OXFORD

Briefings in Bioinformatics, 2023, **24(5)**, 1–13
https://doi.org/10.1093/bib/bbad289
Advance access publication date 14 August 2023
Review

Artificial intelligence-aided protein engineering: from topological data analysis to deep protein language models

Yuchi Qiu and Guo-Wei Wei

Corresponding author. Guo-Wei Wei, Department of Mathematics, Michigan State University, East Lansing, 48824, MI, USA. E-mail: weig@msu.edu

Abstract

Protein engineering is an emerging field in biotechnology that has the potential to revolutionize various areas, such as antibody design, drug discovery, food security, ecology, and more. However, the mutational space involved is too vast to be handled through experimental means alone. Leveraging accumulative protein databases, machine learning (ML) models, particularly those based on natural language processing (NLP), have considerably expedited protein engineering. Moreover, advances in topological data analysis (TDA) and artificial intelligence-based protein structure prediction, such as AlphaFold2, have made more powerful structure-based ML-assisted protein engineering strategies possible. This review aims to offer a comprehensive, systematic, and indispensable set of methodological components, including TDA and NLP, for protein engineering and to facilitate their future development.

Keywords: topological data analysis, protein language models, protein engineering, deep learning and machine learning

INTRODUCTION

Protein engineering aims to design and discover proteins with desirable functions, such as improving the phenotype of living organisms, enhancing enzyme catalysis, and boosting antibody efficacy [1]. It has tremendous impacts on drug discovery, enzyme development and applications, the development of biosensors, diagnostics, and other biotechnology, as well as understanding the fundamental principles of the protein structure-function relationship and achieving environmental sustainability and diversity. Protein engineering has the potential to continue to drive innovation and improve our lives in the future.

Two traditional protein engineering approaches include directed evolution [2] and rational design [3, 4]. Directed evolution is a process used to create proteins or enzymes with improved or novel functions [5]. The method involves introducing mutations into the genetic code of a target protein and screening the resulting variants for improved function. The process is 'directed' because it is guided by the desired outcome, such as increased activity, stability, specificity, binding affinity, and fitness. Rational design involves using knowledge of protein structure and function to engineer desirable specific changes to the protein sequence and/or structure [4, 6]. Both approaches resort to experimental screening of astronomically large mutational space, i.e. 20^N for protein of N amino acid residues, which is expensive, time-consuming, and intractable [7]. As a result, only a small fraction of the mutational space can be explored experimentally even with the most advanced high-throughput screening technology.

Recently, data-driven machine learning has emerged as a new approach for directed evolution and protein engineering [8, 9]. Machine learning-assisted protein engineering (MLPE) refers to the use of machine learning models and techniques to improve the efficiency and effectiveness of protein engineering. MLPE not only reduces the cost and expedites the process of protein engineering, but also optimizes the screening and selection of protein variants [10], leading to the higher efficiency and productivity. Specifically, by using machine learning to analyze and predict the effects of mutations on protein function, researchers can rapidly generate and test large numbers of variants, which establish the protein-to-fitness map (i.e. fitness landscape) from sparsely sampled experimental data [11, 12]. This approach accelerates the process of protein engineering.

The process of data-driven MLPE typically involves several elements, including data collection and preprocessing, model design, feature extraction and selection, algorithm selection and design, model training and validation, experimental validation, and iterative model optimization. Driven by technological advancements in high-throughput sequencing and screening technologies, there has been a substantial accumulation of general-purpose experimental datasets on protein sequences, structures, and functions [13, 14]. These datasets, along with numerous protein-engineering

Yuchi Qiu is a postdoc at Department of Mathematics in Michigan State University. He received his Ph.D. in mathematics from University of California, Irvine, in 2020. His research interests include computational and mathematical biology, and artificial intelligence.
Guo-Wei Wei is an MSU foundation professor at Department of Mathematics, Department of Biochemistry and Molecular Biology, and Department of Electrical and Computer Engineering in Michigan State University. He received his Ph.D. from University of British Columbia in 1996. His research interests include mathematical foundations of data science and biosciences, deep learning, drug discovery, and computational geometry, topology, and graph.
Received: May 8, 2023. **Revised:** July 14, 2023. **Accepted:** July 26, 2023

specific deep mutational scanning (DMS) libraries [15], provide valuable resources for machine learning training and validation.

Data representation and feature extraction are crucial steps in the design of machine learning models, as they help to reduce the complexity of biological data and enable more effective model training and prediction. There are several typical types of feature embedding methods, including sequence-based, structure-based [16, 17], physics-based [18, 19], and hybrid methods [20]. Among them, sequence-based embeddings have been dominant due to the success of various natural language processing (NLP) methods such as long short-term memory (LSTM) [21], autoencoders [22], and Transformers [23], which allow unsupervised pre-training on large-scale sequence data. Structure-based embeddings take advantage of existing protein three-dimensional (3D) structures in the Protein Data Bank (PDB) [13] and advanced structure predictions such as AlphaFold2 [24]. These methods further exploit advanced mathematical tools, such as topological data analysis (TDA) [25, 26], differential geometry [27, 28], or graph approaches [29]. Physics-based methods utilize physical models, such as density functional theory [30], molecular mechanics [31], Poisson-Boltzmann model [32], etc. While these methods are highly interpretable, their performance often depends on model parametrization. Hybrid methods may select a combination of two or more types of features.

The designs and selections of MLPE algorithms depend on the availability of data and efficiency of experiments. In real-world scenarios, small labeled training datasets are prevalent, and as a result, simple machine learning algorithms such as support vector machines and ensemble methods are often employed for small training datasets. In contrast, deep neural networks are more suitable for large training datasets. In addition to regression models, unsupervised zero-shot learning methods can also be utilized to address scenarios with limited labeled data availability [33, 34]. The iterative interplay between experiments and models is another crucial component in MLPE by iteratively screening new data to refine the models. Consequently, the selection of an appropriate MLPE model is influenced by factors like experimental frequency and throughput. This iterative refinement process enables MLPE to deliver optimized protein engineering outcomes.

MLPE has the potential to significantly accelerate the development of new and improved proteins, revolutionizing numerous areas of science and technology (Figure 1). Despite considerable advances in MLPE, challenges remain in many aspects, such as data preprocessing, feature extraction, integration with advanced algorithms, and iterative optimization through experimental validation. This review examines published works and offers insights into these technical advances. First, we review current advanced NLP-based models and efficient MLPE approaches. Then we place particular emphasis on the advanced mathematical TDA approaches, aiming to make them accessible to general readers. Last, we discuss potential future directions in the field.

SEQUENCE-BASED DEEP PROTEIN LANGUAGE MODELS

In artificial intelligence, natural language processing (NLP) has recently gained much attention for representing and analyzing human language computationally [35]. NLP covers a wide range of tasks, including language translation, sentiment analysis, chatbot development, speech recognition, and information extraction, among others. The development and advancement of various machine learning models have been instrumental in tackling the complex challenges posed by NLP tasks.

Similar to human language, the primary structure of a protein is also represented by a string of amino acids, with 20 canonical amino acids. The analogy between protein sequences and human languages has inspired the development of computational methods for analyzing and understanding proteins using models adopted from NLP (Figure 1A). The self-supervised sequence-based protein language models have been applied to study the underlying patterns and relationships within protein sequences, predict their structural and functional properties, and facilitate protein engineering. These language models are pretrained on a given data allowing to model protein properties for each given protein. There are two major types of protein language models utilizing different resources of protein data [33] (Table 1). The first one is the local evolutionary models which focus on homologs of the target protein such as multiple sequence alignments (MSAs) to learn the evolutionary information from the mostly related mutations. The second one is the Table 1 global evolutionary models which learn from large protein sequence databases such as UniProt [14] and Pfam [36].

Local evolutionary models

To train a local evolutionary model, MSAs search strategies such as jackhmmer [51] and EvCouplings [52] are first employed. Taking MSAs as inputs, local evolutionary models learn the probabilistic distribution of mutations for a target protein. Probabilistic models, including Hidden Markov Models (HMMs) [37, 53] and Potts-based models [38], are popular in modeling mutational effects. Transformer models have been introduced to learn distribution from MSAs. The MSA Transformer [39] introduces a row- and column-attention mechanism. Recent years, variational autoencoders (VAEs) [54] serve as the alternate to model MSAs by including the dependency between residues and aligning all sequences to a probability distribution. The VAE model DeepSequence [22] and the Bayesian VAE model EVE [40] exhibit excellent performance in modeling mutational effects [20, 33, 55].

Global evolutionary models

With large-size data, global evolutionary models usually adopt the large NLP models. Convolutional Neural networks (CNNs) [56] models and residual network (ResNet) [57] have been employed for protein sequence analysis [41]. Large-scale models, such as long short-term memory (LSTM) [58], have also gained popularity as seen in Bepler [42], UniRep [21], and eUniRep [43]. In recent years, the Transformer architecture has achieved state-of-the-art performance in NLP by introducing the attention mechanism and the self-supervised learning via the masked filling training strategy [59, 60]. Inspired by these advances, Transformer-based protein language models provide new opportunities for building global evolutionary models. A variety of Transformer-based models have been developed such as evolutionary scale modeling (ESM) [23, 44], ProGen [47], ProteinBERT [49], Tranception [15] and ESM-2 [50].

Hybrid approach via fine-tune pre-training

Although global evolutionary models can learn a variety of sequences derived from natural evolution, they face challenges in concentrating on local information when predicting the effects of site-specific mutations of a target protein. To enhance the performance of global evolutionary models, fine-tuning strategies are subsequently implemented. Specifically, fine-tune strategy further refines the pre-trained global models with local information using MSAs or target training data. The fine-tuned eUniRep [43] shows significant improvement over UniRep [21].

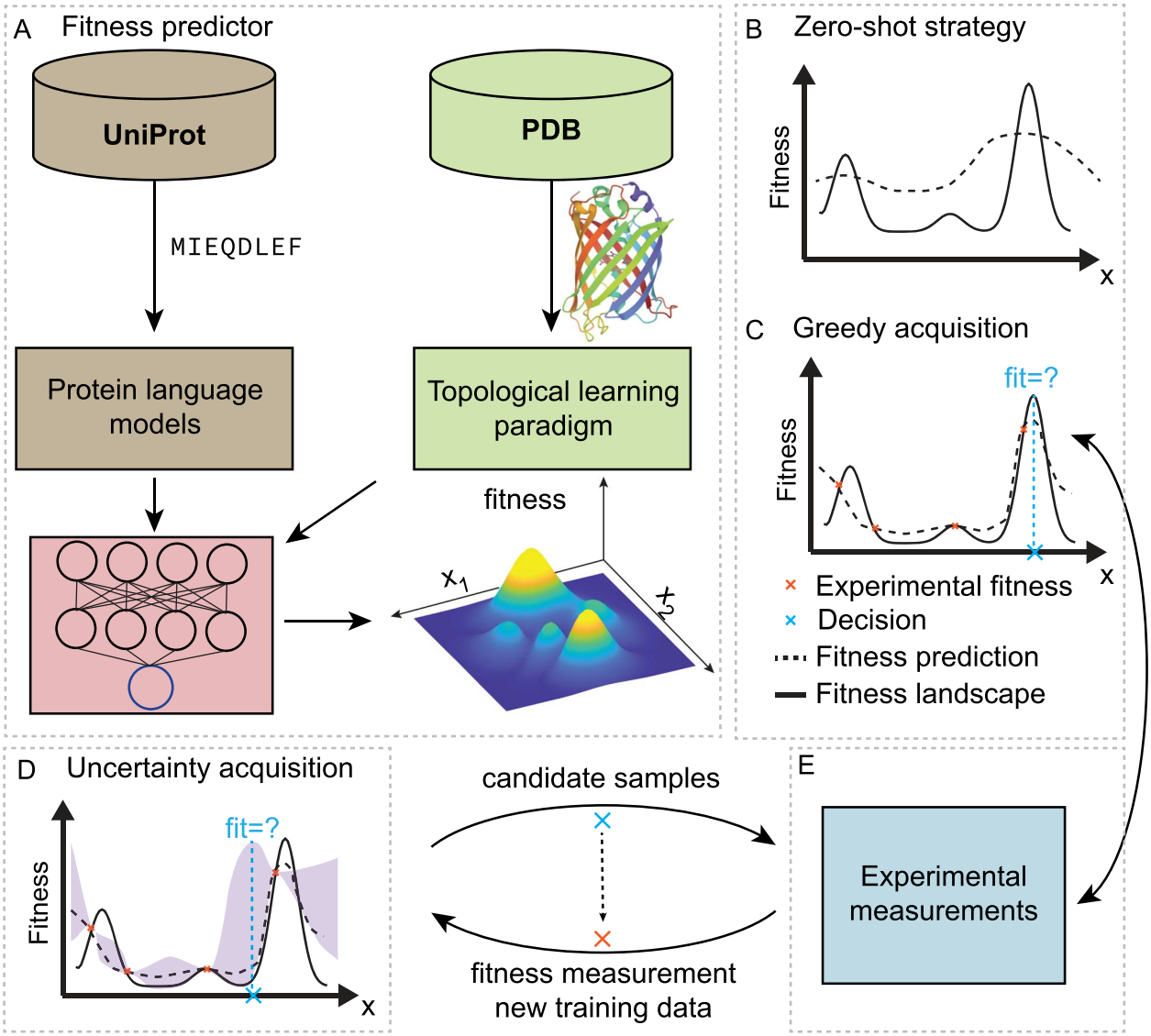

Figure 1. Machine learning-assisted protein engineering (MLPE). (**A**). Machine learning models build fitness predictor using structure and sequence protein data. (**B**). Zero-shot predictors navigate fitness landscape without labeled data. (**C**). Greedy acquisition exploits fitness using fitness predictions. (**D**). Uncertainty acquisition balances exploitation and exploration. The example shows a Gaussian upper confidence bound (UCB) acquisition. (**E**). Experimental measurements query fitness of candidate proteins in sequential optimization.

Similar improvement was also reported for ESM models [23, 44]. The Tranception model proposed a hybrid approach combining a global autoregressive inference and a local retrieval inference from MSAs [15]. Tranception achieved the advanced performance over other global and local models.

With various language models proposed, comprehensive studies on various models and the strategy in building downstream model is necessary. A study explored different approaches utilizing the sequence embedding to build downstream models [61]. Two other studies further benchmarked many unsupervised and supervised models in predicting protein fitness [33, 55].

STRUCTURE-BASED TOPOLOGICAL DATA ANALYSIS (TDA) MODELS

Aided by advanced NLP algorithms, sequence-based models have become the dominant approach in MLPE [11, 12]. However, sequence-based models suffer from a lack of appropriate description of stereochemical information, such as cis-trans isomerism, conformational isomerism, enantiomers, etc. Therefore, sequence embeddings cannot distinguish stereoisomers, which are widely present in biological systems and play a crucial role in many chemical and biological processes. Structure-based models offer a solution to this problem. TDA has became a successful tool in building structure-based models for MLPE [20].

TDA is a mathematical framework based on algebraic topology [62, 63], which allows us to characterize complex geometric data, identify underlying geometric shapes, and uncover topological structures present in the data. TDA finds its applications in a wide range of fields, including neuroscience, biology, materials science, and computer vision. It is especially useful in situations where the data is complex, high-dimensional, and noisy, and where traditional statistical methods may not be effective. In this section, we provide an overview of various types of TDA methods (Table 2). In addition, we review graph neural networks, which

Table 1. Summary of protein language models. # para: number of parameters which are only provided for deep learning models. Max len: maximum length of input sequence. Dim: latent space dimension. Size: pre-trained data size where it refers to number of sequences without specification except MSA transformer includes 26 millions of MSAs. K: thousands; M: millions; B: billions. [1]: Time for the first preprint. The input data size, hidden layer dimension, and number of parameters are only provided for global models

Model	Architecture	Max len	Dim	# para	Pretrained data		Time[1]
					Source	Size	
Local models							
Profile HMMs [37]	Hidden Markov	–	–	–	MSAs	–	Oct 2012
EvMutation [38]	Potts models	–	–	–	MSAs	–	Jan 2017
MSA transformer [39]	Transformer	1024	768	100M	UniRef50 [14]	26M	Feb 2021
DeepSequence [22]	VAEs	–	–	–	MSAs	–	Dec 2017
EVE [40]	Bayesian VAEs	–	–	–	MSAs	–	Oct 2021
Global models							
TAPE ResNet [41]	ResNet	1024	256	38M	Pfam [36]	31M	Jun 2019
TAPE LSTM [41]	LSTM	1024	2048	38M	Pfam [36]	31M	Jun 2019
TAPE transformer [41]	Transformer	1024	512	38M	Pfam [36]	31M	Jun 2019
Bepler [42]	LSTM	512	100	22M	Pfam [36]	31M	Feb 2019
UniRep [21]	LSTM	512	1900	18M	UniRef50 [14]	24M	Mar 2019
eUniRep [43]	LSTM	512	1900	18M	UniRef50 [14]; MSAs	24M	Jan 2020
ESM-1b [23]	Transformer	1024	1280	650M	UniRef50 [14]	250M	Dec 2020
ESM-1v [44]	Transformer	1024	1280	650M	UniRef90 [14]	98M	Jul 2021
ESM-IF1 [45]	Transformer	–	512	124M	UniRef50 [14]; CATH [46]	12M sequences; 16K structures	Sep 2022
ProGen [47]	Transformer	512	–	1.2B	UniParc [14]; UniprotKB [14]; Pfam [36]; NCBI taxonomy [48]	281M	Jul 2021
ProteinBERT [49]	Transformer	1024	–	16M	UniRef90 [14]	106M	May 2021
Tranception [15]	Transformer	1024	1280	700M	UniRef100 [14]	250M	May 2022
ESM-2 [50]	Transformer	1024	5120	15B	UniRef90 [14]	65M	Oct 2022

are deep learning frameworks cognizant of topological structures, along with their applications in protein engineering. For those readers who are interested in the deep mathematical details of TDA, we have added a supplementary section dedicated to two TDA methods - persistent homology and persistent spectral graph (PSG) in Supplementary Methods.

Homology

The basic idea behind TDA is to represent the data as a point cloud in a high-dimensional topological space, and then study the topological invariants of this space, such as the genus number, Betti number, and Euler characteristic. Among them, the Betti numbers, specifically Betti zero, Betti one, and Betti two, can be interpreted as representing connectedness, holes, and voids, respectively [76, 77]. These numbers can be computed as the ranks of the corresponding homology groups in appropriate dimensions.

Homology groups are algebraic structures that are associated with topological spaces [76]. They provide information about the topological connectivity of geometric objects. The basic idea behind homology is to consider the cycles and boundaries in a space. Loosely speaking, a cycle is a set of points in the space that form a closed loop, while a boundary is a set of points that form the boundary of some region in the space. The homology group of a space is defined as the group of cycles modulo the group of boundaries. That is, we identify two cycles that differ by a boundary and consider them to be equivalent. The resulting homology group encodes information about the Betti numbers of the space.

Homology theory has many applications in mathematics and science. It is used to classify topological spaces in category theory, to study the properties of manifolds in differential geometry and algebraic geometry, and to analyze data in various scientific fields [76]. However, the original homology groups offer truly geometry-free representations and are too abstract to carry sufficient geometric information of data. Persistent homology was designed to improve homology groups' ability for data analysis.

Persistent homology

Persistent homology is a relatively new tool in algebraic topology that is designed to incorporate multiscale topological analysis of data [62, 63]. The basic idea behind persistent homology is to construct a family of geometric shapes of the original data by filtration (Figure 2C). Filtration systematically enlarges the radius of each data point in a point cloud, leading to a family of topological spaces with distinct topological dimensions and connectivity. Homology groups are built from the family of shapes, giving rise to systematic changes in topological invariants, or Betti numbers, at various topological dimensions and geometric scales. Topological invariants based on Betti numbers are expressed in terms of persistence barcodes [78] (Figure 2D), persistence diagrams [79], persistence landscapes [80], or persistence images [81]. Persistent topological representations are widely used in applications, particularly in association with machine learning models [82].

Persistent homology is the most important approach in TDA (see Table 2 for a summary of major TDA approaches). It reveals the shape of data in terms of the topological invariants and has had tremendous success in scientific applications, including image and signal processing [83], machine learning [84], biology [82], and neuroscience[85]. Nonetheless, to effectively analyze complex biomolecular data, persistent homology requires further refinement and adjustment. [86].

Table 2. Summary of topological data analysis (TDA) methods for structures

Method	Topological space	Node attribute	Edge attribute
Homology-based			
Persistent homology [62, 63]	Simplicial complex	None	None
Element-specific PH (ESPH) [16]	Simplicial complex	Group labeled	Group labeled
Persistent cohomology [64]	Simplicial complex	Labeled	Labeled
Persistent path homology [65]	Path complex	Path	Directed
Persistent flag homology [66]	Flag complexes	None	Directed
Evolutionary homology [67]	Simplicial complex	Weighted	Weighted
Weighted persistent homology [68]	Simplicial complex	Weighted	Weighted
Laplacian-based			
Persistent spectral graph [69, 70]	Simplicial complex	None	None
Persistent Hodge Laplacians [71]	Manifold	Continuum	Continuum
Persistent sheaf Laplacians [72]	Cellular complex	Labeled	Sheaf relation
Persistent path Laplacians [73]	Path complex	Path	Direction
Persistent hypergraph [74]	Hypergraph	Hypernode	Hyperedge
Persistent directed hypergraphs [75]	Hypergraph	Hypernode	Directed hyperedge

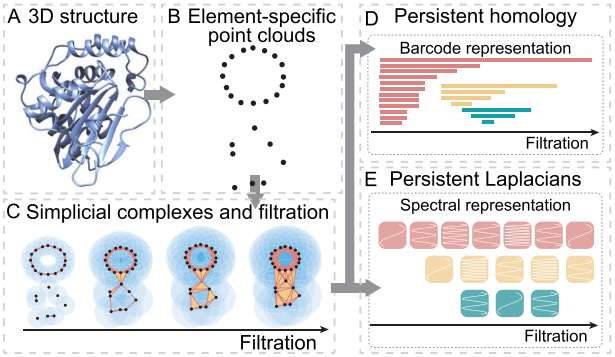

Figure 2. Conceptual illustration of the TDA-based protein modeling. (**A**). A three-dimensional protein structure. (**B**). Point cloud representation of protein structure. (**C**). Simplicial complexes and filtration provide multiscale topological representation of the point cloud. (**D**). Persistent homology characterizes topological evolution of the point cloud. (**E**). Persistent Laplacian characterizes shape evolution of the point cloud.

Persistent cohomology and element-specific persistent homology

One major limitation of persistent homology is that it fails to describe heterogeneous information of data point [64]. In other words, it treats all entries in the point cloud equally without considering other important information about the data. Biomolecules, for example, contain many different element types and each atom may have a different atomic partial charge, atomic interaction environment, and electrostatic potential function that cannot be captured by persistent homology. Thus, it is crucial to have a topological technique that can incorporate both geometric and nongeometric information into a unified framework.

Persistent cohomology was developed to provide such a mathematical paradigm [64]. In this framework, nongeometric information can either be prescribed globally or reside locally on atoms, bonds, or many-body interactions. In topological terminology, nongeometric information is defined on simplicial complexes. This persistent cohomology-based approach can capture multiscale geometric features and reveal non-geometric interaction patterns through topological invariants, or enriched persistence barcodes. It has been demonstrated that persistent cohomology outperforms other methods in benchmark protein-ligand binding affinity prediction datasets [64], which is a non-trivial problem in computational drug discovery.

An alternative approach for addressing the limitation of persistent homology is to use element-specific persistent homology (ESPH) [16]. The motivation behind ESPH is the same as that for persistent cohomology, but ESPH is relatively simple. Basically, atoms in the original biomolecule are grouped according to their element types, such as C, N, O, S, H, etc. Then, their combinations, such as CC, CN, CO, etc., are identified, and persistent homology analysis is applied to the atoms in each element combination, resulting in ESPH analysis. As a result, ESPH reduces geometric and biological complexities and embeds chemical and biological information into topological abstraction. The ESPH approach was used to win the D3R Grand Challenges, a worldwide competition series in computer-aided drug design [87].

Persistent topological Laplacians

However, aforementioned TDA methods are still limited in describing complex data, such as its lack of description of non-topological changes (i.e. homotopic shape evolution) [20], its incapability of coping with directed networks and digraphs (i.e. atomic partial charges and polarizations, gene regulation networks), and its inability to characterize structured data (e.g. functional groups, binding domains, and motifs) [86]. These limitations necessitate the development of innovative strategies.

Persistent topological Laplacians (PTLs) are a new class of mathematical tools designed to overcome the aforementioned challenges in TDA [86]. One of the first methods in this class is the PSG [69], also known as persistent combinatorial Laplacians [69] or persistent Laplacians [70]. PSGs have both harmonic spectra with zero eigenvalues and non-harmonic spectra with non-zero eigenvalues (Figure 2E). The harmonic spectra recover all the topological invariants from persistent homology, while the non-harmonic spectra capture the homotopic shape evolution of data that cannot be described by persistent homology [86]. PSGs have been used for accurate forecasting of emerging dominant SARS-CoV-2 variants BA.4/BA.5 [88], facilitating machine learning-assisted protein engineering predictions [20], and other applications [89].

Like persistent homology, persistent Laplacians are limited in their ability to handle directed networks and atomic polarizations.

To address these limitations, persistent path Laplacians have been developed [73]. Their harmonic spectra recover the topological invariants of persistent path homology [65], while their non-harmonic spectra capture homotopic shape evolution. Both persistent path Laplacians and persistent path homology were developed as a generalization of the path complex [90].

None of the PTLs mentioned above are capable of handling different types of elements in a molecule as persistent cohomology does. To overcome this limitation, persistent sheaf Laplacians [72] were designed, inspired by persistent cohomology [64], persistent Laplacians [69], and sheaf Laplacians for cellular sheaves [91]. The aim of persistent sheaf Laplacians is to discriminate between different objects in a point cloud. By associating a set of non-trivial labels with each point in a point cloud, a persistent module of sheaf cochain complexes is created, and the spectra of persistent sheaf Laplacians encode both geometrical and non-geometrical information [72]. The theory of persistent sheaf Laplacians is an elegant method for the fusion of different types of data and opens the door to future developments in TDA, geometric data analysis, and algebraic data analysis.

Persistent hypergraph Laplacians enable the topological description of internal structures or organizations in data [74]. Persistent hyperdigraph Laplacians further allow for the topological Laplacian modeling of directed hypergraphs [75]. These persistent topological Laplacians can be utilized to describe intermolecular and intramolecular interactions. As protein structures are inherently multiscale, it is natural to apply persistent hypergraph Laplacians and persistent hyperdigraph Laplacians to delineate the protein structure-function relationship.

Finally, unlike all the aforementioned PTLs, evolutionary de Rham-Hodge Laplacians or persistent Hodge Laplacians are defined on a family of filtration-induced differentiable manifolds [71]. They are particularly valuable for the multiscale topological analysis of volumetric data. Technically, a similar algebraic topology structure is shared by persistent Hodge Laplacians and persistent Laplacians, but the former is a continuum theory for volumetric data and the latter is a discrete formulation for point cloud. As such, their underlying mathematical definitions, i.e. differential forms on manifolds and simplicial complexes on graphs, are sharply different.

Deep graph neural networks and topological deep learning

Similar to topological data analysis, graph- and topology-based deep learning models have been proposed to capture connectivity and shape information of protein structure data. Graph neural networks (GNNs) consider the low-order interactions between vertices by aggregating information from neighbor vertices. A variety of popular graph neural network layers has been proposed, such as convolution graph networks (GCN) [92], graph attention networks (GAT) [93], graph sample and aggregate (GraphSAGE) [94], Graph Isomorphism Network (GIN) [95], and gated graph neural network [96].

With variety of architectures of GNN layers, self-supervised learning models are widely used for representation learning of graph-based data. Graph autoencoder (GAE) and variational graph autoencoder (VGAE) consist of both encoder and decoder, where the decoder employ a linear inner product to reconstruct adjacent matrix [97]. While most of graph-based self-supervised models only have encoder. Deep graph infomax (DGI) maximizes mutual information between a graph's local and global features to achieve self-supervised learning [98]. Graph contrastive learning (GRACE) constructs positive and negative pairs from a single graph, and trains a GNN to differentiate between them [99]. Self-supervised graph transformer (SSGT) uses masked node prediction to train the model. Given a masked graph, it tries to predict the masked node's attributes from the unmasked nodes [100].

In applications to learning protein structures, GCNs have been widely applied to building structure-to-function map of proteins [101, 102]. Moreover, self-supervised models provide powerful pre-trained model in learning representation of protein structures. GeoPPI [103] proposed a graph neural network-based autoencoder to extract structural embedding at the protein-protein binding interface. The subsequent downstream models allow accurate predictions for protein-protein binding affinity upon mutations [103] and further design effective antibody against SARS-CoV-2 variants [104]. GRACE has been applied to learn geometric representation of protein structures [105]. To adopt the critical biophysical properties and interactions between residues and atoms in protein structures, graph-based self-supervised learning models have been customized to achieve the specific functions. The inverse protein folding protocol was proposed to capture the complex structural dependencies between residues in its representation learning [45, 106]. OAGNNs was proposed to better sense the geometric characteristics such as nner-residue torsion angles, inter-residue orientations in its representation learning [107].

Topological deep learning, proposed by Cang and Wei in 2017 [108], is an emerging paradigm. It integrates topological representations with deep neural networks for protein fitness learning and prediction [20, 87, 108]. Similar graph and topology-based deep learning architectures have also been proposed to capture connectivity and shape information of protein structure data [75, 88]. Inspired by TDA, high-order interactions among neural nodes were proposed in k-GNNs [109] and simplicial neural networks [110].

ARTIFICIAL INTELLIGENCE-AIDED PROTEIN ENGINEERING

Protein engineering is a typical black-box optimization problem, which focuses on finding the optimal solution without explicitly knowing the objective function and its gradient. In protein engineering, the goal in designing algorithms for this problem is to efficiently search for the best sequence within a large search space:

$$x^* = \arg\max_{x \in \mathcal{S}} f(x), \tag{1}$$

where $\mathcal{S}$ is an unlabeled candidate sequence library, x is a sequence in the library and $f(x)$ is the unknown sequence-to-fitness map for optimization. The fitness landscape, $f(\mathcal{S})$, is a high-dimensional surface that maps amino acid sequences to properties such as activity, selectivity, stability, and other physicochemical features.

There are two practical challenges in protein engineering. First, the fitness landscape is usually epistatic [111, 112], where the contribution of individual amino acid residues to protein fitness have dependency to each other. The interdependence leads to complex, non-linear interactions among different residues. In other word, the fitness landscape contains large number of local optima. For example, in a four-site mutational fitness landscape for GB1 protein with $20^4 = 160\,000$ mutations, 30 local maximum fitness peaks were found [111]. Either traditional directed evolution experiments such as single-mutation walk and recombination, or machine learning models, is difficult to find the global optima without trapped at local one. Second, protein engineering

process usually collects limited number of data comparing to the huge sequence library. There are an enormous number of ways to mutate any given protein: for a 300-amino-acid protein, there are 5700 possible single-amino-acid substitutions and 32 381 700 ways to make just two substitutions with the 20 canonical amino acids [12]. Even with high-throughput experiments, only a small fraction of the sequence library can be screened. Despite this, many systems only have low-throughput assays such as membrane proteins [113], making the process more difficult.

With enriched data-driven protein modeling approaches from protein sequences to structures, recent advanced machine learning methods have been widely developed to accelerate protein engineering in silico (Figure 1A) [1, 11, 12, 114, 115]. Utilizing a limited experimental capacity, machine learning models can effectively augment the fitness evaluation process, enabling the exploration of a vast search space $\mathcal{S}$. This approach facilitates the discovery of optimal solutions within complex design spaces, despite constraints on the number of trials or experiments.

Using a limited number of experimentally labeled sequences, machine learning models can carry out zero-shot or few-shot predictions [11]. The accuracy of these predictions largely depends on the distribution of the training data, which influences the model's ability to generalize to new sequences. Concretely, if the training data is representative or closer to a given sequence, the model is more likely to make accurate predictions for that specific sequence. Conversely, if the training data is not representative or distant from the given sequence, the model's predictive accuracy may be compromised, leading to less reliable results. Therefore, MLPE are usually an iterative process between machine learning models and experimental screens. Incorporating the exploration-exploitation trade-off in this context is essential for achieving optimal results. During the iterative process, the model must balance exploration, where it seeks uncertain regions that machine learning models have low accuracy, with exploitation, where it refines and maximizes fitness based on previously gained knowledge. A right balance is critical to preventing overemphasis on either exploration or exploitation leading, which may lead to suboptimal solutions. In particular, the epistatic nature of protein fitness landscapes influences the exploration-exploitation trade-off in the design process.

MLPE methods need to take the experimental capacity into account when attempt to balance the exploitation-exploration. In this section, we discuss different strategies upon the number of experimental capacity. First, we discuss zero-shot strategy when no labeled experimental data is available. Second, we discuss supervised models for performing greedy search (i.e. exploitation). Last, we discuss uncertainty quantification models that balance exploration and exploitation trade-off.

Unsupervised zero-shot strategy

First, we review the zero-shot strategy that interrogates protein fitness with an unsupervised manner (Figure 1B and Table 3). This is designed for the scenarios in the early stage designs where no experiments have been conducted or the experimentally labeled data is too limited allowing accurate fitness predictions from supervised models [11, 20]. They delineate a fitness landscape at the early stage of protein engineering. Essential residues can be identified and prioritized for mutational experiments, allowing for a more targeted approach to protein engineering [22]. Additionally, the initial fitness landscape can be utilized to filter out protein candidates with a low likelihood of exhibiting the desired functionality. By focusing on sequences with higher probabilities, protein engineering process can be made more efficient and effective [34].

Table 3. Comparisons for fitness predictors. Results were adopted from TopFit [20]. Performance was reported by average Spearman correlation over 34 DMS datasets and 20 repeats. Supervised model use ensemble regression from 18 regression models [20]

Zero-shot predictors				
Model name	training set size			
	0			
ESM-1b PLL [23, 33]	0.435			
eUniRep PLL [127]	0.411			
EVE [40]	0.497			
Tranception [15]	0.478			
DeepSequence [22]	**0.504**			
Supervised models				
Embedding name	training set size			
	24	96	168	240
Persistent homology [20]	0.263	0.432	0.496	0.534
Persistent Laplacian [20]	**0.280**	**0.457**	**0.525**	**0.564**
ESM-1b [23]	0.219	0.421	0.494	0.537
eUniRep [43]	0.259	0.432	0.485	0.515
Georgiev [127]	0.169	0.326	0.402	0.446
UniRep [21]	0.183	0.347	0.420	0.462
Onehot	0.132	0.317	0.400	0.450
Bepler [42]	0.139	0.287	0.353	0.396
TAPE LSTM [41]	0.259	0.436	0.492	0.522
TAPE ResNet [41]	0.080	0.216	0.305	0.358
TAPE transformer [41]	0.146	0.304	0.371	0.418

Zero-shot predictions rely on the model's ability to recognize patterns in naturally observed proteins, enabling it to make informed predictions for new sequences without having direct training data for the target protein. As discussed in Section 2, protein language models, particularly generative models, learn the distribution of naturally observed proteins which are usually functional. The learned distribution can be used to assess the likelihood that a newly designed protein lies within the distribution of naturally occurring proteins, thus providing valuable insights into its potential functionality and stability [11].

VAEs are popular local evolutionary models for zero-shot predictions such as DeepSequence [22] and EVE models [40]. In VAEs, the conditional probability distribution $p(x \mid z, \theta)$ is the decoder in a form of neural network with parameters θ, where x is the sequence being query and z is its latent space variable. Similar, encoder, $q(z \mid x, \phi)$, is modeled by another neural network with parameters ϕ to approximate the true posterior distribution $p(z \mid x)$. For a given sequence x, its probabilistic likelihood in VAEs is $p(x \mid \theta)$ parameterized by parameters θ. Direct computation of this probability, $p(x \mid \theta) = \int p(x \mid z, \theta)\, dz$, is intractable in the general case. The evidence lower bound (ELBO) forming a variation inference [54] provides a lower bound of the log likelihood:

$$\log p(x \mid \theta) \geq \mathrm{ELBO}(x) = \mathbb{E}_q \log p(x \mid z, \theta) - \mathrm{KL}(q(z \mid x, \phi) \mid p(z)). \quad (2)$$

ELBO is taken as the scoring function to quantify the mutational likelihood of each query sequence. The ELBO-based zero-shot predictions show advanced performance reported in multiple works [20, 33, 55].

Transformer is the currently state-of-the-art model which has been used in many supervised tasks [23]. It learns a global distribution of nature proteins. It has also been proved to have advanced performance for zero-shot predictions [33, 44]. The training of Transformer uses mask filling that refers to the process of predicting masked amino acid in a given input sequence by leveraging the contextual information encoded in the Transformer's self-attention mechanism [59, 60]. The mask filling procedure creates a classification layer on the top of the Transformer architecture. Given a sequence x, the masked filling classifier generate probability distributions for amino acids at masked positions. Suppose x has L amino acids $x = x_1x_2\cdots x_L$, by masking a single amino acid at ith position, the classifier calculates the conditional probability of $p(x_i \mid x^{(-i)})$, where $x^{(-i)}$ is the remaining sequence excluding the masked ith position. To reduce the computational cost, the pseudo-log-likelihoods (PLLs) are usually used to estimate the log-likelihood of a given sequence [33, 34]:

$$\mathrm{PLL}(s) = \sum_{i=1}^{L} \log P(s_i \mid s^{(-i)}). \tag{3}$$

The PPLs assume the independence between amino acids. To consider the dependence between amino acids, one can calculate the conditional probability by summing up all possible factorization [34]. But this approach leads to much higher computational cost.

Furthermore, many different strategies have been employed to make zero-shot predictions. Fine-tune model can improve the predictions by combining both local and global evolutionary models [43]. Tranception scores combine global autoregressive inference and an local MSAs retrieval inference to make more accurate predictions. In addition to these sequence-based models, the structure-based GNN-based models including ESM-1F [45] and RGC [116] have also been proposed by utilizing large-scale structural data from AlphaFold2. However, the structure-based model is still limited in accuracy comparing to sequence-based models.

Supervised regression models

Supervised regression models are among the most prevalent approaches used in guiding protein engineering, as they enable greedy search strategies to maximize protein fitness (Figure 1C). These models, including statistical, machine learning, and deep learning techniques, rely on a set of labeled data as their training set to predict the fitness landscape. By leveraging the information contained within the training data, supervised regression models can effectively estimate the relationship between protein sequences and their fitness, providing valuable insights for protein engineering and optimization [1, 12].

A variety of supervised models have been applied to predict protein fitness. In general, statistical models and machine learning models such as linear regression [117], ridge regression [33], support vector machine (SVM) [118], random forest [119], gradient boosting tree [120] have accurate performance for small training set. And deep learning methods such as deep neural networks [121], convolutional neural networks (CNNs) [17], attention-based neural networks [122] are more accurate with large size of training data. However, in protein engineering, the size of training data increases sequentially which make the supervised models difficult to provide accurate performance all time. Alternatively, the ensemble regression was proposed to provide robust fitness predictions despite of training data size [11, 123]. The ensemble regression average predictions from multiple supervised models and they provide more accurate and robust performance than single model [20]. To remove the inaccurate models in the average, cross-validation is usually used to rank accuracy of each model and only top models are taken to average the predictions. Paired with the zero-shot strategy, the ensemble regression trained on informed training set pre-selected by zero-shot predictions can efficiently pick up the global optimal protein with a few round of experiments [34, 124, 125]. And such approach has been applied to enable resource-efficient engineering CRISPR-Cas9 genome editor activities [126].

Rather than the architectures of supervised models, the predictive accuracy highly rely on the amount of information obtained from the featurization process (Table 3). The physical-chemical properties extract the properties of individual amino acids or atoms [127]. The energy-based scores provide descriptions for the overall property of the target protein [18]. However, neither of them successfully take the complex interactions between residues and atoms into account. To tackle this challenge, recent mathematics-initiated topological and geometric descriptors achieved great success in predicting protein fitness including protein-protein interactions [17], protein stability [120], enzyme activity, and antibody effectivity [20]. The aforementioned descriptors (Section 3.1) extract structural information from atoms at different characteristic lengths. Furthermore, the sequence-based protein language models provide another featurization strategies. The deep pre-trained models have the latent space which provide the informative representation of each given sequence. Building supervised models from the deep embedding exhibits accurate performance [20, 128]. Recent works combine different types of sequence-based features [33, 129] or combine structure-based and sequence-based features [20] show the complementary roles of different featurization approaches.

Active learning models for exploration-exploitation balance

With the extensive accurate protein-to-fitness machine learning models, active learning further designs iterative strategy between models and experiments to sequentially optimize fitness with the consideration of exploitation-exploration trade-off (Figure 1D–E) [115].

To balance the exploitation-exploration trade-off, the supervised models require to predict not only the protein fitness but also quantify the uncertainty of the given protein [130]. The most popular uncertainty quantification in protein engineering is Gaussian process (GP) [131], which automatically calibrate the balance. Especially, GP using the upper confidence bounds (UCBs) acquisition has efficient convergent rate theoretically for solving the black-box optimization (Equation 1). A variety protein engineering employed GP to accelerate the fitness optimization. For examples, the light-gated channelrhodopsins (ChRs) were engineered to improve photocurrence and light sensitivity [132, 133], green fluorescent protein has been engineered to become yellow fluorescence [134], acyl-ACP reductase was engineered to improve fatty alcohol production [135], P450 enzyme has been engineered to improve thermostability [136].

The tree-based search strategy is also efficient by building a hierarchical search path, such as the hierarchical optimistic optimization (HOO) [137], the deterministic optimistic optimization (DOO), and the simultaneous optimistic optimization (SOO) [138]. To handle the discrete mutational space in protein engineering, an unsupervised clustering approach was employed to construct the hierarchical tree structure [124, 125].

Recently, researchers have turned to generative models to quantify uncertainty in protein engineering, employing methods

such as Variational Autoencoders (VAEs) [22, 40, 54], generative adversarial networks (GANs) [139, 140], and autoregressive language models [15, 141]. Generative models are a class of machine learning algorithms that aim to learn the underlying data distribution of a given dataset, in order to generate new, previously unseen data points that resemble the training data. These models capture the inherent structure and patterns present in the data, enabling them to create realistic and diverse samples that share the same characteristics as the original data. For examples, ProGen [47] is a large language model that generate functional protein sequences across diverse families. A Transformer-based antibody language models utilize fine-tuning processes to assist design antibody [142]. Recently, a novel Transformer-based model called ReLSO has been introduced [143]. This innovative approach simultaneously generates protein sequences and predicts their fitness using its latent space representation. The attention-based relationships learned by the jointly trained ReLSO model offer valuable insights into sequence-level fitness attribution, opening up new avenues for optimizing proteins.

CONCLUSIONS AND FUTURE DIRECTIONS

In this review, we have discussed the advanced deep protein language models for protein modeling. We further provided an introduction of topological data analysis methods and their applications in protein modeling. Relying on both structure-based and sequence-based models, MLPE methods were widely developed to accelerate protein engineering. In the future, various machine learning and deep learning will have potential perspectives in protein engineering.

Accurate structure prediction methods enhanced accurate structure-based models

Comparing to sequence data, three-dimensional protein structural data offer more comprehensive and explicit descriptions of the biophysical properties of a protein and its fitness. As a result, structure-based models usually provide superb performance than sequence-based models for supervised tasks with small training set [20, 120].

As protein sequence databases continue to grow, self-supervised models demonstrate their ability to effectively model proteins using large-scale data. The protein sequence database provides a vast amount of resources for building sequence-based models, such as UniProt [14] database contains hundreds of millions sequences. In contrast, protein structure databases are comparatively limited in size. The largest among them, Protein Data Bank (PDB), contains only 205 thousands of protein structures as of 2023 [13]. Due to the abundance of data resources, sequence-based models typically outperform structure-based models significantly [116].

To address the limited availability of structure data, researchers have focused on developing highly accurate deep learning techniques aimed at enabling large-scale structure predictions. These state-of-the-art methodologies have the potential to significantly expand the database of known protein structures. Two prominent methods are AlphaFold2 [24] and RosettaFold [144], which have demonstrated remarkable capabilities in predicting protein structures with atomic-level accuracy. By harnessing the power of cutting-edge deep learning algorithms, these tools have successfully facilitated the accurate prediction of protein structures, thus contributing to the expansion of the structural database.

Both AlphaFold2 and RosettaFold are alignment-based, which rely on MSAs of the target protein for structure prediction. Alignment-based approaches can be highly accurate when there are sufficient number of homologous sequences (that is, MSAs depth) in the database. Therefore, these methods may have reduced accuracy with low MSAs depth in database. In addition, the MSAs search is time consuming which slows down the prediction speed. Alternatively, alignment-free methods have also been proposed to tackle these limitations [145]. An early work RGN2 [146] exhibits more accurate predictions than AlphaFold2 on orphans proteins which lack of MSAs. Supervised transformer protein language models predict orphan protein structures [147]. With the development of variety of large-scale protein language models in recent years, the alignment-free structural prediction methods incorporate with these models to exhibit their accuracy and efficiency. For example, ESMFold [50] and OmegaFold [148] achieve similar accuracy with AlphaFold2 with faster speed. Moreover, extensive language model-based methods were developed for structural predictions of single-sequence and orphan proteins [149–152]. Large-scale protein language models will provide powerful toolkits for protein structural predictions.

In building protein fitness model, the structural TDA-based model has exemplified that the AlphaFold2 structure is as reliable as the experimental structure [20]. The zero-shot model, ESM-IF1, also shows advanced performance with coupling with the large structure AlphaFold database [45]. In the light of the revolutionary structure predictive models, structure-based models will open up a new avenue in protein engineering, from directed evolution to de novo design [153, 154]. More sophisticated TDA methods will be demanded to handle the large-scale datasets. Large-scale deep graph neural networks will need to be further developed, for example, to consider the high-order interactions using simplicial neural networks [110, 155].

Large highthroughput datasets enabled larger scale models

Current MLPE methods are usually designed for limited training set. The ensemble regression is an effective approach to accurately learn the fitness landscape with small but increasing size of training sets from deep mutational scanning [34].

The breakthrough biotechnology, next-generation sequencing (NGS) [156] largely enhances the capacity of DMS for collecting supervised fitness data in various protein systems [111, 112, 157]. The resulting large-scale deep mutational scanning databases expand the exploration range of protein engineering. Deeper machine learning models are emerging to enhance the accuracy and adaptivity for protein engineering.

Key Points

- Machine learning and deep learning techniques are revolutionizing protein engineering.
- Topological data analysis enables advanced structure-based machine learning-assisted protein engineering approaches.
- Deep protein language models extract critical evolutionary information from large-scale sequence databases.

SUPPLEMENTARY DATA

Supplementary data are available online at http://bib.oxfordjournals.org/.

FUNDING

This work was supported in part by NIH grants (R01GM126189, R35GM148196, and R01AI164266), NSF grants (DMS-2052983, DMS-1761320, and IIS-1900473), NASA grant (80NSSC21M0023), Michigan Economic Development Corporation, MSU Foundation, Bristol-Myers Squibb (65109), and Pfizer.

AUTHORS' CONTRIBUTION

Y.Q. and G.W.W. conceived, wrote, and revised the manuscript.

DATA AVAILABILITY

No new data were generated or analysed in support of this research.

REFERENCES

1. Narayanan H, Dingfelder F, Butté A, *et al.* Machine learning for biologics: opportunities for protein engineering, developability, and formulation. *Trends Pharmacol Sci* 2021;**42**(3): 151–65.
2. Arnold FH. Design by directed evolution. *Acc Chem Res* 1998;**31**(3):125–31.
3. Karplus M, Kuriyan J. Molecular dynamics and protein function. *Proc Natl Acad Sci* 2005;**102**(19):6679–85.
4. Boyken SE, Chen Z, Groves B, *et al.* De novo design of protein homo-oligomers with modular hydrogen-bond network–mediated specificity. *Science* 2016;**352**(6286):680–7.
5. Romero PA, Arnold FH. Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Biol* 2009;**10**(12): 866–76.
6. Bhardwaj G, Mulligan VK, Bahl CD, *et al.* Accurate de novo design of hyperstable constrained peptides. *Nature* 2016;**538**(7625):329–35.
7. Pierce NA, Winfree E. Protein design is NP-hard. *Protein Eng* 2002;**15**(10):779–82.
8. Siedhoff NE, Schwaneberg U, Davari MD. Machine learning-assisted enzyme engineering. *Meth Enzymol* 2020;**643**:281–315.
9. Mazurenko S, Prokop Z, Damborsky J. Machine learning in enzyme engineering. *ACS Catal* 2019;**10**(2):1210–23.
10. Diaz DJ, Kulikova AV, Ellington AD, Wilke CO. Using machine learning to predict the effects and consequences of mutations in proteins. *Curr Opin Struct Biol* 2023;**78**:102518.
11. Wittmann BJ, Johnston KE, Zachary W, Arnold FH. Advances in machine learning for directed evolution. *Curr Opin Struct Biol* 2021;**69**:11–8.
12. Yang KK, Zachary W, Arnold FH. Machine-learning-guided directed evolution for protein engineering. *Nat Methods* 2019;**16**(8):687–94.
13. Berman HM, Westbrook J, Feng Z, *et al.* The protein data bank. *Nucleic Acids Res* 2000;**28**(1):235–42.
14. The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Res* 2021;**49**(D1):D480–9.
15. Notin P, Dias M, Frazer J, *et al.*. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In: Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, Sivan Sabato (eds). *International Conference on Machine Learning*, vol. **162**. Baltimore, Maryland, USA: PMLR, 2022, 16990–7017.
16. Cang Z, Wei G-W. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *Int J Numer Methods Biomed* 2018;**34**(2): e2914.
17. Wang M, Cang Z, Wei G-W. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nat Mach* 2020;**2**(2):116–23.
18. Schymkowitz J, Borg J, Stricher F, *et al.* The FoldX web server: an online force field. *Nucleic Acids Res* 2005;**33**(suppl_2):W382–8.
19. Leman JK, Weitzner BD, Lewis SM, *et al.* Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nat Methods* 2020;**17**(7):665–80.
20. Qiu Y, Wei G-W. Persistent spectral theory-guided protein engineering. *Nat Comput Sci* 2023;**3**(2):149–63.
21. Alley EC, Khimulya G, Biswas S, *et al.* Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods* 2019;**16**(12):1315–22.
22. Riesselman AJ, Ingraham JB, Marks DS. Deep generative models of genetic variation capture the effects of mutations. *Nat Methods* 2018;**15**(10):816–22.
23. 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.
24. Jumper J, Evans R, Pritzel A, *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* 2021;**596**(7873): 583–9.
25. Edelsbrunner H, Harer J. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
26. Zomorodian A, Carlsson G. Computing persistent homology. *Discrete Comput Geom* 2005;**33**(2):249–74.
27. Nguyen DD, Wei G-W. DG-GL: differential geometry-based geometric learning of molecular datasets. *Int J Numer Methods Biomed Eng* 2019;**35**(3):e3179.
28. Wee JJ, Xia K. Ollivier persistent Ricci curvature-based machine learning for the protein–ligand binding affinity prediction. *J Chem Inf Model* 2021;**61**(4):1617–26.
29. Nguyen DD, Wei G-W. AGL-Score: algebraic graph learning score for protein–ligand binding scoring, ranking, docking, and screening. *J Chem Inf Model* 2019;**59**(7):3291–304.
30. Ryczko K, Strubbe DA, Tamblyn I. Deep learning and density-functional theory. *Phys Rev A* 2019;**100**(2):022512.
31. Butler KT, Davies DW, Cartwright H, *et al.* Machine learning for molecular and materials science. *Nature* 2018;**559**(7715): 547–55.
32. Chen J, Geng W, Wei G-W. MLIMC: machine learning-based implicit-solvent Monte Carlo. *Chin J Chem Phys* 2021;**34**(6): 683–94.
33. Hsu C, Nisonoff H, Fannjiang C, Listgarten J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat Biotechnol* 2022;**40**(7):1114–22.
34. Wittmann BJ, Yue Y, Arnold FH. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst* 2021;**12**(11):1026–45.
35. Khurana D, Koli A, Khatter K, Singh S. Natural language processing: state of the art, current trends and challenges. *Multimed Tools Appl* 2023;**82**(3):3713–44.
36. El-Gebali S, Mistry J, Bateman A, *et al.* The Pfam protein families database in 2019. *Nucleic Acids Res* 2019;**47**(D1):D427–32.
37. Shihab HA, Gough J, Cooper DN, *et al.* Predicting the functional, molecular, and phenotypic consequences of amino acid

substitutions using hidden Markov models. *Hum Mutat* 2013;**34**(1):57–65.
38. Hopf TA, Ingraham JB, Poelwijk FJ, *et al.* Mutation effects predicted from sequence co-variation. *Nat Biotechnol* 2017;**35**(2):128–35.
39. Rao RM, Liu J, Verkuil R, *et al.* MSA transformer. In: Marina Meila, Tong Zhang (eds). *International Conference on Machine Learning*, vol. **139**. PMLR, 2021, 8844–56.
40. Frazer J, Notin P, Dias M, *et al.* Disease variant prediction with deep generative models of evolutionary data. *Nature* 2021;**599**(7883):91–5.
41. Rao R, Bhattacharya N, Thomas N, *et al.* Evaluating protein transfer learning with TAPE. *Adv Neural Inf Process* 2019;**32**:9689.
42. Bepler T and Berger B. Learning protein sequence embeddings using information from structure. In: *International Conference on Learning Representations*. New Orleans, Louisiana, United States, 2018.
43. Biswas S, Khimulya G, Alley EC, *et al.* Low-N protein engineering with data-efficient deep learning. *Nat Methods* 2021;**18**(4):389–96.
44. Meier J, Rao R, Verkuil R, *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *Adv Neural Inf Process Syst* 2021;**34**. https://openreview.net/forum?id=uXc42E9ZPFs.
45. Hsu C, Verkuil R, Liu J, *et al.* Learning inverse folding from millions of predicted structures. In: Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, Sivan Sabato (eds). *International Conference on Machine Learning*, vol. **162**. Baltimore, Maryland, USA: PMLR, 2022, 8946–70.
46. Orengo CA, Michie AD, Jones S, *et al.* Cath–a hierarchic classification of protein domain structures. *Structure* 1997;**5**(8):1093–109.
47. Madani A, Krause B, Greene ER, *et al.* Large language models generate functional protein sequences across diverse families. *Nat Biotechnol* 2023;1–8. https://www.nature.com/articles/s41587-022-01618-2#citeas.
48. Federhen S. The NCBI Taxonomy database. *Nucleic Acids Res* 2012;**40**(D1):D136–43.
49. Brandes N, Ofer D, Peleg Y, *et al.* ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics* 2022;**38**(8):2102–10.
50. Lin Z, Akin H, Rao R, *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* 2023;**379**(6637):1123–30.
51. Eddy SR. Accelerated profile HMM searches. *PLoS Comput Biol* 2011;**7**(10):e1002195.
52. Hopf TA, Green AG, Schubert B, *et al.* The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics* 2019;**35**(9):1582–4.
53. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 1989;**77**(2):257–86.
54. Kingma DP, Welling M. Auto-encoding variational bayes. Preprint, arXiv:1312.6114, 2013.
55. Livesey BJ, Marsh JA. Using deep mutational scanning to benchmark variant effect predictors and identify disease mutations. *Mol Syst Biol* 2020;**16**(7):e9380.
56. Kim Y. Convolutional neural networks for sentence classification. In: Alessandro Moschitti, Bo Pang, Walter Daelemans (eds). *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, 1746–51.
57. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770–8.
58. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**(8):1735–80.
59. Vaswani A, Shazeer N, Parmar Niki, *et al.* Attention is all you need. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds). *Advances in Neural Information Processing Systems*, vol **30**. 2017, 5998–6008.
60. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint, arXiv:1810.04805, 2018.
61. Detlefsen NS, Hauberg S, Boomsma W. Learning meaningful representations of protein sequences. *Nat Commun* 2022;**13**(1):1914.
62. Edelsbrunner H, Harer J, *et al.* Persistent homology-a survey. *Contemp Math* 2008;**453**(26):257–82.
63. Zomorodian A, Carlsson G. Computing persistent homology. In: Jack Snoeyink, Jean-Daniel Boissonnat (eds). *Proceedings of the Twentieth Annual Symposium on Computational Geometry*. New York, NY, United States: Association for Computing Machinery, 2004, 347–56.
64. Cang Z, Wei G-W. Persistent cohomology for data with multicomponent heterogeneous information. *SIAM J Math Data Sci* 2020;**2**(2):396–418.
65. Chowdhury S, Mémoli F. Persistent path homology of directed networks. In: Artur Czumaj (ed) *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2018, 1152–69.
66. Lütgehetmann D, Govc D, Smith JP, Levi R. Computing persistent homology of directed flag complexes. *Algorithms* 2020;**13**(1):19.
67. Cang Z, Munch E, Wei G-W. Evolutionary homology on coupled dynamical systems with applications to protein flexibility analysis. *J Appl Comput Topol* 2020;**4**:481–507.
68. Meng Z, Vijay Anand D, Lu Y, *et al.* Weighted persistent homology for biomolecular data analysis. *Sci Rep* 2020;**10**(1):2079.
69. Wang R, Nguyen DD, Wei G-W. Persistent spectral graph. *Int J Numer Methods Biomed Eng* 2020;**36**(9):e3376.
70. Mémoli F, Wan Z, Wang Y. Persistent Laplacians: properties, algorithms and implications. *SIAM J Math Data Sci* 2022;**4**(2):858–84.
71. Chen J, Zhao R, Tong Y, Wei G-W. Evolutionary de Rham-Hodge method. *Discrete Continuous Dyn Syst Ser B* 2021;**26**(7):3785.
72. Wei X, Wei G-W. Persistent sheaf Laplacians. Preprint, arXiv:2112.10906, 2021.
73. Wang R, Wei G-W. Persistent path Laplacian. *Found Data Sci* 2023;**5**:26–55.
74. Liu X, Feng H, Jie W, Xia K. Persistent spectral hypergraph based machine learning (PSH-ML) for protein-ligand binding affinity prediction. *Brief Bioinform* 2021;**22**(5):bbab127.
75. Chen D, Liu J, Wu J, Wei G-W. Persistent hyperdigraph homology and persistent hyperdigraph Laplacians. Preprint, arXiv:2304.00345, 2023.
76. Kaczynski T, Mischaikow KM, and Mrozek M.. *Computational Homology*, vol. **3**, Springer, 2004.
77. Wasserman L. Topological data analysis. *Annu Rev Stat* 2018;**5**:501–32.
78. Ghrist R. Barcodes: the persistent topology of data. *Bull New Ser Am Math Soc* 2008;**45**(1):61–75.
79. Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of persistence diagrams. In: Joe Mitchell, Günter Rote (eds). *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*.

New York, NY, United States: Computing Machinery, 2005, 263–71.
80. Bubenik P, *et al.* Statistical topological data analysis using persistence landscapes. *J Mach Learn Res* 2015;**16**(1):77–102.
81. Adams H, Emerson T, Kirby M, *et al.* Persistence images: a stable vector representation of persistent homology. *J Mach Learn Res* 2017;**18**(8):1–35.
82. Cang Z, Mu L, Wu K, *et al.* A topological approach for protein classification. *Comput Math Biophys* 2015;**3**(1). https://www.degruyter.com/document/doi/10.1515/mlbmb-2015-0009/html.
83. Clough JR, Byrne N, Oksuz I, *et al.* A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Trans Pattern Anal Mach Intell* 2020;**44**(12):8766–78.
84. Pun CS, Xia K, Lee SX. Persistent-homology-based machine learning and its applications–a survey. Preprint, arXiv:1811.00252, 2018.
85. Stolz BJ, Harrington HA, Porter MA. Persistent homology of time-dependent functional networks constructed from coupled time series. *Chaos* 2017;**27**(4):047410.
86. Wei G-W. Topological data analysis hearing the shapes of drums and bells. Preprint, arXiv:2301.05025, 2023.
87. Nguyen DD, Cang Z, Kedi W, *et al.* Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges. *J Comput Aided Mol Des* 2019;**33**:71–82.
88. Chen J, Qiu Y, Wang R, Wei G-W. Persistent Laplacian projected Omicron BA.4 and BA.5 to become new dominating variants. *Comput Biol Med* 2022;**151**:106262.
89. Meng Z, Xia K. Persistent spectral–based machine learning (PerSpect ML) for protein-ligand binding affinity prediction. *Sci Adv* 2021;**7**(19):eabc5329.
90. Grigor'yan AA, Lin Y, Muranov YV, Yau S-T. Path complexes and their homologies. *J Math Sci* 2020;**248**:564–99.
91. Hansen J, Ghrist R. Toward a spectral theory of cellular sheaves. *J Appl Comput Topol* 2019;**3**:315–58.
92. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. Preprint, arXiv:1609.02907, 2016.
93. Veličković P, Cucurull G, Casanova A, *et al.* Graph attention networks. Preprint, arXiv:1710.10903, 2017.
94. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Adv Neural Inf Process Syst* 2017;**30**.
95. Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? Preprint, arXiv:1810.00826, 2018.
96. Li Y, Tarlow D, Brockschmidt M, Zemel R. Gated graph sequence neural networks. Preprint, arXiv:1511.05493, 2015.
97. Kipf TN, Welling M. Variational graph auto-encoders. Preprint, arXiv:1611.07308, 2016.
98. Veličković P, Fedus W, Hamilton WL, *et al.* Deep graph infomax. Preprint, arXiv:1809.10341, 2018.
99. You Y, Chen T, Sui Y, *et al.* Graph contrastive learning with augmentations. *Adv Neural Inf Process Syst* 2020;**33**:5812–23.
100. Rong Y, Bian Y, Tingyang X, *et al.* Self-supervised graph transformer on large-scale molecular data. *Adv Neural Inf Process Syst* 2020;**33**:12559–71.
101. Li S, Zhou J, Xu T, *et al.* Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In: Feida Zhu, Beng Chin Ooi (eds). *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. New York, NY, United States: Computing Machinery, 2021, 975–85.
102. Gligorijević V, Renfrew PD, Kosciolek T, *et al.* Structure-based protein function prediction using graph convolutional networks. *Nat Commun* 2021;**12**(1):3168.
103. Liu X, Luo Y, Li P, *et al.* Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. *PLoS Comput Biol* 2021;**17**(8):e1009284.
104. Shan S, Luo S, Yang Z, *et al.* Deep learning guided optimization of human antibody against SARS-CoV-2 variants with broad neutralization. *Proc Natl Acad Sci* 2022;**119**(11):e2122954119.
105. Zhang Z, Xu M, Jamasb A, *et al.* Protein representation learning by geometric structure pretraining. Preprint, arXiv:2203.06125, 2022.
106. Ingraham J, Garg V, Barzilay R, Jaakkola T. Generative models for graph-based protein design. *Adv Neural Inf Process Syst* 2019;**32**.
107. Li J, Luo S, Deng C, *et al.* Orientation-aware graph neural networks for protein structure representation learning. 2022. https://openreview.net/forum?id=WcTLZrpzfe.
108. Cang Z, Wei G-W. TopologyNet: topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput Biol* 2017;**13**(7):e1005690.
109. Morris C, Ritzert M, Fey M, *et al.* Weisfeiler and Leman go neural: higher-order graph neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. **33**. Palo Alto, California USA: AAAI Press, 2019, 4602–9.
110. Ebli S, Defferrard M, Spreemann G. Simplicial neural networks. Preprint, arXiv:2010.03633, 2020.
111. Wu NC, Dai L, Olson CA, *et al.* Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife* 2016;**5**:e16965.
112. Podgornaia AI, Laub MT. Pervasive degeneracy and epistasis in a protein-protein interface. *Science* 2015;**347**(6222):673–7.
113. Zhang Y, Jiang Y, Gao K, *et al.* Structural insights into the elevator-type transport mechanism of a bacterial ZIP metal transporter. *Nat Commun* 2023;**14**(1):385.
114. Freschlin CR, Fahlberg SA, Romero PA. Machine learning to navigate fitness landscapes for protein engineering. *Curr Opin Biotechnol* 2022;**75**:102713.
115. Hie BL, Yang KK. Adaptive machine learning for protein engineering. *Curr Opin Struct Biol* 2022;**72**:145–52.
116. Tian X, Wang Z, Yang KK, *et al.* Sequence vs. structure: delving deep into data driven protein function prediction. *bioRxiv* 2023, 2023–04.
117. Fox RJ, Davis SC, Mundorff EC, *et al.* Improving catalytic function by ProSAR-driven enzyme evolution. *Nat Biotechnol* 2007;**25**(3):338–44.
118. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Res* 2008;**36**(9):3025–30.
119. Zhang N, Chen Y, Lu H, *et al.* MutaBind2: predicting the impacts of single and multiple mutations on protein-protein interactions. *iScience* 2020;**23**(3):100939.
120. Cang Z, Wei G-W. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics* 2017;**33**(22):3549–57.
121. Aghazadeh A, Nisonoff H, Ocal O, *et al.* Epistatic Net allows the sparse spectral regularization of deep neural networks for inferring fitness functions. *Nat Commun* 2021;**12**(1):5225.
122. Dallago C, Mou J, Johnston KE, *et al.* FLIP: benchmark tasks in fitness landscape inference for proteins. *bioRxiv* 2021, 2021–11.
123. Bryant DH, Bashir A, Sinai S, *et al.* Deep diversification of an AAV capsid protein by machine learning. *Nat Biotechnol* 2021;**39**(6):691–6.
124. Qiu Y, Hu J, Wei G-W. Cluster learning-assisted directed evolution. *Nat Comput Sci* 2021;**1**(12):809–18.

125. Qiu Y, Wei G-W. CLADE 2.0: evolution-driven cluster learning-assisted directed evolution. *J Chem Inf Model* 2022;**62**(19): 4629–41.
126. Thean DGL, Chu HY, Fong JHC, *et al.* Machine learning-coupled combinatorial mutagenesis enables resource-efficient engineering of CRISPR-Cas9 genome editor activities. *Nat Commun* 2022;**13**(1):2219.
127. Georgiev AG. Interpretable numerical descriptors of amino acid space. *J Comput Biol* 2009;**16**(5):703–23.
128. Shen L, Feng H, Qiu Y, Wei G-W. SVSBI: sequence-based virtual screening of biomolecular interactions. *Communication Biology* 2023;**6**:536.
129. Luo Y, Jiang G, Yu T, *et al.* ECNet is an evolutionary context-integrated deep learning framework for protein engineering. *Nat Commun* 2021;**12**(1):1–14.
130. Greenman KP, Soleimany A, Yang KK. Benchmarking uncertainty quantification for protein engineering. In: *ICLR2022 Machine Learning for Drug Discovery*, 2022.
131. Rasmussen, Carl Edward. Gaussian processes in machine learning. In: Bousquet, Olivierand von Luxburg, Ulrikeand Rätsch, Gunnar (eds). *Advanced Lectures on Machine Learning: ML Summer Schools*. Berlin Heidelberg: Springer, 2003, 63–71.
132. Bedbrook CN, Yang KK, Robinson JE, *et al.* Machine learning-guided channelrhodopsin engineering enables minimally invasive optogenetics. *Nat Methods* 2019;**16**(11):1176–84.
133. Bedbrook CN, Yang KK, Rice AJ, *et al.* Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLoS Comput Biol* 2017;**13**(10):e1005786.
134. Saito Y, Oikawa M, Nakazawa H, *et al.* Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synth Biol* 2018;**7**(9):2014–22.
135. Greenhalgh JC, Fahlberg SA, Pfleger BF, Romero PA. Machine learning-guided acyl-ACP reductase engineering for improved in vivo fatty alcohol production. *Nat Commun* 2021;**12**(1): 5825.
136. Romero PA, Krause A, Arnold FH. Navigating the protein fitness landscape with gaussian processes. *Proc Natl Acad Sci* 2013;**110**(3):E193–201.
137. Bubeck S, Munos R, Stoltz G, Szepesvári C. X-armed bandits. *J Mach Learn Res* 2011;**12**(5):1655–95.
138. Munos R. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Adv Neural Inf Process Syst* 2011;**24**:783–91.
139. Creswell A, White T, Dumoulin V, *et al.* Generative adversarial networks: an overview. *IEEE Signal Process Mag* 2018;**35**(1): 53–65.
140. Gupta A, Zou J. Feedback GAN for DNA optimizes protein functions. *Nat Mach Intell* 2019;**1**(2):105–11.
141. Shin J-E, Riesselman AJ, Kollasch AW, *et al.* Protein design and variant prediction using autoregressive generative models. *Nat Commun* 2021;**12**(1):2403.
142. Bachas S, Rakocevic G, Spencer D, *et al.* Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness. *bioRxiv* 2022, 2022–08.
143. Castro E, Godavarthi A, Rubinfien J, *et al.* Transformer-based protein generation with regularized latent space optimization. *Nat Mach Intell* 2022;**4**(10):840–51.
144. Baek M, DiMaio F, Anishchenko I, *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* 2021;**373**(6557):871–6.
145. Kandathil SM, Lau AM, Jones DT. Machine learning methods for predicting protein structure from single sequences. *Curr Opin Struct Biol* 2023;**81**:102627.
146. Chowdhury R, Bouatta N, Biswas S, *et al.* Single-sequence protein structure prediction using a language model and deep learning. *Nat Biotechnol* 2022;**40**(11):1617–23.
147. Wang W, Peng Z, Yang J. Single-sequence protein structure prediction using supervised transformer protein language models. *Nat Comput Sci* 2022;**2**(12):804–14.
148. Wu R, Ding F, Wang R, *et al.* High-resolution de novo structure prediction from primary sequence. *bioRxiv* 2022, 2022–07.
149. Fang X, Wang F, Liu L, *et al.* HelixFold-Single: MSA-free protein structure prediction by using protein language model as an alternative. Preprint, arXiv:2207.13921, 2022.
150. Barrett TD, Villegas-Morcillo A, Robinson L, *et al.* So manyfolds, so little time: efficient protein structure prediction with pLMs and MSAs. *bioRxiv* 2022, 2022–10.
151. Wu J, Wu F, Jiang B, *et al.* tFold-Ab: fast and accurate antibody structure prediction without sequence homologs. *bioRxiv* 2022, 2022–11.
152. Weissenow K, Heinzinger M, Steinegger M, Rost B. Ultra-fast protein structure prediction to capture effects of sequence variation in mutation movies. *bioRxiv* 2022, 2022–11.
153. Bordin N, Dallago C, Heinzinger M, *et al.* Novel machine learning approaches revolutionize protein knowledge. *Trends Biochem Sci* 2022;**48**(4):345–59.
154. Chidyausiku TM, Mendes SR, Klima JC, *et al.* De novo design of immunoglobulin-like domains. *Nat Commun* 2022;**13**(1):5661.
155. Keros AD, Nanda V, Subr K. Dist2Cycle: a simplicial neural network for homology localization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. **36**. Palo Alto, California USA: AAAI Press, 2022, 7133–42.
156. Schuster SC. Next-generation sequencing transforms today's biology. *Nat Methods* 2008;**5**(1):16–8.
157. Sarkisyan KS, Bolotin DA, Meer MV, *et al.* Local fitness landscape of the green fluorescent protein. *Nature* 2016;**533**(7603): 397–401.