ExplainableFold: Understanding AlphaFold Prediction with Explainable AI

Juntao Tan
Rutgers University, New Brunswick, NJ, US
juntao.tan@rutgers.edu

Yongfeng Zhang
Rutgers University, New Brunswick, NJ, US
yongfeng.zhang@rutgers.edu

ABSTRACT

This paper presents ExplainableFold (xFold), which is an Explainable AI framework for protein structure prediction. Despite the success of AI-based methods such as AlphaFold (αFold) in this field, the underlying reasons for their predictions remain unclear due to the black-box nature of deep learning models. To address this, we propose a counterfactual learning framework inspired by biological principles to generate counterfactual explanations for protein structure prediction, enabling a dry-lab experimentation approach. Our experimental results demonstrate the ability of ExplainableFold to generate high-quality explanations for AlphaFold's predictions, providing near-experimental understanding of the effects of amino acids on 3D protein structure. This framework has the potential to facilitate a deeper understanding of protein structures. Source code and data of the ExplainableFold project are available at https://github.com/rutgerswiselab/ExplainableFold.

CCS CONCEPTS

• **Applied computing** → **Molecular structural biology**; • **Mathematics of computing** → **Computing most probable explanation**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

AlphaFold; Protein Structure Prediction; Explainable AI; Counterfactual Reasoning

ACM Reference Format:
Juntao Tan and Yongfeng Zhang. 2023. ExplainableFold: Understanding AlphaFold Prediction with Explainable AI. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599337

1 INTRODUCTION

The protein folding problem studies how a protein's amino acid sequence determines its tertiary structure. It is crucial to biochemical research because a protein's structure influences its interaction with other molecules and thus its function. Current machine learning models have gained increasing success on 3D structure prediction [3, 55]. Among them, AlphaFold [31] provides near-experimental accuracy on structure prediction, which is considered an important achievement in recent years. Nevertheless, one of the significant challenges with AlphaFold, as well as other deep learning models, is that they cannot provide explanations for their predictions. Essentially, the *why* question still remains largely unsolved: the model gives limited understanding of why the proteins are folded into the structures they are, which hinders the model's ability to provide deeper insights for human scientists.

However, explainability is a critically perspective in AI for Science research (Explainable AI for Science) [35], since science is not only about understanding the "how", but also, and perhaps more importantly, the "why". Specifically, in protein structure prediction research, it is crucial to understand the mechanism of protein folding from both AI and scientific perspectives. From the AI perspective, explainability has long been an important consideration. State-of-the-art protein structure prediction models leverage complex deep and large neural networks, which makes it difficult to explain their predictions or debug the trained model for further improvement. From the scientific perspective, scientists' eagerness to conquer knowledge is not satisfied with just knowing the prediction results, but also knowing the *why* behind the results [35]. In particular, structural biologists not only care about the structure of proteins, but also need to know the underlying relationship between protein primary sequences and tertiary structures [16, 17].

It has been established that certain amino acids play significant roles in the protein folding process. For instance, one single disorder in the HBB gene can significantly change the structure of hemoglobin, the protein that carries oxygen in blood, causing the sickle-cell anaemia [32]. Knowing the relationship between amino acids and protein structure helps scientists to produce synthetic proteins with precisely controlled structures [53] or modify existing proteins with desired properties [1, 38, 50], which are essential for advanced research directions such as drug design. Additionally, in certain research tasks, scientists would like to modify the amino acids without drastically changing the protein structure, which requires the knowledge of "safe" residue substitutions [7], i.e., the knowledge of which amino acids are not the most crucial ones in the folding process.

While currently there are few Explainable AI-based methods to study the mechanism of protein folding, many previous biochemical studies have been conducted for this purpose. One of the best known methods is via site-directed mutagenesis [9, 29, 44]. To test the role of certain residues in protein folding, biologists either delete them from the sequence (i.e., site-directed deletion) [4, 19, 23, 24] or replace them with other types of amino acids (i.e., site-directed substitution) [5, 7, 23] and then measure their influences on the 3D structure. However, these approaches suffer from several limitations: 1) So far, modification of such residues

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599337

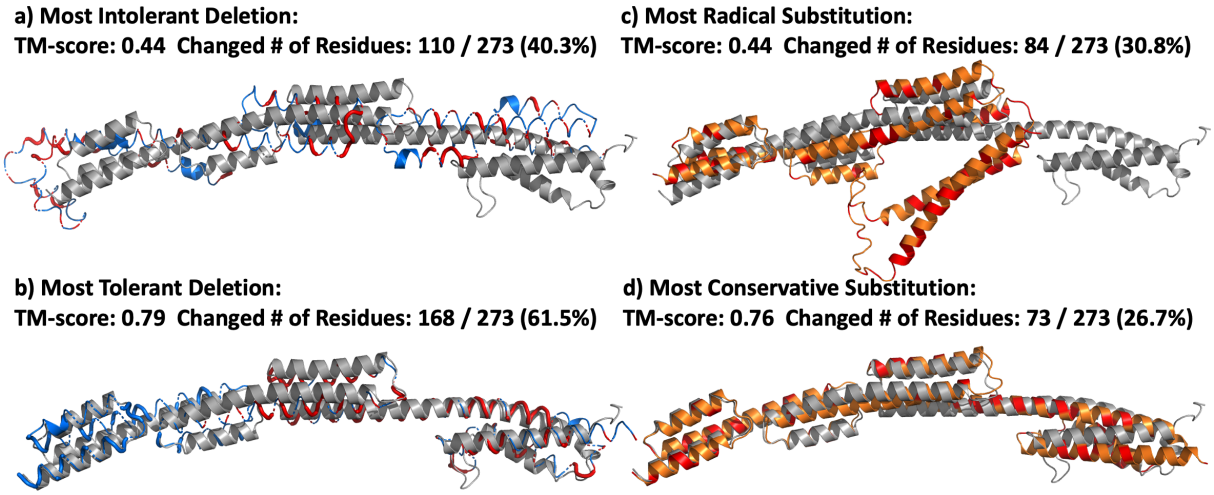


Figure 1: The original protein is colored gray; structures following amino acid deletion and substitution are blue and orange, respectively, with red indicating the altered residues. (a) Some amino acids play crucial roles in protein folding. By removing the effects of a relative small set of these residues, the predicted structure will be different. (b) Some other residues are less important. Despite deleting a large set of these residues, the protein still folds into a similar structure. (c) Some substitutions are radical to the protein structure and even a small number of such substitutions can drastically change the structure. (d) Some other substitutions are conservative and have small effect on the protein structure.

can be limited by methods for their installation and the chemistry available for reaction, and the modification of some residues can be very challenging [47], 2) Wet-lab methods for determining protein structures are very difficult and time-consuming [30], and 3) The wet lab experiments described above have many prerequisites and obstacles, and may not be completely safe for many researchers.

Recently, AI-based dry-lab methods such as AlphaFold provide near-experimental protein structure predictions [31], which sheds light on the possibility to generate insightful understandings of protein folding by explaining AlphaFold's inference process. Such (Explainable) AI-driven dry-lab approach will largely overcome the aforementioned limitations and can be very helpful for human scientists. Fortunately, we observe that the process of testing the effects of residues on protein structure by site-directed mutagenesis is fundamentally similar to counterfactual reasoning, a commonly used technique for generating explanations for machine learning models [12, 26, 51, 52, 54]. Intuitively, counterfactual reasoning perturbs parts of the input data, such as interaction records of a user [51], nodes or edges of a graph [52], pixels of an image [26], or words of a sentence [54], and then observes how the model output changes accordingly.

In this paper, we propose ExplainableFold, a counterfactual explanation framework that generates explanations for protein structure prediction models. ExplainableFold mimics existing biochemical experiments by manipulating the amino acids in a protein sequence to alter the protein structure through carefully designed optimization objectives. It provides insights about which residue(s) of a sequence is crucial (or indecisive) to the protein's structure and how certain changes on the residue(s) will change the structure, which helps to understand, e.g., what are the most impactful amino acids on the structure, and what are the most radical (or safe) substitutions when modifying a protein structure. An example of applying our framework on CASP14 target protein T1030 is shown in Figure 1, which shows that deletion or substitution of a small number of residues can result in significant changes to the protein structure, while some other deletions or substitutions may have very small effects. We evaluate the framework based on both standard explainable AI metrics and biochemical heuristics. Experiments show that the proposed method produces more faithful explanations compared to previous statistical baselines. Meanwhile, the predicted relationship between amino acids and protein structures are highly positively correlated with wet-lab biochemical experimental results.

2 RELATED WORK

The essential idea of the proposed method is to integrate counterfactual reasoning and site-directed mutagenesis analysis in a unified machine learning framework. We discuss the two research directions in this section.

2.1 Residue Effect Analysis by Site-directed Mutagenesis

Many studies in molecular biology, such as those involving genes and proteins, rely on the use of human-induced mutation analysis [48]. Early mutagenesis methods were not site-specific, resulting in entirely random and indiscriminate mutations [21]. In 1978, Hutchison et al. [29] proposed the first method that modifies biological sequences at desired positions with specific intentions, known as site-directed mutagenesis. Later, more precise and effective tools have been developed [18, 42]. Site-directed mutagenesis is widely utilized in biomedical research for various applications. In this section, we focus on the use of site-directed mutagenesis to study the impact of amino acid mutations on protein structures [49].

Two common approaches to site-directed mutagenesis are amino acid deletion and substitution [11]. The deletion approach involves the deletion of certain residues from the sequence and observes the effects on the structure. For instance, Glück and Wool [24] identified

the amino acids that are essential to the action of the ribotoxin restrictocin by systematic deletion of its amino acids. Flores-Ramírez et al. [23] proposed a random deletion approach to measure the amino acids' effects on the longest loop of GFP. Arpino et al. [4] conducted experiments to measure the protein's tolerance to random single amino acid deletion. The substitution approach, on the other hand, replaces one or multiple residues with other types of amino acids to test their influence. For example, Clemmons [13] substituted a small domain of the IGF-binding protein to measure whether specific domains account for specific structures and functions. Zhang et al. [64] mutated a specific amino acid on the surface of a Pin1 sub-region, known as the WW domain, and observed significant structural change on the protein structure. Guo et al. [28] randomly replaced amino acids to test proteins' tolerance to substitution at different positions.

When developing our framework, we draw insights from the aforementioned biochemical methods, which were proven effective in wet-lab experiments. We aim to translate the wet-lab methods of understanding protein structures into a dry-lab AI-driven approach. We note that there have been existing attempts which built models to understand the relationship between protein structures and their residues [39, 40, 46]. However, they were mostly based on statistical analysis on wet-lab experimentation data. Our method is the first AI-driven machine learning method developed for understanding protein structure predictions.

2.2 Counterfactual Reasoning for Explainable AI

Counterfactual explanation is a type of model-agnostic explainable AI method that tries to understand the underlying mechanism of a model's behavior by perturbing its input. The basic idea is to investigate the difference of the model's prediction before and after changing the input data in specific ways [57]. Since counterfactual explanation is well-suited for explaining black-box models, it has been an important explainable AI method and has been employed in various applications, such as recommender system [51], computer vision [26, 56], natural language processing [34, 54, 61], graph and molecular analysis [36, 52], and software engineering [12].

In this paper, we explore counterfactual explanation to explain the amino acids' effects on protein folding. However, counterfactual explanation for protein folding has unique challenges compared with previous tasks. For example, 1) most of the aforementioned applications are classification tasks, for which the explanation goal is very clear—find a minimal change on the input that alters the predicted label. However, protein structure prediction is a generation task in a continuous space, which requires careful design of the counterfactual reasoning objective; 2) protein structure prediction models such as AlphaFold take complicated input besides the primary sequence, e.g., the MSA and templates; 3) it is easier to evaluate the explanations for the classification tasks, nevertheless, as a new AI task, protein structure prediction poses unique challenges on the evaluation of explanation. We will show how to overcome these challenges in the following parts of the paper.

3 PROBLEM FORMULATION

In this section, we first provide formulation of the ExplainableFold problem. Given that a protein tertiary (3D) structure is uniquely determined by its primary structure (amino acid sequences) [17, 58], according to the key idea of counterfactual explanation, we define the explanation as identifying the most crucial residues that cause the proteins to fold into the structures they are.

Suppose a protein consists of a chain of l residues, where the i-th residue is encoded as a 21-dimensional one-hot column vector $\boldsymbol{r}_i$. The "1" element in $\boldsymbol{r}_i$ indicates the type of the residue, which can be one of the 20 common amino acids or an additional dimension for unknown residue. By concatenating all the residue vectors, a protein P is denoted as $\boldsymbol{P} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_l]$, where $\boldsymbol{P} \in \{0,1\}^{21\times l}$ is called the protein embedding matrix. Many state-of-the-art protein structure prediction models predict the 3D structure not only based on the residue sequence, but also utilize supplementary evolutionary information [31, 45] by extracting Multiple Sequence Alignment (MSA) [20] from protein databases. Suppose m proteins are retrieved from the evolutionary database based on their similarity with protein P, the constructed MSAs can be encoded as another matrix $\boldsymbol{M}(\boldsymbol{P}) \in \{0,1\}^{m\times 21\times l}$. A protein structure prediction model f_θ predicts the protein 3D structure $\boldsymbol{S}$ based on the residue sequence and MSA embeddings:

$$\boldsymbol{S} = f_\theta\big(\boldsymbol{P}, \boldsymbol{M}(\boldsymbol{P})\big) \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{P})$ can be omitted if the model only takes the residue sequence information. Though a structure prediction model may predict the positions of all atoms, in many structural biology research, only the backbone of residues are used for comparing the similarities of protein structures [59, 62, 65, 66]. Therefore, we adopt the same idea in this paper, where $\boldsymbol{S} \in \mathbb{R}^{3\times l}$ only contains the predicted $(x, y, z)^T$ coordinates of the α-carbon atom of each amino acid residue.

The explanation is expected to be a subset of residues extracted from the protein sequence, expressed as $\mathcal{E}$. The objective of the ExplainableFold problem is to find the **minimum** set of $\mathcal{E}$ that contains the **most influential** information for the prediction of the 3D structure.

4 THE EXPLAINABLEFOLD FRAMEWORK

In biochemistry, the most common methods for studying the effects of amino acids on protein structure fall into two categories: amino acid deletion and substitution [11]. We design the ExplainableFold framework from both of the two perspectives, and we introduce them separately in the following.

4.1 The Residue Deletion Approach

The deletion approach simulates the biochemical studies that detect essential residues for a protein by deleting one or more residues and measuring the protein's tolerance to such deletion [4, 23, 24]. The key idea is to apply a residue mask that removes the effect of certain residues from the sequence and then measure the change of the protein structure. From the counterfactual machine learning perspective, this can be considered from two complementary views [27, 52]: 1) Identify the minimal deletion that will alter the predicted structure and the deleted residues will be the *necessary* explanation; 2) Identify the maximal deletion that still keeps the predicted structure and the undeleted residues will be the *sufficient*

explanation. We design the counterfactual explanation algorithm from these two views accordingly.

4.1.1 ***Necessary Explanation (Most Intolerant Deletion).*** From the necessary perspective, we aim to find the **minimal** set of residues in the original sequence which, if deleted, **will change** the AI model's (such as AlphaFold's) predicted structure. The **deleted** residues thus contain the most necessary information for the model's original prediction.

We can express the perturbation on the original sequence as a multi-hot vector $\Delta = \{0,1\}^{1\times l}$, where $\delta_i = 1$ means that the i-th residue will be deleted and $\delta_i = 0$ means it will be kept. Then the counterfactual protein embedding matrix $\boldsymbol{P}^{\Delta}$ can be expressed as:

$$\boldsymbol{P}^{\Delta} = \boldsymbol{P} \odot (\mathbf{1} - \Delta) + \boldsymbol{U} \odot \Delta \tag{2}$$

where $\odot$ is the element-wise product, $\mathbf{1}$ is an all-1 vector with length l, and $\boldsymbol{U} \in \{0,1\}^{21\times l}$ denotes an "unknown" matrix of the same shape with $\boldsymbol{P}$, but with all elements being 0 except for the last row being 1 (i.e., unknown type amino acid). Thus, for $\delta_i = 0$, the i-th residue in the original sequence will be preserved, while for $\delta_i = 1$, the i-th residue will be treated as an unknown amino acid without any specific chemical property.

Motivated by the Occam's Razor Principle [6], we aim to find *simple* and *effective* explanations. The simpleness can be characterized by the number of residues that need to be deleted, which should be as few as possible, while effectiveness means that the predicted protein structure should be different before and after applying the deletions. We can use zero-norm $\|\Delta\|_0$ to represent the number of deletions (for simpleness), while using the TM-score between the original and the new protein structures $\mathrm{TM}(\boldsymbol{S}, \boldsymbol{S}^*)$ to represent the degree of change on the structure (for effectiveness). TM-score is a standard measurement for comparing aligned protein structures, where TM-score > 0.5 suggests the same folding and TM-score ≤ 0.5 suggests different foldings [59, 65]. The counterfactual explanation algorithm then learns the optimal explanation by solving the following constrained optimization problem:

$$\begin{aligned} &\text{minimize } \|\Delta\|_0 \\ &\text{s.t. } \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S}^*) \leq 0.5,\ \Delta = \{0,1\}^{1\times l} \\ &\text{where } \boldsymbol{S}^* = f_\theta(\boldsymbol{P}^{\Delta}, \boldsymbol{M}(\boldsymbol{P}^{\Delta})) \end{aligned} \tag{3}$$

where the objective $\|\Delta\|_0$ aims to find the minimal deletion, while the constraint guarantees the effectiveness of the deletion, i.e., the deletion will change the predicted protein structure to be different from before.

Due to the exponential combinations of sub-sequences for a given sequence, it is impractical to search for an optimal solution on the discrete space. To solve the problem, we use a continuous relaxation approach to solve the optimization problem by relaxing the multi-hot vector Δ to a real-valued vector. We also relax the hard constraint in Eq.(3) and combine them into a single trainable loss function:

$$\begin{aligned} &\mathcal{L}_1 = \mathrm{LeakyReLU}\big(\mathrm{TM}(\boldsymbol{S}, \boldsymbol{S}^*) - 0.5 + \alpha\big) + \lambda_1 \|\sigma(\Delta)\|_1 \\ &\text{s.t. } \Delta \in \mathbb{R}^{1\times l}, \text{where } \boldsymbol{S}^* = f_\theta(\boldsymbol{P}^{\sigma(\Delta)}, \boldsymbol{M}(\boldsymbol{P}^{\sigma(\Delta)})) \end{aligned} \tag{4}$$

where the LeakyReLU function is $\mathrm{LeakyReLU}(x) = \max(0,x) + \text{negative\_slope}\cdot\min(0,x)$ [37] and we set negative_slope $= 0.1$, the sigmoid function $\sigma(\cdot)$ is applied so that $\sigma(\Delta) \in (0,1)^{1\times l}$ approximates the probability distribution between the original residues and unknown residues, and α is the margin value whose default value is 0.1. This relaxation approach has been justified in several previous studies which also learn explanation on discrete inputs [26, 52]. The 1-norm regularizer assures the learned perturbation $\sigma(\Delta)$ to be sparse [8], i.e., the learned explanation only contains a small set of residues. λ_1 is a hyper-parameter that controls the trade-off between the complexity and strength of the generated explanation.

We use the LeakyReLU loss function as a variation of the original hinge loss function $\mathrm{hinge}(x) = \max(0,x)$ to make the loss optimization process more stable and smooth, which will be explained in the implementation details subsection 5.2. Eq.(4) can be easily optimized with gradient descent. After optimization, we convert $\sigma(\Delta)$ to a binary vector with the threshold of 0.5.

4.1.2 ***Sufficient Explanation (Most Tolerant Deletion).*** Symmetrically, from the sufficiency perspective, we aim to find the **maximal** set of residues in the orignal sequence which, if deleted, **will not change** the AI model's predicted structure. The **undeleted** residues thus contain the most sufficient information for the model's original prediction.

This can be formulated as a similar but reversed optimization process as Eq.(3), which looks for the maximal perturbation Δ while keeping the same folding (TM-score > 0.5). Therefore, the optimization problem is formulated as:

$$\begin{aligned} &\text{maximize } \|\Delta\|_0 \\ &\text{s.t. } \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S}^*) > 0.5,\ \Delta = \{0,1\}^{1\times l} \\ &\text{where } \boldsymbol{S}^* = f_\theta\big(\boldsymbol{P}^{\Delta}, \boldsymbol{M}(\boldsymbol{P}^{\Delta})\big) \end{aligned} \tag{5}$$

Similarly, we relax Eq.(5) to a differentiable loss function:

$$\begin{aligned} &\mathcal{L}_2 = \mathrm{LeakyReLU}\big(0.5 - \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S}^*) + \alpha\big) - \lambda_2 \|\sigma(\Delta)\|_1 \\ &\text{s.t. } \Delta \in \mathbb{R}^{1\times l}, \text{where } \boldsymbol{S}^* = f_\theta(\boldsymbol{P}^{\sigma(\Delta)}, \boldsymbol{M}(\boldsymbol{P}^{\sigma(\Delta)})) \end{aligned} \tag{6}$$

Contrary to the necessary explanation, the sufficient explanation consists the undeleted residues. Hence, after optimization, we filter the residues according to $\big(1 - \sigma(\Delta)\big) > 0.5$ and include them into the sufficient explanation.

4.2 The Residue Substitution Approach

Another popular approach in biochemistry, site-directed substitution, studies the influence of the amino acids on protein folding by replacing certain residues with other known-type residues [5, 7, 23]. Different replacements may have different effects on protein structures, and they can be classified into two types: conservative substitution and radical substitution [15, 63]. A conservative substitution is considered as a "safe" substitution for which the amino acid replacement usually have no or minor effects on the protein structure. A radical substitution is considered "unsafe," which usually causes significant structural changes. Based on the above concepts, we design the substitution approach from these two different perspectives.

4.2.1 ***Radical Substitution Explanation.*** From the radical substitution perspective, we aim to find the **mimimal** set of residue replacements which will lead to a **different** folding, and then the

learned substitutions are the most **radical** substitutions for the protein.

For a target protein with binary embedding matrix $\boldsymbol{P}$, we learn a counterfactual binary protein embedding $\boldsymbol{P'}$, which has the same shape as the original embedding matrix. The number of substitutions is represented by $\|\boldsymbol{P} - \boldsymbol{P'}\|_0$, which is the 0-norm of the difference between the two matrices. To find the minimal residue substitution that changes the original folding, the optimization problem is defined as:

$$\begin{aligned} &\text{minimize } ||\boldsymbol{P} - \boldsymbol{P'}||_0 \\ &\text{s.t. } \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S'}) \leq 0.5,\ \boldsymbol{P'} \in \{0,1\}^{21\times l} \\ &\text{where } \boldsymbol{S'} = f_\theta\big(\boldsymbol{P'}, \boldsymbol{M}(\boldsymbol{P'})\big) \end{aligned} \tag{7}$$

Due to the exponential search space of the substitutions, we use the similar continuous relaxation method as in the deletion approach. First, we relax the binary counterfactual embedding matrix $\boldsymbol{P'}$ to continuous space. We also relax the hard constraint in Eq.(7) and define the differentiable loss function as:

$$\begin{aligned} &\mathcal{L}_3 = \mathrm{LeakyReLU}\big(\mathrm{TM}(\boldsymbol{S}, \boldsymbol{S'}) - 0.5 + \alpha\big) + \lambda_3 \|\boldsymbol{P} - \sigma(\boldsymbol{P'})\|_1 \\ &\text{s.t. } \boldsymbol{P'} \in \mathbb{R}^{21\times l}, \text{where } \boldsymbol{S'} = f_\theta\big(\boldsymbol{P'}, \boldsymbol{M}(\boldsymbol{P'})\big) \end{aligned} \tag{8}$$

After optimization, we convert the learned continuous matrix $\sigma(\boldsymbol{P'})$ into binary by setting the maximum value of each column as 1 and others as 0. Then, the changed residues between $\boldsymbol{P}$ and $\boldsymbol{P'}$ are the radical substitution explanations.

4.2.2 ***Conservative Substitution Explanation.*** From the conservative substitution perspective, we aim to find the **maximal** set of residue replacements which however lead to the **same** folding, and then the learned substitutions are the most **conservative** substitutions for the protein.

On the contrary to Eq.(7), we formulate an inverse optimization problem as:

$$\begin{aligned} &\text{maximize } ||\boldsymbol{P} - \boldsymbol{P'}||_0 \\ &\text{s.t. } \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S'}) > 0.5,\ \boldsymbol{P'} \in \{0,1\}^{21\times l} \\ &\text{where } \boldsymbol{S'} = f_\theta\big(\boldsymbol{P'}, \boldsymbol{M}(\boldsymbol{P'})\big) \end{aligned} \tag{9}$$

With the same relaxation process, the loss function is:

$$\begin{aligned} &\mathcal{L}_4 = \mathrm{LeakyReLU}\big(0.5 - \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S'}) + \alpha\big) - \lambda_4 \|\boldsymbol{P} - \sigma(\boldsymbol{P'})\|_1 \\ &\text{s.t. } \boldsymbol{P'} \in \mathbb{R}^{21\times l}, \text{where } \boldsymbol{S'} = f_\theta\big(\boldsymbol{P'}, \boldsymbol{M}(\boldsymbol{P'})\big) \end{aligned} \tag{10}$$

After learning $\sigma(\boldsymbol{P'})$ and getting the binary matrix, again, the changed residues between $\boldsymbol{P}$ and $\boldsymbol{P'}$ are the conservative substitution explanations.

4.3 Phased MSA Re-alignment

It is impractical to re-compute MSAs in each training step. Therefore, we propose a phased MSA re-alignment strategy. When learning the explanations, we fix the generated MSAs and only learn the changes on the sequence embedding for t training steps ($t = 100$ by default), which is one phase. Then, we re-align the MSAs and start another training phase.

5 EXPERIMENTS

We first introduce the datasets and implementation details. Then, we introduce the evaluation results of the deletion approach and substitution approach, respectively.

5.1 Datasets

We test the ExplainableFold framework on the 14th Critical Assessment of protein Structure Prediction (CASP-14) dataset[1] [43]. CASP consecutively establishes protein data with detailed structural information as a standard evaluation benchmark for protein structure prediction. Following Jumper et al. [31], we remove all sequences for which fewer than 80 amino acids had the alpha carbon resolved and remove duplicated sequences.

5.2 Implementation Details

Though the ExplainableFold framework can be applied on any model that predicts protein 3D structures, we choose Alphafold2 [31], the state-of-the-art model, as the base model in the experiments. More specifically, we use the OpenFold [2] implementation, and load the official pre-trained AlphaFold parameters[2].

When learning the explanations, the pre-trained parameters of AlphaFold are fixed, and only the perturbation vectors on the input (Δ for the deletion approach and $\boldsymbol{P'}$ for the substitution approach) will be optimized. However, it still requires computing the gradient through the entire Alphafold network, as a result, the learning process requires extensive memory consumption. To solve the problem, we follow exactly the same training procedure as introduced in the original AlphaFold paper [31]. More specifically, we use the gradient checkpointing technique to reduce the memory usage [10]. Meanwhile, if a protein has more than 384 residues, we cut it to different chunks for each consecutive 384 residues and generate explanations for each chunk [31]. Except for memory effciency, this also makes it meaningful to compute and compare the explanation size since the maximum sequence length is bounded.

We employ the same training strategy for both deletion and substitution explanation methods: for each training phase between MSA re-alignments, we optimize the perturbation vector for 100 steps with Adam optimizer [33] and learning rate 0.01. After each training loop, we re-align the MSAs with the AlphaFold HHblits / JackHMMER pipeline. We repeat the training and alignment process for 3 phases when generating explanations for each protein. All experiments are conducted on NVIDIA A5000 GPUs. The entire training process, including all 3 phases, for one protein takes approximately 5 hours. We set $\alpha = 0.2$ and $\lambda = 0.00001$, $\lambda = 0.002$, $\lambda = 0.01$, $\lambda = 0.0001$ in Equations (4)(6)(8)(10), respectively. To realize an incremental deletion/substitution process, we initialize the counterfactual protein embedding matrix as a near duplication of the original protein embedding matrix, i.e., we initialize Δ with near 0's and initialize $\boldsymbol{\sigma(P')}$ approximately equal to the original $\boldsymbol{P}$.

However, the above initialization may lead to unstable optimization if we use hinge loss as the loss function. Take Eq.(6) as an example, the initial TM score is close to 1 under this initialization, making the value of $0.5 - \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S^*}) + \alpha$ negative. As a result, if we use hinge loss in Eq.(6), i.e., $\max\big(0, 0.5 - \mathrm{TM}(\boldsymbol{S}, \boldsymbol{S^*}) + \alpha\big) - \lambda_2 \|\sigma(\Delta)\|_1$,

[1] https://predictioncenter.org/casp14/
[2] https://github.com/deepmind/alphafold

Table 1: PN Evaluation. Deletion* is the necessity optimization.

	Ave Explanation Size ($\|\mathcal{E}\|$) ↓	Ave Complexity ($\|\mathcal{E}\|/l$) ↓	Ave TM-score TM(S, S^*) ↓	PN score↑
Random	85.22	0.33	0.83	0.07
Evolutionary [40]	88.42	0.33	0.77	0.16
Deletion (necessity)*	**83.33**	**0.31**	**0.59**	**0.40**

Table 2: PS Evaluation. Deletion* is the sufficiency optimization.

	Ave Explanation Size ($\|\mathcal{E}\|$) ↓	Ave Complexity ($\|\mathcal{E}\|/l$) ↓	Ave TM-score TM(S, S^*) ↑	PS score↑
Random	129.78	0.50	0.44	0.36
Evolutionary [40]	134.89	0.51	0.49	0.42
Deletion (sufficiency)*	**106.25**	**0.49**	**0.65**	**0.76**

then the hinge part of the loss will not take any effect during early phase of the optimization, making the optimization process unstable. Therefore, we use the LeakyReLU function as a varianiton of the original hinge loss function to ensure a stable learning process.

5.3 Evaluation of the Deletion Approach

Counterfactual explanations can be evaluated by their complexity, sufficiency and necessity [25, 52]. First, according to the Occam's Razor Principle [6], we hope an explanation can be as simple as possible so that it is cognitively easy to understand for humans. This can be evaluated by the complexity of the explanation, i.e., the percentage of residues that are included in the explanation:

$$\text{Complexity} = |\mathcal{E}|/l \tag{11}$$

where l is the length of the protein.

Sufficiency and necessity measure how crucial the generated explanations are for the protein structure. We follow the definition in causal inference theory [25] and existing explainable AI research [52] and measure the explanations with two causal metrics: Probability of Necessity (PN) and Probability of Sufficiency (PS).

PN measures the necessity of the explanation. A set of explanation residues is considered a necessary explanation if, by removing their effects from the protein sequence, the predicted structure of the protein will have a different folding (TM-score < 0.5). Suppose there are N proteins in the testing data, then PN is calculated as:

$$\text{PN} = \frac{\sum_{k=1}^{N} \text{PN}_k}{N},\ \text{PN}_k = \begin{cases} 1, \text{if TM}(S_k, S_k^*) \leq 0.5 \\ 0, \text{else} \end{cases} \tag{12}$$

Intuitively, PN measures the percentage of proteins whose explanation residues, if removed, will change the protein structure, and thus their explanation residues are necessary.

PS measures the sufficiency of the explanation. A set of explanation residues is considered a sufficient explanation if, by removing all of the other residues and only keeping the explanation residues, the protein still has the same folding. Similarty, PS is calculated as:

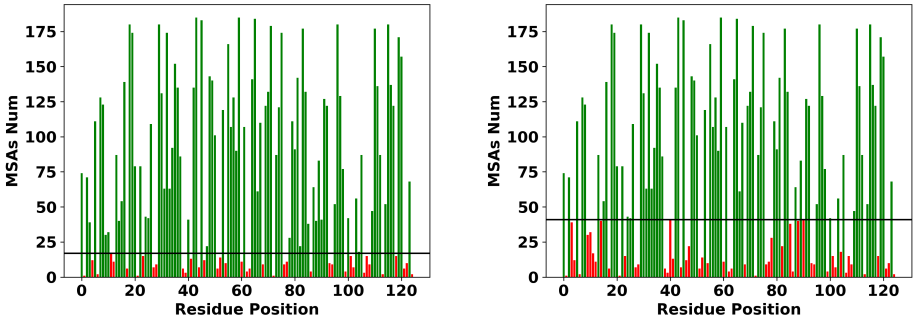
$$\text{PS} = \frac{\sum_{k=1}^{N} \text{PS}_k}{N},\ \text{PS}_k = \begin{cases} 1, \text{if TM}(S_k, S_k^*) > 0.5 \\ 0, \text{else} \end{cases} \tag{13}$$

Intuitively, PS measures the percentage of proteins whose explanation residues alone can keep the protein structure unchanged, and thus their explanation resides are sufficient.

5.3.1 ***Baselines.*** We compare the model performance with a common computational biology baseline [40], which analyzes a protein's tolerance to the change on each residue by extracting the data from evolutionary database. More specifically, proteins are not tolerant to the mutations at evolutionarily conserved positions. However, they are capable of withstanding certain mutations at other positions. When implementing the baseline, we refer to a protein's MSAs and select the evolutionarily conserved residues as the explanation. This is illustrated in Figure 2 using protein CASP14 target T1029 as an example, where for each residue position, we count the number of MSAs that conserve the residue at this position, and show the top 30% and 40% conserved residues. We also randomly select residues as explanation and compute PN and PS scores as another baseline to measure the general difficulty of the evaluation task, and more details are provided in the following subsection.

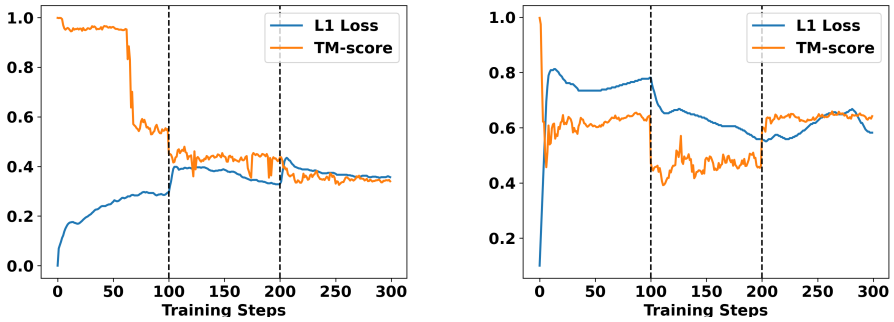
5.3.2 ***Results.*** The results of PN and PS evaluation are reported in Table 1 and Table 2, respectively. The explanation complexities of our method (31% for necessary explanation and 49% for sufficient explanation) are automatically decided by our optimization process. However, the baselines do not have the ability to decide the optimal explanation complexity. For fair comparison, we set the complexities of the baselines to be similar our method (33% for necessary explanation and 50% for sufficient explanation). Therefore, the baselines will have a small advantage over our method because they are allowed to use more residues to achieve the necessity or sufficiency goals.

For PN evaluation, the results of the random baseline shows that protein structures tend to be robust to residue deletions. For example, when randomly removing the effects of 33% residues, only 7% of the proteins fold into different structures, which indicates that finding necessary explanations is a challenging problem. The evolutionary baseline is able to select more necessary residues with a PN score of 0.16, which is 128.6% better than random selection. Compared to them, our method shows much better performance:



(a) Top 30% conserved residues (b) Top 40% conserved residues

Figure 2: Evolutionary conserved residues are considered more important for the protein structure (the residues in red).



(a) Necessary Optimization (b) Sufficient Optimization

Figure 3: Learning Curves of the Deletion Approach, including three phases between MSA re-alignment, 100 traing steps each.

with a smaller number of residues, the generated explanations are able to cause 41% of the proteins fold into different structures, outperforming the evolutionary baseline by 150%.

For PS evaluation, the evolutionary baseline is not noticeably better than randomly selecting residues. The reason may be that despite the proteins' less tolerance to the evolutionary conserved residues, there is no guarantee that the evolutionary conserved residues alone contain sufficient information to preserve the protein structure. In comparison, our method does generate more sufficient explanations, outperforming the evolutionary baseline by 81.0% according to the PS score with less complex explanations. Meanwhile, our TM score is $> 50\%$, indicating that the protein structure is indeed preserved under our sufficient explanation.

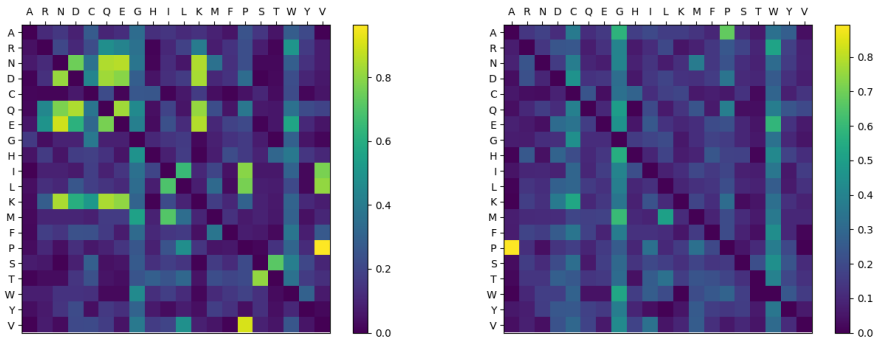
Additionally, we show the learning curve of the optimization for CASP14 target protein T1030 in Figure 3. For necessary optimization, the algorithm gradually deletes the protein residues until reaching a TM-score near 0.3 (i.e., $0.5 - \alpha$, see Eq.(4)). Then, the explanation complexity slightly drops back while keeping the TM-score at the same level. During sufficient optimization, the L1-loss drastically increases initially, which suggests that the algorithm is trying to delete as many residues as possible while keeping the original folding structure unchanged. However, after re-computing MSAs, the TM-score becomes too low. Thus, the algorithm increases the number of preserved residues to keep TM-score near 0.7 (i.e., $0.5 + \alpha$, see Eq.(6)). Note that the TM-scores change sharply when re-computing MSAs at the end of each training loop. More frequent MSA realignments result in a smoother optimization process.

5.4 Evaluation of the Substitution Approach

The substitution approach identifies the most radical or conservative amino acid substitutions, which are of particular interest in biochemical research [63]. Previously, it was impractical to conduct wet-lab experiments to investigate the relative "safety" of replacing specific residues with alternative amino acids due to their prohibitive cost [7]. Alternatively, scientists infer the *exchangeability* of two types of amino acids either through the use of heuristics based on their physical or chemical properties or through the analysis of evolutionary data, such as:

- Epstein's distance [22]: Predict the impact of switching two amino acids based on their size and polarity.
- Miyata's distance [41]: Predict the impact based on their volume and polarity.
- Evolutionary indicator [7]: Detect "safe" substitutions based on evolutionary data.

Note that these indicators are rather suggestions than ground-truth. They provide general trends that are better than random selection but cannot be expected to be precise in every scenario [7]. These methods are not perfectly consistent with each other, but are linearly related.



(a) Correlation with Epstein's distance (b) Correlation with Miyata's distance

Figure 4: The correlation between the exchangeability provided by our conservative optimization method and (a) Epstein's distance as well as (b) Miyata's distance.

Table 3: Correlation between our method and each of the biochemical indicators. Metrics with "*" are originally distance metrics, for which we take the inverse to reprenst the exchangeability. The results are significant at $p < 0.001$ under two-tailed test.

	Epstein*	Miyata*	Evolutionary
Radical	0.388	0.602	0.382
Conservative	0.494	0.796	0.405

Therefore, we utilize the amino acid substitution data generated by our method to caculate the pair-wise exchangeability between the amino acids, and test the correlation between our exchangeability with the above three existing exchangeability indicators. The details of the pair-wise substitution statistics and the calculation of pair-wise exchangeability are provided in the Appendix.

In Table 3, we report the correlation of our generated pair-wise exchangeability with the three aforementioned indicators by a non-parametric method: Pearson's correlation r. Besides, the correlation among the three biochemical methods themselves range from 0.438 to 0.578. Additionally, the correlation is visualized in Figure 4, where darker color indicates higher correlation. For Pearson's correlation, a value greater than 0 indicates a positive association, where $r > 0.1$, $r > 0.3$, $r > 0.5$ represents small, medium, and large correlations, accordingly [14]. Both Table 3 and Figure 4 show that our method has clear positive correlations with all of the three biochemical methods, indicating that ExplainableFold can provide informative exchangeability signals [60]. Besides, the results generated by ExplainableFold may further improve when larger protein datasets are available or applied on even better base models in the future.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose ExplainableFold—an Explainable AI framework that helps to understand the deep learning based protein structure prediction models such as AlphaFold. Technically, we develop a counterfactual explanation framework and implement the framework based on two approaches: the residue deletion approach and the residue substitution approach. Intuitively, ExplainableFold aims to find simple explanations that are effective enough to keep or change the protein's folding structure. Experiments are conducted on CASP-14 protein datasets and results show that our approach outperforms the results from traditional biochemical methods. We believe Explainable AI is fundamentally important for AI-driven scientific research because science not only pursues the answers for the "what" questions but also (or even more) for the "why" questions. In the future, we will further improve our framework by considering more protein modification methods beyond deletion and substitution. We will also generalize our framework to other scientific problems due to the flexibility of our framework.

ACKNOWLEDGEMENT

We thank the reviewers for their constructive suggestions. The work was supported in part by NSF IIS 1910154, 2007907, 2046457, 2127918 and NIH CA277812-02.

APPENDIX

A STATISTICAL ANALYSIS OF AMINO ACID SUBSTITUTIONS

Table 4 shows the total count of each amino acid in the testing proteins. In Table 5, we show how many times a specific type of substitution happens in the generated explanations learned by the conservative substitution method. For instance, the substitution of $A \rightarrow R$ happens 19 times. The exchangeability of $X \rightarrow Y$ can be easily calculated by $|X \rightarrow Y|/|X|$ [7, 40]. The same statistics for radical substitution is provided in Table 6. For radical substitution, the higher the number in Table 6, the lower the exchangeability, and thus the exchangeability of $X \rightarrow Y$ is calculated as the reciprocal $|X|/|X \rightarrow Y|$ [7, 40].

Table 4: Total number of each amino acid in testing data

	A	R	N	D	C	Q	E	G	H	I
#	782	581	827	776	175	520	848	853	309	904
	L	K	M	F	P	S	T	W	Y	V
#	1122	911	273	600	506	916	746	149	595	780

Table 5: Structural Conservative Statistics

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	0	19	25	16	50	18	21	78	23	26	22	22	19	14	39	28	15	42	35	8
R	8	0	15	23	23	11	15	37	12	12	19	18	9	15	23	18	11	49	18	9
N	15	33	0	32	51	19	18	56	16	19	11	19	50	21	33	21	16	42	22	14
D	7	29	22	0	57	21	25	42	11	19	23	19	16	21	44	16	15	32	16	11
C	2	1	1	2	0	7	1	9	8	4	5	5	2	2	2	4	0	8	2	5
Q	5	12	18	12	33	0	14	42	15	18	7	15	21	7	33	5	7	32	21	18
E	14	14	14	50	47	30	0	62	15	35	19	19	18	29	32	16	11	79	19	11
G	18	11	19	19	62	18	18	0	23	26	29	9	18	25	29	22	12	46	15	9
H	0	15	4	15	11	14	9	28	0	5	9	7	5	12	9	9	2	21	9	7
I	9	16	16	14	46	16	25	58	33	0	49	19	56	35	29	14	14	54	18	32
L	5	28	23	50	57	30	25	70	21	53	0	19	47	40	40	21	19	51	22	33
K	2	44	22	56	78	22	28	51	22	35	18	0	29	19	47	7	11	49	21	15
M	4	5	5	5	9	8	8	26	5	12	28	5	0	7	9	5	4	16	7	8
F	8	19	16	25	35	18	9	36	14	21	19	7	21	0	21	9	5	46	21	2
P	8	11	5	12	16	9	14	25	9	28	9	14	28	16	0	4	14	30	16	2
S	21	26	22	33	49	14	28	53	23	30	26	21	29	22	36	0	40	65	36	19
T	9	19	21	35	33	22	21	40	9	33	42	22	30	28	29	22	0	47	25	19
W	0	2	4	4	7	1	1	12	4	7	5	0	5	7	7	4	0	0	7	4
Y	7	9	16	23	25	18	15	36	11	9	5	7	29	26	15	16	8	37	0	9
V	8	12	7	29	44	25	9	54	25	49	14	14	43	19	11	15	11	40	18	0

Table 6: Structural Radical Statistics

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	0	28	22	16	39	5	19	22	30	28	25	5	25	22	33	14	14	64	16	14
R	8	0	11	33	19	5	28	22	16	25	8	2	11	36	25	5	5	33	11	11
N	11	16	0	8	47	11	22	16	14	19	25	22	19	25	25	5	8	25	14	19
D	11	11	11	0	25	14	22	16	19	25	11	16	44	16	16	11	0	22	14	25
C	2	0	0	0	0	2	2	11	2	0	8	8	2	2	5	8	5	2	0	5
Q	2	19	5	11	25	0	22	16	14	14	14	16	14	16	25	5	5	19	8	8
E	5	11	5	19	58	5	0	22	14	19	14	11	28	36	19	5	8	39	25	19
G	2	25	2	16	56	11	14	0	8	33	28	22	53	28	14	14	5	44	22	8
H	2	5	0	0	16	14	8	11	0	11	8	8	0	0	14	2	0	5	5	14
I	25	28	22	36	33	5	22	44	5	0	2	8	16	8	30	11	11	47	5	11
L	22	28	25	30	64	28	28	72	19	25	0	47	33	11	56	28	22	36	28	19
K	14	2	8	25	81	22	19	30	11	14	8	0	16	30	33	5	19	64	19	19
M	2	0	16	11	5	2	2	25	8	8	0	8	0	2	19	2	0	14	5	5
F	16	11	11	22	28	8	28	19	11	16	11	19	14	0	39	19	2	11	8	5
P	8	11	2	19	22	5	16	11	8	2	14	16	36	14	0	2	5	36	22	8
S	22	8	5	14	44	16	22	30	22	28	28	28	25	22	25	0	16	58	16	14
T	11	14	16	22	56	8	19	42	14	5	19	8	33	22	19	14	0	58	16	11
W	2	0	0	2	2	8	2	14	2	2	0	0	2	2	5	5	2	0	2	8
Y	25	14	2	5	28	5	8	25	16	11	5	19	25	8	19	8	8	14	0	11
V	8	19	16	36	25	19	30	53	14	8	11	11	44	16	19	11	8	56	8	0

REFERENCES

[1] Gary K Ackers and Francine R Smith. Effects of site-specific amino acid modification on protein interactions and biological function. *Annual review of biochemistry*, 54(1):597–629, 1985.

[2] Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, Arkadiusz Nowaczynski, Bei Wang, Marta M Stepniewska-Dziubinska, Shang Zhang, Adegoke Ojewole, Murat Efe Guney, Stella Biderman, Andrew M Watkins, Stephen Ra, Pablo Ribalta Lorenzo, Lucas Nivon, Brian Weitzner, Yih-En Andrew Ban, Peter K Sorger, Emad Mostaque, Zhao Zhang, Richard Bonneau, and Mohammed AlQuraishi. Openfold: Retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *bioRxiv*, 2022. doi: 10.1101/2022.11.20.517210.

[3] Mohammed AlQuraishi. Machine learning in protein structure prediction. *Current opinion in chemical biology*, 65:1–8, 2021.

[4] James AJ Arpino, Samuel C Reddington, Lisa M Halliwell, Pierre J Rizkallah, and D Dafydd Jones. Random single amino acid deletion sampling unveils structural tolerance and the benefits of helical registry shift on gfp folding and structure. *Structure*, 22(6):889–898, 2014.

[5] Matthew J Betts and Robert B Russell. Amino acid properties and consequences of substitutions. *Bioinformatics for geneticists*, 317:289, 2003.

[6] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[7] Domenico Bordo and Patrick Argos. Suggestions for "safe" residue substitutions in site-directed mutagenesis. *Journal of molecular biology*, 217(4):721–729, 1991.

[8] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[9] Paul Carter. Site-directed mutagenesis. *Biochemical Journal*, 237(1):1, 1986.

[10] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[11] Yongwook Choi and Agnes P Chan. Provean web server: a tool to predict the functional effect of amino acid substitutions and indels. *Bioinformatics*, 31(16): 2745–2747, 2015.

[12] Jürgen Cito, Isil Dillig, Vijayaraghavan Murali, and Satish Chandra. Counterfactual explanations for models of code. In *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*, pages 125–134, 2022.

[13] David R Clemmons. Use of mutagenesis to probe igf-binding protein structure/function relationships. *Endocrine reviews*, 22(6):800–817, 2001.

[14] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

[15] Tal Dagan, Yael Talmor, and Dan Graur. Ratios of radical to conservative amino acid replacement are affected by mutational and compositional factors and may not be indicative of positive darwinian selection. *Molecular biology and evolution*, 19(7):1022–1025, 2002.

[16] Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *science*, 338(6110):1042–1046, 2012.

[17] Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. The protein folding problem. *Annual review of biophysics*, 37:289, 2008.

[18] Jon A Doering, Sehan Lee, Kurt Kristiansen, Linn Evenseth, Mace G Barron, Ingebrigt Sylte, and Carlie A LaLone. In silico site-directed mutagenesis informs species-specific predictions of chemical susceptibility derived from the sequence alignment to predict across species susceptibility (seqapass) tool. *Toxicological Sciences*, 166(1):131–145, 2018.

[19] Clifford N Dominy and David W Andrews. Site-directed mutagenesis by inverse pcr. In *E. coli Plasmid Vectors*, pages 209–223. Springer, 2003.

[20] Robert C Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373, 2006.

[21] Martin Egli, Andy Flavell, Anna Marie Pyle, W David Wilson, S Ihtshamul Haq, Ben Luisi, Julie Fisher, Charlie Laughton, Stephanie Allen, and Joachim Engels. *Chapter 5.6 Nucleic Acids in Biotechnology*. The Royal Society of Chemistry, 2006. ISBN 978-0-85404-654-6. doi: 10.1039/9781847555380.

[22] Charles J Epstein. Non-randomness of ammo-acid changes in the evolution of homologous proteins. *Nature*, 215(5099):355–359, 1967.

[23] Gabriela Flores-Ramírez, Manuel Rivera, Alfredo Morales-Pablos, Joel Osuna, Xavier Soberón, and Paul Gaytán. The effect of amino acid deletions and substitutions in the longest loop of gfp. *BMC chemical biology*, 7(1):1–10, 2007.

[24] Anton Glück and Ira G Wool. Analysis by systematic deletion of amino acids of the action of the ribotoxin restrictocin. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, 1594(1):115–126, 2002.

[25] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[26] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[27] Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019.

[28] Haiwei H Guo, Juno Choe, and Lawrence A Loeb. Protein tolerance to random amino acid change. *Proceedings of the National Academy of Sciences*, 101(25): 9205–9210, 2004.

[29] Clyde A Hutchison, Sandra Phillips, Marshall H Edgell, Shirley Gillam, Patricia Jahnke, and Michael Smith. Mutagenesis at a specific position in a dna sequence. *Journal of Biological Chemistry*, 253(18):6551–6560, 1978.

[30] Andrea Ilari and Carmelinda Savino. Protein structure determination by x-ray crystallography. *Bioinformatics*, pages 63–87, 2008.

[31] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[32] Gregory J Kato, Frédéric B Piel, Clarice D Reid, Marilyn H Gaston, Kwaku Ohene-Frempong, Lakshmanan Krishnamurti, Wally R Smith, Julie A Panepinto, David J Weatherall, Fernando F Costa, et al. Sickle cell disease. *Nature Reviews Disease Primers*, 4(1):1–22, 2018.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Orestis Lampridis, Riccardo Guidotti, and Salvatore Ruggieri. Explaining sentiment classification with synthetic exemplars and counter-exemplars. In *International Conference on Discovery Science*, pages 357–373. Springer, 2020.

[35] Zelong Li, Jianchao Ji, and Yongfeng Zhang. From Kepler to Newton: Explainable AI for Science Discovery. In *ICML 2022 2nd AI for Science Workshop*, 2022.

[36] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021.

[37] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA, 2013.

[38] Dailén G Martínez, Stefan Hüttelmaier, and Jean B Bertoldo. Unveiling druggable pockets by site-specific protein modification: Beyond antibody-drug conjugates. *Frontiers in Chemistry*, 8:586942, 2020.

[39] Majid Masso and Iosif I Vaisman. Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. *Bioinformatics*, 24(18):2002–2009, 2008.

[40] Majid Masso, Zhibin Lu, and Iosif I Vaisman. Computational mutagenesis studies of protein structure-function correlations. *Proteins: Structure, Function, and Bioinformatics*, 64(1):234–245, 2006.

[41] Takashi Miyata, Sanzo Miyazawa, and Teruo Yasunaga. Two types of amino acid substitutions in protein evolution. *Journal of molecular evolution*, 12:219–236, 1979.

[42] Ken Motohashi. A simple and efficient seamless dna cloning method using slice from escherichia coli laboratory strains and its application to slip site-directed mutagenesis. *BMC biotechnology*, 15(1):1–9, 2015.

[43] J Moult, K Fidelis, A Kryshtafovych, T Schwede, and M Topf. Critical assessment of techniques for protein structure prediction, fourteenth round. *CASP 14 Abstract Book*.

[44] Gobinda Sarkar and Steve S Sommer. The" megaprimer" method of site-directed mutagenesis. *Biotechniques*, 8(4):404–407, 1990.

[45] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[46] Cristina Sotomayor-Vivas, Enrique Hernández-Lemus, and Rodrigo Dorantes-Gilardi. Linking protein structural and functional change to mutation using amino acid networks. *Plos one*, 17(1):e0261829, 2022.

[47] Christopher D Spicer and Benjamin G Davis. Selective chemical protein modification. *Nature communications*, 5(1):1–14, 2014.

[48] Peter D Stenson, Matthew Mort, Edward V Ball, Katy Evans, Matthew Hayden, Sally Heywood, Michelle Hussain, Andrew D Phillips, and David N Cooper. The human gene mutation database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies. *Human genetics*, 136:665–677, 2017.

[49] Romain A Studer, Benoit H Dessailly, and Christine A Orengo. Residue mutations and their impact on protein structure and function: detecting beneficial and pathogenic changes. *Biochemical journal*, 449(3):581–594, 2013.

[50] David E Szymkowski. Creating the next generation of protein therapeutics through rational drug design. *CURRENT OPINION IN DRUG DISCOVERY AND DEVELOPMENT*, 8(5):590, 2005.

[51] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1784–1793, 2021.

[52] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 1018–1027, 2022.

[53] Yi Tan, Hongxiang Wu, Tongyao Wei, and Xuechen Li. Chemical protein synthesis: advances, challenges, and outlooks. *Journal of the American Chemical Society*, 142(48):20288–20298, 2020.

[54] George Tolkachev, Stephen Mell, Stephan Zdancewic, and Osbert Bastani. Counterfactual explanations for natural language interfaces. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 113–118, 2022.

[55] Mirko Torrisi, Gianluca Pollastri, and Quan Le. Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal*, 18: 1301–1310, 2020.

[56] Tom Vermeire, Dieter Brughmans, Sofie Goethals, Raphael Mazzine Barbossa de Oliveira, and David Martens. Explainable image classification with evidence counterfactual. *Pattern Analysis and Applications*, pages 1–21, 2022.

[57] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[58] Marco Wiltgen. Structural bioinformatics: From the sequence to structure and function. *Current Bioinformatics*, 4:54–87, 01 2009. doi: 10.2174/157489309787158170.

[59] Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

[60] Lev Y Yampolsky and Arlin Stoltzfus. The exchangeability of amino acids in proteins. *Genetics*, 170(4):1459–1472, 2005.

[61] Linyi Yang, Eoin M Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. Generating plausible counterfactual explanations for deep transformers in financial text classification. *arXiv preprint arXiv:2010.12512*, 2020.
[62] Adam Zemla. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.
[63] Jianzhi Zhang. Rates of conservative and radical nonsynonymous nucleotide substitutions in mammalian nuclear genes. *Journal of molecular evolution*, 50(1): 56–68, 2000.
[64] Meiling Zhang, David A Case, and Jeffrey W Peng. Propagated perturbations from a peripheral mutation show interactions supporting ww domain thermostability. *Structure*, 26(11):1474–1485, 2018.
[65] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.
[66] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.