The Protein Journal (2022) 41:1–26
https://doi.org/10.1007/s10930-021-10031-8

Protein–Protein Docking: Past, Present, and Future

Sharon Sunny[1] · P. B. Jayaraj[1]

Accepted: 1 November 2021 / Published online: 17 November 2021

Abstract
The biological significance of proteins attracted the scientific community in exploring their characteristics. The studies shed light on the interaction patterns and functions of proteins in a living body. Due to their practical difficulties, reliable experimental techniques pave the way for introducing computational methods in the interaction prediction. Automated methods reduced the difficulties but could not yet replace experimental studies as the field is still evolving. Interaction prediction problem being critical needs highly accurate results, but none of the existing methods could offer reliable performance that can parallel with experimental results yet. This article aims to assess the existing computational docking algorithms, their challenges, and future scope. Blind docking techniques are quite helpful when no information other than the individual structures are available. As more and more complex structures are being added to different databases, information-driven approaches can be a good alternative. Artificial intelligence, ruling over the major fields, is expected to take over this domain very shortly.

Keywords Protein–protein docking · Deep learning · Artificial intelligence · Nature-inspired algorithms · Geometric algorithms · Searching and scoring

1 Introduction

Proteins are essential biological molecules whose interactions are at the pivot of many biological systems, including the immune and nervous systems. The interactions can happen between proteins of any size, i.e., any protein oligomer is a potential candidate for interaction. Their interactions are very specific, and hence any mutation at interaction interfaces are more threatening than at the non-interface regions [1]. Interfacial mutations are reported to trigger diseases like cancer [2]. Aggregation of proteins causes diseases like Alzheimer's disease and Parkinson's disease; a cure is not yet found for these diseases. This points to the importance of interaction studies on proteins. Structural biology gains popularity in studying protein interactions as their function and structure are highly interrelated. Along with the advent of experimental techniques like X-ray crystallography, NMR (Nuclear Magnetic Resonance) spectroscopy, and cryo-EM (Electron Microscopy) have come radical changes in interaction studies. X-ray crystallography proved simple and cheap and can generate high-resolution structures, while NMR can accommodate interaction dynamics. On the negative side, the former demands crystallizable structures and hence cannot account for the possible conformational changes, and the later procedure describes the structures in terms of observed distances which makes it difficult to use in large proteins. The prediction of protein structure greater than 50 kDa is nearly impossible using NMR. With the introduction of cryo-EM, better quality structures could be obtained even for large molecules. Both hardware and software improvements contribute to its better performance. It utilizes the enhanced computational power provided by GPU and improvements in image processing techniques accompanied by a better electron microscope and detector to deliver better structures. Yip et al. recently reported that using cryo-EM, they could obtain a high-quality structure of 1.25Å resolution for apoferritin [3]. This technique is appropriate for samples of high molecular weight only. Apart from the aforementioned limitations, the experimental techniques are laborious and expensive. There is a considerable gap between the number of individual protein structures and the protein complex

✉ Sharon Sunny
sharon_p180018cs@nitc.ac.in

P. B. Jayaraj
jayarajpb@nitc.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Calicut, India

structures available in PDB databank. Thus computational techniques become a viable alternative in structure prediction tasks.

Computational techniques for protein complex structure prediction are broadly classified as template-based methods and template-free methods. In template-based methods, the onus of generating accurate structures is fulfilled only with the available templates. A substantive improvement in the quality of predictions is visible by the application of refinement techniques. On the other hand, a template-free method starts the prediction procedure from scratch, with no reference structure available. Protein–protein docking is one of its kind that predicts the near-native orientations of proteins. Input to a docking algorithm can be any oligomeric protein. Largest among the input proteins is designated as the receptor and the smallest as the ligand for computational benefits. Predictions by these techniques are utilized to delineate interactions' characteristics, the role of mutation in interaction patterns, and affinity predictions; this information can aid in drug discovery.

Another exciting classification of docking technique is based on the flexibility of interacting proteins. Rigid-body techniques assume that, upon complex formation, the individual proteins preserve their internal geometry, which is not necessarily the case. Changes can happen to side-chain atoms and/or at backbone positions. Methods like Molecular Dynamics (MD), Monte Carlo (MC) methods, Normal Mode Analysis (NMA) help us to account for such conformational changes. A recent review on flexible docking methods [4] observes that technological advancements have driven the field much forward, but the scope for improvement remains.

We know that 20 amino acids can be connected in different combinations and permutations through peptide bonds to generate different proteins. In 1963, Ramachandran et al. [5] inspected the internals of the structures and proposed the Ramachandran plot, which delineates the range of torsional angles in a near-native protein structure. It suggests the probability contour of torsion angles (shown in green color in Fig. 1) based on reference to already known structures. In Fig. 1, left side shows the plot corresponding to a plausible structure where other than few outliers, all the angles are within the contour. In contrast, the angles are seen dispersed in the allowed and prohibited regions in the figure on the right side, showing its distortion. A Ramachandran plot adhering to the torsional angle criterion is necessary but not sufficient to conclude on the quality of the structure under consideration. The plot cannot identify the clashes between residues far apart in the sequence but too closer in structure, as these residues may satisfy angle criteria. Other conditions like negative potential should be added to identify plausible near-native structures.

2 Protein Representation Schemes

A string of amino acids is designated as polypeptide or protein, depending on its length. The largest protein in the human body, titin, contains 27,000 to 33,000 amino acids, each containing at least four atoms. Computations involving such gigantic proteins in all-atom representation demands more execution time and high-end resources. In such situations, the adoption of a good representation technique may help. This drives the experiments in reduced representation schemes. Different computational techniques assume different levels of granularity—from all-atom to residue level—in representing input proteins. Recent studies show that a coarse-grained approach improves the performance in terms of execution time while maintaining the accuracy [6].

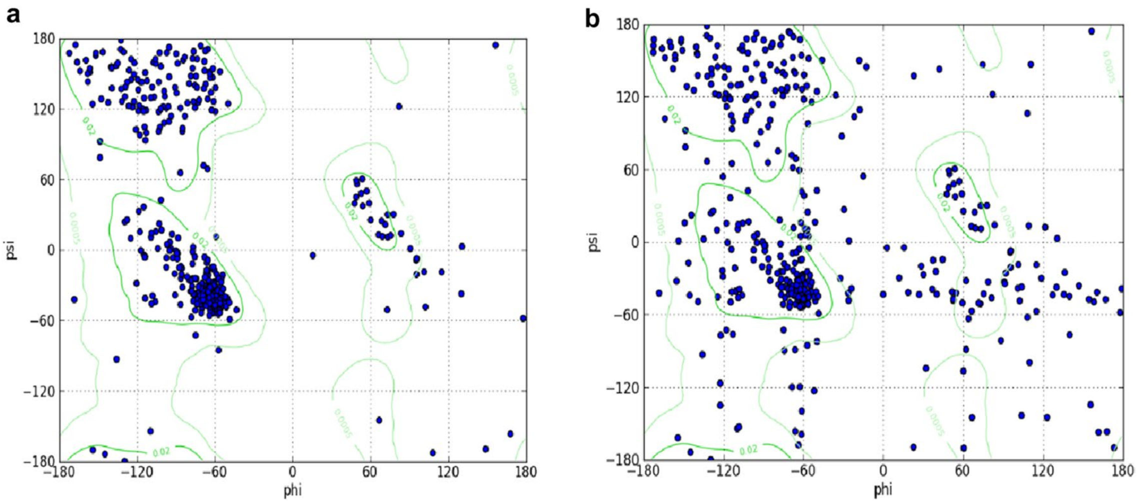


Fig. 1 Ramachandran plot of a plausible structure and distorted structure of a sample protein

Zacharias [7] presented a coarse-grained model where at least two pseudo atoms represent all but Glycine—one pseudo atom represents the Cα atom, and one or two pseudo atoms represent side-chain heavy atoms depending on the size of amino acid. Results show that the minimum energy structure obtained by docking proteins in reduced representation resembles with experimental structure. This scheme is effectively used in many docking techniques [8–10].

Koliński proposed [11] a reduced protein representation, C_α–C_β Side group (CABS) model, where four atoms represent a residue viz. C_α atom, C_β atom, and two pseudo atoms : one representing the center of two C_α atoms, and the other for the center of mass of side-chain atoms. This lattice-based model redefines the force-field parameters based on the extensive statistical analysis of available protein structures. This approach is reported to be widely used in applications related to protein folding, structure prediction tasks [12, 13], protein interactions[14], and protein docking [15].

UNited RESidue (UNRES) model proposed by Khalili et al. [16, 17] reduce the number of representative atoms to two—C_α atom that represents the backbone atom, and center of mass of side-chain atoms—with support for the extended simulation period. The coarse-grained UNRES MODEL is utilized in UNRES-DOCK [18], which is designed for protein–protein and protein-peptide docking.

AWSEM (Associated memory, Water mediated, Structure and Energy Model) [19] uses three atoms to represent a residue: C_α, C_β, and O. The model accommodates both direct and water-mediated interactions. It finds application in the prediction of dimerization interfaces of protein–protein complexes [20]. An extensive study on coarse-grained representation can be found in [21].

With the advancements in problem-solving strategies, there is a paradigm shift in docking methods. Blind docking techniques that demand no a priori information, other than interacting protein structures, usher integrative modeling; this becomes the new order of the day [22]. Recently, deep learning techniques are also gaining ground. This work aims to throw light into the challenges and opportunities in docking techniques, particularly rigid-body docking, how existing methods handle them, and the scope of developing good-sounding concepts.

3 Protein–Protein Docking: A General Pipeline

This section gives an overview of the docking procedure, which may be conducive to understanding its intricacies. The onerous task of protein–protein docking, in general, involves two steps : pose generation and scoring. A schematic diagram of the same is given in Fig. 2. A preprocessing step that precedes the algorithm execution may involve cleaning the bound structure of input proteins, adding missing information on atoms or residues, which is mostly done by MODELLER [23], extracting the surface information, and finding an appropriate representation scheme for input proteins.

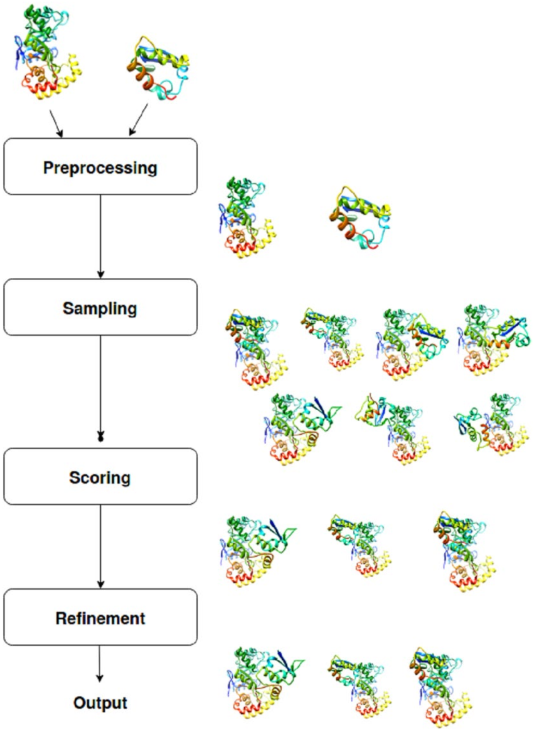


Fig. 2 A general pipeline in protein–protein docking. Here sampling and scoring phases are shown separately for the ease of representation. Usually both stages happen in concert

3.1 Pose Generation

In molecular docking, the pose generation phase is very critical as the exclusion of near-native poses in this step will profoundly affect the efficiency of the method. The basic steps involved in this stage are translation and rotation of interacting proteins, which can be done using an exhaustive or stochastic method; both approaches have advantages and disadvantages. Exhaustive searching, also called systematic searching, brings down the propensity of an algorithm to omit near-native structures by considering all the possible transformations of ligand and receptor at a given interval. It drastically increases the time and space for execution from a different perspective, thus the demand for improved sampling methods. The method is altered so that search confines to regions that can probably lead to near-native structures. The introduction of constraints like shape complementarity and electrostatic complementarity helped in finding potential binding sites. Stochastic methods, on their capacity, can also lead to the accurate prediction of near-native structures, coercing the scoring stage to be more accurate. This technique is more suitable with metaheuristic algorithms that begin execution using a random set of solutions.

3.2 Scoring

This phase involves assigning a score to different poses based on various criteria. The effectiveness of scoring functions may depend on the characteristics of proteins involved in interactions. Scoring functions are classified as force-field-based, knowledge-based, empirical, consensus, and machine learning-based, depending on the parameters used for score calculation. Force-field based scoring functions [24–26] take advantage of non-bonded terms (van der Waals potential and electrostatic potential), combined with bonded terms like bond angle, bond length, and dihedral angle. In the case of rigid-body docking, only the non-bonded terms are applicable. Flexible docking gives more refined results by considering atoms' vibration, and such algorithms use bonded terms, too, for more accurate calculations. Empirical scoring functions [27] utilize intermolecular interactions and changes in the accessible surface area of the proteins to calculate the score. The total score may be attributed to factors like hydrogen bonds, hydrophobicity, and hydrophilicity of residues. Knowledge-based scoring functions [28, 29] use existing knowledge on protein interactions to derive statistical potential mean force. Consensus-based scoring [30] operates on a combination of parameters, considering the different aspects of interactions. With the application of AI techniques, machine learning-based scoring functions [31] are developed.

The best structures chosen after the scoring phase may be subjected to refinement where the side chain or even backbone atoms undergo conformational changes to get the minimum energy structures. A mindful selection of strategies is advised in designing the tools as both steps play a seminal role in accurately predicting structures.

4 Algorithmic Approaches in Protein–Protein Docking

The incalculable computational complexity of docking techniques, owing to the innumerable possibilities it needs to consider, can be tackled by skillfully utilizing the key provisions in different algorithmic approaches. From the introduction of Fast Fourier Transform (FFT) to exhaustive searching techniques, the field was evolving with many dramatic improvements in solution strategies. Different techniques tested with the geometric, energetic, and topological aspects of protein interactions are discussed in this section.

4.1 Computational Geometric Algorithms

Among the many successful methods in docking, geometry-based techniques are one of the prominent because of the shape complementarity exhibited at interface regions of proteins. However, the constraint is not as strict as in protein-ligand docking, where the lock-key strategy is applied. The dominance of shape complementarity largely depends on the nature of interacting proteins. A near-exact complementarity is expected for rigid-body targets, while a relaxed strategy works for antigen-antibody targets [32].

Delaunay tesselation [33], alpha shape [34], convex hull [35, 36], and geometric hashing are some of the geometric concepts that are widely endorsed in the surface generation, interface identification, and pose generation. A general pipeline of geometry-based methods is shown in Fig. 3. The procedure begins with the extraction of surface information of the interacting proteins. The usual practice is to roll a sphere of specific radius over the protein and render the path traced by its surface [37]. The generated surface is fragmented into convex, concave, and flat regions based on the curvature values of the points, and the alignment of these curved regions ensues pose generation. An indispensable entity in protein alignment is a shape descriptor [38, 39] that satisfies the following criteria [40].

Translation and Rotation Invariance: The input proteins, perhaps, have different orientations and may occupy distant coordinate space. The alignment of their complementary regions must not be affected by these factors. The descriptors must be invariant on rotation and translation of the proteins to align them with its counterpart.

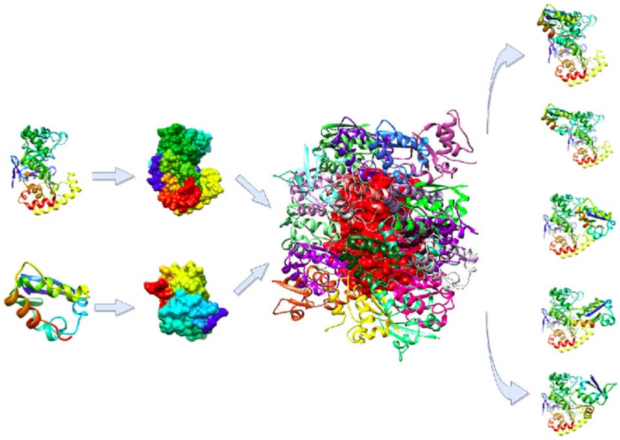


Fig. 3 Geometry based docking. In this figure, first column shows the ribbon structure of input proteins. In the second column, different surface patches are shown in different colors. The solid surface in the 3rd column shows the receptor protein. Some of the possible positions of ligand around the receptor are shown around it as ribbon structures

Statistical Independence: Descriptors of different points must be statistically independent. It indicates the need for a compact representation.

Reliable and Fast: There is a trade-off between correctness and speed of execution. The descriptor must be accurate, and its calculation must not induce a performance bottleneck.

The critical points of alignment may be the center of the surface patch [41] or a point equidistant from the majority of the points in a patch [42]. The curvature value of a point can also make it eligible for a critical point. The alignment of surface patches is pivoted on these critical points. Multipatch alignment can be adopted instead of single or two-patch assembly to ensure that the proteins are arranged based on shape complementarity.

Though not properly addressed, topological complementarity was considered in the pioneering work in automated protein–protein interaction prediction. Wodak et al. [43] applied exhaustive searching, which uses a coarse-grained representation of residues to identify the interacting residue in combination with the free energy of dissociation. The work laid the foundation for further studies in interaction prediction.

A breakthrough in exhaustive searching methods was in 1992 when Katchalski-Katzir et al. [44] introduced FFT to speed up the searching strategy. They formulated a correlation function that can be applied to the discrete molecular representation. The method solely based on geometric complementarity was successful for rigid structures. This revolutionary technique inspired many to modify the same to improve the performance [45–48].

The majority of FFT-based techniques work in 3D transformation space. It requires a new grid for each rotated position, demands separate FFT calculation for each term in the correlation function, and has difficulties in accomodating restraints. Padhorny et al. [49] attempted to apply FFT in 5D rotation space to circumvent these limitations. Compromising energy calculations and offloading the liability of accurate prediction to the clustering stage, the proposed representation in SO(3) contributes largely to fast execution. Cluster centers, chosen as predictions, are selected based on cluster size, balancing the inaccuracy in energy calculation.

Geometric hashing is a computer vision technique that identifies a match between two regions under consideration. It assigns each point an invariant representation with respect to a selected reference frame and compares them for similarity. An agreement in the representation of data points indicates a match between the regions. This technique is successfully implemented in docking algorithms to find the complementary regions on proteins during pose generation [50, 51]. The problems introduced by the non-uniform distribution of data, in the case of proteins, during hashing can be discounted by the application of rehashing techniques [52] or self-organizing maps [53]. Venkatraman et al. [54] proposed an alternative hashing strategy for the same problem, which uses kd-tree to search the matching points; steps to calculate the invariant representation is the same. Recently, Christoffer et al. [55] released the LZerD webserver capable of doing pairwise and multiple protein docking. Estrin et al. [56] proposed the SnapDock algorithm that utilizes hashing in template-based docking. The method calculates hash values of templates in Dockground [57] and PIFACE [58] dataset. Similar processing is applied to the query sequence to find the matching interface region. The identified regions are then aligned to generate the actual prediction.

Another geometric technique adopted in docking is triangulation. When the interaction between atoms is hindered by intermediate atoms, calculating distance-based potential will cause unjustifiable computational costs. With this assumption, Jafari et al. [59] proposed a Delaunay triangulation-based scoring function. It calculates the score only for the triangulated interprotein atoms that are within a threshold distance. The method is proved to have superior performance in calculating the potential but underperforming in structure prediction, which may be rectified by proper tuning.

4.2 Population Based Metaheuristic Algorithms

Among the many factors determining the interaction between proteins, van der Waals force, electrostatic force, and desolvation potential are prominent. Upon complex formation, these forces of attraction or repulsion may induce conformational changes in proteins due to their fundamental nature to exist in minimum stable energy. Thus structure prediction can be modeled as an optimization problem where the system tries to minimize the energy of generated structures. Due to the hardness of the docking problem, a population-based metaheuristic algorithm may be necessary to find a suitable solution. These algorithms, in each iteration, try to improvise the solutions by applying different operators.

Evolutionary algorithms are metaheuristic algorithms that mimic natural evolution. These algorithms balance exploration and exploitation of solution space by applying reproduction, mutation, recombination, and selection operations. The foundation stone for evolutionary algorithms was laid in the 1970s when John Holland [60] proposed Genetic Algorithm (GA). The algorithm begins execution by generating a random set of chromosomes, constituting an initial population. Then, in the course of execution, the algorithm tries to improve the population in each generation by applying cross-over and mutation operations. Finally, to curb the population explosion, only the fittest chromosomes are allowed to survive. In 2001, Gardiner et al. [61] proposed a solution to protein–protein docking based on this algorithm with each chromosome representing the degrees of freedom of ligand in six dimensions. They employed Niche restriction

[62] which chooses a new solution based on its difference with existing ones so that searching is not confined to a local region. Geometric complementarity of the surfaces used to find the fitness of a pose is not implemented as a strict constraint due to the structural characteristics of proteins. Sunny et al. [63] proposed FPDock that uses Flower Pollination Algorithm (FPA) [64]. Pollens, the agents in the algorithm, are quaternions representing the transformation parameters. Pollens try to improve their score in each iteration by combining with the best or some random pollens. These iterations are followed by the purging of unhealthy pollens to avoid population explosion and maintain the population's quality. The pollen score depends on its electrostatic and van der Waals potential, based on which the best pollens are identified. These best pollens may be toppled by better new generation pollen. The algorithm stops execution after a fixed number of iterations as no prior information on the stable potential of the target is available. Choosing multimodal output is satisfied by splitting the initial population into different islands where the earlier steps are applied. Though this split is similar to the island variant of FPA, the migration step is excluded for getting diverse solutions. All the evolutionary algorithms have to deal with a large population, and score calculation can be a bottleneck in execution. The introduction of parallel algorithms can effectively deal with the problem, supported mainly by the island implementation.

Swarm optimization algorithms take the collective behavior of agents in the swarms to find the optimal solution. Figure 4 shows such a system where ligand swarms are formed around the receptor. In Particle Swarm Optimization (PSO) [65], each particle is associated with a position and velocity. These values get updated depending on the best position accomplished by any particle in the swarm, i.e., individual best and swarm best determine particles' current position and velocity. A variant, PSO^2 [66], uses PSO for a local search around selected particles in basic PSO. PSO algorithm along with normal mode analysis is at the pivot of SwarmDock algorithm [67]. Particles, which represent the ligand transformations, are spread around the receptor as in Gaussian distribution. Their fate is determined by distance-dependent potential, which is the sum of electrostatic and van der Waals potential. Best positions on receptors are identified through intensification and diversification of solutions. JabberDock [68] uses PSO "kick reseed", proposed in POW^{er} environment [69], to explore the potential energy surface of the proteins. Unlike PSO, "kick reseed" randomly reinitializes the particle positions, thereby improving the convergence while saving the particles from being trapped in local minima. The docking algorithm starts with an MD simulation followed by the application of the PSO algorithm for sampling. The generated structures are checked for atomic clashes, and those having only negative van der Waals potential are hand-picked for shape complementarity checking. The proposed STID maps are proved to be suitable in devising appropriate scoring functions, and it leads to improved performance of the tool. The tool is found to be efficient for transmembrane proteins, too [70].

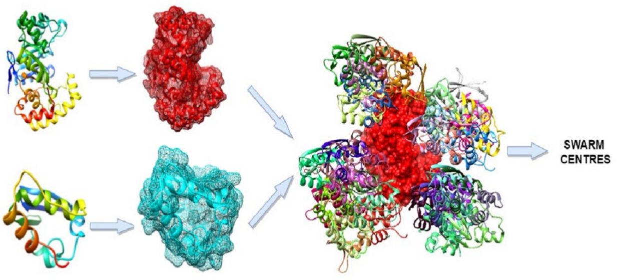


Fig. 4 Population-based docking. From the input proteins, surface information is extracted, shown in column 2. Ligand swarms, shown as ribbon like structures, formed around the receptor are shown as different clusters. The solid surface in the 3rd column shows the receptor protein

A notable work in docking was proposed by Jiménez-García et al. [71] which uses Glowworm Swarm Optimization (GSO). LightDock initializes the swarm centers around the receptor where the swarms get flourished. Each swarm has a swarm center with the highest luciferin value, to which glowworms in its vicinity get attracted. This multimodal optimization algorithm that returns swarm centers accommodates multiple scoring functions. It allows users to design their scoring function or use the default nine scoring functions in any combination. The disadvantage of getting trapped in local minima is avoided here by the multimodal nature of the GSO algorithm. Anisotropic Network Model (ANM) [72, 73] in the framework supports partial flexibility of backbone atoms. A problem that can arise in a 3D system that describes rotation in terms of Euler angles is Gimbal lock problem. This occurs after several rounds of rotations when any two axes of rotation become parallel to each other, and consequently the system loses a degree of freedom in space. Quaternion offers a solution to this problem, and hence it is adopted in this technique. It is observed that the proposed method works best for flexible complexes than for rigid complexes. The blind docking model of LightDock is later revised to a data-driven model that can make predictions based on information about binding sites [74]. This addition made a drastic increase in the success rate of the tool. When compounded with HADDOCK refinement [6], LightDock could work well for membrane-associated complexes, utilizing the topological information of membranes [75]. Faster execution time in LightDock is attributed to the parallel execution supported by the swarms in the system.

4.3 Monte Carlo Algorithms

Monte Carlo (MC) algorithms, indeed, are a class of algorithms that predicts the probability values based on historical data. They have an inherent element of randomness in their execution, with a slim chance of returning an incorrect solution. Nevertheless, these algorithms offer solutions when many fail to do so, making it favorable to find solutions to hard problems. For instance, problems involving numerous degrees of freedom of particles can be solved using MC algorithms. It offers non-deterministic solution to a stochastic problem. The main steps involved in solution determination are defining a solution space, choosing random inputs from a probability distribution and its evaluation, and getting the compounded result. The docking problem has a similar interpretation where solutions are identified from the conformation space based on its probability of being a near-native structure.

RosettaDock [76, 77] uses one such method where MC search joins hands with optimization of atom coordinates to get proper docking results. Here, the rigid-body perturbations, rotamer packing, and minimization of rigid-body displacement followed by the scoring phase precisely determine the quality of predictions. Rosetta has a two-stage sampling procedure—a coarse-grained rigid-body sampling and a fine-grained, flexible sampling where side-chain conformations are acceptable. The unavoidable computational cost incurred by the implementation of MC in high-resolution sampling can be alleviated by parallel tempering or replica exchange tempering [78, 79]. Parallel tempering, proposed to improve physicochemical simulations, starts by randomly initializing N copies of the system at different temperatures; the replicas are allowed to exchange their configurations. A replica can sample over a large space at high temperatures, while local sampling is possible at low temperatures. An ensemble-based method involving frequent temperature exchange between hot and cold samples can avoid a possible convergence at local minima. A deliberate choice of temperature and number of replicas can help in handling computational cost. Another alternative is the use of Hamiltonian exchange among ensembles [80], where replicas show differences in interactions. The allowable degrees of freedom cause a bottleneck in its working as the algorithm needs an overlap in energy levels of ensembles. A well-tempered ensemble (WTE), which holds energy as the collective variable, offers a solution to this problem. Zhang et al. [81] compared the standard MC, Temperature Replica Exchange Monte Carlo (REMC), well-tempered ensemble temperature Replica Exchange Monte Carlo (WTE-REMC), and well-tempered ensemble two-dimensional Hamiltonian Replica Exchange Monte Carlo (WTE-H-REMC) to find the best method that can be added with RosettaDock. The authors observe that the concocted WTE-H-REMC outperformed the other methods.

Siebenmorgen et al. [82] presented repulsive scaling replica exchange MD (RS-REMD) simulation for complex structure prediction. It assigns an increased van der Waals radius and reduced van der Waals attraction, reducing the effect of Lennard Jones (LJ) potential, which is a functional form used to describe van der Waals interactions, and other potential parameters. Compounding this with repulsive bias helps the system to escape from suboptimal binding sites. Existing methods for MD simulation require the starting point to be near the actual binding site. RS-REMD can assure the convergence to a near-native solution with no such constraints.

4.4 Graph Based Algorithm

For problems involving dependencies between entities, a graphical representation of the data perhaps paves the way for better and easier solution strategies. In the basic definition, a graph $G = (V, E)$ consists of a set of vertices V connected by edges in set E. A one-to-one correspondence between two graphs is called graph isomorphism, where the edges between nodes are maintained. In an alternative graph definition, a graph can be expressed as a mapping $f : X- > Y$. i.e., $G = (x, f(x)|x \in X)$. These definitions are applicable in a protein, where there is a relation between atoms connected through bonds. Spatial relations between atoms can be treated as a connection. A residue level graph also holds information about interactions. Apart from the proximity data, it conveys information about physicochemical characteristics as attributes of nodes and edges. Vishveshwara et al. [83] performed a detailed analysis on the construction of protein graphs in various applications. Graph properties like isomorphism and graph spectra are widely used in protein folding, function, and dynamics studies. Clique, which is a complete subgraph in a graph, has numerous applications in different fields. It is used in protein structure comparison, especially in drug discovery applications [84]. Complementarity finding can be easily mapped to similarity finding problems if both inside and outside the protein surface are treated similarly. Grindley et al. [85] proposed a maximum common subgraph (MCS) based solution to the similarity finding problem. The method constructs a correspondence graph that contains all the possible equivalence between the graphs. The following steps are carried out to generate such a graph. Let A and B are two graphs whose MCS needs to be identified. Firstly, generate a set $C = \{(n_a, n_b)|n_a \in A \ and \ n_b \in B\}$. Each element in C occupies a node in the correspondence graph, CG. Add an edge between two nodes in CG if an edge exists between corresponding nodes in graphs A and B. Now cliques in CG can be interpreted as MCSs in A and B. Gardiner et al.

[86] adopted this idea in finding regions that satisfy both shape and hydrogen-bonding complementarity. They identified hydrogen-bonding donor-acceptor pairs in both proteins and constructed a docking graph. The inclusion of an edge in the graph implies that the distances between these atoms in two proteins are comparable with a threshold value. Cliques identified in this graph represent similar regions and can be interpreted as complementary regions. Observing the inner and outer regions alike can lead to the identification of similar regions, too, along with complementary regions, ensuing clashes in docking predictions. Axenopoulos et al. [42] solved the issue by considering the ligand surface and negative of receptor surface; the idea is later successfully adopted in SP-Dock [41], a shape complementarity based docking tool.

Like sampling phase, scoring can also take advantage of graph representation. We know that a tree is an acyclic graph. A kd-Tree is an efficient data structure in handling *k* dimensional data. It follows the principles of a binary search tree, where elements to the left of a node are less than its value and elements to the right are greater than or equal to its value, and hence searching in *k*-D space is easy. The proficiency of a multidimensional binary search tree in tracking down solutions is utilized by TreeDock [87] in exploring the energy landscape of proteins. It generates a kd-tree using the polar coordinates of the atoms in two proteins—one is fixed (F), and the other is assumed to be movable (M)—promoting the easy searching of atoms within a threshold distance from partner protein. This implementation drastically speeds up the energy calculation step, resulting in an overall improvement in performance. Generation of clash-free and low-energy structures is fundamental to perceiving a proper docking tool, which TreeDock achieves by merry-go-round algorithm. In this algorithm, a movable protein explores the space around a fixed protein to find the lowest energy structure. Firstly, using the generated kd-Tree, the merry-go-round algorithm finds those M atoms that can be moved closer to F atoms by rotation about the z-axis. It, then, removes those transformations leading to atomic clashes. Each transformation applied to the movable structure attempts to update the lowest energy values and the corresponding configurations.

The implementation of random walk graph kernel (RWGK) in GraphRank [88] to find subgraph similarity is exploited by Borgwardt et al. [89] to predict functions of proteins. Geng et al. [90] utilized the same for choosing the best docking model. The proposed tool, iScore, is equipped with an SVM classifier to segregate the positive and negative models. An interface graph representing protein–protein interaction is a bipartite graph constructed using interface residues. Each node in this graph is annotated with an evolutionary conservation profile of residues. This 20x1 vector obtained from Position Specific Scoring Matrix (PSSM) represents the log-likelihood ratio between the observed probability of an amino acid to appear in a particular position and its expected probability in a random sequence. RWGK applied concurrently on the two interface graphs calculates the similarity score between them. This is integrated with HADDOCK energy terms to boost the performance of the scoring function. The combination of energetic information and evolutionary information enables iScore to outperform GraphRank. The main bottleneck in the implementation of the tool is the generation of interface graph, and computation of RWGK [91], which is solved by MPI implementation with offloading of tasks to GPU.

4.5 Machine Learning Based Algorithms

Machine learning is a branch of AI that generates models that learn from training data without explicit coding. Statistics and probability form the base of these algorithms. During the training phase, the model tries to identify hidden patterns in the training data. Based on this knowledge, it can later perform similar tasks on new input. ML techniques can be classified on the basis of learning strategies as supervised, unsupervised, semi-supervised and reinforcement learning. Supervised learning techniques require labeled data for training. Once trained, it can perform classification and regression tasks on unknown data. Classification involves categorizing the data into some classes, while regression tasks demand the prediction of numerical values. Unsupervised techniques that do not require labeled training data draw inferences from the given data and divide the data into different classes after several rounds of refinement based on predefined criteria. This approach is best suited for problems like clustering and dimensionality reduction of data. Semi-supervised learning supports the use of a mix of labeled and unlabeled data. The most promising reinforcement learning algorithms take in feedback from the environment, this feedback is then used to ascertain the next step.

The journey towards machine intelligence began in 1944 when Warren McCullough and Walter Pitts implemented a neural network. There were ups and downs in the popularity of the model during the period of its development. The idea of backpropagation put forth in the 1980s triggered a second surge of its acceptance. With the introduction of voluminous data, there was a paradigm shift from knowledge-based methods to data-driven methods. Thus the objective, learning from rules, changed to learning the rules. The current success of machine learning techniques, particularly deep learning, is attributed to parallel computing and advancements in hardware technologies. A fundamental characteristic of machine learning-based methods is that it demands user-supplied features. Classical ML models include Artificial Neural Network (ANN), Support Vector Machine (SVM), and decision trees. SVM uses a kernel function to find the maximal distant hyperplane capable of separating two data classes as shown in Fig. 5. Classification and regression tasks in protein–protein interactions have been successfully implemented using the ML algorithms mentioned earlier.

Being data-driven, the type of data and its volume are primary concerns in developing a good model. Equally important is the number of samples in the different data classes. An imbalance in data may lead to biased predictions, which is unacceptable. E.g., in the case of protein–protein interactions, the number of interacting residue pairs is much lesser than the number of non-interacting pairs. This data imbalance affects the performance of the model. Techniques adopted by existing methods to tackle this issue are discussed in the remainder of the text.

A protein interface is usually defined by changes in the accessible surface area upon complex formation or those residues having heavy atoms within a distance threshold of a partner protein. The threshold is usually taken as 6Å. The commonly used sequence-based features are sequence profile, sequence conservation score, and one-hot encoded residue name. Structure-based features include but are not limited to the relative accessible surface area, half-sphere exposure, geometrical descriptor, hydrophobicity, protrusion index, residue depth, and one-hot encoded secondary structure. Since the properties of interacting residues depend not only on their features but also on their neighbors, the structural and sequential environment features are considered for predictions.

Earlier, SVM-based techniques were adopted for designing scoring functions. The use of probabilistic SVM along with the structural, biological, and physicochemical properties, outperformed many scoring functions [92]. SVM-based scoring function combined with geometric correlation and amino acid-specific scoring functions also yielded good results [93]. Later, this technique was adopted in interface prediction methods. Unlike many sequence-based methods, PAIRPred [94] uses both sequence and structural data to predict partner-specific binding information. An SVM fed with a pairwise kernel predicts the score for each interacting pair of residues. A neighborhood averaging following this step gives the actual interactions. Tests show that bringing in structural features has a positive effect on the results. Das et al. [95] proposed an SVM based classifier for classification of interfacial residues. They generated a non-redundant dataset of complexes using CD-HIT [96] and BLASTp [97]. The method generates complexes from individual protein structures using PatchDock [51] and its identified interfaces are analysed, using PISA [98, 99], for feature extraction. An SVM now does the classification based on training. The proposed method predicts the nearness of a predicted interface to a known interface. This is available online as Protein Complex Prediction by the Interface Properties web server.

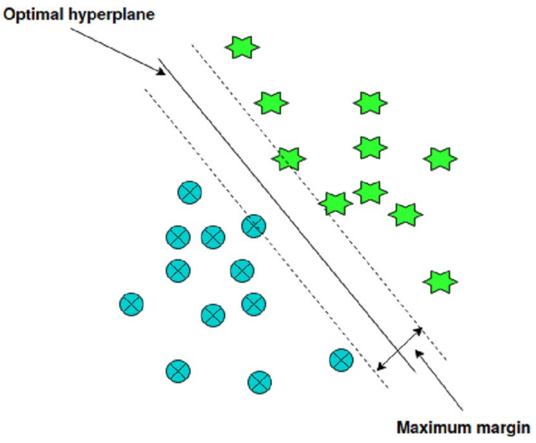


Fig. 5 Support vector machine

Ensemble methods are a class of machine learning methods that base their decision on a set of models created during execution. For instance, averaging and voting are ensemble methods. It outperforms other methods due to its collective decision-making ability. A popular ensemble machine learning algorithm is the random forest, which is widely used in studies related to drug discovery [100] and protein–protein interactions [101, 102]. BIPSPI [103] uses Extreme Gradient Boosting (XGBoost) in identifying the interfacial residues. Like PAIRPred, it works on both sequence and structural data. In its implementation, two consecutive XGBoost classifiers are fed with the feature set; these classifiers then predict interacting pairs, which are finalized by applying a scoring function. Another ensemble-based learning method for protein–protein interaction site prediction is EL-SMURF proposed by Wang et al. [101]. While DLPred [104], a sequence-based method that uses simplified LSTM, achieved an accuracy of 71.1%, 73.1%, and 71.8% for PDBtestset164, Dtestset72, and Dset186, respectively, ELSMURF has an accuracy of 77.7%, 79.1%, and 77.1% on the same datasets. It is interesting to look into the intricacies of the method. The algorithm works on sequence profile features and evolutionary information (Residue Evolution Rate). The application of oversampling to circumvent the data imbalance problem is followed by a multidimensional scaling algorithm to deal with dimensionality reduction of features. The use of a random forest classifier compounded by expert system voting for data integration yields good results in classification.

Though ML methods can work utilizing a small amount of data, they have some disadvantages. These methods require human intervention in feature selection. The importance of chosen features influence the overall performance of the model. Hence it becomes an esoteric job and demands in-depth domain knowledge to develop an efficient model. Another potential demerit is its inability to learn beyond a level, which limits its performance. i.e., the supply of an

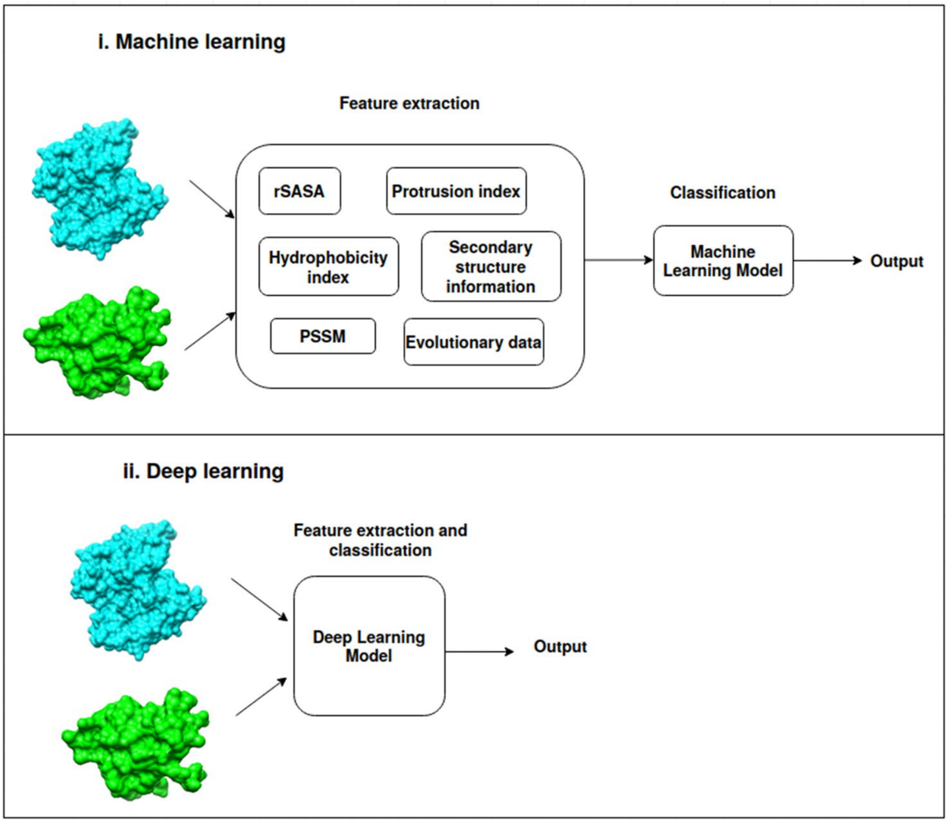


Fig. 6 Machine learning vs deep learning techniques. Machine learning needs human intervention to extract features upon which machine learning model can act. Deep learning models themselves are capable of extracting features

enormous amount of data does not always improve the performance of an ML model.

4.6 Deep Learning Techniques

A boom in deep learning (DL) techniques accompanied revolutionary changes in hardware technologies. DL is a particular class of machine learning methods that relieve the programmer from hand-picking features. They are deep neural networks that can learn from self-designed feature sets, and the supply of abundant data can significantly improve its performance. Equally important is its black box nature, which demands experience to fine-tune the model to get the intended level of accuracy. A schematic diagram showing the difference in ML and DL is shown in Fig. 6. A deep learning model utilizes forward propagation and backward propagation to learn the relation between input and output. The forward propagation step checks whether the model can generate the intended result, and the backward propagation step tunes the network parameters. The ability of an ANN to mimic biological neurons forms the backbone of deep learning networks. A basic flow diagram is shown in Fig. 7. It has different connected nodes triggered based on the signal from other nodes. A neural network in its basic structure has an input layer, at least one hidden layer, and an output layer. The input layer gets the input for the hidden layers to process. The output of all nodes, but input layer nodes, is a function of data from the previous layer, weight associated with the edges connecting it with the previous layer nodes, and the bias, which is an additional adjustment for the output. The activation function applied to the obtained result determines the triggering of a particular node. These functions add nonlinearity to the input, making it suitable for handling complex relations. Also, its differentiability supports the backpropagation step. The nodes that have values above a threshold can pass the data to the next layer. At the output layer, the node having the highest value determines the output of the model. A loss function calculates the difference in expected and predicted outputs based, the backpropagation step then adjusts the weight matrices in different layers. The repeated application of these steps yields the optimal solution. The selection of the loss and activation functions depend on the nature of the problem to be solved. Convolutional Neural Networks (CNN) [105, 106], Recurrent Neural Networks (RNN) [107, 108], Long Short Term Memory networks (LSTM) [109], Generative networks [110, 111], transformers [112, 113] are some of the most popular deep learning architectures.

Recently, the scientific community witnessed the enormous power of deep learning (DL), amassed through years of research, in predicting the tertiary structure of proteins from amino acid sequences. The groundbreaking solution

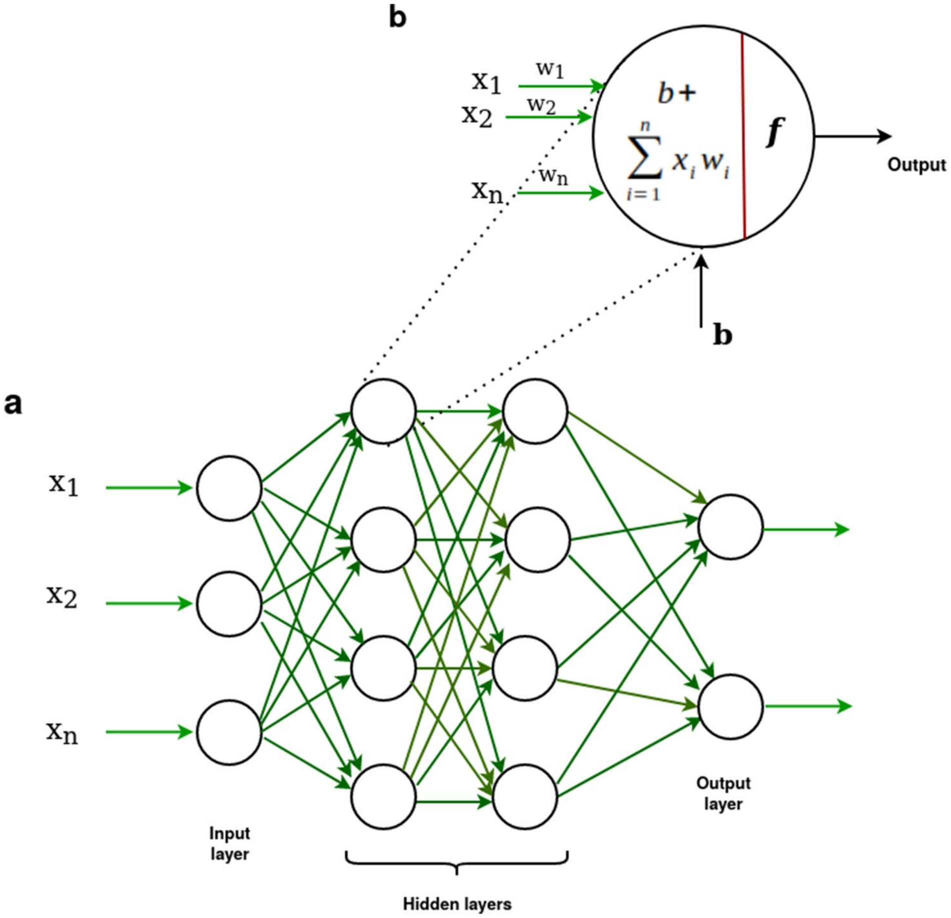


Fig. 7 **a**. Artificial neural network architecture. **b**. The structure of a single node in ANN. x_1, x_2,...x_n represent the input values, w_1, w_2,...w_n represent the weights, b is the bias applied to a neuron, and f is the activation function

to this hard problem was proposed by DeepMind [114]. The program, AlphaFold2, utilized an attention mechanism on a graph along with the evolutionary information. It was reported to be trained on 170,000 structures publicly available in protein databank and other data sources. The performance of AlphaFold in CAPS opened many discussions on the effectiveness of deep learning in structure prediction. Baek et al. [115] used a three-track network that uses the sequence data, distance map, and 3D coordinate information to elucidate the working of AlphaFold. In this method, an SE(3) transformer refines the 3D coordinates of atoms generated by a graph transformer that accepts MSA features. In addition to protein structure, the model could predict protein complex structures too, but with low resolution.

Voluminous data in appropriate representation is required to propel the model to its full performance in DL. One notable advantage of this technique is its ability to identify features independently, thereby exempting the user from the burden of providing essential features. At the same time, this black box nature results in poorly interpretable models. The scientific community has started experiments with deep learning in tasks like identification of binding sites, interface regions, prediction of protein–protein interactions, implementation of scoring functions, and structure prediction of complexes.

4.6.1 Convolutional Neural Networks (CNN)

One of the breakthroughs in deep learning algorithms was in 1998 when Yann LeCun developed CNNs for document recognition [105]. Motivated by the development of Neocognitron [116] in 1988, LeCun started working towards convolutional networks. In its inception, the architecture was called LeNet [117, 118]. Convolutional Neural Networks (CNN or ConvNets) features and architecture are most suitable for handling tensors. The inclusion of the convolution operation led to remarkable changes in the output of these deep learning models. CNN mainly contains four types of layers as shown in Fig. 8. In a model, there can be one or more convolutional layers, which implements convolution operation. In each convolutional layer, filters are convolved over the multi-dimensional input data to extract features. Feature extraction is incremental; the first layer extracts only low-level features. Consecutive layers should extract higher-level features. The results of the dot product of filters and input are known as feature maps. During the training phase, randomly initialized filters are modified to reduce the loss function value. After

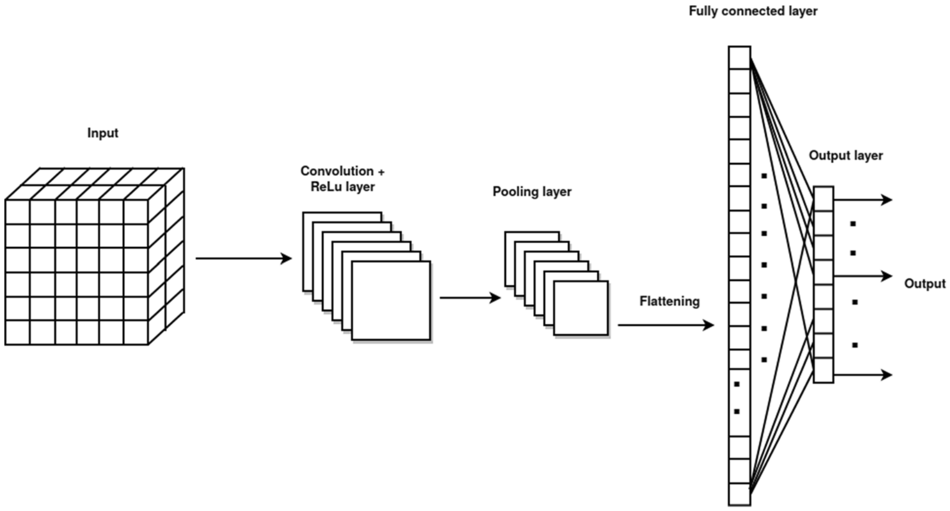


Fig. 8 A convolutional neural network architecture. Multi-dimensional input is passed through convolutional, pooling and fully connected layer to generate the desired output

the convolutional layer, there is pooling layer. The use of too many parameters in a layer may be computationally challenging. Aggregation operations on the feature map can deal with the issue by reducing the dimension. A pooling layer takes up this responsibility. In the pipeline, a ReLU (Rectified Linear Unit) layer, which acts as an activation function, replaces all the negative values in the feature map with zero. The last layer in this architecture is the fully connected layer that processes the flattened data from the previous layer to predict the output. This architecture outperformed all the then state-of-the-art architectures with their prowess to specially treat important input aspects. Apart from this, the application of filters gives a reduced representation making it suitable for handling massive inputs. Inter-protein interactions are orchestrated not only by the participating residues. The neighboring residues rather influence the probability of interaction. A simple neural network can never ought to this demand. CNN equipped with convolution operation is best suitable for this need.

Townshend et al. [119] describe the architecture of SASNet, which uses the spatial information of atoms in proteins to identify the interfacial residues. The method does not use any handcrafted features but applies siamese like 3D CNN on voxelized input proteins containing the atom type information to generate feature vectors. It has improved performance compared with BIPSPI [103] and Node average method using GCN [120] due to the use of a large dataset, DIPS (Database of Interacting Proteins) containing 42826 binary protein interactions, for training. A degradation in the tool's performance when trained using Docking Benchmark 5 (DB5) is due to fewer samples in the dataset. Still, the proposed method outperforms the other two tools.

Xie et al. [121] introduced a CNN-based method for finding the interaction sites utilizing the binding propensity of residues. Given a sequence of amino acids and its associated information like sequence profile and physicochemical properties, and the structural features like accessible surface area, relative accessible surface area, protrusion index, and depth index in addition to its hydrophobicity of residues in the inputs, the algorithm predicts the interaction propensity of the residues. In this implementation, the interface region includes those residues having any atom within a distance of 6Å from any atom in the partner protein. This definition sometimes causes the inclusion of false-positive residues in the interface. The authors observed that these false positives in training data badly affect the prediction accuracy.

Zhu et al. [122] proposed ConvsPPIS in an attempt to devise an ensemble deep convolutional neural network tool to predict interface residues. The method includes evolutionary information in addition to the sequence and structural data to generate separate feature graphs. The association of an ensemble predictor with three deep CNNs trained on these feature graphs exposes residues' interactability. The model is found to be effective and has an accuracy of 88% on the DBv5-Sel dataset.

Hadarovich et al. [123] attempted to predict the structure of homodimers using a CNN model. The first stage generates a contact map using a deep CNN model with binary cross-entropy as the loss function, which is then fed along with the 3D structure of the protein in its unbound state to a gradient descent algorithm. The burden of finding the affine transformation that can be applied to the second protein to get the dimer structure is on the shoulders of the optimization algorithm. The method failed to output an impressive result. Nevertheless, the attempt was appreciated and can be an indicator of future trials.

The efficiency of 2D and 3D CNNs [124–126] are proved undoubtedly through many works in image processing. DOcking decoy selection with Voxel-based deep neural nEtwork (DOVE) [127] is the first attempt to test the applicability of 3D CNNs in protein–protein docking. As the name suggests, it does voxelization of the decoys,

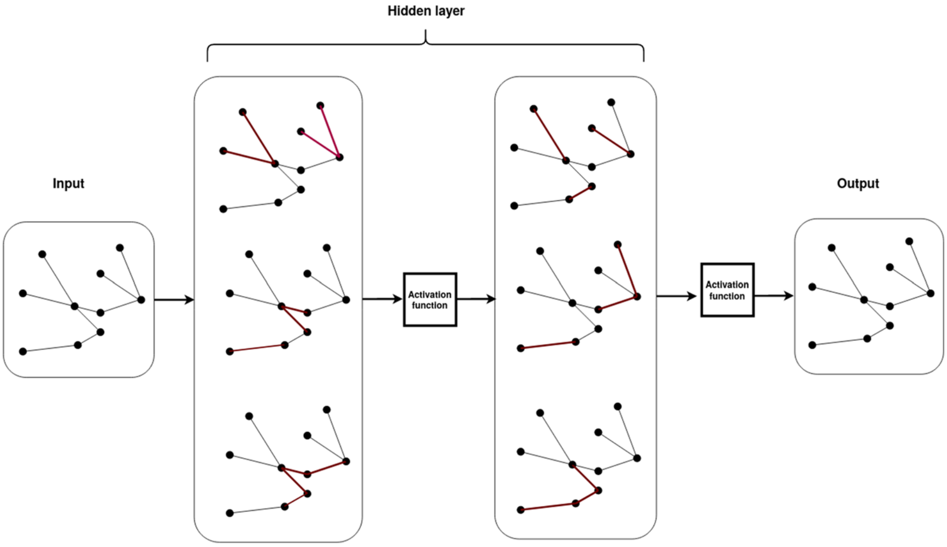


Fig. 9 Multilayer graph convolutional neural network [129]

analyses the interfacial residues, and calculates the potential developed upon complex formation using a 3D CNN to separate the near-native from other structures. The method uses TMscore [128], which measures the similarity between structures using Eq. (1), to remove redundant poses in the input dataset. TMscore uses the number of amino acids in the protein, $\mathrm{L_N}$, number of its matching residues with any reference protein, $\mathrm{L_T}$, distance between ith matching residues, $\mathrm{d_i}$, and the normalization factor, $\mathrm{d_o}$ to calculate the resemblance.

$$TMscore = Max\left[\frac{1}{L_{\mathrm{N}}}\sum_{i=1}^{L_{\mathrm{T}}}\frac{1}{1+\left(\frac{d_{\mathrm{i}}}{d_{\mathrm{o}}}\right)^2}\right] \tag{1}$$

At the same time, to cope with the problem of unbalanced data, positive samples are generated by differently orienting the original structures. It is worth noting that each voxel is associated with total atomic potentials. The performance of the model revealed the power of CNN in scoring too.

4.6.2 Graph Neural Networks(GNN) and Graph Convolutional Networks (GCN)

CNN-like models are suitable when dealing with data that lies in the n-dimensional linear space, $\mathbb{R}^{\mathrm{n}}$. However, not all data can be mapped to $\mathbb{R}^{\mathrm{n}}$. Sometimes, graphs can replace n-dimensional grids to represent the relationship between the different dimensions of input data. Graphs are better structures for representing the hierarchical relation between a set of entities, the same applies to graph neural networks. Input to GNN is a graph that represents the relations between different entities in a problem. Each node embeds information about its neighbors by message passing, which includes an *aggregate* operation on neighboring node data followed by a *combine* operation with the nodes feature. Computations at lth layer can be mathematically written as follows:

$$a_v^{(l)} = Aggregate^{(l)}\left(\left\{h_u^{(l-1)}\ \forall\, u \in \mathcal{N}(v)\right\}\right) \tag{2}$$

where v is the node, $\mathcal{N}()$ returns the neighborhood, $a_v^{(l)}$ is the aggregated node feature of neighborhood, and $h_u^{(l-1)}$ is the neighborhood node feature in $(l-1)$th iteration.

$$h_v^l = Combine^{(l)}\left(h_v^{(l-1)}, a_v^{(l)}\right) \tag{3}$$

Like any other network, stacking of layers can improves the model's performance. Summation of the individual node encodings gives a representation of the whole graph. When the neighbors need to be treated differently, there is a demand to fall back on convolution. Graph convolution networks save the situation. A GCN has a convolution layer, linear layer/fully connected layer, and non-linear activation layer to carry out the intended task. A multilayer GCN is shown in Fig. 9 [129]. The models are scalable as changes need localized modifications to get new embeddings. Another variant of GNN is the Graph autoencoder which encodes the graphical data. The encoder part generates a latent vector representation while the decoder generates the graph from this encoded vector.

Fout [120] proposed a GCN-based model, which classifies the residue pairs as interacting or not, based on structural and sequence data. It separately treats the two proteins using GCN, and the final merging operation facilitates the prediction of interacting residue pairs. The results showed that convolution has a significant role in improving the results. In interface prediction, adding residue binding propensity to separate positive and negative residue pairs is successfully implemented in [121]. This implies a difference in the nature of residues occurring at the surface and interior of proteins. Also, there is residue preference for interaction due to the change in properties of different residues. An integrated graph neural network and CNN-based approach [130] which uses structural, sequential, and high order interaction data was proposed recently. High-order interaction data includes details about adjacent residue pairs as they have a role in the interactions. GNN models generate a sequential representation of individual proteins. This sequential data is summed or concatenated to form a tensor which then passes through CNN layers to yield predictions for pairwise interaction. The inclusion of in-protein interaction information helps to deal with the imbalance in training labels.

Cao et al.[131] use graph convolution networks to implement a scoring function. The network accepts both intra and inter-molecular contact maps as input. This spatial relationship between residues is used to calculate the intra and inter-protein energy values. The model is trained to reduce the difference in calculated Gibbs energy and the actual energy of the complex. Also, the generated score value can be relied on for binding affinity prediction. Improved performance can be expected if the model is trained with data, which is a distribution of test data. Wang et al. [132] presented a GNN based scoring function where interfacial information is embedded in a graph. Two graphs representing the covalent and non-covalent interactions are treated with a gated graph attention network which predicts the probability of a structure to be near-native.

4.6.3 Generative Models

Generative models in deep learning are designed with immense power to generate new data instances. Hence, its introduction has a widespread impact on many fields. For instance, data scientists could compensate for the low data availability to train deep learning models. It is interesting to know the working of generative models. In simple terms, the method attempts to generate data that follows the distribution of training data. When strict constraints on data distribution may make the task impossible, near-exact compliances can be expected from a successful model. Variational autoencoder (VAE) [133] and generative adversarial networks (GAN) are examples of generative models. An autoencoder has an encoder and a decoder network, working hand-in-hand to reconstruct its input at the output. The encoder generates a latent vector of the input on which the decoder operates to generate the actual input. Latent vectors are a reduced representation of the input, and hence the architecture is widely used in dimensionality reduction. Autoencoders with only one hidden layer are vanilla autoencoders. They have the power to generate data from an encoded representation but fail to create variations from the input data. VAE, which uses Bayesian inference to update the probabilities, has the power of a generative model. Instead of encoding the data, VAE tries to encode the data distribution. Thus a sample from this encoded data can be used for generating data that follows the distribution.

GAN [134] is similar to VAE but varies in the training method. Fig. 10 shows the working of a simple GAN. A GAN has a generative network, G, to generate the data, which the discriminator, D, classifies as real or fake based on its training. The objective of the method is to find Nash equilibrium between the two networks, where the generator and discriminator make optimal moves to minimize and maximize the loss, respectively. In short, the model is an

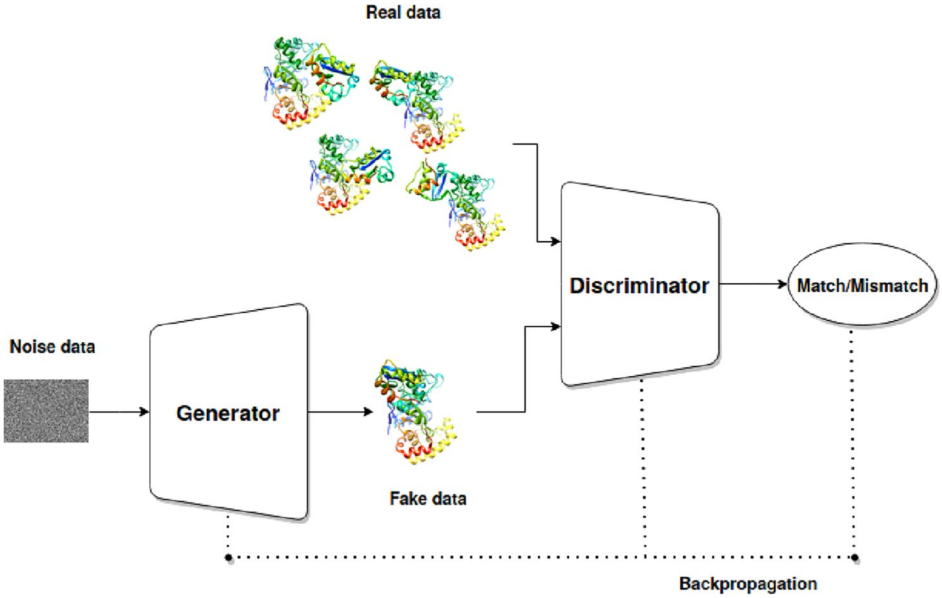


Fig. 10 Overview of generative adversarial network

implementation of a min-max game that works on the following equation:

$$ {}_{G}min\ {}_{D}max\ V(D, G) = \mathbb{E}_{x}[logD(x)] + \mathbb{E}_{z}[log(1 - D(G(z)))] \quad (4)$$

where $\mathbb{E}_z$ is the expectation of noise data and $\mathbb{E}_x$ is the expectation of real data. The discriminator is first trained with actual data. The data generated from the supplied noise data is fed to a discriminator, and its results are propagated back separately to train the two networks. The generator tries to maximize the probability of the discriminator making a mistake. On each iteration, both networks improve their skills until a balance is attained.

Degiacomi [135] tested the ability of autoencoders in generating protein structures. He showed that the decoder part of a well-trained autoencoder suffices to generate a conformation space. Structures generated by MD simulation were used in autoencoder training, which generates latent vector representations of the structures. The trained model can generate a conformation space from these latent vectors. Recently, Ramaswamy et al. [136] improved the conformation space generation procedure using 1D convolutions in both encoder and decoder. The combination of force-field parameters and the Mean Squared Error (MSE) loss function helps to generate low-energy structures. While the non-bonded terms (van der Waals and electrostatic potential) guide the transition to a conformation, bonded terms (bond angle, bond length, and dihedral angles) help in refining the structure by making local alterations in atom positions. Results showed that the inclusion of physics-based terms helped in generating biologically relevant conformations. Nguyen et al. [137] made a similar attempt with generative adversarial networks in D3R Grand Challenge 4. Along with generator and discriminator modules, the proposed method accommodates a mathcentre. The generator module is an auto decoder that generates a structure using latent space and noise input. The mathcentre converts this structure to a low-dimensional mathematical representation utilizing algebraic topology, differential geometry, and algebraic graph. The discriminator, which is an autoencoder, encodes the mathematical representation to a latent space and checks its quality. While the discriminator tries to maximize the loss function, the generator tries to minimize it. Their adversarial action helps the network to find the optimal model parameters.

An image-to-image translation system is designed to accept images from one domain and manipulate them to generate images in another domain that follows a different style. One of the main hurdles in implementing this translation system was the requirement of paired training data. CycleGAN [138], an unsupervised learning technique, was first introduced to circumvent this demand for supervised learning. The network has two pairs of generators and discriminators. One of the notable features of CycleGAN is the cycle consistency loss, which ensures forward-backward consistency of the generated data. Mol-CycleGAN [139] utilized the power of CycleGAN in generating optimized molecular structures. A more precise description would be that, given a molecule, Mol-CycleGAN generates a molecule of a similar structure but with optimized characteristics on par with the standards. A step to generate latent vector containing a component of junction tree scaffold and molecular graph using JT-VAE (Junction Tree VAE) [140] is carried out before the application of cycleGAN. The model works on two sets of latent vectors—an active set, Y, and an inactive set, X. Two mapping functions $G : X- > Y$ and $F : Y- > X$ are defined along with corresponding discriminators $\mathrm{D_Y}$ and $\mathrm{D_X}$ respectively. The discriminator and generator functions try to outperform their adversary till convergence is achieved. For instance, $\mathrm{D_Y}$ forces G to focus on the distribution of Y. It is the identity mapping loss that ensures similarity of input and generated structure. i.e., it ensures that input to one generator and output of the other generator are matching. Once trained, this model has learned the rules for transforming an inactive molecule to its active state. Thus given an input molecule, $x \in X$, G generates an embedding of x that follows the distribution of Y and is similar to x. The generated embedding can be fed to the decoder of JT-VAE to obtain the optimized molecule. The method finds application in generating molecules that are difficult to synthesize.

4.6.4 Other Deep Learning Architectures

Though 3D CNN can handle translation, it has an inherent inability to afford rotation invariance, and hence it might be unwise to use it to manipulate point cloud data. Tensor field networks [141] are special networks that can work with such data as it offers rotation and translation invariance. This distinct quality eschews the data augmentation step in addition to its support for solving problems related to classical mechanics molecular structure. Eismann et al. [142] utilized tensor field networks in PAUL, an end-to-end implementation of a scoring function. Unlike other methods that feed in the physicochemical properties, this method uses only three atom level features—3D coordinates, corresponding atom name, and whether it belongs to receptor or ligand. In the implementation, the convolution operation is carried out by taking the tensor product of input and truncated series of spherical harmonics with learnable radial function to ensure the equivariance of input. Hierarchial learning is achieved by atomic and residue-level aggregations in different layers to finally give a single feature vector representation for the whole structure. Since the physical forces are significant over a locality, convolution is applied within the neighborhood only. The model is trained for the regression task of

Ligand Root Mean Square Deviation (LRMSD) prediction and the classification task of segregating acceptable and unacceptable structures. Each layer of processing reduces the granularity of input data, and in the last fully connected layer, a scalar output is generated. The equivariant network, hierarchical learning, and the nearest neighbor convolutions form the backbone of this model. Results show that augmenting PAUL with other scoring functions promises better prediction.

Gainza et al. [143] explored the capacity of geometric deep learning in checking the interactability of proteins. The methods divide the surface into fragments of fixed geodesic radius to which biophysical and chemical/structural properties are assigned as an encoded descriptor. For a matching pair of surface fragments, the distance between the descriptors will be significantly less, and such a pair can form the base of interaction. The authors describe three critical applications of the descriptor: MaSIF-site, MaSIF-ligand, and MaSIF-search. MaSIF-site identifies the interaction site between two proteins, MaSIF-ligand can find the binding pockets of ligands, and MaSIF-search helps filter out the best probable interactions. The method, using bound structures, is compared with PatchDock, ZDOCK [144], and ZDOCK+ZRANK and is found to have comparable performance but with much-reduced execution time. This is because the fragmentation of surface into patches serves the simultaneous processing of multiple patches. However, the model needs to be improved to get good results for unbound structures.

5 Implementation Aspects: Parallel Computing in Docking

The twenty-first century witnessed many advancements in the hardware domain. It led to evolutionary changes in computation; serial programming paves the way for parallel computing, which can effectively utilize the improved hardware. There are provisions to support instruction-level parallelism, thread-level parallelism, data-level parallelism, and transaction-level parallelism. Porting an existing serial system to parallel execution must be handled carefully to get the intended results. An APOD (Assess, Parallelize, Optimize, and Deploy) strategy can be employed to get the maximum benefits. A proper analysis of the program is required to identify the potential candidates for parallelization. It is always recommended to conduct automated profiling of the code to support the manual analysis. Once profiled, independent steps that take more execution time can be parallelized to get improved efficiency. The use of GPU-accelerated libraries, OpenACC directives, and GPU programming languages helps in accomplishing the task. A code that complies with the best practices adds to the optimized performance and can be deployed successfully. OpenMP, MPI, CUDA, OpenACC, and OpenCL are popular platforms that offer parallel programming capabilities. Protein docking, being a computationally intensive job and has the scope of parallel execution, can benefit from this technique to a great extend. Some of the specific works that adopt parallel programming are discussed below.

MegaDock 4.0 [145] uses the method proposed by Katchalski-Katzir to compute the score of generated poses. Unlike the proposed correlation functions, a single correlation function that accommodates all factors is utilized, making the execution faster. This method offloads all processing into GPU, including voxelization, ligand transformation, score calculation, and final prediction. The provision for simultaneously utilizing multiple cores and GPUs makes the execution even faster. In each of the 420 nodes available for computing, 2 Intel Xenon X5670 CPUs and 3 NVIDIA Tesla K20X GPUs are accommodated. A boost on the software side is achieved using hybrid CUDA, MPI, and OpenMPI. Shimoda et al. [146] studied the effect of different computational environments – GPU and MIC (Many Integrated Cores) – in the working of MEGADOCK and GPU proved to be better.

Cell-Dock [147] implemented an array of techniques for faster execution. It parallelizes both rotation and discretization steps on a PlayStation 3 (PS3) and Cell BE Blade in addition to the data localization techniques employed. It offers two versions – CELL-256 and CELL-128 – out of which the former has a better performance. CELL-256 on a dual-processor Cell BE-based blade contained two SMT-enabled Cell BE processors at 3.2 GHz with 2 GB DD2.0 XDR RAM (1 GB per processor). The integration of task-level parallelism with data-level parallelism improved its performance compared with FTDock[45], which is an FFT-based technique that uses shape and electrostatic complementarity for filtering the structures.

Sukhwani et al. [148] analyze the ways to accelerate the PIPER [46], an FFT-based algorithm . A profile analysis identified FFT as a good candidate for applying parallelization. Along with this, other steps were also considered as the application of Amdahl's law suggests that modifications to the above step could speed up the whole procedure by a factor of 11 only. Finally, the work compared the performance of FPGA and GPU in accelerating docking and observed the dominance of GPU in protein–protein docking.

Deep learning techniques, randomized algorithms, population-based algorithms, and other traditional approaches are suitable candidates for parallel execution. Such techniques can drastically reduce the execution time in addition to effectively utilizing the resources.

6 Critical Assessment of PRediction of Interactions (CAPRI)

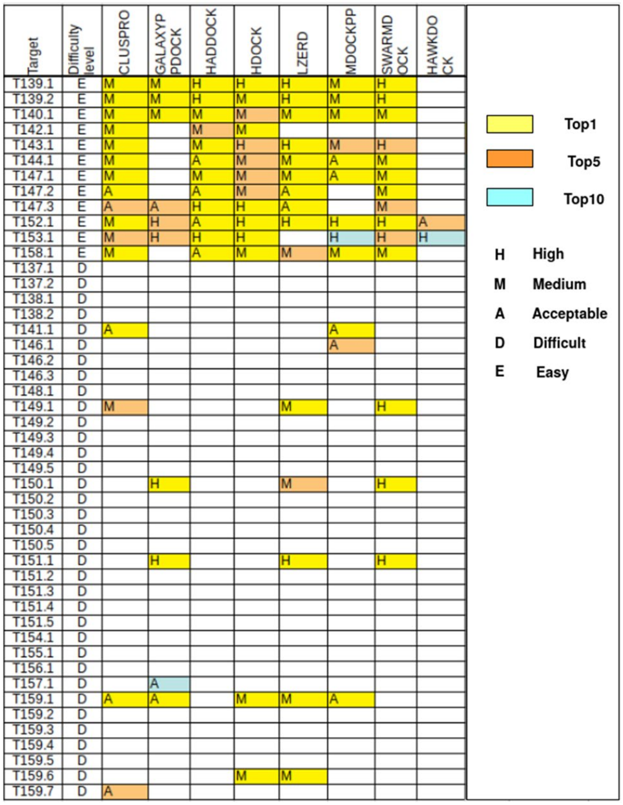
Critical Assessment of Structure Prediction (CASP), started in 1994, to bring state-of-the-art techniques in protein structure prediction from amino acid sequences to a single platform. It is treated as a world championship where reputed labs test their tools using unknown targets, whose structures are not published anywhere. In 2001, EMBL-EBI started Critical Assessment of PRediction of Interactions (CAPRI), following the path laid down by CASP, to showcase the performance of scorers and predictors in the protein–protein docking process. Given the coordinates of input proteins, predictors generate ten possible near-native structures of the complex. In addition, they may supply large number of structures for the scorers to check the efficiency of their scoring functions. CAPRI offers separate tracks for servers and human predictors. Targets, whose experimentally determined complex structures are yet to be published, are chosen in different CAPRI rounds. Emphatically, the individual structures are available in free form. Ab initio methods were adopted for docking at its inception, and only protein–protein complexes were chosen as targets. Later, the arena opened for other protein-bound targets too. In 2014, CASP joined hands with CAPRI for a biennial event CASP-CAPRI. Most of the targets in these events are homo-oligomers, and homology modeling may suffice to predict their complex structures. By this time, predictors relied on template-free methods only on the unavailability of target templates. The introduction of clustering techniques improved the methods by effectively choosing appropriate templates from a set of similar structures. Targets having different structural domains posed a real problem for both predictors and scorers [149, 150]. Apart from the changes in input and procedure, CAPRI brought new objectives to the event. In addition to structure prediction of targets, the participants were assigned the task of predicting position of interfacial water molecules [151], and binding affinity [152–154]. Overall performance of 4 servers which are consistently gaining success in CASP-CAPRI events is shown in Table 1; the values are taken from [149, 150, 155].

Lensink et al. [155] reported the details of CASP13-CAPRI held in 2019. Participants were 24 human predictors, eight predictor servers, 14 human scorers, and five scorer servers. In a competition where template structures are supplied, the general trend of docking servers is to adopt template-based techniques wherever possible. Cluspro server identified appropriate templates of given sequence data using HHPred [156]. On the unavailability of templates for complex, Cluspro [48] opted for free docking, which starts by applying the FFT-based PIPER algorithm [46]. The generated structures were scored using van der Waals, electrostatic, and desolvation potentials. Finally, the clustered structures' centers are chosen and cleaned of atomic clashes by implementing van der Waals minimization that uses CHARMM potentials. MDockPP also uses the FFT-based structure generation technique, followed by an optimization step and scoring using ITScorePP [29]. The final models were selected from clustered data based on the biological information. GalaxyPPDock takes advantage of GalaxyHomomer [157] to segregate the templates identified by HHsearch [158]. It performed FFT-based ab initio docking and then refined the generated structures using GalaxyRefineComplex [159]. PSO-based structure prediction technique uses HHBlits [160] for finding the homologous sequences and generate the individual structures using constricted PSO^2 with Dfire potential [161]. The standard SwarmDock algorithm to find the optimal binding location is then used together with ranking SVM [31] to select the desirable structures. HADDOCK employed ab initio, template-based, and information-driven approaches to predict a near-native structure. However, the ab initio method could not generate any successful targets. LZerD server used a combination of PSI-BLAST and HHpred to find the template structures. Therefore, the application of the LZerD algorithm is limited to cases for which templates could be identified and the ability to identify the correct templates predominantly affects the performance of servers. Optimization of the generated structure is indeed encouraged in the case of difficult targets. Among the predictor servers, HADDOCK outperformed all other servers in quality of prediction and the number of targets solved. This analysis clearly shows that all methods apply template-based techniques wherever possible to obtain high-quality results. Furthermore, with the addition of more protein–protein complex structures, the tool's performance could be improved. Fig. 11 shows the performance of different docking servers based on the rank and quality of the generated structures as reported in [155]. The majority of the servers got through for almost all easy targets but the results were not promising for difficult targets.

Table 1 Composition of docking performance of different servers in CASP-CAPRI experiments. Results of each experiment are taken from corresponding publications [149, 150, 155]

Server	CASP11-CAPRI	CASP12-CAPRI	CASP13-CAPRI
Cluspro	16/8**	7/3**	12/10**
HADDOCK	16/9**	6/1***/1**	9/3***/3**
SwarmDock	11/4**	5/1***/1**	9/5***/4**
LZERD	3	5/1***/2**	9/3***/6**

**Indicates medium quality structures, ***indicates high quality structures, and the numbers indicate the count of targets for which the server has successfully predicted near-native structures



Target	Difficulty level	CLUSPRO	GALAXYPDOCK	HADDOCK	HDOCK	LZERD	MDOCKPP	SWARMDOCK	HAWKDOCK
T139.1	E	🟨 M	🟨 M	🟨 H	🟨 H	🟨 H	🟨 M	🟨 H	
T139.2	E	🟨 M	🟨 M	🟨 H	🟨 M	🟨 H	🟨 M	🟨 H	
T140.1	E	🟨 M	🟨 M	🟨 M	🟧 M	🟨 M	🟨 M	🟨 M	
T142.1	E	🟨 M		🟧 M	🟨 M				
T143.1	E	🟨 M		🟨 M	🟧 H	🟨 H	🟧 M	🟧 H	
T144.1	E	🟨 M		🟨 A	🟧 M	🟨 M	🟨 A	🟨 M	
T147.1	E	🟨 M		🟨 M	🟧 M	🟨 M	🟨 A	🟨 M	
T147.2	E	🟨 A		🟨 A	🟧 M	🟨 A		🟨 M	
T147.3	E	🟧 A	🟧 A	🟨 H	🟨 H	🟨 A		🟧 M	
T152.1	E	🟨 M	🟧 H	🟨 A	🟨 H	🟨 H	🟨 H	🟨 H	🟧 A
T153.1	E	🟧 M	🟧 H	🟨 H	🟨 H		🟦 H	🟧 H	🟦 H
T158.1	E	🟨 M		🟨 A	🟨 M	🟧 M	🟨 M	🟨 M	
T137.1	D								
T137.2	D								
T138.1	D								
T138.2	D								
T141.1	D	🟨 A					🟨 A		
T146.1	D						🟧 A		
T146.2	D								
T146.3	D								
T148.1	D								
T149.1	D	🟧 M				🟨 M		🟨 H	
T149.2	D								
T149.3	D								
T149.4	D								
T149.5	D								
T150.1	D		🟨 H			🟧 M		🟨 H	
T150.2	D								
T150.3	D								
T150.4	D								
T150.5	D								
T151.1	D		🟨 H			🟨 H		🟨 H	
T151.2	D								
T151.3	D								
T151.4	D								
T151.5	D								
T154.1	D								
T155.1	D								
T156.1	D								
T157.1	D		🟦 A						
T159.1	D	🟨 A	🟨 A		🟨 M	🟨 M	🟨 A		
T159.2	D								
T159.3	D								
T159.4	D								
T159.5	D								
T159.6	D				🟨 M	🟨 M			
T159.7	D	🟧 A							

🟨 Top1
🟧 Top5
🟦 Top10

H High
M Medium
A Acceptable
D Difficult
E Easy

Fig. 11 Performance of different docking servers in CASP13-CAPRI in terms of rank and quality of best prediction

7 Benchmark Set and Evaluation Criteria

The truest test for docking tools is their ability to predict near-native structures for flexible targets. There must be ways to compare the performance of different docking techniques. Protein–protein docking benchmark, DOCKGROUND, and PPI4Dock are docking datasets that provide the opportunity for the same.

The protein–protein docking benchmark, proposed by Chen et al. [162] in 2003, was the first of its kind. They took special attention to include targets of all difficulty levels—rigid-body, medium, and difficult—while avoiding redundant structures. This benchmark set was refined many times to generate different versions [163–166]. The latest version protein–protein docking benchmark version 5.5 [167] contains 162 rigid-body, 60 medium difficulty, and 35 difficult targets. It should be noted that these benchmark sets contain experimentally generated structures, and the individual structures are available in a bound and unbound form.

DOCKGROUND [57, 168], a dataset compiled by Douguet et al. in 2006, offers support for multiple aspects in protein–protein complex structure prediction. Apart from experimental structures, computationally obtained bound and unbound structures are also included in this dataset. Specifically, X-ray bound, X-ray unbound, simulated unbound, model-model complexes, X-ray docking decoys, and docking templates are supplied by this dataset. X-ray bound structures, whose details are stored in a PostgreSQL database, are downloaded directly from the PDB repository, and X-ray unbound structures are identified with the help of ProPairs software [169]. Generation of computational unbound structures utilizes the service provided by Langevin dynamics simulation on the bound structures separated from their partners. DOCKGROUND generates decoys structures of complexes using GRAMM-X [170].

The largest dataset for docking hitherto is PPI4Dock [171], which contains 1417 targets obtained by homology modeling. These targets are classified as easy, very_easy, hard, very_hard, and super_hard, depending on the deviation of template structure from the original crystal structure. The authors observe that a tool that could generate near-native structures for very_easy, easy, and hard targets can be considered efficient.

The well-accepted criteria for checking the quality of structures generated by different docking tools is CAPRI evaluation criteria, which uses LRMSD, IRMSD, and f_{nat} values for the same. LRMSD is the root mean square deviation of ligand backbone atoms when receptor backbone atoms of predicted and crystal structure are superposed. Similarly, Interface Root Mean Square Deviation (IRMSD) measures the change in interface coordinates of the prediction and original structure. Fraction of native residues (f_{nat}) keeps a record of the number of interface residues in a native structure that is reproduced in the predicted structure. Based on the values of these parameters, generated structures are classified high, medium, acceptable, and incorrect, as shown in Table 2.

However, this criteria is not robust as it uses three different quantities in qualifying the structures. Even slight changes in the values of any of these parameters may change the quality class of a generated structure. Also, a large LRMSD value may be compensated by a good f_{nat} value, making the structure counted as acceptable. Basu et al. [172] proposed DockQ as an alternative measure that returns a single score value between 0 and 1. It uses scaled IRMSD and LRMSD values obtained by the inverse square scaling technique. The final score is calculated as follows [172]:

$$DockQ = (f_{nat} + LRMSD_{scaled} + IRMSD_{scaled})/3 \tag{5}$$

where

$$RMSD_{scaled} = \frac{1}{1 + \left(\frac{RMSD}{d_i}\right)^2} \tag{6}$$

The values of d_i are 8.5Å and 1.5Å to calculate LRMSD and IRMSD respectively. Even CAPRI competition started

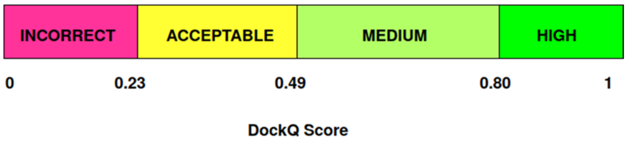


Fig. 12 Range of DockQ score

using DockQ for quality analysis [155]. Fig. 12 shows the range of DockQ scores used in classification of structures into different quality classes.

8 Discussion

The majority of docking tools prefer high-quality X-ray crystallographic structures as input. These structures could reveal the atomic details of proteins. The scarcity of such high-quality X-ray crystallographic data stresses the need to investigate the utility of low-quality structures. Structures obtained through small-angle X-ray scattering (SAXS) and homology modeling may be useful in this regard [173, 174]. The usefulness of augmenting input data with additional theoretical knowledge to get around its frailty can also be a point of investigation.

A docking algorithm is required to take only the necessary information using which it can perform well. One of the most striking examples is the performance of HADDOCK-CG with Martini force-field [6]. A notable feature of coarse-grained representation is that it smoothens the energy landscape and speeds up the execution. Also, it is conducive to deal with slight changes in conformation. Thus careful selection of information may contribute to the performance of the method.

Every sensible docking algorithm that tries to accomplish improvement takes care of its computation cost too. Score calculation and refinement are the main bottlenecks in the majority of the docking procedures. Increased execution time is attributed to the pairwise energy calculation in interprotein residues in score calculation. Since interaction does not happen in the core region, many docking techniques calculate the pairwise energy only for residues at a specific depth. Measures to reduce the size of the conformation space can also help in reducing the computational cost.

A water molecule rich in hydrogen-protein bonds is called a highly coordinated water molecule. Such molecules have a significant role in the interaction [175]. Adding its contribution to the scoring function improves the sampling of conformation space. A two-stage approach that explicitly treats water molecules adds to the performance of the scoring stage [176]. Also, an interacting molecule perhaps displaces water molecules to get associations, and hence it may be utilized as an indicator of binding sites. The use of explicit water molecules in MD simulation can improve results [177].

Evolutionary algorithms are used in many protein-related tasks. Underutilization of these algorithms in docking may be due to the difficulty in fixing the convergence criteria. A possible solution is to fix the number of iterations. Furthermore, the addition of SVM with these algorithms may help to fix the parameters [178, 179]. Another problem with these algorithms is their computational complexity which the application of embarrassingly parallel execution can solve.

Dealing with flexibility is a primary concern in protein docking. An efficient docking tool may have to deal with both side-chain and backbone flexibility. Proper employment of refinement techniques is crucial to get near-native structures for flexible targets. These refinement techniques, in their effort to reduce the physical energy, give better structures. Nevertheless, assuming flexibility to targets may not always result in good structure prediction as a few proteins prefer to be in a rigid state, exhibiting limited segmental flexibility [180, 181]. Thus information about the characteristics of input has an important role. The current trend in docking techniques is towards integrative modeling, which combines information from different sources. X-ray crystallography, NMR spectroscopy, Electron microscopy, footprinting, chemical crosslinking, FRET spectroscopy, SAXS, and proteomics are counted as data sources for such techniques.

The participants in CAPRI opted for ab initio methods only when there were no available templates. The current

Table 2 CAPRI evaluation criteria [184]

CAPRI quality	Conditions
High	f_{nat} ≥0.5 AND (LRMSD≤1.0 OR IRMSD≤1.0)
Medium	(f_{nat} ≥0.3 AND f_{nat} <0.5) AND (LRMSD≤5.0 OR IRMSD≤2.0) OR (f_{nat} ≥0.5 AND LRMSD>1.0 AND IRMSD>1.0)
Acceptable	(f_{nat} ≥0.1 AND f_{nat} <0.3) AND (LRMSD≤10.0 OR IRMSD≤4.0) OR (f_{nat} ≥0.3 AND LRMSD>5.0 AND IRMSD>2.0)
Incorrect	f_{nat} <0.1 OR (LRMSD>10.0 AND IRMSD>4.0)

Reprinted from [184]

trend shows that template-based methods are gaining popularity. With the addition of more and more complex structures, such methods could perform better.

8.1 Challenges and Future in Deep Learning Techniques

The transition from ML to DL was gradual and was boosted by the technological advancements in GPUs. Backed by the high computational power, DL models are responsible for feature extraction from training data on the precondition of supply of a large quantity of data. Apart from the theoretical aspects, in practice, the claim of DL methods extracting features is specious. The input need not always be raw data; a preprocessing step perhaps precedes a deep learning model. Max pooling layer is one instance of hard-coded features. However, the pretense of requiring fewer hand-coded features compared to ML models is ensured in DL models.

The model design demands handcrafted features due to the reduced number of known protein complex structures. A major hurdle in using deep learning-based approaches on protein docking or any other protein-based application is data representation. It is challenging to effectively represent both the positional information of the atoms and the interactions between them. Furthermore, using a 3-dimensional space to represent the positional details of the protein is also difficult due to its sparsity. The uneven distribution of atoms adds to the trouble of representation scheme selection. A sparse convolution offers a solution to the sparsity problem. Another possibility is the use of graph representation which graph networks can process. Though graphs can give spatial proximity data, they lack implicit data on the location and direction of atoms/residues and can be added as an explicit node feature. A novice designer can take the assistance of encoder networks to get the proper representation schemes. It not only returns a reduced representation but also avoids less critical features. When dealing only with sequence data, BERT and transformers are also helpful in finding the embeddings for input data.

As with any deep learning model, overfitting and underfitting can shadow the performance. The model may learn the training data in every detail, including the noise, making it unfit for predicting new data. The same can happen when the model fails to understand the underlying rule of the data. Proper training with adequate data can solve this issue; the distribution of data matters here. Training data must contain data from different data distributions with sufficient size.

Finding the fitting parameters is essential for the model performance. Equally important is the tuning of hyperparameters. For instance, varying the learning rate during the training phase possibly improves the learning of the model. The available computational capabilities may constrain the selection of batch size.

Deep learning techniques are computationally demanding due to the requirement for processing huge-sized data and the depth of the network. Deeper networks may use a huge number of parameters and thus may be stopped by computational limitations. Furthermore, compared to the evolutionary changes in algorithmic approaches, hardware improvements are far behind, limiting the model performance. Researchers observe that only computationally efficient methods can sustain in the future as computational limits are fast approaching [182].

A promising deep learning model for protein–protein complex structure prediction is GAN. Generator and discriminator networks in GAN are ideally working against each other. Hence it is crucial to find the correct balance between their working. As with any other model, the selection of loss function demands utmost care. Since GAN is designed for data generation, further augmenting data for training may negatively affect the results. Another possibility is using tensor field networks, which are helpful when dealing with point cloud data that demands transformation invariance. This property avoids the need for data augmentation.

Protein complex structure prediction from sequence data is also worth mentioning. Discussions about quaternary structure prediction followed the much-celebrated success of Alphafold [183] in tertiary structure prediction. A transformer is one of the successful neural network models capable of dealing with sequential data. The main crux of the transformer lies in its self-attention module, which computes how strongly the information should be routed from one token to another. The transformer suits best as any protein-related task depends on the interaction between the amino acids in the sequence. However, it considers the relations between every pair of tokens in the input, which may not be desirable for protein sequences where only local attention is needed. In addition, a fully connected graph for the long protein sequences may overburden the computational resources. The recently proposed graph transformers seem a solution to this problem. It gives local attention to the input tokens. In other words, it examines only the relationship between connected nodes, making it suitable for tasks related to interaction prediction. The idea can be further extended to structure prediction tasks. Unlike RNNs and other sequential models, the transformer performs better because it can be parallelized, and also, the attention module makes the information flow much more concrete.

The application of deep learning in the structure prediction of protein complexes has to overcome other challenges too. First, training a model for the same purpose requires voluminous data. The number of complex structures available in different databases counts to a few thousand, and this may not be sufficient for a model to learn the rules. Second, any learning procedure ought to deal with the data imbalance problem. Third, since a model cannot afford to learn from low-quality data, there is a stringent need for high-quality experimental data. A generative model may be a pathbreaking solution to this problem.

There is immense scope for the application of deep learning techniques in protein-related tasks. Interaction prediction of proteins, interfacial region identification, classification of interfacial residue pairs, interprotein contact map prediction, implementation of scoring function, and generation of conformation space are actively probed by the research community. In addition, attempts to generate protein complex structures are on their way to development.

9 Conclusion

Study on protein interactions is crucial in understanding the working of biological systems. Computational techniques have overtaken experimental methods in popularity but need much refinement for the former to replace the later. Conventional techniques for protein docking in their capacity can work with less volume of data. These methods analyze the physicochemical and geometric properties of proteins to predict probable near-native structures. The addition of more and more protein–protein complex structures to different databases favors templates-based methods in docking. Template-based methods are more efficient than ab initio methods, provided proper templates. Also, experiments for integrating available knowledge in structure prediction are increasing due to its improved performance. The latest trend in problem-solving is the application of deep learning techniques. They demand a vast amount of data for training. A full-fledged DL model for protein–protein docking shall be expected soon as the algorithmic techniques and the size of protein–protein complex data are increasing.

Acknowledgements The authors acknowledge the Department of Computer Science and Engineering, NIT Calicut, for facilitating the access to the research articles referenced. Special thanks to Mr. Abdulfathaah Shamsuddin, Mr. Gokul Prasad S, Mr. Joel Mathew Cherian, Mr. Roshan MSB, and Mr. Saimanoj Akondi of NIT Calicut for their valuable suggestions.

Author Contributions SS has conceptualised the idea, collected the papers, and prepared the manuscript. PBJ has reviewed the manuscript.

Funding Not applicable.

Data Availability Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interest The authors declare no conflicts of interests.

Ethical Approval Not applicable.

References

1. David A, Sternberg MJ (2015) The contribution of missense mutations in core and rim residues of protein-protein interfaces to human disease. J Mol Biol 427(17):2886–2898
2. Porta-Pardo E, Garcia-Alonso L, Hrabe T, Dopazo J, Godzik A (2015) A pan-cancer catalogue of cancer driver protein interaction interfaces. PLoS Comput Biol 11(10):e1004518
3. Yip KM, Fischer N, Paknia E, Chari A, Stark H (2020) Atomic-resolution protein structure determination by cryo-EM. Nature 587(7832):157–161
4. Harmalkar A, Gray JJ (2021) Advances to tackle backbone flexibility in protein docking. Curr Opin Struct Biol 67:178–186
5. Ramachandran GN, Ramakrishnan C, Sasisekharan V (1963) Stereochemistry of polypeptide chain configurations. J Mol Biol 7:95–99
6. Roel-Touris J, Don CG, Honorato RV, Rodrigues JP, Bonvin AM (2019) Less is more: coarse-grained integrative modeling of large biomolecular assemblies with HADDOCK. J Chem Theory Comput 15(11):6358–6367
7. Zacharias M (2003) Protein-protein docking with a reduced protein model accounting for side-chain flexibility. Protein Sci 12(6):1271–1282
8. Zacharias M (2005) ATTRACT: protein-protein docking in CAPRI using a reduced protein model. Proteins: Struct Funct Bioinform 60(2):252–256
9. Ruiz Echartea ME, Chauvot de Beauchêne I, Ritchie DW (2019) EROS-DOCK: protein-protein docking using exhaustive branch-and-bound rotational search. Bioinformatics 35(23):5003–5010
10. Ruiz Echartea ME, Ritchie DW, Chauvot de Beauchêne I (2020) Using restraints in EROS-DOCK improves model quality in pairwise and multicomponent protein docking. Proteins Struct Funct Bioinform 88(8):1121–1128
11. Koliński A (2004) Protein modeling and structure prediction with a reduced representation. Acta Biochim Pol 51:349–371
12. Blaszczyk M, Jamroz M, Kmiecik S, Kolinski A (2013) CABS-fold: server for the de novo and consensus-based prediction of protein structure. Nucleic Acids Res 41(W1):W406–W411
13. Jamroz M, Kolinski A, Kmiecik S (2013) CABS-flex: server for fast simulation of protein structure fluctuations. Nucleic Acids Res 41(W1):W427–W431
14. Verkhivker GM, Di Paola L (2021) Integrated biophysical modeling of the SARS-CoV-2 spike protein binding and allosteric interactions with antibodies. J Phys Chem B 125(18):4596–4619
15. Kurcinski M, Jamroz M, Blaszczyk M, Kolinski A, Kmiecik S (2015) CABS-dock web server for protein-peptide docking with significant conformational changes and without prior knowledge of the binding site: PJ-022. Nucleic Acid Res. https://doi.org/10.1093/nar/gkv456
16. Khalili M, Liwo A, Rakowski F, Grochowski P, Scheraga HA (2005) Molecular dynamics with the united-residue model of polypeptide chains. I. Lagrange equations of motion and tests of numerical stability in the microcanonical mode. J Phys Chem B 109(28):13785–13797
17. Khalili M, Liwo A, Jagielska A, Scheraga HA (2005) Molecular dynamics with the united-residue model of polypeptide

chains. II. Langevin and Berendsen-bath dynamics and tests on model -helical systems. J Phys Chem B 109(28):13798–13810

18. Krupa P, Karczyńska AS, Mozolewska MA, Liwo A, Czaplewski C (2020) UNRES-dock-protein-protein and peptide-protein docking by coarse-grained replica-exchange MD simulations. Bioinformatics 37(11):1613–1615
19. Davtyan A, Schafer NP, Zheng W, Clementi C, Wolynes PG, Papoian GA (2012) AWSEM-MD: protein structure prediction using coarse-grained physical potentials and bioinformatically based local structure biasing. J Phys Chem B 116(29):8494–8503
20. Zheng W, Schafer NP, Davtyan A, Papoian GA, Wolynes PG (2012) Predictive energy landscapes for protein-protein association. Proc Natl Acad Sci 109(47):19244–19249
21. Kmiecik S, Gront D, Kolinski M, Wieteska L, Dawid AE, Kolinski A (2016) Coarse-grained protein models and their applications. Chem Rev 116(14):7898–7936
22. Yan Y, He J, Feng Y, Lin P, Tao H, Huang SY (2020) Challenges and opportunities of automated protein-protein docking: HDOCK server vs human predictions in CAPRI Rounds 38-46. Proteins: Struct Funct Bioinform 88(8):1055–1069
23. Šali A, Blundell TL (1993) Comparative protein modelling by satisfaction of spatial restraints. J Mol Biol 234(3):779–815
24. Feng T, Chen F, Kang Y, Sun H, Liu H, Li D, Zhu F, Hou T (2017) HawkRank: a new scoring function for protein-protein docking based on weighted energy terms. J Cheminform 9(1):1–5
25. Zhang C, Lai L (2011) SDOCK: a global protein-protein docking program using stepwise force-field potentials. J Comput Chem 32(12):2598–2612
26. Kynast P, Derreumaux P, Strodel B (2016) Evaluation of the coarse-grained OPEP force field for protein-protein docking. BMC Biophys 9(1):1–7
27. Roy AA, Dhawanjewar AS, Sharma P, Singh G, Madhusudhan MS (2019) Protein Interaction Z Score Assessment (PIZSA): an empirical scoring scheme for evaluation of protein-protein interactions. Nucleic Acids Res 47(W1):W331–W337
28. Andreani J, Faure G, Guerois R (2013) InterEvScore: a novel coarse-grained interface scoring function using a multi-body statistical potential coupled to evolution. Bioinformatics 29(14):1742–1749
29. Huang SY, Zou X (2008) An iterative knowledge-based scoring function for protein-protein recognition. Proteins: Struct Funct Bioinform 72(2):557–579
30. Chermak E, Petta A, Serra L, Vangone A, Scarano V, Cavallo L, Oliva R (2015) CONSRANK: a server for the analysis, comparison and ranking of docking models based on inter-residue contacts. Bioinformatics 31(9):1481–1483
31. Moal IH, Barradas-Bautista D, Jiménez-García B, Torchala M, van der Velde A, Vreven T, Weng Z, Bates PA, Fernández-Recio J (2017) IRaPPA: information retrieval based integration of biophysical models for protein assembly selection. Bioinformatics 33(12):1806–1813
32. Yan Y, Huang SY (2019) Pushing the accuracy limit of shape complementarity for protein-protein docking. BMC Bioinform 20(25):1
33. Mitra P, Pal D (2010) New measures for estimating surface complementarity and packing at protein-protein interfaces. FEBS Lett 584(6):1163–1168
34. Albou LP, Schwarz B, Poch O, Wurtz JM, Moras D (2009) Defining and characterizing protein surface using alpha shapes. Proteins: Struct Funct Bioinform 76(1):1–2
35. Leman JK, Lyskov S, Bonneau R (2017) Computing structure-based lipid accessibility of membrane proteins with mp_lipid_acc in RosettaMP. BMC Bioinform 18(1):1–9
36. Zhao R, Cang Z, Tong Y, Wei GW (2018) Protein pocket detection via convex hull surface evolution and associated Reeb graph. Bioinformatics 34(17):i830-7
37. Sanner MF, Olson AJ, Spehner JC (1995) Fast and robust computation of molecular surfaces. In: Proceedings of the eleventh annual symposium on Computational geometry, pp 406–407
38. Laga H, Schreck T, Ferreira A, Godil A, Pratikakis I, Veltkamp R (2011) Bag of words and local spectral descriptor for 3D partial shape retrieval. In: Proceedings of the Eurographics workshop on 3D object retrieval (3DOR’11), pp 41-48
39. Reuter M, Wolter FE, Peinecke N (2006) Laplace-Beltrami spectra as ‘Shape-DNA’ of surfaces and solids. Comput Aided Des 38(4):342–366
40. Park F (2011) Shape descriptor/feature extraction techniques. UCI iCAMP2011, pp 1–25
41. Axenopoulos A, Daras P, Papadopoulos GE, Houstis EN (2012) SP-dock: protein-protein docking using shape and physicochemical complementarity. IEEE/ACM Trans Comput Biol Bioinform 10(1):135–150
42. Axenopoulos A, Daras P, Papadopoulos G, Houstis E (2011) A shape descriptor for fast complementarity matching in molecular docking. IEEE/ACM Trans Comput Biol Bioinform 8(6):1441–1457
43. Wodak SJ, Janin J (1978) Computer analysis of protein-protein interaction. J Mol Biol 124(2):323–342
44. Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA (1992) Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. Proc Natl Acad Sci 89(6):2195–2199
45. Gabb HA, Jackson RM, Sternberg MJ (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. J Mol Biol 272(1):106–120
46. Kozakov D, Brenke R, Comeau SR, Vajda S (2006) PIPER: an FFT-based protein docking program with pairwise potentials. Proteins: Struct Funct Bioinform 65(2):392–406
47. Jiménez-García B, Pons C, Fernández-Recio J (2013) pyDockWEB: a web server for rigid-body protein-protein docking using electrostatics and desolvation scoring. Bioinformatics 29(13):1698–1699
48. Kozakov D, Hall DR, Xia B, Porter KA, Padhorny D, Yueh C, Beglov D, Vajda S (2017) The ClusPro web server for protein-protein docking. Nat Protoc 12(2):255
49. Padhorny D, Kazennov A, Zerbe BS, Porter KA, Xia B, Mottarella SE, Kholodov Y, Ritchie DW, Vajda S, Kozakov D (2016) Protein-protein docking by fast generalized Fourier transforms on 5D rotational manifolds. Proc Natl Acad Sci 113(30):E4286–E4293
50. Fischer D, Lin SL, Wolfson HL, Nussinov R (1995) A geometry-based suite of moleculardocking processes. J Mol Biol 248(2):459–477
51. Schneidman-Duhovny D, Inbar Y, Nussinov R, Wolfson HJ (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. Nucleic Acids Res 33(suppl–2):W363–W367
52. Lifshits M, Blayvas I, Goldenberg R, Rivlin E and Rudzsky M (2004) Rehashing for Baysian geometric hasing. In: Proceedings of the 17th international conference on ICPR’04, vol 3, pp 99–102
53. Bebis G, Georgiopoulos M, Lobo NV (1998) Using self-organizing maps to learn geometric hash functions for model-based object recognition. IEEE Trans Neural Netw 9(3):560–70
54. Venkatraman V, Yang YD, Sael L, Kihara D (2009) Protein-protein docking using region-based 3D Zernike descriptors. BMC Bioinform 10(1):1–21
55. Christoffer C, Chen S, Bharadwaj V, Aderinwale T, Kumar V, Hormati M, Kihara D (2021) LZerD webserver for

pairwise and multiple protein-protein docking. Nucleic Acids Res 49(W1):W359–W365

56. Estrin M, Wolfson HJ (2017) SnapDock-template-based docking by geometric hashing. Bioinformatics 33(14):i30–i36
57. Douguet D, Chen HC, Tovchigrechko A, Vakser IA (2006) Dockground resource for studying protein-protein interfaces. Bioinformatics 22(21):2612–2618
58. Cukuroglu E, Gursoy A, Nussinov R, Keskin O (2014) Nonredundant unique interface structures as templates for modeling protein interactions. PLoS ONE 9(1):e86738
59. Jafari R, Sadeghi M, Mirzaie M (2016) Investigating the importance of Delaunay-based definition of atomic interactions in scoring of protein-protein docking results. J Mol Graph Modell 66:108–14
60. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
61. Gardiner EJ, Willett P, Artymiuk PJ (2001) Protein docking using a genetic algorithm. Proteins: Struct Funct Bioinform 44(1):44–56
62. Beasley D, Bull DR, Martin RR (1993) A sequential niche technique for multimodal function optimization. Evolut Comput 1(2):101–25
63. Sunny S, Jayaraj PB (2021) FPDock: protein-protein docking using flower pollination algorithm. Comput Biol Chem 93:107518
64. Kazemian M, Ramezani Y, Lucas C, Moshiri B (2006) Swarm clustering based on flowers pollination by artificial bees. Swarm intelligence in data mining. Springer, Berlin, pp 191–202
65. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95- IEEE international conference on neural networks, vol 4, p 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
66. Khairy M, Fayek MB, Hemayed EE (2011) Evolutionary computation (CEC). IEEE, pp 1826–1832
67. Moal IH, Bates PA (2010) SwarmDock and the use of normal modes in protein-protein docking. Int J Mol Sci 11(10):3623–3648
68. Rudden LS, Degiacomi MT (2019) Protein docking using a single representation for protein surface, electrostatics, and local dynamics. J Chem Theory Comput 15(9):5135–5143
69. Degiacomi MT, Dal Peraro M (2013) Macromolecular symmetric assembly prediction using swarm intelligence dynamic modeling. Structure 21(7):1097–1106
70. Rudden LS, Degiacomi MT (2021) Transmembrane protein docking with JabberDock. J Chem Inf Model 61(3):1493–1499
71. Jiménez-García B, Roel-Touris J, Romero-Durana M, Vidal M, Jiménez-González D, Fernández-Recio J (2018) LightDock: a new multi-scale approach to protein-protein docking. Bioinformatics 34(1):49–55
72. Doruker P, Atilgan AR, Bahar I (2000) Dynamics of proteins predicted by molecular dynamics simulations and analytical approaches: application to amylase inhibitor. Proteins: Struct Funct Bioinform 40(3):512–524
73. Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I (2001) Anisotropy of fluctuation dynamics of proteins with an elastic network model. Biophys J 80(1):505–515
74. Roel-Touris J, Bonvin AM, Jiménez-García B (2020) LightDock goes information-driven. Bioinformatics 36(3):950–952
75. Roel-Touris J, Jiménez-García B, Bonvin AM (2020) Integrative modeling of membrane-associated protein assemblies. Nat Commun 11(1):1–1
76. Lyskov S, Gray JJ (2008) The RosettaDock server for local protein-protein docking. Nucleic Acids Res 36(suppl–2):W233–W238
77. Zhang Z, Lange OF (2013) Replica exchange improves sampling in low-resolution docking stage of RosettaDock. PLoS ONE 8(8):e72096
78. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D (2003) Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J Mol Biol 331(1):281–299
79. Sugita Y, Okamoto Y (1999) Replica-exchange molecular dynamics method for protein folding. Chem Phys Lett 314(1–2):141–151
80. Fukunishi H, Watanabe O, Takada S (2002) On the Hamiltonian replica exchange method for efficient sampling of biomolecular systems: application to protein structure prediction. J Chem Phys 116(20):9058–9067
81. Zhang Z, Schindler CE, Lange OF, Zacharias M (2015) Application of enhanced sampling Monte Carlo methods for high-resolution protein-protein docking in Rosetta. PLoS ONE 10(6):e0125941
82. Siebenmorgen T, Engelhard M, Zacharias M (2020) Prediction of protein-protein complexes using replica exchange with repulsive scaling. J Comput Chem 41(15):1436–1447
83. Vishveshwara S, Brinda KV, Kannan N (2002) Protein structure: insights from graph theory. J Theor Comput Chem 1(01):187–211
84. Jayaraj PB, Rahamathulla K, Gopakumar G (2016) A GPU based maximum common subgraph algorithm for drug discovery applications. In: 2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW), pp 580–588
85. Grindley HM, Artymiuk PJ, Rice DW, Willett P (1993) Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. J Mol Biol 229(3):707–721
86. Gardiner EJ, Willett P, Artymiuk PJ (2000) Graph-theoretic techniques for macromolecular docking. J Chem Inf Comput Sci 40(2):273–279
87. Fahmy A, Wagner G (2002) TreeDock: a tool for protein docking based on minimizing van der Waals energies. J Am Chem Soc 124(7):1241–1250
88. He H, Singh AK (2006) Graphrank: Statistical modeling and mining of significant subgraphs in the feature space. In: Sixth international conference on data mining (ICDM'06), pp 885–890
89. Borgwardt KM, Ong CS, Schönauer S, Vishwanathan SV, Smola AJ, Kriegel HP (2005) Protein function prediction via graph kernels. Bioinformatics 21(suppl–1):i47-56
90. Geng C, Jung Y, Renaud N, Honavar V, Bonvin AM, Xue LC (2020) iScore: a novel graph kernel-based function for scoring protein-protein docking models. Bioinformatics 36(1):112–121
91. Renaud N, Jung Y, Honavar V, Geng C, Bonvin AM, Xue LC (2020) iScore: an MPI supported software for ranking protein-protein docking models based on a random walk graph kernel and support vector machines. SoftwareX 11:100462
92. Martin O, Schomburg D (2008) Efficient comprehensive scoring of docked protein complexes using probabilistic support vector machines. Proteins: Struct Funct Bioinform 70(4):1367–78
93. Heuser P, Schomburg D (2007) Combination of scoring schemes for protein docking. BMC Bioinform 8(1):1–1
94. Afsar Minhas FU, Geiss BJ, Ben-Hur A (2014) PAIRpred: partner-specific prediction of interacting residues from sequence and structure. Proteins: Struct Funct Bioinform 82(7):1142–1155
95. Das S, Chakrabarti S (2021) Classification and prediction of protein-protein interaction interface using machine learning algorithm. Sci Rep 11(1):1–2
96. Fu L, Niu B, Zhu Z, Wu S, Li W (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 28(23):3150–3152

97. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25(17):3389–3402
98. Krissinel E, Henrick K (2007) Inference of macromolecular assemblies from crystalline state. J Mol Biol 372(3):774–797
99. Krissinel E (2010) Crystal contacts as nature's docking solutions. J Comput Chem 31(1):133–143
100. Jayaraj PB, Ajay MK, Nufail M, Gopakumar G, Jaleel UA (2016) GPURFSCREEN: a GPU based virtual screening tool using random forest classifier. J Cheminform 8(1):1
101. Wang X, Yu B, Ma A, Chen C, Liu B, Ma Q (2019) Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. Bioinformatics 35(14):2395–2402
102. Wei ZS, Han K, Yang JY, Shen HB, Yu DJ (2016) Protein-protein interaction sites prediction by ensembling SVM and sample-weighted random forests. Neurocomputing 193:201–212
103. Sanchez-Garcia R, Sorzano CO, Carazo JM, Segura J (2019) BIPSPI: a method for the prediction of partner-specific protein-protein interfaces. Bioinformatics 35(3):470–477
104. Zhang B, Li J, Quan L, Chen Y, Lü Q (2019) Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. Neurocomputing 357:86–100
105. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
106. LeCun Y, Haffner P, Bottou L, Bengio Y (1999) Object recognition with gradient-based learning. Shape, contour and grouping in computer vision. Springer, Berlin, pp 319–345
107. Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S (2010) Recurrent neural network based language model. In: Interspeech vol 2, p 1045-1048
108. Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D 404:132306
109. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
110. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. Commun ACM 63(11):139–144
111. Pu Y, Gan Z, Henao R, Yuan X, Li C, Stevens A, Carin L (2016) Variational autoencoder for deep learning of images, labels and captions. Adv Neural Inf Process Syst 29:2352–2360
112. Jaderberg M, Simonyan K, Zisserman A (2015) Spatial transformer networks. Adv Neural Inf Process Syst 28:2017–2025
113. Yun S, Jeong M, Kim R, Kang J, Kim HJ (2019) Graph transformer networks. Adv Neural Inf Process Syst 32:11983–11993
114. Callaway E (2020) "It will change everything": DeepMind's AI makes gigantic leap in solving protein structures. Nature 588:203–204
115. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, Wang J, Cong Q, Kinch LN, Schaeffer RD, Millán C (2021) Accurate prediction of protein structures and interactions using a three-track neural network. Science 373(6557):871–876
116. Fukushima K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw 1(2):119–130
117. LeCun Y (1989) Generalization and network design strategies. Connect Perspect 19:143–155
118. LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackel L (1990) Handwritten digit recognition with a back-propagation network. Adv Neural Inf Process Syst 2:396–404
119. Townshend R, Bedi R, Suriana P, Dror R (2019) End-to-end learning on 3d protein structure for interface prediction. Adv Neural Inf Process Syst 32:15642–15651
120. Fout AM (2017) Protein interface prediction using graph convolutional networks (Doctoral dissertation, Colorado State University)
121. Xie Z, Deng X, Shu K (2020) Prediction of protein-protein interaction sites using convolutional neural network and improved data sets. Int J Mol Sci 21(2):467
122. Zhu H, Du X, Yao Y (2020) ConvsPPIS: identifying protein-protein interaction sites by an ensemble convolutional neural network with feature graph. Curr Bioinform 15(4):368–378
123. Hadarovich A, Kalinouski A, Tuzikov AV (2020) Deep learning approach with rotate-shift invariant input to predict protein homodimer structure. In: International symposium on bioinformatics research and applications. pp 296–303
124. Fang B, Bai Y, Li Y (2020) Combining spectral unmixing and 3d/2d dense networks with early-exiting strategy for hyperspectral image classification. Remote Sens 12(5):779
125. Yu C, Han R, Song M, Liu C, Chang CI (2020) A simplified 2D–3D CNN architecture for hyperspectral image classification based on spatial-spectral fusion. IEEE J Sel Top Appl Earth Obs Remote Sens 13:2485–2501
126. Maturana D, Scherer S (2015) Voxnet: a 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 922–928
127. Wang X, Terashi G, Christoffer CW, Zhu M, Kihara D (2020) Protein docking model evaluation by 3D deep convolutional neural networks. Bioinformatics 36(7):2113–2118
128. Zhang Y, Skolnick J (2004) Scoring function for automated assessment of protein structure template quality. Proteins: Struct Funct Bioinform 57(4):702–710
129. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. Preprint at arXiv:1609.02907
130. Liu Y, Yuan H, Cai L, Ji S (2020) Deep learning of high-order interactions for protein interface prediction. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 679–687
131. Cao Y, Shen Y (2020) Energy-based graph convolutional networks for scoring protein docking models. Proteins: Struct Funct Bioinform 88(8):1091–1099
132. Wang X, Flannery ST, Kihara D (2021) Protein docking model evaluation by graph neural networks. Front Mol Biosci 8:402
133. Hawkins-Hooker A, Depardieu F, Baur S, Couairon G, Chen A, Bikard D (2021) Generating functional protein variants with variational autoencoders. PLoS Comput Biol 17(2):e1008736
134. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial networks. Preprint at arXiv:1406.2661
135. Degiacomi MT (2019) Coupling molecular dynamics and deep learning to mine protein conformational space. Structure 27(6):1034–1040
136. Ramaswamy VK, Musson SC, Willcocks CG, Degiacomi MT (2021) Deep learning protein conformational space with convolutions and latent interpolations. Phys Rev X 11(1):011052
137. Nguyen DD, Gao K, Wang M, Wei GW (2020) MathDL: mathematical deep learning for D3R grand challenge 4. J Comput Aided Mol Des 34(2):131–147
138. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
139. Maziarka Ł, Pocha A, Kaczmarczyk J, Rataj K, Danel T, Warchoł M (2020) Mol-CycleGAN: a generative model for molecular optimization. J Cheminform 12(1):1–8
140. Jin W, Barzilay R, Jaakkola T (2018) Junction tree variational autoencoder for molecular graph generation. In: International conference on machine learning, pp 2323–2332

141. Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, Riley P (2018) Tensor field networks: rotation-and translation-equivariant neural networks for 3d point clouds. Preprint at arXiv:1802.08219

142. Eismann S, Townshend RJ, Thomas N, Jagota M, Jing B, Dror RO (2021) Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. Proteins: Struct Funct Bioinform 89(5):493–501

143. Gainza P, Sverrisson F, Monti F, Rodola E, Boscaini D, Bronstein MM, Correia BE (2020) Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nat Methods 17(2):184–92

144. Chen R, Li L, Weng Z (2003) ZDOCK: an initial-stage protein-docking algorithm. Proteins: Struct Funct Bioinform 52(1):80–7

145. Ohue M, Shimoda T, Suzuki S, Matsuzaki Y, Ishida T, Akiyama Y (2014) MEGADOCK 4.0: an ultra-high-performance protein-protein docking software for heterogeneous supercomputers. Bioinformatics 30(22):3281–3283

146. Shimoda T, Suzuki S, Ohue M, Ishida T, Akiyama Y (2015) Protein-protein docking on hardware accelerators: comparison of GPU and MIC architectures. BMC Systems Biology 9:1–10

147. Pons C, Jiménez-González D, González-Álvarez C, Servat H, Cabrera-Benítez D, Aguilar X, Fernández-Recio J (2012) Cell-dock: high-performance protein-protein docking. Bioinformatics 28(18):2394–2396

148. Sukhwani B, Herbordt MC (2009) GPU acceleration of a production molecular docking code. In: Proceedings of 2nd workshop on general purpose processing on graphics processing units, pp 19–27

149. Lensink MF, Velankar S, Kryshtafovych A, Huang SY, Schneidman-Duhovny D, Sali A, Segura J, Fernandez-Fuentes N, Viswanath S, Elber R, Grudinin S (2016) Prediction of homoprotein and heteroprotein complexes by protein docking and template-based modeling: a CASP-CAPRI experiment. Proteins: Struct Funct Bioinform 84:323–348

150. Lensink MF, Velankar S, Baek M, Heo L, Seok C, Wodak SJ (2018) The challenge of modeling protein assemblies: the CASP12-CAPRI experiment. Proteins: Struct Funct Bioinform 86:257–273

151. Lensink MF, Moal IH, Bates PA, Kastritis PL, Melquiond AS, Karaca E, Schmitz C, van Dijk M, Bonvin AM, Eisenstein M, Jiménez-García B (2014) Blind prediction of interfacial water positions in CAPRI. Proteins: Struct Funct Bioinform 82(4):620–632

152. Fleishman SJ, Whitehead TA, Strauch EM, Corn JE, Qin S, Zhou HX, Mitchell JC, Demerdash ON, Takeda-Shitaka M, Terashi G, Moal IH (2011) Community-wide assessment of protein-interface modeling suggests improvements to design methodology. J Mol Biol 414(2):289–302

153. Moretti R, Fleishman SJ, Agius R, Torchala M, Bates PA, Kastritis PL, Rodrigues JP, Trellet M, Bonvin AM, Cui M, Rooman M (2013) Community-wide evaluation of methods for predicting the effect of mutations on protein-protein interactions. Proteins: Struct Funct Bioinform 81(11):1980–1987

154. Lensink MF, Wodak SJ (2013) Docking, scoring, and affinity prediction in CAPRI. Proteins: Struct Funct Bioinform 81(12):2082–2095

155. Lensink MF, Brysbaert G, Nadzirin N, Velankar S, Chaleil RA, Gerguri T, Bates PA, Laine E, Carbone A, Grudinin S, Kong R (2019) Blind prediction of homo-and hetero-protein complexes: the CASP13-CAPRI experiment. Proteins: Struct Funct Bioinform 87(12):1200–1221

156. Zimmermann L, Stephens A, Nam SZ, Rau D, Kübler J, Lozajic M, Gabler F, Söding J, Lupas AN, Alva V (2018) A completely reimplemented MPI bioinformatics toolkit with a new HHpred server at its core. J Mol Biol 430(15):2237–2243

157. Baek M, Park T, Heo L, Park C, Seok C (2017) GalaxyHomomer: a web server for protein homo-oligomer structure prediction from a monomer sequence or structure. Nucleic Acids Res 45(W1):W320–W324

158. Söding J (2005) Protein homology detection by HMM-HMM comparison. Bioinformatics 21(7):951–960

159. Heo L, Lee H, Seok C (2016) GalaxyRefineComplex: refinement of protein-protein complex model structures driven by interface repacking. Sci Rep 6(1):1

160. Remmert M, Biegert A, Hauser A, Söding J (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods 9(2):173–175

161. Yang Y, Zhou Y (2008) Specific interactions for ab initio folding of protein terminal regions with secondary structures. Proteins: Struct Funct Bioinform 72(2):793–803

162. Chen R, Mintseris J, Janin J, Weng Z (2003) A protein-protein docking benchmark. Proteins: Struct Funct Bioinform 52(1):88–91

163. Mintseris J, Wiehe K, Pierce B, Anderson R, Chen R, Janin J, Weng Z (2005) Protein-protein docking benchmark 2.0: an update. Proteins: Struct Funct Bioinform 60(2):214–216

164. Hwang H, Pierce B, Mintseris J, Janin J, Weng Z (2008) Protein-protein docking benchmark version 3.0. Proteins: Struct Funct Bioinform 73(3):705–709

165. Hwang H, Vreven T, Janin J, Weng Z (2010) Protein-protein docking benchmark version 4.0. Proteins: Struct Funct Bioinform 78(15):3111–3114

166. Vreven T, Moal IH, Vangone A, Pierce BG, Kastritis PL, Torchala M, Chaleil R, Jiménez-García B, Bates PA, Fernandez-Recio J, Bonvin AM (2015) Updates to the integrated protein-protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. J Mol Biol 427(19):3031–3041

167. Guest JD, Vreven T, Zhou J, Moal I, Jeliazkov JR, Gray JJ, Weng Z, Pierce BG (2021) An expanded benchmark for antibody-antigen docking and affinity prediction reveals insights into antibody recognition determinants. Structure 29(6):606–621

168. Kundrotas PJ, Anishchenko I, Dauzhenka T, Kotthoff I, Mnevets D, Copeland MM, Vakser IA (2018) Dockground: a comprehensive data resource for modeling of protein complexes. Protein Sci 27(1):172–181

169. Krull F, Korff G, Elghobashi-Meinhardt N, Knapp EW (2015) ProPairs: a data set for protein-protein docking. J Chem Inf Model 55(7):1495–1507

170. Tovchigrechko A, Vakser IA (2006) GRAMM-X public web server for protein-protein docking. Nucleic Acids Res 34(suppl–2):W310–W314

171. Yu J, Guerois R (2016) PPI4DOCK: large scale assessment of the use of homology models in free docking over more than 1000 realistic targets. Bioinformatics 32(24):3760–3767

172. Basu S, Wallner B (2016) DockQ: a quality measure for protein-protein docking models. PLoS ONE 11(8):e0161879

173. Jiménez-García B, Bernadó P, Fernández-Recio J (2020) Structural characterization of protein-protein interactions with pyDockSAXS. Structural bioinformatics. Humana, New York, pp 31–144

174. Quignot C, Rey J, Yu J, Tufféry P, Guerois R, Andreani J (2018) InterEvDock2: an expanded server for protein docking using evolutionary and biological information from homology models and multimeric inputs. Nucleic Acids Res 46(W1):W408–W416

175. Levy Y, Onuchic JN (2004) Water and proteins: a love-hate relationship. Proc Natl Acad Sci 101(10):3325–3326

176. Pavlovicz RE, Park H, DiMaio F (2020) Efficient consideration of coordinated water molecules improves computational protein-protein and protein-ligand docking discrimination. PLoS Comput Biol 16(9):e1008103

177. Parikh HI, Kellogg GE (2014) Intuitive, but not simple: including explicit water molecules in protein-protein docking simulations improves model quality. Proteins: Struct Funct Bioinform 82(6):916–932
178. Liu L, Chen X, Wong KC (2021) Early cancer detection from genome-wide cell-free DNA fragmentation via shuffled frog leaping algorithm and support vector machine. Bioinformatics. https://doi.org/10.1093/bioinformatics/btab236
179. Mahmoudi N, Orouji H, Fallah-Mehdipour E (2016) Integration of shuffled frog leaping algorithm and support vector regression for prediction of water quality parameters. Water Resour Manage 30(7):2195–2211
180. Cozzini P, Kellogg GE, Spyrakis F, Abraham DJ, Costantino G, Emerson A, Fanelli F, Gohlke H, Kuhn LA, Morris GM, Orozco M, Pertinhez TA, Rizzi M, Sotriffer CA (2008) Target flexibility: an emerging consideration in drug discovery and design. J Med Chem 51(20):6237–6255
181. Huber R (1987) Flexibility and rigidity, requirements for the function of proteins and protein pigment complexes. Eleventh Keilin memorial lecture. Biochem Soc Trans 15(6):1009–1020
182. Thompson NC, Greenewald K, Lee K, Manso GF (2020) The computational limits of deep learning. Preprint at arXiv:2007.05558
183. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596(7873):583–589
184. Méndez R, Leplae R, Lensink MF, Wodak SJ (2005) Assessment of CAPRI predictions in rounds 35 shows progress in docking procedures. Proteins: Struct Funct Bioinform 60(2):150–169

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.