Protein sequence profile prediction using ProtAlbert transformer[1]

Armin Behjati[1], Fatemeh Zare-Mirakabad[1*], Seyed Shahriar Arab[2], Abbas Nowzari-Dalini[3]

[1]Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran.
[2]Department of Biophysics, Faculty of Biological Sciences, Tarbiat Modares University, Tehran, Iran.
[3]Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran

* Corresponding author: f.zare@aut.ac.ir

Abstract

Protein sequences can be viewed as a language; therefore, we benefit from using the models initially developed for natural languages such as transformers. ProtAlbert is one of the best pre-trained transformers on protein sequences, and its efficiency enables us to run the model on longer sequences with less computation power while having similar performance with the other pre-trained transformers. This paper includes two main parts: transformer analysis and profile prediction. In the first part, we propose five algorithms to assess the attention heads in different layers of ProtAlbert for five protein characteristics, nearest-neighbor interactions, type of amino acids, biochemical and biophysical properties of amino acids, protein secondary structure, and protein tertiary structure. These algorithms are performed on 55 proteins extracted from CASP13 and three case study proteins whose sequences, experimental tertiary structures, and HSSP profiles are available. This assessment shows that although the model is only pre-trained on protein sequences, attention heads in the layers of ProtAlbert are representative of some protein family characteristics. This conclusion leads to the second part of our work. We propose an algorithm called PA_SPP for protein sequence profile prediction by pre-trained ProtAlbert using masked-language modeling. PA_SPP algorithm can help the researchers to predict an HSSP profile while there are no similar sequences to a query sequence in the database for making the HSSP profile.

Keywords:

Transformer Analyzing, HSSP profile, nearest-neighbor interactions, biochemical and biophysical properties of amino acids, protein secondary structure, protein tertiary structure.

1. Introduction

Proteins consist of linear chains of twenty types of amino acids, each with different chemical properties. Proteins are the most versatile organic molecules in cells or living organisms and play critical roles in the body. The diversity of proteins functions is generally related to their diverse structures. The sequence of amino acids determines a unique protein tertiary structure which directly impacts its specific function[1].
New sequencing technologies have led to an explosion in generating biological data such as protein sequences in the past two decades. UniProt[2] Archive and Swiss-Prot[3] databases contain most of the publicly available protein sequences globally. These sequences grow exponentially every few years[2]. Despite the strong interest in protein structure determination, there is currently a massive gap between the number of known sequences and experimentally determined structures deposited in the Protein Data Bank[4] (PDB), highlighting the difficulties of structure elucidation[5]. Therefore, computationally predicting protein

[1] This article was submitted to "proteins-structure function and bioinformatics" journal: 18 September 2021

structure from the query sequence remains to be largely unsolved[6 7 8]. Homology modeling is a common approach for protein structure prediction. In this approach, homologous proteins of the query sequence are found by sequence comparison in a database. Then, a sequence profile is created to show the conservative and non-conservative regions in the homologous sequences [9 10].

Profiles are used in many bioinformatics problems. For example, they are applied to model protein families[11], predict protein domains[12], detect protein homology[13 14], design proteins[15 16], and identify orthologous genes and proteins[17]. Homology-derived Secondary Structure of Proteins (HSSP) database includes a sequence profile for each PDB protein. In HSSP, a Multiple Sequence Alignment (MSA) of putative homologs is prepared to construct a profile for each PDB protein. The list of homologous sequences is the result of an iterative database search in Swiss-Prot[18]. A well-defined profile can group information of similar sequences on conserved regions. It helps us to assign a query sequence to the family. This assignment is challenging when the query sequence length is short, and there is little similarity between this sequence and any sequences in the profile.

Protein structures are more conserved than protein sequences. Homologous proteins sharing a common evolutionary ancestor can have high sequence-level variations[19], and when the protein sequence similarity is below 30% at the amino acid level, the alignment score usually falls into a twilight zone[20 21]. Therefore, simply comparing sequence similarities often fails to capture global structural and functional similarities of proteins.

Concerning the above discussion, improving the profile prediction methods to get more information about the sequence and families is an active research area in bioinformatics. In this paper, our primary goal is to predict a profile for query protein sequence using transformers.

In the following, we review the transformer-based models processing protein sequences. Proteins, as a linear chain of amino acids, can be viewed precisely as a language. Therefore, they can be modeled using Language Models (LMs) taken from Natural Language Processing (NLP). These LMs are used for biology identity representation and new prediction tools in various bioinformatics problems. The central concept behind this approach is to interpret protein sequences as sentences of characters (amino acids) and each character as a single word[22 23 24]. Recent research has shown that contextualized representations in NLP work well for contextual protein representation learning[25 26]. In the training phase, LMs learn to extract useful features from many samples and generate appropriate representations of these features [27 28 29 30]. In these papers, architectures inspired by NLP are employed for protein processing. Also, pre-training tasks such as Masked-Language Modeling (MLM) and autoregressive generation are utilized to investigate protein-specific pre-training tasks.

One of the latest architectures that showed significant superiority over previous models is transformers[31]. Devlin et al.[32] introduced a new language representation model based on transformers called Bidirectional Encoder Representations from Transformers (BERT). This model is designed to pre-train deep bidirectional representations from unlabeled text to create state-of-the-art models for a wide range of tasks. Bepler and Berge[33] proposed a framework for mapping any protein sequence to a sequence of vector embeddings that encode structural information. Also, they defined a novel similarity measure between these arbitrary length vectors to learn useful position-specific embeddings. Similarly, Alley et al.[34] used a Recurrent Neural Network (RNN) named UniRep to learn statistical representations of proteins and demonstrated that such representations predict the stability of natural and de novo designed proteins, as well as the quantitative function of molecularly diverse mutants. Rao et al.[35] introduced TAPE as a new benchmark consisting of five relevant semi-supervised tasks for assessing such protein representation.

Elnaggar et al.[29] trained two auto-regressive language models (Transformer-XL, XLNet) and two auto-encoder models (BERT, ALBERT) on data extracted from UniProt Reference Clusters (UniRef) datasets and Big Fat Database (BFD). They showed the effects of these pre-training models upon the success of the subsequent supervised training for predicting secondary structure, subcellular localization, and membrane-bound or water-soluble protein problems. Lu et al.[36] applied the principle of mutual information maximization between local and global information as a self-supervised pre-training signal for protein embeddings to introduce a contrastive loss that trains an RNN to discriminate fragments from a source sequence versus randomly sampled fragments from other sequences. Min et al.[37] introduced a novel pre-training scheme for protein sequence modeling called PLUS consisting of masked language modeling and a complementary protein-specific pre-training task, namely same-family prediction. They showed the advances of the PLUS on six out of seven protein biology tasks. Sturmfels et al.[38] introduced a new pre-training task for protein sequence models. They used profile-hidden Markov models derived from MSAs as labels during pre-training for profile prediction. They utilized the model on a set of five downstream tasks for protein modeling and demonstrated that the model outperforms masked language modeling alone on all five tasks.

Although most previous studies on using transformer models for embedding protein sequences in different bioinformatics problems show acceptable results, they apply the model as a black box.

Here, we analyze heads in layers of a pre-trained transformer on protein sequences to find representative heads for some protein characteristics. The results of the analyses lead us to propose an algorithm for protein sequence profile prediction.

At the first step, we select pre-trained ProtAlbert, because its efficiency enables us to run the model on longer sequences with less computation power while having similar performance with the other pre-trained transformers. Then, we propose five algorithms called RLH_NNI, RH_SAA, RH_BBP, RH_PSS, and RH_PTS to analyze five protein characteristics, nearest-neighbor interactions, type of amino acids, biochemical and biophysical properties of amino acids, protein secondary structure, and protein tertiary structure at attention heads in the layers of ProtAlbert.

For this assessment, we make a dataset by extracting 55 proteins from CASP13 which their sequences, experimental tertiary structures, and HSSP profiles are available. In addition, we perform our analysis on three proteins to show no difference between the average result of CASP13 and case studies.

After executing each of the transformer head analyzer algorithms, we reach the following results:

- RLH_NNI[i] algorithm detects representative heads in the layers of the ProtAlberl model for interaction between amino acids located at k $(1 \leq k \leq 5)$ distances on the protein sequence.
- RH_SAA[ii] algorithm finds specific heads for aspartic acid, glutamic acid, proline, tryptophan, and histidine.
- RH_BBP[iii] algorithm announces representative heads for amino acids classified based on R-group.
- RH_PSS[iv] algorithm identifies some heads which contain significant attention weights from helix to helix, coil to coil, and sheet to sheet.
- RH_PTS[v] algorithm finds a representative head for protein contact map, which is a simple tertiary structure representation.

Generally, these analyses show the representative heads of the pre-trained ProtAlbert on protein sequences to detect protein family features. So, we propose an algorithm called PA_SPP[vi] for sequence profile prediction by pre-trained ProtAlbert on protein sequences using MLM. Next, the predicted profiles are compared to the HSSP profiles. The result shows the high similarity between the predicted and HSSP

profiles. PA_SPP algorithm can help the researchers to predict a profile similar to the HSSP profile while there are no similar sequences to the query sequence in the database for making the HSSP profile.

2. Material and Method

This section first introduces the basic definitions needed to interpret ProtAlbert as a transformer model. Next, we propose five algorithms for assessing the layers and heads of ProtAlbert to identify some protein characteristics. Then, our approach is illustrated for the sequence profile prediction problem in more detail. In the end, we introduce the dataset used for evaluation.

2.1 Notation and Definition

The sequence of protein P with length n is represented by:

$$S^P = s_1^P \ldots s_n^P, \quad s_i^P \in AA, \quad |S^P| = n,$$

where $AA = \{a_1, \ldots, a_{20}\}$ shows the set of amino acids. We define amino acid s_{i+k}^P as a k-neighbor of s_i^P in sequence S^P. The positive (negative) value of k shows that the position i attends from left to right (from right to left) of the sequence to find the neighboring amino acid at distance k.

In protein folding, the sidechain backbone of nearest-neighbor interactions may restrict the accessible conformations to a chain of protein[39]. Neighboring amino acids can be structurally categorized according to their separation in the primary sequence as proximal (1-4 positions apart) and otherwise distal[40]. For each protein P, the k-neighbor interaction is defined based on the interaction of each position i with position $i + k$ on the sequence S^P.

In addition to the effect of the nearest neighbor amino acids on protein folding, each amino acid has different biochemical and biophysical properties that can effectively determine the protein structure. Amino acids are classified based on R-group[vii] into five classes $\mathbb{C} = \{N, H, U, A, B\}$(see Table 1).

Table 1: Classification of amino acids based on R-group: $\mathbb{C} = \{N, H, U, A, B\}$.

Name of class	Amino acids	Biochemical and biophysical properties
N	{G,A,V,L,I,P,M}	Hydrophobic, Nonpolar, Aliphatic
H	{F,Y,W}	Hydrophobic, Aromatic
U	{S,T,C,N,Q}	Hydrophilic, Uncharged, Polar,
A	{D,E}	Acidic, Negatively charged
B	{R,H,K}	Basic, Positively charged

The experimental structure of proteins can be extracted from PDB[viii]. Therefore, the 3D coordinate of each atom of amino acids in the protein sequence is available. Here, we represent the tertiary structure of protein P with length n, by contact map $D_{n\times n}^P$ as follows:

$$D^P[i,j] = \begin{cases} 1 & dis\left(\left(x_i^P, y_i^P, z_i^P\right), \left(x_j^P, y_j^P, z_j^P\right)\right) \le \theta, \\ 0 & otherwise, \end{cases} \quad (1)$$

where $dis(.,.)$ is the Euclidian distance and $\left(x_i^P, y_i^P, z_i^P\right)$ shows the 3D coordinate of the atom c_α for amino acid s_i^P at position i of protein P. The value of θ is set 4.87 based on paper[41]. Each element $D^P[i,j]$ with value 1 indicates that two amino acids s_i^P and s_j^P are in contact.

The secondary structure of protein P is extracted from the tertiary structure using DSSP[ix] software. This method provides eight classes, 3-helix, 4-helix, 5-helix, β-strand, β-bridge, turn, bend, and coil. Typically, the DSSP states are converted into three classes using the following convention. 3-helix, 4-helix, and 5-helix are considered helix (H). β-strand and β-bridge are displayed by a sheet (E). The rest of the states are shown as a coil (C). The secondary structure of protein P with length n is displayed as follows:

$$E^P = e_1^P \ldots e_n^P, \qquad e_i^P \in \{H, E, C\}, \qquad |E^P| = n.$$

As mentioned in[42], the secondary structure of each position in the protein sequence is dependent on its neighbors. The length of each type of regular secondary structure[43] is about 6. We define a secondary structure matrix named $H^P_{n\times n}$ on protein P with length n as follows:

$$H^P[i,j] = \begin{cases} 1 & |i-j| \le 6 \ \ \& \ \ e_i^P = e_j^P, \\ 0 & otherwise, \end{cases} \qquad (2)$$

where $H^P[i,j] = 1$ indicates the same secondary structure between two amino acids s_i^P and s_j^P with distance less than 7 in sequence S^P.

For each protein P with length n in PDB database, a profile named $R^P_{n\times 20}$ is extracted from the HSSP database[18]. In this database, there is an MSA of all available homologous sequences properly aligned to protein sequence S^P. This MSA is constructed based on searching in the Swiss-Prot database considering the sequence family and structure. Each sequence of MSA is more than 30% identical to S^P. Using MSA, the profile $R^P_{n\times 20}$ is generated where $R^P[i,j]$ shows the probability of amino acid $a_j \in AA$ at position i of MSA.

In the following, we assume that dataset $\Delta = \{P_1, \ldots, P_t\}$ includes t proteins where their sequences, experimental tertiary structures, and HSSP profiles are available.

2.2 ProtAlbert as a pre-trained transformer model on protein sequences

As described earlier, protein sequences can be viewed as a language, and therefore, we can benefit from using the models initially developed for natural languages. One of the latest architectures that showed significant superiority over previous models is transformers.

As it was mentioned, BERT[32]is a method of pre-training language representations. It means that after training a general-purpose language understanding model on a large corpus of text, the model can be used on downstream tasks. BERT is an example of auto encoding language modeling trained using MLM. During the training, 15% of the input is randomly masked, and the model is asked to predict the masked tokens. This process lets the model predicts the masked tokens based on the other available tokens. It shows that the model has a good idea about the language and the context. This self-supervised pre-training method, which means the labels are in the training corpus, got better results in many downstream tasks.

A year after BERT[32], ALBERT[44] was released by Google research that improved state-of-the-art performance in 12 NLP tasks. The main idea in the ALBERT was to allocate capacity more efficiently. They made two design changes to BERT, but the training process was MLM. First, while the input level embeddings need to be context-independent representations, the hidden-state embeddings need to take context into account. This was addressed by splitting the embedding matrix between a low dimension input-level embedding

with length 128 and a higher dimension hidden-layer embedding with size 4096. The second critical change was removing redundancy and therefore increasing the capacity of the model to learn. Previously, it was observed that the various layers of BERT with different parameters in the model learned similar operations. This possible redundancy was eliminated in ALBERT by parameter sharing in different layers. These two design changes resulted in 90% parameter reduction compared to BERT with slightly decreased accuracy. However, this reduction allows scaling the hidden size from 768 in BERT to 4096 in ALBERT. It is shown that the bigger hidden layer embeddings can capture and represent the context better[44].

We base our experiments on ProtAlbert, a transformer-based model on ALBERT architecture from the ProtTrans project[29]. ProtAlbert is pre-trained on 216 million protein sequences from the UniRef100 dataset. In this paper, we do not train or fine-tune the model. In the ProtAlbert model, the protein sequences are tokenized using a single space between each amino acid (indicating words), and each sequence is stored in a separate line (indicating sentences). Also, all non-generic or unresolved amino acids (B,O,U,Z) are mapped to the unknown token X. This model can process sequences with lengths of up to 40K, although this length is bound by the hardware capacity. The details of the ProtAlbert model are available in Table 2.

Table 2: ProtAlbert Parameters.

Hyperparameter	ProtAlbert
Dataset	UniRef100
Number of Layers	12
Hidden Layers Size	4096
Hidden Layers Intermediate Size	16384
Number of Heads	64
Positional Encoding Limits	40K
Target Length	512/2048

Our work contains two main parts, transformer analysis, and profile prediction. For the first part, a protein sequence is given as an input to the ProtAlbert transformer. Then, we analyze and interpret the attention weights at attention heads in different layers. In the second part, protein profile is predicted using ProtAlbert and masked token prediction. In other words, a protein sequence with some masked amino acids is fed to the model for predicting the most likely amino acids in the masked positions.

We choose ProtAlbert[29] because its efficiency enables us to run the model on longer sequences with less computation power while having similar performance with ProtBert[29], which is a great advantage. The ProtBert model is a pre-trained BERT-based language model with 420M parameters from the ProTrans project that has been trained on the same dataset as the ProtAlbert model with 224M parameters.

2.3 Proposed algorithms for analyzing ProtAlbert transformer to identify protein characteristics

In this sub-section, we propose five algorithms to analyze the attention heads and layers of ProtAlbert for finding the specific properties of proteins (see Table 3). This analysis is essential because it shows that ProtAlbert transformer can learn some biological features from only protein sequences. It allows us not to apply the transformer as a black box but to select the ProtAlbert features specific to the bioinformatics problems.

Table 3: Five protein characteristics.

Nearest-neighbor interaction
Type of amino acids
biochemical and biophysical properties of amino acids
Protein secondary structure
Protein tertiary structure

The input and output of this assessment are defined as follows:

- **Input**: Sequence $S^P = s_1^P \ldots s_n^P$ of protein P.
- **Output:** Extracting attention matrix $A_{n\times n}^{P,l,h}$ from ProtAlbert for each head h in layer l to interpret the properties of protein P displayed in Table 3.

ProtAlbert includes 12 encoder layers, and each encoder has 64 attention heads. Each protein sequence $S^P = s_1^P \ldots s_n^P$ is given to the model as an input, then it goes through the encoder layers, and the attention mechanism in each layer generates output to go to the next layer.

For the input sequence S^P with length n, each attention head h ($1 \le h \le 64$) in the layer l ($1 \le l \le 12$) produces a matrix of positive attention weights named $A_{n\times n}^{P,l,h}$. The value $A^{P,l,h}[i,j]$ shows the attention weight from amino acid s_i^P to s_j^P and $\sum_{j=1}^{n} A^{P,l,h}[i,j] = 1$. So, each amino acid at head h in layer l can attend to all other amino acids in the sequence, but the level of the attention is determined by the $A^{P,l,h}[i,j]$.

Based on the attention matrix $A_{n\times n}^{P,l,h}$, adjacency matrix $M_{n\times n}^{P,l,h}$ is constructed as follows:

$$M^{P,l,h}[i,j] = \begin{cases} 1 & A^{P,l,h}[i,j] \ge \theta, \\ 0 & otherwise, \end{cases} \quad (3)$$

where the value of θ is determined by its application. We use attention and adjacency matrices for introducing our approaches to quantify representative heads in layers for some protein features (see Table 3).

2.3.1 RHL_NNI algorithm to quantify the representative heads and layers of ProtAlbert for nearest-neighbor interaction

We propose the RHL_NNI algorithm to determine if head h in layer l of the ProtAlbert model represents the interaction of k-neighbor amino acids in dataset Δ . The main steps of this algorithm are defined as follows:

1. For each protein $P \in \Delta = \{P_1, \ldots, P_t\}$,
   i. The ***interaction of k-neighbor amino acids*** from the sequence S^P ($|S^P| = n$) is quantified, as:

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad T^{P,l,h}[k] = \begin{cases} \sum_{i=1}^{n-k} M^{P,l,h}[i,i+k] & 1 \le k \le n-1, \\ \sum_{i=1-k}^{n} M^{P,l,h}[i,i+k] & -(n-1) \le k \le -1, \end{cases} \quad (4)$$

   where adjacency matrix $M^{P,l,h}$ is generated based on Eq.3 for each head h in layer l .

   ii. The ***normalized k-neighbor interaction*** is defined like this:

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad -(n-1) \le k \ne 0 \le n-1,\quad N^{P,l,h}[k] = \frac{T^{P,l,h}[k]}{n - |k|},$$

where $|.|$ shows the absolute function. For each head h in layer l, $N^{P,l,h}[k]$ indicates the percentage of positions in protein P which attend to the k^{th} amino acid in the neighbor.

iii. The ***weighted quantification of k-neighbor interaction*** is computed based on attention matrix, as :

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad W^{P,l,h}[k] = \frac{1}{T^{P,l,h}[k]} \begin{cases} \sum_{i=1}^{n-k} M^{P,l,h}[i, i+k] A^{P,l,h}[i, i+k] & 1 \le k, \\ \sum_{i=1-k}^{n} M^{P,l,h}[i, i+k] A^{P,l,h}[i, i+k] & k \le -1. \end{cases} \quad (5)$$

2. The ***average of normalized k-neighbor interaction*** is computed on dataset Δ, as:

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad -(n-1) \le k \ne 0 \le n-1,\quad \bar{N}^{l,h}[k] = \frac{1}{|\Delta|} \sum_{P \in \Delta} N^{P,l,h}[k].$$

3. The maximum interaction value of neighbor amino acids at each head in the layer is computed to determine the ***nearest neighbor radius for interaction***, as:

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad k_{max}^{l,h} = \underset{-(n-1) \le k \ne 0 \le n-1}{argmax} \bar{N}^{l,h}[k].$$

4. For each $1 \le l \le 12$ and $1 \le h \le 64$, if $\bar{N}^{l,h}\left[k_{max}^{l,h}\right] \ge \theta$,

i. Head h in layer l is announced ***representative for $k_{max}^{l,h}$-neighbor interaction*** on dataset Δ.

ii. For head h in layer l, the ***average of weighted quantification of $k_{max}^{l,h}$-neighbor interaction*** is computed on dataset Δ, as:

$$\forall\ 1 \le l \le 12,\quad 1 \le h \le 64,\quad \bar{W}^{k_{max}^{l,h}} = \frac{1}{|\Delta|} \sum_{P \in \Delta} W^{P,l,h}\left[k_{max}^{l,h}\right].$$

The average of weighted quantification ($\bar{W}^{k_{max}^{l,h}}$) and the average of normalized interaction ($\bar{N}^{l,h}[k_{max}^{l,h}]$) is compared to show the effect of discretizing of attention weights in the adjacency matrix.

2.3.2 RH_SAA algorithm to quantify representative heads of ProtAlbert for specific amino acids

Here, we introduce the RH_SAA algorithm to investigate if the head h of ProtAlbert attends significantly to a specific amino acid in the protein dataset Δ. To quantify the quality of attention head h for amino acid $a \in AA$, we apply the ***F-measure criterion*** to evaluate the occurrence rate of amino acid a versus the rest in this head. In the following, this algorithm is described in more detail:

1. For each protein $P \in \Delta = \{P_1, \ldots, P_l\}$,

i. True positive $TP_a^{P,h}$ is defined based on the number of amino acid a in protein sequence $S^P = s_1^P \ldots s_n^P$ which is attended by at least one position of the sequence at head h in at least one layer:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad TP_a^{P,h} = \sum_{j=1}^{n} z_a^h[j],$$

where

$$z_a^h[j] = \begin{cases} 1 & \exists\ 1 \le l \le 12,\ \ 1 \le i \le n,\ \ s_j^P = a,\ \ M^{P,l,h}[i,j] = 1, \\ 0 & otherwise, \end{cases} \quad (6)$$

where adjacency matrix $M^{P,l,h}$ is generated based on Eq.3 for head h in layer l.

ii. False positive, $FP_a^{P,h}$, is obtained as follows:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad FP_a^{P,h} = \sum_{q \in AA - \{a\}} TP_q^{P,h},$$

where $TP_q^{P,h}$ represents the number of amino acid $q \neq a$ attended by at least one position of the sequence at head h in at least one layer.

iii. False negative, $FN_a^{P,h}$, is computed as follows:

$$\forall\ 1 \le h \le 64,\ \ a \in AA,\ \ FN_a^{P,h} = \mathrm{B}_a^P - TP_a^{P,h},$$

where B_a^P shows the frequency of amino acid $a \in AA$ in sequence S^P.

iv. ***F-measure criterion***, $\mathcal{F}_a^{P,h}$, is computed to quantify head h for amino acid a:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad \mathcal{F}_a^{P,h} = \frac{2 \times TP_a^{P,h}}{2 \times TP_a^{P,h} + FP_a^{P,h} + FN_a^{P,h}}.$$

v. The ***relative occurrence of amino acid*** a of protein P in head h is computed as:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad \varphi_a^{P,h} = \frac{TP_a^{P,h}}{B_a^P}.$$

vi. The ***weighted occurrence of amino acid*** a of protein P is calculated as follows:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad \omega_a^{P,h} = \sum_{l=1}^{12} \sum_{j=1}^{n} \sum_{i=1}^{n} M^{P,l,h}[i,j] A^{P,l,h}[i,j] \chi_a^{s_j^P}, \quad (7)$$

where

$$\chi_g^f = \begin{cases} 1 & f = g, \\ 0 & otherwise. \end{cases}$$

vii. The ***normalized weighted occurrence of amino acid*** a of protein P is calculated as follows:

$$\forall\ 1 \le h \le 64,\quad a \in AA,\quad W_a^{P,h} = \frac{\omega_a^{P,h}}{\sum_{q \in AA} \omega_q^{P,h}}.$$

2. The ***average of F-measure*** is computed on the dataset Δ :

$$\forall a \in AA,\quad \bar{\mathcal{F}}_a^h = \frac{1}{|\Delta|} \sum_{P \in \Delta} \mathcal{F}_a^{P,h}.$$

3. The ***candidate representative head*** for amino acid a is computed, as:

$$\forall a \in AA,\quad h_{max}^a = \underset{1 \le h \le 64}{argmax}\ \bar{\mathcal{F}}_a^h.$$

4. For each $a \in AA$, if $\bar{\mathcal{F}}_a^{h_{max}^a} \ge \theta$:

i. Head h^{a}_{max} is announced as a ***representative head*** for amino acid a.

ii. In head h^{a}_{max}, the ***average of normalized weighted occurrence of amino acid*** a is computed on dataset Δ, as:

$$\forall a \in AA, \quad \bar{W}^{h^{a}_{max}} = \frac{1}{|\Delta|} \sum_{P \in \Delta} W_{a}^{P, h^{a}_{max}}.$$

where the normalized weighted occurrence of amino acid a shows the effect of attention weights attending from each amino acid to a at head h^{a}_{max}.

iii. The ***average of the relative occurrence of amino acid*** a at head h^{a}_{max} is computed on dataset Δ as:

$$\forall a \in AA, \quad \bar{\varphi}^{h^{a}_{max}} = \frac{1}{|\Delta|} \sum_{P \in \Delta} \varphi_{a}^{P, h^{a}_{max}}.$$

where the relative occurrence of amino acid a shows the probability of amino acid a detection at head h^{a}_{max}.

2.3.3 RH_BBP algorithm to quantify representative heads of ProtAlbert for biochemical and biophysical properties of amino acids

In this sub-section, we illustrate the RH_BBP algorithm to find representative heads of ProtAlbert on the biochemical and biophysical properties using the classification of amino acids based on the R-group. Table 1 shows this classification, $\mathbb{C} = \{N, H, U, A, B\}$. The algorithm is very similar to RH_SAA, which identifies specific heads for amino acids. In the following, the details of RH_BBP are available:

1. For each protein $P \in \Delta = \{P_1, \ldots, P_l\}$,

   i. For class $C \in \mathbb{C}$, true positive is defined by $TP_{C}^{P,h}$ to show the number of amino acids from class C in protein sequence S^P ($|S^P| = n$) attended by at least one position of the sequence at head h in at least one layer:

$$\forall\ 1 \le h \le 64, \quad C \in \mathbb{C}, \quad TP_{C}^{P,h} = \sum_{j=1}^{n} z_{C}^{h}[j],$$

   where

$$z_{C}^{h}[j] = \begin{cases} 1 & \exists\, 1 \le l \le 12, \quad 1 \le i \le n, \quad s_{j}^{P} \in C, \quad M^{P,l,h}[i,j] = 1, \\ 0 & otherwise. \end{cases} \qquad (8)$$

   ii. For class $C \in \mathbb{C}$, false positive, $FP_{C}^{P,h}$, is obtained as follows:

$$\forall\ 1 \le h \le 64, \quad C \in \mathbb{C}, \quad FP_{C}^{P,h} = \sum_{q \in \mathbb{C} - \{C\}} TP_{q}^{P,h},$$

   where $TP_{q}^{P,h}$ represents the number of amino acids from class $q \neq C$ attended by at least one position of the sequence S^P at head h in at least one layer.

   iii. For class $C \in \mathbb{C}$, false negative, $FN_{C}^{P,h}$, is computed as follows:

$$\forall\ 1 \le h \le 64, \quad C \in \mathbb{C}, \quad FN_{C}^{P,h} = B_{C}^{P} - TP_{C}^{P,h},$$

where B_C^P shows the frequency of the amino acids from class C in sequence S^P.

iv. For class $C \in \mathbb{C}$, ***F-measure criterion***, $\mathcal{F}_C^{P,h}$, is computed to quantify head h at this class:

$$\forall\, 1 \le h \le 64, \quad C \in \mathbb{C}, \quad \mathcal{F}_C^{P,h} = \frac{2 \times TP_C^{P,h}}{2 \times TP_C^{P,h} + FP_C^{P,h} + FN_C^{P,h}}.$$

v. The ***relative occurrence of class*** C for protein P in head h is computed as:

$$\forall\, 1 \le h \le 64, \quad C \in \mathbb{C}, \quad \varphi_C^{P,h} = \frac{TP_C^{P,h}}{B_C^P}.$$

vi. The ***weighted occurrence of class*** C for protein P is calculated as follows:

$$\forall\, 1 \le h \le 64, \quad C \in \mathbb{C}, \quad \omega_C^{P,h} = \sum_{l=1}^{12} \sum_{j=1}^{n} \sum_{i=1}^{n} M^{P,l,h}[i,j] A^{P,l,h}[i,j] \chi_C^{s_i^P}, \quad (9)$$

where

$$\chi_g^f = \begin{cases} 1 & f \in g, \\ 0 & otherwise. \end{cases}$$

vii. The ***normalized weighted occurrence of class*** C for protein P is calculated as follows:

$$\forall\, 1 \le h \le 64, \quad C \in \mathbb{C}, \quad W_C^{P,h} = \frac{\omega_C^{P,h}}{\sum_{q \in \mathbb{C}} \omega_q^{P,h}}.$$

2. The ***average of F-measur***e is computed on dataset Δ :

$$\forall\, C \in \mathbb{C}, \quad \bar{\mathcal{F}}_C^{h} = \frac{1}{|\Delta|} \sum_{p \in \Delta} \mathcal{F}_C^{P,h}.$$

3. The ***candidate representative head*** for class C is calculated, as:

$$\forall\, C \in \mathbb{C}, \quad h_{max}^C = \underset{1 \le h \le 64}{argmax} \bar{\mathcal{F}}_C^{h}.$$

4. For each $C \in \mathbb{C}$, if $\bar{\mathcal{F}}_C^{h_{max}^C} \ge \theta$:

i. Head h_{max}^C is announced as a ***representative head*** for class C.

ii. In head h_{max}^C, the ***average of normalized weighted occurrence of class*** C is computed on dataset Δ , as:

$$\forall\, C \in \mathbb{C}, \quad \bar{W}^{h_{max}^C} = \frac{1}{|\Delta|} \sum_{P \in \Delta} W_C^{P,h_{max}^C}.$$

where the normalized weighted occurrence of class C shows the effect of attention weights attending from each amino acid to the amino acids in class C at head h_{max}^a.

*iii. The **average of the relative occurrence of class** C at head h_{max}^C on dataset Δ is computed as:*

$$\forall\, C \in \mathbb{C}, \quad \bar{\varphi}^{h_{max}^C} = \frac{1}{|\Delta|} \sum_{P \in \Delta} \varphi_C^{P,h_{max}^C}.$$

where the relative occurrence of class C shows the probability of class C detection at head h_{max}^C.

2.3.4 RH_PSS algorithm to quantify representative heads of ProtAlbert for protein secondary structure

Although ProtAlbert only has been pre-trained on protein sequences, we propose the RH_PSS algorithm on dataset Δ to assess attention heads about the protein secondary structure matrix (see Eq.2). The detail of this algorithm is as below:

1. For each protein $P \in \Delta = \{P_1, \ldots, P_l\}$,
   i. Predicting the secondary structure matrix $\mathscr{H}^{P,h}_{n\times n}$, of protein P with length n for each head h as follows:

$$\forall\ 1 \le h \le 64,\quad \mathscr{H}^{P,h}[i,j] = \begin{cases} 1 & \exists\, 1 \le l \le 12,\quad M^{P,l,h}[i,j] = 1, \\ 0 & otherwise, \end{cases} \tag{10}$$

   where adjacency matrix $M^{P,l,h}$ is constructed based on Eq.3 for each head h in layer l.
   ii. Making the natural secondary structure $H^{P}_{n\times n}$ for protein P based on Eq.2.
   iii. Computing the cosine similarity between $\mathscr{H}^{P,h}_{n\times n}$ and $H^{P}_{n\times n}$ for each h, $1 \le h \le 64$, $\cos(\mathscr{H}^{P,h}, H^{P})$.
2. The average of cosine similarity is computed as:

$$\forall\ 1 \le h \le 64,\qquad \bar{C}^{h} = \frac{1}{|\Delta|}\sum_{P\in\Delta} \cos(\mathscr{H}^{P,h}, H^{P}).$$

2.3.5 RH_PTS algorithm to quantify representative heads of ProtAlbert for protein tertiary structure

We propose the RH_PTS algorithm to compare the natural protein contact map to the predicted contact map from head h on dataset Δ. The main steps of this algorithm are as follows:

1. For each protein $P \in \Delta = \{P_1, \ldots, P_l\}$,
   i. Making matrix $\mathfrak{D}^{P,h}_{n\times n}$ as:

$$\forall\ 1 \le h \le 64,\quad \mathfrak{D}^{P,h}[i,j] = \sum_{l=1}^{12} A^{P,l,h}[i,j],$$

   where $A^{P,l,h}$ represents the attention matrix of protein P in layer l and head h.
   ii. Normalizing matrix $\mathfrak{D}^{P,h}_{n\times n}$ as bellow:

$$\mathbb{D}^{P,h}[i,j] = \frac{\mathfrak{D}^{P,h}[i,j] - min}{max - min},\quad min = \min_{1\le i,j\le n} \mathfrak{D}^{P,h}[i,j],\quad max = \max_{1\le i,j\le n} \mathfrak{D}^{P,h}[i,j].$$

   iii. Predicting contact map based on matrix $\mathbb{D}^{P,h}_{n\times n}$, as:

$$\mathscr{D}^{P,h}[i,j] = \begin{cases} 1 & \mathbb{D}^{P,h}[i,j] \ge \theta, \\ 0 & otherwise. \end{cases} \tag{11}$$

   iv. Making real contact map $D^{P}_{n\times n}$ for protein P based on Eq.1.
   v. Computing the cosine similarity between $\mathscr{D}^{P,h}_{n\times n}$ and $D^{P}_{n\times n}$ for each$1 \le h \le 64$, $\cos(\mathscr{D}^{P,h}, D^{P})$.
2. Computing the average of cosine similarity as:

$$\forall\ 1 \le h \le 64,\quad \bar{C}^{h} = \frac{1}{|\Delta|}\sum_{P \in \Delta} \cos\left(\mathcal{D}^{P,h}, D^{P}\right).$$

3. Finding head h_{max} to indicate the maximum similarity between natural and predicted contact maps:

$$h_{max} = \underset{1 \le h \le 64}{argmax}\, \bar{C}^{h},$$

where head h_{max} is known as a representative head for contact maps.

2.4 Proposed algorithm for sequence profile prediction problem

In the second part of our work, we propose the PA_SPP algorithm for the sequence profile prediction problem. The input and output of this problem are defined as follows:

- **Input**: Sequence $S^P = s_1^P \ldots s_n^P$ of protein P.
- **Output:** Predicting profile $\mathfrak{R}^P_{n\times 20}$ using pre-trained ProtAlbert.

To solve this problem, we apply pre-trained ProtAlbert to predict the masked token of an input sequence containing unknown amino acids in one position of the sequence S^P. ProtAlbert model generates the most likely amino acids for that position. In other words, the model predicts the masked amino acid in the sequence based on the context of other amino acids surrounding it. This process is called masked token prediction and represented by

$$(\Upsilon^P, \Pi^P) = Masking\left(s_1^P \ldots s_{i-1}^P [Mask] s_{i+1}^P \ldots s_n^P\right),$$

where generates two vectors $\Upsilon^P = [\gamma_1^P \ldots \gamma_{20}^P]$ and $\Pi^P = [\pi_1^P \ldots \pi_{20}^P]$. Vectors Υ^P and Π^P represent the type of amino acids and the score for each amino acid replaced at the masked position in the sequence S^P. For each $1 \le j \le 20$, π_j^P shows the score of substitution of amino acid γ_j^P at position i of sequence S^P. Figure 1 illustrates the PA_SPP algorithm for solving the profile prediction problem. In the first step, the sequence S^P is given as an input to the algorithm. In the second step, a zero-matrix named $\mathfrak{R}^P_{n\times 20}$ is defined where $\mathfrak{R}^P[i,j]$ is updated during algorithm running by predicting the probability of j^{th} amino acid at the i^{th} position of sequence S^P. The third step selects each position i, $1 \le i \le n$, in sequence S^P for masking. In the fourth step, temporary memory T is defined to keep the sequence S^P with masking position i. In the fifth step, sequence T is fed to $Masking$ process of ProtAlbert. The model generates two vectors Υ^P and Π^P for position i. In the sixth step, we set the probability vector Π^P into the i^{th} row of matrix $\mathfrak{R}$ according to the order of amino acids in Υ^P. In the end, we call $\mathfrak{R}^P_{n\times 20}$ the predicted profile for protein P.

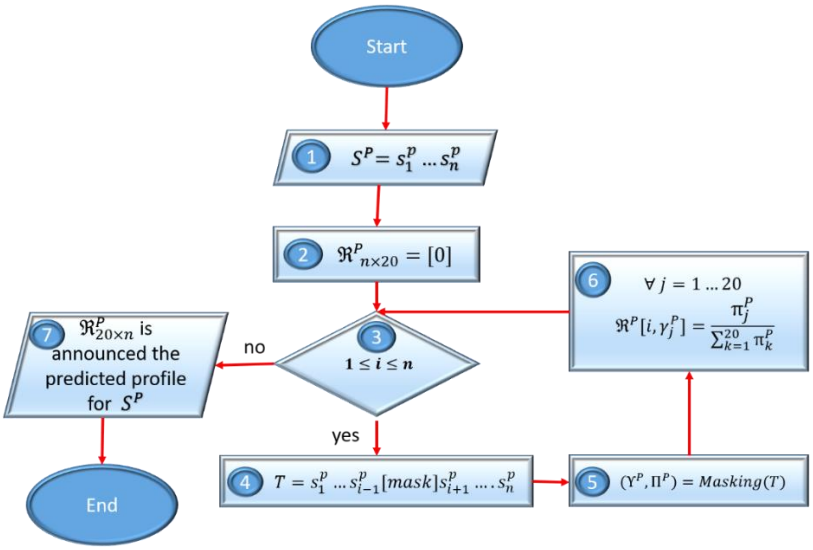


Figure 1: PA_SPP algorithm for protein profile prediction.

2.5 Dataset

In this study, we use the CASP13[x] dataset. This dataset includes 194 proteins. We select 55 proteins (see Supplementary 1) whose profiles are available in the HSSP database. We call the selected proteins from CASP13, dataset Δ where $|\Delta| = 55$. The tertiary structure and sequence of each protein $P \in \Delta$ are extracted from the PDB database. In addition, their HSSP profiles are downloaded from xssp site.

The distribution of the extracted target sequences lengths is shown in Figure 2. In addition, Figure 3 represents the frequency of amino acids in the sequences of dataset Δ.

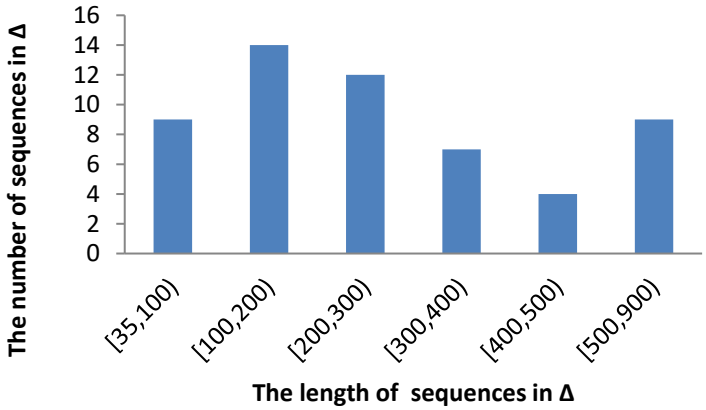


Figure 2: Distribution of the length of protein sequences in Δ.

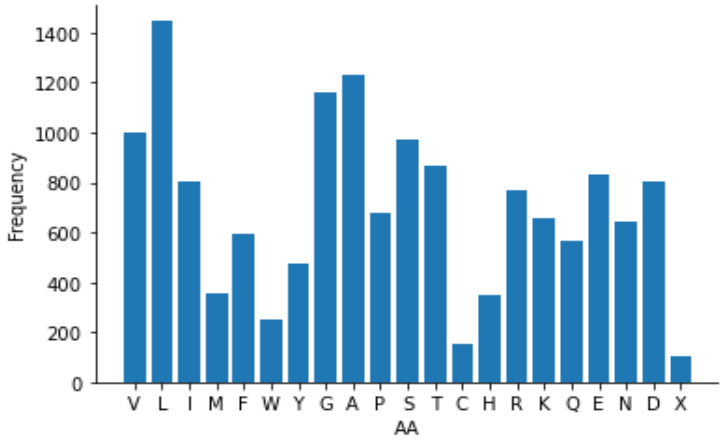


Figure 3: The frequency of amino acids (AA) in the Δ.

In addition to dataset Δ, we select three essential proteins (see Table 4) in different organisms for case studies to show that our result is generally reliable. The details of these proteins are available in Supplementary 2.

Table 4: Details of three case study proteins.

Abb	Protein Name	Chain	Length
LuxB	Alkanal monooxygenase beta chain	A	325
Mpro	Replicase polyprotein 1ab Fragment: 3C-like proteinase (Main protease)	A	306
Taq	Taq DNA polymerase I	A	832

3. Result and Discussion

In this section, we apply $\Delta \subseteq CASP13$ and three case study proteins, LuxB, Mpro, and Taq, to analyze ProtAlbert as a pre-trained transformer on protein sequences. We find representative heads of ProtAlbert for five protein characteristics (Table 3). This part assures us that the heads contain the information required by a family of proteins. Then, we use this dataset for profile prediction. In the end, we compare the predicted profiles to the HSS profiles.

3.1 Analyzing ProtAlbert as a pre-trained transformer on protein sequences

Here, we find representative heads in the layers of ProtAlbert for five protein characteristics displayed in Table 3 using algorithms RLH_NNI, RH_SAA, RH_BBP, RH_PSS, and RH_PTS. In these algorithms, we use some cutoffs obtained by our trial and error. Cutoffs are set high for sequence feature analysis because ProtAlbert has been pre-trained on the protein sequences. For structures feature analysis, cutoffs are set low.

3.1.1 Assessment of nearest-neighbor interactions at heads in layers of ProtAlbert

As mentioned in[40], k-neighbor interaction where $-4 \leq k \neq 0 \leq 4$ is known as proximal interaction, which is effective in the first step of protein folding. Here, we apply RHL_NNI algorithm on dataset $\Delta \subseteq \mathrm{CASP13}$ to find the representative heads in the layers of ProtAlbert for the nearest neighbor radius of amino acids interaction.

In Eq.4 and Eq.5 of this algorithm, we consider threshold 0.5 to make an adjacency matrix from the attention matrix. In the fourth step of RHL_NNI, we select representative head h in layer l for the interaction of $k_{max}^{l,h}$-neighbor amino acids in dataset Δ, if $\bar{N}^{l,h}\left[k_{max}^{l,h}\right] \geq 0.5$. For each selected head h in layer$l$ and protein $P \in \{LuxB,\ Mpro,\ Taq\}$, $N^{P,l,h}\left[k_{max}^{l,h}\right]$ is computed. Table 5 shows the representative heads in layers for the interaction of $k_{max}^{l,h}$-neighbor amino acids. The results show that the average of normalized $k_{max}^{l,h}$-neighbor interaction is close to the normalized $k_{max}^{l,h}$-neighbor interaction on each case study protein. Also, the average of the weighted quantification of $k_{max}^{l,h}$-neighbor interaction, $\bar{\mathrm{W}}^{k_{max}^{l,h}}$is calculated. Also, the weighted quantification for each case study protein P, $W^{P,l,h}\left[k_{max}^{l,h}\right]$, is available in this table. The values of W are close to N ones; it shows that attention weights are high in $k_{max}^{l,h}$-neighbor on the dataset and cases study proteins. The results show that

- head 10 in layers 2-9, head 21 in layers 1-9, and heads 14 and 44 in layer 1 represent interactions at one position apart.
- head 23 in layers 1- 8 and head 33 in layer 1 are specific for interactions at two positions apart.
- heads 3 and 51 in layer 1 indicate interactions between each amino acid and its third neighbor in the sequence.
- head 51 in layers 2 – 8, head 53 in layers 1- 8, head 2 in layers 1-2 represent the interaction between each amino acid and its fourth neighbor in the sequence.
- head 56 in layer 1 is specific for interactions at five positions apart.

In conclusion, we have identified the representative heads in different layers for proximal positions in proteins. According to [40], proximal positions are essential in the first step of protein folding.

Table 5: Representative heads in layers of ProtAlbert for nearest-neighbor interaction.

Nearest-neighbor			Dataset=**Δ**		Protein=LuxB		Protein=Mpro		Protein=Taq	
$k_{max}^{l,h}$	h	l	$\bar{W}^{k_{max}^{l,h}}$	$\bar{N}^{l,h}[k_{max}^{l,h}]$	$W^{P,l,h}[k_{max}^{l,h}]$	$N^{P,l,h}[k_{max}^{l,h}]$	$W^{P,l,h}[k_{max}^{l,h}]$	$N^{P,l,h}[k_{max}^{l,h}]$	$W^{P,l,h}[k_{max}^{l,h}]$	$N^{P,l,h}[k_{max}^{l,h}]$
4	2	1	0.974	0.995	0.974	1	0.975	1.000	0.937	0.944
		2	0.878	0.940	0.889	0.963	0.882	0.954	0.889	0.955
	51	2	0.920	0.958	0.911	0.966	0.915	0.960	0.913	0.930
		3	0.962	0.981	0.961	0.991	0.950	0.987	0.974	0.993
		4	0.970	0.980	0.985	0.997	0.950	0.983	0.984	0.996
		5	0.965	0.980	0.988	0.997	0.940	0.980	0.988	0.998
		6	0.965	0.977	0.986	0.997	0.933	0.990	0.990	0.999
		7	0.970	0.977	0.990	0.997	0.944	0.993	0.992	0.999
		8	0.953	0.974	0.966	0.997	0.944	0.980	0.984	0.998
-3	3	1	0.944	0.997	0.943	1.000	0.940	1.000	0.909	0.954
1	10	2	0.991	0.994	0.993	1	0.992	1.000	0.987	0.998
		3	0.990	0.991	0.994	0.997	0.993	0.993	0.991	0.998
		4	0.989	0.989	0.997	0.997	0.986	0.997	0.992	0.996
		5	0.987	0.987	0.996	0.997	0.985	0.997	0.995	0.998
		6	0.985	0.987	0.995	0.997	0.985	0.997	0.996	0.998
		7	0.985	0.986	0.996	0.997	0.982	0.993	0.997	0.998
		8	0.987	0.986	0.996	0.997	0.983	0.997	0.997	0.998
		9	0.953	0.975	0.919	0.985	0.980	0.993	0.973	0.995
	14	1	0.944	0.997	0.946	1	0.946	1	0.900	0.953
-1	10	1	0.968	0.996	0.970	1.000	0.969	1.000	0.902	0.941
	21	1	0.879	0.997	0.881	1.000	0.873	1.000	0.863	0.960
		2	0.826	0.875	0.804	0.858	0.840	0.859	0.825	0.883
		3	0.856	0.922	0.808	0.867	0.866	0.954	0.848	0.929
		4	0.887	0.953	0.877	0.981	0.850	0.948	0.878	0.972
		5	0.893	0.956	0.924	0.991	0.852	0.957	0.909	0.982
		6	0.895	0.957	0.918	0.985	0.832	0.961	0.926	0.986
		7	0.932	0.972	0.955	0.985	0.870	0.974	0.957	0.987
		8	0.974	0.987	0.985	1.000	0.958	0.987	0.988	0.995
		9	0.951	0.970	0.934	0.985	0.985	0.990	0.972	0.993
	44	1	0.719	0.832	0.713	0.873	0.713	0.839	0.712	0.623
-2	23	1	0.984	0.998	0.986	1.000	0.985	1.000	0.960	0.963
		2	0.859	0.960	0.849	0.957	0.893	0.997	0.847	0.951
		3	0.805	0.912	0.761	0.873	0.851	0.977	0.787	0.901
		4	0.794	0.898	0.769	0.932	0.833	0.970	0.773	0.905
		5	0.775	0.871	0.775	0.926	0.820	0.961	0.758	0.900
		6	0.758	0.827	0.759	0.867	0.779	0.944	0.755	0.878
		7	0.758	0.802	0.754	0.805	0.764	0.938	0.754	0.861
		8	0.726	0.686	0.696	0.570	0.742	0.898	0.729	0.724
2	33	1	0.976	0.997	0.978	1	0.977	1.000	0.944	0.961
3	51	1	0.949	0.996	0.951	1	0.948	0.977	0.921	0.954
-4	53	1	0.978	0.997	0.977	1.000	0.977	1.000	0.951	0.959
		2	0.994	0.996	0.997	1.000	0.998	1.000	0.993	1.000
		3	0.996	0.995	0.998	1.000	0.998	1.000	0.997	1.000
		4	0.995	0.995	0.998	1.000	0.998	1.000	0.998	1.000
		5	0.993	0.994	0.998	1.000	0.995	1.000	0.998	1.000
		6	0.991	0.992	0.997	1.000	0.994	0.993	0.997	1.000
		7	0.989	0.990	0.997	1.000	0.992	0.993	0.997	1.000
		8	0.961	0.980	0.941	1.000	0.982	0.990	0.977	0.998
-5	56	1	0.736	0.985	0.735	1.000	0.731	0.997	0.707	0.740

3.1.2 Assessment of the type of amino acids at heads of ProtAlbert

In this sub-section, we use the RH_SAA algorithm to find a representative head for each amino acid on dataset $\Delta \subseteq CASP13$. In Eq.6 and Eq.7 of this algorithm, we consider threshold 0.4 to make adjacency matrix from attention matrix. In the third step of RH_SAA, we select candidate representative head h^a_{max} for amino acid a. At the fourth step, head h^a_{max} is announced as a representative head for amino acid a, if $\bar{\mathcal{F}}_a^{h^a_{max}} \geq 0.3$. Meanwhile, we compute the F-measure criterion, $\mathcal{F}_a^{P,h^a_{max}}$, for amino acid a in each protein $P \in \{LuxB,\ Mpro,\ Taq\}$ at head h^a_{max}. Table 6 shows that the average of the F-measure is similar to the F-measure of each case study.

Moreover, this table represents the average of the relative occurrence of amino acid a in dataset Δ and each case study protein $P \in \{LuxB,\ Mpro,\ Taq\}$ by $\bar{\varphi}^{h^a_{max}}$ and $\varphi_a^{P,h^a_{max}}$, respectively. In addition, the average of normalized weighted occurrence of amino acid a in dataset Δ and case study protein P are shown by $\bar{W}^{h^a_{max}}$ and $W_a^{P,h^a_{max}}$, respectively. As a result, we find that

- the average of F-measure on the dataset is close to case study ones,
- heads 8 and 18 can support hydrophilic acidic amino acids, aspartic acid (D) and glutamic acid (E),
- heads 13, 20, and 63 are specific for proline (P), tryptophan (W), and histidine (H), respectively.

To better understand the selected heads for specific amino acids, Figure 4 shows the weighted stacking of amino acids ($\bar{W}^{h^a_{max}} \times \bar{\varphi}^{h^a_{max}}$) at attention heads 8, 13, 18, 20, and 63.

Table 6: The representative heads for amino acids found based on F-measure ($\bar{\mathcal{F}}_a^{h^a_{max}} \geq 0.3$).

Head	Amino acid	Dataset= Δ			Protein=LuxB			Protein=Mpro			Protein=Taq		
		$\bar{\mathcal{F}}_a^{h^a_{max}}$	$\bar{\varphi}^{h^a_{max}}$	$\bar{W}^{h^a_{max}}$	$\mathcal{F}_a^{P,h^a_{max}}$	$\varphi_a^{P,h^a_{max}}$	$W_a^{P,h^a_{max}}$	$\mathcal{F}_a^{P,h^a_{max}}$	$\varphi_a^{P,h^a_{max}}$	$W_a^{P,h^a_{max}}$	$\mathcal{F}_a^{P,h^a_{max}}$	$\varphi_a^{P,h^a_{max}}$	$W_a^{P,h^a_{max}}$
8	E,D	0.41,0.41	0.85,0.89	0.34,0.40	0.45,0.48	0.90,0,90	0.42,0.46	0.20,0.44	0.78,1.0	0.24,0.49	0.59,0.39	0.78,0.85	0.53,0.33
13	P	0.52	0.9	0.39	0.48	1	0.32	0.61	1.0	0.41	0.47	0.72	0.29
18	D	0.42	0.97	0.56	0.48	1	0.6	0.43	1.0	0.56	0.46	0.9	0.6
20	W	0.57	0.94	0.58	0.44	1	0.67	0.75	1.0	0.57	0.81	0.92	0.86
63	H	0.32	0.5	0.37	0.31	0.3	0.42	0.44	0.57	0.48	0.37	0.44	0.38

Figure 4:Logo consists of the weighted stacking of amino acids relative to the occurrences of amino acids in the protein sequences at heads 8, 13, 18, 20, and 63.

3.1.3 Assessment of biochemical and biophysical properties of amino acids at heads of ProtAlbert

In the previous sub-section, we found representative heads 8 and 18 for amino acids D and E. They are hydrophilic acidic amino acids. In the following, we assess the heads in layers to find more biochemical and biophysical properties based on the R-group of amino acids. This classification, $\mathbb{C}=\{N,H,U,A,B\}$, is shown in Table 1. To do the assessment, we apply the RH_BBP algorithm on dataset $\Delta \subseteq CASP13$. In Eq.8 and Eq.9 of this algorithm, we consider threshold 0.4 to make adjacency matrix from attention matrix. In the third step of RH_BBP, head h^{C}_{max} is selected to identify the maximum quantity for class $C \in \mathbb{C}$. At the fourth step, we announce that head h^{C}_{max} is representative for class C if $\bar{\mathcal{F}}^{h^{C}_{max}}_{C} \geq 0.3$. Meanwhile, we compute $\mathcal{F}^{P,h^{C}_{max}}_{C}$ for each protein $P \in \{LuxB,\ Mpro,\ Taq\}$at the selected head h^{C}_{max} . Table 7 shows the average F-measure for representative class C at head h^{C}_{max} is similar to the case study ones.

Moreover, this table represents the average relative occurrence of class C for dataset Δ and each case study protein P by $\bar{\varphi}^{h^{C}_{max}}$and $\varphi^{P,h^{C}_{max}}_{C}$, respectively. In addition, the average weighted occurrence of class C and each case study protein P at this head are shown by $\bar{W}^{h^{C}_{max}}$ and $W^{P,h^{C}_{max}}_{C}$, respectively.

In conclusion, representative heads 8, 44, and 49 show hydrophilic acidic, hydrophobic aliphatic, and hydrophobic aromatic amino acids, respectively. Also, head 43 can represent both polar and basic amino acids. Figure 5 consists of the weighted stacking of amino acids ($\bar{W}^{h^{C}_{max}} \times \bar{\varphi}^{h^{C}_{max}}$) at attention heads 8, 43, 44, and 49.

Table 7: The representative heads for the classes in set $\mathbb{C}$ based on F-measure ($\bar{\mathcal{F}}^{h^{C}_{max}}_{C} \geq 0.3$).

Head	Class Amino acids	Dataset= Δ			Protein=LuxB			Protein=Mpro			Protein=Taq		
		$\bar{\mathcal{F}}^{h^{C}_{max}}_{C}$	$\bar{\varphi}^{h^{C}_{max}}$	$\bar{W}^{h^{C}_{max}}$	$\mathcal{F}^{P,h^{C}_{max}}_{C}$	$\varphi^{P,h^{C}_{max}}_{C}$	$W^{P,h^{C}_{max}}_{C}$	$\mathcal{F}^{P,h^{C}_{max}}_{C}$	$\varphi^{P,h^{C}_{max}}_{C}$	$W^{P,h^{C}_{max}}_{C}$	$\mathcal{F}^{P,h^{C}_{max}}_{C}$	$\varphi^{P,h^{C}_{max}}_{C}$	$W^{P,h^{C}_{max}}_{C}$
8	A={E,D}	0.66	0.86	0.74	0.72	0.81	0.88	0.53	0.92	0.61	0.77	0.81	0.86
43	B={R,H,K}	0.46	0.64	0.28	0.51	0.71	0.33	0.36	0.69	0.15	0.56	0.60	0.33
	U={ S,T,C,N,Q }	0.40	0.40	0.44	0.46	0.45	0.43	0.58	0.56	0.60	0.14	0.37	0.35
44	N={G,A,V,L,I,P,M}	0.63	0.99	0.67	0.56	1	0.58	0.62	1	0.59	0.71	0.97	0.73
49	H={F,Y,W}	0.45	0.86	0.5	0.5	0.83	0.46	0.54	0.94	0.48	0.46	0.86	0.33

Figure 5: The logo consists of weighted stacking of amino acids in class $C \in \mathbb{C}$ relative to the occurrences of these amino acids in the protein sequences at heads 8, 43, 44, and 49.

3.1.4 Assessment of the protein secondary structure at heads of ProtAlbert

ProAlbert has been pre-trained on protein sequences, but we use the RH_PSS algorithm on dataset $\Delta \subseteq CASP13$ to show that some heads with high attention weights are attending from helix to helix, sheet

to sheet, and coil to coil. In Eq.10 of this algorithm, we consider threshold 0.1 to make an adjacency matrix from the attention matrix. At step two of RH_PSS, the average of cosine similarity, $\bar{C}^{h}$, between the predicted and natural secondary structure matrices.

Figure 6 shows the heatmaps of $\bar{C}^{h}$, at each head h, $1 \leq h \leq 64$ on data set $\Delta \subseteq CASP13$. In addition, the cosine similarity, $\cos(\mathscr{H}^{P,h}, H^{P,h})$, for each case study protein $P \in \{\text{Taq}, \text{Mpro}, \text{LuxB}\}$is computed. The high similarity between the predicted and natural protein secondary matrices can be seen at heads 2, 3, 8, 9, 10, 13,14, 18, 20, 21, 23, 32, 49, 51, 53, 56, and 63. Some of these heads are common with the heads in nearest-neighbor interaction. After removing the common heads, we find that heads 8, 9, 13,18, 20, 32, 49, and 63 are only informative about the secondary structure. These heads show more attention from each amino acid secondary structure to the same structure, with less than 6 amino acids in neighbors.

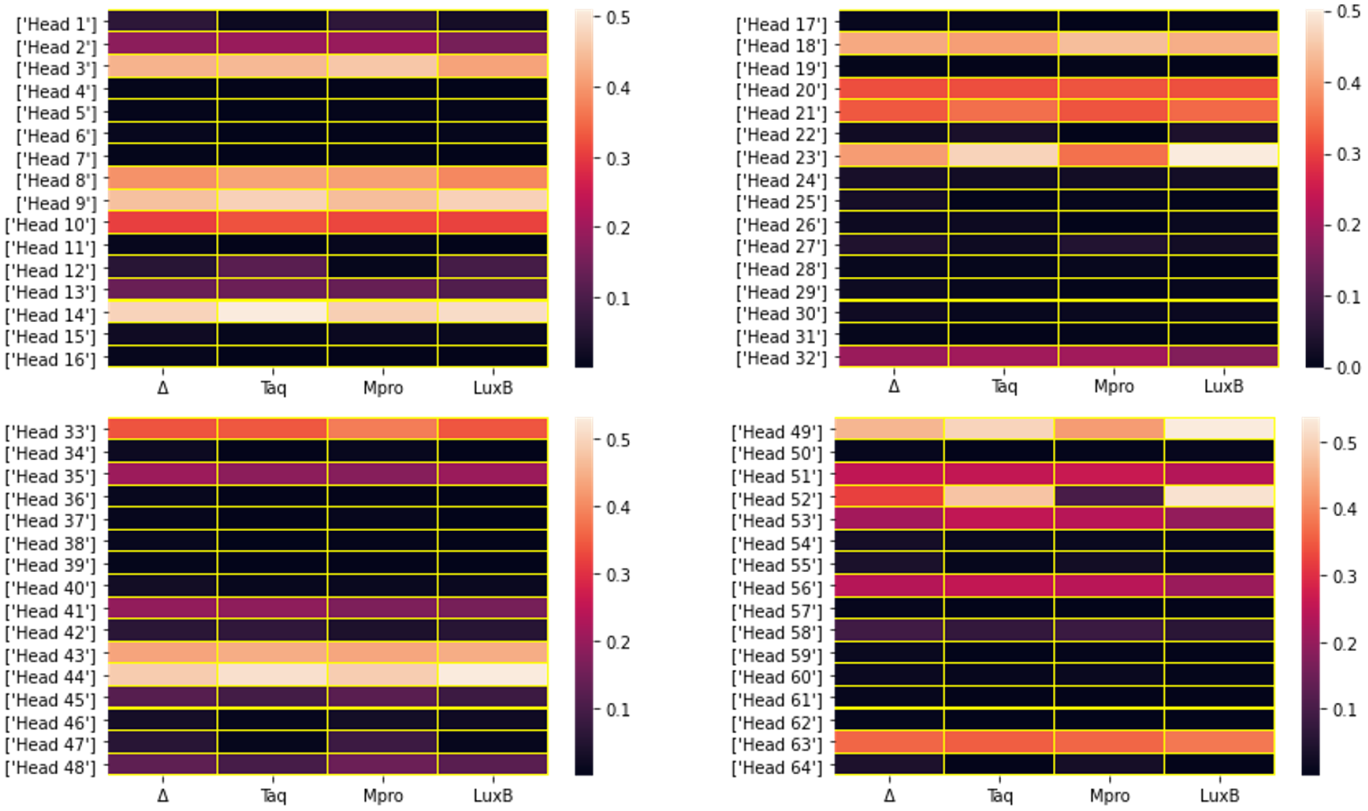


Figure 6: Heatmap of cosine similarity between the predicted and natural protein secondary structure matrices

3.1.5 Assessment of the protein tertiary structure at heads of ProtAlbert

In this sub-section, we compute the similarity between the natural contact map (see Eq.1) and predicted contact map of protein P using the RH_PTS algorithm on dataset $\Delta \subseteq \text{CASP13}$. In this algorithm, threshed 0.1 is defined for Eq.11 to discretize the predicted contact map.

Table 8 shows the average similarity between the predicted and natural contact maps at $h_{max} = 10$ on dataset Δ obtained from step three of the algorithm. Then $\bar{C}^{h_{max}}$ is calculated based on the second step of RH_PTS. In addition, the cosine similarity, $\cos(\mathscr{D}^{P,h_{max}}, D^{P})$ is computed for each case study protein $P \in \{Taq,\ Mpro, LuxB\}$. It seems that head 10 can show appropriate information on contact maps.

Table 8: Cosine similarity between natural and predicted contact map for proteins at head 10.

Data	Cosine similarity
Dataset=Δ	$\bar{C}^{10}{}_{max}$=0.7949
P=Mpro	$\cos(\mathcal{D}^{P,10}, D^P)$ =0.7299
P=Taq	$\cos(\mathcal{D}^{P,10}, D^P)$ =0.8071
P=LuxB	$\cos(\mathcal{D}^{P,10}, D^P)$ =0.8167

3.2 Predicting profile using ProtAlbert

The above assessment shows that transformers can extract some protein features from the sequence to represent the protein family. These features can lead us to find appropriate information about the homologous sequences of each protein sequence given as an input to ProtAlbert. Therefore, the PA_SPP algorithm (see Figure 1) employs pre-trained ProtAlbert to predict a profile for a query sequence. Here, we compare the predicted profiles to real ones obtained from the homologous sequences (HSSP profile).
For each protein $P \in \Delta \subseteq \mathrm{CASP13}$ and three case study proteins, Taq, Mpro, and LuxB, the PA_SPP algorithm predicts profile $\mathfrak{R}^P$. Then, we compare the similarity of the predicted profile $\mathfrak{R}^P$ to HSSP profile R^P using cosine similarity. We want to show that the predicted profile is close to the HSSP profile. It should be noted; some HSSP profiles are more reliable than the other ones because the number distribution of sequences aligned to the query sequence is different. For example, some profiles are obtained by less than 100 aligned sequences, and some are made based on more than 1000 aligned sequences. Therefore, the HSSP profile constructed with more aligned sequences is more reliable. So, the weighted average similarity between predicted and HSSP profiles are computed by the number of aligned sequences. Figure 7 shows that the predicted profiles are more similar to the HSSP profiles with more alignment sequences.

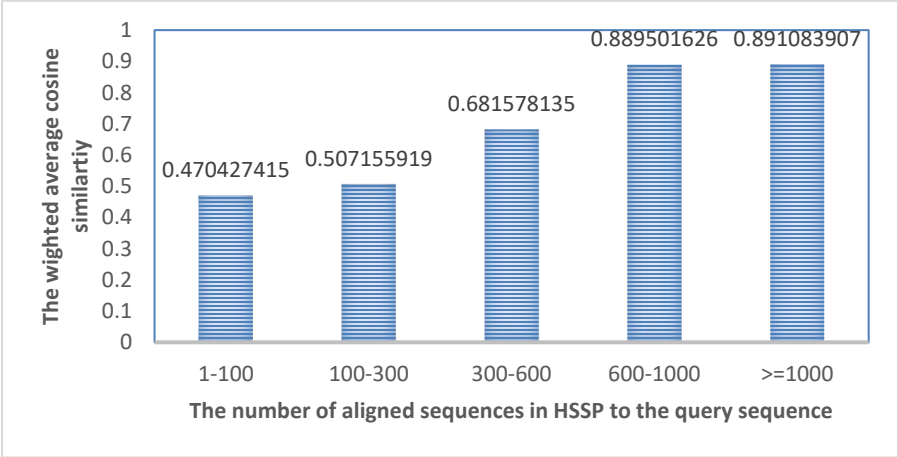


Figure 7: The weighted cosine similarity between predicted profiles and HSSP profiles based on the number of aligned sequences to the query sequence.

4. Conclusion

This paper contained two parts, ProtAlbert model analysis and profile prediction. Most previous studies used pre-trained transformer models to generate an embedding for protein sequence in different

bioinformatics problems as a black box. Here, we would like to find the representative heads in layers for some protein characteristics. For this assessment, we used ProtAlbert because its efficiency enables us to run the model on longer sequences with less computation power while having similar performance with the other pre-trained transformers on proteins which is a great advantage for us.

In this study, we did not train or fine-tune ProtAlbert. In other words, we used pre-trained ProtAlbert to determine the interaction of nearest-neighbor amino acids, type of amino acids, biochemical and biophysical properties of amino acids, protein secondary structures, and tertiary structures at attention heads in different layers. This analysis is crucial because it shows that ProtAlbert learns some protein family features from only sequences. It led us to propose an algorithm called PA_SPP for profile prediction from a query sequence using ProtAlbert. The results showed that the predicted profile is close to the profile obtained from the homologous sequences.

We believe that the proposed algorithm for profile prediction can help the researchers to make a profile for a query sequence while there are no similar sequences to the query sequence in the database. In the future, we can improve this predictor with new transformer models.

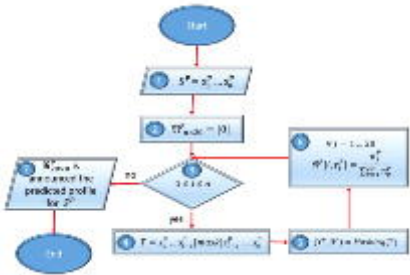
5. Reference

1. Alberts B, Johnson AD, Lewis J, et al. *Molecular Biology of Cell*. W. W. Norton & Company; 2014.
2. Consortium TU. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res*. 2019;47(D1):D506-D515.
3. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*. 2000;28(1):45. doi:10.1093/NAR/28.1.45
4. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res*. 2000;28(1):235-242.
5. Seffernick JT, Lindert S. Hybrid methods for combined experimental and computational determination of protein structure. *J Chem Phys*. 2020;153(24):240901.
6. Bhattacharya D, Cao R, Cheng J. UniCon3D:de novoprotein structure prediction using united-residue conformational search via stepwise, probabilistic sampling. *Bioinformatics*. 2016;32(18):2791-2799.
7. Guo Z, Hou J, Cheng J. DNSS2: Improved ab initio protein secondary structure prediction using advanced deep learning architectures. *Proteins Struct Funct Bioinforma*. 2021;89(2):207-217.
8. Hou J, Wu T, Cao R, Cheng J. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins Struct Funct Bioinforma*. 2019;87(12):1165-1178.
9. Eddy SR. Profile hidden Markov models. *Bioinformatics*. 1998;14(9):755-763.
10. Haft DH, Loftus BJ, Richardson DL, et al. TIGRFAMs: a protein family resource for the functional identification of proteins. *Nucleic Acids Res*. 2001;29(1):41-43.
11. Nguyen N, Nute M, Mirarab S, Warnow T. HIPPI: highly accurate protein family classification with ensembles of HMMs. *BMC Genomics*. 2016;17(10):89-100.
12. Galzitskaya O V., Melnik BS. Prediction of protein domain boundaries from sequence alone. *Protein Sci*. 2003;12(4):696.
13. Chen J, Liu B, Huang D. Protein remote homology detection based on an ensemble learning approach. *Biomed Res Int*. 2016;2016:5813645.
14. Liu B, Chen J, Wang X. Application of learning to rank to protein remote homology detection. *Bioinformatics*. 2015;31(21):3492-3498.

15. Pan X, Kortemme T. Recent advances in de novo protein design: Principles, methods, and applications. *J Biol Chem*. 2021;296:100558.
16. Zhang Y, Chen Y, Wang C, et al. ProDCoNN: Protein design using a convolutional neural network. *Proteins Struct Funct Bioinforma*. 2020;88(7):819-829.
17. Hulsen T, Huynen MA, de Vlieg J, Groenen PMA. Benchmarking ortholog identification methods using functional genomics data. *Genome Biol*. 2006;7(4):R31.
18. Schneider R, de Daruvar A, Sander C. The HSSP database of protein structure-sequence alignments. *Nucleic Acids Res*. 1997;25(1):226-230.
19. Creighton T. *Proteins Structures and Molecular Properties*. W.H. Freeman and Company.; 1993.
20. Rost B. Twilight zone of protein sequence alignments. *Protein Eng Des Sel*. 1999;12(2):85-94.
21. Liu B, Wang X, Lin L, Dong Q, Wang X. A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. *BMC Bioinformatics*. 2008;9:510.
22. Ingraham J, Garg VK, Barzilay R, Jaakkola T. Generative models for graph-based protein design. In: *Advances in Neural Information Processing Systems 32*. ; 2019:9689-9701.
23. Heinzinger M, Elnaggar A, Wang Y, et al. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*. 2019;20(1):723.
24. Armenteros JJA, Johansen AR, Winther O, Nielsen H. Language modelling for biological sequences – curated datasets and baselines. *bioRxiv*. Published online March 9, 2020. doi:10.1101/2020.03.09.983585
25. McCann B, Bradbury J, Xiong C, Socher R. Learned in Translation: Contextualized Word Vectors. In: *Advances in Neural Information Processing Systems*. ; 2017:6294-6305.
26. Peters ME, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol 1. ; 2018:2227-2237.
27. Madani A, McCann B, Naik N, et al. ProGen: Language Modeling for Protein Generation. *bioRxiv*. Published online March 13, 2020. doi:10.1101/2020.03.07.982272
28. Rives A, Meier J, Sercu T, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci*. 2021;118(15):e2016239118.
29. Elnaggar A, Heinzinger M, Dallago C, et al. ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Trans Pattern Anal Mach Intell*. 2021;14(8):1-16.
30. Strodthoff N, Wagner P, Wenzel M, Samek W. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*. 2020;36(8):2401-2409.
31. Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Vol 30. ; 2017:6000-6010.
32. Devlin J, Chang M-W, Lee K, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ; 2019:4171-4186.
33. Bepler T, Berger B. Learning protein sequence embeddings using information from structure. In: *7th International Conference on Learning Representations, ICLR 2019*. ; 2019.
34. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods*. 2019;16(12):1315-1322.

35. Rao R, Bhattacharya N, Thomas N, et al. Evaluating Protein Transfer Learning with TAPE. *Adv Neural Inf Process Syst*. 2019;32:9689.
36. Lu AX, Zhang H, Ghassemi M, Moses A. Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximization. *bioRxiv*. Published online September 6, 2020. doi:10.1101/2020.09.04.283929
37. Min S, Park S, Kim S, Choi H-S, Yoon S. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *arXiv*. Published online November 25, 2019. Accessed September 11, 2021. http://arxiv.org/abs/1912.05625
38. Sturmfels P, Allen PG, Vig J, Madani A, Rajani NF. Profile Prediction: An Alignment-Based Pre-Training Task for Protein Sequence Models. *arXiv*. Published online 2020. Accessed September 11, 2021. https://arxiv.org/abs/2012.00195
39. Kovacs JM, Mant CT, Kwok SC, Osguthorpe DJ, S.Hodges R. Quantitation of the Nearest-neighbour Effects of Amino Acid Side-Chains that Restrict Conformational Freedom of the Polypeptide Chain using Reversed-Phase Liquid Chromatography of Synthetic Model Peptides with L- and D-amino Acid Substitutions. *J Chromatogr A*. 2006;1123(2):212-224.
40. Brocchieri L, Karlin S. How are close residues of protein structures distributed in primary sequence? *Biophysics (Oxf)*. 1995;92:12136-12140.
41. Pietal MJ, Bujnicki JM, Kozlowski LP. GDFuzz3D: a method for protein 3D structure reconstruction from contact maps, based on a non-Euclidean distance function. *Bioinformatics*. 2015;31(21):3499-3505.
42. Salzberg S, Cost S. Predicting protein secondary structure with a nearest-neighbor algorithm. *J Mol Biol*. 1992;227(2):371-374.
43. Ashok Kumar T. CFSSP: Chou and Fasman Secondary Structure Prediction server. *Wide Spectr*. 2013;1(9):15-19.
44. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv*. Published online September 26, 2019. Accessed September 13, 2021. https://arxiv.org/abs/1909.11942v6

[i] **R**epresentative **H**eads in **L**ayers of ProtAlbert for **N**earest **N**eighbor **I**nteractions
[ii] **R**epresentative **H**eads of ProtAlbert for **S**pecific **A**mino **A**cids
[iii] **R**epresentative **H**ead of ProtAlbert for **B**iochemical and **B**iophysical **P**roperties of amino acids
[iv] **R**epresentative **H**eads of ProtAlbert for **P**rotein **S**econdary **S**tructure
[v] **R**epresentative **H**eads of of ProtAlbert for **P**rotein **T**ertiary **S**tructure
[vi] Using **P**rot**A**lbert for **S**equence **P**rofile **P**rediction
[vii] https://microbenotes.com/amino-acids-properties-structure-classification-and-functions/
[viii] https://www.rcsb.org/
[ix] https://www3.cmbi.umcn.nl/xssp/
[x] https://www.predictioncenter.org/casp13/domains_summary.cgi

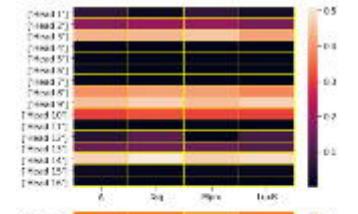
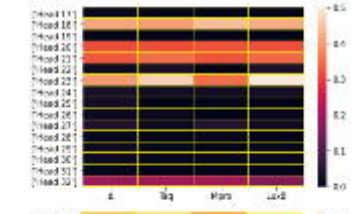
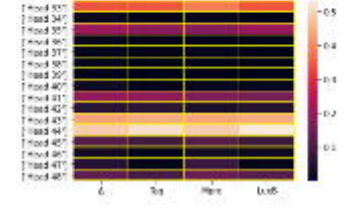
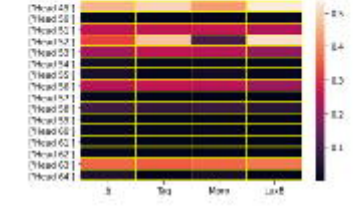

Start
No
Yes
End


The number of sequences in A
16
14
12
10
8
6
4
2
0
[0, 100]
[100, 200]
[200, 300]
[300, 400]
[400, 500]
[500, 600]
The length of sequences in A


Frequency
1400
1200
1000
800
600
400
200
0
V L I M T H Y G A P S T C H R K Q E N D X
AA


information (bits)
0.0
0.2
0.4
0.6
-5
-15
-25
-35
-45


information (bits)
0.0
0.2
0.4
0.6
0.8


The aligned average score similarity
0.6
0.5
0.4
0.3
0.2
0.1
0
0.6815781355
1-100
100-200
200-300
300-400
400-500
>500
The number of aligned sequences in HSSP to the query sequence