Highly accurate protein structure prediction with AlphaFold

Loghman Samani

Introduction

Proteins, which are fundamental to life, play a crucial role in biological processes, so understanding their structures is essential for deciphering their functions [1]. Despite experimental efforts that have led to structural insights for about 100,000 unique proteins, this is only a fraction of the millions of known protein sequences [7]. The bottleneck in structural coverage is due to the laborious and time-consuming nature of experimental determination [3]. To close this gap and enable large-scale structural bioinformatics, accurate computational approaches are essential. Predicting the three-dimensional structure of a protein based solely on its amino acid sequence, a key aspect of the protein folding problem, has been a challenge for over 50 years. While experimental structures cover only 17% of human protein residues [2], AlphaFold2 represents a breakthrough computational solution that utilizes machine learning methods on an unprecedented scale to predict protein structures with atomic accuracy.

AlphaFold covers a remarkable 98.5% of the human proteome and provides reliable predictions for 58% of residues, with an exceptional subset showing a very high degree of confidence [2]. In this article, I would like to explain the intricate details of the AlphaFold system (Figure 1) and break down its methodology step by step. As illustrated in Figure 1 the system can be divided into three main components: Database Search, Evoformer Module and Structure Module. Each of these segments plays a pivotal role in the system, and the flow of information retrieved from the protein database through the various parts of the system leads to an extremely accurate 3D structure of a protein.

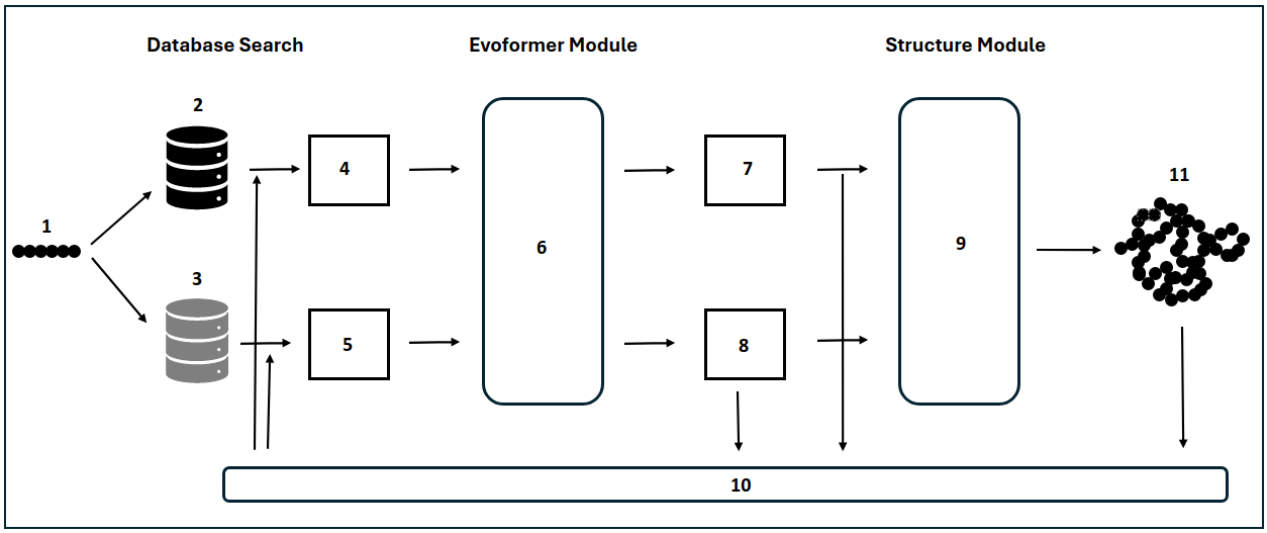

Figure 1: The AlphaFold system. The system consists of three main components: Database Search, Evoformer Module and Structure Module. The explanations of each component and its layers (represented by numbers) are detailed in the corresponding part of the article.

Genetic and Structure Database Searches

The initial stage of the AlphaFold system is the Data Pipeline, which is the process of gathering both sequential and structural information essential for effective model training and enhancing the accuracy of the inference process. In addition to amino acid sequence files in the FASTA format, another resource utilized during the training phase of the system is the Macromolecular Crystallographic Information File (mmCIF). As implied by its name, mmCIF contains a diversity of sequential and structural protein information derived from proteins collected through experimental methods such as X-ray crystallography. During the inference phase, the system requires only the sequence identifier, which is then used to retrieve the amino acid sequence of the protein in the FASTA format. The system parses the input files (mmCIF and fasta) and extracts relevant information, in the case of the mmCIF this is the name of the sequence, the amino acid sequence, the atomic coordinates, the release date and the resolution. In the other case (fasta), the amino acid sequence and its name are the only information required. With the information retrieved from the files, two different database searches are performed, in one case, the genetic database search (Figure 1. 2), the system searches the Big Fantastic Database (BFD) [4] for similar sequences (in this case, the system used specialized applications such as JackHMMER and HHBlits) and the result is called multiple sequence alignment (MSA), then the result is modified by removing duplicate sequences and those that are not long enough to be included in the process. The remaining sequences are then assembled into a matrix of sequences called the *MSA representation*. Each column of this matrix encodes a single residue of the input sequence in different organisms, while each row represents the entire sequence in a particular organism. In another case, the system uses an approach called HHSearch to search in the Protein Data Bank (PDB), Structure Database Search (Figure 1. 3), to retrieve the 3d structures of the proteins, that have a similar primary structure to the original input (Figure 1. 1).

Embedding process

Before the collected sequential (MSA information) and structural information can be used in the alphafold system, it must be embedded. This is the process of converting categorical information, such as the alphabetical representations (G, V, A, ... Each letter represents an amino acid) found in MSA, as well as the structural features, into numerical vectors or probabilities [5]. For this purpose, each input symbol or word is converted into a vectorized representation. These numerical vectors serve as input for neural networks, enabling them to effectively process and learn patterns from the sequential data. Regarding the MSA, the resulting embedded information is denoted as the MSA representation (Figure 1. 4). This representation takes the form of a matrix with dimensions (number of sequences × length of the longest sequence). Conversely, in the other case, the embedded outcome is termed the Pair Representation (Figure 1. 5), which manifests as a matrix with dimensions (length of sequence × length of sequence). These distinct yet crucial matrices serve as the primary input parameters during both the training and inference phases of the system. They encapsulate essential structural and sequential information, facilitating subsequent steps in the algorithm.

Evoformer module

The Evoformer (Figure 1. 6) stands out as a pivotal and potent module within the AlphaFold system. Fundamentally, it operates as a variant of the transformer neural network, originally introduced by Google Brain in 2017 [8]. Initially designed for language translation tasks, transformers excel with sequential data [9]. In the context of AlphaFold, the Evoformer takes both MSA-representation and pair representation as input. Leveraging this information, it produces an output that is a prediction of the protein structure as a graph inference problem [6] in 3D space. In this 3D space, the nodes of the graph correspond to amino acid residues in proximity, while the edges represent the spatial relationships between these residues.

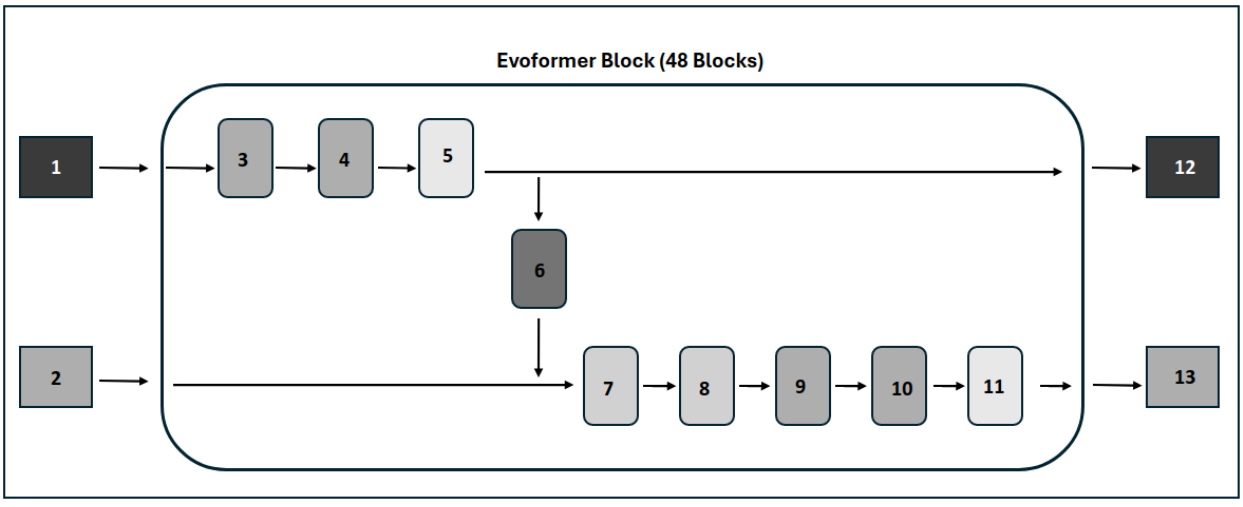

Figure 2: The Evoformer network. It consists of 48 (by default) blocks, each block has an MSA representation (1) and a pair representation (2) as its input and output and processes them within several layers. The detailed explanations of each layer, the core components of this network's functionality, are meticulously expounded in the corresponding part of the article.

The Evoformer block (figure 2) comprises two transformer models, each dedicated to processing one of the primary input datasets, MSA representation (Figure 2, 1) and pair representation (Figure 2, 2). These transformers collaboratively exchange information across two distinct layers within the block. In the initial layer of the block, information flows from pair representation to MSA representation. Meanwhile, in the sixth layer (outer product mean, Figure2, 6), information is reciprocally transmitted from the MSA representation transformer to the pair representation transformer. This intricate information exchange mechanism enhances the Evoformer's ability to integrate insights from both MSA and pair representations, contributing to the overall refinement and accuracy of the protein structure prediction. Within each of the two transformers in the Evoformer block, there are two main components. The initial part is referred to as self-attention, which allows the model to weigh the significance of different positions within the input sequence. The second part is denoted as the Feed Forward Neural Network (FFNN), which processes the information from the self-attention mechanism, aiding in capturing intricate patterns and dependencies within the data.

Self-Attention

The main idea behind self-attention in a Transformer is to enable the model to weigh the importance of different words in a sequence when processing each word. It allows the model to consider the relationships and dependencies between words in a flexible and adaptive way, capturing long-range dependencies and improving its ability to understand the context of each word. In the context of protein structure, a diverse set of 20 unique amino acids (analogous to words) in each sequence plays a vital role in determining distinct folding patterns. It is essential to gather detailed information about each residue and understand how they interact with neighboring residues. This comprehensive knowledge is crucial for inferring higher-order structures such as secondary, tertiary, and quaternary structures in proteins. To gain this necessary information alphafold defines axial self-attention (Figure 2. 3 & 4) in the MSA representation transformer and triangular self-attention (Figure 2. 9 & 10) in the pair representation transformer.

Axial self-attention in the MSA stack

The initial step in the Evoformer's self-attention mechanism involves creating three vectors—query (q), key (k), and value (v)—from each row (row-attention, Figure 2. 3) or column (column-attention, Figure 2. 4) of the embedded MSA-representation. For each vector, a corresponding weight matrix (Wq, Wk, and Wv) is employed, and these matrices are then multiplied by the

embedded MSA (X) to generate the Q, K, and V matrices. This procedure is known as positional encoding, wherein a matrix of weights is added to each input embedding. The positional encoding matrix adheres to a specific pattern learned by the model, aiding in determining the position of each residue or the distance between different residues in the sequence.

Following the creation of the Q, K, and V matrices, the next step in the Evoformer's self-attention mechanism involves activating these matrices using the SoftMax activation function. This activation method ensures that the scores are positive and sum up to 1. As depicted in Figure 2-a, there exists a connection between the MSA-transformer and pair-transformer, which demonstrated the flow of information from pair representation to the MSA-representation. In this process, additional information from pair representation is incorporated to influence the bias terms of the MSA attention.

$Wq \times X = Q$
$Wk \times X = K$
$Wv \times X = V$

column-attention: $Z = \text{SoftMax}((Q \cdot K.T)/\text{sqrt}(d)) \times V$; d = dimension of the keys, queries, and values matrices
row-attention: $Z = \text{SoftMax}((Q \cdot K.T)/\text{sqrt}(d) + b) \times V$; b = pair representation information

Feed Forward Neural Networks

After row-wise and column-wise attention, the next layer of the MSA transformer contains a 2-layer MLP, multi-layer perceptron, known as FFNN (Figure 2. 5), as the transition layer [10]. This stage of processing in the Evoformer block operates across features, refining the representation using a non-linear transform. The main idea behind a feed-forward neural network is to process input data through a series of layers, where each layer consists of nodes, connected to nodes in the subsequent layer. During the training process of a Feedforward Neural Network (FFNN), each weight matrix associated with the layers of the network is optimized using Adam optimization algorithm, which is a gradient based optimization algorithm, to improve the accuracy of predicted structures. Notably, in a conventional neural network architecture, weights are typically shared across different layers. However, in the case of the Evoformer, each block operates with its own set of weights, and these weights are not shared with other blocks. This contrasts with the usual neural network setup, where weights are often shared across layers. The unique characteristic of Evoformer lies in the independence of weights for each block, allowing for more localized and specific adaptations during the training process.

$$FFN(X) = \max(0, X \times W1 + b1) \times W2 + b2$$

Outer Product Mean

This layer of the Evoformer (Figure 2. 6) serves the purpose of transforming the MSA-representation into an update for the pair representation. The transformation involves computing the mean of the outer product of each two columns (for instance, Ci and Cj), and this result is utilized as an update for the element (Cij) in the pair representation. This mechanism ensures the integration of information from the MSA into the Pair representation module.

Mathematically, this process can be represented as:

$$\text{Outer product matrix} = C^1(m \times n) \times C^2(m \times n).\ T$$
$$\text{Outer product mean} = C^3 = \text{mean (outer product mean)}$$

Triangle multiplicative update and triangle self-attention

In the Evoformer block, the process of updating the pair representation is designed to implement an attention mechanism akin to the one applied in the Multiple Sequence Alignment representation. The objective is to capture intricate relationships between residues in a three-dimensional protein structure, enabling a nuanced understanding of how each residue influences the spatial representation of others in the molecular space (Figure 2. 7,8,9,10).

As illustrated in Figure 3. a, a graph representation is constructed from the pair representation matrix. In this graph, each node represents an amino acid, and the edges correspond to entries in the matrix. The circles within the graph symbolize individual residues. To simplify the explanation, we focus on updating a single edge, ij, leveraging information from other edges.

During the Triangular Multiplicative Update step, the edge ij undergoes an update by assimilating information from the other edges. It is noteworthy that there exist two symmetric versions of this update, differentiating between outgoing and incoming edges. This ensures a comprehensive consideration of information flow within the graph structure (Figure 3. b).

Following the Triangular Multiplicative Update, the process continues with the Triangular Self-Attention step. In this step, the edge ij is further updated by incorporating values from all edges that share the same starting node. This comprehensive attention mechanism facilitates the integration of information from neighboring residues, contributing to a refined and context-aware representation of the 3D spatial relationships between amino acids in the protein structure.

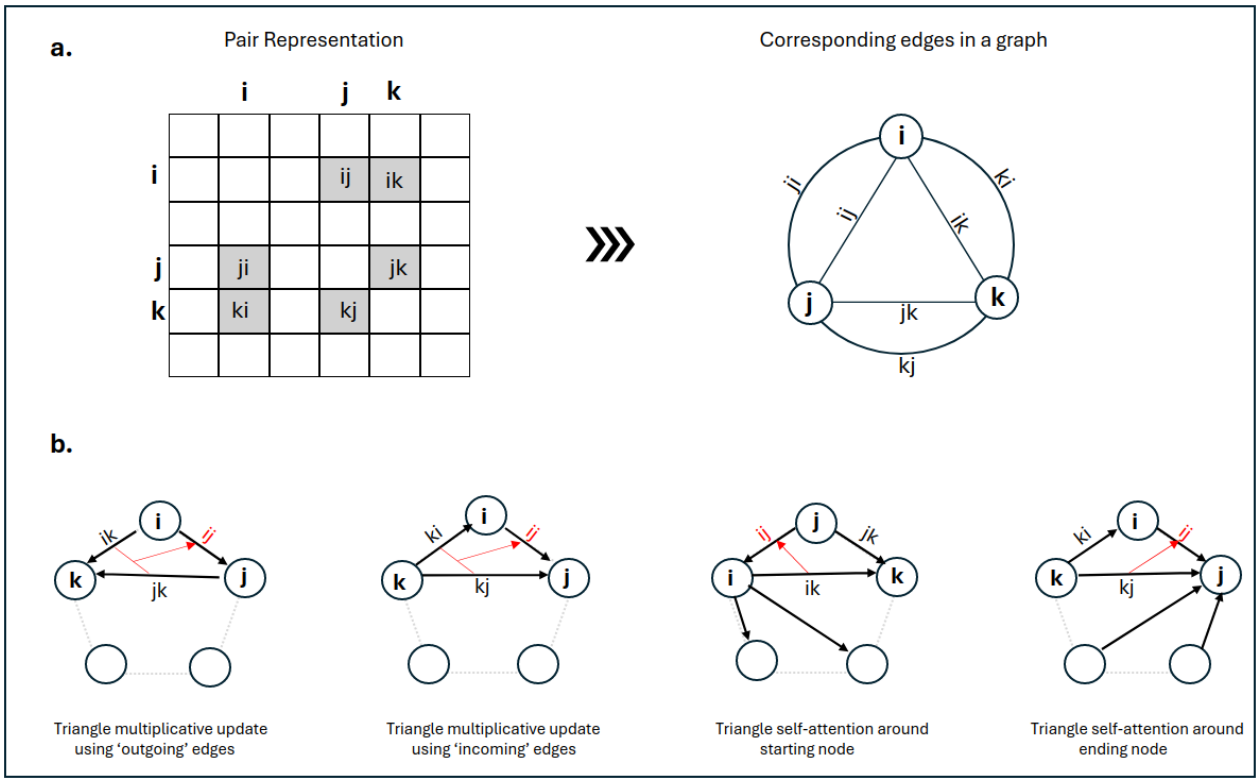

Figure 3: Triangle Multiplicative Update and Triangle Self-Attention. a) The pair representation interpreted as directed edges in a graph. b) Triangle multiplicative update and triangle self-attention. The circles represent residues. Entries in the pair representation are illustrated as directed edges, and in each diagram, the edge being updated is ij.

After the attention mechanism processes the pair representation in the Evoformer block, the subsequent step involves a transition layer, feedforward neural network, (Figure 2. 11). This transition layer is analogous to the one found in the MSA representation transformer. In this layer, the pair representation is further refined and transformed through the application of a feedforward neural network.

The structure module

In the final step of AlphaFold system, the algorithm extracts the first row of the highly processed MSA representation (Figure 4. 2), which corresponds to the original sequence, and utilizes the pair representation (Figure 4. 1). These two representations serve as the foundation for the construction of a three-dimensional model of the protein structure. The outcome of this process is a comprehensive list of Cartesian coordinates presented in the Protein Data Bank (PDB) format. These coordinates delineate the precise positions of each atom within the protein structure, encompassing not only the backbone but also the intricate details of side chains.

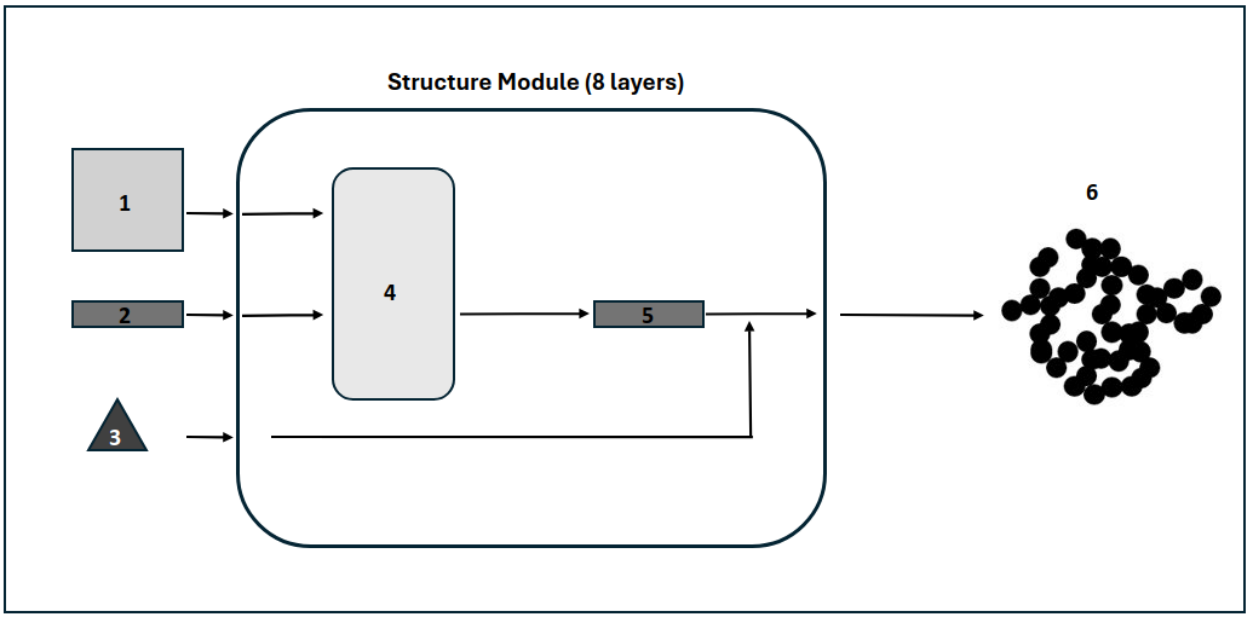

Figure 4: The structure module.

The single representation and pair representation and backbone frames, spatial representations, (Figure 4. 3) are used as input of the structure module. This module utilizes a gradient descent-based algorithm, Adam, to optimize the predicted 3D structure of the protein (Figure 4. 6). The structure module comprises 8 iterations by default, and in each iteration, the weights are shared among the layers, differing from the Evoformer module where each block has distinct weights.

During each iteration of the Structure Module, the following steps take place:

Invariant Point Attention

In this step of the Structure Module, known as Invariant Point Attention (IPA) (Figure 4. 4), a specialized type of attention mechanism designed for handling 3D structures is employed. The general attention mechanism, previously explained in detail in the Evoformer module, is adapted for use in this specific context. The Pair Representation and Single Representation (first row of the MSA representation) play important roles in this sub-step. These representations are utilized to iteratively refine the backbone frames, which collectively define the 3D structure of the protein.

In each iteration of the Structure Module during the training phase, the information (backbone frames) generated by IPA and the Single Representation is utilized to predict the 3D structure of the protein. The predicted structure at the end of each iteration is then compared to the actual structure using two key metrics:

Frame Aligned Point Score

This metric measures the difference between all atom predicted coordinates and the actual coordinates. It provides assessment of the alignment between the predicted and actual positions of each atom in the protein structure.

Torsion Angle Loss

Torsion angles represent the rotations around the chemical bonds in the protein structure, including both side chain and backbone torsion angles. The torsion angle loss quantifies the difference between the predicted torsion angles and the actual angles.

The calculated values for Frame Aligned Point Score and Torsion Angle Loss are combined, to form a unified loss function. This combined loss is then employed in the gradient descent algorithm during the training process. The gradient descent algorithm adjusts the model's parameters to minimize this loss, facilitating the refinement of the predicted 3D protein structure. By iteratively optimizing the model based on the disparity between predictions and actual observations, the algorithm converges towards a more accurate and biologically relevant protein structure.

Recycling

Following the prediction of the final protein structure in the Structure Module, all the generated and refined information, including the Multiple Sequence Alignment (MSA) representation, pair representation, and the predicted 3D structure, serves as input for Evoformer blocks. This recycling mechanism (Figure 1. 10) enables the model to iteratively refine its predictions. This iterative refinement process is repeated three times by default. In each iteration, the output from the previous step (Figure 1. 11) is embedded and utilized as additional input to the model. This iterative embedding of information ensures that the model progressively refines its understanding and representation of the protein structure. By incorporating the refined predictions from earlier iterations, the model can capture finer details, improve precision, and enhance the overall accuracy of its final predictions. The recycling mechanism contributes to the robustness and effectiveness of the AlphaFold algorithm in predicting highly accurate protein structures.

Conclusion

In conclusion, AlphaFold represents a groundbreaking advance in the field of protein structure prediction, bridging the gap between experimental efforts and the vast landscape of known protein sequences. The intricate interplay of its three main components - database search, Evoformer module and structure module - contributes to the remarkable achievement of predicting protein structures with unprecedented accuracy. Using machine learning methods to an impressive extent,

AlphaFold not only covers 98.5% of the human proteome, but also provides reliable predictions for 58% of residues, with a subset showing an exceptional level of confidence. This groundbreaking computational approach has the potential to significantly impact structural bioinformatics by overcoming the challenges posed by laborious experimental determinations and providing insights into the three-dimensional shape of proteins that are critical to understanding their functions. As we delve into the details of the AlphaFold methodology, it becomes clear that the fusion of genetic database searches, Evoformer's neural network architecture and the structural module results in an exceptionally accurate 3D representation of proteins. The success of AlphaFold underscores the power of computational methods in unlocking the secrets of protein structures, providing insight into the intricate world of molecular biology and opening new avenues for research and discovery.

References

[1] Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589 (2021). Accessed 14 January 2024.

[2] Tunyasuvunakool, K., Adler, J., Wu, Z. et al. Highly accurate protein structure prediction for the human proteome. Nature 596, 590–596 (2021). Accessed 14 January 2024.

[3] Senior, A.W., Evans, R., Jumper, J. et al. Improved protein structure prediction using potentials from deep learning. Nature 577, 706–710 (2020). Accessed 14 January 2024.

[4] Big Fantastic Database. Accessed 14 January 2024.

[5] Wang, S., Zhou, W. & Jiang, C. A survey of word embeddings based on deep learning. *Computing* **102**, 717–740 (2020). https://doi.org/10.1007/s00607-019-00768-7. Accessed 14 January 2024.

[6] Chunyan Xu, Zhen Cui, Xiaobin Hong, Tong Zhang, Jian Yang, Wei Liu. Graph Inference Learning for Semi-supervised Classification. Accessed 14 January 2024.

[7] Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic acids research* 47, no. D1 (2019): D520-D528. Accessed 14 January 2024.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. Accessed 14 January 2024.

[9] Richard E. Turner. An Introduction to Transformers. Accessed 14 January 2024.

[10] G. Bebis and M. Georgiopoulos, Feed-forward neural networks, in *IEEE Potentials*, vol. 13, no. 4, pp. 27-31, Oct.-Nov. 1994, doi: 10.1109/45.329294. Accessed 14 January 2024.