Review

Protein–protein interaction prediction with deep learning: A comprehensive review

Farzan Soleymani a, Eric Paquet b,*, Herna Viktor c, Wojtek Michalowski d, Davide Spinello a

a Department of Mechanical Engineering, University of Ottawa, Ottawa, ON, Canada
b National Research Council, 1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
c School of Electrical Engineering and Computer Science, University of Ottawa, ON, Canada
d Telfer School of Management, University of Ottawa, ON, K1N 6N5, Canada

ABSTRACT

Most proteins perform their biological function by interacting with themselves or other molecules. Thus, one may obtain biological insights into protein functions, disease prevalence, and therapy development by identifying protein–protein interactions (PPI). However, finding the interacting and non-interacting protein pairs through experimental approaches is labour-intensive and time-consuming, owing to the variety of proteins. Hence, protein–protein interaction and protein–ligand binding problems have drawn attention in the fields of bioinformatics and computer-aided drug discovery. Deep learning methods paved the way for scientists to predict the 3-D structure of proteins from genomes, predict the functions and attributes of a protein, and modify and design new proteins to provide desired functions. This review focuses on recent deep learning methods applied to problems including predicting protein functions, protein–protein interaction and their sites, protein–ligand binding, and protein design.

Keywords:
Protein–protein interaction
Deep learning
Protein design
Sequence-based
Structure-based

Article history:
Received 22 June 2022
Received in revised form 29 August 2022
Accepted 30 August 2022
Available online 19 September 2022

E-mail addresses:
Fsoleyma@uottawa.ca (F. Soleymani), Eric.Paquet@nrc-cnrc.gc.ca (E. Paquet), hviktor@uottawa.ca (H. Viktor), wojtek@telfer.uottawa.ca (W. Michalowski), dspinell@uottawa.ca (D. Spinello).

https://doi.org/10.1016/j.csbj.2022.08.070
2001-0370/Crown Copyright © 2022 Published by Elsevier B.V. on behalf of Research Network of Computational and Structural Biotechnology. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
1. Introduction

Proteins are organic molecules abundant in living systems and conduct a wide range of unique functions such as transport, storage, membrane composition, and enzymatic action [1], among others. Proteins may interact with DNA, RNA, ligands, and other proteins to carry out cellular and biological functions [2]. The latter occurs by physical interaction between two or more proteins [3,4]. These interactions ought to comply with two conditions: first, the interaction must be by design, i.e., the result of a specific biomolecular event; second, the interaction has evolved to serve a certain non-generic function [3–5]. Thus, one may obtain biological insights into protein functions, disease prevalence, and therapy development by identifying interaction amongst protein pairs [6–8]. Hence, protein–protein interaction (PPI) and protein–ligand binding problems have drawn attention in bioinformatics and computer-aided drug discovery [7,9,10]. Computational methods paved the way for scientists to predict the 3-D structures of proteins from genomes and, hence, to predict their functions and attributes, allowing them to modify proteins and design new ones to target desired functions. However, experimental validation benchmarking remains challenging [11].

Protein–protein interactions compose complexes to conduct numerous biological processes and functions such as metabolic cycles, signal transduction, DNA transcription and replication, catalysis, and immune response [12–18]. The activities of cells and their functions are affected by abnormalities in protein interactions, leading to numerous diseases such as cancer and chronic degenerative diseases [19]. Comprehensive identification of PPIs can help to decode the molecular mechanisms of the specific biological functions involved [19]. The proximity of proteins in PPI is of paramount importance for specific functionality. Despite significant efforts in molecular biology and genomics, the functions of most proteins are not yet established [20–22]. It has been demonstrated by Jansen et al. [23] that the interaction between known and unknown functional proteins can significantly contribute toward deciphering many protein functions. Therefore, predicting PPIs has become a crucial challenge in the field of bioinformatics [19].

PPIs may help in decoding the functionality of unannotated proteins [19,24]. Therefore, many experimental studies have been conducted to identify PPI, among which the yeast two–hybrid [25–27], mass spectrometry [28–32], protein microarrays [33–36] are often used [37,38]. However, these approaches are laborious and time-consuming, which makes them difficult to employ for all protein pairs [9,39,40]. Moreover, the validity of the experimental techniques is highly dependent on how well one implements the assay protocols in target organisms [41]. Therefore, one may use computational methods as pre-treatment in advance of the experimental methods, aiming to reduce false-positive and false-negative results [24,41,42].

A protein comprises a unique linear sequence of amino acids called its primary structure, which determines the folded shape or conformation. The local secondary structure elements, such as strands, helices, and random coils, are created as a result of interactions between the protein backbone, the side-chains, and the environment and extended to the ultimate 3-D structure of the protein [43]. The large number of possible configurations of the peptide backbone, and the desirable chemical bonding geometry and interactions, make the problem of modelling protein structures challenging [17]. This paper reviews recent advances in deep learning methods developed and/or applied to problems, including predicting protein functions, protein–protein interactions and their sites, protein–ligand binding, and protein design.

This review is structured as follows. We first outline protein structure architectures in Section 2. Next, we describe PPI shapes in Section 3. Then, we present some of the main resources for protein structures and sequences in Section 4. Section 5 briefly explains some of the most commonly used deep learning methods, and Section 6 provides an overview of PPI prediction. Section 6.2 and Section 6.3 discuss structure-based PPI prediction methods and their computational solutions, respectively. Sequence-based PPI prediction methods and their associated computational solutions are described in Section 6.4 and Section 6.5. Section 7 reviews deep learning methods addressing protein design problem and Section 8 concludes the review.

2. Protein Structure

Proteins are a broad class of biomolecules forming more than 50% of the dry weight of cells [44]. Their diverse functionality and abundance determine the function and structure of cells, with each protein being an agent performing a specific biological role [44]. Genes are the basic physical and functional units of inheritance and act as instructions to create the proteins that are the agents of biological function. In fact, a unique protein structure is encoded by each gene in cellular DNA, which leads to numerous possible structures [1]. The uniqueness of proteins originates in the amino acid sequences and the bonds that hold them together.

The interaction between proteins is mainly non-covalent [43] except for covalent disulfide bonds (formed by the coupling of two thiol (–SH) groups), between the cysteine amino acid residues of the interacting partner proteins. Hydrogen bonding between proteins in a specific PPI is the most important type of non-covalent interaction.

The main and side-chain atoms of the different amino acid residues are involved in the hydrogen bonding between interacting protein partners. The ion pairs, which form mainly between an acidic and a basic amino acid in the proteins, form the second most important non-covalent interaction between protein partners [45]. The stability of protein structures is also affected by long-range interactions. The impact of short, medium and long-range interactions on various structural classes of proteins are discussed in [46–48].

As stated in [47], the all-α protein class, i.e., the proteins whose secondary structure is completely formed by α-helices apart from a few β-sheets on the edges [49], are governed by medium-range interactions. In contrast, long-range interactions dominate in all-β proteins, in which the secondary structure is mainly composed of β-sheets aside from some α-helices on the edges [49].

The primary to quaternary protein structures are examined in more detail in the following sections.

2.1. Primary structure

The primary structure, as shown in Fig. 1, is a unique, linear, amino acid sequence that forms the backbone of protein. Intramolecular bonding and folding of the linear amino acid chain
eventually establish the protein's three-dimensional shape. The sequence of protein is determined by the gene encoding it, so changing the gene's DNA sequence may alter the protein's amino acid sequence and, thus, the protein's overall structure and function.

Amino acids, the building blocks of proteins, are small organic molecules composed of a central carbon atom, called the \( \alpha \)-carbon, attached to an amino group (-NH\(_2\)), a carboxyl group (-COOH), and a hydrogen atom \([50]\). The carboxyl group is typically deprotonated and carries a negative charge at physiological pH (7.2–7.4) \([51]\), whereas the amino group is typically protonated and shows a positive charge. The identity of each amino acid depends on its R group, which is an atom or group attached to the central atom. For example, the R group of glycine, as shown in Fig. 2 is a hydrogen atom, while the R group of alanine is a methyl group (-CH\(_3\)). Fig. 2 illustrates the twenty common amino acids, each of which has a unique side chain. The side chains govern each acid's chemical behaviour, e.g. whether it is acidic, basic, polar, or nonpolar. Nonpolar amino acids contain aliphatic (hydrocarbon) chains, while polar neutral amino acids contain a hydroxyl (-OH), sulfur, or amide in the R group. Polar acidic amino acids have a carboxylic acid group in the side chain, in addition to the one in the backbone. Polar basic amino acids contain an amine group (which may be neutral or charged) in the side chain, in addition to that in the backbone.

The physicochemical properties of 20 common amino acids are reported in Table 1. SASA represents the solvent-accessible surface area, and the side-chain net charge number is given by NCN \([52]\). These properties help determine the feasibility of protein’s interaction.
Databases for PPI prediction

Table 2

| Name                          | Symbol | Abbr |
|-------------------------------|--------|------|
| Aspartic acid or Asparagine   | A      | Asx  |
| Leucine or Isoleucine         | J      | Xle  |
| Pyrolysine                    | O      | Pyl  |
| Selenocysteine                | U      | Sec  |
| Glutamic acid or Glutamine    | Z      | Glx  |
| unknown amino acid            | X      | Unk  |

Table 3

| Charge     | Positive | H, K, R |
|------------|----------|---------|
| Negative   | D, E     |         |
| Neutral    | A, C, N, P, Q, S, F, G, I, L, M, T, V, W |
| Polarity   | Polar    | Y       |
| Aromaticity| Aliphatic| A, F, I, L, M, P, V, W |
| Aromatic   | I, L, V  |         |
| Size       | Small    | A, C, D, E, Q, R, S, G, K, M, N, P, T |
|           | Medium   | A, G, P, S |
|           | Large    | D, N, T |

Table 4

| Type                     | Database    | Description                                                                 | Last update | URL               |
|--------------------------|-------------|-----------------------------------------------------------------------------|-------------|-------------------|
| Protein–Protein Interactions | STRING [109] | Functional associations between protein pairs, which contains 67,592,464 proteins from 14094 organisms; 20,052,394,042 interactions. | 2021        | https://string-db.org/ |
|                          | IntAct [116] | Contains manually curated datasets (topical), interactomes (for 16 different species) and annotations of experimental evidence. | 2021        | https://www.ebi.ac.uk/interactome/ |
|                          | Biogrid [111] | Contains 2,467,140 protein and genetic interactions, 29,417 chemical interactions and 1,128,339 PTMs from major model organism species. | 2020        | http://www.thebiogrid.org/ |
|                          | DIP [112]    | Experimentally determined PPI database including biological information of proteins, PPIs and experimental techniques for identifying interactions. | 2020        | https://dip.doe-mbi.ucla.edu/dip/Main.cgi |
|                          | Negatome 2.0 [108] | Contains 21,795 interactions, with scores of zero and one, using text mining from literature and analysing protein complexes from PDB. | 2014        | http://negatome.muenchen.de/proj/ppi/ |
|                          | MINT [114]   | Experimentally curated PPI database that includes approximately 117001 PPIs from 607 different species. | 2012        | https://mint.bio.uniroma2.it/ |
|                          | HPRD [115]   | Consists of 41,327 PPIs, 93,710 PTMs, 22,490 Subcellular Localizations and 112,158 Protein Expressions. | 2010        | http://www.hprd.org |
|                          | BIND [113]   | PPIs collected from of humans, yeasts, nematodes, etc. | 2005        | http://download.baderlab.org/BINDTranslation |
| Protein sequences         | UniProt [106] | A collection of protein sequence and functional information, including UniProtKB, UniParc, UniRef and Proteomes. UniProtKB contains 567,483 reviewed (Swiss–Prot)—manually annotated, and 231,354,261 unreviewed (TrEMBL)—computationally analysed, protein sequences. | 2020        | http://www.uniprot.org |
|                          | SWISS-MODEL [107] | A web-based integrated service providing information for protein structure homology modelling. The repository contains 2,217,470 models from SWISS-MODEL for UniProtKB targets, as well as 180,107 structures from PDB with mapping to UniProtKB. | 2020        | https://swissmodel.expasy.org/ |
|                          | PIR [119]    | Integrated protein resources, including protein sequences and high-quality annotations by integrating more than 90 biological databases. | 2022        | http://pir.georgetown.edu/ |
| Higher-level structures   | RCSB PDB [110] | Information about the 3-D structure of proteins, nucleic acids, and complex assemblies. 191144 structures, 57340 human sequence structures, and 14406 nucleic acid-containing Structures | 2021        | https://www.rcsb.org/ |
|                          | SCOP [122]   | Classification of known proteins and a comprehensive description of the structural and evolutionary relationships between them. As of 2022-05-30, this dataset contains 72,448 non-redundant domains, representing 858,316 protein structures. | 2022        | http://scop.mrc-lmb.cam.ac.uk/scop |
| Genomic information       | CGD [124]    | A resource for genomic sequence data, genes and protein information for Candida albicans and related species. | 2022        | http://www.candidagenome.org/ |

The nonstandard amino acids [65–67]. These nonstandard amino acids are presented in Table 2. Sometimes it is not possible to differentiate two closely related amino acids. Therefore, we have the indeterminate residues in protein sequences as represented by symbols B, J, Z and X.

One way to classify amino acids is based on the side chains, as shown in Table 3 [68,69], in which case.

Beyond the common amino acids shown in Table 1, there are also nonstandard amino acids [61]. These are also known as biosynthetic amino acids, and require complex synthetic and translational mechanisms that differ from the canonical enzymatic system used for the 20 standard amino acids [62] namely, pyrolysine [63] and selenocysteine [64]. These nonstandard amino acids are presented in Table 2. Sometimes it is not possible to differentiate two closely related amino acids. Therefore, we have the indeterminate residues in protein sequences as represented by symbols B, J, Z and X.

The physicochemical attributes of amino acids, such as hydrophathy [53], isoelectric, the pH at which the molecule carries no net charge [54,55], and charge, play crucial roles in identifying the interaction between protein sequences [56].

The common amino acids shown in Table 1, there are also nonstandard amino acids [61]. These are also known as biosynthetic amino acids, and require complex synthetic and translational mechanisms that differ from the canonical enzymatic system used for the 20 standard amino acids [62]) namely, pyrolysine [63] and selenocysteine [64]. These nonstandard amino acids are presented in Table 2. Sometimes it is not possible to differentiate two closely related amino acids. Therefore, we have the indeterminate residues in protein sequences as represented by symbols B, J, Z and X.

One way to classify amino acids is based on the side chains, as shown in Table 3 [68,69], in which case.

Multiple amino acids are linked together by peptide bonds, forming a long chain called the polypeptide. The order of the amino acids determines the polypeptide’s functionality. Polypeptides are classified by the number of amino acid units in the chain. Each amino acid is linked covalently to its neighbours by peptide bonds, in a dehydration synthesis (condensation) reaction. Each protein is composed of one or more polypeptide chains. During protein synthesis, the carboxyl group (–COOH) of the amino acid at the end of the growing polypeptide chain reacts with the amino group of an incoming amino acid, forging a peptide bond and releasing a water molecule. Peptide bonds connect the carbon of the carboxyl group of one amino acid to the nitrogen of the amino group of the next, as shown in Fig. 3.
Polypeptide chains are directional, i.e. its ends are chemically distinct from one another. The end with a free amino group is called the amino terminus or N-terminus, while the other end has a free carboxyl group, and is known as the carboxyl terminus or C-terminus (see Fig. 3).

Most of the side chains are nonpolar, several are positively or negatively charged, some are polar but not charged. These features, and their consequent bonds, are responsible for protein structure and functionality by maintaining the protein in a specific shape or conformation. The polar side chains can form hydrogen bonds, while the charged side chains can form ionic bonds. Hydrophobic side chains interact via van der Waals interactions [1]. Consequently, protein folding is directed by the side-chain interactions, the sequence and the location of amino acids in that protein. The order of the acids, i.e. the primary structure, determines which bond types can form at each location along the polypeptide, and thus governs the protein’s tertiary structures [70].

2.2. Secondary structure

Secondary structures result from interactions between parts of the polypeptide chain. The most common folding patterns are α-helices and β-pleated sheets [44].

In an α-helix, the hydrogen bonding occurs between the carbonyl group (C = O) of one amino acid and the hydrogen atom of the amino acid four places further along the chain. This bonding pattern draws the polypeptide chain into a helix, with each turn of containing 3.6 amino acids. The R groups stick outwards from the α-helix, and are free to interact. In a β-pleated sheet, segments of a polypeptide chain align next to each other, making a sheet structure coupled by hydrogen bonds between carbonyl and amino groups of backbone, while the R groups extend above and below the plane of the sheet.

The strands of a β-pleated sheet may be parallel (i.e. their N- and C-termini match up), or anti-parallel (i.e. the N-terminus of one strand alongside the C-terminus of the next). In certain cases, the amino acids are not found in α-helices or β-pleated sheets. For instance, proline is known as a “helix breaker” owing to its unusual R group, which bonds to the amino group to form a ring creating a bend in the chain that prevents helix formation. Proline is generally found in bends, unstructured regions between secondary structures. Proteins can contain α-helices, β-pleated sheets or both, or may form neither type.

2.3. Tertiary structure

The tertiary structure, as shown in Fig. 5, is formed as the polypeptide chains of protein molecules fold into a more compact shape with a low surface-to-volume ratio. The tertiary structure results mainly from electrostatic forces between the R groups. For instance, oppositely charged R groups bond ionically, while similarly charged R groups repel one another. Similarly, polar R groups may form hydrogen bonds and other dipole–dipole interactions.

A cluster of amino acids with nonpolar, hydrophobic R groups on the inside of the protein leaves the hydrophilic amino acids on the outside to interact with nearby water molecules.

The tertiary structure can also be produced by disulfide bonds. Disulfide bonds are covalent and hence keep parts of the polypeptide firmly attached to each other [44]. A synthesis of a tertiary structure is portrayed in Fig. 6.

2.4. Quaternary structure

Many proteins comprise two or more polypeptide chains that interact to form a stable folded structure, known as a subunit of...
the protein. The amino acid sequences of each subunit can either be identical (as in tobacco mosaic virus protein), similar (as in the α and β chains of hemoglobin), or entirely different (as in aspartate transcarbamoylase see Fig. 7). Subunit arrangement establishes the protein’s quaternary structure.

In the following section, protein shapes are discussed.

3. Protein Shapes

Proteins may be classified on shape and solubility into three global classes: fibrous (Fig. 8), globular (Fig. 9), or membrane (Fig. 10).

In general, fibrous proteins have relatively simple, regular linear structures, and often provide cells with structural functions.
Fibrous proteins are usually insoluble in water and dilute salt solutions. A well-known example of such proteins is collagen, abundant in all animals [71]. As illustrated in Fig. 8, collagen is composed of three chains, each containing 1400 amino acids, twisted together into a triple helix. Glycine appears in every third position along each chain, and, due to its small size, it perfectly fits inside the helix. Proline and hydroxyproline [72] fill numerous positions on a chain. There are numerous types of collagen, all comprising a long stretch of triple helix attached to different ends.

On the other hand, globular proteins are nearly spherical, as shown in Fig. 9 and very soluble in aqueous solutions. Examples include haemoglobin in the red blood cells, that binds to oxygen.

Membrane proteins have hydrophobic side chains directed outwards, and interact with the nonpolar phase within membranes. Therefore, membrane proteins are insoluble in aqueous solutions but can be solubilised in solutions of detergents. Bacteriorhodopsin (Fig. 10) represents an example of such proteins which is made by halophilic (salt-loving) bacteria. This protein pumps protons across cell membranes, powered by sunlight [44].

PPI prediction and protein design may benefit from classifying deformable protein shapes. A novel classification method for protein shapes, based on their macromolecular surfaces, is introduced in [73]. They proposed a novel description, based on bifractional Fokker–Planck and Dirac–Kähler equations for deformable shapes.

3.1. Protein folding

Over the past two decades, considerable efforts have been made in the protein design field, which has further expanded due to the evolution of computational methods and machine learning algorithms. Some of the successful examples include novel folds in protein design [74,75], enzymes [76,77], antibodies [78–80], vaccines [78,81], ligand-binding proteins [82,83], protein assemblies [84–88], and membrane proteins [89–91]. Some of the most recent comprehensive reviews in this field are presented in [92–95]. Generally, the backbone structure of a target protein forms the input for computational protein design. An optimal sequence can be generated using computational sampling methods, seeking potential folding into the desired structure for experimental validation.

A vital component of the solution process involves the scoring function, which can distinguish folds that are or are not physically compatible with a given amino acid sequence [96]. One approach to defining the scoring function considers van der Waals and electrostatic energy along with knowledge-based terms such as backbone dihedral preference statistics about protein structures [97,98], and side-chain rotamers [99]. There is a gap between automated protein design and current approaches, which mostly depend on human experience. This is due to restrictions on artificially created sequences which must comply with various factors such as in silico folding free-energy landscape [100,101] and shape complementarity [87].

Despite the rapidly growing number of known protein structures, the number of unique protein folds is converging, suggesting
that statistical learning based on existing structures leads to progress in design methods [102–104]. This statistical potential enables machine learning, especially deep-learning neural networks, to be used for accurate prediction and feature extraction [105].

Some of the commonly used resources for the structure of proteins and sequence are discussed in the following section.

4. PPI Databases

There are several known PPI databases, such as Uniprot [106], SWISS-MODEL [107], Negatome 2.0 [108], STRING [109], RCSB PDB [110], BioGRID [111], DIP [112], BIND [113], MINT [114], HPRD [115] and IntAct [116]. However, among these databases, some are not currently being maintained, such as BIND and HPRD, and are thus rarely used [117]. STRING, IntAct and MINT provide interaction scores from different sources to indicate their reliability. The Negatome 2.0 dataset comprises the manually curated interacting protein pairs from literature and analysed protein complexes from PDB, with scores of zero and one to indicate non-interacting and interacting pairs [108].

Computational methods often use the proteins’ biological information, including protein sequences and protein structures. The biological characteristics and high-level structure of proteins are affected significantly by their primary structure. Therefore, one may use the knowledge extracted from protein sequences to estimate the interaction likelihood between protein pairs [118]. Protein sequences can be obtained from the STRING [109], PDB [110], Uniprot [106], PIR [119], SWISS-MODEL [107], and TrEMBL [120] databases. Information on higher-level protein structures can be acquired from PDB [121] and SCOP [122]. Dandekar et al. have asserted that proteins encoded by conserved gene pairs physically interact [123]. That basis is used, in genomic-based computational methods, for prediction. Genomic information can be found in The Candida Genome Database (CGD) [124].

In the following section, some of deep learning methods are briefly explained.

5. Deep Learning Models

Autoencoders, as illustrated in Fig. 11, are a type of unsupervised feedforward neural network reconstructing the output from the high-dimensional and possibly correlated input feature space [125]. It consists of two parts, the encoder and the decoder. The encoder maps the input data into a low-dimensional and uncorrelated features space, called the latent layer, while the decoder reconstructs the input data from the latent layer. Autoencoders remove redundancies and correlations while extracting highly informative features [126,127].

Recurrent neural networks (RNNs) can capture contextual information when mapping input to output sequences. However, RNNs often suffer from vanishing gradients, limiting the context range they can access [128,129]. To address this problem, long-short term memory (LSTM) architecture was introduced [130], as illustrated in Fig. 12.

The LSTM architecture consists of recurrently connected memory blocks and corresponding control gates, the forget gate $f_t$, the input gate $i_t$, and the output gate $o_t$, which update and control the cell states [131]. The input and forget gates control current network memory and the flow of new information. Specifically, as new information flows into the network, the forget gate manages the information that needs to be removed from cell states, while the input gate controls the information that needs to be stored in cell states. Finally, the output gate determines the encoded information that needs to be forwarded as the input for the next step.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$h_t = o_t \odot \tanh(S_t)$$

where $\sigma$ is the sigmoid activation function, $W$ is the weight matrix, $b$ is the bias vector, and $\odot$ is the point-wise product. The initial operation is performed by the forget gate $f_t$, Eq. 1a, which determines whether the information should be kept or removed. The LSTM
The architecture contains a hidden state $h_t$, Eq. 1b that is formed by sequential information.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$  \hspace{1cm} (2a)

$$\tilde{S}_t = \sigma(W_c \cdot [h_{t-1}, x_t] + b_c)$$  \hspace{1cm} (2b)

The next step involves storing the new input information in the cell state via the input gate $i_t$, Eq. 2a. Therefore, the cell state can be modified through candidate values $\tilde{S}_t$, Eq. 2b, and Eq. 3a. Finally, the LSTM determines the output of each unit as Eq. 3b.

$$S_t = f_t \odot S_{t-1} + i_t \odot \tilde{S}_t$$  \hspace{1cm} (3a)

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$  \hspace{1cm} (3b)

Despite the many advantages of LSTM, it is a computationally demanding architecture and slow to train.

The convolutional neural network (CNN) architecture is illustrated in Fig. 13. Its input is a matrix of the encoded representation of two proteins stacked in two columns. In this example, the CNN architecture comprises three 2-D convolutional layers and three dense layers. The convolutional and max-pooling layers reduce the size of the input tensor. The dropout layers are used to reduce overfitting and improve generalisation error [132]. The flatten layer reduces the dimensionality of the input. In addition, three densely connected layers reduce the features to the desired size. Finally, the output is obtained from a densely connected layer with the softmax activation function, classifying interactions into interacting and noninteracting pairs.

Most data used in deep learning can be readily represented in Euclidean space [133], where the convolution operation is properly defined [134]. However, when data cannot be represented on a regular grid due to the complex nature of their correlations [135,136], standard convolution cannot be directly applied to non-Euclidean geometries, limiting the applicability of CNNs [137,138].

However, the convolution theorem states that [137] convolution may be evaluated using Fourier transform. The Fourier transform is first performed for both the input and the filter. Then, both transformations are multiplied by the Hadamard product. Finally, the inverse Fourier transform of the Hadamard product is evaluated. If the Fourier transform is defined correctly, the convolution theorem remains valid under non-Euclidean geometry [137,134], allowing the application of CNNs to non-Euclidean geometries [138]. The spectral graph convolution in the non-Euclidean domain can be obtained by applying the Fourier transform graph and convolution theorem to both the input signal and the convolving filter [139].

Graph convolution extracts underlying local information by collecting node information in the local neighbourhood. Localisation can be achieved by expressing the filters in terms of Chebyshev polynomials of the first kind [140,141]. Fig. 14 illustrates a graph convolutional network (GCN) with stacked layers to extract multi-scale substructure features [138]. The propagation rule for the multi-layer GCN is given by:

$$f_i^{l+1} = \sigma \left( \Phi \mathbf{G}_l \phi ^l f_i^l \right), \quad i = 1, \ldots, q, \quad j = 1, \ldots, p$$  \hspace{1cm} (4)

Where $\sigma$ is the nonlinear activation function ($l$) and $\mathbf{G} = \text{diag}(g_i(i))$. The number of features is denoted by ($q$), and the number of assets is denoted by ($p$).

Locality is assumed for all nodes in GCN. As the size of the neighbourhood increases, algorithmic time and space complexity also increase [142]. This issue violates the purpose of using deep
models. While few studies have addressed this issue (e.g., skip connection-based models), how to construct a deep architecture that can better adaptively exploit deeper structural graph patterns is still an open challenge [136].

Generative models aim to model the underlying distribution of the data, enabling the generation of new samples with comparable properties to those on which the model was trained [143,144]. Numerous generative models have been developed on the basis of deep neural networks, such as Variational Autoencoder (VAE) [145–147], Generative adversarial Network (GAN) [148], and deep autoregressive models [149–151].

In their original form, GAN algorithms are composed of two components, namely, generator and discriminator, with the generator producing synthetic data while the discriminator evaluates the discrepancy between the generated data and the real data. Each network attempts to improve its performance until an equilibrium is reached, where the discriminator is unable to detect the fake samples and the generator fails to produce better samples [148,152–154].

As illustrated in Fig. 15, given a data distribution, \( x \sim p_x, x \in \mathcal{X} \), the generator learns the distribution \( p_G \) which maps the latent variable drawn from a prior distribution \( z \sim p_z, z \in \mathcal{Z} \) to the sample space as \( G : \mathcal{Z} \rightarrow \mathcal{X} \), while the discriminator is trained to distinguish between fake and real samples via a score \( D : \mathcal{X} \rightarrow \mathbb{R} \) [148].

The VAE architecture, as shown in Fig. 16, is a class of generative models based on variational Bayesian inference with multivariate prior distribution [155–157], initially introduced in [145]. The VAEs comprise two linked models that are individually parameterized, namely the encoder or recognition model and the decoder or generative model. Unlike autoencoders, in which the encoder compresses the input features into real-valued latent features, the encoder in a VAE stochastically maps the observed variables’ \( x \)-space to a probabilistic latent \( z \)-space (latent variable) [158].

Fractionally strided convolutions, also known as transposed convolutions, perform a reverse spatial transformation by switching the forward and backward pass [159]. Fractionally strided convolutions may allow for recovering the shape of the initial feature map but do not guarantee retrieving the input itself [159]. This allows the network to learn its own spatial downsampling and upsampling. An extension of the 2-D GAN framework, called conditional GAN, has been proposed in [160] that applies conditions on class labels for both the generator and the discriminator networks. Multimodal data generation is better represented using conditional GANs.

Both generator and discriminator are trained based on an additional information placed as condition \( y \) in the input layer, as depicted in Fig. 17. The adversarial training framework allows for flexible-joint hidden representations composed from input noise \( p_z(x) \) and \( y \) in the generator [160]. The fake samples are generated as \( G(x, z) = \mathbf{x}'|y \) (\( \mathbf{x}' \) is synthetic sample given \( y \) as a condition) aiming to resemble real samples as well as possible. The discriminator receives real samples with labels \( (x, y) \) and fake samples from generator \( (x', y, y) \). The discriminator outputs a single probability through a sigmoid activation function (\( \sigma \)) indicating its decision on fake and real inputs.

The following section represents PPI prediction methods.
facilitated the generation of synthetic protein domains which
gereliability [177–180]. Nevertheless, computational methods have
protein sequence, it is difficult to determine the source of the poor
performance of sampling algorithms for jointly exploring the pro-
accuracy of complex energy functions for protein physics and the
designs frequently fail, entailing multiple trial-and-error cycles
great achievements, current approaches are unreliable as initial
[87], and enzymes [169]. Even though these methods have shown
so as in the design of novel 3-D folds [74], protein complexes
enzymes [76,77,169], the development of new therapies
[170,171], biosensors [172], and small-molecule binders [82]. However, these methods are mainly suited to modifying naturally
enzymes [76,77,169], biosensors [172], and small-molecule binders [82]. However, these methods are mainly suited to modifying naturally
found proteins [173]. On the other hand, creating proteins de novo provides full control over their structure and function [92,174]. Hence, a new objective is to discover new, non-native folds or structural elements as building blocks for novel proteins [173].

Computational protein design mainly aims to automate the fab-
rication of proteins with specific structural and functional properties [9,73]. This field has gained traction in the past two decades, such as in the design of novel 3-D folds [74], protein complexes [87], and enzymes [169]. Even though these methods have shown great achievements, current approaches are unreliable as initial designs frequently fail, entailing multiple trial-and-error cycles [175,176]. Since these approaches are highly dependent on the accuracy of complex energy functions for protein physics and the performance of sampling algorithms for jointly exploring the protein sequence, it is difficult to determine the source of the poor reliability [177–180]. Nevertheless, computational methods have facilitated the generation of synthetic protein domains which mimic natural folds using sequences unlike those in nature [181–183].

Quick computational testing of many possible outcomes, poten-
tially narrowing the set of necessary experiments, would ulti-
mately save time. Among the computational methods addressing the PPI prediction problem, some use extracted features as inputs to learn the model [184], while others extract new protein information [185–187]. These methods are further explained in Section 6.5. The information extracted from a tertiary structure of proteins may be used to predict PPI. There exist several experimental techniques for determining a tertiary structure of proteins, including X-ray crystallography and NMR spectroscopy [188]. It is suggested in [189] that locations of protein–protein binding sites are engraved in the proteins’ structures. Although experimentally determined 3-D protein structures may facilitate the detection of interaction sites and the understanding of protein functions, experimental biological methods are laborious and time-consuming, and consequently, the geometries of only a small fraction of known proteins have been determined [189–194]. To address this short-
coming, various studies use deep learning to predict, from protein structure and other protein features, potential PPI [185,186,195–197]. Some of these methods are discussed further in Section 6.3.

6.1. PPI Site Prediction

Identifying PPI sites is crucial for understanding the mecha-
nisms of disease and for novel drug design. PPI binding sites consist of amino acid residues forming chemical bonds with a part of another molecule [40]. Identifying interaction domains in sequences helps in understanding cell regulatory mechanisms, locating drug targets and predicting protein functions [198]. Yuan et al. have addressed PPI site prediction as a graph node classification problem, modelling proteins as undirected graphs. They de-
veloped GraphPPIS [199] to predict PPI sites.

PPI site predictions are roughly categorised into three cate-
gories: protein–protein docking, structure-based, and sequence-
based methods. Docking methods aim to generate structures of the resulting protein complex [200], as proposed in [201], by defining a scoring function for novel shape complementarity at the initial docking stage. Some of the recent sequence-based methods for predicting protein–protein interaction sites include: attention-
based convolutional neural networks [197], simplified LSTM [191], the DeepPPISP method which uses a combination of local contextual and global sequence features [196], CNN with a residue binding propensity to address data imbalance [202], and the DEL-
PHI method which comprises an ensemble structure as a combina-
tion of CNN and recurrent neural network (RNN) [203]. CNN and LSTM architectures are illustrated in Fig. 13 and 12 respectively.

Structure-based methods for PPI prediction are addressed in the next subsection.

6.2. Structure-based PPI Prediction

Proteins adopt complex 3-D structures to perform biological functions via physical contact between effectors and regulators. The effectors may be characterised as the molecules that activate or suppress the regulator’s function and alter gene expression as a result [204,205]. Therefore, predicting which residues are involved in PPIs may help structure-based drug discovery, improve the accuracy of protein–protein docking, and obtain richer annotation of protein function [206,207]. A protein may interact with multiple partners over different or overlapping sections of its surface. These interactions may occur at different times or, when the interaction site is large, simultaneously.

Structure-based methods exploit information such as similarity in protein structure to predict PPIs [208]. For instance, two pro-
teins, A’ and B’, structured similarly to two interacting proteins A
and B, respectively, can be assumed also interact with each other
[209]. Structure-based techniques often employ empirical scoring functions, physics-based methods, knowledge-based approaches,
or quantitative structure–activity relationship methods to determine both the binding affinities and structural orientations of PPIs [210]. Protein–protein docking techniques can model the orientation of two interacting proteins and their binding affinity and identify key residues in PPIs [210].

Docking-based methods use the structures of individual proteins to predict the structure of the complex. Generally, the only information available is the structure of these individual proteins. The docking method includes two steps. Firstly, the binding orientations of two interacting proteins are identified. Secondly, the binding free energy between the interacting proteins is estimated [211,212]. A global search is conducted by holding the target protein (receptor) stationary while moving the ligand around it. After modelling all possible orientations, the interactions between the two proteins are determined [210]. The global search method demands an unlimited number of translations and rotations, making it a computationally expensive approach. To address this issue, a fast Fourier transform (FFT) approximation has been used in [213].

The local docking technique may improve solutions found by the global docking approach. In the global docking scenario, the sampling starts from a random point, whereas the local techniques assume a known starting point (binding mode) and restrict the sampling search around it [214,215]. The ZDOCK server is among the commonly adopted docking resources which employ FFT-based global search [216]. RosettaDock is a local protein–protein docking algorithm based on a Monte Carlo search. It allows for user-defined initial poses or random orientation of the two proteins. RosettaDock aims to find the system with the lowest energy, initially through a low-resolution optimisation, followed by a high-resolution refinement [217]. Finally, the docking score is estimated by an all-atom energy function [218–220]. The structural features and physicochemical properties are used for showing the models of unknown PPIs. MEGADOCK is a template-free docking methods [221] identify the most promising interactions from a large set of potential interaction sites by assessing the unbound protein components. This method investigates a protein docking approach based on the tertiary structures of the target proteins and physicochemical properties. The docking calculation is accelerated using a novel scoring function called the real Pairwise Shape Comple-mentarity (rPSC) score. Although docking methods have proven successful for some proteins, they fail to deliver the same performance for proteins that sustain conformational changes during interaction [222].

Homologous proteins, i.e. proteins exhibiting similarity through common ancestors sequences [223], are apt to adopt the same binding interfaces [224]. However, the PPI interfaces may be structurally similar, even though their global structures differ [225]. Template-based docking techniques predict PPIs by comparing a protein–protein complex under examination against templates, i.e. other, experimentally determined, protein–protein complex structures [168,211,226–229]. In general, these techniques operate in five steps i) developing the template library, (ii) selecting the target set, (iii) searching for the similarities between target and template, (iv) refinement and (v) scoring. Developing the template library is the most crucial step. In the last decade, the number of experimentally determined structures has grown exponentially, thus improving the performance of template-based techniques [210].

The search for similarity between target and template is performed globally and locally, and can be conducted through sequence alignment [230], structural alignment, and threading [231–234]. Alignments can be obtained from sequences, structures, or feature information from both sequences and structures. The structure framework in the aligned regions of the template with the highest alignment score is selected as the basis of the target protein structure [222,235] (See Fig. 18).

The next subsection will review computational methods predicting PPI using protein 3-D structure.

### 6.3. Structure-based PPI Prediction Using Computational Methods

In order to address the high-dimensionality problem of protein structure, several dimensional reduction techniques have been applied, such as random forests [237] and the support vector machine (SVM) and its derivatives [238,239]. Northey et al. introduced a multi-layer perception network (MLP)-based method called IntPred [240] to predict interaction by splitting proteins into a group of patches that integrates 3-D structural information into a feature set.

In recent years, many graph convolutional network (GCN) variants (see Fig. 14) [134,241] have been successfully employed in a variety of tasks with graph-structured data [242], such as protein solubility prediction [243], genomic analysis [244] and drug discovery [245]. A GCN-based approach is proposed in [246] to acquire positional information in PPIs. Their representation method combines the information from the amino acid sequence and the protein positions. In order to determine the amino acids of an interacting protein interface, Fout et al. integrated 3-D structures into a GCN [247]. To accurately predict interactions between query proteins entirely from 3-D structural data, Baranwal et al. proposed a GCN-based mutual attention classifier called Struct2Graph [248]. The generative model proposed in [177] is a graph-based model which captures the joint distribution of the full protein sequence, which is founded on long-range interactions resulting from the protein structure. A multimodal approach based on LSTM is proposed in [187], which predicts PPI by integrating structural and sequential information about proteins into the input feature set.

The advantages and disadvantages of these methods are listed in Table 5. Moreover, Table 6 lists the datasets used by each method.
The sequence-based methods for PPI prediction are reviewed in the next section.

6.4. Sequence-based PPI Prediction

Traditional methods often analyse protein sequences based on multiple sequence alignments. This leads to a simple inference of functional and structural constraints from sequence data [265]. While protein design and engineering have benefited from evolutionary information of alignments [266–268], adding distant proteins will induce large and unreliable alignments [269], restricting the diversity of sequences.

Unlike docking and structure-based methods, sequence-based methods do not require structural data and instead leverage the abundance of existing protein sequence data from sequencing technology, especially since the introduction of metagenomics [106,270,271]. One may predict PPIs from amino acid sequence similarity in the known interactions, depending on interactions already identified in one species to infer interaction in different species [24,272]. The sequence-based method hence focuses on primary structure, disregarding the protein’s 3-D shape [273].

In domain-based technique, specific sequences in the protein structure are represented by conserved domains. Conserved Domains may be defined by local multiple sequence alignments, including a wide range of organisms to display sequence regions

| Framework | Description | Advantage | Disadvantage |
|-----------|-------------|-----------|--------------|
| GCN-based [2017] [247] | This study proposed a pairwise classification architecture in which one or more graph convolution layers process the neighbourhood of a residue in each protein. Then, the representation of two residues is paired and passed through a dense layer for classification. This study analysed several GCN-based methods, concluding that neighbourhood-based convolution methods outperform diffusion-based convolution and SVM-based methods. | The proposed convolution operators and obtained features may be helpful for other applications, including protein function, catalytic and other functional residues, and protein interactions with DNA and RNA. | The accuracy of this approach is examined based on a limited number of labelled training examples. |
| IntPred [2018] [240] | This method uses a random forest to predict protein–protein interface sites at both the surface patch and residue levels. | The performance of a binary classifier can be evaluated using different measurements, such as the Matthews’ correlation coefficient (MCC), sensitivity, precision, and specificity [240]. IntPred outperformed the methods Promate [189], PIER [249], PINUP [250], and meta-PPISP [251], but not SPPIDER [252], based on MCC. | The performance of this method depends on the application. For instance, IntPred was better suited to cases in which false positives are less well tolerated than false negatives. |
| Graph-based generative model [2019] [177] | This method uses a graph transformer model for designing protein sequences given graph representations of 3-D protein structures, leveraging the spatial locality of dependencies in molecular structures. | | |
| Struct2Graph [2020] [248] | In this method, graph embeddings of each protein are obtained using an assigned GCN. Next, relevant geometric features associated with query protein pairs are extracted using a mutual attention network. Finally, a feedforward neural network performs a binary classification between interacting and noninteracting pairs. | Struct2Graph only uses 3-D structural information to predict the PPI. They have reported state-of-the-art performance on both balanced and unbalanced datasets. | Limited availability of 3-D structural information may restrict the applicability of this method. |
| LSTM-based [2020] [187] | The proposed method integrates the 3-D structure and sequence-based information of proteins to predict PPIs. The 3-D coordinate information, hydrophathy index, isoelectric point, and amino acid charges of each protein are fed into a pre-trained ResNet50 model to extract features from these attributes. A stacked autoencoder obtains the compact form of encoded proteins using autocovariance and conjoint triad. The structural features from ResNet50 are passed through LSTM and concatenated with features from the stacked autoencoder. The merged features are then fed into the classifier to predict protein pair labels. | This method performs well despite being trained on a low number of instances. | Limited availability of 3-D structural information may restrict the applicability of this method. Additionally, LSTM models are computationally demanding and slow. |
containing the same, or comparable, patterns of amino acids [274]. This idea can be used to predict the subcellular location of the protein and the class and subclass of the enzyme, to find functional interactions, and to identify the membrane protein [275]. Computational approaches have been developed to predict PPIs based on the information this technique provides [275]. A domain-based method to estimate the interaction map of E.coli, for example, is developed in [276]. Another domain-based method is introduced by Kim et al. [277] which estimates the probability of the interaction between interacted domains. Using the relevant vector machine (RVR) and domain features with support vector regression (SVR), Kamada et al. identified PPIs [278]. DomainGA is a multi-parameter optimisation approach which is developed to predict the score of PPI [279].

The ortholog-based techniques also use similarities between amino acid sequences [210]. The annotations to a functionally determined protein sequence are transferred to similar parts of a target sequence. This work relies on databases of annotated proteins to construct the homologous model of the studied protein [280]. Significant sequence similarities may be shared among multiple proteins from an organism in systems in diverse organisms. Thus, if a significant similarity is found between an input protein and an annotated protein (with known functions), the input protein may be hypothesised to possess similar properties or functions. In order to identify these functions, paralog and ortholog approaches are employed. Orthologs are homologous genes that evolved by vertical descent from a single ancestral gene; in contrast, paralogs evolved by duplication [223,281]. For instance, the orthologs of two interacting proteins, A and B, can interact similarly in different species [210].

Computational methods predicting PPI using protein sequence are addressed in the next subsection.

6.5. Sequence-based PPI Prediction Using Computational Methods

The Interface Weighted RAptor (iWRAP) integrates a boosting classifier with a novel linear programming formulation for interface alignment to predict interacting proteins encoded by the entire yeast genome. The interface profiles are constructed using SCOPPI [282], based on the sequence and structural similarity of the interface [283]. The Universal In Silico Predictor of Protein–protein Interactions (UNISPPI) uses the primary structure to classify protein pairs as interacting or non-interacting [284]. A matrix-based representation of protein sequence coupled with the SVM algorithm is proposed in [285], using the order of primary structure and its dipetide information. The sequence-based methods may be split into two distinctive techniques: domain-based and ortholog-based [273]. PPI prediction is conducted based on sequences in [286] by defining units of three adjacent amino acids and measuring the frequency of those units in a protein sequence. Other methods such as amino acid index distribution [287], conjoint triad method (CT) [288] and autocovariance (AC) [288] are used to extract features such as locations of amino acids, frequencies and physicochemical properties, with the aim of representing a protein sequence.

An SVM based approach, ACT-SVM, is developed in [289] to extract features from protein sequences as the input vector for the classifier. A sequence-based human PPI prediction is developed in [18] based on a Stacked autoencoder (SAE). Another sequence-based PPI prediction approach is introduced in [165] called D-
Table 7
Summary of advantages and disadvantages of sequence-based Deep Learning methods regarding PPI prediction.

| Framework                | Description                                                                 | Advantage                                                                 | Disadvantage                                                                 |
|--------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| SVM-conjoint triad       | Each protein sequence was represented in this study by a vector of amino acid features. The model was developed based on a support vector machine (SVM) integrated with a kernel function and a conjoint triad feature for describing amino acids. This method mapped different types of PPI networks using only sequence information, which could be applied to explore networks for any newly discovered protein with unknown biological relationships. They suggested that methods without local environments for amino acids are often unreliable, so a conjoint triad method was used. | The 20 standard amino acids were clustered into several classes based on their dipoles and side chain volumes to achieve dimensionality reduction of the vector space. This method might predict PPI networks created by pairwise PPIs. | The limited available information on protein pairs restricts the applicability of this method. Additionally, it mainly considers the properties of two nearby amino acids, overlooking long-range interactions. |
| SVM-autocovariance       | This method combined a new feature representation using autocovariance (AC) and a support vector machine (SVM). AC considers the interactions between more distant amino acids in the protein sequence, specifically long-range interactions. This is an improvement over the method proposed in [286]. | The conjoint triad (CT) method only considered the attributes of an amino acid and its two neighbouring amino acids [286], while long-range interactions are accounted for by the AC method. | The model achieved a low prediction accuracy of 58.42% in a negative dataset created using the Prcp method [286]. |
| UNISPPI [2013]           | This method used a decision tree model, predicting PPIs using only 20 amino acid frequency combinations from interacting and noninteracting proteins as learning features. This study indicated that asparagine, cysteine, and isoleucine frequencies are important features for discriminating between interacting and noninteracting protein pairs. | This method extracts more information hidden in protein primary sequences than amino acid dipeptide composition. | Instances with a classification score of 0.50 were classified as neither PPIs nor non-PPIs, limiting the applicability of this method. Additionally, the obtained accuracy of 79.4% for interacting and 72.6% for noninteracting pairs are relatively low. |
| SVM-based method [2015]  | PPI prediction was addressed by integrating a support vector machine (SVM) and a novel matrix-based representation of the sequence order and dipeptide information of the primary protein sequence, extracting more information than amino acid dipeptide composition. The SVM classified the interaction between protein pairs using these feature vectors. | This method extracts more information hidden in protein primary sequences than amino acid dipeptide composition. | SVM algorithms performed relatively poorly with noisy data and are unsuitable for large datasets since training time may increase significantly [303]. Moreover, finding a proper kernel function was difficult. |
| DeepPPI [2017]           | The DeepPPI method used a deep neural network architecture network for each protein to extract high-level discriminative features from common protein descriptors. The interaction between two proteins was determined using the one-hot encoding label. This method comprises two different architectures: DeepPPI-Sep, which uses two separate networks as input for each protein, and DeepPPI-Con, which directly links two proteins in a single network. | This method can capture informative features of protein pairs by a layer-wise abstraction. In addition, DeepPPI can automatically learn an internal distributed feature representation from the data. | The accuracy of DeepPPI for All Human/Yeast dataset is relatively low, and the accuracy of methods proposed in [304] exceeds that of DeepPPI. |
| Stacked Autoencoder (SAE) [2017] | This method used a stacked autoencoder to predict PPI. The feature extraction from protein sequences was performed using autocovariance (AC) and the conjoint triad (CT). | SAE can learn hidden interaction features of protein sequences. | They used a synthetic negative interaction dataset, and the accuracy of this model for negative interactions is relatively low. |
| DPPi [2018]              | This method performed sequence-based PPI prediction using a deep, Siamese-like convolutional neural network combined with random projection and data augmentation. This method captured the composition information, sequential order of amino acids, and co-occurrence of interacting sequence motifs in a protein pair. Each protein was characterised as a probabilistic sequence profile generated by PAST-BLAST. The patterns in each sequence were identified using the convolutional module, comprising multiple layers. The representations learned by the convolutional module were projected to two different spaces using the random projection module, allowing DPPi to explore the combination of protein motifs. | DPPi addresses interactions for both homodimeric and heterodimeric proteins. Moreover, this method could model binding affinities. | This method yields lower PPI prediction accuracy on the S.cerevisiae core dataset from PIPR based on 5-fold cross-validation compared to PIPR [299] and DeepTrio [302]. |
### Table 7 (continued)

| Framework | Description | Advantage | Disadvantage |
|-----------|-------------|-----------|--------------|
| PPIR [2019][299] | This study proposed an end-to-end framework for PPI prediction based on amino acid sequences using a deep residual recurrent convolutional neural network in the Siamese architecture. This method leveraged an automatic multi-granular feature selection to capture local significant and sequential features from protein sequences. In this method, the PPI network was regarded as an undirected graph. This framework comprised three parts. First, coding the raw protein sequences. Second, the S-VGAE model extracted vector embedding for each protein with sequence information and graph structure. Finally, a simple three-layer softmax classifier. This model was inspired by the variational graph autoencoder (VGAE) [306] that uses latent variables to learn interpretable representations for undirected graphs. | The Siamese-based learning architecture captured the mutual influence of protein pairs and allowed for generalising to address different PPI prediction tasks without needing predefined features. | RCNN was built using bidirectional gated recurrent units (bidirectional-GRU). However, GRUs suffer from slow convergence and low learning efficiency [305]. |
| S-VGAE [2020][300] | This model proposed a signed variational graph autoencoder (SVGAEE) that combined sequence information and graph structure. In this method, the PPI network was regarded as an undirected graph. This framework comprised three parts. First, coding the raw protein sequences. Second, the S-VGAE model extracted vector embedding for each protein with sequence information and graph structure. Finally, a simple three-layer softmax classifier. This model was inspired by the variational graph autoencoder (VGAE) [306] that uses latent variables to learn interpretable representations for undirected graphs. | They have observed that SVM method outperforms K-Nearest Neighbour (KNN), ANN, RFM, Naive Bayes, Logistic Regression, s for the H. pylori protein pairs. | Finding the proper kernel and hyperparameters was challenging, and training time for SVM classifiers increases with dataset size [307]. |
| ACT-SVM [2020][289] | This method performed feature extraction on the protein sequence to obtain a vector, composition, and transition descriptor and integrated them into a vector. Then, the feature vector was fed into the SVM classifier. The performance of this method was evaluated using 5-fold, 8-fold, and 10-fold cross-validation on H. pylori and human datasets. | | |
| D-SCRIPT [2021][165] | Deep sequence contact residue interaction prediction transfer (D-script) is an interpretable deep learning method generating structurally informative features given protein sequences using a pre-trained language model from [290]. This method used projection modules to reduce the dimension of features, including the residue-contact map of the protein. Finally, the interaction probability was predicted based on the contact maps. | D-SCRIPT generalised to new species considering the sparsity of training data for most model organisms (i.e., it was relatively accurate for cross-species PPI prediction). | Despite its performance for cross-species PPI prediction, D-SCRIPT underperformed on within-species evaluations. The training dataset only included proteins with 50–800 amino acids, limiting the applicability of this method. |
| SPNet [2021][9] | The Siamese pyramid network (SPNet) architecture used self-binding and folding amino acid sequences to predict the binding probability for two proteins based solely on their amino acid sequences. Subsequent screening through potential candidates was performed based on binding probabilities. | This architecture consisted of a multilevel pyramid feature structure encompassing various PPI mechanisms to reduce gradient explosion and disappearance, a multilevel Siamese neural network with an attention mechanism, and a multilevel, trainable binding probability prediction network. SPNet extracted the sequence and position of the biological information in the protein sequence. | |
| BiLSTM-RF [2021][291] | The BiLSTM-RF model extracted features of protein pairs in the human database. BiLSTM comprises forward and backwards LSTMs and is capable of bidirectional encoding (i.e., encoding front-to-back and back-to-front information). A random forest classifier (RF) was built with 100 trees and used a voting strategy to integrate these results to predict the interaction. | The protein node contained protein attribute and network structure information by integrating local and global features. | LSTM models are computationally demanding and slow. Moreover, a large number of trees in the random forest leads to a longer training time. |
| Heterogeneous Network [2021][296] | PPI prediction was performed using a computational sequence and network representation learning-based model. Local features were extracted from the protein sequence using the k-mer method (k = 3), while global features were extracted from the heterogeneous network. The latter captured network structure and obtained potential linked information. This method integrated local features with global features to represent protein nodes. | | Model accuracy is relatively low compared to other deep learning methods such as DPPI [298]. |

(continued on next page)
Table 7 (continued)

| Framework     | Description                                                                 | Advantage                                                                 | Disadvantage                                                                 |
|---------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| OR-RCNN [2021]| This method was called ordinal regression and recurrent convolutional neural network (OR-RCNN), which predicted PPIs based on their confidence score. The architecture comprised two recurrent convolutional neural networks (RCNNs) encoders, which shared the same parameters, to extract robust local features and sequential information from protein pairs. Then, one novel embedding vector was obtained by element-wise multiplication of the two embedding vectors from RCNNs. The second part of the architecture performed an ordinal regression model via multiple sub-classifiers that use the ordinal information behind the confidence score. Finally, the confidence score determined the existence of PPI with a threshold. DeepTrio is available both online and offline. | This method offered better accuracy compared to some existing models, such as autocovariance [288] and composition transition distribution (CTD) descriptor [308] for feature description, and random forest (RF) [309], extreme gradient boosting (XGBoost) [310], and support vector machine (SVM) [311] for the prediction. DeepTrio yields lower PPI prediction accuracy on the S. cerevisiae core dataset from PIPR based on 5-fold cross-validation compared to PIPR [299]. DeepTrio achieves lower PPI prediction accuracy on the S. cerevisiae core dataset from DeepFE-PPI based on 5-fold cross-validation compared to DeepFE-PPI [312]. | The RCNN was built using bidirectional gated recurrent units (bidirectional-GRU). However, GRUs suffer from slow convergence and low learning efficiency [305]. |
| DeepTrio [2022] | The DeepTrio method used a deep-learning framework based on a mask multiscale CNN architecture that performed binary PPI prediction by capturing multiscale contextual information of protein sequences using multiple parallel filters. This method used a single-protein class, allowing it to distinguish relative and intrinsic properties. This method was also made available as an online tool to address cross-platform usage and dependency-related issues. |                                                                                                                                                 |                                                                                   |

SCRIPT (Deep Sequence Contact Residue Interaction Prediction Transfer), which models protein structure using a pre-trained language model from [290]. A novel deep learning approach called Siamese Pyramid Network (SPNet) architecture is proposed in [9], which predicts the binding probability of two proteins based on their amino acid sequences. This method is employed to discover the proteins that potentially bind with the 2019-nCoV spike, in order to find future vaccines.

Learning protein pair representation is tackled by deep learning methods such as BiRNN-RF, which uses LSTM to extract the features of protein pair sequences and a random forest classifier [291], and DeepPPI [292] that uses a separate network for each protein and learns high-level features from raw protein features. The EnsDNN approach extracts the interaction information of proteins from amino acid sequences, using AC descriptor [293], local descriptor (LD) [294,295] and multi-scale continuous and discontinuous local descriptor (MCD) [15,237]. A heterogeneous network for PPI prediction is presented in [296] which uses the concatenation of local and global features to present protein node. The local features are extracted from protein sequence by the k-mer method (k = 3)1, while the global features are extracted from the heterogeneous network, and heterogeneous networks by LINE (Large-scale Information Network Embedding), respectively, and random forests to classify and predict potential protein pairs.

DPPI [298] performs sequence-based PPI prediction using a deep, Siamese-like convolutional neural network combined with random projection and data augmentation. This method captures the composition information, sequential order of amino acids, and co-occurrence of interacting sequence motifs in a protein pair. The PIPR method [299] predicts PPI by integrating a deep residual recurrent convolutional neural network (residual-RCNN) in the Siamese architecture, leveraging both robust local features and contextualised information on the protein sequence. The signed variational graph auto-encoder (S-VGAE) graph-based method [300] considers the PPI network as an undirected graph, combining sequence information and graph structure to predict PPI. A deep learning method called OR-RCNN is developed to predict PPI [301], which is composed of two recurrent convolutional neural networks (RCNNs) to extract local features and sequential information from the protein pairs and ordinal regression to construct multiple sub-classifiers. The sequence-based PPI prediction approach DeepTrio [302] uses mask parallel convolutional neural networks.

Table 7 represents further analysis of the sequence-based methods. Additionally, the datasets of each of these methods are reported in Table 8.

Some of the methods addressing protein design problems, including protein function, sequence, and structure, are discussed in the following section.

7. Protein Design

Most protein design problems require profound knowledge and subjective expertise to analyse obstacles and obtain optimal design strategy. However, with the emergence of deep neural networks, computational capacity and the available historical data, new computational methods have shown advantageous in many cases, such as RNNs’ successful application in generating SMILES (Simplified Molecular Input Line Entry System) sequences for de novo drug discovery [328,329] and optimising the RNN output to obtain specific properties through transfer learning and fine-tuning on desired sequences [330]. Recently, numerous studies have been conducted on predicting protein properties and generating new molecules and DNA sequences. These include, for example, graph neural networks for molecule representation [331–334], prediction of amino acid sequence for a particular structure using deep neural networks [104], using GAN to generate DNA sequence [335], and structure prediction methods using neural networks [180,336,337].

7.1. Protein Function

In several instances, functional folded proteins have been acquired from random-sequence libraries; however, this process

---

1 k-mers are the substrings with length k within a biological sequence [297]
Datasets of sequence-based methods.

| Framework                  | Data Processing                                                                                                                                                                                                 |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SVM-conjoint triad [2007] | A dataset comprising 16,443 nonredundant entries of experimentally verified PPI was extracted from the Human Protein Reference Database (HPRD; version 2005–0913; www.hprd.org). These interactions are primarily based on individual in vivo (e.g., coimmunoprecipitation) or in vitro (e.g., GST pull-down) experiments [286]. The negative dataset was created by excluding pairs that appeared in the positive dataset. For example, if AB and IJ are interacting pairs, AI, AJ, BI, and BJ may be noninteracting pairs. Additional conditions were applied, including an equal number of negative and positive pairs (16,443 in this study) and harmonious contributions of proteins forming the negative set. Therefore, the training set is equally distributed, comprising 32,486 protein pairs. The test set contained another 400 protein pairs. Both positive and negative pairs were randomly selected. The SVM data was extracted from the S. cerevisiae core subset of the Database of Interacting Proteins (DIP; version 20070219) [112], containing 5,966 interaction pairs. The expression profile reliability (EPR) and paralogous verification method (PVM) were used to test the reliability of this core subset [313]. By removing proteins with fewer than 50 amino acids, 5,943 protein pairs formed the final positive data set. The CD-HIT program was used to obtain a nonredundant subset with a sequence identity of 40%. The negative dataset was created using the Psb method, assuming that proteins located in different subcellular localisations do not interact. The subcellular location information was extracted from Swiss-Prot (http://www.expasy.org/sprot/). This method excluded proteins without subcellular localisation information and those marked as ‘putative’ or ‘hypothetical’, while proteins localised to the cytoplasm, nucleus, endoplasmic reticulum, Golgi apparatus, lysosome, and mitochondrion remained. The noninteracting pairs were generated by pairing proteins from one subset with those from the other. This strategy must satisfy the following conditions: (1) The DIP yeast interacting pairs do not include any noninteracting pairs, (2) there is an equal number of negative and positive pairs, and (3) the negative set should have a harmonious contribution. The non-redundant and negative pairs were obtained according to Guo et al. [288]. Therefore, the PPI dataset included 11,188 interacting and noninteracting pairs. The H. pylorn dataset contained 2,916 protein pairs (1,458 interacting and 1,458 noninteracting) following [314]. DeepPPI [2017] The dataset evaluating the DeepPPI comprised 11,188 negative and positive protein pairs from S. cerevisiae obtained from the Database of Interacting Proteins (DIP; version 20160731), 1,458 interacting and 1,458 noninteracting pairs from H.pylori, 3,899 interacting and 4,262 noninteracting pairs from humans, 4,013 interacting pairs from C.elegans, 6,954 interacting pairs from E. coli, 1,412 interacting pairs from H. sapiens, 313 interacting pairs from M. musculus, and one additional H. pylorn data set of 1,420 interacting pairs used in [315]. The negative dataset was created by pairing proteins from one subcellular location information extracted from Swiss-Prot (http://www.expasy.org/sprot/) with proteins from other locations. The Pan’s PPI dataset was acquired from [264], comprising 36,630 positive PPIs from the human protein reference database (HPRD, version 2007). Negative PPIs were generated by pairing proteins discovered in different subcellular locations from the Swiss-Prot database (version 57.3). After removing proteins with fewer than 50 amino acid residues, 2,184 unique proteins from six subcellular locations (cytoplasm, nucleus, endoplasmic reticulum, Golgi apparatus, lysosome, and mitochondrion) remained. The addition of negative pairs from the Negatome dataset [108] provided 36,480 total negative pairs. Protein pairs with nonstandard amino acids such as U and X were removed, resulting in a benchmark dataset of 36,545 positive and 36,323 negative pairs. The pre-training set contained 33,052 positive and 32,816 negative pairs, while 7,000 randomly selected pairs (3,493 positive and 3,507 negative) formed the test set. Stacked Autoencoder (SAE) [2017] Pan’s PPI dataset was acquired from [264], comprising 36,630 positive PPIs from the human protein reference database (HPRD, version 2007). Negative PPIs were generated by pairing proteins discovered in different subcellular locations from the Swiss-Prot database (version 57.3). After removing proteins with fewer than 50 amino acid residues, 2,184 unique proteins from six subcellular locations (cytoplasm, nucleus, endoplasmic reticulum, Golgi apparatus, lysosome, and mitochondrion) remained. The addition of negative pairs from the Negatome dataset [108] provided 36,480 total negative pairs. Protein pairs with nonstandard amino acids such as U and X were removed, resulting in a benchmark dataset of 36,545 positive and 36,323 negative pairs. The pre-training set contained 33,052 positive and 32,816 negative pairs, while 7,000 randomly selected pairs (3,493 positive and 3,507 negative) formed the test set. Pre-training and testing used 10-fold cross-validation. The external test sets used in this study included the 2010 version of the HPRD dataset, the 2010 HPRD NR dataset, the DIP dataset, and the HIPPIE dataset. DPPI [2018] This study used human and yeast datasets from [316]. The human PPI dataset was created by taking the 10% top-scoring interactions from the Hippie database v1.2 [317]. The yeast PPI dataset was retrieved from DPPI database [318]. The noninteracting pairs were generated by randomly sampling from all proteins, where a 10:1 negative-to-positive ratio was considered [316]. Additionally, data redundancy regarding sequence similarity in PPI (>40%) was removed following the strategy of [316]. Finally, a 10-fold cross-validation was performed. PIPR [2019] Guo’s dataset [288] comprised 2,497 proteins forming 11,188 PPI pairs, half representing positive pairs and half representing negative pairs. Interaction pairs for H. sapiens were obtained from the STRING database (version 10.5) [319]. Three thousand randomly selected pairs and 8,000 proteins that shared < 40% sequence identity formed two subsets. Finally, protein binding affinity data were obtained from the SKEMPI database [320], comprising 3,047 binding affinity changes after mutation of protein subunits within a protein complex for use in the affinity estimation task. ACT-SVM [2020] Following [321] a nonredundant dataset including H. pylori and human PPI was created. The H. pylori dataset comprised 1,458 interacting and 1,457 noninteracting protein pairs, while the human dataset comprised 3,899 interacting and 4,262 noninteracting protein pairs. S-VGAE [2020] This study used data from the human protein reference database (HPRD) and the Database of Interacting Protein (DIP) for humans, Drosophila, E.coli, and C.elegans.

(continued on next page)
been proposed in [341] to tackle the protein function prediction due to their capability to exploit these data unsupervised. Sequence variation. However, these data are not labelled, which is often laborious and limited in the types of protein they can model [338–340,92].

Machine learning algorithms offer an alternative and possibly complementary approach capable of using the information available in protein sequence and structure databases. The information about the structural and biophysical constraints on the amino acid sequence within functional proteins can be found in natural sequence variation. However, these data are not labelled, which presents a challenge for straightforward supervised learning techniques. This is where generative modelling methods show promise due to their capability to exploit these data unsupervised.

A GAN-based data augmentation approach, FFPred-GAN, has been proposed in [341] to tackle the protein function prediction problem by learning the distribution of protein amino acid sequence-based biophysical features and producing high-quality artificial protein feature samples. In the presence of auxiliary information, generative models can conduct the generative process by modelling the data distribution conditioned on the auxiliary variables. In particular, designing a protein may entail preserving a special function while modifying a property such as stability or solubility. An example of these models is conditional GAN [160]. Such a generative framework is proposed in [177], which learns a conditional generative model for protein sequences by considering a certain target structure represented by a graph over the R-group of amino acids. A VAE-based method is developed in [342] to generate novel variants of bacterial luciferase, an enzyme that emits light through the oxidation of flavin mononucleotide (FMN/H2). A combination of reinforcement learning and RNNs is proposed in [343] to generate optimal molecules for biological activity.

7.2. Structure Design

Generative models for protein structures and modelling have been studied in [177,344], among which [104,345,346] have employed neural network-based models for sequences given their 3-D structure, modelling the amino acids independently from each other. Deep generative models have enabled new and viable protein structures [347]. Predicting the missing segments of corrupted protein structures can also be achieved using GAN, as represented by [173], in which the training data is restricted to structural information about the distances between adjacent α-carbons on the protein backbone.

7.3. Sequence Design

In addition to developing structure-based models, deep generative models have gained considerable attention for analysing protein sequences in individual protein families [348–351]. Even
though these approaches have proven effective, they require that a large number of sequences from a particular family are already available. This assumption cannot be met when designing novel proteins that diverge significantly from natural sequences, owing to an unbalanced dataset, non-interacting proteins [177]. Protein-GAN [352] is a GAN-based method with a customized temporal convolutional network [353] and self-attention mechanism [354] that aims to learn vital long-range inter-residue interactions and sequence motifs, as well as focusing on functional areas [355].

A conditional variational autoencoder (CVAE) model was developed in [356] to design protein sequences conditioned on a 1-D, context-free, grammar-based specification for folding topology. In [357,358], the conditional distributions of single amino acids are modelled, considering the encompassing structure and sequence context of the given protein, using convolutional neural networks. The generative model proposed in [177] is a graph-based model that captures the joint distribution of the full protein sequence, established on long-range interactions resulting from the protein 3-D structure.

Building on several recent successful studies using deep learning methods in modelling protein sequences such as contact prediction [359], prediction of secondary structure [360,361], and prediction of the fitness effects of mutations [348], generative modelling methods have begun to show potential for designing new sequences [177,349,356,363,350,364,365]. An LSTM is used in [366] as a generative approach in terms of amino acid sequences and peptide de novo design.

8. Conclusion

In the last decade, advancements in deep learning algorithms and GPUs as accelerators for high-performance computing have facilitated resolving intricate problems [367] concerning protein–protein and protein–ligand interaction, and drug discovery. This review offers an outline of protein structures and how they interact with other proteins, towards understanding their wide range of functionalities. Additionally, we outlined several deep learning methods and their applications to predicting protein–protein interaction, new drug delivery methods, and the improvement of existing solutions.

The available datasets for deep learning methods can be divided into structure-based and sequence-based. However, there is more sequential information available for proteins than there is 3-D structural information, thus driving progress in the development of sequence-based methods [368]. In fact, all the vital information required to identify PPIs is encoded in the proteins’ amino acid sequence [369]. Several studies have been conducted by combining structural and sequential information.

Nevertheless, the viability of these techniques is yet to be verified experimentally. Continuing work is needed, such as analysing the strengths and limitations of different methods and the possibilities for incorporation into existing engineering operations. First and foremost, representation of the proteins to the network is a matter of importance [342]. This issue has been tackled using graph-based representations to model protein sequences and their 3-D structures [177,365]. Additionally, we may bolster our limited knowledge of protein folding mechanisms using deep reinforcement learning methods, aiming to find possible trajectories from extended protein chains to well-folded protein structures [355].

Following the Covid-19 pandemic and the dire need for rapid and reliable methods to create vaccines, one may see the potential of deep learning methods for solving such problems [370]. Additionally, a range of neurodegenerative diseases, infectious diseases, and cancers are closely related to abnormal protein–protein interactions [371–373]. Therefore, identifying protein–protein interactions using deep learning methods helps pave the way towards developing new drugs and targeted therapeutic approaches [8,14,374].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

[1] M.A. Clark, J. Choi, Biology (2018).
[2] Zhang F, Shi W, Zhang J, Zeng M, Li M, Kurgan L. Probselect: accurate prediction of protein-binding residues from proteins sequences via dynamic predictor selection. Bioinformatics 2020;36:i735–44.
[3] Chatr-aryamontri A, Ceol A, Licata L, Cesareni G. Protein interactions: integration leads to belief. Trends in Biochemical Sciences 2008;33(6):241–241.
[4] De Las Rivas J, Fontanillo C. Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. PLoS Computational Biology 2010;6(6):e1000807.
[5] Mackay JP, Sunde M, Lowry JA, Crossley M, Matthews JM. Protein interactions: the strengths and limitations of different methods and the possibilities using deep learning methods helps pave the way towards developing new drugs and targeted therapeutic approaches [8,14,374].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

[1] M.A. Clark, J. Choi, Biology (2018).
[2] Zhang F, Shi W, Zhang J, Zeng M, Li M, Kurgan L. Probselect: accurate prediction of protein-binding residues from proteins sequences via dynamic predictor selection. Bioinformatics 2020;36:i735–44.
[3] Chatr-aryamontri A, Ceol A, Licata L, Cesareni G. Protein interactions: integration leads to belief. Trends in Biochemical Sciences 2008;33(6):241–241.
[4] De Las Rivas J, Fontanillo C. Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. PLoS Computational Biology 2010;6(6):e1000807.
[5] Mackay JP, Sunde M, Lowry JA, Crossley M, Matthews JM. Protein interactions: the strengths and limitations of different methods and the possibilities using deep learning methods helps pave the way towards developing new drugs and targeted therapeutic approaches [8,14,374].
Gdr HB, Sharon N, Australia 

EW. Nomenclature and symbolism for amino acids. In: Maloy S, Hughes K, editors. Brenner's Encyclopedia of Genetics (Second Edition). Edition: Academic Press; San Diego; 2013. p. 1088–10.

Yang Y, Han Y, Zhang H, Li W, Dai Y. Prediction of protein–protein interactions with local weight-sharing mechanism in deep learning. BioMed Research International 2020.

A. Droit, G.G. Poirier, J.M. Hunter, Experimental and bioinformatic approaches to long-range interactions in complex bio-molecular reactions, including protein–protein interactions. ChemMedChem 2016;11(8):738.

Go N, Taketomi H, Ito T, Tashiro K, Muta S, Ozawa R, Chiba T, Nishizawa M, Yamamoto K, Kuhara S, Sakaki Y. Toward a protein–protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. Proceedings of the National Academy of Sciences 2000;97(3):1143–7.

Brizuela L, Braun P, LaBaer J. Flexgene repository: from sequenced genomes to function. Bioinformatics 2001;17:2492–501.

Zhang C, Zheng W, Cheng M, Ommen GS, Freddolino PL, Zhang Y. Functions of essential genes and a scale-free protein interaction network revealed by structure-based function and interaction prediction for a minimal genome. PLoS computational biology research 2021;20(2):1178–89.

Chirgadze Y, Boshkovska E, Kargatov A, Chirgadze N. Functional identification of hypothetical protein-structures with unknown function. Journal of Bioinformatics and Computational Biology 2022:1–5.

Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M. Bayesian networks approach for predicting protein–protein interactions from genomic data. Science 2003;302:449–53.

Schwikowski B, Utzet P, Fields S. A network of protein–protein interactions in yeast. Nature biotechnology 2000;18:1257–61.

Melkikh AV, Meijer DK. On a generalized levinthal’s paradox: The role of long-range and short range interactions in complex bio-molecular reactions, including protein and dna folding. Progress in Biophysics and Molecular Biology 2018;132:57–79.

Jancarik J, Sanderson CM. Prediction of protein–protein interactions from amino acid sequences based on continuous and discrete wavelet transform features. Molecules 2018;23(4):823.

Hatfield DL, Gladyshev VN. How selenium has altered our understanding of human health. Croatian medical journal 2012;53(6):535–50.

Bussow K, Nordhoff E, Lübbert C, Lehrach H, Walter G. A human cdna library enriched for hypothetical protein-structures with unknown function. Journal of Molecular Endocrinology 34 (2) (01 Apr. 2005) 263–280.

Klahre G, Xue Z, Buehler K, Dehoux M, Wang W, Metzger J, Hahn H. An integrated wavelet transform features. Molecules 2018;23(4):823.

Desmond DJ, Echeverria CM, Frankel WA, Nelson SE, Haines JL, et al. Prediction of protein–protein interactions. ChemMedChem 2016;11(8):738.

Waters AL, Li L, Huang T, Zhang Y, Hu Y. A network of protein–protein interactions in yeast. Nature biotechnology 2000;18:1257–61.

Jiang L, Althoff EA, Zanghellini A, Dym O, et al. Kemp elimination catalysts based on deep learning, and the fractional fokkerplanck and khlerdirac equations., IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019;41(2):363–77.

Kalodimos CG, Johnsson K, Stoddard BL, et al. Computational design of ligand-based enzyme design. Nature 2008;453:190–5.
Chen J, Zheng S, Zhao H, Yang Y. Structure-aware protein solubility prediction.

Wang C, Bradley P, Baker D. Protein–protein docking with backbone.

Tatusov RL, Koonin EV, Lipman DJ. A genomic perspective on protein families.

Kundrotas PJ, Zhu Z, Janin J, Vakser IA. Templates are available to model.

Rao J, Zhou X, Lu Y, Zhao H, Yang Y. Imputing single-cell rna-seq data by.

Tuncbag N, Gursoy A, Nussinov R, Keskin O. Predicting protein-protein.

Porollo A, Meller J. Prediction-based fingerprints of protein–protein.

Vakser IA. Protein-protein docking: From interaction to interactome.

Yan Y, Wen Z, Wang X, Huang S-Y. Addressing recent docking challenges: A.

Chaudhury S, Lyskov S, Gray JJ. Pyrosetta: a script-based interface for.

Balakrishnan S, Kamisetty H, Carbonell JG, Lee S-I, Langmead CJ. Learning.

You Z-H, Chan KC, Hu P. Predicting protein-protein interactions from primary.

Liang S, Zhang C, Liu S, Zhou Y. Addressing recent docking challenges: A.

Mashiach E, Nussinov R, Wolfson HJ. Fiberdock: flexible induced-fit backbone.

Huang Y-A, You Z-H, Gao X, Wong L, Wang L. Using weighted sparse.

Marks DS, Colwell LJ, Sheridan R, Hopf TA, Pagnani A, Zecchina R, Sander C.

Tuncbag N, Gursoy A, Keskin O. Prediction of protein–protein interactions:.

Ohue M, Matsuzaki Y, Uchikoga N, Ishida T, Akiyama Y. Megadock: an all-to-

Qin S, Zhou H-X. meta-ppisp: a meta web server for protein-protein.

Fukuhara N, Kawabata T. Homcos: a server to predict interacting protein.

Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, Zecchina R,

Szilagyi A, Zhang Y. Template-based structure modeling of protein–protein.

Huang Y-A, You Z-H, Gao X, Wong L, Wang L. Using weighted sparse.

Chaudhury S, Lyskov S, Gray JJ. Pyrosetta: a script-based interface for.

Balakrishnan S, Kamisetty H, Carbonell JG, Lee S-I, Langmead CJ. Learning.

You Z-H, Yu J-Z, Zhu L, Li S, Wen Z-K. A mapbased reduced parallel sv.

Northey TC, Barelis A, Martin AC. Intspec: a structure-based predictor of.

Topham J, Byrd, B. Shariat, A. Ben-Hur. Protein interface prediction using.

M. Baranwal, A. Magner, J. Saldinger, E.S. Turali-Emre, S. Kozarekar, P. Elvati, J.

A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using.

Liu L, Zhu X, Ma Y, Piao H, Yang Y, Hao X, Fu Y, Wang L, Peng J. Combining.

Y. Song, S. Zheng, Z. Niu, Z.-H. Fu, Y. L, Y. Yang, Communicative representation.

Cloning and sequencing of a new graph distance for deformable 3d.

Chen J, Zheng S, Zhao H, Yang Y. Structure-aware protein solubility prediction.

Rao J, Zhou X, Yu L, Zhao H, Yang Y. Imputing single-cell rna-seq data by.

Kundrotas PJ, Zhu Z, Janin J, Vakser IA. Templates are available to model.

Topham J, Byrd, B. Shariat, A. Ben-Hur. Protein interface prediction using.

M. Baranwal, A. Magner, J. Saldinger, E.S. Turali-Emre, S. Kozarekar, P. Elvati, J.

A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using.

Liu L, Zhu X, Ma Y, Piao H, Yang Y, Hao X, Fu Y, Wang L, Peng J. Combining.

Y. Song, S. Zheng, Z. Niu, Z.-H. Fu, Y. L, Y. Yang, Communicative representation.

Cloning and sequencing of a new graph distance for deformable 3d.

Chen J, Zheng S, Zhao H, Yang Y. Structure-aware protein solubility prediction.
quality-controlled protein–protein association networks, made broadly accessible, Nucleic acids research (2016) gkw9537.

[320] Moal IH, Hernández-Recio J. Skempa: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. Bioinformatics 2012;28(20):2600–7.

[321] Kong M, Zhang Y, Xu D, Chen W, Dehmer M. Ftcp-wsrc: protein–protein interactions prediction via weighted sparse representation based classification. Frontiers in genetics 2020;11:186.

[322] Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 2006;22(13):1658–9.

[323] Fu J, Liu B, Zhu Z, Li W, Gao W, Cai G. Cd-hit-accelerated for clustering the next-generation sequencing data. Bioinformatics 2012;28(23):3150–2.

[324] F. Richoux, C. Servantie, C. Bories, S. Téletchéa, Comparing two deep learning sequence-based models for protein-protein interaction prediction, arXiv preprint arXiv:1901.06268.

[325] Li W, Jaroszewski L, Godzik A. Clustering of highly homologous sequences to reduce the size of large protein databases. Bioinformatics 2001;17(3):282–3.

[326] Oughtred R, Stark C, Breitbart W, Rust J, Boucher L, Chang C, Kolas N, O’Donnell L, Leung G, McAdam R, et al. The biogrid interaction database: 2019 update. Nucleic acids research 2019;47(1):D529–41.

[327] U. Consortium. Uniprot: a worldwide hub of protein knowledge. Nucleic acids research 2019;47(1):D705–15.

[328] Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. Molecular science 2018;4(1):1–12.

[329] Semmler MJ, Paige B, Hernandez-Lobato JM. Grammar variational autoencoder. International Conference on Machine Learning, PMLR 2017:1945–54.

[330] Kusner MJ, Paige B, Hernandez-Lobato JM. Grammar variational autoencoder. A deep learning network approach to ab initio protein secondary structure prediction, in: International conference on machine learning, PMLR, 2019, pp. 7354–7363.

[331] Zhang J, Goodfellow D, Metaxas A, Odena, Self-attention generative adversarial networks, in: International conference on machine learning, PMLR, 2019, pp. 7354–7363.

[332] Ding, K. Nakai, H. Gong, Protein design via deep learning, Briefings in Bioinformatics.

[333] Greener FC, Moffat J, Jones DT. Design of metalloproteins and novel protein folds using variational autoencoders. Scientific reports 2018;8(1):1–12.

[334] Boomsma, J. Frenkels, Spherical convolutions and their application in molecular modelling, in: NIPS, vol. 2, 2017, p. 6.

[335] M. Wu, M. Wei, M. Geiger, M. Wolfs, B. Wong, B. Mooney, T. Cohen, 3d steeple cxyz: Learning rotationally equivariant features in volumetric data, arXiv preprint arXiv:1807.02547.

[336] Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate de novo prediction of protein contact map by ultra-deep learning model. PloS Computational Biology 2017;13(1):e1005324.

[337] Spencer M, Eickholt J, Cheng J. A deep learning network approach to ab initio protein secondary structure prediction, PLoS ONE 2015;10(1):e116164.

[338] Wang S, Feng J, Ma J, Xu J. Protein secondary structure prediction using deep convolutional neural networks. Scientific reports 2016;6(1):1–11.

[339] Das, K. Wadhawan, O. Chang, T. Secco, C. D. Santos, M. Riener, V. Chenthamarakshan, I. Padhu, A. Mojsilovic, Pepave: Semi-supervised targetted design of antimicrobial peptide sequences, arXiv preprint arXiv:1810.07743.

[340] De Repecka, V. Janinskis, L. Karpius, E. Rembeza, J. Zrimec, S. Poviloniene, L. Rokaitis, A. Laurynenas, W. Abuaiaj, O. Savolainen, et al. Functional protein sequence space using generative adversarial networks, bioRxiv (2019) 787917.

[341] A. Madami, B. McCane, N. Naik, N.S. Keskar, N. Anand, R.R. Eugchi, P.-S. Huang, R. Socher, Progen: Language modeling for protein prediction, arXiv preprint arXiv:2004.03497.

[342] Stricko A. Beverda C. Corbi-Verge C. Perez-Riba A. Kim PM. Fast and flexible protein function prediction. PLoS ONE 2013;8(6):e62996.

[343] Boomsma, J. Frenkels, Spherical convolutions and their application in molecular modelling, in: NIPS, vol. 2, 2017, p. 6.

[344] M. Wu, M. Wei, M. Geiger, M. Wolfs, B. Wong, B. Mooney, T. Cohen, 3d steeple cxyz: Learning rotationally equivariant features in volumetric data, arXiv preprint arXiv:1807.02547.

[345] Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate de novo prediction of protein contact map by ultra-deep learning model. PloS Computational Biology 2017;13(1):e1005324.

[346] Spencer M, Eickholt J, Cheng J. A deep learning network approach to ab initio protein secondary structure prediction, PLoS ONE 2015;10(1):e116164.

[347] Wang S, Feng J, Ma J, Xu J. Protein secondary structure prediction using deep convolutional neural networks. Scientific reports 2016;6(1):1–11.

[348] Das, K. Wadhawan, O. Chang, T. Secco, C. D. Santos, M. Riener, V. Chenthamarakshan, I. Padhu, A. Mojsilovic, Pepave: Semi-supervised targetted design of antimicrobial peptide sequences, arXiv preprint arXiv:1810.07743.

[349] De Repecka, V. Janinskis, L. Karpius, E. Rembeza, J. Zrimec, S. Poviloniene, L. Rokaitis, A. Laurynenas, W. Abuaiaj, O. Savolainen, et al. Functional protein sequence space using generative adversarial networks, bioRxiv (2019) 787917.

[350] A. Madami, B. McCane, N. Naik, N.S. Keskar, N. Anand, R.R. Eugchi, P.-S. Huang, R. Socher, Progen: Language modeling for protein prediction, arXiv preprint arXiv:2004.03497.

[351] Strokach A, Becerra D, Corbi-Verge C, Perez-Riba A, Kim PM. Fast and flexible protein function prediction. PLoS ONE 2013;8(6):e62996.

[352] Boomsma, J. Frenkels, Spherical convolutions and their application in molecular modelling, in: NIPS, vol. 2, 2017, p. 6.