Received: 2 April 2023 | Revised: 24 May 2023 | Accepted: 9 June 2023

DOI: 10.1002/pmic.202200292

Proteomics
Proteomics and Systems Biology

REVIEW

State-of-the-art computational methods to predict protein–protein interactions with high accuracy and coverage

Neal Kewalramani[1] | Andrew Emili[2] | Mark Crovella[3]

[1]Program in Bioinformatics, Boston University, Boston, Massachusetts, USA

[2]OHSU Knight Cancer Institute, Portland, Oregon, USA

[3]Department of Computer Science and Program in Bioinformatics, Boston University, Boston, Massachusetts, USA

Correspondence
Andrew Emili, Program in Bioinformatics, Boston University, Boston, MA, USA.
Email: aemili@bu.edu

Funding information
internal Boston University funds and NIH, Grant/Award Number: RO1AI146941

Abstract

Prediction of protein–protein interactions (PPIs) commonly involves a significant computational component. Rapid recent advances in the power of computational methods for protein interaction prediction motivate a review of the state-of-the-art. We review the major approaches, organized according to the primary source of data utilized: protein sequence, protein structure, and protein co-abundance. The advent of deep learning (DL) has brought with it significant advances in interaction prediction, and we show how DL is used for each source data type. We review the literature taxonomically, present example case studies in each category, and conclude with observations about the strengths and weaknesses of machine learning methods in the context of the principal sources of data for protein interaction prediction.

KEYWORDS
bioinformatics, data processing and analysis, mass spectrometry, technology

1 | INTRODUCTION

Proteins are the basic building blocks of organisms, but a protein does not function solely on its own. Rather, proteins interact physically and specifically with one another to perform particular cellular processes. These interactions occur through either transient or stable non-covalent bonds between amino acid side chains, which guide the quaternary superstructure of macromolecular complexes and enable functional properties [1].

Because of their biological complexity, identifying protein–protein interactions (PPIs) remains a major challenge for researchers.

Understanding which proteins interact with each other, either in a pairwise fashion or as components in a multi-subunit complex, is an important task because these interactions reveal basic functional mechanisms and suggest the potential druggability surfaces of molecules for pharmacological modulation.

Traditionally, as for 3D protein structure determination, PPIs have been mapped using a diversity of experimental techniques [1]. There exist many experimental methods such as yeast two-hybrid and protein microarrays to find PPIs [2, 3]. Some of these methods are high-throughput, being able to test many PPIs at the same time, but experimental elucidation of PPIs requires time and resource dedication, and these strategies are subject to diverse sources of technical errors [4]. As a result of these drawbacks, recent effort has gone into the development of computational methods to predict PPIs. In concert with advances in 3D structure prediction [5, 6], computational methods have now emerged as a viable technique to infer PPIs because of their advantage of being scalable with less resource dedication.

Computational methods, in this context, refer to the prediction methods used to convert a source of biological data to a PPI prediction. The field has mainly been using machine learning (ML) algorithms to convert the biological data to predictions, so this paper will be covering the ML methods used and what advances have been made in the PPI prediction field.

Because of the recent increase in the use of computational methods to predict PPIs, there exists a need for a review of current approaches

to the problem. This paper will review the recent progress in computational methods used to predict PPIs. We categorize the work in the PPI prediction field based on the data inputs to the prediction models. There are three main data inputs used in the PPI prediction field: protein sequences, protein structure, and co-fractionation mass spectrometry data. In the body of this paper, we review work using each of these data sources, and we present case studies exploring methods that use each of these data sources. Table 1 presents an overview of the computational methods we survey.

From Table 1, it is shown that the datasets used to train and benchmark models are generally consistent. *Saccharomyces cerevisiae*, *Mus musculus*, *Homo sapiens*, *Caenorhabditis elegans*, *Escherichia coli*, and *Helicobacter pylori* are all well-studied organisms and are common benchmark datasets for PPI prediction models. Some researchers have made their own datasets using PPIs from the organisms mentioned above to create more balanced datasets for training and testing. Therefore, the Dsets we see in the table are also prevalent in the PPI prediction field. The datasets mentioned are curated from publicly available datasets and their respective description and statistics on how the datasets were optimized to allow for good performance are available through the corresponding citation in Table 1. It is important to note that between models that datasets taken from the same organism can be different depending on considerations made by the different authors, so statistics and performance can vary depending on the considerations the authors took into account. Dsets mentioned in the table are curated datasets from specific organisms by other researchers, so Dsets are the same when mentioned in the table.

2 | METHODOLOGICAL BACKGROUND

PPI prediction models aim to use some source of protein data to predict known interactors. Before reviewing these computational models for predicting protein interactions, we provide a brief overview of concepts to provide background on the methods discussed.

2.1 | Machine learning

ML is a subfield of artificial intelligence that focuses on using data to learn associations and make predictions [68]. In this review, we focus on the methods of ML that use supervised learning for classification since these are the most relevant to PPI prediction. Supervised ML refers to the use of labeled training data, in which data items having associated features that are equipped with labels. For PPI prediction, these labels are strictly binary: whether the two input proteins are interactors. In this setting, the goal of classification is to construct a model that infers labels for test data – that is, data coming from the same distribution as the training data, but without labels. In the context of PPI prediction, the input to an ML model is two protein representations, and the label associated with the input is whether the two proteins are known interactors. Essentially, the ML model tries to extract the relevant features from each of the protein representations, and then, use these features for a PPI prediction.

In supervised ML, training data is used to adjust model parameters toward settings that tend to predict known interactors accurately. Once trained, we expect the model to predict labels in the training data accurately. Test data, that serves as unseen data, provide information on how well the ML model performs on data it has not seen to test the model's ability to generalize.

2.2 | Classical machine learning

We use the term *classical* ML to refer to algorithms developed before the advent of deep learning (DL) techniques (which are discussed further below). Classical ML methods include *decision trees*, *support vector machines* (SVMs), and *random forest classifiers* (RFCs). Decision trees construct a tree-structured partition of the data's feature space [69]. Each node in the tree represents a single logical test of a single data feature's values – such as the presence of an alpha helix in a protein and so splits the data space based on that feature. Leaves in the tree are associated with a label and correspond to predictions. In contrast, SVMs non-linearly map input feature vectors to a high-dimensional space and construct linear decision boundaries in that space to assign labels for classification [70]. Finally, RFCs use an ensemble method comprised of many decision trees constructed from random samples of the training data and uses a voting system among these decision trees to determine the predicted classification [71].

2.3 | Deep learning

DL is a subset of the ML field that refers to the use of artificial neural networks (ANNs) for tasks such as classification [72]. The basic building block of a neural network architecture is the use of artificial neurons which typically take in a weighted sum of inputs and then applies a nonlinear transformation to produce an output that is shared with other neurons downstream. Neurons are organized into layers, and each layer consists of multiple neurons that takes their input either from data or from a previous layer's output. Learning occurs by adjustment of the weight parameters in a manner that seeks to achieve good classification performance. In the context of PPI prediction, weights would be optimized to maximize the number of correct predictions of known interactors and non-interactors. A *neural architecture* refers to the particular number of layers and interconnection pattern linking those layers. A number of well-studied neural architectures are typically found; in the following subsections, we discuss architectures prominently used in protein interaction prediction.

2.3.1 | Convolutional neural networks

The convolutional neural network (CNN) is one of the most popular architectures used in DL [72]. A CNN architecture uses convolutional

TABLE 1 Overview of computational methods comprehensive list of PPI prediction pipelines.

Name of pipeline	Data inputs	ML models used	Datasets/species used
DeepFE-PPI [7]	Text embedding	Fully connected network	*S. cerevisiae* [8], Human [9], *C. elegans*, *E. coli*, *H. sapiens*, *M. musculus*, *H. pylori* [10]
DeepPPI [11]	Metric extractions (amino acid composition, dipeptide composition, etc.)	Fully connected network	*S. cerevisiae* [12], *H. pylori* [13, 10], Human [9], *C. elegans*, *E. coli*, *H. sapiens*, *M. musculus* [10]
SGPPI [14]	AF protein database structures	Graph convolutional network	Profppikernel dataset [15], Human Reference Interactome [16], Pan's dataset [17]
TAGPPI [18]	Protein sequence, AF output	Convolutional neural network, graph attention network	*S. cerevisiae* [19], *E. coli*, *C. ele-gans* [10], *D. melanogaster*, multi-species dataset [20], multi-class dataset [21]
GRAPHPPIS [22]	PDB structures	Graph convolutional network	Dset 186, Dset 72 [23], Dset 164 [24]
DPPI [25]	Protein sequences	Convolutional neural network	*H. sapiens*, *S. cerevisiae*, *S. pombe*, *M. musculus*, *D. melanogaster*, *C. elegans*, *A. thaliana*, *B. subtilis*, *B. taurus*, *E. coli*, *R. norvegicus* [26]
PIPR [20]	Protein sequence encoding, amino acid embedding	Convolutional neural network	Guo's dataset [8], Multi-species dataset (*C. elegans*, *E. coli*, *D. melanogaster*), STRING *H. sapiens* dataset [21], SKEMPI dataset [27]
EnsDNN [28]	Metric extractions (auto covariance, multi-continuous descriptor, etc.)	Fully connected network	*S. cerevisiae* [11], *C. elegans*, *E. coli*, *H. sapiens*, *M. musculus*, *H. pylori* [10]
SDNN-PPI [4]	Metric representations (amino acid composition, conjoint triad, auto covariance)	Fully connected network, attention mechanism	*S. cerevisiae* [8], Human [29], Human- *B. Anthracis*, Human-*Y. pestis* [30], *C. elegans*, *E. coli*, *H. sapiens*, *M. musculus* [31]
Jha et al [32]	PDB structures	Graph attention network	Pan's dataset [17], *S. cerevisiae*
Sun et al [33]	Metric extractions (autocovariance, conjoint triad)	Autoencoder	Pan's dataset [17], 2010 HRPD dataset, 2010 HRPD NR dataset, DIP dataset [8], HIPPIE dataset, inWeb inbiomap, 2005 Martin dataset [17], *E. coli*, *Drosophila*, *C. elegans* [8]
DeepCF-PPI [34]	Metric extractions (amino acid composition, pseduo-amino acid composition, etc.), text embedding	Fully connected network, attention mechanism	*S. cerevisiae* [35], Human [9], *C. elegan*, *E. coli*, *H. sapiens*, *H. pylori*, *M. musculus* [36], CD9, Wnt, Cancer [11, 37, 38]
ctP2ISP [39]	Protein sequence, metric extractions (biophyical information)	Convolutional neural network, transformer	Dset 186, Dset 72 [23], Dset 164 [24], Dset 448 [40], Dset 355 [41], Dset 70 [42]
GOSeqPPI [43]	Text embedding, one-hot encoding	Recurrent neural network, convolutional neural network, attention mechanism	*S. cerevisiae*, SHS27k, SHS148k
ISPRED-SEQ [44]	One-hot encoding	Convolutional neu-ral network	Dset 335 [41], Dset 448 [40], HomoTE, HeterTE [45]
SECAT [46]	CF/MS	PyProhet ML algorithm [47]	HeLa-CC SEC-SWATH-MS dataset, HEK293-EG SEC-SWATH-MS dataset
ADH-PPI [48]	Text embedding	Long short-term memory, convolutional layer, attention mechanism	*S. cerevisiae*, *H. pylori*, *Saccharomyces*, *C. elegans*, *H. sapiens*, *M. musculus*, *E. coli*
CCprofiler [49]	CF/MS Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. issn: 0885-	Peak detection algorithm	Composite SWATH/DIA dataset [50]
Topsy-Turvy [51]	Feature embedding, metric extraction	Convolutional neural network	Human, *M. musculus*, *D. melanogaster*, *C. elegans*, *S. cerevisiae*, *E. coli* [52, 21]
Ahmed et al [53]	Metric extractions(amino acid triplets, quadruplets, etc.)	Fully connected network	Human [54], *B. anthracis* [55, 56]

(Continues)

TABLE 1 (Continued)

Name of pipeline	Data inputs	ML models used	Datasets/species used
Struct2Graph [57]	PDB structures	Graph convolutional network, attention	*S. cerevisiae*, *H. sapiens*, *E. coli*, *C. elegans*, *S. aures* from IntAct [58], STRING [59]
PrInCE [60]	CF/MS	Protein clustering algorithm (ClusterONE) [61]	Four co-fractionation datasets [62, 63, 64]
EPIC [65]	CF/MS	SVM, RFC	*C. elegans*
DeepPPISP [42]	Metric extractions (position-specific scoring matrix, secondary structure), raw protein sequence, encoding	Convolutional neural network	Dset 186, Dset 72 [23], PDBset 164 [66]
PITHIA [67]	Multiple sequence alignment, metric extractions	Transformer	Dset 72, Dset 186 [23], Dset 164 [24], Dset 448 [40]

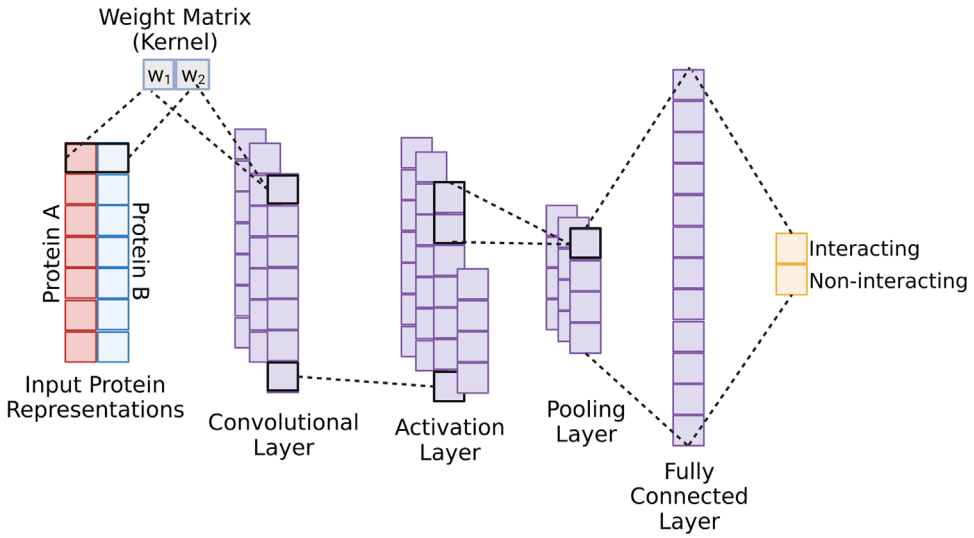

FIGURE 1 Example architecture of a convolutional neural network (CNN) for PPI prediction. Adapted from Alzubaidi et al. [72].

layers, which are layers that employ a kernel to transform output of the previous layer. The kernel is a matrix of learned weights that "moves" over the input data and performs a dot product with the corresponding region in the input. Because of this calculation, the kernel condenses the input into a smaller size. In contrast, a network having an individual weight for each element in the input – referred to as a fully connected layer – can suffer from excessive memory size and overfitting (too many parameters compared to the training data size). CNNs address this issue by only using a limited set of weight (those in the kernel); hence the weights in a CNN are referred to as shared weights. CNNs are capable of identifying long-range interactions in data; this capability is important for protein sequences as amino acid residue interactions in a protein sequence are long-range and can be captured by a CNN. One architectural challenge in using ANNs is the question of how to process inputs of variable length (such as sequence strings).

Figure 1 is a general pipeline for a CNN architecture for PPI prediction. The convolutional layer incorporates a weight matrix (kernel) to identify long-range relationships in the input data – in this case, two protein representations. An activation layer applies a nonlinear function to convert the numbers in the convolutional layer to something more interpretable for the neural network. Common examples of nonlinear functions include ReLu and sigmoid. The pooling layer condenses the activation layer – this can be done by taking an average or max across the window of values. In the fully connected layer, the vectors generated in the previous layer are stacked on top of each other to create one vector and is then used to find a prediction of whether the two proteins are interacting. It is important to note that the activation, pooling, and fully connected layers are not unique to CNN architectures but are included to help process the outputs from the CNN layer into a prediction.

2.3.2 | Recurrent neural networks

Early attempts to handle variable length inputs led to the recurrent neural network (RNN) architecture. In an RNN, feedback loops exist between layers, so that data can be input in sequence, and previously seen data can affect the classification of subsequent data [72]. RNNs

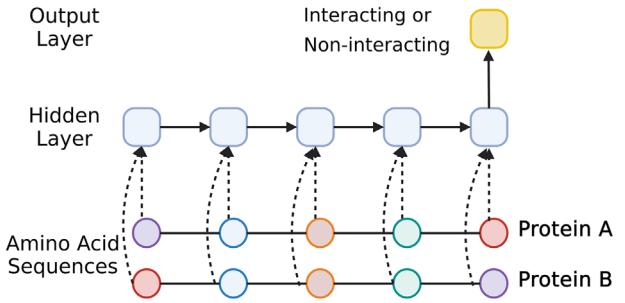

FIGURE 2 Example architecture of an unrolled recurrent neural network (RNN) for PPI prediction. Adapted from Alzubaidi et al. [72].

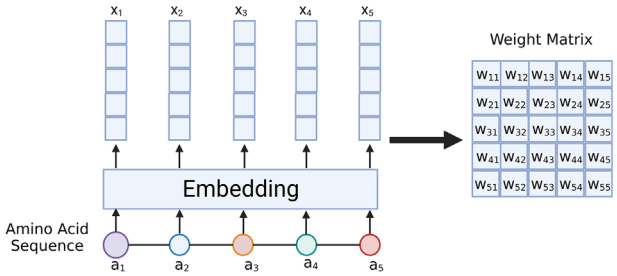

FIGURE 3 Example attention mechanism. Each element in the weight matrix is the result of the dot product of the corresponding vectors. Adapted from Vaswani et al. [73].

are capable of identifying relationships between the different inputs, making them optimal for applications in video processing and natural language processing. Because the protein sequence-to-PPI prediction problem can be seen as a derivation of problems in the natural language processing field, protein sequence models have a solid foundation of text models and research to adopt from for PPI predictions. Just like words in a sentence need context to understand their meanings in a sentence, amino acid residues need context from other amino acid residues to understand their roles in the protein sequence. Therefore, text-based architectures are a good fit and have been used for the protein sequence to PPI prediction problem.

Figure 2 is a general pipeline for an RNN architecture for PPI prediction. In this example, the RNN is displayed in its "unrolled" state. Because RNNs are cyclical, RNN diagrams are displayed in an unrolled state over the input sequence (protein amino acid sequence) to display its architecture. This example RNN is referred to as a many-to-one RNN as it takes an input sequence and generates one output (PPI prediction value).

Unfortunately, due to the presence of feedback loops, RNNs are subject to the vanishing gradient problem, meaning that the network can sometimes fail to propagate useful information from the early layers of the network during training. The long short term memory (LSTM) architecture has been developed to overcome this issue; it introduces gates within the architecture to better control the flow information from layer to layer [72].

Nonetheless, training an RNN or LSTM with variable-length protein sequence data is computationally expensive since the network requires long training time to capture long-range amino acid residue interactions in the input sequence.

2.3.3 | Transformers and attention

The *attention* mechanism is the key advance leading to the *transformer* neural architecture [73]. Similar to RNNs, transformers are an encoder-decoder architecture but use semi-supervised learning, unlike the previously discussed methods. Transformers were developed to address the problems with RNNs and LSTMs: while RNNs process the input sequentially leading to long training times, transformers process all elements in the input simultaneously, allowing for faster computations. Google first introduced the attention mechanism with the transformer architecture. Simply, the attention mechanism represents each sequence in the input as a numerical representation of the other sequences in the input. This vectorization of each sequence assigns relationships to each other sequence input, giving the neural network context for each sequence. Transformers use these contextualized vectors to decode into a new representation. Because of this, transformers are popular in language translation models as they can capture the individual meaning and context of each word in a sentence and translate it to a different language. Both the transformer architecture and the attention mechanism have applications for PPI prediction. The transformer architecture takes in sequential data, like a protein's primary sequence, and produces an output representation. However, as discussed in the coming sections, the attention mechanism has been ubiquitous in the PPI prediction field and for good reason. Future sections will discuss the application of attention in the PPI prediction field.

Figure 3 is a general example for an attention mechanism. The input amino acid sequence is converted to a vector through some embedding. The weight matrix is then constructed taking the dot product of the vectors. The weight matrix can then be multiplied by the output vectors from the embedding to create an amino acid representation that is context aware.

2.3.4 | Embeddings and encodings

Embeddings are representations of feature-rich inputs that are scaled down to reduce the dimensions from the input. This scaling down process highlights important features within the network. Embeddings come from pre-trained neural networks that generate feature representations as an output. For example, Word2Vec is a neural network architecture that represents words in a sentence as vectors in a sentence that capture their semantic and syntactic attributes [74]. It is important to note that embeddings are of a fixed length; therefore, the method that produces the embedding can take a variable length input and yet still produce a fixed-length output. Word2Vec and other relevant embedding architectures that have been adapted for protein representations will be explained further below.

In this review, we use the terms *embeddings* and *encodings* interchangeably. Embeddings and encodings refer to data transformations that either approximate or provide a direct one-to-one transformation of an input. We briefly describe some examples of encodings and embeddings that will be referenced in future sections. An example of an encoding is the one-hot encoding. The one-hot encoding converts categorical data, like amino acids in a protein sequence, into an $n \times L$ dimensional vector, where n is the number of categories and L is the length of the input. For example, a one-hot encoding of a protein's amino acid sequence would be a $20 \times L$ dimensional vector, where 20 is the number of categories (amino acids) and L is the length of the protein's amino acid sequence. This encoding has a direct map allowing conversion back to the input.

Word2Vec: Word2Vec is a neural network architecture that, as the name suggests, converts words into a vector. The network learns word associations by training on a body of text and represents these word associations in vectors. The goal of this vectorization is to be able to perform simple algebraic operations to combine the associations of different words. For example, with Word2Vec, it has been shown that $\text{vector}(King) - \text{vector}(Man) + \text{vector}(Woman) \approx \text{vector}(Queen)$ [74].

FastText: FastText is similar to Word2Vec in that its goal is to produce a vectorization of words. However, FastText's main contribution lies in its architecture: words are represented as a combination of vector representations of characters. This approach makes FastText, as suggested by its name, a faster method for training and producing these vector representations [75].

BERT: BERT is a masked-language model that aims to predict a missing word in a sentence. BERT is trained using sentences with a masked token on some words. BERT will then try to predict these masked words given context from the other words in the sentence [76].

2.3.5 | Autoencoders

Autoencoders are different from the previously mentioned architectures because they use unsupervised learning. Simply, autoencoders try to project their input into a different dimensional feature vector (encoding) and then use this feature vector to recreate the input (decoding). The goal of this architecture is to create an encoder that is capable of projecting an input to a different dimension. Most commonly, autoencoders are used to project data into a lower dimension in order to enhance signals, highlight features, and reduce noise. This attribute of autoencoders is important for mass spectrometry data, which will be discussed in a future section.

2.4 | Training and testing data partitioning

Since the goal of ML is to be able to make predictions on unseen data (i.e., generalization), one commonly splits available data into training and testing sets, with model training taking place on the training set and model testing performed using the (previously unseen) test set.

However, performing the train/test split requires care in the context of PPI predictions. One must avoid "leaking" information from the training data to the testing data. If the set of interactions are simply partitioned, typically many proteins will appear in both the training and testing set, thereby leaking information from training to testing. For example, if a highly connected protein appears in both testing and training, the ML model may learn to simply predict that the protein has further interaction partners without taking into account the properties of the partners. This can lead to unrealistically high accuracies. To address this issue, the best practice is to partition the available data based on proteins themselves and not on PPIs. This split means that proteins in the training set will not be seen in the test set, giving more confidence in the model's ability to generalize [77].

3 | PROTEIN SEQUENCES FOR PPI PREDICTIONS

Having covered the necessary ML background, we now turn to the use of protein sequences and the computational methods used to integrate these sequences into a PPI prediction pipeline. PPI prediction models that use protein sequences are plentiful mainly because of how readily available protein sequences are. Below, we discuss the current general pipeline and give examples of existing pipelines in the field.

Protein sequence refers to the primary structure of proteins: the amino acid sequence. The amino acid sequence is a 1D chain of twenty possible amino acids. This specific arrangement of amino acids is referred to as the protein's primary structure [1]. While this 1D sequence's relationship to the protein's 3D structure is not easily apparent, there exist patterns, such as certain amino acid combinations, within the protein sequence that define folding motifs (e.g., alpha-helix, beta-fold) within the protein's structure. The goal of using computational methods with protein sequences for PPI prediction is to capture these patterns from the interfaces of known interacting polypeptides to predict PPIs based on the similarity between other proteins' sequences. Here, we discuss the computational work that has been done to assess the interaction between two candidate proteins based on their sequences alone.

Much of the work that focuses on computational methods for PPI predictions uses protein sequences as its sole input. These protein sequences are far more readily available as compared to the 3D protein structural models: there are around 189 million protein sequences available on UniProt, while there are only a little more than 200,000 protein structures deposited in the Protein Data Bank (PDB) [78, 79]. The plethora of protein sequences and the ease of accessing them has contributed to an abundance of computational models that predict PPIs based solely on protein sequences alone.

When it comes to training data for PPI prediction based on reference protein sequences, researchers can access a diversity of experimentally derived PPIs based on well-studied organisms [4, 11, 20, 33, 34, 80]. They can split these into and training and test (hold out) sets to demonstrate that a particular model is able to perform well, capable of getting high accuracy and not overfit to one particular reference study.

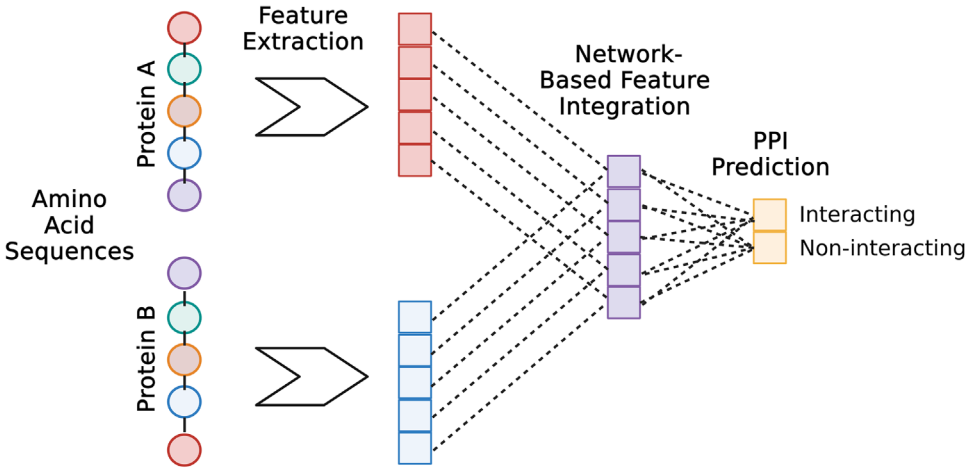

FIGURE 4 General pipeline of PPI prediction models using protein sequences. The input is the primary amino acid sequence. A feature extraction step is used to distill important features from the amino acid sequence. This step includes metric representations, embeddings, encodings, or some use of the original protein sequence. A network architecture then integrates the two separate protein feature vectors and outputs a binary prediction.

Commonly referenced datasets include *S. cerevisiae* [4, 11, 34], *E. coli* [34], and Human datasets [4, 11, 34].

Early sequence-based PPI prediction techniques have used well known ML tools such as SVMs and RFCs, but the field started to shift toward novel DL within the past few years because of the advantages neural networks offer in terms of performance. Coupled with the recent advancements with the attention mechanism, neural networks have become significantly more powerful compared to classical ML algorithms. Indeed, the vast number of architectures developed for text-based problem such as text mining and language translation can be leveraged toward PPI prediction. Strategies used with neural networks such as transfer learning [81] and model adaptation allows for an easy transition from text-based models to PPI predictions using pairs of protein sequences as input [42, 48]. In addition, research has shown that data processing such as using metric representations or embeddings (see Sections 3.1 and 3.2) for neural networks can reduce experimental noise or unimportant features and highlight important features in the data, which leads to a reduced training time [82, 83].

Below, we discuss the shift toward neural network-based models in the protein sequence-to-PPI prediction field and the different strategies taken by different authors to improve predict performance. Sequence-based models can be distinguished mainly on what representation of the protein sequences is used as inputs into the neural networks. Since input features affect the neural network architectures that can be used, careful consideration needs to be given to how on how best to represent interacting polypeptides. When it comes to representing protein sequences for input into neural networks, the field has developed three different encoding strategies: metric representations, text or neural network embeddings, or simply raw protein sequences as input. Figure 4 illustrates the general pipeline of many models that use protein sequences solely.

3.1 | Metric representations

Distilling protein sequences to metric representations not only reduces training time but also avoids deeper networks, which are usually necessary to capture interactions. It is important to mention that extracting these features from protein sequences is a relatively simple task. Metric representations involve distilling the raw protein sequence into statistical, physical, or chemical properties and seems to be the most popular method in the field currently. These representations include properties such as hydrophobicity, secondary structures, backbone angles, amino acid composition (AAC), and many other features [11, 20, 28, 33, 34, 42–44, 80, 84–87]. Other commonly used features including AAC and conjoint triad (CT) provide a global representation of the protein, while also fixing the length of the input feature [4, 34]. While a fixed-length input is important for most neural network architectures, RNNs do not require a fixed-length input. When one considers the higher payoff and lower effort needed to generate these metrics, it becomes evident why this approach is among the most popular in the field.

3.1.1 | Case example of metric representations and attention: SDNN-PPI

As an example of systems that use metric representations, we will review the design of SDNN-PPI [4]. SDNN-PPI is a neural network-based method that pre-processes protein sequences into metric representations before their input to the neural network, producing a binary PPI prediction output. The three derived features are an AAC, a CT distribution, and an autocovariance (AC) which considers the proximity effects of nearby amino acids. To obtain the CT, SDNN-PPI first clusters the amino acids into seven different clusters based on the

biophysical properties of their side chains and dipoles: if two amino acids are in the same cluster, they are treated the same. The CT score is then calculated as the distribution of amino acid triplets with respect to these clusters. The AC is determined by replacing the amino acids in an input protein sequence with a number corresponding to some basic biophysical property (i.e., hydrophobicity, hydrophilicity, net charge index, polarity, polarizability, soluble surface area, and side chains). The AC of each such sequence is calculated and used as the third input feature to SDNN-PPI.

The SDNN-PPI architecture model architecture is based on a feed-forward network with six fully connected layers. The architecture itself is simple in that it uses one of the most basic layers in DL – a fully connected layer that also includes the use of attention. The authors tested multiple different encodings of protein sequences and were able to get an area under the receiving operator (ROC) curve (AUC) value of 0.986 when the algorithm was tested on multiple high confidence PPI datasets. ROC is a metric used for classification methods that plots the true positive rate as a function of the false positive rate. AUC is the area under this ROC curve – a metric that measures the model's performance across all thresholds for classification. The highest AUC score possible is 1.00, meaning the 0.986 achieved by SDNN-PPI is high. The authors used the *S. cerevisiae* dataset to evaluate SDNN-PPI's performance. While the authors evaluated across different combination of metrics, no comparison to other methods of encoding such as neural network sequences or raw protein embeddings [4].

SDNN-PPI highlights the power of the attention mechanism, namely achieving excellent performance in PPI prediction based purely on sequences and a simple DL architecture. An advancement in attention networks is influencing the field heavily as the most recent papers have started to shift to architectures with the self-attention mechanism, and most recent papers have largely shifted to these attention mechanisms over the span of a few years. SDNN-PPI also highlights the trajectory in which protein sequence pipelines are moving: pipelines that leverage attention mechanisms along with readily available protein sequences are able to produce remarkably accurate results with only the primary protein structure. However, as the authors have pointed out, metric representations do not provide comprehensive protein characterization such as structural, evolutionary, and protein-residue relationship information [4]. In addition, information regarding how the data for training and test set was not specified, which means that information leakage, as described previously in the Methodological Background, could have occurred.

3.2 | Embeddings

The next strategy for protein sequence representation is the use of embeddings and encoders. Encoders such as one hot encodings can be used to represent protein sequences, while embeddings such as Word2Vec, FastText, and BERT [7, 34, 43, 44, 48, 67, 88] are commonly used to transform the protein sequence into an interpretable feature for a neural network. The main advantage to this strategy is that it keeps some resemblance of the entire protein sequence while also providing an interpretable feature representation for a neural network [43]. In addition, neural network embeddings can project an input sequence to a lower dimension. The advantage of this projection is that it can reduce noise and extract important features from the input [82, 83]. Conversely, the main drawback to this approach is the issue of interpretability: while metric-derived features have meaning (since they represent statistical or biophysical representations of the protein), embeddings from neural networks create features that lead to output representations that do not have direct interpretation. While this disadvantage may not affect how well a model predicts PPI using these embeddings, it causes issues when trying to interpret the features the model is using to make these predictions (black box phenomena).

3.2.1 | Case example of embeddings and encodings: GOSeqPPI

An example of embeddings and encodings in a PPI pipeline is GOSeqPPI [43]. GOSeqPPI incorporates both a neural network embedding and a text encoding to represent both the protein sequence and associated gene ontology (GO) annotations [89, 90] the protein into a neural network architecture. To represent the protein sequence, GOSeqPPI uses a one hot encoding, distilling the protein sequence into a $(20 \times L)$ – dimensional feature vector. Twenty for the number of possible amino acids and L for the length of the protein sequence. A pre-trained NCBI-blueBERT model [91], which is a version of the BERT model that was specifically trained on biomedical text data, was used to distill the GO annotations into a $(768 \times N)$ – dimensional feature vector. The one hot encoding and the BERT embedding are passed into a CNN layer and an LSTM layer to combine the features from the protein sequence and GO annotations to create an embedded representation of pairs of proteins. Finally these feature representations are passed into a fully connected network layer with an attention mechanism for a PPI prediction.

Compared to extracted protein features, GOSeqPPI's embeddings and encodings have a distinct advantage: the ability to incorporate additional information. The BERT model extracted important features the GO annotations and was able to use them for PPI predictions. Because neural network embeddings project the input data into a lower dimension, they reduce noise and capture important features. Indeed, compared against other models such as PIPR, GoSeqPPI shows improved accuracy across multiple datasets. However, this method makes the extracted features uninterpretable to the human eye.

3.3 | Raw protein sequences

The research in PPI predictions seems focused primarily on global interactions: a binary output classification on whether two proteins are interacting. However, some of the research looks at local interactions: determining which residues from a pair of proteins are interacting. The literature for predicting residue interaction interfaces is not as rich as

inferring global PPIs. Most models that predict interaction networks tend to use statistical and physiochemical representations for PPI predictions. This decision makes sense since global interaction models are not concerned with the specific residues of the two interacting proteins. However, models that predict local interaction tend to use a combination of statistical, physiochemical representations as well as some representation of the overall protein sequence that captures local features of the protein (e.g., overall fold or domains). As described in the previous sections, protein sequence representations encompass encoding methods such as metric representations, text embeddings, and neural network feature embeddings, but some groups have also leveraged raw protein sequences [39, 42, 44, 67, 86, 87]. Using unprocessed protein sequences for PPI prediction creates an issue for neural network architectures since most models depend on an input of fixed length. This issue means that the architectures used in these models must either use RNNs, which can handle variable length inputs or else somehow fix the input lengths of a given protein sequence. Given the limitations to using RNNs noted before, efforts in this area opt to fix the input length of the proteins using neural network embedding models, text embeddings, or a pre-determined length [39, 44, 87]. Overall, however, these strategies have seen less success than approaches that alter the protein sequence.

3.3.1 | Case example of using protein sequences: ctP2ISP

ctP2ISP [39] is a pipeline that interprets protein sequences to predict protein interaction sites between a pair of interacting protein at the individual amino acid residue level. To achieve this precision, ctP2ISP inputs six different features, representing the physicochemical properties of each protein to the neural network. The first input, covering the first 500 residues of a protein consists of a number from 1 to 20, representing different amino acids. If the protein is less than 500 residues, the input feature is padded with zeros. The algorithm then uses SPIDER3 [92], a protein structure prediction method, to generate information about a protein's secondary structure, including its solvent accessible surface area, and peptide backbone angles. This information is presented as a 1D vector to the neural network pipeline. The other two features are a position-specific scoring matrix, and basic biophysical properties (such as charge, volume, and hydrophobicity).

ctP2ISP separates itself into a local and global block for semantic feature mining. The global block is the part of the architecture that uses global metrics of the protein sequence along with the 500 amino acid sequence of the protein. The local block uses the 30 amino acid fragment for interaction site prediction. Even though both blocks contain transformer layers, the input to the global block consists of the entire 500 residue protein sequence, while the input to the local block is a 30 residue protein subsequence from using a sliding window. The sliding window allows for data augmentation, generating more data to train and test the model without accessing more datasets. Using protein sequences is not necessary for global PPI predictions but is most likely needed for determining local protein interactions. The results are concatenated and sent into fully connected layers for target classification.

The advantage of ctP2ISP's architecture means that all protein sequence inputs are of a fixed-length, allowing the use of the CNN architecture instead of an RNN, but as noted above, other workarounds may perform better to represent an entire protein sequence without sacrificing the ability to use other architectures.

4 | PROTEIN STRUCTURE FOR PPI PREDICTIONS

A protein's 3D structure is a result of biophysical interactions among the primary amino acid sequence. The side chains on the amino acid sequence make energetically favorable contacts with other residues in the sequence. The types of interactions that drive protein folding include hydrogen bond formations, electrostatic interactions, and van der Waals [1]. Experimentally determined protein structures curated in the PDB display physically interacting residues as a graph structure. However, deducing protein structures is more complicated than determining protein sequences, which is why protein sequences are more readily available compared to protein structures [78, 79]. Nevertheless, innovative computational methods based on DL for determining protein structure has recently become a popular field.

In contrast to physics-based methods like ClusPro [93], which depend on free energy minimizations for determining how a protein folds, ground-breaking methods like AlphaFold (AF) use multiple sequence alignments (MSAs) to gain insight into protein structure [6]. Because of AF's recent prominent success in the protein structure prediction field, an AF derived Protein Database was produced, consisting of around 200 million protein structure predictions [94].

Recently, the PPI prediction field has started to integrate protein structures, known ones from PDB or predicted ones from the AF Protein Database, into computational pipelines to see if they improve prediction performance. The general idea behind using protein structure for PPI prediction is that a protein's structure contains key features that can be extracted and then used to compare for complimentary (i.e., lock-and-key fit) against another protein's structure for PPI prediction.

Below, we discuss how protein structure has been leveraged for PPI prediction. The field so far has evolved two main strategies for PPI prediction using structure: PPI prediction using AF-inspired models and graphical representations based on known protein structures. Figure 5 shows the general pipeline of models in this part of the field.

4.1 | AlphaFold

Historically, the implementation of protein structure has been more challenging compared to the use of protein sequences. However, the pioneering AF tool [6] has increased activity around structure for PPI predictions. By demonstrating the power of attention-based DL, the work in the PPI prediction field has now shifted to using this

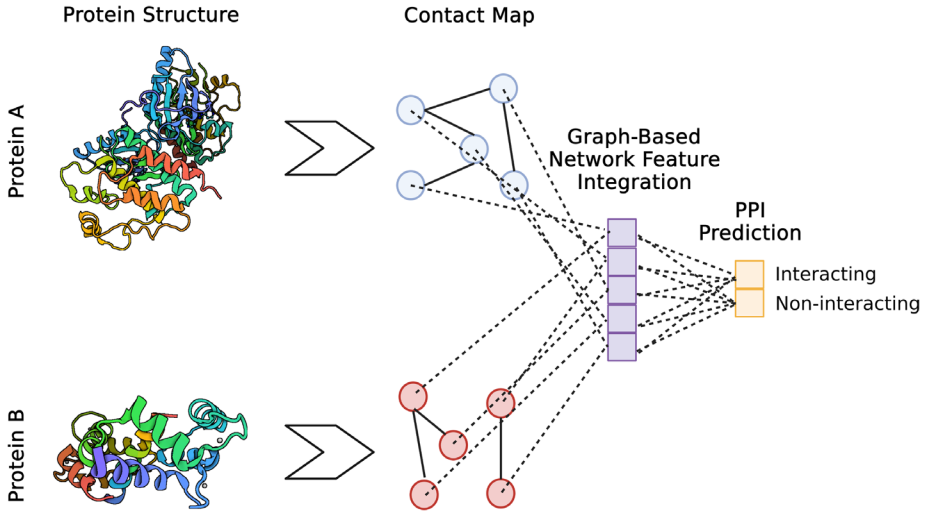

FIGURE 5 General pipeline of PPI prediction models using protein structure models. The input is the 3D structure of a pair of proteins. Protein structures consist of experimentally verified or computationally predicted structures. These 3D structures are converted to residue contact graphs to make it an interpretable graph for the graph neural network architecture. The graph neural network synthesizes the two graphs into a feature vector, which is then used for a PPI prediction.

innovative DL mechanism. AF has moved the PPI field forward by both leveraging the power of DL while also making use of accurate large-scale protein structure models for PPI prediction. Notably, use of protein structure for PPI prediction is no longer restricted to proteins with experimentally derived protein structure, rather researchers can use computational protein structure models for PPI predictions, leveraging the AF Protein Structure Database collection of about 200 million predicted protein structure [6, 94].

While the advent of the AF Protein Structure Database and AF-Multimer is inspiring, considerable progress in the PPI prediction field protein structure-based PPI prediction is still not as extensive as studies using protein sequences. It is also important to mention that protein structure models do not have the same vast starting foundation that protein sequence models had, which draws from extensive previous research with text-based tools, so much of the field is currently working with off-the-shelf models. Rather, existing structure-based methods still depend on protein sequence but use it in conjunction with protein structure information.

4.2 | Multiple sequence alignments

AF's success comes partly from the power of MSA alignment of three or more homologous protein sequences from a diverse set of species to both define similar functions in different organisms and identify boundaries on sequence variation, which reflect the protein's 3D structure [95, 96]. Because of this detectable relationship between homologous protein sequences, MSAs have been widely used to identify evolutionary relationships needed for protein structure prediction. More specifically, MSAs can reveal both global and local structural features, including secondary structures, backbone angles, and even residue–residue interactions [96]. MSAs and other evolutionary information have been popular for protein structure prediction for some time [97–99], but the emergence of the attention mechanism in DL tools such as AF truly allow this information to be extrapolated for PPI predictions.

The success of AF has inspired an AF-Multimer model, which produces medium quality protein docking structures at scale [5]. Because of its limited reliability, using AF features directly for PPI prediction was somewhat disappointing initially. Nevertheless, the AF-Multimer model has been used as a high throughput method to generate candidate PPIs and even multi-protein complexes [100]. Researchers have found that using the AF model in conjunction with paired MSAs yields the best results, achieving an AUC value up to 0.87 when predicting PPIs for *E. coli* [101]. However, AF has a long run time to find the necessary sequences for the MSA needed for the model. This drawback means that performing modifications to AF architecture to enhance PPI predictions will require not only resources to handle AF's computationally intensity but also long run times, which makes this strategy not as feasible compared to using pure protein sequences for PPI prediction. Techniques to speed up AF's MSA generation have been reported [101, 102], but, the long run time still pose an issue.

In theory, long run times can be circumvented by using protein structure predictions in the AF Protein Database, but this strategy means that the model will not be able to use intermediate computed features in AF but rather just AF's output. Hence, with work in this part of the field is still in its early stages, literature for using AF-inspired PPI prediction models remains sparse.

4.2.1 | Case example of MSAs: PITHIA

An example of a neural network architecture incorporating MSAs is PITHIA [67], which uses not only MSAs but also uses other side

features such as embeddings from protein language models and other representations such as a PSSM. While PITHIA does not exactly make PPI predictions, it can find protein interaction sites on a singular protein.

The authors tested the performance of multiple different ML architectures including a multilayer perceptron, an RNN, a CNN, and a transformer with self-attention with the latter performing the best. Surprisingly, however, when testing the impact of additional features such as a PSSM, physiochemical characteristics, or evolutionary conservation, the use of MSAs alone yielded the best results [67]. PITHIA demonstrates the power of MSAs within the attention mechanism compared against other architectures and features.

4.2.2 | Case example of AF output for PPI prediction: TAGPPI

Using a slightly different approach, some recent work in the field has looked at using parts of AF instead of MSAs for protein structure prediction. TAGPPI, for example, incorporates AF's protein structure prediction by creating a residue level contact map to pass onto a graph neural network with an attention mechanism – referred to as a graph attention network (GAT), along with a CNN to interpret the protein sequence after pre-processing. The authors tested multiple variations of their method, including versions that used the CNN and GAT layers independently versus together. They found that the addition of the structure feature provided by AF provided the highest accuracy [18]. It is important to note that this model incorporates both sequence and structure, and compared against other models such as PIPR, a neural network model that only uses protein sequences, TAGPPI improved PPI prediction performance albeit marginally [18].

The combined use of protein sequence and protein structure in TAGPPI somewhat obscures the impact of protein structure on PPI prediction accuracy. To assess the role of protein structure for PPI predictions, it would be helpful to look at a pipeline that does not look at direct sequence information and how this impacts the model performance.

4.2.3 | Case example of pure protein structure for PPI prediction: SGPPI

SGPPI uses protein structures from the AF Protein Structure Database and additional information regarding protein secondary structures but does not use a direct metric representation of the protein sequence. The protein structure is converted into a contact map and passed onto a graph convolutional neural network (GCN) for a PPI prediction. While the authors did not report the overall accuracy of their model, they did indicate an F1-score of 0.375 by a 10-fold cross validation on Pan's dataset [14]. An F1-Score is a function of the model's sensitivity and specificity; the closer the score is to 1, the better the model's performance. Pan's dataset is a generated dataset based on the Human Protein References Database and is commonly used for benchmarking [14]. While each model was tested with a different benchmark, these results suggest using protein sequence has a greater impact on performance relative to protein structure, again stemming from the solid foundation provided by previous natural language processing research. Since protein structure models have lagged behind protein sequence models, the difference in performance could result from the individual field being able to move forward more quickly and is not a representation of the inherent utility for PPI predictions.

4.3 | Non-AF methods

Multiple models have used known protein structures instead of putative structures computationally derived by AF for PPI predictions. The general methodology is to convert the structures into a graph, usually a residue contact map, and apply a graph neural network to extract key features to predict PPIs [22, 57, 103].

4.3.1 | Case example of using experimentally derived protein structures: Struct2Graph

Struct2Graph is a network model that converts experimentally-derived PDB structure files into graphs. These graphs are then passed into a GAT to identify similarities in the graphs. The authors report that Struct2Graph outperforms the other state-of-the-art models using an attention network with protein structures, achieving a claimed accuracy $\geq$98.89% by five-fold cross validation [57]. These results indicate that introducing known structures into a PPI prediction model can increase prediction performance. However, the AF Protein Database currently has about 200 million proteins, while the PDB consists of only 200,000 [78, 94]. Hence, models that work with predicted protein structures have a harder problem since they scale to 100-fold more proteins. Because PDB is much smaller (fewer proteins to be trained or tested on), there is inherently less variability that neural network architectures need to capture, so the problem for known protein structures is an easier task. On the other hand, working with known protein structures is intrinsically limiting. Therefore, for the field to move forward and incorporate inferred protein structure, improvements to AF-inspired models are needed for a highly accurate yet scalable method for PPI predictions.

While still dynamic, a few papers have successfully combined attention networks with protein structure information [57, 103]. So far, models that incorporate known protein structures seem to generate better results compared to those using the predicted protein structures [57, 103]. Yet, the AF model is an important step forward for PPI prediction field as it allows structural information to be extracted for PPIs with only protein sequence available. The attention mechanism seems to provide the best performance – one recent study found that the GAT outperformed the GCN [103]. However, obstacles hindering faster progress include the MSA alignment because the process to finding appropriate protein sequences is computationally expensive [102].

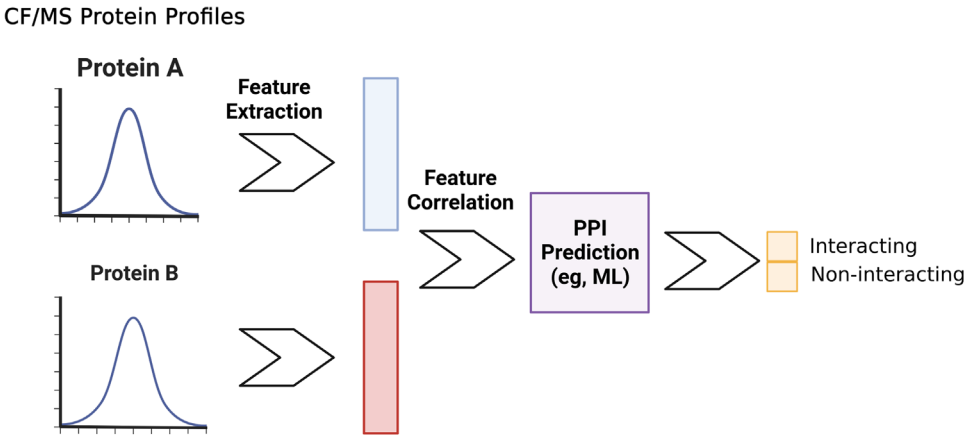

FIGURE 6 General pipeline of PPI prediction models using co-fractionation to mass spectrometry (CF/MS) data. The input is a pair of protein profiles. A feature extraction step does data processing to reduce the influence of experimental noise. The processed profiles are converted to correlations and then used as input to some PPI prediction step such as a machine learning (ML) model. The prediction step outputs a PPI prediction.

5 | CF/MS DATA FOR PPI PREDICTIONS

Biochemical co-fractionation to mass spectrometry (CF/MS) is a powerful experimental method for mapping PPIs on a large-scale that critically depends on extensive computational modeling. In CF/MS experiments, intact soluble protein complexes in a cellular lysate are fractionated by high performance liquid chromatography (HPLC) prior to proteolysis and standard denaturing liquid chromatography mass spectrometry (LCMS) [104]. Since subunits of stable multi-subunit complexes are expected to co-fractionate together, bioinformatics pipelines can be used to compare pairwise sets of protein profiles with highly correlated pairs used to predict PPIs. One key advantage of CF/MS is that it can be to examine different experimental contexts. However, a major challenge is chance co-elution, which can lead to incorrectly predicting interactions among functionally unrelated proteins. Multiple pipelines have been derived to interpret CF/MS data to predict PPIs by addressing the issue of chance co-elution [49, 60, 65].

From a computational perspective, the individual protein profiles recorded by CF/MS can be represented in three different dimensions: peptides (detected proteolytic sequences), fractions (HPLC retention times), and conditions (experimental variables) [65]. Two key steps are data processing to remove inherent sources of experimental noise that can produce error and protein correlation analysis. ML models such as SVMs, RFCs, and Naive Bayes classifiers have been introduced to best interpret CF/MS data [46, 49, 60, 65]. The inputs to these models are pre-processed MS data to predict PPIs. Some pipelines use additional information such as functional annotations to aid in eliminating spurious correlations [65]. Below, we discuss some current strategies for CF/MS data processing and PPI predictions while noting how the field can leverage recent advancements in neural networks. Figure 6 demonstrates the general methodology of models that use CF/MS protein profiles for PPI prediction.

5.1 | Data processing

Data processing is an important step for CF/MS analysis since this step aims to enhance signal-to-noise during subsequent correlation analysis to better detect PPIs. A multitude of different strategies have been introduced, including data normalizing, correlation analysis, and signal processing. For data normalization, the data is scaled to account for spurious measurement variations. Z-score scaling and fitting Gaussian models are some examples of ways to reduce experimental variance [60, 65, 105].

Because interacting proteins co-elute together, a common strategy is to find correlations between protein elution profiles. Correlation metrics include Jaccard, Euclidean distance, and Bayes correlation [65]. Determining coordinate changes in protein abundance (e.g., HPLC peak height, width, retention time) is another strategy for characterizing CF/MS profiles for PPI predictions [46, 60].

5.1.1 | Case example of CF/MS pipeline: EPIC

One of the first tools to process CF/MS data, EPIC (elution profile-based inference of complexes) [65] calculates eight different correlation scores between all proteins profiled in an experiment. These values are then used as an input to an ML engine (RFC and SVM) for PPI prediction. These classifiers are trained by using annotated protein complexes obtained from the CORUM database – if two proteins are curated to same complex, they are deemed as expected PPI. When fully optimized, EPIC achieves an overall accuracy score of 0.65 when applied to *C. elegans* data [65]. Given the accuracy of EPIC is not as high as some of the computational due to chance co-elution, CF/MS data can be used to reveal dynamic rewiring of PPI networks, which is hard to infer computationally.

5.2 | Improving performance in CF/MS field

Notably, CF/MS-based models have yet to exploit advanced neural network architectures. Therefore, there is ample opportunity to improve the accuracy of CF/MS-based predictions by using neural networks and incorporating other sources of information. The first point to consider is neural networks are capable increasing the signal-to-noise ratio in CF/MS. Autoencoders and transformers (see Section 2) are unsupervised and semi-supervised architectures, respectively, that reduce noise present in the inputs. This capability makes them an ideal candidate architecture for a CF/MS prediction pipeline as chance co-elution is one of the leading issues hindering CF/MS bioinformatics pipelines. Transformers also incorporate attention into their architecture [73].

Another point to consider is the use of additional information to aid CF/MS predictions. There are multiple examples of PPI predictions pipelines that use additional information to aid in prediction such as EPIC using functional annotations [65] and GOSeqPPI using GO annotations [43]. CF/MS data, specifically, can benefit from additional information in order to reduce chance co-elution. Neural networks are useful in this instance as they can seamlessly integrate multiple sources of information into a PPI prediction. As an example, a text-based neural network model was used to interpret GO annotations in the GOSeqPPI pipeline [43], and a similar approach can be done with CF/MS data. Protein sequence models have shown success, and integrating protein sequences with CF/MS data is a feasible strategy to improve accuracy in CF/MS predictions.

6 | DISCUSSION

Overall, attention networks have been impactful in the PPI prediction field. The trend in the field is that the current models the field is publishing involve attention. PITHIA demonstrated that the use of the transformer architecture with attention yielded the best results when testing different architectures of the model [67]. Therefore, for the foreseeable future, in order for models to compete with the current state-of-the-art, they will need to incorporate an attention mechanism.

When reviewing the current state of the PPI prediction field, it seems as though AF has had an important impact on the work being done. Neural networks that use protein sequence and/or protein structure have started to use attention in their architectures after AF's publication. While AF has been influential with PPI prediction models using protein sequences and structures, the CF/MS field has barely started to move toward the use of neural networks. There exists an opportunity to progress the CF/MS field through the use of attention networks. Because of chance co-elution, it may be beneficial for the CF/MS field to expand on PPI prediction architectures that use protein sequences or protein structures. Protein sequence models specifically have seen some success, so adapting an architecture and incorporating CF/MS data through an autoencoder, to enhance the signal-to-noise ratio, can be an important first step to progress the CF/MS PPI prediction field.

When comparing PPI prediction models that use protein sequences versus those that use protein structure models, these models seem to yield better results, but this is not a reflection of their superiority over models that use protein structures but rather possibly an artifact. Protein sequence models seem to have a more consistent set of benchmarking datasets based on well-studied organisms. Along with these datasets and the wealth of models from the natural language processing field, the protein sequence PPI prediction field has the necessary tools to create well-performing models. While protein structure models have not moved as far, the AF Protein Database [94] can make way for models that are more applicable to more proteins.

Overall, the PPI prediction field has moved far within the past few years and has the potential to move further within the next few years too. Protein sequence PPI prediction models have leveraged natural language processing models and created a solid foundation for the field to move forward. It may be best for future models to start incorporating multiple sources of information, leveraging the advantages from each data input: amino acid residue information from protein sequences, protein characterization from protein structures, and different experimental contexts from CF/MS.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Andrew Emili https://orcid.org/0000-0001-8995-246X

REFERENCES

1. Stollar, E. J., & Smith, D. P. (2020). Uncovering protein structure. *Essays in Biochemistry*, *64*(4), 649–680. https://doi.org/10.1042/ebc20190042
2. Brückner, A., Polge, C., Lentze, N., Auerbach, D., & Schlattner, U. (2009). Yeast two-hybrid, a powerful tool for systems biology. *International Journal of Molecular Sciences*, *10*(6), 2763–2788. https://doi.org/10.3390/ijms10062763
3. Sutandy, F. X. R., Qian, J., Chen, C., & Zhu, H. (2013). Overview of protein microarrays. *Current Protocols in Protein Science*, *72*(1), 27.1.1–27.1.16. https://doi.org/10.1002/0471140864.ps2701s72
4. Li, X., Han, P., Wang, G., Chen, W., Wang, S., & Song, T. (2022). SDNN-PPI: Self-attention with deep neural network effect on protein-protein interaction prediction. *BMC Genomics*, *23*(1), 474. https://doi.org/10.1186/s12864-022-08687-2
5. Evans, R. et al. (2021). Protein complex prediction with AlphaFold-Multimer. *bioRxiv*. p. 2021.10.04.463034. https://doi.org/10.1101/2021.10.04.463034
6. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2
7. Yao, Y., Du, X., Diao, Y., & Zhu, H. (2019). An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, *7*, e7126. https://doi.org/10.7717/peerj.7126

8. Guo, Y., Yu, L., Wen, Z., & Li, M. (2008). Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, *36*(9), 3025–3030. https://doi.org/10.1093/nar/gkn159
9. Huang, Y.-A., You, Z.-H., Gao, X., Wong, L., & Wang, L. (2015). Using weighted sparse representation model combined with discrete cosine transformation to predict protein-protein interactions from protein sequence. *BioMed Research International*, *2015*, 902198. https://doi.org/10.1155/2015/902198
10. Zhou, Y. Z., Gao, Y., & Zheng, Y. Y. (2011). Prediction of protein-protein interactions using local description of amino acid sequence. In: *Advances in computer science and education applications*. Communications in Computer and Information Science (pp. 254–262). Springer, Berlin Heidelberg.
11. Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y., & Zhang, Y. (2017). DeepPPI: Boosting prediction of protein–protein interactions with deep neural networks. *Journal of Chemical Information and Modeling*, *57*(6), 1499–1510. https://doi.org/10.1021/acs.jcim.7b00028
12. You, Z.-H., Zhu, L., Zheng, C.-H., Yu, H.-J., Deng, S.-P., & Ji, Z. (2014). Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC Bioinformatics*, *15*(Suppl 15), S9. https://doi.org/10.1186/1471-2105-15-s15-s9
13. Martin, S., Roe, D., & Faulon, J.-L. (2005). Predicting protein–protein interactions using signature products. *Bioinformatics*, *21*(2), 218–226. https://doi.org/10.1093/bioinformatics/bth483
14. Huang, Y., Wuchty, S., Zhou, Y., & Zhang, Z. (2023). SGPPI: Structure-aware prediction of protein–protein interactions in rig- orous conditions with graph convolutional network. *Briefings in Bioinformatics*, bbad020. https://doi.org/10.1093/bib/bbad020
15. Hamp, T., & Rost, B. (2015). Evolutionary profiles improve protein–protein interaction prediction from sequence. *Bioinformatics*, *31*(12), 1945–1950. https://doi.org/10.1093/bioinformatics/btv077
16. Luck, K., Kim, D.-K., Lambourne, L., Spirohn, K., Begg, B. E., Bian, W., Brignall, R., Cafarelli, T., Campos-Laborie, F. J., Charloteaux, B., Choi, D., Coté, A. G., Daley, M., Deimling, S., Desbuleux, A., Dricot, A., Gebbia, M., Hardy, M. F., Kishore, N., … Calderwood, M. A. (2020). A reference map of the human binary protein interactome. *Nature*, *580*(7803), 402–408.
17. Pan, X.-Y., Zhang, Y.-N., & Shen, H.-B. (2010). Large-scale prediction of human protein—protein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, *9*(10), 4992–5001. https://doi.org/10.1021/pr100618t
18. Song, B., Luo, X., Luo, X., Liu, Y., Niu, Z., & Zeng, X. (2022). Learning spatial structures of proteins improves protein–protein interaction prediction. *Briefings in Bioinformatics*, *23*(2), bbab558. https://doi.org/10.1093/bib/bbab558
19. Salwinski, L. (2004). The database of interacting proteins: 2004 Update. *Nucleic Acids Research*, *32*(Suppl 1), D449–D451.
20. Chen, M., Ju, C. J.-T., Zhou, G., Chen, X., Zhang, T., Chang, K.-W., Zaniolo, C., & Wang, W. (2019). Multifaceted protein–protein interaction prediction based on Siamese residual RCNN. *Bioinformatics*, *35*(14), i305–i314. doi: 10.1093/bioinformatics/btz328
21. Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N. T., Roth, A., Bork, P., Jensen, L. J., & Von Mering, C. (2017). The STRING database in 2017: Quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research*, *45*(D1), D362–D368. https://doi.org/10.1093/nar/gkw937
22. Yuan, Q., Chen, J., Zhao, H., Zhou, Y., & Yang, Y. (2021). Structure-aware protein–protein interaction site prediction using deep graph convolutional network. *Bioinformatics*, *38*(1), 125–132. https://doi.org/10.1093/bioinformatics/btab643
23. Murakami, Y., & Mizuguchi, K. (2010). Applying the Naĺive Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. *Bioinformatics*, *26*(15), 1841–1848. https://doi.org/10.1093/bioinformatics/btq302
24. Dhole, K., Singh, G., Pai, P. P., & Mondal, S. (2014). Sequence-based prediction of protein–protein interaction sites with L1-logreg classifier. *Journal of Theoretical Biology*, *348*, 47–54. https://doi.org/10.1016/j.jtbi.2014.01.028
25. Hashemifar, S., Neyshabur, B., Khan, A. A., & Xu, J. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, *34*(17), i802–i810. doi: 10.1093/bioinformatics/bty573
26. Das, J., & Yu, H. (2012). HINT: High-quality protein interactomes and their applications in understanding human disease. *BMC Systems Biology*, *6*(1), 92–92.
27. Moal, I. H., & Fernt'andez-Recio, J. (2012). SKEMPI: A structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, *28*(20), 2600–2607. https://doi.org/10.1093/bioinformatics/bts489
28. Zhang, L., Yu, G., Xia, D., & Wang, J. (2019). Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing*, *324*, 10–19. doi: 10.1016/j.neucom.2018.02.097
29. You, Z.-H., Huang, W.-Z., Zhang, S., Huang, Y.-A., Yu, C.-Q., & Li, L.-P. (2019). An efficient ensemble learning approach for predicting protein- protein interactions by integrating protein primary sequence and evolutionary information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *16*(3), 809–817.
30. Kosesoy, I., Gok, M., & Oz, C. (2019). A new sequence based encoding for prediction of host–pathogen protein interactions. *Computational Biology and Chemistry*, *78*, 170–177.
31. Chen, C., Zhang, Q., Ma, Q., & Yu, B. (2019). LightGBM-PPI: Predicting protein-protein interactions through light- GBM with multi-information fusion. *Chemometrics and Intelligent Laboratory Systems*, *191*, 54–64.
32. Jha, K., Saha, S., & Singh, H. (2022). Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, *12*(1), 8360. doi: 10.1038/s41598-022-12201-9
33. Sun, T., Zhou, B., Lai, L., & Pei, J. (2017). Sequence-based prediction of protein protein interaction using a deep- learning algorithm. *BMC Bioinformatics*, *18*(1), 277. https://doi.org/10.1186/s12859-017-1700-2
34. Tran, H.-N., Xuan, Q. N. P., & Nguyen, T.-T. (2023). DeepCF-PPI: Improved prediction of protein-protein interactions by combining learned and handcrafted features based on attention mechanisms. *Applied Intelligence*, 1–16. https://doi.org/10.1007/s10489-022-04387-2
35. Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., & Jiang, H. (2007). Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(11), 4337–4341.
36. Xenarios, I. (2002). DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, *30*(1), 303–305.
37. Yao, Y., Du, X., Diao, Y., & Zhu, H. (2019). An integration of deep learning with feature embedding for protein-protein interaction prediction. *PeerJ (San Francisco, CA)*, *7*, e7126–e7126.
38. Yu, B., Chen, C., Wang, X., Yu, Z., Ma, A., & Liu, B. (2021). Prediction of protein–protein interactions based on elastic net and deep forest. *Expert Systems with Applications*, *176*, 114876.
39. Li, K. et al., (2022). ctP2ISP: Protein-protein interaction sites prediction using convolution and transformer with data augmentation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *99*, 1–1. doi: 10.1109/tcbb.2022.3154413
40. Zhang, J., & Kurgan, L. (2019). SCRIBER: Accurate and partner type-specific prediction of protein-binding residues from proteins sequences. *Bioinformatics*, *35*(14), i343–i353.

41. Li, Y., Golding, G. B., & Ilie, L. (2021). DELPHI: Accurate deep ensemble model for protein interaction sites prediction. *Bioinformatics*, *37*(7), 896–904.
42. Zeng, M., Zhang, F., Wu, F.-X., Li, Y., Wang, J., & Li, M. (2019). Protein–protein interaction site prediction through combining local and global features with deep neural networks. *Bioinformatics*, *36*(4), 1114–1120. https://doi.org/10.1093/bioinformatics/btz699
43. Zhao, L., Wang, J., Hu, Y., & Cheng, L. (2020). Conjoint feature representation of GO and protein sequence for PPI prediction based on an inception RNN attention network. *Molecular Therapy – Nucleic Acids*, *22*, 198–208. https://doi.org/10.1016/j.omtn.2020.08.025
44. Manfredi, M., Savojardo, C., Martelli, P. L., & Casadio, R. (2023). ISPRED-SEQ: Deep neural networks and embeddings for predicting interaction sites in protein sequences. *Journal of Molecular Biology*, 167963. https://doi.org/10.1016/j.jmb.2023.167963
45. Hou, Q., Dutilh, B. E., Huynen, M. A., Heringa, J., & Feenstra, K. A. (2015). Sequence specificity between interacting and non-interacting homologs identifies interface residues – A homodimer and monomer use case. *BMC bioinformatics*, *16*(1), 325–325.
46. Rosenberger, G., Heusel, M., Bludau, I., Collins, B. C., Martelli, C., Williams, E. G., Xue, P., Liu, Y., Aebersold, R., & Califano, A. (2020). SECAT: Quantifying protein complex dynamics across cell states by network-centric analysis of SEC-SWATH-MS profiles. *Cell Systems*, *11*(6), 589–607.e8. https://doi.org/10.1016/j.cels.2020.11.006
47. Teleman, J., Röst, H. L., Rosenberger, G., Schmitt, U., Malmström, L., Malmström, J., & Levander, F. (2015). DIANA – Algorithmic improvements for analysis of data-independent acquisition MS data. *Bioinformatics*, *31*(4), 555–562. https://doi.org/10.1093/bioinformatics/btu686
48. Asim, M. N., Ibrahim, M. A., Malik, M. I., Dengel, A., & Ahmed, S. (2022). ADH-PPI: An attention based deep hybrid model for protein protein interaction prediction. *iScience*, *25*, 105169. https://doi.org/10.1016/j.isci.2022.105169
49. Bludau, I., Heusel, M., Frank, M., Rosenberger, G., Hafen, R., Banaei-Esfahani, A., Van Drogen, A., Collins, B. C., Gstaiger, M., & Aebersold, R. (2020). Complex-centric proteome profiling by SEC-SWATH-MS for the parallel detection of hundreds of protein complexes. *Nature Protocols*, *15*(8), 2341–2386. https://doi.org/10.1038/s41596-020-0332-6
50. Rosenberger, G., Koh, C. C., Guo, T., Röst, H. L., Kouvonen, P., Collins, B. C., Heusel, M., Liu, Y., Caron, E., Vichalkovski, A., Faini, M., Schubert, O. T., Faridi, P., Ebhardt, H. A., Matondo, M., Lam, H., Bader, S. L., Campbell, D. S., & Deutsch, E. W. (2014). A repository of assays to quantify 10,000 human proteins by SWATH-MS. *Scientific Data*, *1*(1), 140031–140031.
51. Singh, R., Devkota, K., Sledzieski, S., Berger, B., & Cowen, L. (2022). Topsy-Turvy: Integrating a global view into sequence-based PPI prediction. *Bioinformatics*, *38*(1), i264–i272. https://doi.org/10.1093/bioinformatics/btac258
52. Sledzieski, S. et al. (2021). Sequence-based prediction of protein-protein interactions: A structure-aware interpretable deep learning model. *bioRxiv*. p. 2021.01.22.427866. https://doi.org/10.1101/2021.01.22.427866
53. Ahmed, I., Witbooi, P., & Christoffels, A. (2018). Prediction of human-Bacillus anthracis protein–protein interactions using multi-layer neural network. *Bioinformatics*, *34*(24), 4159–4164. https://doi.org/10.1093/bioinformatics/bty504
54. Cui, G., Fang, C., & Han, K. (2012). Prediction of protein-protein interactions between viruses and human by an SVM model. *BMC Bioinformatics*, *13*(7.S7), S5–S5.
55. Hermjakob, H. (2004). IntAct: An open source molecular interaction database. *Nucleic Acids Research*, *32*(Suppl 1), D452–D455.
56. Wattam, A. R., Abraham, D., Dalay, O., Disz, T. L., Driscoll, T., Gabbard, J. L., Gillespie, J. J., Gough, R., Hix, D., Kenyon, R., Machi, D., Mao, C., Nordberg, E. K., Olson, R., Overbeek, R., Pusch, G. D., Shukla, M., Schulman, J., Stevens, R. L., … Sobral, B. W. (2014). PATRIC, the bacterial bioinformatics database and analysis resource. *Nucleic Acids Research*, *42*, D581–D591.
57. Baranwal, M., Magner, A., Saldinger, J., Turali-Emre, E. S., Elvati, P., Kozarekar, S., Vanepps, J. S., Kotov, N. A., Violi, A., & Hero, A. O. (2022). Struct2Graph: A graph attention network for structure based pre- dictions of protein–protein interactions. *BMC Bioinformatics*, *23*(1), 370. https://doi.org/10.1186/s12859-022-04910-9
58. Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., Del-Toro, N., Duesbury, M., Dumousseau, M., Galeota, E., Hinz, U., Iannuccelli, M., Jagannathan, S., Jimenez, R., Khadake, J., Lagreid, A., … Hermjakob, H. (2014). The MIntAct project–IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, *42*, D358–D363.
59. Szklarczyk, D., Gable, A. L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N. T., Morris, J. H., Bork, P., Jensen, L. J., & Mering, C. V. (2019). STRING v11: Protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Research*, *47*(D1), D607–D613.
60. Stacey, R. G., Skinnider, M. A., Scott, N. E., & Foster, L. J. (2017). A rapid and accurate approach for prediction of interactomes from co-elution data (PrInCE). *BMC Bioinformatics*, *18*(1), 457. https://doi.org/10.1186/s12859-017-1865-8
61. Nepusz, T., Yu, H., & Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, *9*(5), 471–472. https://doi.org/10.1038/nmeth.1938
62. Kristensen, A. R., Gsponer, J., & Foster, L. J. (2012). A high-throughput approach for measuring temporal changes in the interactome. *Nature Methods*, *9*(9), 907–909.
63. Scott, N. E., Rogers, L. D., Prudova, A., Brown, N. F., Fortelny, N., Overall, C. M., & Foster, L. J. (2017). Interactome disassembly during apoptosis occurs independent of caspase cleavage. *Molecular Systems Biology*, *13*(1), 906.
64. Scott, N. E., Brown, L. M., Kristensen, A. R., & Foster, L. J. (2015). Development of a computational framework for the analysis of protein correlation profiling and spatial proteomics experiments. *Journal of Proteomics*, *118*, 112–129.
65. Hu, L. Z., Goebels, F., Tan, J. H., Wolf, E., Kuzmanov, U., Wan, C., Phanse, S., Xu, C., Schertzberg, M., Fraser, A. G., Bader, G. D., & Emili, A. (2019). EPIC: Software toolkit for elution profile-based inference of protein complexes. *Nature Methods*, *16*(8), 737–742. doi: 10.1038/s41592-019-0461-4
66. Singh, G., Dhole, K., Pai, P. P., & Mondal, S. (2014). SPRINGS: Prediction of protein-protein interaction sites using artificial neural networks. *PeerJ Preprints*, *2*, e266v2. doi: 10.7287/peerj.preprints.266v2
67. Hosseini, S. M., & Ilie, L. (2022). PITHIA: Protein interaction site prediction using multiple sequence alignments and attention. *International Journal of Molecular Sciences*, *23*(21), 12814. https://doi.org/10.3390/ijms232112814
68. Mahesh, B. (2020). Machine learning algorithms – a review. *International Journal of Science and Research (IJSR)*, *9*, 381–386.
69. Breiman, L. (1984). Classification and regression trees (1st ed.). Routledge. https://doi.org/10.1201/9781315139470
70. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018
71. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324
72. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

73. Vaswani, A. et al. (2017). Attention is all you need. *arXiv*. eprint: 1706.03762. doi: 10.48550/arxiv.1706.03762
74. Mikolov, T. et al. (2013). Efficient estimation of word representations in vector space. *arXiv*. eprint: 1301.3781. doi: 10.48550/arxiv.1301.3781
75. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051
76. Devlin, J. et al. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*. https://orcid.org/10.48550/arxiv.1810.04805
77. Park, Y., & Marcotte, E. (2012). Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, *9*(12), 1133–1134. https://doi.org/10.1038/nmeth.2254
78. RCSB Protein Data Bank. *PDB statistics: Overall growth of released structures per year*. https://www.rcsb.org/stats/growth/growth-released-structures, Retrieved March 15th, 2023
79. Bateman, A., Martin, M.-J., Orchard, S., Magrane, M., Agivetova, R., Ahmad, S., Alpi, E., Bowler-Barnett, E. H., Britto, R., Bursteinas, B., Bye-A-Jee, H., Coetzee, R., Cukura, A., Da Silva, A., Denny, P., Dogan, T., Ebenezer, T., Fan, J., Castro, L. G., … Teodoro, D. (2020). UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, *49*(D1), D480–D489. https://doi.org/10.1093/nar/gkaa1100
80. Xiang, Z., Gong, W., Li, Z., Yang, X., Wang, J., & Wang, H. (2021). Predicting protein–protein interactions via gated graph attention signed network. *Biomolecules*, *11*(6), 799. https://doi.org/10.3390/biom11060799
81. Zhuan, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. in *Proceedings of the IEEE*, *109*(1), pp. 43–76. https://doi.org/10.1109/JPROC.2020.3004555
82. Hassan, M., & Hassan, I. (2021). Improving artificial neural network based stream-flow forecasting models through data preprocessing. *KSCE Journal of Civil Engineering*, *25*(9), 3583–3595. https://doi.org/10.1007/s12205-021-1859-y
83. Zhou, Q., & Ooka, R. (2021). Influence of data preprocessing on neural network performance for reproducing CFD simulations of non-isothermal indoor airflow distribution. *Energy and Buildings*, *230*, 110525. https://doi.org/10.1016/j.enbuild.2020.110525
84. Ahmed, I., Witbooi, P., & Christoffels, A. (2018). Prediction of human-Bacillus anthracis protein–protein interactions using multi-layer neural network. *Bioinformatics*, *34*(24), 4159–4164. https://doi.org/10.1093/bioinformatics/bty504
85. Debnath, S., & Mollah, A. F. (2022). A supervised machine learning approach for sequence based protein-protein interaction (PPI) prediction. *arXiv*. eprint: 2203.12659. doi: 10.48550/arxiv.2203.12659
86. Wang, S., Chen, W., Han, P., Li, X., & Song, T. (2022). RGN: Residue-based graph attention and convolutional network for protein–protein interaction site prediction. *Journal of Chemical Information and Modeling*, *62*(23), 5961–5974. https://doi.org/10.1021/acs.jcim.2c01092
87. Xie, Z., Deng, X., & Shu, K. (2020). Prediction of protein–protein interaction sites using convolutional neural network and improved data sets. *International Journal of Molecular Sciences*, *21*(2), 467. https://doi.org/10.3390/ijms21020467
88. Sun, T., Zhou, B., Lai, L., & Pei, J. (2017). Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, *18*(1), 277. https://doi.org/10.1186/s12859-017-1700-2
89. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene Ontology: Tool for the unification of biology. *Nature Genetics*, *25*(1), 25–29. https://doi.org/10.1038/75556
90. Carbon, S., Douglass, E., Good, B. M., Unni, D. R., Harris, N. L., Mungall, C. J., Basu, S., Chisholm, R. L., Dodson, R. J., Hartline, E., Fey, P., Thomas, P. D., Albou, L.-P., Ebert, D., Kesling, M. J., Mi, H., Muruganujan, A., Huang, X., Mushayahama, T., … Elser, J. (2020). The Gene Ontology resource: Enriching a GOld mine. *Nucleic Acids Research*, *49*(D1), D325–D334. https://doi.org/10.1093/nar/gkaa1113
91. Peng, Y., Yan, S., & Lu, Z. (2019). Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 5865, Association for Computational Linguistics, Florence, Italy.
92. Heffernan, R., Yang, Y., Paliwal, K., & Zhou, Y. (2017). Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*, *33*(18), 2842–2849. https://doi.org/10.1093/bioinformatics/btx218
93. Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., & Vajda, S. (2017). The ClusPro web server for protein–protein docking. *Nature Protocols*, *12*(2), 255–278. https://doi.org/10.1038/nprot.2016.169
94. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Z-Dek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., … Velankar, S. (2021). AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, *50*(D1), D439–D444. https://doi.org/10.1093/nar/gkab1061
95. Kuhlman, B., & Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, *20*(11), 681–697. https://doi.org/10.1038/s41580-019-0163-x
96. Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., & Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS ONE*, *6*(12), e28766. https://doi.org/10.1371/journal.pone.0028766
97. Cong, Q., Anishchenko, I., Ovchinnikov, S., & Baker, D. (2019). Protein interaction networks revealed by proteome coevolution. *Science*, *365*(6449), 185–189. https://doi.org/10.1126/science.aaw6718
98. Liu, C. H., Li, K.-C., & Yuan, S. (2013). Human protein–protein interaction prediction by a novel sequence-based co-evolution method: Co-evolutionary divergence. *Bioinformatics*, *29*(1), 92–98. doi: 10.1093/bioinformatics/bts620
99. Wang, L., You, Z.-H., Xia, S.-X., Liu, F., Chen, X., Yan, X., & Zhou, Y. (2017). Advancing the prediction accuracy of protein-protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier. *Journal of Theoretical Biology*, *418*, 105–110. https://doi.org/10.1016/j.jtbi.2017.01.003
100. Yu, D., Chojnowski, G., Rosenthal, M., & Kosinski, J. (2022). AlphaPulldown – A Python package for protein-protein interaction screens using AlphaFold-Multimer. https://doi.org/10.1101/2022.08.05.502961
101. Bryant, P., Pozzati, G., & Elofsson, A. (2022). Improved prediction of protein-protein interactions using AlphaFold2. *Nature Communications*, *13*(1), 1265. doi: 10.1038/s41467-022-28865-w
102. Mirdita, M., Schµtze, K., & Moriwaki, Y. (2022). ColabFold – Making protein folding accessible to all. *Nat Methods*, *19*, 679–682. https://doi.org.ezproxy.bu.edu/10.1038/s41592-022-01488-1
103. Jha, K., Saha, S., & Singh, H. (2022). Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, *12*(1), 8360. doi: 10.1038/s41598-022-12201-9

104. Low, T. Y., Syafruddin, S. E., Mohtar, M. A., Vellaichamy, A., Rahman, N. S., Pung, Y.-F., & Tan, C. S. H. (2021). Recent progress in mass spectrometry-based strategies for elucidating protein–protein interactions. *Cellular and Molecular Life Sciences*, *78*(13), 1–15. https://doi.org/10.1007/s00018-021-03856-0
105. Mertens, B. J. A. (2017). Transformation, normalization and batch effect in the analysis of mass spectrometry data for omics studies. (Eds: S. Datta, B. Mertens), *Statistical Analysis of Proteomics, Metabolomics, and Lipidomics Data Using Mass Spectrometry. Frontiers in Probability and the Statistical Sciences*. Springer, Cham. https://doi.org.ezproxy.bu.edu/10.1007/978-3-319-45809-0_1

How to cite this article: Kewalramani, N., Emili, A., & Crovella, M. (2023). State-of-the-art computational methods to predict protein–protein interactions with high accuracy and coverage. *Proteomics*, e2200292.
https://doi.org/10.1002/pmic.202200292