



Review

Protein Function Analysis through Machine Learning

Chris Avery^{1,†}, John Patterson^{1,†}, Tyler Grear^{1,2,†} , Theodore Frater¹ and Donald J. Jacobs^{2,*} 

¹ Department of Bioinformatics and Genomics, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

² Department of Physics and Optical Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA

* Correspondence: djacobs1@uncc.edu

† These authors contributed equally to this work.

‡ Affiliate faculty of the UNC Charlotte School of Data Science.

Abstract: Machine learning (ML) has been an important arsenal in computational biology used to elucidate protein function for decades. With the recent burgeoning of novel ML methods and applications, new ML approaches have been incorporated into many areas of computational biology dealing with protein function. We examine how ML has been integrated into a wide range of computational models to improve prediction accuracy and gain a better understanding of protein function. The applications discussed are protein structure prediction, protein engineering using sequence modifications to achieve stability and druggability characteristics, molecular docking in terms of protein–ligand binding, including allosteric effects, protein–protein interactions and protein-centric drug discovery. To quantify the mechanisms underlying protein function, a holistic approach that takes structure, flexibility, stability, and dynamics into account is required, as these aspects become inseparable through their interdependence. Another key component of protein function is conformational dynamics, which often manifest as protein kinetics. Computational methods that use ML to generate representative conformational ensembles and quantify differences in conformational ensembles important for function are included in this review. Future opportunities are highlighted for each of these topics.

Keywords: machine learning; protein structure prediction; protein–protein interactions; protein dynamics; protein function; allostery; conformational sampling; force fields; molecular docking



Citation: Avery, C.; Patterson, J.; Grear, T.; Frater, T.; Jacobs, D.J. Protein Function Analysis through Machine Learning. *Biomolecules* **2022**, *12*, 1246. <https://doi.org/10.3390/biom12091246>

Academic Editors: Jianlin Cheng, Vladimir N. Uversky and Irina Nesmelova

Received: 16 July 2022

Accepted: 31 August 2022

Published: 6 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Our review presents important questions about proteins and their functions that have been around for several decades in terms of molecular biology, biochemistry, and biophysics. This scientific knowledge is where practiced researchers have an advantage; however, there is a major difference in how machine learning (ML) is used in computational biology today compared to how it was used 20 years ago [1]. There is now a convergence between the two fields as illustrated in Figure 1. This new way of thinking puts computational biology at a stage where problems that were unsolvable by old methods can become solvable, or at least limited solutions can provide increased accuracy and/or speed by incorporating ML methods. The promise that ML can help solve challenging problems applies to all areas of science, engineering, and beyond (e.g., medicine, art, music, economics, public policy) [2–4].

In this review, we make no distinction between old or new ML methods, or whether the ML method is based on clustering, decision trees, support vectors, projection pursuit or deep/shallow neural networks. If any form of ML has been used to help elucidate protein function, we mention the method if we found it in our literature searches. For those ML methods that escaped our attention, this was not intentional, and we regret our oversight. We also discuss experimental data when they are a critical component of a method in

computational biology, such as the prediction of protein structure. We discuss limited aspects of experimental methodology only if this process is part of a method within computational biology. Not all aspects of computational biology are covered in equal depth, either because of our bias toward what we believe is the most interesting aspects of protein function, or because of what is available in the computational biology literature. The biases we have towards the topics covered in this review are unabashedly applied to help researchers fuse structural bioinformatics and computational biology with ML.

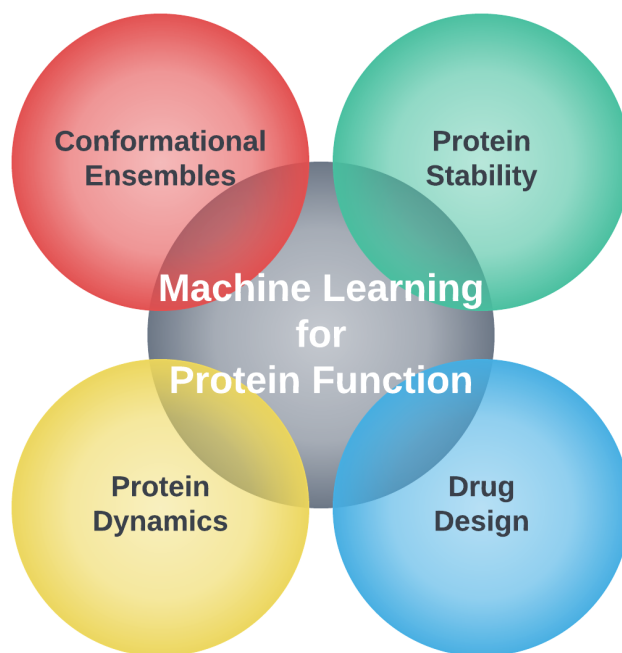


Figure 1. Machine learning meets computational biology. This review is about how machine learning (gray center circle) intersects with multiple aspects of computational biology (colored circles).

The rapid introduction of novel ML methods has already transformed the field of computational biology [5]. Experts in the field and early doctoral students are on close footing regarding development of new methods that use ML at a fast pace, which is not expected to slow down soon. This environment may foster many successful handshakes between old and new schools of thought, creating a breeding ground for new ideas. We believe it would be a mistake to promote a particular methodology that uses ML without performing an independent and extensive fair comparison test between methods. This is often carried out in the field of computational biology with certain blind competitions, which we cite when available. Besides making the scope of the review too large by including technical and controlled head-to-head comparisons, the reader should be aware that changes are being made so quickly that a research group will take ideas from others and create a competitive scheme in turn-key fashion. With easy-to-use ML tools and rapid developments in artificial intelligence (AI), this surge in high-intensity research is beneficial. Nevertheless, there are two frustrating elements that will occur before things start to settle down. Firstly, not all good ideas will survive, and not all ideas that survive will be explainable. Secondly, it is unfortunate that not all results will be reproducible, because researchers generally do not always precisely define the methodology and specify critical details. One way to mitigate these problems is to set standard datasets for training models to establish controls and benchmarks. This is happening in some areas of computational biology and ML [6] that we briefly mention.

Given the emergence of a new playing field that crosses ML with computational biology, we have set out to answer this two-part question: what is the current state of the art, and how does it further our knowledge on protein function? By reading the literature, we have answered this question fairly robustly. We hope that this review will help others

as much as it has helped us understand the changing landscape in computational biology in terms of protein function analysis using ML. Section 2 of the paper highlights the biophysical aspects of protein function and presents a brief historical overview of how computational biology and machine learning have converged. Section 3 then discusses a wide range of computational biology applications using ML. We conclude in Section 4, where we highlight the remaining challenges and suggest future opportunities.

2. Foundational Concepts of Protein Function

The biochemical activity of life is orchestrated by proteins, polymer chains of amino acid molecules called residues. Their function can be autonomous or in concert, as multiple-chain complexes. They play a critical biological role by conducting interactions and controlling a myriad of processes in the crowded molecular environment of cells. The *in vivo* environment consists of a large and diverse population of proteins mixed with other molecules that include osmolytes, ions, fatty hydrocarbons, and other macro- and meso-molecules. This large mixture of molecules within cells and at other locations of a living organism, such as the extracellular matrix, continuously undergoes chemical reactions driven by thermodynamic and kinetic considerations. To maintain reliability in the dynamically changing network of chemical reactions, proteins evolved to maintain a functionally relevant conformational ensemble that satisfies thermodynamic and kinetic stability conditions [7]. Protein function builds on the physical and chemical aspects of its environment that can drive specificity for a particular process. This specificity is encoded in the primary structure to form the basis for the central dogma of biology and the sequence–structure–function paradigm in structural biology. However, the sequence–structure–function paradigm is shallow, analogous to the title of a story, whereas the description of protein function requires a novel for each protein.

Christian Anfinsen and co-workers showed that the 3D structure of proteins is due to thermodynamics [8]. Thermodynamics spontaneously drive the protein into a stable, free-energy basin that is consistent with the environment. In proteins, structure rather than sequence tends to be conserved among proteins that perform the same function, even in proteins that are analogous across species [9,10]. From this perspective, a protein's 3D structure is arguably the most important physical attribute of a protein. Nevertheless, a single structure is a representation of an ensemble of conformations. Indeed, an intrinsically disordered protein (IDP) has an ensemble of conformations that cannot be accurately represented by a single native 3D structure [11]. In general, a protein is rigid when its conformational ensemble acutely preserves a native 3D structure, or perhaps it is so flexible that there is no conserved single domain that can be identified within the conformational ensemble. There is a broad spectrum of conformational characteristics that bridge these two extremes.

The common characteristic responsible for protein function is how it interacts with partners [12–14]. The strength of a protein's interaction with other proteins or types of molecules can range from weak to strong, depending on its sequence and environment. The interaction between different partners is modulated by the environment, which can include the effects of post-translational modulation on a protein, such as glycosylation, gradients of solutes or solvents, or state changes such as pressure, pH, or temperature. The environment controls a large network of intra- and inter-molecular interactions that strongly depend on molecular concentrations. Although observed properties of a protein depend on its environment, it is common to focus on intrinsic properties where environmental effects are largely ignored. This approach is fruitful because the conditions for most life on planet Earth are constrained. As water is essential for life as we know it, organisms have evolved to live in extreme conditions such as in hot springs or deep within the ocean, but generically operate under similar molecular constraints. This can be observed for a given family of proteins from organisms living in diverse environments that share the same biomolecular function [15].

In general, proteins that are members of the same functional family lose their function when they are placed in conditions too different from their native environment. When comparing proteins in their respective native environments, we find that they tend to have similar intrinsic properties [16–18]. A careful analysis shows that proteins within the same functional family have large segments of their sequence conserved. This can be determined by aligning protein sequences using bioinformatic sequence alignment methods [19–21]. Multiple sequence alignment reveals corresponding amino acids between proteins when accounting for the possibility of mutations, insertions, and deletions [22,23].

As illustrated in Figure 2 the important components for quantifying protein function are sequence, structure, stability, dynamics and knowledge of the partners with which a protein interacts: ligands, ions, membranes and other proteins. A ligand can be a protein, and protein–protein interactions are an important aspect of protein function that is central in systems biology [24,25]. When applied to understanding the characteristics of proteins and their functions, computational biology rarely attempts to model all the effects described above at the same time. Different aspects of protein function are usually considered separately, and approximations are inevitably made [26,27]. This review examines how ML is used to better quantify protein function, and takes into account aspects of sequence, structure, dynamics and binding.

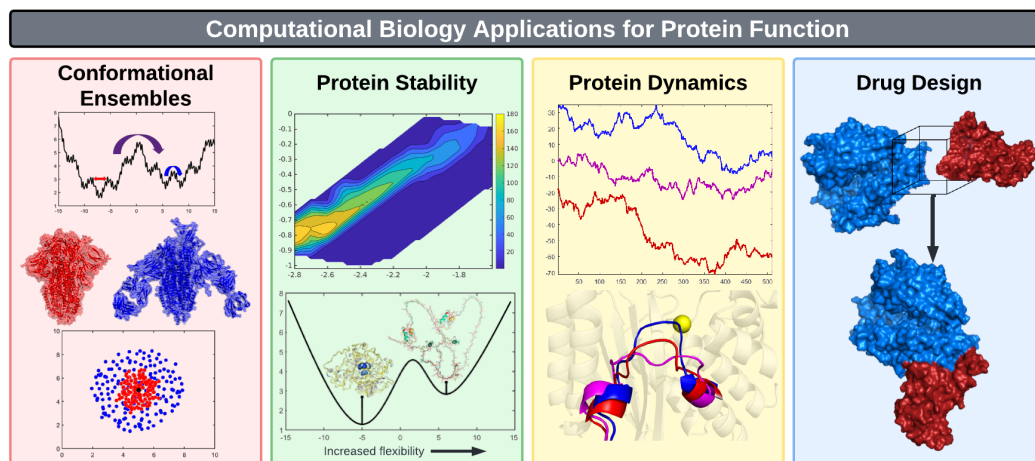


Figure 2. Four pillars of computational biology: The role of protein structure is critical in all panels. Conformational ensembles, red panel: ML helps with: (i) enhanced sampling; (ii) identifying collective variables; (iii) automated potential biasing; and (iv) Markovian state space partitioning. Protein stability, green panel: ML helps with (i) modeling the role of environment; (ii) protein engineering through mutagenesis; (iii) characterization of protein–protein interactions; and (iv) the role of rigidity for protein function. Protein dynamics, yellow panel: ML helps with (i) protein flexibility/conformational dynamics; (ii) dynamic allostery; and (iii) potential energy/force fields. Drug discovery, blue panel: ML helps with (i) molecular docking; and (ii) binding affinity prediction.

2.1. Elucidating Protein Function Using Computational Biology: A Short History

In the early days of computational biology, upon the advent of the digital computer, the main interest was using computers to help with taxonomy, where similar traits classify organisms. Many experimental comparative studies at the molecular level of proteins were also conducted, which helped establish the idea of molecular evolution [28]. The emphasis gradually shifted to genetics and evolutionary relationships found in DNA sequences, especially after the introduction of dynamic programming for alignments presented by Needleman and Wunsch [29].

In the 1970s, structural considerations of proteins began to be of interest. As it became easier to solve the inverse problem for structure determination using X-ray crystallography, protein structures became more readily available. In 1971, the protein data bank (PDB) was initiated with seven protein structures [30]. The forethought for building such a database

is remarkable, as it has been an indispensable resource for serving as the backbone of structural bioinformatics and computational biology.

Computationally addressing how proteins dynamically fold also started in the 1970s [31], although computer power and molecular models were limited. In the late 1980s and throughout the 1990s, simplified bead models, including simulations on lattices [32], were used to understand the basic physics of the protein folding process. It was also common to consider unfolding from native states, expanding upon the Go model [33], which also increased the need to know about more protein structures. In the early 1990s, it became clear that accurate prediction of protein structure is essential when it comes to understanding protein function, even if the dynamic protein folding pathways remain unknown. Protein function began to be simulated in the late 1990s once molecular dynamics (MD) of proteins exhibited numerical stability over long timescales (approaching milliseconds with large computing resources). These computational studies required starting with a known protein structure from the PDB.

As attention began to shift to native state dynamics of globular proteins, the mechanisms of action underlying protein function were able to be studied computationally. This included mutation studies and monitoring changes in dynamics of proteins as certain residues mutated (usually on timescales of 1 μ s or less). Native state dynamics also stimulated interest in boosting the accuracy of dynamic allostery models [34]. This capability naturally led to the development of models and algorithms for computational protein design and protein stability prediction. Unfortunately, MD simulation is not an appropriate tool for calculating protein stability because of its inability to properly generate ensembles large enough to calculate thermodynamic properties, not to mention approximations in force fields.

To avoid brute force MD simulation, thermodynamic models that assumed additivity in free-energy components were proposed to rapidly calculate thermodynamic quantities. Unfortunately, these simple additive models failed miserably [35] due to the non-additivity of conformational entropy [36]. However, in the early 2000s, with proper accounting for molecular constraints using rigidity theory, the non-additivity found in conformational entropy was accurately incorporated using a distance constraint model (DCM) [37]. For example, the DCM accurately described the heat capacity of proteins [38–40], including cold denaturation, because the model accounts for solvent effects [41]. From the DCM, quantitative stability and flexibility relationships are determined for proteins [16], which has been useful for understanding protein evolution and helps with protein design [42–44].

In tandem with interest in native state dynamics, starting in the mid 1990s, there was a strong interest in docking small molecules to proteins, characterize binding sites in proteins, and estimate binding affinities through computational means. Many docking methods were created, but once again these methods lacked the proper conformational sampling that considers the flexibility of the protein and the ligand. In addition, additive models for free-energy shifts used in scoring functions generally ignored entropic contributions to binding. For these reasons, developing methods for accurate molecular docking has been a major challenge in computational biology. The need for efficient and accurate docking methods in drug discovery drove the development of such methods, which led to large databases of potential small-molecule drugs, and myriad algorithms that combined data mining with quantitative structure–activity relationships (QSAR) [45].

These days, a systems biology approach is popular for tracking protein–protein interactions, which is critically important for biological regulation and maintaining homeostasis. However, predicting protein–protein interactions involves a docking problem that is particularly difficult because of the innate flexibility and size of binding regions on proteins, as well as the accuracy of scoring functions. This topic represents a current challenge in computational biology.

2.2. Convergence of Machine Learning and Computational Biology

Along with computational biology, ML has its own interesting historical development. Initially, methods for discriminant analysis were developed in statistics that are now considered ML [46,47]. However, the term “machine learning” was first coined by Arthur Samuel, who developed a computer program to play checkers in the early 1950s [48,49]. Frank Rosenblatt coined the term “perceptron” by combining the ideas of neural biology and optimization of game-playing objectives introduced by Samuel [50]. Over the years, artificial neural networks have gained popularity due to better algorithms and faster/larger computers. Especially with GPUs, deep learning with several or more perceptron layers is tractable. Moreover, the concept of rectifying unit or activation function has generalized and replaced the original perceptron concept.

At present, many ML methods are linked to statistical analysis, with optimization through automated procedures to draw statistical conclusions. The domain of ML is often divided into three main categories: unsupervised, supervised and reinforcement learning. Unsupervised and supervised learning uses data to train a model to perform a particular task. Unsupervised learning methods aim to learn patterns within a dataset without additional categorical information on how the data are structured. Supervised learning uses additional information during the training, such as class labels. Reinforcement learning takes the approach of self-learning, in which an agent learns to perform a particular action through trial and error, guided by a reward system, to perform a particular task when triggered by certain environmental conditions.

Other frameworks of ML have emerged, such as semi-supervised learning, which uses both labeled and unlabeled data to learn a particular task. The goal of ML is to train a model to perform a useful action or task. Some common functions of ML algorithms include: clustering, binary and multi-class classification, regression, generative modeling, natural language processing (NLP) and dimensionality reduction (DR). Table 1 provides an overview of general ML algorithms, along with where they fall in the landscape of ML.

Table 1. Survey of commonly used machine learning models. Abbreviations defined here are used throughout this review.

Method	Task	Paradigm	Abbreviation
k-Means	clustering	Unsupervised	-
Agglomerative Clustering	clustering	Unsupervised	-
Spectral Clustering	clustering	Unsupervised	-
Self-Organizing Maps	clustering	Unsupervised	SOM
Principal Component Analysis	DR	Unsupervised	PCA
Time-Lagged Independent Component Analysis	DR	Unsupervised	tICA
t-Distributed Stochastic Neighbor Embedding	DR	Unsupervised	t-SNE
Supervised Projection Learning for Orthogonal Completeness	clustering/classification/DR	Unsupervised	SPLOC
Naive Bayes	classification/regression	Supervised	NB
Support Vector Machines	classification	Supervised	SVM
Decision Trees	classification/regression	Supervised	DT
Random Forest	classification/regression	Supervised	RF
Gaussian Mixture Model	classification/regression	Supervised	GMM
Artificial Neural Network	classification/regression/NLP	Supervised	ANN
Shallow Neural Network	classification/regression/NLP	Supervised	SNN
Deep Neural Network	classification/regression/NLP	Supervised	DNN
Convolutional Neural Network	classification/regression/NLP	Supervised	CNN
3D Convolutional Neural Network	classification/regression/NLP	Supervised	3D-CNN
Recurrent Neural Network	classification/regression/NLP	Supervised	RNN
Autoencoder	DR/generative modeling	Supervised	-
Variational Autoencoder	DR/generative modeling	Supervised	VAE
Generative Adversarial Network	classification/generative modeling	Supervised	GAN
Message Passing Neural Network	classification/regression/DR	Supervised	MPNN
Graph Neural Network	classification/regression/DR	Supervised	GNN
Graph Convolutional Neural Network	classification/regression/DR	Supervised	GCNN

Table 1 demonstrates the breadth of different types of algorithms which make up the current landscape of the ML field. This landscape is constantly being sculpted by new

developments and, as such, is constantly changing. Currently, method development has accelerated dramatically in many areas of ML, such as generative modeling and NLP, which has made its way into the latest methods in computational biology. Generative modeling refers to algorithms which are able to learn patterns from the training data and use these patterns to create new data points from scratch. These methods have been used extensively in application domains such as image analysis. The variational autoencoder (VAE) [51] is an example of a generative model. VAEs consist of two networks that learn to encode samples to a low-dimensional space and decode (reconstruct) samples from the low-dimensional space, respectively. After training, the latter network can be used independently to construct new samples. VAEs are widely used in computational biology for dimension reduction, as well as its generative function. Generative Adversarial Networks (GAN) [52] are another example of popular generative models where the model consists of a generator and a discriminator. The generator constructs random samples, while the discriminator tries to identify which of two samples presented to it are real. The two are trained in opposition to each other, where the generator learns to trick the discriminator by generating more realistic samples. The VAE and GAN generative models have recently made their way into computational biology applications.

Natural language processing (NLP) is popular in ML research, where the goal is to analyze and generate sequence data such as text analysis/speech recognition or AI-powered chatbot models. Traditionally, recurrent models such as LSTM [53] have been used for these tasks because they inherently treat the data as a sequence. More recently, transformer networks are used in highly popular models such as GPT-3 or BERT [54,55] for text generation. Transformer networks have made several advancements in sequence models [56]. In particular, the inclusion of attention mechanisms allowed the model to put emphasis on the order of inputs, providing context and long-range correlations that can be exploited in an input sequence. NLP models are a popular choice in computational biology with respect to data-mining protein and DNA sequences.

Another model gaining significant traction in both the ML and computational biology communities is graph neural networks (GNN) [57]. These models treat the data as a graph, or a set of nodes that each contain a vector of features connected by edges according to a topology. For example, the nodes may represent atoms in a molecule and the edges defined according to the bond structure. A mechanism called message passing allows information to flow across the edges of the graph according to a set of message-passing rules, updating the feature representation at the nodes each time messages are passed. This allows the model to learn deeply encoded representations of data structure, which can then be used in a top-level model that performs a task. End-to-end training allows the GNN to learn representations specific to the task being performed. Graph convolutional neural networks (GCNN) have been shown to be very effective for molecular fingerprinting [58,59] and are being explored in a wide array of applications in computational biology.

Although ML addresses some of the limitations of traditional methods in computational biology, it also has its own caveats. Many methods require large amounts of data to train the model. When the data are high dimensional, a preprocessing step of DR is often required. As the amount of data and variables analyzed increases, algorithm performance typically wanes due to the curse of dimensionality [60]. Another problem is over-fitting to random fluctuations present in training data, or under-fitting due to incomplete sampling, which is associated with lower variance in the training sub-sample compared to the population sample. Protein function analysis often deals with high-dimensional data and low statistics, which push ML to its limits. We are now seeing a convergence between ML and computational biology, because the field of ML is now addressing the sampling problems encountered in computational biology. Of course, advances in ML are driven by applications. This convergence will help advance computational biology as much as ML. We are at the beginning of a new chapter in computational biology, meaning many gold-standard algorithms will become obsolete in the near future.

3. Selected Applications of Machine Learning in Computational Biology

In this section, we will focus on four pillars of computational biology, as shown in Figure 2. Although well known, but often neglected in the way one visualizes protein function, the thematic connection we make through this section is conformational ensembles, and how ML can help advance computational biology by providing better low-dimensional representations of the characteristics of these ensembles.

3.1. Protein Structure Prediction

The vast number of functions performed by proteins are facilitated by their 3D structure. Accurate protein structure prediction (PSP) opens the door to engineering novel protein functions in medicine, furthering our understanding of mechanisms through simulation and experimentation. Historically, the two most important computational approaches to PSP are template-based and template-free modeling. Template-based modeling is based on the principle that similar amino acid sequences share the same folds [61–63]. As a result, known empirical structures can be used to model unknown structures: homology modeling. The protein structure initiative (PSI) of the early 2000s aimed to distribute 3D structural information for naturally occurring proteins through experimental methods such as: (i) X-ray crystallography (XRC); (ii) nuclear magnetic resonance (NMR); (iii) cryo-electron microscopy (cryo-EM), each with its advantages and disadvantages [64,65]. Other available experimental methods include SAXS, CD, FRET, and Raman spectroscopy [66]. Notably, many protein structures are resolved when bound to a ligand. Some protein structures are obtained without a bound ligand, and some proteins have many structures, each with a different ligand. This empirical information is extremely useful for training and testing the docking software [67] often needed in drug design.

The research community religiously deposits experimentally resolved protein structures (and other non-protein macro molecular structures) into the PDB, where these data are used to improve the accuracy of template-based modeling. Template-free modeling generates protein structures without the use of homologs in the PDB [68]. This method typically relies on physics-based energy functions and is much more computationally intensive than template-free modeling. Knowledge-based approaches dominate the field, yielding deterministic designs for stable proteins or homology-based structural predicts [69–71]. Of course, there are algorithms that combine the these approaches. With ML, the lines between template-based and template-free modeling have become more seamless, and combinations of the two approaches are more commonplace [62,72].

The critical assessment of methods of protein structure prediction (CASP) is a biennial competition where the amino acid sequences of experimentally determined structures are provided to participants through double-blind assignment. The experimental structures used in CASP competitions are initially not deposited into the PDB until after the competition. Participants are expected to submit predicted models using their unique method of determination. Models are evaluated using the Z-scores of backbone conformational similarity using various measures for accuracy. The most successful methods over the last few rounds of CASP are given in Table 2 and represented in Figure 3 [73]. Although ML has made great strides in the field of PSP there are methods such as Feig-R2, which used physics-based refinement through MD simulations [74]. The Feig-R2 method was competitive against the ML methods from the Baker and Zhang groups.

It is important to use ML whenever it enhances computational efficiency, but ML need not dominate an approach, nor turn everything into a black box model. However, having said this, we see ML methods such as AlphaFold and AlphaFold2 have produced substantially better results. Researchers should be aware that this new paradigm does not make other methods for PSP obsolete. CASP is a competition, but more importantly, CASP provides a conduit for PSP experts worldwide to share ideas. In our view, ML is a powerful multifaceted tool, but ML will not always be the best tool that solves a problem. Ideally, ML should be used to enhance models rooted in basic physics and chemistry principles.

Table 2. Top ML methods for PSP in the regular targets category from CASP12 to CASP14.

CASP	Method/Group	Year	Model ¹	Stand Alone ²	Webserver ³
12	Baker [75]	2016	Rosetta	Yes	Yes
	LEEab [76]	2016	GOAL	No	No
	Zhang [77]	2016	I-TASSER	Yes	Yes
	LEE [76]	2016	GOAL	No	No
	VoroMQA-select [78]	2016	VoroMQA	Yes	Yes
13	A7D (AlphaFold) [79,80]	2018	FreeM + DNN	Yes	No
	Zhang [81]	2018	TripletRes	Yes	Yes
	MULTICOM [82]	2018	CDNN + ab initio	Yes	Yes
	QUARK [83]	2018	C-QUARK	Yes	Yes
	Zhang-Server [83]	2018	C-I-TASSER	Yes	Yes
14	AlphaFold2 [84]	2020	TR/ATT + ANN	Yes	Yes
	Baker [85]	2020	ResNet + trRosetta	Yes	Yes
	Baker-exp. [86]	2020	ResNet + trRosetta	Yes	Yes
	FEIG-R2 [87]	2020	PREFMD2	Yes	Yes
	Zhang [88]	2020	UnknownI ⁴	No ⁴	No ⁴

¹ This column lists the name of the model used by the CASP competition group. Due to the extensive architectures used, some models do not have ML shorthand nomenclature. In-depth descriptions of the models can be found in the corresponding reference. ² This column indicates whether a standalone program is available (“Yes” or “No”). The hyperlink for “Yes” redirects to the corresponding repository. ³ This column indicates whether a webserver is available to users (“Yes” or “No”). The hyperlink for “Yes” redirects to the corresponding webserver URL. ⁴ The exact model was ambiguous, as this lab submitted multiple high-performing models.

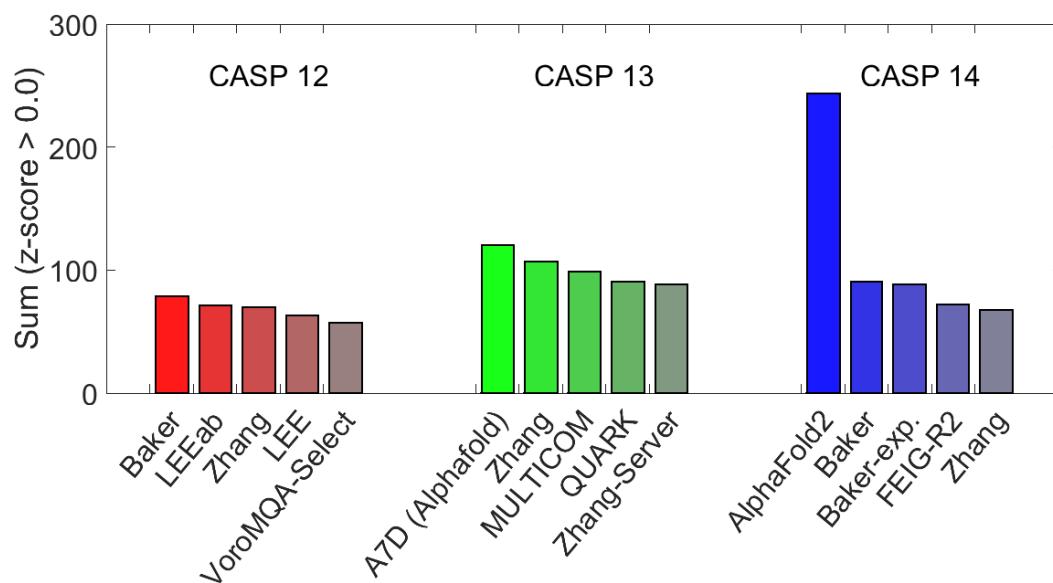


Figure 3. CASP 12–14 top 5 competitors per year (x-axis). The performance for each competition was based primarily on the summation of positive Z-scores (y-axis) with respect to GDT_TS for each of the proposed structure prediction models. The accuracy metric, GDT_TS, is a multiscale indicator for the proximity of C α atoms in a model to those in the corresponding experimental structure.

Debuting at CASP-Round XIV, AlphaFold2 by Google Deep Mind has raised the bar in PSP by achieving prediction accuracy near the experimental limits, and scoring in the competition far higher than their competitors [84]. AlphaFold2 is a completely reworked model of the AlphaFold method presented at CASP13 [73,89]. The success of AlphaFold2 comes from the unique implementation of deep neural networks and evolutionary history via computed MSA [72,84,90–92]. The high accuracy of AlphaFold2 should not be seen

as a sign that we are close to the endgame for PSP. Rather, the success of AlphaFold2 points to new horizons to explore, shown by similar works such as RoseTTAfold [93] or an alphafold reproduction such as openfold [94]. Naturally, many areas of improvement are needed, such as accounting for the role of the environment or predicting structures without the need for multiple sequence alignments (MSAs), as seen in OmegaFold [95]. A new chapter has begun, and programs such as openFold [94] and RoseTTAfold [93] are emerging. OpenFold is a recreation of AlphaFold using pytorch and RoseTTAfold is a “three track” neural network [93,94].

It cannot be overemphasized that once structures are determined, mechanisms of action can be better understood by other computational methods. With a model structure in hand, new therapeutics can be developed much more rapidly, often enhanced by computer-aided structure-based drug discovery (SBDD) methods [96,97]. The effectiveness of SBDD has been enhanced by ML as well as discussed in [98,99]. Of course, many proteins cannot be crystallized because they have intrinsically disordered regions [100]. Sometimes only parts of proteins can be crystallized, as there are proteins with stable native structure combined with disordered regions. This raises a more general question of what a model structure means. The crystal structure represents a snapshot of the protein in an environment different from in vitro and in vivo conditions. In solution, proteins explore many conformations, and those that are accessible are called an ensemble. Structure prediction must shift to predicting distributions that model conformational ensembles, which is discussed next.

3.2. Conformational Ensembles

The native state of a protein usually refers to a folded structure, which is a functionally relevant conformation of the molecule within the global minimum of the free-energy surface (FES). From the perspective of the free-energy landscape, the native state of a protein corresponds to a native free-energy basin associated with a native conformational ensemble. Zooming into the native free-energy basin at high resolution generally reveals multiple functionally relevant basins separated by low free-energy barriers. MD simulations provide a means to explore the conformational ensemble of a native basin, and possibly transition into metastable basins at longer time scales due to larger free-energy barriers that separate basins for metastable states. Thermal fluctuations typically govern dynamic processes on millisecond or longer timescales where many functional processes take place, as shown in Figure 4. This puts a severe limitation on MD simulations to reach time scales that are functionally relevant due to computing constraints.

The practical goal of performing MD simulations is to observe molecular processes in more detail than can be obtained from experiments by sampling conformations in the form of a time series. A theoretical goal that is difficult to achieve in practice is to compute thermodynamic properties through the partition function of statistical mechanics. How much information can be extracted from time series data greatly depends on whether the process is stationary, and the sampling adequacy. The partition function requires a sum over all accessible states of a system with appropriate statistical weights. Fortunately, almost all accessible states have negligible probability. Only the most probable states need to be explored in practice. Furthermore, by obtaining probability distributions based on counting from sampling, it is possible to obtain free energy without directly calculating a partition function. While these simplifications tremendously help mitigate the problem of under-sampling, barriers in the FES greatly hinder MD from exploring the functionally relevant conformational space. In general, accuracy increases as more conformations are sampled from different free-energy basins. To increase the ability for MD to sample a wider range of conformations, it is necessary to consider enhanced sampling methods.

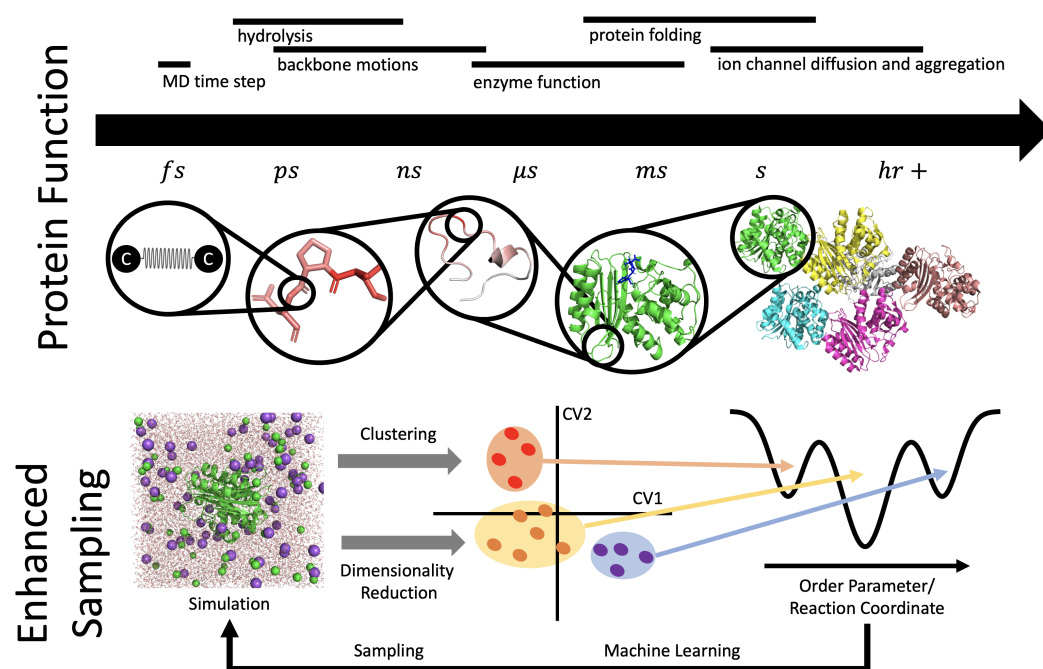


Figure 4. Time scales of protein function. Protein function occurs on time scales that span many orders of magnitude. MD simulation time steps are limited to the femtosecond range, necessitating enhanced sampling methods for the analysis of long time scale processes. Enhanced sampling typically starts with an unsupervised simulation of proteins to initially explore conformation space. Clustering and dimensionality reduction inform methods such as metadynamics on how to bias further simulations for the exploration of unsampled or poorly sampled regions in the free-energy landscape. Machine learning has been deployed to achieve better and faster sampling of this landscape.

3.2.1. Enhanced Sampling Methods

A few approaches are available to increase sampling to capture the relevant conformations for protein function. The oldest approach is to use coarse-grained models to simulate the motions of proteins on longer time scales [101–103]. All coarse-grained models sacrifice some information from the system to speed up calculations, achieving more sampling for the same amount of computation time. It is worth noting that there are other types of models that are not along the lines of MD simulation that model protein dynamics and/or conformational ensembles. The elastic network model (ENM) [104] and the DCM [37,38] are two very different examples of a coarse-grained model. A coarse-grained model commonly used today for MD simulation is the Martini force field [105,106], which replaces certain groups of about 4 atoms in amino acids by single elements called beads and applies physics-based effective potentials. Another choice is to use a forcefield based on the potential of mean force (PMF), such as the United Residue (UNRES) [107,108] model. This model groups atoms by their interactions, such as dipole interactions and electrostatics. Coarse-grained forcefields remove information about a structure it simulates; however, through backmapping [109], approximations to the original all-atom coordinates can be obtained. Several approaches have integrated deep learning into the prediction of coarse-grained forces, including DNNs [110,111], GNNs [112], and ensemble-based gradient boosting methods [113].

Replica exchange [114,115] is a physics-based method that involves performing many MD simulations in parallel at different temperatures. At a periodic time interval, the atomic coordinates are randomly swapped between systems and momenta scaled appropriately, which forces the conformations observed at different conditions to re-equilibrate. Higher-temperature systems will be able to move between FES minima more easily. One can also use the same idea with other variables, such as a collective variable or even forcefield

parameters [115,116], as seen in the method of Hamiltonian Replica Exchange or (Replica Exchange Umbrella Sampling).

Simulated annealing [117] can be used to generate conformations during the exploration of the FES, which is controlled by starting at a fictive high temperature and cooling slowly. However, in this method, many such runs are repeated so that different basins are explored. Steps are taken stochastically using Monte Carlo methods that always move to lower-energy states when found, but have some probability to move to a higher-energy state during the exploration process. At high temperatures, the system can take large steps to explore the FES, then, as the temperature decreases at a constant rate, the system explores the full shape of the FES. This method can be used, for example, in folding simulations to find the lowest energy conformation. When performed in a mass parallel scheme, such as a swarm approach, a set of lowest energy conformers can be found by aggregating results. This simulated annealing-driven MD approach enables the native protein structure (or a set of representatives for non-native states) to be determined [118].

Lastly, non-equilibrium MD combined with Monte-Carlo methods are used to enhance conformational sampling. This approach can be classified into two schemes: introducing biases based on collective variables that are predefined, or biases placed on the Hamiltonian of the system with a potential energy that fills in the FES the longer the system spends time in a basin. Both methods increase in bias to push the system out of a basin the longer it resides there. In this way, more basins can be explored much faster than in physical reality. The objective is to capture a diverse set of conformations, not to provide real-time dynamics. These methods are often referred to as metadynamics [119,120].

In the first class of methods, a biasing force estimates the free energy along a particular reaction coordinate by computing the average force, and computing the FES by thermodynamic integration. In the second class, one way to fill up free energy basins is to add positive potentials in the form of Gaussian functions. Another method called Adaptive Biasing Force (ABF) [121,122] uses this by invoking the concept of the PMF to try and flatten out the the potential barrier along a reaction coordinate by applying a force which counteracts the PMF. When bias forces are used to perturb the underlying potential energy, re-weighting of probabilities is required to obtain equilibrium statistics and [123].

These methods have been widely applied to MD simulation, and combinations of these classic methods with ML algorithms to further improve FES sampling have been explored extensively. In the following subsections, applications of ML to various aspects of these enhancement algorithms are reviewed.

3.2.2. Identifying Collective Variables

Many enhanced sampling algorithms rely on an *a priori* defined collective variable (CV), defined using expert knowledge by hand-selecting a set of residues to track coordinates, distance pairs or dihedral angles. Recent work has incorporated empirical data for CV selection, such as using co-evolutionary residue couplings [124] or the NMR S^2 parameter as a metric for conformational entropy [125]. More generally, ML provides an automated approach to CV discovery. The benefit of non-curated collective coordinates is that an unbiased approach to coordinate discovery allows hidden information with functional relevance to be identified without investigator bias.

The natural description of protein dynamics is the 3D coordinates of each of the N atoms in the molecule. However, the $3N$ dimensional space is too large to efficiently sample functionally relevant processes. Collective coordinates should be low-dimensional representations of the complex biomolecular process that is to be sampled [126,127]. The transformation to CVs should be a differential function of the $3N$ atomic coordinates. A traditional approach for finding collective coordinates is to use dimension reduction via unsupervised ML such as PCA [128]. Normally, it is best to use only a select set of heavy atoms, such as carbon alpha atoms, along the backbone. PCA has been implemented directly in many MD packages, including GROMACS, as well as part of standalone MD simulation analysis

software such as JEDi [129], MDAnalysis [130] or ModeTask [131]. These methods are excellent for extracting the large-scale motions from a protein.

Large-scale motions are effectively represented in a low-dimensional space and are used to accelerate MD simulation [132] and identify sampling boundaries of the FES to seed further rounds of MD simulations [133]. Non-linear approaches such as kernel methods or manifold learning provided added complexity, but removed much of the intuition for a learned CV. These methods include local linear embedding [134], isomap [135], sketch-map [136], and diffusionmap [137]. Recently it was demonstrated how classification algorithms in supervised ML, such as SVM and linear regression, can be used to define CVs to differentiate two known end states of a biochemical process [138].

A popular method for CV determination as well as analysis of MD simulation data is time-lagged Independent Component Analysis (tICA) [139]. ICA is a method of signal processing that aims to reduce a high-dimensional dataset into a small set of the most statistically independent components. The original aim of ICA was to solve the so-called “cocktail party” problem, in which a mixed signal is transformed into its component signals [140]. In tICA, the ICA problem is extended to the generalized eigenvalue problem:

$$C(t + t_0)V = C(t)\Lambda V ,$$

where V is a matrix whose columns represent the collective independent components, Λ is a diagonal matrix of eigenvalues, $C(t)$ is the system covariance matrix, and $C(t + t_0)$ represents the time-lagged covariance matrix, with t_0 being the lag time between samples being compared [141]. The time lag is chosen depending on the time-scale of the processes of interest. By solving this generalized eigenvalue problem, the independent components, V , that maximize autocorrelation are found. The eigenvalues, Λ , represent the autocorrelation of the data at the lag time. The time scale of the motions represented by the i -th independent component, τ_i is estimated by $\tau_i = \frac{t_0}{\ln(\lambda_i)}$ [139]. Usually the slowest changing processes are of interest. Because tICA is able to identify long time-scale motions, it has been used to generate CVs for metadynamics [142] and is widely used for constructing Markov models [143]. The software PyEMMA [144] and MSMBuilder [145] construct Markov models for molecular simulation using tICA, as well as other reaction coordinates.

Neural networks have also been used to parameterize CVs because they are able to be computed on the fly during the MD simulation. Moreover, neural networks are differentiable, which lends to computing biasing forces [146]. Autoencoders are neural network architectures which can be used for dimension reduction and generative modeling. A neural network called the encoder predicts a low-dimensional latent space representation of the input, and the latent space points can be used to reconstruct the input by a corresponding decoder neural network. The encoder then acts as the transformation to the CV, and the decoder provides the inverse function. Autoencoders have been used to find CVs or bias potentials for enhanced sampling of molecular systems [147]. Work has also been carried out to generalize these models to allow the latent space variables to express periodic CVs and impose a hierarchical ordering on learned CVs [148]. The ability to rank CVs occurs naturally in other algorithms such as PCA, and is generally useful if one wishes to interpret the information expressed by a CV.

The low-dimensional representations learned by autoencoder models have been used in several applications to construct a Markov state model (MSM) to characterize long-timescale motions [149,150]. Similar to tICA, a time-lagged version of the model aims to predict a sample or latent point t_0 later [151]. An ensemble of autoencoders can be used to iteratively sample and bias simulations on the fly, first trained on unbiased simulation and then used to determine conformational states that are poorly sampled. Furthermore, autoencoders can be combined with umbrella sampling to sample the FES [147].

The variation autoencoder, VAE, is used for probabilistic generative modeling. VAEs work similarly to autoencoders in that they learn an encoder and decoder model. However, rather than learning discrete points in the latent space, the model learns the parameters $\{\mu\}$ and $\{\sigma\}$ of a multivariate distribution over the latent space variables z . This distribution

is sampled to produce a point z_0 , which is fed into the decoder to reproduce the input x during training, and subsequently generates a new sample during test time. The model is trained to minimize both reconstruction error, so that the decoder can generate new points of x given a latent point z , and the Kullback–Leibler divergence between the learned distribution in the latent space of z and a normal distribution. This second training term constrains the learned distribution on z so that it may be efficiently sampled without the need for the encoder after training. VAEs have been used in combination with other methods to find reaction coordinates [152]. As an example, preselected features using the Automatic Mutual Information Noise Omission (AMINO) [153] to train a re-weighted Autoencoded Variational Bayes model (RAVE) [154] for a study on the dissociation of drugs from GPCR.

A related method which blends ideas from VAEs and tICA has recently been described for finding slow reaction coordinates by characterizing a MSM for protein dynamics, called the Variational Approach to Markov Processes (VAMP) [155]. In VAMP a variational method attempts to directly approximate the left and right singular functions of the Koopman operator which controls the time dynamics of the system on a set of low-dimensional coordinates [156–158]. In this approach [159], a neural network architecture called VAMPnets was developed, where two fully connected networks parallelly predict the transformation of \vec{x}_t and $\vec{x}_{t+\tau}$ onto the Koopman coordinates by training the network to minimize the VAMP-2 score (R_2), given by $R_2 = \|C_{00}^{-1/2}C_{0\tau}C_{\tau\tau}^{-1/2}\|_F^2$. C_{00} and $C_{\tau\tau}$ are the non-time-lagged covariance matrices of the Koopman coordinates at time $t = 0$ and $t = \tau$, $C_{0\tau}$ is the time-lagged covariance matrix, and $\|\cdot\|_F^2$ represents the Frobenius norm [159]. This approach has been used to explore the kinetic landscape of protein dynamics by using the learned Koopman coordinates to parameterize Markov models [160,161].

3.2.3. Automated Potential Biasing

Methods such as metadynamics [162] and adaptive biasing force (ABF) rely on perturbing the dynamics of the system, such that the model is driven toward regions of low sampling. To do this, models must either compute the biasing force or the biasing potential to add to the Hamiltonian of the system. The system is often projected onto CVs before biasing forces or potentials are computed to reduce the complexity of the problem. This process is performed adaptively on the fly, constructing the bias from many shorter MD simulation runs. In metadynamics, the bias is built by successively adding small Gaussian perturbations, which accumulate in well sampled areas of the FES, so that the bias potential effectively shadows the underlying potential energy surface. ABF attempts to lower barriers between free-energy basins by matching the estimated PMF. Both methods rely on sufficient sampling of the underlying potential energy surface before updates to the adaptive bias can be made.

Here, we highlight three examples of how neural networks can be used to compute adaptive bias potentials. The first method, called NN2B [163], is similar to metadynamics. However, NN2B replaces the computation of a biasing potential to add with a high-dimensional density estimator that gets smoothed over configuration space by an ANN. While the biasing potential cannot be computed in real time, the neural network and density estimation allows the model to handle higher dimensional representations. Another model, Force-biasing Using Neural Networks (FUNN) [164] trains a neural network to directly predict biasing forces on a system, rather than potential energy, at a given CV value based on Bayesian regularized neural networks [165]. The disadvantage of this method is that each bin in CV space must be sampled to train the network. However, once the network is trained, it can be used to predict the mean force for any value of the CV, even if it has not been previously sampled. The third method for enhanced sampling is Deep Enhanced Sampling of Proteins (DESP), which uses a VAE-based model [166].

3.2.4. Clustering Conformations and Markov Model State Space Partitioning

Clustering data is a task that is ubiquitously used in data science and within ML methods. This section will describe many clustering methods, but the main application of interest is to quantify the similarity between conformations. Clustering conformations is an essential element of constructing a MSM. Often, a trajectory can be clustered into hundreds to thousands of microstates to describe the kinetics of biophysical processes.

Clustering is an unsupervised ML task in which a space or set of inputs is divided into groups based on how similar or different inputs are from each other [167]. At the most basic level, clustering methods can be clustered into two types of attributes: (1) what similarity/distance measures to choose and (2) what clustering algorithm to follow. Often, the number of states is left as a hyper-parameter for the researcher to optimize. Clustering of a trajectory can occur in the full coordinate space of the system, but is more commonly applied after the system is transformed into collective coordinates which characterize the progression of the biophysical process being modeled.

Clustering algorithms are sensitive to the type of distance metrics used to quantify similarity between two conformations. A common approach is to use the RMSD and TM score [168]. The RMSD distance metric compares the euclidean distance between the 3D coordinates of two structurally aligned structures [129,169–171]. Although commonly used due to its simplicity, RMSD is sensitive to the size of the systems being compared. The TM score is normalized on the range $[0, 1]$ to make it independent of the system size. The TM score uses a weighting scheme to normalize distances at different size scales.

An ongoing issue in comparing molecular structures is the requirement of a consensus set of residues to facilitate the comparison. After the residues are selected, the conformations to be compared must first be structurally aligned into the same frame of reference. In general, as the consensus set of residues becomes large, distance comparisons become less useful because of the curse of dimensionality. This is why the coordinate space is often transformed into low-dimensional, informative representations prior to clustering. Linear distance metrics such as euclidean distance $d_{ij} = \sqrt{(\vec{x}_i - \vec{x}_j) \cdot (\vec{x}_i - \vec{x}_j)}$ or the Manhattan distance $d_{ij} = \sum |\vec{x}_i - \vec{x}_j|$ have been widely used for clustering applications. These metrics are part of the Minkowski family of distances and are natural choices for clustering vectored data such as conformational coordinates. However, this logical approach performs poorly when different features have different scales [172]. To remove the multiple-scale problem, features are normalized prior to clustering or non-linear distance metrics with saturation limits can be used. Some metrics that are commonly used in clustering are the mahalanobis distance $d_{ij} = \sqrt{(\vec{x}_i - \vec{x}_j)^T \Sigma_{ij} (\vec{x}_i - \vec{x}_j)}$, cosine similarity $d_{ij} = \vec{x}_i \cdot \vec{x}_j / \sqrt{\vec{x}_i^2 \vec{x}_j^2}$, or Pearson correlation [173].

As applied to molecular simulation, the major classes of clustering that are popular are: geometric or partitioning methods, hierarchical methods, and model-based methods including spectral clustering [167]. Geometric clustering typically divides the data into clusters by partitioning the space of points into different clusters. The k-means clustering algorithm is one of the simplest, yet most commonly used method for this application. It assigns the data to a cluster based on which of a set of k cluster centroids it is closest to. The centroid locations are recomputed, and this process is iterated until convergence is reached. Geometric clustering is easy to implement and has been effective with MD data, and thus has been included in analysis packages such as MSMBuilder [145].

Hierarchical clustering constructs a dendrogram of the data points by iteratively grouping data points or clusters of data points one at a time until either the root is reached, or a specified number of distinct clusters is found. Hierarchical clustering can be employed using a top-down approach which starts at the root and works its way to the leaves, or a bottom-up approach that starts with the leaves and works its way to the root. The bottom-up approach can be referred to as agglomerative clustering. An example of hierarchical methods used in clustering protein conformations is the Bayesian Agglomerative Clustering Engine (BACE) method.

BACE [174] is an agglomerative clustering method which merges microstates by computing the Bayesian likelihood factor, or BACE Bayes factor, that the micro-states belong to the same or different macrostates. This method was inspired by the hierarchical nature of protein FES, where microstates belonging to the same macrostate will be much more statistically similar to each other than microstates in another macrostate. This method addresses the problem of statistical uncertainty in PCCA and PCCA+.

It is worth noting that model-based clustering assumes an underlying model for the data, which is exploited to cluster data points. To analyze MD trajectories, spectral clustering methods are used, which decompose a particular matrix using eigen-vector/value (spectral) decomposition [175]. In particular, a set of data points is treated as nodes of a graph which are connected by edges whose weight corresponds to similarity. A graph Laplacian matrix, $L = 1 - D^{-1}W$ is constructed from the graph adjacency matrix W , converted to similarities by a transformation of the users choice, and diagonalized. The resulting matrix of the first n_c nonzero-eigenvalue eigenvectors (n_c being the number of clusters) per data point are used as a new vector representing the data point, and can be clustered by another method such as k -means. According to spectral clustering theory, the number of clusters within a dataset corresponds to the number of eigenvalues of the Laplacian matrix that are equal to 1, or practically, within a tolerance. The most commonly used spectral clustering approach is Perron-Cluster Cluster Analysis (PCCA) and Robust PCCA (PCCA+).

PCCA and PCCA+ [176,177] is a model-based spectral clustering method which is ideal for clustering states for Markov chains. It uses the transfer matrix of a Markov chain, P , to construct the Laplacian matrix, where in the above equation for L , $D^{-1}W = P$. For a perfectly uncoupled Markov chain, the matrix P will become block diagonal under appropriate permutations, indicating that data points within a cluster only form edges with other data points in the same cluster within the graph representation of the dataset. The eigenvalue spectrum will have a Perron Cluster of eigenvalues at 1, allowing for the spectral clustering. This method has been used to construct Markov models and analyze rare dynamic processes in MD simulation [178]. A weakness of this method is that a sufficiently large amount of MD data must be collected so that the transition matrix is well sampled.

Methods of clustering are not restricted to an established clustering method in ML. As an example, Super-Level-Set Hierarchical Clustering [179] blends multiple methodologies together as it clusters conformations using the density of states. Initially, the conformations are clustered into microstates via k -means and then hierarchically ordered based on the density of the state, which is proportional to the number of conformations within the state. The microstates are then divided into n clusters, such that the number of conformations (not microstates) are about equal between them, so that the resulting set with less microstates is more dense in conformation space. Density sets are accumulated into a super density set, such that the i -th super density level contains all of the density sets 0 to i . Spectral clustering is performed at each super density level to reveal kinetic clusters, which form successively more connected graph representations with increasing set level. This allows clustering to be performed hierarchically and reveals structured information about the underlying FES. Further extensions of this algorithm, called the Hierarchical Nyström Extension Graph, have been published by the same authors [180] to account better for the high-dimensional nature of protein dynamics.

Another approach to clustering is the Most Probable Paths algorithm [181], which takes a set of n microstates and converts them into more coarse-grained macrostates. The trajectory is first reduced into a low-dimensional form by PCA, then the free energy is computed and microstate transition probabilities are approximated via the assumption that $G_i = k_b T \log(P_i)$, where G_i is the free energy of state i , $k_b T$ is the Boltzmann constant times temperature, and P_i is the probability of being in state i . The transition path is followed by jumping to the nearest neighbor state with the highest probability (the lowest G_i) until the path terminates or reaches a loop. All states that terminate at the same lowest free-energy

state are lumped into a composite state, and the algorithm is repeated until the number of macrostates converges.

Other clustering methods are available, as previously reviewed [182] and described in much more detail. It is noteworthy to mention Renormalization Group Clustering [183], Automatic Partitioning for Multi Body Systems [184], Sapphire based clustering [185], and self-organizing maps as unsupervised neural networks [186]. It is clear from this brief survey that additional clustering algorithms that learn on MD simulation data will be developed and introduced. From our own research, we recommend a new direction toward developing conformational comparisons that are not directly tied to common sets of residues. The traditional approach of atom to atom comparison makes it difficult to compare conformational ensembles across larger sets of proteins that have large variations in sequence identity. This paradigm shift would require replacing structural alignments between corresponding atoms [129] in terms of physical properties that go beyond atomic coordinates. Such an approach would allow environmental and dynamic effects to be quantified, assist protein evolution studies and benefit ML methods for structure prediction.

3.3. Protein Stability

Stability is a question that is central to protein science. In the 1950s, Pauling's elucidation of the native contacts of the protein backbone in helical structures spurred the field of structural biology [7,187,188]. This quickly evolved to the current day perspective: stability of proteins is functionally important from the mechanical, thermodynamic and kinetic point of view as depicted in Figure 5. From the results of XRC, the common view among biochemists and molecular biologists is that functional proteins are stable with a well-defined, mostly rigid structure, and some flexible loop regions protruding into solvent. This corresponds to the case when a protein maintains a tight conformational ensemble at the bottom of a free-energy funnel, consistent with an enthalpic dominance in the free energy of the protein.

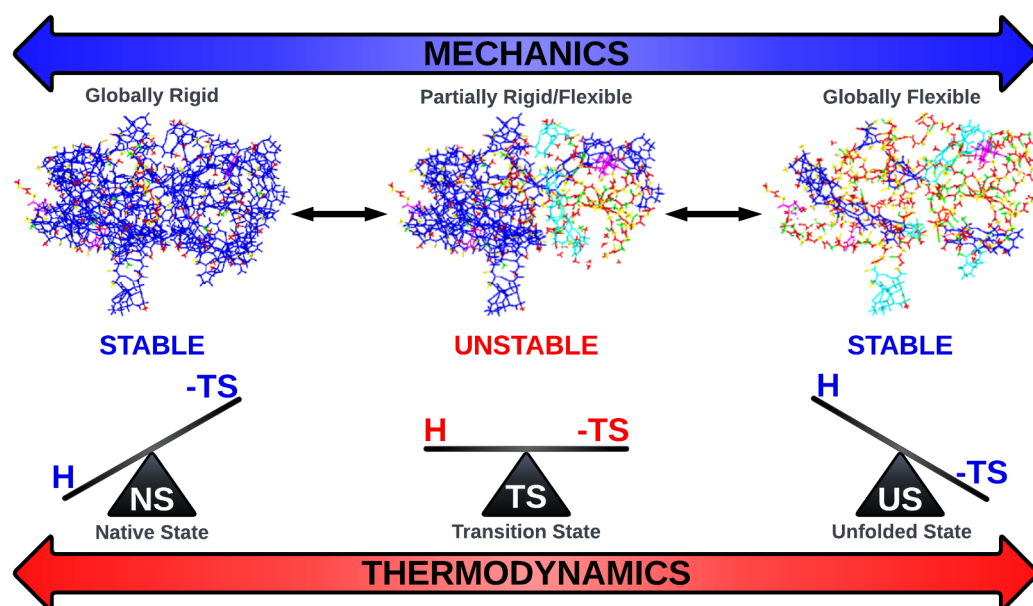


Figure 5. Protein stability paradigm for globular proteins. Mechanical and thermodynamic stability are intimately related. The native state of globular proteins is driven by favorable enthalpic interactions through the hydrogen bond network and packing interactions. The unfolded state is driven by conformational entropy, associated with an increase in conformational flexibility. The transition state represents a mixture of opposing thermodynamic and mechanical elements, which determines protein folding pathways.

Empirical characterization methods for protein structure and stability are biased one way or another. The biases originate from what is measured. For example, in crystallography, it is only possible to observe mechanically stable proteins, because they pack tightly when placed in an environment with significant concentration in the solution state. In contrast, NMR measures protein dynamics in solvent, and includes information about the solvent. For example, it is possible to observe wakes in solvent paramagnetic relaxation enhancement NMR [189]. The merging of this heterogeneous empirical data over the last half century has transformed fundamental physics concerns into data science. It is now important to develop methods and concepts that work with conformational ensembles through their probability distributions and describe how different environmental conditions modulate conformational ensembles, ushering in big data analysis.

Since the operative word in data science is “data”, it is worth describing where data for protein function come from. Databases such as SCOP and CATH have been valuable for learning how to perform secondary structure prediction and early homology modeling [9,190–192]. Historically, the accession of structural and sequence-based data spawned regression and statistical methods to obtain predictive models [191,193,194]. In a similar spirit, other empirically derived data yield metrics such as the gravity score among many other sequence propensity scales [195]. Notice that the depositories concern themselves with data that define a particular property that is not directly related to function. The protein function analysis part of the problem comes in because these data are clustered based on categorical labels that specify which protein is functional or not, making data mining plus clustering a standard bioinformatics approach. Although this data mining approach has been extraordinarily useful, methods that rely on data collections are subject to hypothesis formation bias [196].

An important class of proteins are Intrinsically Disordered Proteins (IDPs). An IDP has disordered regions, which can mix with ordered domains or small well-defined structural motifs. Often IDPs undergo induced conformation change upon binding [197]. Disorder-based functionality complements the known function of ordered proteins and domains [198]. The prominent characteristic of IDPs is that these biologically active proteins/domains do not coalesce into stable 3D structures under physiological conditions without some inter-molecular interaction. Naturally, structural data for IDPs or intrinsically disordered regions are lacking. As such, the contrast between globular proteins and IDPs highlights how training data can lead to hypothesis formation bias due to sampling bias inherent in a structural database. When ML methods are trained on structural and other empirical data, potential sources for bias hypothesis formation should be considered.

Working with conformational ensembles provides a direct route to elucidate protein stability, dynamics and function. However, there remain open questions about the best way to generate, represent and quantify conformational ensembles. Early work includes taking a statistical approach [199], knowledge-based analysis [10] and unsupervised clustering [200,201]. Eventually, neural networks were applied to recognize protein folds [202] or to self-improve a predicted folded structure [203]. Subsequently, predicting optimal sidechain packing was addressed in pytorch implementations [204,205]. Using specialized transformations, such as Voronoi tessellations, and CNN has been shown to yield state of the art fold modeling [206]. In addition, contact map predictions have been performed using various methods utilizing aggregated sequence and statistical data [207], which in part was leveraged by alphafold, where a DNN is applied to contact maps and can be leveraged to predict multiple conformations [84,208].

Predicting stable conformations using ML has been achieved using neural networks [209] by operating over local conformation changes generated through physics-based minimization. In the context of thermodynamic and kinetic stability, ML has added a new twist into protein fold prediction. To simplify the analysis as much as possible, a large body of work has focused on fast folders or ultra-fast folders [210] as model systems. Utilizing Bayesian decision trees in tandem with MD has been shown to properly identify folding

pathways [211]. To reduce the computational costs of protein folding MD simulations, ML-guided dynamic frameworks have been introduced with some success [212].

Typically, for initial descriptors, these models use a combination of structural features, sequence information using singleton or multiple sequence alignment, and structural topology representations such as contact or distance matrices. Feature crafting is a key element of well-designed learning algorithms. In addition to sequence propensity scores, structural features include atomic packing [213], electrostatics [214], meta dynamics from MD, and structural families, such as those found in SCOPe [10]. These features in combination with various software packages for parsing and curating structural elements with adaptable objects suitable for ML, such as pytorch objects, will likely pave the way for the next generation of structural bioinformatics [215].

There is a resurgent interest in predicting protein stability, mainly from sequence information, and possibly from static structures. For example, emergent sequence features from transformer processes with alternating self-attention with feed-forward neural network connections [216] have been applied to embed sequence spaces. Along these lines, with byte pair encoding compression and training on $\Delta\Delta G$ data, a mutational predictor for structural stability directly from a sequence was created [217]. Furthermore, a self-attention-based variant of a GAN has been applied to learn the sequence diversity to generate new functional protein sequences [218]. To obtain improved predictions for stability, solubility and binding affinity, recent works integrated 3D information into sequence embeddings using a message-passing neural network consisting of encoder-decoder layers trained on atom distances [219]. Possibly, self-learning will take into account thermodynamic attributes such as enthalpy and entropy, of stable folds of proteins, along with folding pathways and functional classifications in tandem [220].

In our view, a purely sequential approach will fail to give robust protein stability predictions. The environment dictates the nature of the conformational ensemble in conjunction with the constraints placed by a sequence. Without context, stability is undetermined. Specifically, solvent and conformational entropy effects cannot be neglected when quantifying protein stability. Therefore, the ML methods that incorporate environmental and conformational ensemble information will likely perform well in the long run. In the short term, good results may be obtained by working with small datasets that have systematic trends; however, experimental evidence in the late 1980s and early 1990s [35,36] suggest to us that the same mistakes will be repeated under the guise of non-linear black-box magic.

3.3.1. Role of Environment

Protein stability highly depends on environmental factors. Most work takes into account how protein stability can be altered with different formulations or how stability can be shifted by small molecular ligands. While numerous factors are present in the crowded cell, we limit our review to the description of protein stability as a function of the chemical composition of small solute molecules. This simpler situation is already a rich and important problem, germane to synthesis and purification of proteins in practice.

Practical computational approaches employ datasets that characterize protein sequence/structure characteristics and correlate this information to solubility using ML models. A graph-convolutional network trained on soluble protein datasets, such as eSOL, treats solubility prediction as a regression problem. By including contact maps along with a variety of sequence and structural information about the protein, predicted accuracy benchmarks had an R^2 of 0.48 and AUC above 0.8 [221].

In another line of research, improving solubility by modifying moieties through solubility tags is of great interest. An algorithm that creates these for peptides was found to increase solubility in empirical studies by over 100%. This work also trained on the eSOL database using support vector regression to inform a genetic algorithm to optimize the tag additions for the given sequence [222]. Similar work utilizing NLP techniques learned good soluble fragments from the TargetDB [223]. Another approach combined dilated

CNN layers with residual ANN and a Squeeze-and-Excitation layer in effort to extrapolate long-range relationships along a sequence to relate to measured solubility [224].

The ability to increase protein stability without loss in protein function efficacy is the objective of designing protein formulations. A key lacking aspect seems to be understanding this for a specified aqueous co-solutes, whether ionic or aliphatic in nature. One cannot separate the conformational ensemble of a protein in an aqueous solution without taking into account the exchange of protons from the protein to the medium. From explicit MD simulations of constant pH, training a convolutional neural network in tandem with dense layers for pKa prediction was shown to be fairly robust for soluble proteins by using MD simulations to train ML models [225]. A model for force computation called high-dimensional neural network potentials have been shown to handle atomic electrostatics with good accuracy with moderate computation times [226]. Going forward, it should be possible to include details of electrostatics in such an evaluation in order to create correlations between protonation states based on pKa in aqueous solutions to the electrostatic potential of the protein.

The environment of a protein need not be of aqueous nature. There has been recent work with ML to predict membrane interfaces of proteins using reinforcement learning on structural data generated from MD, after determining appropriate CVs. Utilizing an ensemble of predictors tied together with a meta classifier, exhaustively tested and optimized, yielded state-of-the-art accuracy under limited conditions [227]. In a similar fashion of training, MD-based training data to generate CVs that are tractable have also been used to understand aggregation of larger proteins, such as antibodies, where the environmental question is defined by the protein target of interest [228,229]. Post-translational modification (PTM) sites have also been tackled, such as glycosylation [230] using RF algorithms to predict sites from sequence input or phosphorylation sites in a similar fashion [231].

In each of the cases discussed above, a key dataset is used to train a model for inferring relationships between sequence/structural information and how the protein will react to the environment. In each of these applications, extrinsic information is being inferred on data from a secondary source. A key opportunity is to look for transferability in the ML model to account for changes in the environment.

3.3.2. Protein Engineering Through Mutagenesis

Mutating residues to modify protein function or achieve a specific molecular property is the earliest examples of protein design, also known as re-design and protein engineering. We are interested in the computational biology underpinnings of in silico structural based platforms that proceed along these lines [232–236]. Historically, plasmid insertions gave biochemists the ability to deduce functional sites and facilitate molecular engineering, and these procedures eventually led to modern adaptations [237,238]. Combinatorial approaches, threading along backbones and more detailed knowledge-based methods have dominated the field [69,191,239]. Fixed backbone designs lacked predictive power for binding events due to the necessity of understanding dynamics or flexibility of the protein [240]. Early attempts to solve this problem applied dimension reduction techniques or SVM [193,241].

To help quantify the shifting of stability in a protein, ProTherm was created as an empirical mutational database that includes the attributes of $\Delta\Delta G$ and change in melting temperature. Application of ML models to the curated ProTherm dataset generated mutational landscapes utilizing SVM and RF decision regression. These approaches obtained fair predictive power for the dataset, when compared to naive Bayes classifier, K nearest neighbor, partial least squares, and an ANN [242,243]. A deep neural network predictor of $\Delta\Delta G$ using this same training dataset outputs $\Delta\Delta G$ values for mutated sequences, as determined by multiple sequence alignments [244]. When this model was validated using the same data source, linear correlations in the range from 0.6 to 0.7 were obtained. However, when tested against novel sequences, the correlation ranged from 0.7 down to 0.1, with predictions sensitive to specific attributes of the protein function. Optimizing protein

interfaces through mutagenesis has been successfully achieved using a RF approach [245], which compared favorably to other mutagenesis predictors such as SKEMPI. Most results to date use training datasets that over-represent specific protein families, which leads to poor translation to other families. It has been noted that generalizing a mutational stability model must be undertaken with great care for each specific problem addressed [196].

Most recent advancements in predicting stability changes of a protein due to mutation include hallucinations via diffusion models applied to protein structures to generate novel composite single and multi-domain globular proteins [246]. Improving interactions and stability of structural space from their relationship to sequence space was deployed using a message-passing neural network with empirically deduced success [219]. Methods that minimize the number of mutations to shift stability are available [247]. In addition to addressing the solubility problem, TopologyNET a DCNN also predicts stability changes upon mutagenesis using a neural network design, demonstrating that the two problems are often entwined [248]. Pipelines are being developed that piece together ML packages with evaluation methods such as alphafold, Rosetta, or MD simulation evaluation to find possible realistic folding proteins [249].

In most works to date, the process of prediction followed by refinement using known features is applied. Prediction of stability shifts due to mutagenesis requires structural and thermodynamic considerations, which makes it challenging to achieve reliability.

3.3.3. Protein–Protein Interactions

Understanding molecular function extends past the intrinsic characteristics of individual structures/sequences and involves the dynamics of protein–protein interactions (PPIs). These interactions can occur in many ways, such as permanent (i.e., irreversible binding) or transient relationships (i.e., intracellular signaling interactions) [250]. Predicting if and how proteins will interact is an ongoing computational/experimental challenge that is driven by a large number of confounding factors. *In vivo*, a protomer's localization, concentration, and local environment can affect the interactions between proteins, which are vital to control the composition and oligomeric state of protein complexes [13]. The majority of PPIs cannot be neatly categorized into a dichotomy of obligate or non-obligate interactions; there exists a gradient between the two. Furthermore, the stability of complexes is heavily dependent on both the physiological conditions and environment [13,17]. The ML methods aimed at predicting PPIs can be broadly categorized into two forms: structure and sequence-based approaches; additionally, there are various objectives for PPI prediction. These objectives include: the binary task as to whether two proteins generally interact (class 1) or not (class 2); PPI site prediction; and PPI binding affinity prediction. In this section, the focus will be on the general problem posed as to whether two given structures interact or not.

Sequence-based PPI prediction methods generally begin with two protein sequences and result in a score which ranks the probability that an interaction occurs. Encoding the complex information required for robust PPI predictions is of vital importance while performing feature engineering [251]. Furthermore, an issue arises due to the variable length of proteins under PPI analysis. Many ML methods require input feature vectors of equal length; this often involves the padding/truncation/aggregation of sequences [252], where the inevitable loss or distortion of information will occur. A non-exhaustive list of ML methods utilized for sequence-based PPI prediction is presented in Table 3. Structure-based PPI predictors are not constrained by the same problems as sequence-based approaches, such as the loss of important predictive information during feature construction. A list of structure-based PPI prediction models from 2018 is also presented in Table 3.

The majority of ML methods for PPI prediction are sequence based (static structures), and do not account for conformational ensembles or the role played in PPIs by local environment. Despite the use of high-throughput techniques in discovering PPIs, the coverage of experimentally determined PPI data remains poor [253]. This translates to a lack of labeled data for ML model training, which directly affects the generalizability of predictions on

previously unseen systems. Given the delicate nature of constructing the correct features for sequence-based PPI prediction, it is a natural extension to utilize neural networks that have the capability of identifying useful latent features simultaneously during model training. This trend can be observed from Table 3 as the successful PPI prediction ML models have largely shifted into deep learning architectures.

Table 3. Machine learning methods for PPI prediction from 2018 to 2022.

Paradigm	Method	Year	Model ¹	Stand Alone ²	Webserver ³
Sequence	DPPI [254]	2018	CNN	Yes	No
	EnsDNN [255]	2019	DNN	Yes	No
	MDPN [256]	2019	DPN	No	No
	CNN-FSRF [257]	2019	CNN + RF	No	No
	S-VGAE [258]	2020	GCNN + VAE	Yes	No
	EnAmDNN [259]	2020	DNN + Att	Yes	No
	PCPIP [260]	2021	SVM	No	Yes
	CAMP [261]	2021	CNN	Yes	No
	Balogh et al. [262]	2022	GAN	Yes	No
	TAGPPI [263]	2022	GCNN	Yes	No
Structure	Daberdaku et al. [264]	2018	SVM	Yes	No
	BIPSPI-structure [265]	2018	DT	Yes	Yes
	IntPred [266]	2020	RF	No	No
	MaSIF [267]	2021	ANN	Yes	No
	GraphPPIS [268]	2021	GCNN	Yes	Yes

¹ The nomenclature for ML models utilized: “SVM” (support vector machine); “CNN” (convolutional neural network); “DNN” (deep neural network); “MDPN” (multimodal deep polynomial network); “RF” (random forest); “GCNN” (graph convolutional neural network); “VAE” (variational autoencoder); “Att” (attention mechanism); “SVM” (support vector machine); “GAN” (generative adversarial network); “GNN” (graph neural network); “DT” (decision trees); “ANN” (artificial neural network). ² This column indicates whether a standalone program is available (“Yes” or “No”). The hyperlink for “Yes” redirects to the corresponding repository. ³ This column indicates whether a webserver is available to users (“Yes” or “No”). The hyperlink for “Yes” redirects to the corresponding webserver URL.

3.3.4. Role of Rigidity in Disordered Proteins

Beyond the classical structure–function paradigm, which is grounded in a general lock-and-key model driven by the assumption of unique structures, is a continued interest in the role of IDPs and protein hybrids. Protein hybrids are comprised of intrinsically disordered protein regions (IDPRs) and ordered domains. Disorder-based functionality complements the known function of these ordered proteins and domains [198]. The prominent characteristic of IDPs and IDPRs is that these biologically active proteins/domains do not coalesce into stable 3D structures under physiological conditions. Consequently, they lack the structural rigidity that is often cited as necessary for function in the structure–function paradigm [269,270]. From a physics perspective, an important characteristic of IDP/IDPRs is that they exhibit a flat Gibbs free-energy landscape, not favoring one molecular conformation over another [271]. The IDP/IDPR systems exist as highly dynamic structural ensembles at the secondary and/or tertiary levels [272,273]. Understanding the conformational state space that is spanned by disordered structures is vital for a robust structure–function paradigm that will allow for high levels of recognition specificity and provide further insight into how IDPs/IDPRs interact with other proteins and molecules.

Of particular interest are the IDPRs known as molecular recognition features (MoRFs), which execute their function through a phenomenon known as disorder-to-order transitions (induced folding) [197]. While the vast majority of an IDP is flexible, these MoRFs form

relatively stable structures as a rigid structural motif. Machine learning predictions of IDPs/IDPRs are especially important, as they contribute to the experimental discovery of PPIs. Previous studies have compiled/analyzed intrinsic disorder predictors [274,275]; here, a select list of predictors is provided in Table 4, which was constrained to ML methods since 2016.

The gold standard for IDP/IDPR predictor evaluation is the critical assessment of protein intrinsic disorder prediction (CAID). This is a community-based blind experiment aimed at identifying state-of-the-art IDP/IDPR predictors. In the 2021 CAID benchmark [276], 43 methods were evaluated while using a dataset of 646 proteins curated by DisProt [277]. To evaluate the methods across different applications, three forms of the base dataset were utilized: (1) fully disordered proteins (IDPs) from Disprot; (2) fully disordered proteins (IDPs) from PDB; and (3) a binding challenge dataset where positive labels correspond to residues annotated as intrinsically disordered binding residues (IDBRs) in the DisProt database. It should be noted that the threerd dataset contained 414/646 target IDPs that were absent of any positive labels that were derived from the experiment. Two overarching challenges were set; the first of these assessed performance of the predictors when the objective was the identification/prediction of fully disordered proteins (IDPs) using datasets 1 and 2. The second challenge evaluated the performance of the predictors when the objective focused on the prediction of disordered protein binding regions (DPBRs), benchmarked using dataset 3.

Table 4. Machine learning methods for IDP/IDPR prediction from 2016 to 2022.

Method	Year	Model ¹	Stand Alone ²	Webserver ³
MoRFchibi [278]	2016	SVM	Yes	Yes
AUCpreD [279]	2016	CNF	Yes	Yes
Predict-MoRFs [280]	2016	SVM	Yes	No
SPOT-Disorder1 [281]	2017	RNN	Yes	Yes
MoRFPred-plus [282]	2018	SVM	Yes	No
OPAL+ [283]	2018	SVM	Yes	Yes
rawMSA [284]	2019	CNN + RNN	Yes	No
SPOT-Disorder2 [285]	2019	CNN + RNN	Yes	Yes
ODiNPred [286]	2020	SNN	No	Yes
IDP-Seq2Seq [287]	2020	RNN + Att	No	Yes
fIDPnn [288]	2021	DNN+RF	Yes	Yes
fIDPlr [288]	2021	RF	No	No
RFPR-IDP [289]	2021	CNN + RNN	No	Yes
Metapredict [290]	2021	RNN	Yes	No
DeepDISOBind [291]	2022	MTNN	No	Yes
MoRF-FUNCpred [292]	2022	BR + SVM + RF	Yes	No
DisoMine [293]	2022	RNN	No	Yes

¹ The nomenclature for ML models utilized: “SVM” (support vector machine); “CNF” (convolutional neural fields); “RNN” (recurrent neural network); “CNN” (convolutional neural network); “SNN” (shallow neural network); “ANN” (artificial neural network); “RF” (random forest); “MTNN” (multi-task neural network); “BR” (binary relevance). ² This column indicates whether a standalone program is available (“Yes” or “No”). The hyperlink when “Yes” redirects to the corresponding repository. ³ This column indicates whether a webserver is available to users (“Yes” or “No”). The hyperlink for “Yes” redirects to the corresponding webserver URL.

Here, we provide a brief summary for the performance of the participants and discuss general trends. The primary metric for performance during CAID is the maximum F_1 score, F_{max} , or the maximum harmonic mean between precision and recall over all thresholds. This

is a robust metric for two reasons: (i) F_{max} takes into account all predictions across the full sensitivity spectrum; and (ii) F_{max} is invariant to imbalanced datasets. The largest general trend from the assessment was that the best methods utilize deep learning neural network architectures which outperform physiochemical-based methods. For the first challenge over all performance metrics (beyond F_{max}), the predictors fDPnn, SPOT-Disorder2, rawMSA, and AUCpreD consistently perform in the top five, respectively, attaining an F_{max} of 0.48, 0.47, 0.45, and 0.44 on dataset 1. For dataset 2 where all known structured regions are filtered out, these numbers substantially increase for the same methods to an F_{max} : 0.71 (fDPnn), 0.79 (SPOT-Disorder2), 0.75 (RAWmsa), and 0.77 (AUCpreD). This exemplifies that the removal of known structured regions heavily simplifies the prediction problem. For the second challenge (DPBR prediction), the assessment across all methods shows a substantial reduction in predictive performance, indicating a need for advancement. The F_{max} for the top 5 methods here was 0.214 ± 0.0134 .

Given the recent non-incremental increase in the structure prediction field by AlphaFold2, it is pertinent to address recent concerns relating to AlphaFold2 and intrinsic disorder prediction. It has been noted that the disorder predictor one constructs from a predicted AlphaFold2 structure determines accuracy [294]. This is due to non-trivial underlying assumptions such as annotating residues from helices, strands, and H-bond stabilized turns as ordered while the remaining are unordered. This ultimately leads to a dramatic overestimation of disorder [274,275]. We suspect that structure prediction methods will soon embrace quantifying conformational ensemble diversity to allow for more representative predictions for disordered regions. However, this remains an open problem.

3.4. Protein Dynamics

Protein motion, of which there are many types, as schematically shown in Figure 6, is critical to protein function. Identifying flexible regions in proteins can be empirically achieved in several ways. The two most common methods are XRC and NMR. In XRC, the temperature factor (or B-factor) estimates the fluctuations of atoms due to thermal vibrations in their equilibrium positions from the attenuation of scattering [295]. The B-Factor has been moderately found to correlate with structural flexibility, particularly when using just carbon alpha B-factors [296], but it is not a direct measurement of flexibility.

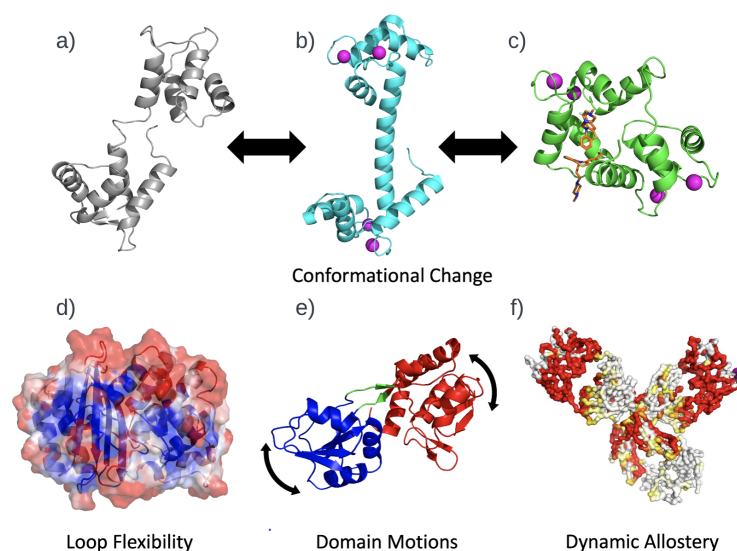


Figure 6. Examples of protein functional dynamics. In (a–c), three critical conformational states [297–299] of calmodulin are shown during the process of ligand binding. The unbound structure (a) binds to calcium ions (b), then the resulting structure is able to bind with a substrate (c) [300]. On the bottom row, (d,e) show the native state motions of proteins including surface loop flexibility [301,302] and concerted domain fluctuations [303]. (f) A visual of dynamic allostery pathways resulting from changes to the vibrational modes of a protein upon binding a ligand [304].

This is because XRC only provides static structures. Crystal packing biases the B-factors during the process of solving the inverse problem for structure from the raw X-ray diffraction data, as resolution errors are folded into the B-factors [305]. In the solution, NMR works with conformational ensembles, which allows flexibility to be sampled. For example, the root mean squared fluctuation (RMSF) is moderately correlated with the S^2 order parameter [306]. There are other modes of NMR operation, and in general, NMR measures dynamics on a limited range of timescales.

In addition to these incompleteness and indirect measurement problems, directly resolving dynamical correlations among flexible regions in a protein remains an experimental challenge. Therefore, MD simulations are typically used to identify flexible regions within a protein and monitor time correlations. However, MD simulations cannot reach long time scales, and there are insufficient statistics at the longest timescales that can be reached. Furthermore, it is a challenge to identify functionally relevant dynamics both experimentally and computationally. This is because the observation of atomic motions does not answer how important these motions are for a mechanism of action. In general, non-functional motions will hide functionally relevant small-amplitude motions, especially when non-functional motions have large amplitude. Therefore, functional dynamics are likely to be missed by methods that intrinsically extract large structural fluctuations in a protein, such as PCA. Since PCA is used as a dimension reduction method, it is possible to lose important information about functional dynamics before more sophisticated ML methods are applied whenever PCA is indiscriminately applied.

3.4.1. Protein Flexibility and Conformational Dynamics

For brevity, we only discuss globular proteins and IDP as two classes of proteins. Globular proteins tend to hold a well-conserved 3D shape, although parts of the structure can flex. The most flexible parts of a globular protein are located in its loop regions. In general, large-scale motions and localized flexible regions are important for protein function. Flexibility in a particular macrostate (e.g., the native state or a metastable macrostate) exists as a hierarchy in the size of certain structural elements and in terms of local minimums in the free-energy funnel. As an example of this, a full domain of a protein may undergo a concerted, global conformational change, while regions of secondary structure remain more rigid due to stabilizing hydrogen bonding. Loops, particularly those protruding into the solvent can be quite flexible in contrast with the the stable core of a molecule [39,307] Finally, at the highest resolution, residue side chains can be flexible in that they switch between different rotamer states. Flexibility at each of these hierarchical levels can help the protein perform a specific function. When comparing two or more proteins in terms of flexibility, root mean square deviation (RMSD) and root mean square fluctuation (RMSF) provide simple metrics that inform differences in global and per-residue flexibility given a ensemble of structures from, e.g., MD simulation. Additional measures are the flexibility index, which quantifies how flexibility is distributed throughout the protein, which is directly dependent on thermodynamic stability [39,308].

One approach for ML to directly predict flexibility in proteins is to use the structure data collected in the PDB to predict the experimental flexibility metrics, such as the B-factor and S^2 order parameter. The Gaussian Network Model (GNM) and the more general Elastic Network Model (ENM) [309] allow RMSF to be calculated, and this quantity moderately correlates with B-factors and S^2 order parameters [39]. Another metric that can be learned by training these networks is the Flexibility–Rigidity Index (FRI) [310,311] which can be computed for each atom (graph node) as a weighted sum of its connections. The sum of all FRI node values provides a global FRI value, which is inversely proportional to flexibility. FRI can be used as a predictor of B-factors and general stability in proteins. More recent work [312] has pitted these network models directly against ML algorithms (RF, gradient boosting trees, and CNNs) and found that CNNs provided the best prediction. Similar work to predict S^2 has used shallow neural networks [313] to predict the parameter directly from feature embeddings of 3D structures. Some more recent work [314] has been

carried out using ML to validate structure distributions from CryoEM, a method gaining considerable traction in structure determination. The method uses a Gaussian Mixture Model, inspired by the structure of VAEs to approximate the electron density map, with the limits of the density corresponding to the probable area for an atom to exist.

In the literature, there are many types of ENMs [309]. In general, an ENM makes a direct assumption that the ensemble of conformations can be described as quasi-harmonic vibrations with respect to a rigid reference structure. This is modeled by sampling the bottom of a harmonic potential well, and modeling the protein as a set of masses connected by springs. The equation of motion for n masses is $M\ddot{\vec{x}} = -K\vec{x}$. In this model, \vec{x} and $\ddot{\vec{x}}$ are the positions and acceleration of each atom or mass point, M is a diagonal matrix containing the mass of each mass point, and the elements of matrix K , k_{ij} are the effective spring constant connecting points i and j . K is called the Hessian matrix for the model, where harmonic potentials connect to mass points. The eigenvectors represent the normal modes of motion for the system, and their eigenvalues are the corresponding frequency of that motion. The mass of each mass point will in general be different because the constituent atoms associated with each mass point are heterogeneous. However, often the masses of all mass points are set equal. These spring connections are heuristic, and the ENM is not the same as a true normal mode analysis since the ENM is set at a coarse grained level description.

In principle, equilibrium MD simulation provides the best way to investigate flexibility by directly sampling the conformational space. However, as discussed above, in practice there is usually an unobtainable computational cost, and to make matters worse, one faces massive amounts of data, which must be analyzed to quantify flexibility through dimension reduction. Nevertheless, the most popular method for extracting flexible motions from equilibrium simulations is Essential Dynamics (ED). Essential Dynamics [128] applies PCA to the ensemble of mean centered conformations by diagonalizing the covariance matrix. The eigenvectors and eigenvalues represent global motions, which can be ranked by size scale using the eigenvalues. The eigenvalues are the variance in the global motion represented by the eigenvector, where larger eigenvalues represent greater amplitude vibrations. ED can be performed on the Cartesian coordinates of an ensemble of structures, or internal coordinates such as distance pairs or dihedral angles.

A link was found between ED and ENM, where the Hessian matrix can be approximated for a sampling of conformations by the inverse of their covariance matrix, which is the matrix used in PCA. Thus, within the quasi-Harmonic approximation, the most variant PCA mode (with the largest eigenvalue) is equivalent to the lowest-frequency ENM mode (with the lowest eigenvalue) [34]. It is important to note that the limitation of both approaches is that they require the conformational ensemble to remain close to a single representative structure. This means jumping between different free-energy basins cannot be accurately described. However, at the coarse-grained level, jumping between basins is modeled as harmonic on long-time scales. In this case, these methods describe large-scale rearrangement rather than intrinsic flexibility. Depending on the application, this can be seen as a limitation or benefit of using these methods.

Some ML methods have been used to predict flexible and dynamic regions in proteins. Much of this research is driven by better docking algorithms that can accommodate flexibility in the receptor and ligand. For example, SVMs have been used as binary classification algorithms to determine whether loops have high mobility or not. Early classification algorithms often use curated features such as solvent-accessible surface areas, B-factors, and other structural quantities, which improve but likely bias predictions. Informative features for describing protein flexibility are required to elucidate which atomic motions correlate with functional characteristics. Random forest is commonly used for classification and regression tasks, but metrics such as Gini impurity [315] allow the model to understand feature importance, which provides context into how the model relates its final prediction to the input features. Exemplifying how Gini impurity can provide insight to how RF models make predictions, RF models were trained to classify conformations in beta

lactamase at various stages of ligand catalysis, and using the Gini index revealed residue pairs and structural regions most important for the classification [111]. The success of using dynamics-based models to inform predictions relating to protein function reveals the deep connection between dynamics and function. ANNs have also been used to identify mobile regions of proteins. One such method to identify flexible residues in proteins is NEAT-FLEX [316]. Importantly, neural evolution is applied using a genetic algorithm to augment typologies for learning the optimal topology in the neural network to predict molecular properties.

Connecting conformational dynamics to function can be built into models that directly work with trajectories that describe protein dynamics. The tICA method, which we discussed in Section 3.2.2, does this by using time lag to let the model focus on dynamics at a particular time scale. This approach has also been applied to t-Distributed Stochastic Neighbor Embedding (t-SNE) [317], a non-linear dimensionality reduction method for efficiently reducing MD trajectories into a low dimensional space. A different approach from the Jacobs lab is to apply discriminant analysis to extract functional dynamics using a Supervised Projection Learning for Orthogonal Completeness (SPLOC) algorithm [318]. This method uses data-driven optimization to learn spatial-scale and temporal-scale independent dynamic features, which best distinguish two classes of molecules which function differently. The target for learning is fully defined by the data presented, rather than underlying assumptions [319] such as variance in PCA. The method was demonstrated by describing the dynamic differences in multiple TEM beta-lactamase, which explain differences in enzyme efficiency among several ligands [302].

Generative models provide an interesting approach to studying the flexibility of proteins, which requires training on known conformational ensembles. This data can be obtained from MD simulation trajectories. The MD data provides the model with a baseline for how the atoms move. Then the model learns low dimensional distributions that are used to generate more conformations to help fill out the low sampled regions. This provides a means for studying molecular motions in the native state. A simple, but effective method for doing this is to determine the normal modes of a protein with an ENM, and then use the modes themselves as displacement vectors for perturbing the structure [320]. This strategy was implemented for an application to investigate the opening of cryptic pockets in various proteins.

Autoencoders have also been used to generate realistic conformational ensembles [321]. A disadvantage of using traditional autoencoders is that the information is not efficiently and continuously mapped to the latent space which prohibits effective sampling [322]. Again, turning to VAE allows the model to learn continuous dynamic processes such as folding or conformation change [212]. These latent spaces can be used to study the dynamics of proteins in great detail, as shown in this study [323] that used them to learn important conformational states of GCase. Another generative method, similar to an autoencoder, but which learns the distributions in a Gaussian kernel space, is the Flexible Backbone Learning by Gaussian Process [324] (Flex-BaL-GP), which uses redundant entries in the PDB to describe alternate backbone conformations [325]. The database Conformation Diversity in the Native State (CoDNaS) is a resource for training ML algorithms to detect flexible regions in proteins.

In the early days of structural bioinformatics and computational biology applied to understanding the structure and dynamics of proteins, methods were developed to predict the most flexible regions in a protein. As reviewed above, this is not a difficult task, as there are many methods (based on ML and/or physics-based models) available to identify flexibility or mobility. It is seen that a wide range of methods yield results that are consistent with experiments that measure mobility. The more interesting question that goes to the heart of protein function is how flexible regions and their motions are correlated. The construction of conformational ensembles using MD simulation can, in principle, uncover this correlation, but to reach the timescale of functional dynamics could require calculation times that are not feasible. Our view is that physics-based modeling will

be the rate-limiting step in characterizing functional dynamics, and successful models must include the effects of thermodynamic and mechanical stability and their interrelationships.

3.4.2. Dynamic Allostery

Allosteric signaling is a long-range communication present in proteins that affects function, conformation stability, or interaction propensities between partner molecules [326]. Note that long range refers to distance in a 3D structure. There are many types of allostery, and there are precise definitions of the phenomena in terms of binding curve shifts [327]. A classic example of allostery is exhibited in the function of hemoglobin, where binding one diatomic oxygen ligand acts as a homotropic allosteric regulator, resulting in a lowered barrier for the binding of subsequent diatomic oxygen ligands [328,329]. Most models of allostery, including obligate and conformational allostery, are based on identifying correlated motions in sets of conformational ensembles. This review is concerned only with dynamic allostery because it has a universal mechanism that is potentially present in all proteins [330,331]. This universal character occurs because in dynamic allostery, the mechanism is driven by modes of vibration in the protein, or alternatively, how rigidity propagates through a protein structure [332].

Several normal-mode-based allosteric site prediction software exist [333–337]. There has been a wealth of studies that benchmark the accuracy of these models/software. It has been shown that dynamic allostery will be present for structured proteins with flexibility in limited regions [338]. Off target effectors create coupled dynamics through vibrations, which potentially alter preferential binding propensity [339]. In addition, if a protein is mutated, its sensitivity in vibrational characteristics is likely to change. As such, like other functional mechanisms, dynamic allostery is affected strongly by evolution as noted in beta-lactamase [302]. It is computationally feasible for methods detecting dynamic allostery to be extended to design proteins with planned allostery communication signals.

Attempts at classifying allosteric signaling using local features generated from MD have been attempted. Results using naive Bayesian inference and SVM yielded poor allosteric predictions [340]. Recently, ML applied to MD for allosteric signaling has been performed, with a larger emphasis on the MD than ML in most applications. This is because there is a need to obtain well-sampled data to characterize the ensemble that describes the phenomenon. Good accuracy was reported in one work using SVM and random forest on MD-docking data with known categorical values for drugs [341]. Deep neural networks have been applied to short MD trajectories using a specialized metadynamics routine, called neural relational inference MD. In this model, an allosteric signal is put through a VAE to interpolate conformations and determine the communication pathways within the protein [342]. The combination of extreme gradient boosting (XGBoost) and GCNNs for allosteric site prediction was found to predict conformational allosteric sites on static structures tested, without a need for heavy simulation computations after being trained on known allosteric proteins to learn topological connections that define the phenomenon [343].

Some limited work involving engineering allosteric sites into existing proteins has been carried out previously [344]. These efforts typically add an allosteric site into an existing functional protein or one adjoining an entire domain [345]. Full de novo allosteric proteins are still within the domain of classic design-and-check methods requiring expert attention at all steps [346]. Despite the success of some ML methods that rely on a single static protein structure, it has been our experience [16,34] that single mutations can often dramatically shift the communication pathways, yet the topological properties of the structure remain largely invariant. Accounting for fluctuations through rigidity networks or vibrational modes, or ensemble of conformations through sampling is likely required for robust results.

3.4.3. Potential Energy and Force Field Calculations

To perform MD simulations, there must be a molecular forcefield, which must be parameterized. The most accurate method is using *ab initio* quantum calculations to determine the exact Hamiltonian of the system. Unfortunately, this direct approach becomes intractable for systems larger than several hundred atoms. In addition, the calculations are difficult in principle because environmental effects (from more atoms) are difficult to model implicitly, leading to a non-local parameterization that does not generalize to all systems. Alternatively, a functional form can be approximated based on modeling physical interactions at a local level (bonds, dihedrals, van der Waals, etc.) plus non-local electrostatics. These models can be parameterized based on fitting the functions to known experimental data. While this is more practical, it is limited by the experimental data available, which often does not generalize to systems and environments much beyond the realm described by the fitting data. Limitations from systematic experimental data availability will be present in all ML methods too, but ML may help the procedures to obtain better models for the data that are available.

In recent reviews of the subject [347,348], requirements for ML-based forcefield models are laid out. The main three requirements for the outputs of the model are: (1) conservative forces, which are (2) rotationally and translationally invariant, and (3) the interactions between particles respect symmetries regarding indistinguishable, identical atoms. Usually the nuclear charge allows the model to be aware of atom types. Most methods choose to predict potential energy functions instead of directly predicting forces. This is because potential energy functions are smoother and it is straightforward to differentiate potential energy functions to compute the forces from the model. Neural networks with rigid input structures are sensitive to not only these requirements, but also systems with variable numbers of atoms. Common approaches to this use permutations to match the input order to a universal form and transform the input system to faithfully provide the required invariance properties using transformations such as coulomb matrices or symmetry functions. Alternatively, many models use distance pairs as inputs to a model because they are intrinsically rotationally and translationally invariant. However, even in this case, additional transformations and features are needed to ensure that they are invariant upon atom permutation. The most problematic aspect of this is that this approach does not scale, and thus is not transferable, because the number of distance pairs makes the calculations intractable for large systems.

Ideally a molecular representation models an atom with its local environment, which influences the forces acting on the atom. To achieve this, as well as the other requirements for representations, Atom Centered Symmetry Functions (ACSFs) [349] were introduced in 2011. These functions had two parts: a cutoff function that identifies atoms in the proximity of the atom of interest, weighting contributions to the overall symmetry function, and a symmetry function which describes the local environment. Each ACSF is a sum over all distance pairs within the radius cutoff; hence, it is rotationally and translationally invariant and invariant upon exchange of atoms. ACSFs proposed in the original paper described the radial, spatial and frequency distribution of atoms in the local environment of the atom of interest but did not take into account the atom species. In later work, a set of weighted ACSFs was developed [350] that weighted each symmetry function contribution to the ACSF by the chemical species.

Several methods of transforming molecular representations for predicting forces with ML have been reviewed [351]. One of these is the Smooth Overlap of Atomic Positions (SOAP), which is a method for representing local atomic geometries as a continuous density for use in ML potentials. SOAP appears to be a superposition pairwise distance of atoms within each region, expanded on radial basis functions and spherical harmonics, then kernelized to describe similarity between environments. Another alternative to finding an invariant representation for the molecular system is to find the optimal aligned frame of reference to compute forces in. Unfortunately, this is not efficient for simulation applications, as forces will continually have to be transformed to the correct frame of reference [351].

The Born–Oppenheimer approximation allows the nuclear and electronic wave functions to be treated independently, where the nuclear degrees of freedom are treated classically. Because classical MD simulation does not simulate electron dynamics, a forcefield need not account for this detail. One approach developed in 2010 [352] is the Gaussian Approximation Potential (GAP). GAP first represents the local atomic environment in a set of Gaussian radial basis functions, and learns regression that predicts the energy for the atom using kernelized features. The Gaussian basis has the benefits of being intrinsically rotationally and translationally invariant, and is easily differentiated in terms of the atomic positions to provide forces [352]. In addition, GAP has been implemented in the QUIP software [353] for MD, which can be plugged into popular MD software such as LAMMPS. GAP appears to be fast and accurate for predicting configurational energy; however, it must be trained using *ab initio* calculations, which limits its generality.

Early work using ML to parameterize forcefields to simulate the dynamics of molecules involved using neural nets to predict parameterization for potential energy surfaces. Refs. [354–357] Neural Networks as potential energy surface approximators were explored further in 2004 [358]. Their models use a neural network to represent a system and output its total energy. These networks represented atomic configurations with symmetry functions, which drastically reduced the input size and allowed the networks to be trained using a reasonable number of data points. Behler and Parinello generalized this in 2007 [359] in their high-dimensional neural network (HDNN) model. The advantage of this model, which allowed it to generalize to higher dimensions, is that it broke the total energy calculation into the sum of energy contributions from each atom. The energy per atom was then computed from a set of subnetworks, one subnetwork representing each type of atom in the system. A detailed review and subsequent developments involving HDNNs can be found at [360].

More recent ML models have used the many advanced architectures and methods developed in the past few decades. PhysNet [361] is an example of a DNN architecture proposed for computing energy, force and dipole moments. CNNs have shown great promise in helping us understand the complex interactions and dynamics of proteins; however, the discrete, grid-like structure of convolution layers made them an unattractive solution for machine-learning-based forcefields. To accommodate the continuous nature of atomic coordinates within the CNN architecture, Schütt et al developed a continuous-filter convolution layer [362] (cf-Convolution) to transform atomic representations into feature representations more salient for energy prediction. The continuous-filter convolution takes in the feature value at layer l , x_i^l and the 3D coordinates of each feature's associated atom, r_i and transforms the features a new representation in layer $l + 1$.

$$x_i^{l+1} = \sum_j x_j^l \cdot W^l ||r_i - r_j|| \quad (1)$$

The matrix W transforms the pairwise distances to the feature space dimensions. The model was implemented in a new network structure called SchNet [363] which has been implemented in a PyTorch toolkit called SchNetPack [364]. Forces on each atom can be extracted by following Newton's laws and differentiating the predicted energy with respect to the atomic coordinates. The full SchNet model is similar to a fully connected GNN in that each atom is represented as a node connected by pairwise edges (displacements) and that the node representation is embedded into a d -dimensional feature space which is updated with each sequential cf-convolution layer added to the model. The difference is that convolutions, rather than message passing, are performed to update the feature representation. Recent trends have seen use of GNNs for molecular fingerprinting extensively explored for predicting molecular properties such as energy and force. Some examples of method that use GNNs for predicting energies and forces include DimeNet [365], GNNFF [366], and NewtonNet [367]. These methods have shown increased force prediction accuracy over other deep learning approaches such as PhysNet and SchNet.

There are several implementations of deep ML potentials that can be used in MD simulation software instead of classical potentials. TorchMD is an MD engine (downloadable via

github) written in pytorch that uses combined classical and deep learning potentials [368]. For TorchMD's Deep Learning potential, SchNets were used. DeePMD-kit [369] is a method for parameterizing deep ML potentials, developed and distributed by deeppmind. This can be installed via conda, and the resulting trained potential can be used in the LAMMPS and GROMACS simulations software.

3.5. Drug Design

One of the strongest drivers of ML method development for computational biology and protein design is drug discovery. The drug discovery process takes an average of 12 years from start to commercialization, with an average cost of USD 1.8 billion [370]; consequently, this necessitates a need to streamline the drug development process. To this end, ML approaches have been widely applied in the field of biomedicine. Typically, ML methods use pattern recognition algorithms to discern mathematical relationships between empirical observations, such as protein secondary structure prediction, drug repositioning, and drug design [371]. In recent years, deep learning has garnered considerable interest from computational chemists and medicinal chemists. Until now, various reviews related to the applications of ML or deep learning in drug design and discovery have been published [372–380]. Here, we focus on two vital computational components of the drug design/discovery process, molecular docking and binding affinity prediction, as highlighted in Figure 7.

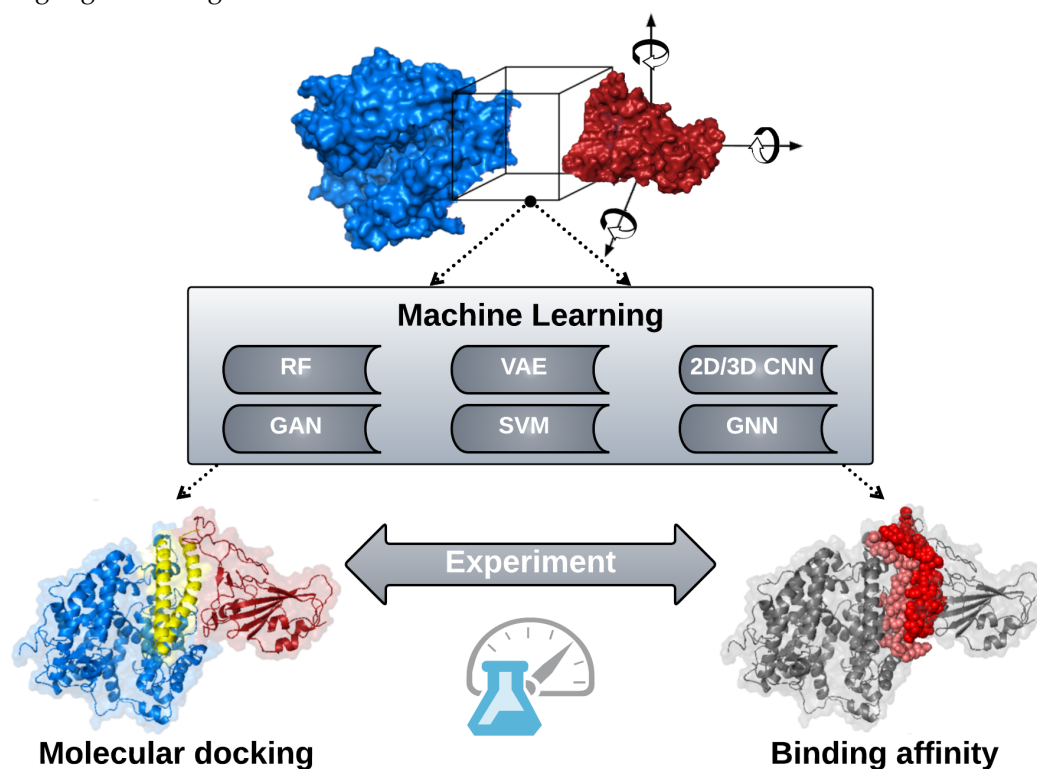


Figure 7. Computational drug discovery relies on the interplay between molecular docking and binding affinity prediction, both of which have been enhanced by ML. Docking methods can use ML to account for subtleties in molecular interaction such as flexibility or predict inter-molecular contacts, while ML-powered binding affinity functions score poses. In both cases, experimental data are used to train models, and methods will continue to improve as more data are collected.

3.5.1. Molecular Docking

Molecular docking is a vital step in the drug design process, and acts as a filter to identify candidate molecules for the purpose of reducing costs by streamlining experimental work. Virtual screening often refers to a high-throughput candidate selection procedure [381]. The pairwise objective with molecular docking can include: protein–

peptide, protein–DNA, protein–RNA, and protein–ligand. Depending on the method, 3D structures, physiochemical properties or a combination of the two are utilized to predict binding characteristics [382]. Molecular docking is a combination of two processes. The first is sampling, which involves generating a set of conformations from a semi-rigid 3D ligand. The method is evaluated based on its capacity to explore the conformational space of the ligand (target), gathering all theoretically possible conformations [383]. The second step is scoring, which approximates the binding affinity of each protein–ligand complex formed (called a pose). Other methods use scaffolds, or molecular fields that represent molecules to make the coarse-grained exploration of conformational space more computationally efficient [384]. Molecular docking can not only predict the binding conformation of a ligand in a target binding pocket, but can also estimate the binding affinity of ligand–target complex [371]; the latter will be expanded upon in the following section.

Classical docking scoring functions assume an additive functional form in order to model a linear relationship between the binding affinity and features characterizing protein–ligand complexes [385]. Unfortunately, a linear relationship is not guaranteed to exist; furthermore, if a non-linear method is used to fit this relationship, the developed scoring function could attain better performance. Traditional ML methods such as random forest [386], SVM [260,387] and neural networks [388,389] have been utilized for molecular docking scoring in lieu of regression-based approaches. Recent algorithmic advances in deep learning have driven promising molecular docking approaches. These deep learning docking methods come in many forms, such as: CNN [388,390–392], GNN [393–396], and generative models (GANs/VAEs) [321,397–399].

We identify three considerations that should be taken into account which ground many of the remaining challenges in molecular docking. First, not all the complexes in the PDB are functional, and this consideration should be reflected when determining protein sets for training ML models [400]. Furthermore, when developing ML models for molecular docking, it is important to train and validate the models over established data sets instead of using synthetic or augmented data sets. This guarantees representativeness, exhaustiveness, variety for the training set, and allows for inter-method comparisons of objective criteria [383]. Second, a general trend is that on average, the size of the interface measured as solvent accessible area buried upon complex formation is larger in biological interfaces compared to crystallographic structures [401]. Lastly, and possibly most important, the three-dimensional structure used for molecular docking will be out of its original environment, often resulting in a change in conformation; thus, the docking result cannot truly reflect the state of the experimental docking [402].

3.5.2. Binding Affinity

Binding affinities are important for evaluating novel drug molecules and their targets. Within computational biology, this falls under applications of virtual screening and docking. The dissociation constant, K_d , is typically used to find binding affinity, as the two quantities are inversely proportional. Other useful metrics, particularly for enzymes and inhibitors, are the inhibition constant K_i , the half maximal inhibitory concentration (IC50) and the minimum inhibitory concentration (MIC). For high-throughput applications such as virtual screening, the measurement of thousands of K_d is inefficient, so there is much motivation for computationally fast and accurate predictions of binding affinities. In molecular docking, methods refer to binding affinity or just affinity as the scoring function for a particular ligand conformer. While some of these models can be based on physical interactions, and the physical binding affinity is certainly dependent on protein–ligand or protein–protein interactions, the predicted affinity is often not physical, and these scores are used primarily as a relative measure that should linearly correlate to true binding affinities in the best cases.

In principle, ML can be trained on experimental data to directly predict binding affinities. Therefore, datasets are important for this approach. A few examples of databases which contain binding affinity information are BioLiP [403], Binding MOAD [404], Bind-

ingDB [405], and PDBbind [406,407]. An early example of how ML can accelerate binding affinity calculations was the RF-Score [408], which used RF models. RF-Score trained on distance pair occurrences between types of protein and ligand atoms, and achieved state-of-the-art performances when it was released in 2010. Random forest ML improved classic binding affinity calculations by training models on features produced by models such as CYSCORE or AutoDock Vina [409,410]. Head-to-head comparisons between classic and ML prediction algorithms [411] have been performed, showing that ML models can outperform classic models in most useful tasks, such as predicting, ranking and docking with binding affinities. Finally, among traditional non-deep ML methods, RF is often the best predictor [412].

With deep learning and large datasets becoming increasingly accessible, various neural network architectures have been exploited, particularly the CNN architecture. Early work on applying CNNs to predict binding affinity was carried out in 2009 [413], where it was found that CNN provided highly accurate affinities for cations binding to common amino acids. In 2017, the subject of using CNNs as scoring functions for virtual screening and binding affinity prediction was picked up [414], and this became the dominant model. To prepare protein ligand structures into the correct input shape for a CNN, the interaction surface is often turned into a 3D grid, with each pixel representing information about the atoms contained within it. This type of CNN that learns directly from the 3D structure of the protein ligand complex is called a 3D CNN. The 3D input structure allows the input to include information about the local environment of the ligand. KDEEP [415], Pafnucy [416], and DeepAtom [417] are examples of a CNN that utilizes these 3D convolutions. While 3D CNNs are powerful predictors, it has been noted that moving from 2D to 3D drastically increases parameter space, allowing for more possibilities of over-fitting and slowing down the prediction time. This was addressed by DeepBindRG [418], which deflated the input size to a 2D array, rather than the 4D array needed for 3D CNNs, and used ResNet for binding affinity prediction. Another popular approach is OnionNet [419,420], which featurizes protein–ligand complexes in a similar way as RF-Score, using atom-pair-specific contacts but at varying spatial scales to create a 3D input array appropriate for 2D convolution learning. There is also work on applying CNNs directly to complexes in sequence space in DeepDTA [421], where the protein is represented by its sequence and the ligand by its SMILES code. The two representations are processed by independent CNN blocks and fed into a single CNN block that predicts the binding affinity. It is worth noting that DeepDTA has been highly influential in binding affinity prediction, with many models directly building off it.

Other deep-learning approaches have been used to predict binding affinities. For example, Zhao et al. [422] replaced the CNNs in the feature extraction blocks of DeepDTA with a set of GANs. With this modification, the GAN learns to generate features in an unsupervised manner. This semi-supervised framework had similar performance to its predecessor, but could use more data, which greatly reduces the need for labeled training data. Again, by building off DeepDTA's sequence approach, attention layers have been shown to increase the effectiveness of the model [423] while making it interpretable and allowing it to predict where binding sites might be in sequence space, with variable accuracy. Finally, GNNs have become popular for feature extraction and prediction tasks. GNNs have previously been shown to create “molecular fingerprints” that act as useful representations of molecules. Two methods which have used GNNs for are GraphDTA and GraphBar [424,425], both of which use GNNs to directly compute binding affinity. GraphDTA is similar to DeepDTA, but the CNN that featurizes the ligand is replaced with a GNN. GraphBar takes a hierarchical approach by constructing multiple graph representations of the complex using increasing distance cutoffs for interactions.

Another random forest model called iSEE predicts changes to binding affinity, $\Delta\Delta G$, due to mutations [245]. Again, databases are needed to train ML models. One such database is SKEMPI, which lists $\Delta\Delta G$ s in response to many mutations that can be used for ML training [426]. The iSEE paper compares to several other $\Delta\Delta G$ predictors including

FoldX [427], CC-PBSA [428], BeAtMuSiC [429], BindProfX [430], which uses evolutionary information from homologs, and mCSM [431,432]. To account for the effects of mutation, either MD simulations for conformational sampling and evolutionary sequence information in terms of models such as Position Specific Scoring Matrices (PSSM) must be exploited to predict changes to binding energies.

Although molecular docking and prediction of binding affinity is a mature area of computational biology that has been impactful in drug design for more than two decades, there remain open challenges. Better methods are needed to robustly identify binding sites on protein structure. This proved to be a difficult task, as binding models must consider environmental and preferential binding effects due to thermodynamic linkage [433–436]. Binding affinity includes thermodynamic and mechanical stability concerns involving enthalpy–entropy compensation, which leads to observable cooperativeness. To accurately model binding affinity, dynamic aspects of the protein and its binding partner must be taken into account beyond local flexibility characteristics. The result of these complexities is that scoring functions typically prove inadequate, because entropic effects are difficult to quantify, and capturing these effects requires considerable exploration of conformation ensembles. Although the experimental side of high-throughput screening is time consuming and costly, it is also the case that simulation of conformational ensembles is also costly and time consuming. We believe that methods that account for conformational ensembles and entropy will dominate the field in the long run, with the remaining methods becoming obsolete, even if they perform best in the short term for limited data sets. Including dynamics and environmental effects is essential to make non-incremental progress in this area, which amounts to combining the four pillars of computational biology.

4. Conclusions

Our review cited over 300 papers on ML methods used in applications of computational biology, and over 100 papers on protein function and computational methods that have historically been successful in protein function analysis. We tied together the ML methods used to elucidate protein function, along with practical applications involving proteins with drug discovery. The context of the underlying biophysics, biochemistry and molecular biology puts into perspective the domain knowledge required to successfully integrate ML into computational biology applications. The convergence of computational biology with ML has already shown many fruitful directions. Due to the complexity of the subject, we believe that the recent gains in model predictions for protein function or functional attributes will continue to increase in the foreseeable future. We have emphasized the conformational ensemble perspective, because there is a natural continuum that bridges proteins with various degrees of conserved structural motifs and disorder. We believe that modeling conformational ensembles and protein dynamics under different environmental conditions is necessary to make the most progress in protein science through computational biology. It is the modeling of these general concepts that also pushes ML to its current state-of-the-art limits. The convergence of computational biology with ML is clearly beneficial for both areas of study, as challenges in protein function analysis are addressed.

Future Opportunities

Despite recent work over the last two decades in terms of accurately describing conformational ensembles of proteins, more work is needed to represent these ensembles in a computationally tractable and mathematically complete way. ML is a tool that can help achieve controllable approximation. We stressed that protein function is linked to specificity, because proteins function in the crowded cellular environment. Understanding how proteins function under environmental changes requires finding representations for environmental effects. It appears ML can improve protein engineering through in silico mutation studies. Changes in the environment and the primary structure of a protein together have significant effects on stability and function, both of which are related to the conformational ensemble.

Another necessary step is to find ways to simulate protein dynamics over long time scales. Due to the intrinsic limitation of the MD methodology of integrating differential equations on the femtosecond time scale, coupled with the problem that biologically important timescales routinely exceed one second, and the desire to simulate ever larger systems, there will always be a major problem with sampling on classical computers in the foreseeable future. Although this sampling problem presents a bleak picture, there are already many ways to address this problem by using coarse-grained models, metadynamics, and other bias techniques to explore functionally relevant dynamics, with ML greatly improving the effectiveness of these methods. There is also the challenge of identifying functional dynamics, for which ML can be used for discriminant analysis and dimensionality reduction of MD simulation data. On these fronts, ML provides a means to solve—or at least mitigate—sampling problems and identify the functionally relevant part of the generated conformational ensemble. To obtain accurate thermodynamic properties, such as binding affinity, it is imperative that the appropriate conformational ensemble is sufficiently sampled. Although more sampling allows flexibility in docking to be accounted for, the docking application remains a challenging problem.

The sampling challenge also motivates the ongoing development of better forcefields for MD simulations and thermodynamic models that take into account non-additivity in conformational entropy. We also reviewed progress on forcefield development using ML methods that have already made progress on these challenges and promise many more advances in the near future. In this time of convergence between ML and computational biology, decades of nurtured ideas have begun to bear fruit. The protein folding problem, while remaining a major challenge in protein science, is on the cusp of being solved, as are many other grand challenges in protein science.

Author Contributions: All authors contributed to all elements in writing this review. D.J.J. acquired funding for this effort. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NIH, grant number R15GM146200 to PI DJJ, and partly funded by a SMART Scholarship awarded to C.A., funded by OUSD/R&E (The Under Secretary of Defense-Research and Engineering), National Defense Education Program (NDEP)/BA-1, Basic Research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Jarvis, R.A.; Patrick, E.A. Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Trans. Comput.* **1973**, C-22, 1025–1034. [[CrossRef](#)]
2. Sturm, B.L.; Ben-Tal, O.; Monaghan, Ú.; Collins, N.; Herremans, D.; Chew, E.; Hadjeres, G.; Deruty, E.; Pachet, F. Machine learning research that matters for music creation: A case study. *J. New Music Res.* **2019**, *48*, 36–55. [[CrossRef](#)]
3. Rodolfa, K.T.; Lamba, H.; Ghani, R. Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy. *Nat. Mach. Intell.* **2021**, *3*, 896–904. [[CrossRef](#)]
4. Brook, T. Music, Art, Machine Learning, and Standardization. *Leonardo* **2021**, 1–11. [_a_02135](#). [[CrossRef](#)]
5. Xu, C.; Jackson, S.A. Machine learning and complex biological data. *Genome Biol.* **2019**, *20*, 76. [[CrossRef](#)] [[PubMed](#)]
6. Alquraishi, M. ProteinNet: A standardized data set for machine learning of protein structure. *BMC Bioinform.* **2019**, *20*, 311. [[CrossRef](#)]
7. Robertson, A.D.; Murphy, K.P. Protein Structure and the Energetics of Protein Stability. *Chem. Rev.* **1997**, *97*, 1251–1268. [[CrossRef](#)]
8. Anfinsen, C.B. Principles that Govern the Folding of Protein Chains. *Science* **1973**, *181*, 223–230. [[CrossRef](#)]
9. Orengo, C.A.; Michie, A.D.; Jones, S.; Jones, D.T.; Swindells, M.B.; Thornton, J.M. CATH—A hierarchic classification of protein domain structures. *Structure* **1997**, *5*, 1093–1109. [[CrossRef](#)]

10. Chandonia, J.M.; Guan, L.; Lin, S.; Yu, C.; Fox, N.K.; Brenner, S.E. SCOPe: Improvements to the structural classification of proteins—Extended database to facilitate variant interpretation and machine learning. *Nucleic Acids Res.* **2022**, *50*, D553–D559. [[CrossRef](#)]
11. Dunker, A.K.; Lawson, J.D.; Brown, C.J.; Williams, R.M.; Romero, P.; Oh, J.S.; Oldfield, C.J.; Campen, A.M.; Ratliff, C.M.; Hipps, K.W.; et al. Intrinsically disordered protein. *J. Mol. Graph. Model.* **2001**, *19*, 26–59. [[CrossRef](#)]
12. Pawson, T.; Nash, P. Assembly of cell regulatory systems through protein interaction domains. *Science* **2003**, *300*, 445–452. [[CrossRef](#)] [[PubMed](#)]
13. Nooren, I.M.A. NEW EMBO MEMBER'S REVIEW: Diversity of protein-protein interactions. *EMBO J.* **2003**, *22*, 3486–3492. [[CrossRef](#)] [[PubMed](#)]
14. Alberts, B.; Heald, R.; Johnson, A.; Morgan, D.; Raff, M.; Roberts, K.; Walter, P. *Molecular Biology of the Cell*, 7th ed.; Garland Science, Taylor and Francis Group: New York, NY, USA, 2022.
15. Liberles, D.A.; Teichmann, S.A.; Bahar, I.; Bastolla, U.; Bloom, J.; Bornberg-Bauer, E.; Colwell, L.J.; de Koning, A.P.J.; Dokholyan, N.V.; Echave, J.; et al. The interface of protein structure, protein biophysics, and molecular evolution. *Protein Sci.* **2012**, *21*, 769–785. [[CrossRef](#)] [[PubMed](#)]
16. Livesay, D.R.; Jacobs, D.J. Conserved quantitative stability/flexibility relationships (QSFR) in an orthologous RNase H pair. *Proteins Struct. Funct. Bioinform.* **2006**, *62*, 130–143. [[CrossRef](#)] [[PubMed](#)]
17. Guerois, R.; Nielsen, J.E.; Serrano, L. Predicting Changes in the Stability of Proteins and Protein Complexes: A Study of More Than 1000 Mutations. *J. Mol. Biol.* **2002**, *320*, 369–387. [[CrossRef](#)]
18. Jacobs, D.J.; Livesay, D.R.; Hules, J.; Tasayco, M.L. Elucidating Quantitative Stability/Flexibility Relationships Within Thioredoxin and its Fragments Using a Distance Constraint Model. *J. Mol. Biol.* **2006**, *358*, 882–904. doi: 10.1016/j.jmb.2006.02.015. [[CrossRef](#)]
19. Sievers, F.; Wilm, A.; Dineen, D.; Gibson, T.J.; Karplus, K.; Li, W.; Lopez, R.; McWilliam, H.; Remmert, M.; Soding, J.; et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **2011**, *7*, 539. [[CrossRef](#)]
20. Katoh, K.; Standley, D.M. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Mol. Biol. Evol.* **2013**, *30*, 772–780. [[CrossRef](#)]
21. Edgar, R.C. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **2004**, *32*, 1792–1797. doi: 10.1093/nar/gkh340. [[CrossRef](#)]
22. Dayhoff, M.O. *Atlas of Protein Sequence and Structure*; National Biomedical Research Foundation: Waltham, MA, USA, 1972.
23. Henikoff, S.; Henikoff, J.G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* **1992**, *89*, 10915–10919. [[CrossRef](#)] [[PubMed](#)]
24. Aloy, P.; Russell, R.B. Structural systems biology: Modelling protein interactions. *Nat. Rev. Mol. Cell Biol.* **2006**, *7*, 188–197. [[CrossRef](#)]
25. Good, M.C.; Zalatan, J.G.; Lim, W.A. Scaffold Proteins: Hubs for Controlling the Flow of Cellular Information. *Science* **2011**, *332*, 680–686. [[CrossRef](#)] [[PubMed](#)]
26. Mehta, P.; Schwab, D.J. Energetic costs of cellular computation. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 17978–17982. [[CrossRef](#)] [[PubMed](#)]
27. Fall, C.P.; Marland, E.S.; Wagner, J.M.; Tyson, J.J. *Computational Cell Biology*; Springer: New York, NY, USA, 2004.
28. Wilke, C.O. Bringing Molecules Back into Molecular Evolution. *PLoS Comput. Biol.* **2012**, *8*, e1002572. [[CrossRef](#)] [[PubMed](#)]
29. Needleman, S.B.; Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **1970**, *48*, 443–453. [[CrossRef](#)]
30. Berman, H.M. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242. [[CrossRef](#)]
31. Levitt, M. The birth of computational structural biology. *Nat. Struct. Biol.* **2001**, *8*, 392–393. [[CrossRef](#)]
32. Dill, K.A.; Bromberg, S.; Yue, K.; Chan, H.S.; Ftebig, K.M.; Yee, D.P.; Thomas, P.D. Principles of protein folding—A perspective from simple exact models. *Protein Sci.* **2008**, *4*, 561–602. [[CrossRef](#)]
33. Takada, S. Gō model revisited. *Biophys. Physicobiol.* **2019**, *16*, 248–255. [[CrossRef](#)] [[PubMed](#)]
34. Ettayapuram Ramaprasad, A.S.; Uddin, S.; Casas-Finet, J.; Jacobs, D.J. Decomposing Dynamical Couplings in Mutated scFv Antibody Fragments into Stabilizing and Destabilizing Effects. *J. Am. Chem. Soc.* **2017**, *139*, 17508–17517. [[CrossRef](#)] [[PubMed](#)]
35. Dill, K.A. Additivity Principles in Biochemistry. *J. Biol. Chem.* **1997**, *272*, 701–704. [[CrossRef](#)] [[PubMed](#)]
36. Mark, A.E.; van Gunsteren, W.F. Decomposition of the free energy of a system in terms of specific interactions. Implications for theoretical and experimental studies. *J. Mol. Biol.* **1994**, *240*, 167–176. [[CrossRef](#)]
37. Jacobs, D.J.; Dallakyan, S.; Wood, G.G.; Heckathorne, A. Network rigidity at finite temperature: Relationships between thermodynamic stability, the nonadditivity of entropy, and cooperativity in molecular systems. *Phys. Rev. E* **2003**, *68*. [[CrossRef](#)]
38. Jacobs, D.J.; Dallakyan, S. Elucidating Protein Thermodynamics from the Three-Dimensional Structure of the Native State Using Network Rigidity. *Biophys. J.* **2005**, *88*, 903–915. [[CrossRef](#)] [[PubMed](#)]
39. Livesay, D.R.; Dallakyan, S.; Wood, G.G.; Jacobs, D.J. A flexible approach for understanding protein stability. *FEBS Lett.* **2004**, *576*, 468–476. [[CrossRef](#)]
40. Li, T.; Tracka, M.B.; Uddin, S.; Casas-Finet, J.; Jacobs, D.J.; Livesay, D.R. Rigidity Emerges during Antibody Evolution in Three Distinct Antibody Systems: Evidence from QSFR Analysis of Fab Fragments. *PLoS Comput. Biol.* **2015**, *11*, e1004327. [[CrossRef](#)]

41. Jacobs, D.J.; Wood, G.G. Understanding the α -helix to coil transition in polypeptides using network rigidity: Predicting heat and cold denaturation in mixed solvent conditions. *Biopolymers* **2004**, *75*, 1–31. [[CrossRef](#)]
42. Jackel, C.; Kast, P.; Hilvert, D. Protein design by directed evolution. *Annu. Rev. Biophys.* **2008**, *37*, 153–173. [[CrossRef](#)]
43. James, L.C.; Tawfik, D.S. Conformational diversity and protein evolution—A 60-year-old hypothesis revisited. *Trends Biochem. Sci.* **2003**, *28*, 361–368. [[CrossRef](#)]
44. Glasner, M.E.; Gerlt, J.A.; Babbitt, P.C. Mechanisms of protein evolution and their application to protein engineering. *Adv. Enzym. Relat. Areas Mol. Biol.* **2007**, *75*, 193–239. [[CrossRef](#)]
45. Cherkasov, A.; Muratov, E.N.; Fourches, D.; Varnek, A.; Baskin, I.I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y.C.; Todeschini, R.; et al. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* **2014**, *57*, 4977–5010. [[CrossRef](#)] [[PubMed](#)]
46. Pearson, K. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [[CrossRef](#)]
47. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [[CrossRef](#)]
48. Samuel, A. Computing Bit by Bit or Digital Computers Made Easy. *Proc. IRE* **1953**, *41*, 1223–1230. [[CrossRef](#)]
49. Samuel, A.L. Artificial Intelligence: A Frontier of Automation. *ANNALS Am. Acad. Political Soc. Sci.* **1962**, *340*, 10–20. [[CrossRef](#)]
50. Rosenblatt, F. Perceptron Simulation Experiments. *Proc. IRE* **1960**, *48*, 301–309. [[CrossRef](#)]
51. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
52. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
53. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
54. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
55. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [[CrossRef](#)]
56. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
57. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [[CrossRef](#)]
58. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370–3388. [[CrossRef](#)] [[PubMed](#)]
59. Liu, K.; Sun, X.; Jia, L.; Ma, J.; Xing, H.; Wu, J.; Gao, H.; Sun, Y.; Boulnois, F.; Fan, J. Chemi-Net: A Molecular Graph Convolutional Network for Accurate Drug Property Prediction. *Int. J. Mol. Sci.* **2019**, *20*, 3389. [[CrossRef](#)] [[PubMed](#)]
60. Friedman, J.H. On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Min. Knowl. Discov.* **1997**, *1*, 55–77. [[CrossRef](#)]
61. Wu, F.; Xu, J. Deep template-based protein structure prediction. *PLoS Comput. Biol.* **2021**, *17*, e1008954. [[CrossRef](#)] [[PubMed](#)]
62. Kuhlman, B.; Bradley, P. Advances in protein structure prediction and design. *Nat. Rev. Mol. Cell Biol.* **2019**, *20*, 681–697. [[CrossRef](#)] [[PubMed](#)]
63. Muhammed, M.T.; Aki-Yalcin, E. Homology modeling in drug discovery: Overview, current applications, and future perspectives. *Chem. Biol. Drug Des.* **2019**, *93*, 12–20. [[CrossRef](#)]
64. Seffernick, J.T.; Lindert, S. Hybrid methods for combined experimental and computational determination of protein structure. *J. Chem. Phys.* **2020**, *153*, 240901. [[CrossRef](#)]
65. Burley, S.K.; Joachimiak, A.; Montelione, G.T.; Wilson, I.A. Contributions to the NIH-NIGMS Protein Structure Initiative from the PSI Production Centers. *Structure* **2008**, *16*, 5–11. [[CrossRef](#)]
66. Bolje, A.; Gobec, S. Analytical Techniques for Structural Characterization of Proteins in Solid Pharmaceutical Forms: An Overview. *Pharmaceutics* **2021**, *13*, 534. [[CrossRef](#)]
67. Li, X.; Li, Y.; Cheng, T.; Liu, Z.; Wang, R. Evaluation of the performance of four molecular docking programs on a diverse set of protein-ligand complexes. *J. Comput. Chem.* **2010**, *31*, 2109–2125. [[CrossRef](#)]
68. Dhingra, S.; Sowdhamini, R.; Cadet, F.; Offmann, B. A glance into the evolution of template-free protein structure prediction methodologies. *Biochimie* **2020**, *175*, 85–92. [[CrossRef](#)] [[PubMed](#)]
69. Roy, A.; Kucukural, A.; Zhang, Y. I-TASSER: A unified platform for automated protein structure and function prediction. *Nat. Protoc.* **2010**, *5*, 725–738. [[CrossRef](#)]
70. Bystroff, C.; Baker, D. Prediction of local structure in proteins using a library of sequence-structure motifs. *J. Mol. Biol.* **1998**, *281*, 565–577. [[CrossRef](#)] [[PubMed](#)]

71. Rohl, C.A.; Strauss, C.E.; Misura, K.M.; Baker, D. Protein structure prediction using Rosetta. *Methods Enzymol.* **2004**, *383*, 66–93. [[CrossRef](#)]
72. Noé, F.; De Fabritiis, G.; Clementi, C. Machine learning for protein folding and dynamics. *Curr. Opin. Struct. Biol.* **2020**, *60*, 77–84. [[CrossRef](#)]
73. Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; Moult, J. Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins* **2019**, *87*, 1011–1020. [[CrossRef](#)]
74. Heo, L.; Feig, M. High-accuracy protein structures by combining machine-learning with physics-based refinement. *Proteins* **2020**, *88*, 637–642. [[CrossRef](#)]
75. Ovchinnikov, S.; Park, H.; Kim, D.E.; Dimaio, F.; Baker, D. Protein structure prediction using Rosetta in CASP12. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 113–121. [[CrossRef](#)]
76. Hong, S.H.; Joung, I.; Flores-Canales, J.C.; Manavalan, B.; Cheng, Q.; Heo, S.; Kim, J.Y.; Lee, S.Y.; Nam, M.; Joo, K.; et al. Protein structure modeling and refinement by global optimization in CASP12. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 122–135. [[CrossRef](#)]
77. Zhang, C.; Mortuza, S.M.; He, B.; Wang, Y.; Zhang, Y. Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. *Proteins Struct. Funct. Bioinform.* **2018**, *86*, 136–151. [[CrossRef](#)]
78. Olechnovič, K.; Venclovas, Č. VoroMQA: Assessment of protein structure quality using interatomic contact areas. *Proteins Struct. Funct. Bioinform.* **2017**, *85*, 1131–1145. [[CrossRef](#)] [[PubMed](#)]
79. Alquraishi, M. AlphaFold at CASP13. *Bioinformatics* **2019**, *35*, 4862–4865. [[CrossRef](#)] [[PubMed](#)]
80. Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.; Bridgland, A.; Penedones, H.; et al. De novo structure prediction with deeplearning based scoring. *Annu. Rev. Biochem.* **2018**, *77*, 6.
81. Li, Y.; Zhang, C.; Bell, E.W.; Yu, D.; Zhang, Y. Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. *Proteins Struct. Funct. Bioinform.* **2019**, *87*, 1082–1091. [[CrossRef](#)]
82. Hou, J.; Wu, T.; Cao, R.; Cheng, J. Protein tertiary structure modeling driven by deep learning and contact distance prediction in CASP13. *Proteins Struct. Funct. Bioinform.* **2019**, *87*, 1165–1178. [[CrossRef](#)]
83. Zheng, W.; Li, Y.; Zhang, C.; Pearce, R.; Mortuza, S.M.; Zhang, Y. Deep-learning contact-map guided protein structure prediction in CASP13. *Proteins Struct. Funct. Bioinform.* **2019**, *87*, 1149–1164. [[CrossRef](#)] [[PubMed](#)]
84. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Židek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [[CrossRef](#)]
85. Anishchenko, I.; Baek, M.; Park, H.; Hiranuma, N.; Kim, D.E.; Dauparas, J.; Mansoor, S.; Humphreys, I.R.; Baker, D. Protein tertiary structure prediction and refinement using deep learning and Rosetta in CASP14. *Proteins Struct. Funct. Bioinform.* **2021**, *89*, 1722–1733. [[CrossRef](#)]
86. Baek, M.; Anishchenko, I.; Park, H.; Humphreys, I.R.; Baker, D. Protein oligomer modeling guided by predicted interchain contacts in CASP14. *Proteins Struct. Funct. Bioinform.* **2021**, *89*, 1824–1833. [[CrossRef](#)]
87. Heo, L.; Janson, G.; Feig, M. Physics-based protein structure refinement in the era of artificial intelligence. *Proteins Struct. Funct. Bioinform.* **2021**, *89*, 1870–1887. [[CrossRef](#)]
88. Zheng, W.; Li, Y.; Zhang, C.; Zhou, X.; Pearce, R.; Bell, E.W.; Huang, X.; Zhang, Y. Protein structure prediction using deep learning distance and hydrogen-bonding restraints in CASP14. *Proteins* **2021**, *89*, 1734–1751. [[CrossRef](#)]
89. Fersht, A.R. AlphaFold—A Personal Perspective on the Impact of Machine Learning. *J. Mol. Biol.* **2021**, *433*, 167088. [[CrossRef](#)] [[PubMed](#)]
90. AlQuraishi, M. Machine learning in protein structure prediction. *Curr. Opin. Chem. Biol.* **2021**, *65*, 1–8. [[CrossRef](#)] [[PubMed](#)]
91. Torrisi, M.; Pollastri, G.; Le, Q. Deep learning methods in protein structure prediction. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 1301–1310. [[CrossRef](#)] [[PubMed](#)]
92. De Oliveira, S.H.P.; Shi, J.; Deane, C.M. Comparing co-evolution methods and their application to template-free protein structure prediction. *Bioinformatics* **2016**, *33*, 373–381. [[CrossRef](#)]
93. Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G.R.; Wang, J.; Cong, Q.; Kinch, L.N.; Schaeffer, R.D.; et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373*, 871–876. [[CrossRef](#)]
94. Ahdritz, G.; Bouatta, N.; Kadyan, S.; Xia, Q.; Gerecke, W.; AlQuraishi, M. OpenFold. *Zenodo* **2021**. [[CrossRef](#)]
95. Wu, R.; Ding, F.; Wang, R.; Shen, R.; Zhang, X.; Luo, S.; Su, C.; Wu, Z.; Xie, Q.; Berger, B.; et al. High-resolution *de novo* structure prediction from primary sequence. *bioRxiv* **2022**. [[CrossRef](#)]
96. Sliwoski, G.; Kothiwale, S.; Meiler, J.; Lowe, E.W. Computational methods in drug discovery. *Pharmacol. Rev.* **2014**, *66*, 334–395. [[CrossRef](#)]
97. Leelananda, S.P.; Lindert, S. Computational methods in drug discovery. *Beilstein J. Org. Chem.* **2016**, *12*, 2694–2718. [[CrossRef](#)]
98. Kokh, D.B.; Kaufmann, T.; Kister, B.; Wade, R.C. Machine Learning Analysis of tauRAMD Trajectories to Decipher Molecular Determinants of Drug-Target Residence Times. *Front. Mol. Biosci.* **2019**, *6*, 36. [[CrossRef](#)] [[PubMed](#)]
99. Lima, A.N.; Philot, E.A.; Trossini, G.H.G.; Scott, L.P.B.; Maltarollo, V.G.; Honorio, K.M. Use of machine learning approaches for novel drug discovery. *Expert Opin. Drug Discov.* **2016**, *11*, 225–239. [[CrossRef](#)] [[PubMed](#)]
100. Zhu, S.; Shala, A.; Bezginov, A.; Sljoka, A.; Audette, G.; Wilson, D.J. Hyperphosphorylation of Intrinsically Disordered Tau Protein Induces an Amyloidogenic Shift in Its Conformational Ensemble. *PLoS ONE* **2015**, *10*, e0120416. [[CrossRef](#)]

101. Joshi, S.Y.; Deshmukh, S.A. A review of advancements in coarse-grained molecular dynamics simulations. *Mol. Simul.* **2021**, *47*, 786–803. [[CrossRef](#)]
102. Liwo, A.; Czaplewski, C.; Sieradzan, A.K.; Lipska, A.G.; Samsonov, S.A.; Murarka, R.K. Theory and Practice of Coarse-Grained Molecular Dynamics of Biologically Important Systems. *Biomolecules* **2021**, *11*, 1347. [[CrossRef](#)]
103. Singh, N.; Li, W. Recent Advances in Coarse-Grained Models for Biomolecules and Their Applications. *Int. J. Mol. Sci.* **2019**, *20*, 3774. [[CrossRef](#)]
104. Togashi, Y.; Flechsig, H. Coarse-Grained Protein Dynamics Studies Using Elastic Network Models. *Int. J. Mol. Sci.* **2018**, *19*, 3899. [[CrossRef](#)]
105. Marrink, S.J.; Risselada, H.J.; Yefimov, S.; Tieleman, D.P.; de Vries, A.H. The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **2007**, *111*, 7812–7824. [[CrossRef](#)]
106. Marrink, S.J.; Monticelli, L.; Melo, M.N.; Alessandri, R.; Tieleman, D.P.; Souza, P.C.T. Two decades of Martini: Better beads, broader scope. *WIREs Comput. Mol. Sci.* **2022**, e1620. [[CrossRef](#)]
107. Rojas, A.; Czaplewski, C.; Liwo, A.; Makowski, M.; O_dziej, S.; Kaz, R.; Scheraga, H.; Murarka, R.; Voth, G. Simulation of Protein Structure and Dynamics with the Coarse-Grained UNRES Force Field. *Coarse-Graining Condens. Phase Biomol. Syst.* **2008**, *1*, 1391–1411.
108. Liwo, A.; Baranowski, M.; Czaplewski, C.; Gołaś, E.; He, Y.; Jagieła, D.; Krupa, P.; Maciejczyk, M.; Makowski, M.; Mozolewska, M.A.; et al. A unified coarse-grained model of biological macromolecule based on mean-field multipole–multipole interactions. *J. Mol. Model.* **2014**, *20*, 2306. [[CrossRef](#)] [[PubMed](#)]
109. Peng, J.; Yuan, C.; Ma, R.; Zhang, Z. Backmapping from Multiresolution Coarse-Grained Models to Atomic Structures of Large Biomolecules by Restrained Molecular Dynamics Simulations Using Bayesian Inference. *J. Chem. Theory Comput.* **2019**, *15*, 3344–3353. [[CrossRef](#)] [[PubMed](#)]
110. Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **2018**, *149*, 034101. [[CrossRef](#)] [[PubMed](#)]
111. Wang, J.; Olsson, S.; Wehmeyer, C.; Perez, A.; Charron, N.E.; de Fabritiis, G.; Noe, F.; Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **2019**, *5*, 755–767. [[CrossRef](#)]
112. Husic, B.E.; Charron, N.E.; Lemm, D.; Wang, J.; Perez, A.; Majewski, M.; Kramer, A.; Chen, Y.; Olsson, S.; de Fabritiis, G.; et al. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **2020**, *153*, 194101. [[CrossRef](#)]
113. Wang, J.; Chmiela, S.; Muller, K.R.; Noe, F.; Clementi, C. Ensemble learning of coarse-grained molecular dynamics force fields with a kernel approach. *J. Chem. Phys.* **2020**, *152*, 194106. [[CrossRef](#)]
114. Zhou, R. Replica exchange molecular dynamics method for protein folding simulation. *Methods Mol. Biol.* **2007**, *350*, 205–223. [[CrossRef](#)]
115. Mori, T.; Miyashita, N.; Im, W.; Feig, M.; Sugita, Y. Molecular dynamics simulations of biological membranes and membrane proteins using enhanced conformational sampling algorithms. *Biochim. Biophys. Acta* **2016**, *1858*, 1635–1651. [[CrossRef](#)]
116. Affentranger, R.; Tavernelli, I.; Di Iorio, E.E. A Novel Hamiltonian Replica Exchange MD Protocol to Enhance Protein Conformational Space Sampling. *J. Chem. Theory Comput.* **2006**, *2*, 217–228. [[CrossRef](#)] [[PubMed](#)]
117. Bernardi, R.C.; Melo, M.C.R.; Schulten, K. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochim. Biophys. Acta* **2015**, *1850*, 872–877. [[CrossRef](#)]
118. Melo, M.C.; Bernardi, R.C.; Fernandes, T.V.; Pascutti, P.G. GSAFold: A new application of GSA to protein structure prediction. *Proteins* **2012**, *80*, 2305–2310. [[CrossRef](#)] [[PubMed](#)]
119. Laio, A.; Parrinello, M. Escaping free-energy minima. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 12562–12566. [[CrossRef](#)] [[PubMed](#)]
120. Barducci, A.; Bussi, G.; Parrinello, M. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Phys. Rev. Lett.* **2008**, *100*, 020603. [[CrossRef](#)] [[PubMed](#)]
121. Comer, J.; Gumbart, J.C.; Henin, J.; Lelievre, T.; Pohorille, A.; Chipot, C. The adaptive biasing force method: Everything you always wanted to know but were afraid to ask. *J. Phys. Chem. B* **2015**, *119*, 1129–1151. [[CrossRef](#)] [[PubMed](#)]
122. Héning, J.; Chipot, C. Overcoming free energy barriers using unconstrained molecular dynamics simulations. *J. Chem. Phys.* **2004**, *121*, 2904–2914. [[CrossRef](#)]
123. Liphardt, J.; Dumont, S.; Smith, S.B.; Tinoco, I.; Bustamante, C. Equilibrium Information from Nonequilibrium Measurements in an Experimental Test of Jarzynski’s Equality. *Science* **2002**, *296*, 1832–1835. [[CrossRef](#)]
124. Shamsi, Z.; Moffett, A.S.; Shukla, D. Enhanced unbiased sampling of protein dynamics using evolutionary coupling information. *Sci. Rep.* **2017**, *7*, 12700. [[CrossRef](#)]
125. Palazzesi, F.; Valsson, O.; Parrinello, M. Conformational Entropy as Collective Variable for Proteins. *J. Phys. Chem. Lett.* **2017**, *8*, 4752–4756. [[CrossRef](#)]
126. Fiorin, G.; Klein, M.L.; Héning, J. Using collective variables to drive molecular dynamics simulations. *Mol. Phys.* **2013**, *111*, 3345–3362. [[CrossRef](#)]
127. Chen, M. Collective variable-based enhanced sampling and machine learning. *Eur. Phys. J. B* **2021**, *94*, 1–17. [[CrossRef](#)]
128. Amadei, A.; Linssen, A.B.; Berendsen, H.J. Essential dynamics of proteins. *Proteins* **1993**, *17*, 412–425. [[CrossRef](#)] [[PubMed](#)]
129. David, C.C.; Avery, C.S.; Jacobs, D.J. JEDi: Java essential dynamics inspector—A molecular trajectory analysis toolkit. *BMC Bioinform.* **2021**, *22*, 226. [[CrossRef](#)] [[PubMed](#)]

130. Michaud-Agrawal, N.; Denning, E.J.; Woolf, T.B.; Beckstein, O. MDAAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.* **2011**, *32*, 2319–2327. [[CrossRef](#)] [[PubMed](#)]
131. Ross, C.; Nizami, B.; Glenister, M.; Sheik Amamuddy, O.; Atilgan, A.R.; Atilgan, C.; Tastan Bishop, O. MODE-TASK: Large-scale protein motion tools. *Bioinformatics* **2018**, *34*, 3759–3763. [[CrossRef](#)]
132. Peng, J.; Zhang, Z. Simulating Large-Scale Conformational Changes of Proteins by Accelerating Collective Motions Obtained from Principal Component Analysis. *J. Chem. Theory Comput.* **2014**, *10*, 3449–3458. [[CrossRef](#)]
133. Shkurti, A.; Styliari, I.D.; Balasubramanian, V.; Bethune, I.; Pedebos, C.; Jha, S.; Laughton, C.A. CoCo-MD: A Simple and Effective Method for the Enhanced Sampling of Conformational Space. *J. Chem. Theory Comput.* **2019**, *15*, 2587–2596. [[CrossRef](#)]
134. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [[CrossRef](#)]
135. Spiwok, V.; Kralova, B. Metadynamics in the conformational space nonlinearly dimensionally reduced by Isomap. *J. Chem. Phys.* **2011**, *135*, 224504. [[CrossRef](#)]
136. Tribello Gareth, A.; Ceriotti, M.; Parrinello, M. Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 5196–5201. [[CrossRef](#)]
137. Rohrdanz, M.A.; Zheng, W.; Maggioni, M.; Clementi, C. Determination of reaction coordinates via locally scaled diffusion map. *J. Chem. Phys.* **2011**, *134*, 124116. [[CrossRef](#)]
138. Sultan, M.M.; Pande, V.S. Automated design of collective variables using supervised machine learning. *J. Chem. Phys.* **2018**, *149*, 094106. [[CrossRef](#)] [[PubMed](#)]
139. Naritomi, Y.; Fuchigami, S. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *J. Chem. Phys.* **2011**, *134*, 065101. [[CrossRef](#)] [[PubMed](#)]
140. Hyvarinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430. [[CrossRef](#)]
141. Perez-Hernandez, G.; Noe, F. Hierarchical Time-Lagged Independent Component Analysis: Computing Slow Modes and Reaction Coordinates for Large Molecular Systems. *J. Chem. Theory Comput.* **2016**, *12*, 6118–6129. [[CrossRef](#)]
142. M, M.S.; Pande, V.S. tICA-Metadynamics: Accelerating Metadynamics by Using Kinetically Selected Collective Variables. *J. Chem. Theory Comput.* **2017**, *13*, 2440–2447. [[CrossRef](#)]
143. Perez-Hernandez, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noe, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, 015102. [[CrossRef](#)] [[PubMed](#)]
144. Scherer, M.K.; Trendelkamp-Schroer, B.; Paul, F.; Perez-Hernandez, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.H.; Noe, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11*, 5525–5542. [[CrossRef](#)]
145. Harrigan, M.P.; Sultan, M.M.; Hernandez, C.X.; Husic, B.E.; Eastman, P.; Schwantes, C.R.; Beauchamp, K.A.; McGibbon, R.T.; Pande, V.S. MSMBuilder: Statistical Models for Biomolecular Dynamics. *Biophys. J.* **2017**, *112*, 10–15. [[CrossRef](#)]
146. Ma, A.; Dinner, A.R. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B* **2005**, *109*, 6769–6779. [[CrossRef](#)]
147. Chen, W.; Ferguson, A.L. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *J. Comput. Chem.* **2018**, *39*, 2079–2102. [[CrossRef](#)]
148. Chen, W.; Tan, A.R.; Ferguson, A.L. Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.* **2018**, *149*, 072312. [[CrossRef](#)] [[PubMed](#)]
149. Jayachandran, G.; Vishal, V.; Pande, V.S. Using massively parallel simulation and Markovian models to study protein folding: Examining the dynamics of the villin headpiece. *J. Chem. Phys.* **2006**, *124*, 164902. [[CrossRef](#)] [[PubMed](#)]
150. Chodera, J.D.; Swope, W.C.; Pitera, J.W.; Dill, K.A. Long-Time Protein Folding Dynamics from Short-Time Molecular Dynamics Simulations. *Multiscale Model. Simul.* **2006**, *5*, 1214–1226. [[CrossRef](#)]
151. Wehmeyer, C.; Noe, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **2018**, *148*, 241703. [[CrossRef](#)] [[PubMed](#)]
152. Lamim Ribeiro, J.M.; Provasi, D.; Filizola, M. A combination of machine learning and infrequent metadynamics to efficiently predict kinetic rates, transition states, and molecular determinants of drug dissociation from G protein-coupled receptors. *J. Chem. Phys.* **2020**, *153*, 124105. [[CrossRef](#)]
153. Ravindra, P.; Smith, Z.; Tiwary, P. Automatic mutual information noise omission (AMINO): Generating order parameters for molecular systems. *Mol. Syst. Des. Eng.* **2020**, *5*, 339–348. [[CrossRef](#)]
154. Ribeiro, J.M.L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, 072301. [[CrossRef](#)]
155. Wu, H.; Noé, F. Variational Approach for Learning Markov Processes from Time Series Data. *J. Nonlinear Sci.* **2020**, *30*, 23–66. [[CrossRef](#)]
156. Koopman, B.O. Hamiltonian Systems and Transformation in Hilbert Space. *Proc. Natl. Acad. Sci. USA* **1931**, *17*, 315–318. [[CrossRef](#)]
157. Koopman, B.O.; Neumann, J.V. Dynamical Systems of Continuous Spectra. *Proc. Natl. Acad. Sci. USA* **1932**, *18*, 255–263. [[CrossRef](#)]
158. Williams, M.O.; Kevrekidis, I.G.; Rowley, C.W. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *J. Nonlinear Sci.* **2015**, *25*, 1307–1346. [[CrossRef](#)]
159. Mardt, A.; Pasquali, L.; Wu, H.; Noe, F. VAMPnets for deep learning of molecular kinetics. *Nat. Commun.* **2018**, *9*, 5. [[CrossRef](#)]

160. Sidky, H.; Chen, W.; Ferguson, A.L. High-Resolution Markov State Models for the Dynamics of Trp-Cage Miniprotein Constructed Over Slow Folding Modes Identified by State-Free Reversible VAMPnets. *J. Phys. Chem. B* **2019**, *123*, 7999–8009. [[CrossRef](#)] [[PubMed](#)]
161. Konovalov, K.A.; Unarta, I.C.; Cao, S.; Goonetilleke, E.C.; Huang, X. Markov State Models to Study the Functional Dynamics of Proteins in the Wake of Machine Learning. *JACS Au* **2021**, *1*, 1330–1341. [[CrossRef](#)]
162. Laio, A.; Gervasio, F.L. Metadynamics: A method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Rep. Prog. Phys.* **2008**, *71*, 126601. [[CrossRef](#)]
163. Galvelis, R.; Sugita, Y. Neural Network and Nearest Neighbor Algorithms for Enhancing Sampling of Molecular Dynamics. *J. Chem. Theory Comput.* **2017**, *13*, 2489–2500. [[CrossRef](#)]
164. Guo, A.Z.; Sevgen, E.; Sidky, H.; Whitmer, J.K.; Hubbell, J.A.; Pablo, J.J.d. Adaptive enhanced sampling by force-biasing using neural networks. *J. Chem. Phys.* **2018**, *148*, 134108. [[CrossRef](#)]
165. Sidky, H.; Whitmer, J.K. Learning free energy landscapes using artificial neural networks. *J. Chem. Phys.* **2018**, *148*, 104111. [[CrossRef](#)]
166. Salawu, E.O. DESP: Deep Enhanced Sampling of Proteins' Conformation Spaces Using AI-Inspired Biasing Forces. *Front. Mol. Biosci.* **2021**, *8*, 587151. [[CrossRef](#)]
167. Ezugwu, A.E.; Ikotun, A.M.; Oyelade, O.O.; Abualigah, L.; Agushaka, J.O.; Eke, C.I.; Akinyelu, A.A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104743. [[CrossRef](#)]
168. Zhang, Y.; Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **2004**, *57*, 702–710. [[CrossRef](#)]
169. Holm, L.; Sander, C. Protein Structure Comparison by Alignment of Distance Matrices. *J. Mol. Biol.* **1993**, *233*, 123–138. [[CrossRef](#)] [[PubMed](#)]
170. Shindyalov, I.N.; Bourne, P.E. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* **1998**, *11*, 739–747. [[CrossRef](#)] [[PubMed](#)]
171. Madej, T.; Lanczycki, C.J.; Zhang, D.; Thiessen, P.A.; Geer, R.C.; Marchler-Bauer, A.; Bryant, S.H. MMDB and VAST+: Tracking structural similarities between macromolecular complexes. *Nucleic Acids Res.* **2014**, *42*, D297–D303. [[CrossRef](#)] [[PubMed](#)]
172. Shirikhorshidi, A.S.; Aghabozorgi, S.; Wah, T.Y. A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLoS ONE* **2015**, *10*, e0144059. [[CrossRef](#)]
173. Mehta, V.; Bawa, S.; Singh, J. Analytical review of clustering techniques and proximity measures. *Artif. Intell. Rev.* **2020**, *53*, 5995–6023. [[CrossRef](#)]
174. Bowman, G.R. Improved coarse-graining of Markov state models via explicit consideration of statistical uncertainty. *J. Chem. Phys.* **2012**, *137*, 134111. [[CrossRef](#)]
175. Baek, M.; Kim, C. A review on spectral clustering and stochastic block models. *J. Korean Stat. Soc.* **2021**, *50*, 818–831. [[CrossRef](#)]
176. Röblitz, S.; Weber, M. Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification. *Adv. Data Anal. Classif.* **2013**, *7*, 147–179. [[CrossRef](#)]
177. Deuffhard, P.; Weber, M. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra Its Appl.* **2005**, *398*, 161–184. [[CrossRef](#)]
178. Huang, R.; Lo, L.T.; Wen, Y.; Voter, A.F.; Perez, D. Cluster analysis of accelerated molecular dynamics simulations: A case study of the decahedron to icosahedron transition in Pt nanoparticles. *J. Chem. Phys.* **2017**, *147*, 152717. [[CrossRef](#)]
179. Huang, X.; Yao, Y.; Bowman, G.R.; Sun, J.; Guibas, L.J.; Carlsson, G.; Pande, V.S. Constructing multi-resolution Markov State Models (MSMs) to elucidate RNA hairpin folding mechanisms. *Pac. Symp. Biocomput.* **2010**, *2010*, 228–239. [[CrossRef](#)]
180. Yao, Y.; Cui, R.Z.; Bowman, G.R.; Silva, D.A.; Sun, J.; Huang, X. Hierarchical Nyström methods for constructing Markov state models for conformational dynamics. *J. Chem. Phys.* **2013**, *138*, 174106. [[CrossRef](#)] [[PubMed](#)]
181. Jain, A.; Stock, G. Identifying Metastable States of Folding Proteins. *J. Chem. Theory Comput.* **2012**, *8*, 3810–3819. [[CrossRef](#)]
182. Wang, W.; Cao, S.; Zhu, L.; Huang, X. Constructing Markov State Models to elucidate the functional conformational changes of complex biomolecules. *WIREs Comput. Mol. Sci.* **2018**, *8*, e1343. [[CrossRef](#)]
183. Orioli, S.; Faccioli, P. Dimensional reduction of Markov state models from renormalization group theory. *J. Chem. Phys.* **2016**, *145*, 124120. [[CrossRef](#)] [[PubMed](#)]
184. Zhu, L.; Sheong, F.K.; Zeng, X.; Huang, X. Elucidation of the conformational dynamics of multi-body systems by construction of Markov state models. *Phys. Chem. Chem. Phys.* **2016**, *18*, 30228–30235. [[CrossRef](#)]
185. Cocina, F.; Vitalis, A.; Caffisch, A. Sapphire-Based Clustering. *J. Chem. Theory Comput.* **2020**, *16*, 6383–6396. [[CrossRef](#)] [[PubMed](#)]
186. Mallet, V.; Nilges, M.; Bouvier, G. quicksom: Self-Organizing Maps on GPUs for clustering of molecular dynamics trajectories. *Bioinformatics* **2021**, *37*, 2064–2065. [[CrossRef](#)] [[PubMed](#)]
187. Pauling, L.; Corey, R.B.; Branson, H.R. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA* **1951**, *37*, 205–211. [[CrossRef](#)]
188. Rao, S.J.A.; Shetty, N.P. Evolutionary selectivity of amino acid is inspired from the enhanced structural stability and flexibility of the folded protein. *Life Sci.* **2021**, *281*, 119774. [[CrossRef](#)] [[PubMed](#)]
189. Walport, L.J.; Low, J.K.K.; Matthews, J.M.; Mackay, J.P. The characterization of protein interactions—What, how and how much? *Chem. Soc. Rev.* **2021**, *50*, 12292–12307. [[CrossRef](#)] [[PubMed](#)]

190. Murzin, A.G.; Brenner, S.E.; Hubbard, T.; Chothia, C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **1995**, *247*, 536–540. [[CrossRef](#)]
191. Frishman, D.; Argos, P. Knowledge-based protein secondary structure assignment. *Proteins Struct. Funct. Bioinform.* **1995**, *23*, 566–579. [[CrossRef](#)] [[PubMed](#)]
192. Kabsch, W.; Sander, C. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **1983**, *22*, 2577–2637. [[CrossRef](#)] [[PubMed](#)]
193. Zhang, C.; Liu, S.; Zhu, Q.; Zhou, Y. A Knowledge-Based Energy Function for Protein–Ligand, Protein–Protein, and Protein–DNA Complexes. *J. Med. Chem.* **2005**, *48*, 2325–2335. [[CrossRef](#)]
194. Dodge, C.; Schneider, R.; Sander, C. The HSSP database of protein structure—Sequence alignments and family profiles. *Nucleic Acids Res.* **1998**, *26*, 313–315. [[CrossRef](#)]
195. Lobry, J.; Gautier, C. Hydrophobicity, expressivity and aromaticity are the major trends of amino-acid usage in 999 Escherichia coli chromosome-encoded genes. *Nucleic Acids Res.* **1994**, *22*, 3174–3180. [[CrossRef](#)]
196. Huang, P.; Chu, S.K.S.; Frizzo, H.N.; Connolly, M.P.; Caster, R.W.; Siegel, J.B. Evaluating Protein Engineering Thermostability Prediction Tools Using an Independently Generated Dataset. *ACS Omega* **2020**, *5*, 6487–6493. [[CrossRef](#)]
197. Mohan, A.; Oldfield, C.J.; Radivojac, P.; Vacic, V.; Cortese, M.S.; Dunker, A.K.; Uversky, V.N. Analysis of Molecular Recognition Features (MoRFs). *J. Mol. Biol.* **2006**, *362*, 1043–1059. [[CrossRef](#)]
198. van der Lee, R.; Buljan, M.; Lang, B.; Weatheritt, R.J.; Daughdrill, G.W.; Dunker, A.K.; Fuxreiter, M.; Gough, J.; Gsponer, J.; Jones, D.T.; et al. Classification of intrinsically disordered regions and proteins. *Chem. Rev.* **2014**, *114*, 6589–6631. [[CrossRef](#)]
199. Rother, D.; Sapiro, G.; Pande, V. Statistical characterization of protein ensembles. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2008**, *5*, 42–55. [[CrossRef](#)] [[PubMed](#)]
200. Bouvier, G.; Desdouits, N.; Ferber, M.; Blondel, A.; Nilges, M. An automatic tool to analyze and cluster macromolecular conformations based on self-organizing maps. *Bioinformatics* **2015**, *31*, 1490–1492. [[CrossRef](#)]
201. Bhattacharyya, M.; Bhat, C.R.; Vishveshwara, S. An automated approach to network features of protein structure ensembles. *Protein Sci.* **2013**, *22*, 1399–1416. [[CrossRef](#)]
202. Jo, T.; Hou, J.; Eickholt, J.; Cheng, J. Improving Protein Fold Recognition by Deep Learning Networks. *Sci. Rep.* **2015**, *5*, 17573. [[CrossRef](#)] [[PubMed](#)]
203. Du, Z.; Su, H.; Wang, W.; Ye, L.; Wei, H.; Peng, Z.; Anishchenko, I.; Baker, D.; Yang, J. The trRosetta server for fast and accurate protein structure prediction. *Nat. Protoc.* **2021**, *16*, 5634–5651. [[CrossRef](#)]
204. Misiura, M.; Shroff, R.; Thyer, R.; Kolomeisky, A.B. DLPacker: Deep learning for prediction of amino acid side chain conformations in proteins. *Proteins Struct. Funct. Bioinform.* **2022**, *90*, 1278–1290. [[CrossRef](#)]
205. King, J.E.; Koes, D.R. SidechainNet: An all-atom protein structure dataset for machine learning. *Proteins Struct. Funct. Bioinform.* **2021**, *89*, 1489–1496. [[CrossRef](#)]
206. Igashov, I.; Olechnovič, K.; Kadukova, M.; Venclovas, Č.; Grudin, S. VoroCNN: Deep convolutional neural network built on 3D Voronoi tessellation of protein structures. *Bioinformatics* **2021**, *37*, 2332–2339. [[CrossRef](#)]
207. Luttrell, J.; Liu, T.; Zhang, C.; Wang, Z. Predicting protein residue-residue contacts using random forests and deep networks. *BMC Bioinform.* **2019**, *20*, 100. [[CrossRef](#)] [[PubMed](#)]
208. Audagnotto, M.; Czechtizky, W.; De Maria, L.; Käck, H.; Papoian, G.; Tornberg, L.; Tyrchan, C.; Ulander, J. Machine learning/molecular dynamic protein structure prediction approach to investigate the protein conformational ensemble. *Sci. Rep.* **2022**, *12*, 10018. [[CrossRef](#)]
209. Duong, V.T.; Diessner, E.M.; Grazioli, G.; Martin, R.W.; Butts, C.T. Neural Upscaling from Residue-Level Protein Structure Networks to Atomistic Structures. *Biomolecules* **2021**, *11*, 1788. [[CrossRef](#)] [[PubMed](#)]
210. Mok, K.H.; Kuhn, L.T.; Goetz, M.; Day, I.J.; Lin, J.C.; Andersen, N.H.; Hore, P.J. A pre-existing hydrophobic collapse in the unfolded state of an ultrafast folding protein. *Nature* **2007**, *447*, 106–109. [[CrossRef](#)] [[PubMed](#)]
211. Nassar, R.; Brini, E.; Parui, S.; Liu, C.; Dignon, G.L.; Dill, K.A. Accelerating Protein Folding Molecular Dynamics Using Inter-Residue Distances from Machine Learning Servers. *J. Chem. Theory Comput.* **2022**, *18*, 1929–1935. [[CrossRef](#)]
212. Wayment-Steele, H.K.; Pande, V.S. Note: Variational encoding of protein dynamics benefits from maximizing latent autocorrelation. *J. Chem. Phys.* **2018**, *149*, 216101. [[CrossRef](#)] [[PubMed](#)]
213. Farmer, J.; Green, S.B.; Jacobs, D.J. Distribution of volume, microvoid percolation, and packing density in globular proteins. *arXiv* **2018**, arXiv:1810.08745.
214. Fried, S.D.; Boxer, S.G. Electric Fields and Enzyme Catalysis. *Annu. Rev. Biochem.* **2017**, *86*, 387–415. [[CrossRef](#)]
215. Jamasb, A.R.; Viñas, R.; Ma, E.J.; Harris, C.; Huang, K.; Hall, D.; Lió, P.; Blundell, T.L. Graphein—A Python Library for Geometric Deep Learning and Network Analysis on Protein Structures and Interaction Networks. *bioRxiv* **2021**. [[CrossRef](#)]
216. Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C.L.; Ma, J.; et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2016239118. [[CrossRef](#)]
217. Kawano, K.; Koide, S.; Imamura, C. Seq2seq Fingerprint with Byte-Pair Encoding for Predicting Changes in Protein Stability upon Single Point Mutation. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *17*, 1762–1772. [[CrossRef](#)]

218. Repecka, D.; Jauniskis, V.; Karpus, L.; Rembeza, E.; Rokaitis, I.; Zrimec, J.; Poviloniene, S.; Laurynenas, A.; Viknander, S.; Abuajwa, W.; et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nat. Mach. Intell.* **2021**, *3*, 324–333. [[CrossRef](#)]
219. Dauparas, J.; Anishchenko, I.; Bennett, N.; Bai, H.; Ragotte, R.J.; Milles, L.F.; Wicky, B.I.M.; Courbet, A.; de Haas, R.J.; Bethel, N.; et al. Robust deep learning based protein sequence design using ProteinMPNN. *bioRxiv* **2022**. [[CrossRef](#)]
220. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maroon, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A Generalist Agent. *arXiv* **2022**, arXiv:2205.06175.
221. Chen, J.; Zheng, S.; Zhao, H.; Yang, Y. Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map. *J. Cheminform.* **2021**, *13*, 7. [[CrossRef](#)] [[PubMed](#)]
222. Han, X.; Ning, W.; Ma, X.; Wang, X.; Zhou, K. Improving protein solubility and activity by introducing small peptide tags designed with machine learning models. *Metab. Eng. Commun.* **2020**, *11*, e00138. [[CrossRef](#)]
223. Chen, L.; Oughtred, R.; Berman, H.M.; Westbrook, J. TargetDB: A target registration database for structural genomics projects. *Bioinformatics* **2004**, *20*, 2860–2862. [[CrossRef](#)]
224. Madani, M.; Lin, K.; Tarakanova, A. DSResSol: A Sequence-Based Solubility Predictor Created with Dilated Squeeze Excitation Residual Networks. *Int. J. Mol. Sci.* **2021**, *22*, 3555. doi: 10.3390/ijms222413555. [[CrossRef](#)]
225. Cai, Z.; Luo, F.; Wang, Y.; Li, E.; Huang, Y. Protein pK (a) Prediction with Machine Learning. *ACS Omega* **2021**, *6*, 34823–34831. [[CrossRef](#)]
226. Ko, T.W.; Finkler, J.A.; Goedecker, S.; Behler, J. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **2021**, *12*, 398. [[CrossRef](#)]
227. Chatzigeorgoulas, A.; Cournia, Z. Predicting protein-membrane interfaces of peripheral membrane proteins using ensemble machine learning. *Briefings Bioinform.* **2022**, *23*, bbab518. [[CrossRef](#)]
228. Lai, P.K.; Fernando, A.; Cloutier, T.K.; Kingsbury, J.S.; Gokarn, Y.; Halloran, K.T.; Calero-Rubio, C.; Trout, B.L. Machine Learning Feature Selection for Predicting High Concentration Therapeutic Antibody Aggregation. *J. Pharm. Sci.* **2021**, *110*, 1583–1591. [[CrossRef](#)] [[PubMed](#)]
229. Li, G.; Qin, Y.; Fontaine, N.T.; Ng Fuk Chong, M.; Maria-Solano, M.A.; Feixas, F.; Cadet, X.F.; Pandjaitan, R.; Garcia-Borràs, M.; Cadet, F.; et al. Machine Learning Enables Selection of Epistatic Enzyme Mutants for Stability Against Unfolding and Detrimental Aggregation. *ChemBioChem* **2021**, *22*, 904–914. [[CrossRef](#)] [[PubMed](#)]
230. Li, F.; Li, C.; Wang, M.; Webb, G.I.; Zhang, Y.; Whisstock, J.C.; Song, J. GlycoMine: A machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. *Bioinformatics* **2015**, *31*, 1411–1419. [[CrossRef](#)] [[PubMed](#)]
231. Maiti, S.; Hassan, A.; Mitra, P. Boosting phosphorylation site prediction with sequence feature-based machine learning. *Proteins Struct. Funct. Bioinform.* **2020**, *88*, 284–291. [[CrossRef](#)]
232. Arnold, F.H. Protein engineering for unusual environments. *Curr. Opin. Biotechnol.* **1993**, *4*, 450–455. [[CrossRef](#)]
233. Prokop, M.; Damborský, J.; Koča, J. TRITON: In silico construction of protein mutants and prediction of their activities *. *Bioinformatics* **2000**, *16*, 845–846. [[CrossRef](#)]
234. Gilis, D.; Rooman, M. PoPMuSiC, an algorithm for predicting protein mutant stability changes. Application to prion proteins. *Protein Eng. Des. Sel.* **2000**, *13*, 849–856. [[CrossRef](#)]
235. Pasquier, C.; Hamodrakas, S. An hierarchical artificial neural network system for the classification of transmembrane proteins. *Protein Eng. Des. Sel.* **1999**, *12*, 631–634. [[CrossRef](#)]
236. Marvin, J.S.; Corcoran, E.E.; Hattangadi, N.A.; Zhang, J.V.; Gere, S.A.; Hellinga, H.W. The rational design of allosteric interactions in a monomeric protein and its applications to the construction of biosensors. *Proc. Natl. Acad. Sci. USA* **1997**, *94*, 4366–4371. [[CrossRef](#)]
237. Barany, F. Single-stranded hexameric linkers: A system for in-phase insertion mutagenesis and protein engineering. *Gene* **1985**, *37*, 111–123. [[CrossRef](#)]
238. Kawai, F.; Nakamura, A.; Visootsat, A.; Iino, R. Plasmid-Based One-Pot Saturation Mutagenesis and Robot-Based Automated Screening for Protein Engineering. *ACS Omega* **2018**, *3*, 7715–7726. [[CrossRef](#)] [[PubMed](#)]
239. Tsai, H.H.; Tsai, C.J.; Ma, B.; Nussinov, R. In silico protein design by combinatorial assembly of protein building blocks. *Protein Sci.* **2004**, *13*, 2753–2765. [[CrossRef](#)] [[PubMed](#)]
240. Mandell, D.J.; Kortemme, T. Backbone flexibility in computational protein design. *Curr. Opin. Biotechnol.* **2009**, *20*, 420–428. [[CrossRef](#)] [[PubMed](#)]
241. Lise, S.; Archambeau, C.; Pontil, M.; Jones, D.T. Prediction of hot spot residues at protein-protein interfaces by combining machine learning and energy-based methods. *BMC Bioinform.* **2009**, *10*, 365. [[CrossRef](#)] [[PubMed](#)]
242. Nikam, R.; Kulandaisamy, A.; Harini, K.; Sharma, D.; Gromiha, M.M. ProThermDB: Thermodynamic database for proteins and mutants revisited after 15 years. *Nucleic Acids Res.* **2021**, *49*, D420–D424. [[CrossRef](#)] [[PubMed](#)]
243. Jia, L.; Yarlagadda, R.; Reed, C.C. Structure Based Thermostability Prediction Models for Protein Single Point Mutations with Machine Learning Tools. *PLoS ONE* **2015**, *10*, e0138022. [[CrossRef](#)]
244. Cao, H.; Wang, J.; He, L.; Qi, Y.; Zhang, J.Z. DeepDDG: Predicting the Stability Change of Protein Point Mutations Using Neural Networks. *J. Chem. Inf. Model* **2019**, *59*, 1508–1514. [[CrossRef](#)] [[PubMed](#)]
245. Geng, C.; Vangone, A.; Folkers, G.E.; Xue, L.C.; Bonvin, A. iSEE: Interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations. *Proteins* **2019**, *87*, 110–119. [[CrossRef](#)]

246. Wang, J.; Lisanza, S.; Juergens, D.; Tischer, D.; Anishchenko, I.; Baek, M.; Watson, J.L.; Chun, J.H.; Milles, L.F.; Dauparas, J.; et al. Deep learning methods for designing proteins scaffolding functional sites. *bioRxiv* **2021**. [[CrossRef](#)]
247. Harteveld, Z.; Bonet, J.; Rosset, S.; Yang, C.; Sesterhenn, F.; Correia, B.E. A generic framework for hierarchical *de novo* protein design. *bioRxiv* **2022**. [[CrossRef](#)]
248. Cang, Z.; Wei, G.W. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.* **2017**, *13*, e1005690. [[CrossRef](#)] [[PubMed](#)]
249. Moffat, L.; Kandathil, S.M.; Jones, D.T. Design in the DARK: Learning Deep Generative Models for De Novo Protein Design. *bioRxiv* **2022**. [[CrossRef](#)]
250. Keskin, O.; Gursoy, A.; Ma, B.; Nussinov, R. Principles of protein–protein interactions: What are the preferred ways for proteins to interact? *Chem. Rev.* **2008**, *108*, 1225–1244. [[CrossRef](#)] [[PubMed](#)]
251. Chen, Z.; Zhao, P.; Li, F.; Marquez-Lago, T.T.; Leier, A.; Revote, J.; Zhu, Y.; Powell, D.R.; Akutsu, T.; Webb, G.I.; et al. iLearn: An integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Briefings Bioinform.* **2020**, *21*, 1047–1057. [[CrossRef](#)]
252. Wang, J.T.L.; Ma, Q.; Shasha, D.; Wu, C.H. New techniques for extracting features from protein sequences. *IBM Syst. J.* **2001**, *40*, 426–441. [[CrossRef](#)]
253. Singh, R.; Park, D.; Xu, J.; Hosur, R.; Berger, B. Struct2Net: A web service to predict protein–protein interactions using a structure-based approach. *Nucleic Acids Res.* **2010**, *38*, W508–W515. [[CrossRef](#)] [[PubMed](#)]
254. Hashemifar, S.; Neyshabur, B.; Khan, A.A.; Xu, J. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics* **2018**, *34*, i802–i810. [[CrossRef](#)]
255. Zhang, L.; Yu, G.; Xia, D.; Wang, J. Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing* **2019**, *324*, 10–19. [[CrossRef](#)]
256. Lei, H.; Wen, Y.; You, Z.; Elazab, A.; Tan, E.L.; Zhao, Y.; Lei, B. Protein–protein interactions prediction via multimodal deep polynomial network and regularized extreme learning machine. *IEEE J. Biomed. Health Inform.* **2018**, *23*, 1290–1303. [[CrossRef](#)]
257. Wang, L.; Wang, H.F.; Liu, S.R.; Yan, X.; Song, K.J. Predicting protein–protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. *Sci. Rep.* **2019**, *9*, 9848. [[CrossRef](#)]
258. Yang, F.; Fan, K.; Song, D.; Lin, H. Graph-based prediction of Protein–protein interactions with attributed signed graph embedding. *BMC Bioinform.* **2020**, *21*, 323. [[CrossRef](#)] [[PubMed](#)]
259. Li, F.; Zhu, F.; Ling, X.; Liu, Q. Protein interaction network reconstruction through ensemble deep learning with attention mechanism. *Front. Bioeng. Biotechnol.* **2020**, *8*, 390. [[CrossRef](#)] [[PubMed](#)]
260. Das, S.; Chakrabarti, S. Classification and prediction of protein–protein interaction interface using machine learning algorithm. *Sci. Rep.* **2021**, *11*, 1761. [[CrossRef](#)]
261. Lei, Y.; Li, S.; Liu, Z.; Wan, F.; Tian, T.; Li, S.; Zhao, D.; Zeng, J. A deep-learning framework for multi-level peptide–protein interaction prediction. *Nat. Commun.* **2021**, *12*, 5465. [[CrossRef](#)]
262. Balogh, O.M.; Benczik, B.; Horváth, A.; Pétervári, M.; Csermely, P.; Ferdinandy, P.; Ágg, B. Efficient link prediction in the protein–protein interaction network using topological information in a generative adversarial network machine learning model. *BMC Bioinform.* **2022**, *23*, 78. [[CrossRef](#)]
263. Song, B.; Luo, X.; Luo, X.; Liu, Y.; Niu, Z.; Zeng, X. Learning spatial structures of proteins improves protein–protein interaction prediction. *Briefings Bioinform.* **2022**, *23*, bbab558. [[CrossRef](#)]
264. Daberdaku, S.; Ferrari, C. Exploring the potential of 3D Zernike descriptors and SVM for protein–protein interface prediction. *BMC Bioinform.* **2018**, *19*, 35. [[CrossRef](#)]
265. Sanchez-Garcia, R.; Sorzano, C.O.S.; Carazo, J.M.; Segura, J. BIPSPI: A method for the prediction of partner-specific protein–protein interfaces. *Bioinformatics* **2019**, *35*, 470–477. [[CrossRef](#)]
266. Northey, T.C.; Barešić, A.; Martin, A.C. IntPred: A structure-based predictor of protein–protein interaction sites. *Bioinformatics* **2018**, *34*, 223–229. [[CrossRef](#)]
267. Gainza, P.; Sverrisson, F.; Monti, F.; Rodola, E.; Boscaini, D.; Bronstein, M.; Correia, B. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat. Methods* **2020**, *17*, 184–192. [[CrossRef](#)]
268. Yuan, Q.; Chen, J.; Zhao, H.; Zhou, Y.; Yang, Y. Structure-aware protein–protein interaction site prediction using deep graph convolutional network. *Bioinformatics* **2022**, *38*, 125–132. [[CrossRef](#)]
269. Tompa, P. Intrinsically disordered proteins: A 10-year recap. *Trends Biochem. Sci.* **2012**, *37*, 509–516. [[CrossRef](#)]
270. Dunker, A.K.; Silman, I.; Uversky, V.N.; Sussman, J.L. Function and structure of inherently disordered proteins. *Curr. Opin. Struct. Biol.* **2008**, *18*, 756–764. [[CrossRef](#)]
271. Uversky, V.N. Unusual biophysics of intrinsically disordered proteins. *Biochim. Biophys. Acta* **2013**, *1834*, 932–951. [[CrossRef](#)]
272. Uversky, V.N. Intrinsically disordered proteins and their “mysterious”(meta) physics. *Front. Phys.* **2019**, *7*, 10. [[CrossRef](#)]
273. Wright, P.E.; Dyson, H.J. Intrinsically unstructured proteins: Re-assessing the protein structure–function paradigm. *J. Mol. Biol.* **1999**, *293*, 321–331. [[CrossRef](#)]
274. Katuwawala, A.; Peng, Z.; Yang, J.; Kurgan, L. Computational Prediction of MoRFs, Short Disorder-to-order Transitioning Protein Binding Regions. *Comput. Struct. Biotechnol. J.* **2019**, *17*, 454–462. [[CrossRef](#)]
275. Zhao, B.; Kurgan, L. Surveying over 100 predictors of intrinsic disorder in proteins. *Expert Rev. Proteom.* **2021**, *18*, 1019–1029. [[CrossRef](#)]

276. Necci, M.; Piovesan, D.; Tosatto, S.C. Critical assessment of protein intrinsic disorder prediction. *Nat. Methods* **2021**, *18*, 472–481. [[CrossRef](#)]
277. Hatos, A.; Hajdu-Soltész, B.; Monzon, A.M.; Palopoli, N.; Álvarez, L.; Aykac-Fas, B.; Bassot, C.; Benítez, G.I.; Bevilacqua, M.; Chasapi, A.; et al. DisProt: Intrinsic protein disorder annotation in 2020. *Nucleic Acids Res.* **2020**, *48*, D269–D276. [[CrossRef](#)]
278. Malhis, N.; Jacobson, M.; Gsponer, J. MoRFchibi SYSTEM: Software tools for the identification of MoRFs in protein sequences. *Nucleic Acids Res.* **2016**, *44*, W488–W493. [[CrossRef](#)] [[PubMed](#)]
279. Wang, S.; Ma, J.; Xu, J. AUCpreD: Proteome-level protein disorder prediction by AUC-maximized deep convolutional neural fields. *Bioinformatics* **2016**, *32*, i672–i679. [[CrossRef](#)] [[PubMed](#)]
280. Sharma, R.; Kumar, S.; Tsunoda, T.; Patil, A.; Sharma, A. Predicting MoRFs in protein sequences using HMM profiles. *BMC Bioinform.* **2016**, *17*, 251–258. [[CrossRef](#)] [[PubMed](#)]
281. Hanson, J.; Yang, Y.; Paliwal, K.; Zhou, Y. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. *Bioinformatics* **2017**, *33*, 685–692. [[CrossRef](#)] [[PubMed](#)]
282. Sharma, R.; Bayarjargal, M.; Tsunoda, T.; Patil, A.; Sharma, A. MoRFPred-plus: Computational identification of MoRFs in protein sequences using physicochemical properties and HMM profiles. *J. Theor. Biol.* **2018**, *437*, 9–16. [[CrossRef](#)]
283. Sharma, R.; Sharma, A.; Raicar, G.; Tsunoda, T.; Patil, A. OPAL+: Length-specific MoRF prediction in intrinsically disordered protein sequences. *Proteomics* **2019**, *19*, 1800058. [[CrossRef](#)]
284. Mirabello, C.; Wallner, B. RAWMSA: End-to-end deep learning using raw multiple sequence alignments. *PLoS ONE* **2019**, *14*, e0220182.
285. Hanson, J.; Paliwal, K.K.; Litfin, T.; Zhou, Y. SPOT-Disorder2: Improved protein intrinsic disorder prediction by ensembled deep learning. *Genom. Proteom. Bioinform.* **2019**, *17*, 645–656. [[CrossRef](#)]
286. Dass, R.; Mulder, F.A.; Nielsen, J.T. ODINPred: Comprehensive prediction of protein order and disorder. *Sci. Rep.* **2020**, *10*, 14780. [[CrossRef](#)]
287. Tang, Y.J.; Pang, Y.H.; Liu, B. IDP-Seq2Seq: Identification of intrinsically disordered regions based on sequence to sequence learning. *Bioinformatics* **2020**, *36*, 5177–5186. [[CrossRef](#)]
288. Hu, G.; Katuwawala, A.; Wang, K.; Wu, Z.; Ghadermarzi, S.; Gao, J.; Kurgan, L. fIDPnn: Accurate intrinsic disorder prediction with putative propensities of disorder functions. *Nat. Commun.* **2021**, *12*, 4438. [[CrossRef](#)] [[PubMed](#)]
289. Liu, Y.; Wang, X.; Liu, B. RFPR-IDP: Reduce the false positive rates for intrinsically disordered protein and region prediction by incorporating both fully ordered proteins and disordered proteins. *Briefings Bioinform.* **2021**, *22*, 2000–2011. [[CrossRef](#)] [[PubMed](#)]
290. Emenecker, R.J.; Griffith, D.; Holehouse, A.S. Metapredict: A fast, accurate, and easy-to-use predictor of consensus disorder and structure. *Biophys. J.* **2021**, *120*, 4312–4319. [[CrossRef](#)]
291. Zhang, F.; Zhao, B.; Shi, W.; Li, M.; Kurgan, L. DeepDISOBind: Accurate prediction of RNA-, DNA- and protein-binding intrinsically disordered residues with deep multi-task learning. *Briefings Bioinform.* **2022**, *23*, bbab521. [[CrossRef](#)]
292. Li, H.; Pang, Y.; Liu, B.; Yu, L. MoRF-FUNCpred: Molecular Recognition Feature Function Prediction Based on Multi-Label Learning and Ensemble Learning. *Front. Pharmacol.* **2022**, *13*, 856417. [[CrossRef](#)] [[PubMed](#)]
293. Orlando, G.; Raimondi, D.; Codice, F.; Tabaro, F.; Vranken, W. Prediction of disordered regions in proteins with recurrent neural networks and protein dynamics. *J. Mol. Biol.* **2022**, *434*, 167579. [[CrossRef](#)]
294. Wilson, C.J.; Choy, W.Y.; Karttunen, M. AlphaFold2: A Role for Disordered Protein/Region Prediction? *Int. J. Mol. Sci.* **2022**, *23*, 4591. [[CrossRef](#)]
295. Sun, Z.; Liu, Q.; Qu, G.; Feng, Y.; Reetz, M.T. Utility of B-Factors in Protein Science: Interpreting Rigidity, Flexibility, and Internal Motion and Engineering Thermostability. *Chem. Rev.* **2019**, *119*, 1626–1665. [[CrossRef](#)]
296. Karplus, P.A.; Schulz, G.E. Prediction of chain flexibility in proteins. *Naturwissenschaften* **2005**, *72*, 212–213. [[CrossRef](#)]
297. Kuboniwa, H.; Tjandra, N.; Grzesiek, S.; Ren, H.; Klee, C.B.; Bax, A. Solution structure of calcium-free calmodulin. *Nat. Struct. Biol.* **1995**, *2*, 768–776. [[CrossRef](#)]
298. Yun, C.H.; Bai, J.; Sun, D.Y.; Cui, D.F.; Chang, W.R.; Liang, D.C. Structure of potato calmodulin PCM6: The first report of the three-dimensional structure of a plant calmodulin. *Acta Crystallogr. D Biol. Crystallogr.* **2004**, *60*, 1214–1219. [[CrossRef](#)] [[PubMed](#)]
299. Vertessy, B.G.; Harmat, V.; Bocskei, Z.; Naray-Szabo, G.; Orosz, F.; Ovadi, J. Simultaneous binding of drugs with different chemical structures to Ca²⁺-calmodulin: Crystallographic and spectroscopic studies. *Biochemistry* **1998**, *37*, 15300–15310. [[CrossRef](#)]
300. Komeiji, Y.; Ueno, Y.; Uebayasi, M. Molecular dynamics simulations revealed Ca(2+)-dependent conformational change of Calmodulin. *FEBS Lett.* **2002**, *521*, 133–139. [[CrossRef](#)]
301. Fonze, E.; Charlier, P.; To'th, Y.; Vermeire, M.; Raquet, X.; Dubus, A.; Frere, J.M. TEM1 beta-lactamase structure solved by molecular replacement and refined structure of the S235A mutant. *Acta Crystallogr. D Biol. Crystallogr.* **1995**, *51*, 682–694. [[CrossRef](#)] [[PubMed](#)]
302. Avery, C.; Baker, L.; Jacobs, D.J. Functional Dynamics of Substrate Recognition in TEM Beta-Lactamase. *Entropy* **2022**, *24*. [[CrossRef](#)]
303. Hsiao, C.D.; Sun, Y.J.; Rose, J.; Wang, B.C. The crystal structure of glutamine-binding protein from Escherichia coli. *J. Mol. Biol.* **1996**, *262*, 225–242. [[CrossRef](#)] [[PubMed](#)]
304. Baker, L.J. Do Dynamic Allosteric Effects Occur in IGG4 Antibodies? Ph.D. Thesis, The University of North Carolina at Charlotte, Charlotte, NC, USA, 2020.
305. Carugo, O.; Argos, P. Protein—Protein crystal-packing contacts. *Protein Sci.* **1997**, *6*, 2261–2263. [[CrossRef](#)] [[PubMed](#)]

306. Berjanskii, M.V.; Wishart, D.S. Application of the random coil index to studying protein flexibility. *J. Biomol. NMR* **2008**, *40*, 31–48. [[CrossRef](#)] [[PubMed](#)]
307. Livesay, D.R.; Huynh, D.H.; Dallakyan, S.; Jacobs, D.J. Hydrogen bond networks determine emergent mechanical and thermodynamic properties across a protein family. *Chem. Cent. J.* **2008**, *2*, 17. [[CrossRef](#)]
308. Li, T.; Tracka, M.B.; Uddin, S.; Casas-Finet, J.; Jacobs, D.J.; Livesay, D.R. Redistribution of flexibility in stabilizing antibody fragment mutants follows Le Châtelier's principle. *PLoS ONE* **2014**, *9*, e92870. [[CrossRef](#)]
309. Atilgan, A.R.; Durell, S.R.; Jernigan, R.L.; Demirel, M.C.; Keskin, O.; Bahar, I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys. J.* **2001**, *80*, 505–515. [[CrossRef](#)]
310. Xia, K.; Opron, K.; Wei, G.W. Multiscale multiphysics and multidomain models—flexibility and rigidity. *J. Chem. Phys.* **2013**, *139*, 194109. [[CrossRef](#)]
311. Opron, K.; Xia, K.; Wei, G.W. Fast and anisotropic flexibility-rigidity index for protein flexibility and fluctuation analysis. *J. Chem. Phys.* **2014**, *140*, 234105. [[CrossRef](#)]
312. Bramer, D.; Wei, G.W. Blind prediction of protein B-factor and flexibility. *J. Chem. Phys.* **2018**, *149*, 134107. [[CrossRef](#)] [[PubMed](#)]
313. Trott, O.; Siggers, K.; Rost, B.; Palmer, A.G., 3rd. Protein conformational flexibility prediction using machine learning. *J. Magn. Reson.* **2008**, *192*, 37–47. [[CrossRef](#)] [[PubMed](#)]
314. Chen, M.; Ludtke, S.J. Deep learning-based mixed-dimensional Gaussian mixture model for characterizing variability in cryo-EM. *Nat. Methods* **2021**, *18*, 930–936. [[CrossRef](#)]
315. Nembrini, S.; König, I.R.; Wright, M.N. The revival of the Gini importance? *Bioinformatics* **2018**, *34*, 3711–3718. [[CrossRef](#)]
316. Grisci, B.; Dorn, M. NEAT-FLEX: Predicting the conformational flexibility of amino acids using neuroevolution of augmenting topologies. *J. Bioinform. Comput. Biol.* **2017**, *15*, 1750009. [[CrossRef](#)]
317. Spiwok, V.; Kriz, P. Time-Lagged t-Distributed Stochastic Neighbor Embedding (t-SNE) of Molecular Simulation Trajectories. *Front. Mol. Biosci.* **2020**, *7*, 132. [[CrossRef](#)]
318. Grear, T.; Avery, C.; Patterson, J.; Jacobs, D.J. Molecular function recognition by supervised projection pursuit machine learning. *Sci. Rep.* **2021**, *11*, 4247. doi: 10.1038/s41598-021-83269-y. [[CrossRef](#)]
319. Patterson, J.; Grear, T.; Jacobs, D.J. Biased Hypothesis Formation From Projection Pursuit 2021. *Adv. Artif. Intell. Mach. Learn.* **2021**, *3*.
320. Zheng, W. Predicting cryptic ligand binding sites based on normal modes guided conformational sampling. *Proteins* **2021**, *89*, 416–426. [[CrossRef](#)] [[PubMed](#)]
321. Degiacomi, M.T. Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space. *Structure* **2019**, *27*, 1034–1040.e3. [[CrossRef](#)] [[PubMed](#)]
322. Tian, H.; Jiang, X.; Trozzi, F.; Xiao, S.; Larson, E.C.; Tao, P. Explore Protein Conformational Space With Variational Autoencoder. *Front. Mol. Biosci.* **2021**, *8*, 781635. [[CrossRef](#)] [[PubMed](#)]
323. Romero, R.; Ramanathan, A.; Yuen, T.; Bhowmik, D.; Mathew, M.; Munshi, L.B.; Javaid, S.; Bloch, M.; Lizneva, D.; Rahimova, A.; et al. Mechanism of glucocerebrosidase activation and dysfunction in Gaucher disease unraveled by molecular dynamics and deep learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 5086–5095. [[CrossRef](#)] [[PubMed](#)]
324. Sun, M.G.F.; Kim, P.M. Data driven flexible backbone protein design. *PLoS Comput. Biol.* **2017**, *13*, e1005722. [[CrossRef](#)]
325. Monzon, A.M.; Rohr, C.O.; Fornasari, M.S.; Parisi, G. CoDNaS 2.0: A comprehensive database of protein conformational diversity in the native state. *Database* **2016**, *2016*, baw038. [[CrossRef](#)]
326. Srivastava, A.; Malgorzata; Uddin, S.; Casas-Finet, J.; Livesay, D.R.; Jacobs, D.J. Mutations in Antibody Fragments Modulate Allosteric Response Via Hydrogen-Bond Network Fluctuations. *Biophys. J.* **2016**, *110*, 1933–1942. [[CrossRef](#)]
327. Guo, J.; Zhou, H.X. Protein Allostery and Conformational Dynamics. *Chem. Rev.* **2016**, *116*, 6503–6515. [[CrossRef](#)]
328. Liu, J.; Nussinov, R. Allostery: An Overview of Its History, Concepts, Methods, and Applications. *PLoS Comput. Biol.* **2016**, *12*, e1004966. [[CrossRef](#)]
329. Perutz, M.F.; Fermi, G.; Luisi, B.; Shaanan, B.; Liddington, R.C. Stereochemistry of cooperative mechanisms in hemoglobin. *Accounts Chem. Res.* **1987**, *20*, 309–321. [[CrossRef](#)]
330. Nussinov, R. Introduction to Protein Ensembles and Allostery. *Chem. Rev.* **2016**, *116*, 6263–6266. [[CrossRef](#)] [[PubMed](#)]
331. Gunasekaran, K.; Ma, B.; Nussinov, R. Is allostery an intrinsic property of all dynamic proteins? *Proteins Struct. Funct. Bioinform.* **2004**, *57*, 433–443. [[CrossRef](#)]
332. Istomin, A.Y.; Gromiha, M.M.; Vorov, O.K.; Jacobs, D.J.; Livesay, D.R. New insight into long-range nonadditivity within protein double-mutant cycles. *Proteins Struct. Funct. Bioinform.* **2008**, *70*, 915–924. [[CrossRef](#)]
333. Skjaerven, L.; Hollup, S.M.; Reuter, N. Normal mode analysis for proteins. *J. Mol. Struct. THEOCHEM* **2009**, *898*, 42–48. [[CrossRef](#)]
334. Tama, F.; Sanejouand, Y.H. Conformational change of proteins arising from normal mode calculations. *Protein Eng. Des. Sel.* **2001**, *14*, 1–6. [[CrossRef](#)]
335. Hayward, S.; Kitao, A.; Berendsen, H.J. Model-free methods of analyzing domain motions in proteins from simulation: A comparison of normal mode analysis and molecular dynamics simulation of lysozyme. *Proteins Struct. Funct. Bioinform.* **1997**, *27*, 425–437. [[CrossRef](#)]
336. Bakan, A.; Meireles, L.M.; Bahar, I. ProDy: Protein Dynamics Inferred from Theory and Experiments. *Bioinformatics* **2011**, *27*, 1575–1577. [[CrossRef](#)]

337. Wells, S.; Menor, S.; Hespeneide, B.; Thorpe, M.F. Constrained geometric simulation of diffusive motion in proteins. *Phys. Biol.* **2005**, *2*, S127–S136. [[CrossRef](#)]
338. Ma, B.; Tsai, C.J.; Haliloğlu, T.; Nussinov, R. Dynamic Allostery: Linkers Are Not Merely Flexible. *Structure* **2011**, *19*, 907–917. doi: 10.1016/j.str.2011.06.002. [[CrossRef](#)]
339. Pandey, R.B.; Jacobs, D.J.; Farmer, B.L. Preferential binding effects on protein structure and dynamics revealed by coarse-grained Monte Carlo simulation. *J. Chem. Phys.* **2017**, *146*, 195101. [[CrossRef](#)] [[PubMed](#)]
340. Ferraro, M.; Moroni, E.; Ippoliti, E.; Rinaldi, S.; Sanchez-Martin, C.; Rasola, A.; Pavarino, L.F.; Colombo, G. Machine Learning of Allosteric Effects: The Analysis of Ligand-Induced Dynamics to Predict Functional Effects in TRAP1. *J. Phys. Chem. B* **2021**, *125*, 101–114. [[CrossRef](#)] [[PubMed](#)]
341. Marchetti, F.; Moroni, E.; Pandini, A.; Colombo, G. Machine Learning Prediction of Allosteric Drug Activity from Molecular Dynamics. *J. Phys. Chem. Lett.* **2021**, *12*, 3724–3732. [[CrossRef](#)] [[PubMed](#)]
342. Zhu, J.; Wang, J.; Han, W.; Xu, D. Neural relational inference to learn long-range allosteric interactions in proteins from molecular dynamics simulations. *Nat. Commun.* **2022**, *13*, 1661. [[CrossRef](#)]
343. Tian, H.; Jiang, X.; Tao, P. PASSer: Prediction of allosteric sites server. *Mach. Learn. Sci. Technol.* **2021**, *2*, 035015. [[CrossRef](#)]
344. Vishweshwaraiah, Y.L.; Chen, J.; Dokholyan, N.V. Engineering an Allosteric Control of Protein Function. *J. Phys. Chem. B* **2021**, *125*, 1806–1814. [[CrossRef](#)]
345. Gorman, S.D.; D’Amico, R.N.; Winston, D.S.; Boehr, D.D. Engineering Allostery into Proteins. *Adv. Exp. Med. Biol.* **2019**, *1163*, 359–384. [[CrossRef](#)]
346. Quijano-Rubio, A.; Yeh, H.W.; Park, J.; Lee, H.; Langan, R.A.; Boyken, S.E.; Lajoie, M.J.; Cao, L.; Chow, C.M.; Miranda, M.C.; et al. De novo design of modular and tunable protein biosensors. *Nature* **2021**, *591*, 482–487. [[CrossRef](#)]
347. Unke, O.T.; Chmiela, S.; Sauceda, H.E.; Gastegger, M.; Poltavsky, I.; Schütt, K.T.; Tkatchenko, A.; Müller, K.R. Machine Learning Force Fields. *Chem. Rev.* **2021**, *121*, 10142–10186. [[CrossRef](#)]
348. Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016**, *145*, 170901. [[CrossRef](#)]
349. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106. [[CrossRef](#)] [[PubMed](#)]
350. Gastegger, M.; Schwiedrzik, L.; Bittermann, M.; Berzsenyi, F.; Marquetand, P. wACSF-Weighted atom-centered symmetry functions as descriptors in machine learning potentials. *J. Chem. Phys.* **2018**, *148*, 241709. [[CrossRef](#)] [[PubMed](#)]
351. Bartok, A.P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B* **2013**, *87*, 184115. [[CrossRef](#)]
352. Bartok, A.P.; Payne, M.C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403. [[CrossRef](#)]
353. Csányi, G.; Winfield, S.; Kermode, J.R.; De Vita, A.; Comisso, A.; Bernstein, N.; Payne, M.C. Expressive Programming for Computational Physics in Fortran 95+. In *IoP Computational Physics Group Newsletter*; Springer: Berlin/Heidelberg, Germany, 2007.
354. Sumpter, B.G.; Noid, D.W. Potential energy surfaces for macromolecules. A neural network technique. *Chem. Phys. Lett.* **1992**, *192*, 455–462. [[CrossRef](#)]
355. Blank, T.B.; Brown, S.D.; Calhoun, A.W.; Doren, D.J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **1995**, *103*, 4129–4137. [[CrossRef](#)]
356. Prudente, F.V.; Acioli, P.H.; Neto, J.S. The fitting of potential energy surfaces using neural networks: Application to the study of vibrational levels of H₃⁺. *J. Chem. Phys.* **1998**, *109*, 8801–8808. [[CrossRef](#)]
357. Hunger, J.; Huttner, G. Optimization and analysis of force field parameters by combination of genetic algorithms and neural networks. *J. Comput. Chem.* **1999**, *20*, 455–471. [[CrossRef](#)]
358. Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* **2004**, *395*, 210–215. [[CrossRef](#)]
359. Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401. [[CrossRef](#)]
360. Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quantum Chem.* **2015**, *115*, 1032–1050. [[CrossRef](#)]
361. Unke, O.T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678–3693. [[CrossRef](#)] [[PubMed](#)]
362. Schütt, K.; Kindermans, P.J.; Sauceda Felix, H.E.; Chmiela, S.; Tkatchenko, A.; Müller, K.R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
363. Schütt, K.T.; Sauceda, H.E.; Kindermans, P.J.; Tkatchenko, A.; Müller, K.R. SchNet—A deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722. [[CrossRef](#)]
364. Schütt, K.T.; Kessel, P.; Gastegger, M.; Nicoli, K.A.; Tkatchenko, A.; Müller, K.R. SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *J. Chem. Theory Comput.* **2019**, *15*, 448–455. [[CrossRef](#)]
365. Gasteiger, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26 April–1 May 2020.

366. Park, C.W.; Kornbluth, M.; Vandermause, J.; Wolverson, C.; Kozinsky, B.; Mailoa, J.P. Accurate and scalable graph neural network force field and molecular dynamics with direct force architecture. *NPJ Comput. Mater.* **2021**, *7*, 73. [[CrossRef](#)]
367. Haghghatlari, M.; Li, J.; Guan, X.; Zhang, O.; Das, A.; Stein, C.J.; Heidar-Zadeh, F.; Liu, M.; Head-Gordon, M.; Bertels, L.; et al. NewtonNet: A Newtonian message passing network for deep learning of interatomic potentials and forces. *Digit Discov.* **2022**, *1*, 333–343. [[CrossRef](#)]
368. Doerr, S.; Majewski, M.; Pérez, A.; Kramer, A.; Clementi, C.; Noe, F.; Giorgino, T.; De Fabritiis, G. Torchmd: A deep learning framework for molecular simulations. *J. Chem. Theory Comput.* **2021**, *17*, 2355–2363. [[CrossRef](#)]
369. Wang, H.; Zhang, L.; Han, J.; E, W. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Comput. Phys. Commun.* **2018**, *228*, 178–184. [[CrossRef](#)]
370. Sinha, S.; Vohora, D. Drug discovery and development: An overview. *Pharm. Med. Transl. Clin. Res.* **2018**, 19–32. [[CrossRef](#)]
371. Shen, C.; Ding, J.; Wang, Z.; Cao, D.; Ding, X.; Hou, T. From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020**, *10*, e1429. [[CrossRef](#)]
372. Lavecchia, A. Machine-learning approaches in drug discovery: Methods and applications. *Drug Discov. Today* **2015**, *20*, 318–331. [[CrossRef](#)]
373. Lo, Y.C.; Rensi, S.E.; Torng, W.; Altman, R.B. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **2018**, *23*, 1538–1546. [[CrossRef](#)]
374. Zhang, L.; Tan, J.; Han, D.; Zhu, H. From machine learning to deep learning: Progress in machine intelligence for rational drug discovery. *Drug Discov. Today* **2017**, *22*, 1680–1685. [[CrossRef](#)]
375. Ghasemi, F.; Mehridehnavi, A.; Pérez-Garrido, A.; Pérez-Sánchez, H. Neural network and deep-learning algorithms used in QSAR studies: Merits and drawbacks. *Drug Discov. Today* **2018**, *23*, 1784–1790. [[CrossRef](#)]
376. Rifaioglu, A.S.; Atas, H.; Martin, M.J.; Cetin-Atalay, R.; Atalay, V.; Doğan, T. Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases. *Briefings Bioinform.* **2019**, *20*, 1878–1912. [[CrossRef](#)]
377. Jing, Y.; Bian, Y.; Hu, Z.; Wang, L.; Xie, X.Q.S. Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *AAPS J.* **2018**, *20*, 58. [[CrossRef](#)]
378. Dana, D.; Gadhiya, S.V.; St. Surin, L.G.; Li, D.; Naaz, F.; Ali, Q.; Paka, L.; Yamin, M.A.; Narayan, M.; Goldberg, I.D.; et al. Deep learning in drug discovery and medicine; scratching the surface. *Molecules* **2018**, *23*, 2384. [[CrossRef](#)]
379. Mouchlis, V.D.; Afantitis, A.; Serra, A.; Fratello, M.; Papadiamantis, A.G.; Aidinis, V.; Lynch, I.; Greco, D.; Melagraki, G. Advances in de novo drug design: From conventional to machine learning methods. *Int. J. Mol. Sci.* **2021**, *22*, 1676. [[CrossRef](#)]
380. Peña-Guerrero, J.; Nguewa, P.A.; García-Sosa, A.T. Machine learning, artificial intelligence, and data science breaking into drug design and neglected diseases. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2021**, *11*, e1513. [[CrossRef](#)]
381. Maia, E.H.B.; Assis, L.C.; De Oliveira, T.A.; Da Silva, A.M.; Taranto, A.G. Structure-based virtual screening: From classical to artificial intelligence. *Front. Chem.* **2020**, *8*, 343. [[CrossRef](#)]
382. Sunny, S.; Jayaraj, P. Protein–protein docking: Past, present, and future. *Protein J.* **2022**, *41*, 1–26. [[CrossRef](#)]
383. Crampon, K.; Giorkallos, A.; Deldossi, M.; Baud, S.; Steffanel, L.A. Machine-learning methods for ligand–protein molecular docking. *Drug Discov. Today* **2021**, *27*, 151–164. [[CrossRef](#)]
384. Eberhardt, J.; Santos-Martins, D.; Tillack, A.F.; Forli, S. AutoDock Vina 1.2. 0: New docking methods, expanded force field, and python bindings. *J. Chem. Inf. Model.* **2021**, *61*, 3891–3898. [[CrossRef](#)]
385. Baum, B.; Muley, L.; Smolinski, M.; Heine, A.; Hangauer, D.; Klebe, G. Non-additivity of functional group contributions in protein–ligand binding: A comprehensive study by crystallography and isothermal titration calorimetry. *J. Mol. Biol.* **2010**, *397*, 1042–1054. [[CrossRef](#)]
386. Wang, C.; Zhang, Y. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *J. Comput. Chem.* **2017**, *38*, 169–177. [[CrossRef](#)]
387. Guedes, I.A.; Barreto, A.; Marinho, D.; Krempser, E.; Kuenemann, M.A.; Sperandio, O.; Dardenne, L.E.; Miteva, M.A. New machine learning and physics-based scoring functions for drug discovery. *Sci. Rep.* **2021**, *11*, 3198. [[CrossRef](#)]
388. Wang, X.; Terashi, G.; Christoffer, C.W.; Zhu, M.; Kihara, D. Protein docking model evaluation by 3D deep convolutional neural networks. *Bioinformatics* **2020**, *36*, 2113–2118. [[CrossRef](#)]
389. Yang, L.; Yang, G.; Chen, X.; Yang, Q.; Yao, X.; Bing, Z.; Niu, Y.; Huang, L.; Yang, L. Deep scoring neural network replacing the scoring function components to improve the performance of structure-based molecular docking. *ACS Chem. Neurosci.* **2021**, *12*, 2133–2142. [[CrossRef](#)]
390. Xie, Z.; Deng, X.; Shu, K. Prediction of protein–protein interaction sites using convolutional neural network and improved data sets. *Int. J. Mol. Sci.* **2020**, *21*, 467. [[CrossRef](#)]
391. Townshend, R.; Bedi, R.; Suriana, P.; Dror, R. End-to-end learning on 3d protein structure for interface prediction. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, CO, Canada, 8–14 December 2019; Volume 32.
392. Zhu, H.; Du, X.; Yao, Y. ConvPPIS: Identifying protein–protein interaction sites by an ensemble convolutional neural network with feature graph. *Curr. Bioinform.* **2020**, *15*, 368–378. [[CrossRef](#)]
393. Liu, Y.; Yuan, H.; Cai, L.; Ji, S. Deep learning of high-order interactions for protein interface prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 679–687.

394. Fout, A.; Byrd, J.; Shariat, B.; Ben-Hur, A. Protein interface prediction using graph convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
395. Cao, Y.; Shen, Y. Energy-based graph convolutional networks for scoring protein docking models. *Proteins Struct. Funct. Bioinform.* **2020**, *88*, 1091–1099. [CrossRef]
396. Wang, X.; Flannery, S.T.; Kihara, D. Protein docking model evaluation by graph neural networks. *Front. Mol. Biosci.* **2021**, *8*, 647915. [CrossRef]
397. Ramaswamy, V.K.; Musson, S.C.; Willcocks, C.G.; Degiacomi, M.T. Deep learning protein conformational space with convolutions and latent interpolations. *Phys. Rev. X* **2021**, *11*, 011052. [CrossRef]
398. Nguyen, D.D.; Gao, K.; Wang, M.; Wei, G.W. MathDL: Mathematical deep learning for D3R Grand Challenge 4. *J. Comput.-Aided Mol. Des.* **2020**, *34*, 131–147. [CrossRef]
399. Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2323–2332.
400. Adeshina, Y.O.; Deeds, E.J.; Karanicolos, J. Machine learning classification can reduce false positives in structure-based virtual screening. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 18477–18488. [CrossRef]
401. Schreiber, G. Protein–Protein Interaction Interfaces and Their Functional Implications. Protein–Protein Interaction Regulators. 2020. Available online: <https://pubs.rsc.org/en/content/chapterhtml/2020/bk9781788011877-00001?isbn=978-1-78801-187-7&sercode=bk> (accessed on 15 July 2022).
402. Fan, J.; Fu, A.; Zhang, L. Progress in molecular docking. *Quant. Biol.* **2019**, *7*, 83–89. [CrossRef]
403. Yang, J.; Roy, A.; Zhang, Y. BioLiP: A semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic Acids Res* **2013**, *41*, D1096–1103. [CrossRef]
404. Smith, R.D.; Clark, J.J.; Ahmed, A.; Orban, Z.J.; Dunbar, J.B., Jr.; Carlson, H.A. Updates to Binding MOAD (Mother of All Databases): Polypharmacology Tools and Their Utility in Drug Repurposing. *J. Mol. Biol.* **2019**, *431*, 2423–2433. [CrossRef]
405. Gilson, M.K.; Liu, T.; Baitaluk, M.; Nicola, G.; Hwang, L.; Chong, J. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* **2015**, *44*, D1045–D1053. [CrossRef]
406. Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model* **2019**, *59*, 895–913. [CrossRef]
407. Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Acc. Chem. Res.* **2017**, *50*, 302–309. [CrossRef]
408. Ballester, P.J.; Mitchell, J.B. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, *26*, 1169–1175. [CrossRef]
409. Li, H.; Leung, K.S.; Wong, M.H.; Ballester, P.J. Improving AutoDock Vina Using Random Forest: The Growing Accuracy of Binding Affinity Prediction by the Effective Exploitation of Larger Data Sets. *Mol. Inform.* **2015**, *34*, 115–126. [CrossRef]
410. Li, H.; Leung, K.S.; Wong, M.H.; Ballester, P.J. Substituting random forest for multiple linear regression improves binding affinity prediction of scoring functions: Cyscore as a case study. *BMC Bioinform.* **2014**, *15*, 291. [CrossRef]
411. Ashtawy, H.M.; Mahapatra, N.R. A Comparative Assessment of Predictive Accuracies of Conventional and Machine Learning Scoring Functions for Protein-Ligand Binding Affinity Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2015**, *12*, 335–347. [CrossRef]
412. Shar, P.A.; Tao, W.; Gao, S.; Huang, C.; Li, B.; Zhang, W.; Shahen, M.; Zheng, C.; Bai, Y.; Wang, Y. Pred-binding: Large-scale protein-ligand binding affinity prediction. *J. Enzym. Inhib. Med. Chem.* **2016**, *31*, 1443–1450. [CrossRef]
413. Jover, J.; Bosque, R.; Sales, J. Quantitative structure-property relationship estimation of cation binding affinity of the common amino acids. *J. Phys. Chem. A* **2009**, *113*, 3703–3708. [CrossRef]
414. Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D.R. Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model* **2017**, *57*, 942–957. [CrossRef]
415. Jimenez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model* **2018**, *58*, 287–296. [CrossRef]
416. Stepniewska-Dziubinska, M.M.; Zielenkiewicz, P.; Siedlecki, P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. *Bioinformatics* **2018**, *34*, 3666–3674. [CrossRef]
417. Li, Y.; Rezaei, M.A.; Li, C.; Li, X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. In Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 18–21 November 2019; pp. 303–310. [CrossRef]
418. Zhang, H.; Liao, L.; Saravanan, K.M.; Yin, P.; Wei, Y. DeepBindRG: A deep learning based method for estimating effective protein-ligand affinity. *PeerJ* **2019**, *7*, e7362. [CrossRef]
419. Zheng, L.; Fan, J.; Mu, Y. OnionNet: A Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein-Ligand Binding Affinity Prediction. *ACS Omega* **2019**, *4*, 15956–15965. [CrossRef]
420. Wang, S.; Liu, D.; Ding, M.; Du, Z.; Zhong, Y.; Song, T.; Zhu, J.; Zhao, R. SE-OnionNet: A Convolution Neural Network for Protein-Ligand Binding Affinity Prediction. *Front. Genet.* **2020**, *11*, 607824. [CrossRef]
421. Ozturk, H.; Ozgur, A.; Ozkirimli, E. DeepDTA: Deep drug-target binding affinity prediction. *Bioinformatics* **2018**, *34*, i821–i829. [CrossRef]

422. Zhao, L.; Wang, J.; Pang, L.; Liu, Y.; Zhang, J. GANsDTA: Predicting Drug-Target Binding Affinity Using GANs. *Front. Genet.* **2019**, *10*, 1243. [[CrossRef](#)]
423. Zhao, Q.; Duan, G.; Yang, M.; Cheng, Z.; Li, Y.; Wang, J. AttentionDTA: Drug-target binding affinity prediction by sequence-based deep learning with attention mechanism. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**. [[CrossRef](#)]
424. Nguyen, T.; Le, H.; Quinn, T.P.; Nguyen, T.; Le, T.D.; Venkatesh, S. GraphDTA: Predicting drug-target binding affinity with graph neural networks. *Bioinformatics* **2021**, *37*, 1140–1147. [[CrossRef](#)]
425. Son, J.; Kim, D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. *PLoS ONE* **2021**, *16*, e0249404. [[CrossRef](#)]
426. Jankauskaite, J.; Jimenez-Garcia, B.; Dapkunas, J.; Fernandez-Recio, J.; Moal, I.H. SKEMPI 2.0: An updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **2019**, *35*, 462–469. [[CrossRef](#)]
427. Schymkowitz, J.; Borg, J.; Stricher, F.; Nys, R.; Rousseau, F.; Serrano, L. The FoldX web server: An online force field. *Nucleic Acids Res.* **2005**, *33*, W382–W388. [[CrossRef](#)]
428. Benedix, A.; Becker, C.M.; de Groot, B.L.; Caflisch, A.; Bockmann, R.A. Predicting free energy changes using structural ensembles. *Nat. Methods* **2009**, *6*, 3–4. doi: 10.1038/nmeth0109-3. [[CrossRef](#)]
429. Dehouck, Y.; Kwasigroch, J.M.; Rooman, M.; Gilis, D. BeAtMuSiC: Prediction of changes in protein-protein binding affinity on mutations. *Nucleic Acids Res.* **2013**, *41*, W333–W339. [[CrossRef](#)]
430. Xiong, P.; Zhang, C.; Zheng, W.; Zhang, Y. BindProfX: Assessing Mutation-Induced Binding Affinity Change by Protein Interface Profiles with Pseudo-Counts. *J. Mol. Biol.* **2017**, *429*, 426–434. [[CrossRef](#)] [[PubMed](#)]
431. Pires, D.E.; Ascher, D.B.; Blundell, T.L. mCSM: Predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics* **2014**, *30*, 335–342. [[CrossRef](#)]
432. Rodrigues, C.H.M.; Myung, Y.; Pires, D.E.V.; Ascher, D.B. mCSM-PPI2: Predicting the effects of mutations on protein-protein interactions. *Nucleic Acids Res.* **2019**, *47*, W338–W344. [[CrossRef](#)] [[PubMed](#)]
433. Timasheff, S.N. Protein-solvent preferential interactions, protein hydration, and the modulation of biochemical reactions by solvent components. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 9721–9726. [[CrossRef](#)] [[PubMed](#)]
434. Ferreon, A.C.M.; Ferreon, J.C.; Bolen, D.W.; Rösger, J. Protein Phase Diagrams II: Nonideal Behavior of Biochemical Reactions in the Presence of Osmolytes. *Biophys. J.* **2007**, *92*, 245–256. [[CrossRef](#)]
435. Duff, M.R.; Howell, E.E. Thermodynamics and solvent linkage of macromolecule–ligand interactions. *Methods* **2015**, *76*, 51–60. doi: 10.1016/j.ymeth.2014.11.009. [[CrossRef](#)]
436. Völker, J.; Breslauer, K.J. Communication between noncontacting macromolecules. *Annu. Rev. Biophys. Biomol. Struct.* **2005**, *34*, 21–42. [[CrossRef](#)]