Sequence Representations and Their Utility for Predicting Protein-Protein Interactions

Dhananjay Kimothi, Pravesh Biyani, James M. Hogan, and Melissa J. Davis

Abstract—Protein-Protein Interactions (PPIs) are a crucial mechanism underpinning the function of the cell. So far, a wide range of machine-learning based methods have been proposed for predicting these relationships. Their success is heavily dependent on the construction of the underlying feature vectors, with most using a set of physico-chemical properties derived from the sequence. Few work directly with the sequence itself. In this paper, we explore the utility of sequence embeddings for predicting protein-protein interactions. We construct a protein pair feature vector by concatenating the embeddings of their constituent sequence. These feature vectors are then used as input to a binary classifier to make predictions. To learn sequence embeddings, we use two established Word2Vec based methods - Seq2Vec and BioVec - and we also introduce a novel feature construction method called SuperVecNW. The embeddings generated through SuperVecNW capture some network information in addition to the contextual information present in the sequences. We test the efficacy of our proposed approach on human and yeast PPI datasets and on three well-known networks: CD9, the Ras-Raf-Mek-Erk-Elk-Srf pathway, and a Wnt-related network. We demonstrate that low dimensional sequence embeddings provide better results than most alternative representations based on physico-chemical properties while offering a far simple approach to feature vector construction.

Index Terms—Sequence representation, machine learning, PPI prediction

1 INTRODUCTION

PROTEINS play a central role in cellular function, often mediated through the protein interactions leading to the formation of transient or stable complexes. Such Protein-Protein Interactions (PPIs) vary in their duration and character, being governed principally by electrostatic forces and constrained by molecular structure [1]. Knowledge of PPIs can help determine the function of new proteins and improve our understanding of cellular processes such as DNA transcription and metabolic cycles [2].

Up to now, several high-throughput experimental techniques have been developed for determining the nature and extent of interactions between proteins. These techniques include yeast two-hybrid screens [3], affinity purification [4], protein chips [5] and mass spectrometry [6]. However, these methods share two significant drawbacks: (i) such methods are expensive and time-consuming, and (ii) they are known to exhibit high false-positive rates [7]. These limitations motivated researchers to develop computational methods for PPI characterization. The earliest such approaches were based on utilizing varied information sources related to proteins such as the protein domains [8], [9], [10], gene-co-expression data and tertiary structures [11], [12], [13]. Some other methods also explored the use of functional annotations (GO labels) [14], [15] for predicting PPIs. Although these computational methods demonstrate promising results, they can be applied to only those protein pairs for which the desired meta-information/annotation is available. Such meta-information may not be available for every protein considered. In contrast, with the advancement of sequencing technology, the amino acid sequence (primary structure) of the proteins is readily available, making them a preferable input for PPI prediction approaches. We term these approaches collectively sequence-based approaches for PPI prediction.

Most sequence-based approaches are machine learning based and primarily rely on representing protein sequences as an n-dimensional feature vector. These feature vectors essentially summarize specific attributes of a protein sequence – such as the distribution of the amino acids, their physico-chemical properties [16], or the composition of a localized region [17], [18]. To apply machine learning approaches for PPI prediction task, protein pair needs to be represented as single mathematical entity. Generally, such a representation for a protein pair is obtained by concatenating the feature vector of its constituent protein sequences, hence keeping the information derived from them intact. Although the sequence encoding methods discussed above are commonly used by researchers for PPI prediction problem, they suffer from three significant drawbacks: (i) such methods require prior knowledge of physico-chemical properties, (ii) the properties used are...
not guaranteed to cover the PPI interaction information adequately and (iii) the resulting feature vectors are usually of high dimension, increasing training and inference time and potentially limiting the effectiveness of the model.

One potential solution to these problems is to learn sequence embeddings that capture biological information that may be present in the sequences themselves, and to use them to represent a protein pair. The earliest work on learning biological sequence embeddings by Asgari et al. [19] shows that it is possible to learn embeddings rich in biological information for sub-sequences \((k\text{-mers})\) directly using the sequence data. These authors and subsequently Kimothi et al. [20] also demonstrated the utility of such learning techniques - generally called representation learning (RepL) - for the protein classification task. Following these primary works, other studies have shown the effectiveness of RepL approaches for a range of other downstream bioinformatics tasks [21], [22], [23], [24] such as trans-membrane and localization prediction. Although learning of sequence embeddings and their usages for downstream bioinformatics tasks has been explored now for the last few years, to date such methods have not been studied in detail for PPI prediction problem.

In this paper, we propose to use sequence embeddings learned through Representation learning (RepL) methods to represent a protein pair for making PPI predictions. Similar to the established methods, we represent a protein pair by concatenating the embedding of its constituent sequences. Here, we hypothesize that the relationship between two proteins can be better determined using the contextual information-rich low dimensional sequence embeddings as compared to the high dimensional physico-chemical property based sequence feature vectors. Note that RepL methods encode the contextual (local) information present in the sequences to generate the corresponding low dimensional embedding. The smaller size of sequence embeddings also makes representation learning methods computationally attractive when compared to the available feature construction methods. The computational requirement becomes crucial for large-scale experiments, such as PPI prediction for all possible protein pairs for an organism.

Apart from the new representation for protein pairs, in this paper we also provide an extensive evaluation and comparison with other protein interaction prediction methods. Most of the established methods, including some of the most recent, use a standard cross-validation setup for evaluation. In this setup, the data is split into training and test sets considering the protein pairs but not their constituent proteins. In such a scenario, many proteins - while apparently in different samples may be present both in the training and test set. In consequence, generalisation is affected by the presence or absence of their constituent proteins in the training set. Hence, as noted by Park et al. [25] such an assessment may not be reflective of the ability of the evaluated methods to generalize for paired input.

Given these findings, we rigorously evaluate our proposed approach according to the evaluation strategy proposed by Park et al. [25]. Here the test sets are based on the presence or absence of the individual protein (from the testing sample) in the training set and hence cover different possible scenarios. For experiments, we use benchmark human and yeast datasets provided in [25] and [16]. The experimental results on these datasets show that even the vanilla embedding approaches are capable of achieving better results than most of the physico-chemical property based vector representation. When compared to the recent deep neural network-based approaches applied to sequence data, we find that even a simple framework like ours can give comparable performance though we stress the need to better understand the relationship between sequences and the possibility of their interaction. In summary, the main contributions of this paper are as follows:

- We use sequence embeddings based protein pair representations for protein-protein interaction prediction task. The representation of each protein pair consists of the embeddings of its constituent protein sequences. We demonstrate that such representations of protein pairs yield improved results when compared to most of the available physico-chemical property based protein-pair representations.
- We introduce a new sequence representation learning model - SuperVecNW. It utilizes the known binary relationships (interaction/non-interaction) along with the contextual information to learn sequence embeddings. We use these embeddings to construct representations for protein pairs that are further used for PPI predictions.
- We demonstrate that our approach achieves comparable performance to that of much more complex end-to-end deep learning approaches on the PPI prediction task.

Before discussing the details of the proposed framework, we briefly review the feature construction methods commonly used for the PPI prediction problem.

1.1 Previous Work
The most prominent feature extraction methods include Auto Covariance (AC), Auto Cross Covariance (ACC), Conjoint Triad (CT), and Local Protein Descriptors (LD). AC and ACC based feature vectors are derived by considering protein sequences as time series and computing their auto and cross-correlation properties. The length of the AC and ACC based feature vectors are 420 and 2940, respectively. The conjoint triad method divides the physico-chemical properties of the system into seven categories based on their dipole and volume scale [26], allowing the representation of the protein sequence based on an array of category labels. Once converted, the feature vector of the sequence is given as the frequency of conjoint triads, i.e., the three category labels. For a protein pair, the CT method gives a 686-dimensional feature vector.

The Local Descriptor method uses the same categories of physico-chemical properties as CT. The feature vector is constructed from three descriptors - composition (C), Transition (T), and distribution (D) for local regions. The descriptors calculated for each of the local regions are then stacked together to give a 630-dimensional vector for each protein sequence. Other methods based on local regions include Multi-scale Local Feature descriptors (MLD) [7] and the Multi-scale Continuous and Discontinuous feature set (MCD)[27]. MLD uses the multi-scale decomposition of protein sequences
to account for overlapping local regions, binary coding to produce a 1134-dimensional vector for a protein pair. MCD works on a similar principle as MLD, though it gives a 1764-dimensional vector.

Apart from these methods, approaches such as the physiological Property Response Matrix combined with Local Phase Quantisation descriptor (PR-LPQ) [28] and Substitution Matrix Representation (SMR) [29] use signal and image processing techniques to compute the feature vector for a sequence. These methods generally operate on an intermediate matrix representation of a sequence; PR-LPQ uses physiological-chemical properties of amino acids, whereas SMR uses the BLOSUM62 [30] matrix to construct the representations.

On top of these feature vectors, a number of different machine learning approaches have been applied to characterise PPIs. These methods have included the Support Vector Machine [16], Random Forests [29] and autoencoders [31]. More recently, studies such as DPPI [32], DNN-PPI [33], and PIPR [34] have explored deep learning frameworks for PPI prediction. Note that these newer deep learning-based approaches are end-to-end classification models and do not specifically focus on the feature construction technique. Also, given their deep architecture, these are complex and hence computationally expensive approaches. Since the success of PPI prediction methods is strongly dependent on the feature vectors extracted from the sequences, new methods for feature extraction remain an active area for researchers.

In the sections below, we discuss the proposed framework in detail, followed by experiments and results.

2 PPI Prediction and Proposed Framework

Notation

We denote M protein pairs as $P = \{p_1, p_2, \ldots, p_M\}$, where $p_i$ consist of two samples from a corpus of N protein sequences, $S = \{s_1, s_2, \ldots, s_N\}$. We write $p_i = (s_i, s_j)$, to uniquely map the sequences with samples in $P$, we add a superscript to $s_i$ i.e., $p_i$ is written as $p_i = (s_i^j, s_j^i)$. We denote the vocabulary of $L$ unique k-mers created from S as $K = \{k_1, k_2, \ldots, k_L\}$ and each sequence $s_i = [k_{i_1}, k_{i_2}, \ldots, k_{i_{L_i}}]$ as an ordered list of k-mers, where $L_i$ denotes the count of k-mers. To avoid notation clutter, we use $s_i$ also to denote its tag. Finally, the embeddings of k-mer $k_i$, sequence $s_i$ and protein pair $p_i$ are denoted as $k_i$, $s_i$, and $p_i$ respectively.

2.1 PPI Prediction Problem

PPI prediction is a paired input problem in which a prediction is made about the relationship between two objects - here proteins. Since machine learning approaches generally operate on a single mathematical entity, for paired input problems, the pair is often represented as a combination of the feature vectors for the individual objects. In a setting where $P_{train}, P_{test} \subset P$ are the training and test set of protein pairs respectively and $P_{train} \cap P_{test} = \emptyset$, the task in PPI prediction problem is to determine whether or not the proteins in $p_i = (s_i, s_j) \in P_{test}$ interact given that a label is available for the samples in training set.

In our proposed framework, we use Repl methods, like BioVec, Seq2Vec, SuperVecX to generate the sequence embeddings that are concatenated for a protein pair, and a binary classifier for making predictions. For a better exposition of the proposed framework, in the section below we first give an overview of the biological sequence representation learning (RepL) methods followed by a description of the complete pipeline of the proposed framework.

2.2 Word2Vec Based Representation Learning Methods for Biological Sequences

Advances in word embedding methods in text processing have inspired several studies of sequence encoding in the bioinformatics domain. These encoding schemes treat sequences as ‘sentences’ and the sub-sequences as ‘words.’ Since the sequences are contiguous strings over some alphabet and there is no notion of ‘words’, they are first split by passing a window of size $k$ across them. The sub-sequences ($k$-mers) thus generated are treated as words. For example, for $k = 3$, a dummy protein sequence “MAFSAED” can be split into k-mers: MA, AF, FS, SA, AE, ED. K-mer size is considered as a hyperparameter in the representation learning models. For a fixed $k$, the number of k-mers increases with the length of the sequence in consideration. For a protein of length $L$ and chosen k-mer size $k$, the number of k-mers generated would be $L - k + 1$.

Some of the earliest biological sequence embedding methods include BioVec [19] and Seq2Vec [20]. Both of these papers utilized Word2Vec [35] based architectures for computing the vector representation for protein sequences and demonstrated the utility of such representations for protein family classification tasks. These models are shallow neural networks with one hidden layer, where the weight matrix between the input/output and hidden layer constitutes the n-dimensional representation of k-mers or sequences in $\mathbb{R}^n$ vector space. The architecture for BioVec and Seq2Vec models are provided in Fig. 1.

The training of these models involves a simple prediction mechanism over samples, consisting of $k$-mers and their neighbors (context) randomly picked from the corpus of sequences. For prediction over a sample, either the $k$-mer is predicted from the context or vice-versa as depicted in Fig. 1. In the BioVec model, the sample consists of a $k$-mer and its neighbours (context), whereas for Seq2Vec, the sample also contains the tag associated with the corresponding sequence. Mathematically, by iterating through the samples, the parameters of these models are learned such that negative log-likelihood probabilities of co-occurrence of $k$-mers are minimized. The cost functions that are minimized in BioVec and Seq2Vec are given in Eqs. (1) and (2) respectively

$$J(K) = \sum_{s_j \in S} \log \frac{1}{\sum_{k_i} \Pr[k_i | s_j]}.$$  

$$J(K, S) = \sum_{s_i \in S} \log \frac{1}{\sum_{C_i} \Pr[C_i | s_i]}.$$  

Here, $k_i$ denotes the $j^{th}$ k-mer of $i^{th}$ sequence, $s_i$ and $C_i$ (context) denotes the set of k-mers surrounding $k_i$.

Once the BioVec model is trained, the sequence embedding is computed by linearly combining the embeddings of its constituent k-mers. For Seq2Vec, the sequence embeddings are directly obtained by passing them through the
trained model. This step is also known as the inference step. Note that these approaches are unsupervised and only need the sequence corpus for training.

The followup work by Kimothi et al. [36] presented supervised approaches for learning representations of protein sequences: SuperVec and SuperVecX. These supervised approaches use sequences as well as their associated labels during the training of the models. Similar to Word2Vec, these models are trained via a prediction task. The SuperVec model joins the two prediction units together to learn contextual and label information. The first unit follows the prediction of a k-mer given its context to capture contextual information. The second unit ensures the inclusion of label information in the training process by forcing the network to predict the tags of sequences that share the label with the considered sample. The joint cost function [36] of SuperVec is given as below, where $J(S, K) = \sum_{i=1}^{n} \left( -\log Pr[k_{i,j} | C_{i,j}, s_i] \right)$ and $\gamma \sum_{z \in I_i^+} \log Pr[s_z | s_i]$. Here $I_i^+$ is the set of sequences that share the label with $s_i$. Note that NN1 and NN2 in Eq. (3) represent the two parts of the complete model, i.e., for learning contextual and label information, respectively. As noted before, SuperVec requires the labels of protein sequences for training purposes. In contrast to SuperVec, training of SuperVecX involves the prediction of the class label using all k-mers of given sequences.

In the PPI prediction problem, along with the sequences, the information regarding the relationships (interaction/non-interaction) of proteins in the training set is also available. Using a similar concept as utilized in SuperVec, we propose a new RepL model - SuperVecNW that utilizes the contextual information and binary relationships of proteins to learn sequence representation. The description of SuperVecNW is given below.

SuperVecNW

In using SuperVecNW we make use of the available network information to label the sequences. The neighborhood of a sequence is looked up in the complete interaction network obtained from the training pairs of proteins. For a given sequence, the sequences connected to it in the PPI network are considered to come from the same class (interacting) and others from different (non-interacting) class. The joint cost function for SuperVecNW is same as for SuperVec as given in Eq. (3); the difference is only in the process of deriving class labels for a sample. For example in Fig. 2, for sample $s_3, I_i^+ = s_1, s_7, s_8$ in Eq. (3).

Note that all of these supervised models - SuperVec, SuperVecX and SuperVecNW, differ in their selection of context and word for the prediction task (refer Table 1).

2.3 Proposed Framework

In the proposed framework, along with the sequence embedding techniques discussed above, we use the Random Forest classifier for making predictions. Note that while the framework is generic and, in principle, any binary classifier can be used, we use the Random Forest (RF) Classifier [37] for our experiments due to its computational efficiency and convenience. An RF classifier requires only one
hyper-parameter to be tuned and the method inherently avoid over fitting on the training data.

A Random Forest is a supervised classifier that uses an ensemble of decision trees to make predictions. These decision trees are constructed using bagging (a bootstrap aggregation technique [37]), and random feature selection. In the bagging method, each of the decision trees is trained using randomly drawn subsets of the training set. For random feature selection, the nodes in each decision tree are split based on a random selection of \( m \) features drawn from the \( n \) features available. Here, \( m < n \). For a test sample, the prediction is made based on voting (averaging) the prediction output from each decision tree. The presence of multiple trees in the random forest helps to avoid overfitting and limits the error due to bias. Also, there is virtually no hyper-parameter to tune except for the number of trees.

In this paper, we represent each sequence as a 100-dimensional feature vector, and the pair of proteins as a 200-dimensional vector. To use SuperVecX, we concatenate the sequences, with the result that the protein pair is represented by a single 100 dimensional feature vector. Note the much lower dimensionality of these representations compared to the other established feature vectors discussed in Section 1.1. For interacting and non-interacting pairs, we use the class labels 1 and 0, respectively. Based on the results on a few train-test splits with different choices of the number of trees (100, 200, 500, 1000) – where each of the trees is built by selecting \( m < 200 \) random features at each node – we find 500 to be a good choice for the detests used in this paper. The prediction for a test sample is made based on the voting of the decision trees.

The complete pipeline of the proposed framework is provided in Fig. 3. Here block A shows the training stage of any supervised representation learning model, A1; when trained, we denote it as A2. A1 is trained using the sequences in training set \( S_{\text{train}} \subseteq S \), \( S_{\text{train}} \) constitutes of all the unique sequences present in protein pairs in \( P_{\text{train}} \). Note that for chosen unsupervised models, A1 is trained with all the available unique sequences (\( S \)) in the corpus. Block B is used for making predictions. It is operated in two phases, namely, the training and testing phase. In the training phase, the samples (protein pairs) from \( P_{\text{train}} \) are passed through B1 to generate the embeddings of their constituent sequences. These embeddings are then concatenated to finally create the feature vector for the protein pairs, which are further used to train the binary classifier, B3. In the testing phase, similar to the training phase, the feature vector for any given test pair from \( P_{\text{test}} \) is generated by passing them through B1 and B2. This feature vector is finally given to the trained binary classifier to predict whether the proteins interacts. It is important to note that the datasets used for Block A and Block B are the same, i.e., the representation learning model is trained on the same dataset (yeast/human), on which further experiments are performed following Block B.

### 3 Results and Discussion

In this section, we present the results of our approach on the benchmark human and yeast PPI datasets provided by Park et al. [25]. These datasets were originally obtained from the protein interaction network analysis platform [38] and were further refined to contain only those sequences that share up to 40% identity. We also provide the results on another widely used benchmark yeast dataset provided by Guo et al. [16]; this dataset was originally obtained from DIP [39] and then refined such that the majority of the remaining protein pairs share less than 40% pairwise sequence identity.

#### 3.1 Evaluation Scheme

For the experiments, we follow the evaluation scheme proposed explicitly for the paired input problem by Park et al. [25]. In paired input problems, the inference is made for a pair of objects (e.g., the PPI prediction) rather than for the single object (e.g., protein family prediction). Based on their experimental results, the authors in [25] showed that the performance of any method for the PPI prediction problem is affected by the presence of the individual proteins of the test sample in the training set, providing better results for the pairs that share protein(s) in the training data.

We cannot, therefore, expect a reliable assessment of generalisation if performance is evaluated on a test set dominated by samples that share components with the training data. Considering this issue, Park et al. [25] suggested that the paired-input prediction methods should be evaluated on three different test classes, namely C1, C2, and C3. Here C1 constitutes the test samples for which both proteins are present in the training data, whereas, in C2, only those test samples are considered that share exactly one protein with the training data. Finally, the C3 test class is constructed from those test samples which share none of the constituent proteins with the training samples, making it the most challenging test class among the three. In contrast, following the standard cross-validation process would split protein pairs data into mutually exclusive and exhaustive training and test set without considering the possibility of possible overlap between individual proteins of protein pairs in the train and test set. So, the CV split test set contains cases from C1, C2 and C3.

Park et al. [25] showed that the datasets previously used for cross-validation are close to the C1 type. These authors also noted that in the HIPPIE database [40], C1-type human

---

### TABLE 1

Context-Word Used per Training Sample \((s_i, k_{ij}, C_{ij})\) for Different Supervised Representation Learning Models

| Models     | NN Architecture | context                                                                 | word |
|------------|-----------------|-------------------------------------------------------------------------|------|
| SuperVec   | NN1             | CBOW                                                                   | \(s_i\) and \(C_{ij}\) \(\{s_j, j = 1, 2, t\}; s_i \) and \(s_j\) have same labels \(k_{ij}\) |
|            | NN2             | Skip-Gram                                                              | \(s_i\) and \(s_j\) are interacting pair \(k_{ij}\) |
| SuperVecNW | NN1             | CBOW                                                                   | \(\{k_{ij}; j = 1, 2, 3, .. | s_i\}; s_i \in L_1(\text{label}(s) \text{for } s_i)\) |
| SuperVecX  | NN1             | CBOW                                                                   | \(l_i\) |

**NN denotes neural network.**
protein pairs account for 19.2% of the samples, whereas, C2 and C3-type account for much higher proportions - 49.2% and 31.6% respectively. Hence, in a typical cross-validation test set, the C1 class is more frequent than the population level, and performance estimates from cross-validation test sets cannot be expected to generalize reliably. Therefore, apart from the evaluation on the cross-validation and C1 test classes, results obtained on the C2 and C3 test classes should be taken into account before making any judgment about the generalization properties of the method.

3.2 Datasets

Dataset1: Park and Marcotte Datasets

We first evaluate our method on the human and yeast PPI datasets provided by Yungki Park and M. Marcotte [25]. The human PPI dataset contains 20,117 proteins and 24,718 PPIs whereas the yeast dataset contains 6,806 proteins and 14,938 PPIs. The experiments are conducted on the 40-fold train-test split for the CV, C1, C2, and C3 classes; for benchmarking, the split files were also as provided by [25].

Dataset2: Guo dataset

Guo et al. [16] provide PPI datasets for several species. These datasets are balanced and contain an equal number of positive and negative samples. In this paper, we use the yeast dataset used as a benchmark in many of the earlier studies [28], [32]. In this dataset, there are 2,450 unique proteins, and a total of 11,188 protein pairs, where half of these pairs are interacting (positive pair) and rest are non-interacting (negative pair). For negative pairs, the proteins that have no evidence of interactions are randomly selected, and are further filtered based on their sub-cellular location.

3.3 Evaluation Measurements

To compare the performance of our approach on Dataset1 with other methods we use the performance measures Area under Receiver Operating Characteristic curve (AUROC) and the area under the Precision-Recall curve (AUPRC). These measures were used by Park et al. [25] to benchmark the performance of previous studies. The ROC is a graph showing the relationship between the False positive rate (FPR) and the True positive rate (TPR). In contrast, the PRC is the graph showing precision and recall values calculated at different thresholds. The AUROC and AUPRC summarize the ROC and PR curves, respectively. The AUROC value indicates the ability of the classifier to distinguish between two classes, whereas AUPRC shows the trade-off achieved between precision and recall. The AUROC and AUPR are evaluated for all 40 (train-test) splits of Dataset1, and the mean value with standard deviation is reported.

The benchmark methods that use Dataset2 for performance evaluation use: accuracy, precision, sensitivity, specificity, F-score, and Matthew’s correlation coefficient (MCC) as evaluation metrics. To compare with benchmark methods, we also evaluate our proposed approach for Dataset2 on these metrics. These are defined as follows:

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{4a}
\]

\[
\text{Specificity} = \frac{TN}{TN + FP} \tag{4b}
\]

\[
FPR = 1 - \text{Specificity} \tag{4c}
\]

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4d}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{4e}
\]
3.4 Experiments and Results

3.4.1 PPI Prediction for Dataset1

In these experiments, we compare the results of the proposed framework with the benchmark methods (M1-M8) that rely on physico-chemical properties for encoding the protein sequences. Here, M1 and M2 are signature-product based methods proposed by Martin et al. [41]. M1 uses the signature-product directly, whereas M2 [42] uses metric learning on the signatures to compute the feature vector for the protein pair. M3 [26] is the Conjoint triad method; in this method, the feature vector of a protein sequence is based on the frequency of conjoint triads. M4 and M5 both use the AC method proposed by Guo et al. [16] for feature construction, but they differ in the classification algorithm, with M4 using the SVM whereas M5 employs a random forest. The M7 method was developed initially for protein-RNA interaction prediction and is based on maximizing the scores computed for interacting protein pairs vis-a-vis non-interacting protein pairs within the vector space. M8, proposed by Ding et al. [45], calculates mutual information-based features based on the functional categories of amino acids. For predictions, M1-M4 use the SVM while M5-M8 use a random forest classifier. All of these methods (M1-M8), except M6 (PIPE), are machine learning-based approaches. PIPE is a sequence similarity-based approach that relies on the shared local similarity between known interacting protein pairs and the test pair. In the comparison for the human dataset, we also report prediction results for another sequence similarity-based approach, SPRINT [46]. In contrast to the machine learning-based approaches, these methods use only positive (interacting) pairs for making predictions and hence do not require the negative (non-interacting) examples.

Our approach is a generic machine learning-based framework in which we can accommodate different representation learning methods for generating the feature vectors. We report the results for four different representation learning techniques - Seq2Vec (M9), BioVec (M10), SuperVecNW (M11) and SuperVecX (M12) - on both the human and yeast datasets.

Results

Tables 2 and 3 summarize the performance of each method on the human and yeast datasets respectively. We report the mean AUROC and AUPRC and their correspond-
TABLE 3
Comparison of the Prediction Performance Between the Proposed Methods and Other State of Art Methods on the Yeast Dataset From Dataset1

| Method                | AUROC | AUPRC |
|-----------------------|-------|-------|
|                       | CV    | C1    | C2    | C3    | CV    | C1    | C2    | C3    |
| M1 (Martin et al. [41]) | 0.83 ± 0.02 | 0.83 ± 0.01 | 0.62 ± 0.02 | 0.57 ± 0.03 | 0.83 ± 0.02 | 0.83 ± 0.01 | 0.62 ± 0.02 | 0.57 ± 0.03 |
| M2 (Vert et al. [42])  | 0.84 ± 0.01 | 0.84 ± 0.01 | 0.60 ± 0.02 | 0.59 ± 0.03 | 0.84 ± 0.01 | 0.84 ± 0.01 | 0.60 ± 0.02 | 0.58 ± 0.03 |
| M3 (Shen et al. [26])  | 0.65 ± 0.02 | 0.65 ± 0.02 | 0.56 ± 0.03 | 0.53 ± 0.07 | 0.65 ± 0.02 | 0.65 ± 0.02 | 0.56 ± 0.03 | 0.53 ± 0.07 |
| M4 (Guo et al. [16])   | 0.76 ± 0.02 | 0.76 ± 0.02 | 0.57 ± 0.02 | 0.54 ± 0.03 | 0.76 ± 0.02 | 0.76 ± 0.02 | 0.58 ± 0.02 | 0.54 ± 0.03 |
| M5 (Park et al. [25])  | 0.78 ± 0.02 | 0.78 ± 0.01 | 0.57 ± 0.02 | 0.54 ± 0.02 | 0.78 ± 0.02 | 0.78 ± 0.01 | 0.57 ± 0.02 | 0.54 ± 0.02 |
| M6 (Futre et al. [43]) | 0.75 ± 0.02 | 0.75 ± 0.02 | 0.59 ± 0.04 | 0.52 ± 0.04 | 0.75 ± 0.02 | 0.76 ± 0.02 | 0.60 ± 0.05 | 0.47 ± 0.07 |
| M7 (Bellucci et al. [44]) | 0.58 ± 0.02 | 0.58 ± 0.01 | 0.54 ± 0.02 | 0.52 ± 0.03 | 0.60 ± 0.02 | 0.60 ± 0.02 | 0.55 ± 0.02 | 0.53 ± 0.02 |
| M8 (Ding et al. [45])  | 0.81 ± 0.02 | 0.81 ± 0.02 | 0.62 ± 0.02 | 0.61 ± 0.02 | 0.84 ± 0.01 | 0.84 ± 0.01 | 0.64 ± 0.02 | 0.62 ± 0.02 |
| M9 (Seq2Vec)           | 0.81 ± 0.02 | 0.81 ± 0.01 | 0.59 ± 0.02 | 0.58 ± 0.02 | 0.83 ± 0.02 | 0.83 ± 0.01 | 0.61 ± 0.02 | 0.59 ± 0.02 |
| M10 (BioVec)           | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.61 ± 0.04 | 0.62 ± 0.02 | 0.82 ± 0.01 | 0.82 ± 0.01 | 0.62 ± 0.03 | 0.62 ± 0.02 |
| M11 (SuperVecNW)       | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.58 ± 0.01 | 0.54 ± 0.01 | 0.84 ± 0.02 | 0.82 ± 0.01 | 0.59 ± 0.01 | 0.53 ± 0.02 |
| M12 (SuperVecX)        | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.55 ± 0.01 | 0.54 ± 0.02 | 0.81 ± 0.01 | 0.81 ± 0.01 | 0.54 ± 0.01 | 0.54 ± 0.01 |
| PIPR                   | 0.83 ± 0.03 | 0.84 ± 0.01 | 0.60 ± 0.02 | 0.56 ± 0.03 | 0.83 ± 0.03 | 0.84 ± 0.02 | 0.60 ± 0.02 | 0.55 ± 0.02 |
| ProtBert               | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.67 ± 0.03 | 0.67 ± 0.04 | 0.87 ± 0.01 | 0.87 ± 0.01 | 0.69 ± 0.03 | 0.68 ± 0.03 |
| ProtAlbert             | 0.86 ± 0.01 | 0.84 ± 0.01 | 0.65 ± 0.03 | 0.65 ± 0.03 | 0.86 ± 0.01 | 0.86 ± 0.01 | 0.67 ± 0.03 | 0.66 ± 0.03 |
| ProtXLNet              | 0.85 ± 0.01 | 0.85 ± 0.01 | 0.66 ± 0.03 | 0.66 ± 0.03 | 0.87 ± 0.01 | 0.87 ± 0.01 | 0.68 ± 0.03 | 0.66 ± 0.03 |

The trend as AUROC values. Overall, SPRINT provides the best results for the C2 and C3 test class.

The results obtained for all methods on the yeast dataset are in general consistent with the results obtained for the human dataset, except for PIPE, which deteriorates for C2 and C3. SPRINT results are not available for the yeast dataset.

Comparing the performance of different Repl techniques i.e., Seq2Vec (M9), BioVec (M10), SuperVecNW (M11) and SuperVecX (M12) on the human dataset, we observe that for CV and C1, M9, M11 and M12 provide slightly better result than M10, whereas for C2 and C3 they provide comparable values. For the yeast data set, M9 and M10 provide comparable results. This difference in performance when compared to the human dataset might be due to the larger number of human protein interactions available for training as compared to the yeast dataset.

In summary, our proposed framework (tested with different Repl approaches) outperforms most of the standard methods for CV and C1 test splits, while giving comparable results for C2 and C3 in most cases. In contrast to other methods, the representations for protein pairs used in the proposed framework are simple to construct and provide computational advantages in training and prediction time due to their lower dimensionality.

3.4.2 PPI Prediction for Dataset2

In these experiments, we compare the results of our approach (tested with M9 and M10) on Dataset2 with baseline approaches SVM-AC [16], kNN-CTD [18], EELM-PCA [47], SVM-MCD [27], MLP [48], RF-LPQ [28], SAE [31], DNN- PPI [33], DPPI and SRGRU, SCNN and PIPR from [34]. The results for the baseline approaches are taken from [34].

Results

As shown in Table 4, M9 and M10 outperform many approaches (SVM-AC, kNN-CTD, EELM-PCA, SVM-MCD, SAE) including deep learning methods (DNN-PPI, SRGRU) and perform at par with DPPI and SCNN. The only method that outperforms our approaches is PIPR with an improvement of approx 2.5% in accuracy. Note that PIPR is a deep learning based end-to-end method whereas our methods comprise a simple feature extraction method with a binary classifier. Also, it is important to mention that these results are reported in the cross-validation setting and hence should not be considered to generalize for the complete dataset. It is appropriate to evaluate these further on the C2 and C3 test classes. Considering this, we evaluate PIPR on Marcotte’s benchmark dataset (Dataset 1) and procedure as discussed below.

3.5 Comparison With Deep Learning Approach - PIPR

Some recent work has demonstrated the applicability of deep learning based approaches for the PPI prediction task. All of these approaches are primarily based on the Convolutional Neural Network and have a deep architecture. The most recent of these approaches, PIPR [34], provides an end-to-end deep GRU (gated recurrent unit) based architecture for PPI prediction. The results reported [34] show an improvement in classification results (refer Table 4) compared to other benchmark methods – including other deep learning based approaches.

These results are computed following a 5-fold cross validation test on Guo’s yeast dataset (Dataset2). As discussed before, the cross-validation results may not reflect the ability of the method to generalize over the complete dataset. We thus evaluate the performance of PIPR on Marcotte’s dataset for C1, C2, C3 test classes along with the cross validation setting. The results for yeast dataset show improvement (2-3%) for CV and C1; for C2 the results are comparable but for C3 the figure is some 2-4% below our approaches. For the human dataset, its performance is generally 2-4% below our methods. These results show that like the other techniques PIPR does not perform well for C2 and C3 test sets. For the human dataset, it is not the best among the machine learning based methods.
TABLE 4
Evaluation of PPI Prediction on the Yeast Dataset (Guo’s) Based on 5 Fold Cross Validation

| Methods    | Accuracy (%) | Precision (%) | Sensitivity (%) | Specificity (%) | F1-score (%) | MCC (%)  |
|------------|--------------|---------------|-----------------|-----------------|--------------|----------|
| SVM-AC     | 87.35 ± 1.38 | 87.82 ± 4.84  | 87.30 ± 5.23    | 87.41 ± 6.33    | 87.34 ± 1.33 | 75.09 ± 2.51 |
| kNN-CTD    | 86.15 ± 1.17 | 90.24 ± 1.34  | 81.03 ± 1.74    | NA              | 85.39 ± 1.51 | NA       |
| EELM-PCA   | 86.99 ± 0.29 | 87.59 ± 0.62  | 86.15 ± 0.43    | NA              | 86.86 ± 0.37 | 77.36 ± 0.44 |
| SVM-MCD    | 91.38 ± 0.4  | 91.94 ± 0.69  | 90.67 ± 0.77    | NA              | 91.3 ± 0.73  | 84.21 ± 0.66 |
| MLP        | 94.43 ± 0.3  | 96.65 ± 0.59  | 92.06 ± 0.36    | NA              | 94.3 ± 0.45  | 88.97 ± 0.62 |
| RF-LPQ     | 93.92 ± 0.36 | 96.45 ± 0.45  | 91.10 ± 0.31    | NA              | 93.7 ± 0.37  | 85.66 ± 0.53 |
| SAE        | 67.17 ± 0.62 | 66.90 ± 1.42  | 68.06 ± 2.50    | 66.30 ± 2.27    | 67.44 ± 1.08 | 34.39 ± 1.25 |
| DNN-PPI    | 76.61 ± 0.51 | 75.1 ± 0.66   | 79.63 ± 1.34    | 73.59 ± 1.28    | 77.29 ± 0.66 | 53.32 ± 1.05 |
| DPPI       | 94.55        | 96.68         | 92.24           | NA              | 94.41        | NA       |
| SRGRU      | 93.77 ± 0.84 | 94.60 ± 0.64  | 92.85 ± 1.58    | 94.69 ± 0.81    | 93.71 ± 0.85 | 87.56 ± 1.67 |
| SCNN       | 95.03 ± 0.47 | 95.51 ± 0.77  | 94.12 ± 1.27    | 95.55 ± 0.77    | 95.00 ± 0.50 | 90.08 ± 0.93 |
| PipR       | 97.09 ± 0.24 | 97.00 ± 0.65  | 97.17 ± 0.44    | 97.00 ± 0.67    | 97.09 ± 0.23 | 94.17 ± 0.48 |
| M9 (Seq2Vec) | 94.24 ± 0.48 | 97.35 ± 0.55  | 90.97 ± 0.91    | 97.52 ± 0.53    | 94.05 ± 0.51 | 88.68 ± 0.93 |
| M10 (BioVec) | 94.18 ± 0.30 | 97.16 ± 0.67  | 91.03 ± 0.67    | 97.34 ± 0.65    | 93.99 ± 0.31 | 88.55 ± 0.61 |
| ProtBert   | 94.41 ± 0.34 | 97.35 ± 0.39  | 91.33 ± 0.57    | 97.50 ± 0.38    | 94.24 ± 0.36 | 89.89 ± 0.68 |
| ProtAlbert | 94.33 ± 0.33 | 97.26 ± 0.41  | 91.25 ± 0.66    | 97.41 ± 0.4     | 94.16 ± 0.35 | 88.83 ± 0.64 |
| ProtXLNet  | 94.79 ± 0.42 | 97.59 ± 0.53  | 91.88 ± 0.61    | 97.72 ± 0.51    | 94.65 ± 0.44 | 89.75 ± 0.84 |

Mean and standard deviation is reported in the table.

3.6 Comparison With Recent Deep Language Models

There has been an increase in focus on adapting deep language models for generating embeddings for molecular sequences over the last two years. Some of the most popular models in this category include SeqVec [49] - an adaptation of LSTM based language model, ELMo [50], ProtBert, ProtAlbert, ProtXLNet by Elnaggar et al. [51] which are adaptation of large Transformer models BERT [52], Albert [53] and XLNet [54], respectively. In contrast to the Word2Vec based approaches, these newer models are more complex and have significantly greater computational and memory requirements for training and inference purposes.

We include three of the latest models - ProtBert, ProtAlbert and ProtXLNet for PPI prediction experiments. For the human dataset (Dataset1), ProtBert provides comparable results (refer Table 2) to the Word2Vec based models (Seq2Vec, BioVec etc), whereas ProtAlbert and ProtXLNet provide significant improvement, specifically for C3 test set (7-10%). For the yeast dataset (Dataset1) all three models provide improved results over all test cases - C1, C2 and C3. For the yeast dataset (Dataset2), the performance of these models is comparable to the Word2Vec based representation learning models.

When comparing these newer models with the Word2Vec based models in the context of PPI prediction, there are a few notable distinctions. First, in contrast to the Seq2Vec and BioVec generated sequence embeddings (of size 100), these models generate high dimensional embeddings. ProtBert, ProtAlbert, and ProtXLNet generate sequence embeddings of sizes 1,024, 4,096 and 1,024; the protein pairs are thus represented with 2,048, 8,192 and 2,048-dimensional feature vectors. Large input size increase training and prediction time for an ML model.

Second, the performance of these models for PPI prediction, specifically for the C3 test set, shows their ability to capture distinctive features for representing sequences. However, the comparable results in Table 4 show that for some cases, we might do just as well by using Word2Vec based models.

Third, these models are large - ProtBert has approximately 420 million parameters to learn; it is trained for millions of sequences for days over 5,161 GPUs [51]. In contrast Word2Vec based models can be run on a standard desktop with minimal memory requirement. Also, the inference time of the Word2Vec based models is significantly lower than these models. The improved performance achieved from these newer models in lieu of the computational cost makes it interesting to optimise these models specifically in the bioinformatics context to reduce their computational requirements. Also, since the Word2Vec based models considered here are trained on significantly smaller amounts of data, it would be interesting to note the possible effect on the quality of sequence embeddings generated through such models when trained on a much larger corpus.

3.7 PPI Network Prediction

A good PPI prediction method is expected to perform well in predicting the interactions of known networks. Previously, Shen et al. [26] and Ding et al. [45] demonstrated the utility of their approaches in predicting the interactions of three different types of networks, namely, one-core, multi-core, and crossover network. The one-core network is a simple network that consists of a core protein that is connected to the other proteins radially. In a multiple-core network, there are core proteins that interact with other proteins. The crossover network is a combination of one-core and multi-core networks.

Here, we also evaluate our approaches on such networks and use the CD9 (a tetraspanin protein) network for one-core, the Ras-Raf1-Mek1-Erk1-Elk1-5r pathway for multi-core, and Wnt-related network for the crossover network types. The CD9 network contains 16 PPIs, the Ras-Raf1-Mek1-Erk1-Elk1-5r pathway network has 189, and the Wnt-related network contains 96 PPIs. For experiments, we follow the same process as described in Fig. 3.

For training, we use human PPI data provided by Guo et al. [16] and ensure that the core protein (CD9, Ras, Raf1, Mek1, Erk1, Elk1, SRF) and proteins from the Wnt-related
TABLE 5
Comparison of the Prediction Performance of Different Methods on PPI Networks

| Methods  | Networks       | CD9  | Ras pathway | Wnt-related |
|----------|----------------|------|-------------|-------------|
| Shen et al. [26] | M9 (Seq2Vec) | 13   | 161         | 73          |
| Ding et al. [45] | M9 (BioVec)  | 14   | 174         | 91          |
|           | M10            | 16   | 185         | 94          |
|           | M10 (BioVec)  | 16   | 183         | 87          |

The entries in each column gives the count of correctly predicted interaction for one of the network. The total number of interactions in CD9, Ras pathway and Wnt-related networks is 16, 189 and 96 respectively.

network are not present. We compare the results with established sequence-based approaches and show the superior performance of our approaches (tested on M9 and M10); the results are provided in Table 5. Among M9 and M10, M9 provides better results.

3.8 Discussion
Predicting PPIs is a challenging problem. In recent years, computational methods have been a valuable alternative to experimental methods owing to their speed and relatively low false-positive rates. Over this period, sequence-based (machine learning) techniques have evolved, and better performance – as much as an 18% improvement over the earliest methods – has been achieved. Sequence-based methods mainly differ in the way the feature vector is constructed. Most of the available methods rely on known physicochemical properties of the protein. Such properties might not cover the interaction information completely and hence may be limited in their value in distinguishing interacting from non-interacting pairs. The availability of sequence data and improvements in automatic extraction of features in text processing have motivated researchers to learn feature vectors directly from the sequences.

This work establishes the fact that representation learning (RepL) based approaches can produce sequence embeddings that are useful for PPI prediction, in many cases providing better results than the physico-chemical property feature vectors without the need for extensive domain knowledge. Another advantage of using learned representations is their low dimensionality, which speeds up the training process and the inference step.

While sequence-based (machine learning) methods have evolved and have been shown to provide a high level of accuracy for PPI prediction, we need to be careful when making a judgment about their ability to generalize. Most of these methods give high cross-validation accuracy, which may not be a reflection on their generalization properties [25]. Therefore, we need to also focus on the results of the C2 and C3 test classes. In our experiments, we found that all methods – including the recent deep learning method PIPR – show good cross-validation accuracy, but when evaluated for the C2 and C3 test class, their performance markedly deteriorates.

Some methods which rely directly on similarity – such as PIPE and SPRINT – show improved results for the C2 and C3 test classes compared to other methods. Results for SPRINT on the Yeast datasets were not available. In contrast to machine learning approaches, PIPE and SPRINT work at a much more granular level, and use sub-sequence comparisons as the cornerstone for making predictions. These methods use the computationally intensive preprocessing step [46] of computing similar sub-sequences from the complete protein set. In contrast, machine learning approaches rely on feature vectors that capture the overall sequence information. Working at such a level of granularity proves to be advantageous for PIPE and SPRINT on the C2 and C3 test classes of the human dataset. Using embedding based approaches that work at different resolutions can potentially yield improvements for the C2 and C3 test cases.

All of the methods considered performed relatively poorly on the C3 test class. To understand this better (specifically for representation learning based methods) we undertook some exploratory investigations. These investigations were based on the interactions between proteins in the neighborhood of each protein in a potentially interacting pair. Here we focused on proteins from the CD9 network and made predictions for all possible protein pairs. There are 17 proteins in the CD9 network, which makes a total of 136 possible protein pairs. If we remove all of these 17 proteins from the training set, this set constitutes a C3 test class.

To label a protein pair as interacting or non-interacting, we use the well known biological database STRING [55], which scores the known protein-protein interactions by taking into consideration various evidence sources (experimental evidence, co-expression, mention in literature etc.). If a protein pair is present in the STRING database (> 0.4 score), we label it as interacting; otherwise it is labelled as non-interacting. The results we obtain with the embeddings in M9 and M10 show that these methods can detect most of the interactions but suffer from an high false-positive rate. We note that the protein pairs that are not present in the STRING database may be the negative or they may include some interacting pairs which might not be documented. By examining the possible links between proteins in the neighborhood of each member of the pair, we sought to obtain some insight into the over-generalisation of many of these methods.

We selected candidates based on: (i) the distance of proteins from the test sample in the proteome and (ii) the interaction/non-interaction of the neighbours of the test sample proteins in the embedding space - constructed of the embeddings of unique proteins present in the training set. We found sufficient evidence to question perhaps 10-20% of the false positives, but the approach was inconclusive with its main value coming as a potential strategy to improve representations. At present, the efficacy of the proposed approach is limited by the embeddings used to construct a feature vector for a protein pair.

Capturing the interaction information from a pair of proteins in a feature vector remains a challenging task. While our methods offer much more convenient construction of feature vectors than some existing alternatives, and performance remains comparable, some insight is lacking and there is a good deal of scope to improve our models for generating feature vectors from raw sequences.

One other possible opportunity lies in focusing on hybrid models that take advantage of available high-quality
annotations and measurement data along with the sequence data. Such methods may be useful in dealing with the C2 and C3 test class samples. Also, we note that none of the sequence-based methods considers the PPI prediction in a specific context (e.g., different cell types). Having models that can utilise both measurement and sequence data can help us to make a better prediction of the PPIs in a specific context. Given that our supervised models (SuperVecNW and SuperVecX) for feature construction are flexible, it might be possible to extend these approaches for context-specific PPI predictions while retaining the low dimensional representation.

4 Conclusion

In this paper, we proposed a framework that uses embeddings learned through Word2Vec based models for the PPI prediction task. Such embeddings require little prior biological knowledge and are learned directly from the sequence data. Moreover, these embeddings offer a far lower dimensional representation than those based on the physico-chemical properties of the proteins and their constituent amino acids.

The experimental results obtained on both human and yeast datasets, and on selected networks confirm that better sequence embeddings for PPI prediction tasks can be generated without necessarily relying on biological domain knowledge. We observe that most of the sequence-based methods perform well when evaluated on CV and C1 but not on the C2 and C3 test classes. The inability of sequence-based methods to perform better for C2 and C3 test classes presents an attractive if challenging opportunity to learn feature vectors that better represent the interaction between two proteins. Perhaps some limited use of prior knowledge in a supervised learning framework or working on a smaller resolution similar to SPRINT may provide the answer – maintaining most of the convenience of our approach while offering improved performance on the more challenging datasets.

As discussed in Section 3.6, we have also seen explorations recently with other types of representation learning models for the creation of embeddings for molecular sequences. These newer models require a massive amount of data and substantial computational resources for training. The landscape now includes a range of alternatives for the generation of these deep-network embeddings. In effect, the choice comes down to a trade-off between accuracy, computational cost and the need for training data. The shallow representation learning models (e.g., BioVec, Seq2Vec) have significant advantages in terms of training data required, the time taken for training, and the time needed for generating embeddings for new sequences. It would be an interesting future direction to explore the possible effect on the quality of sequence embeddings generated by Word2Vec based models in the context of their ability to predict PPIs when trained on a much larger corpus - a data set of the scale used to train some of these newer models.

References

[1] J. De Las Rivas and C. Fontanillo, “Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks,” PLoS Comput. Biol., vol. 6, no. 6, 2010, Art. no. e1000807.
[2] R.-S. Wang, Y. Wang, L.-Y. Wu, X.-S. Zhang, and L. Chen, “Analysis on multi-domain cooperation for predicting protein–protein interactions,” BMC Bioinf., vol. 8, no. 1, 2007, Art. no. 391.
[3] S. Fields and O.-K. Song, “A novel genetic system to detect protein–protein interactions,” Nature, vol. 340, no. 6230, 1989, Art. no. 245.
[4] A.-C. Gavin et al., “Functional organization of the yeast proteome by systematic analysis of protein complexes,” Nature, vol. 415, no. 6868, pp. 141–147, 2002.
[5] U. Rüetschi et al., “Proteomic analysis using protein chips to detect biomarkers in cervical and amniotic fluid in women with intra-amniotic inflammation,” J. Proteome Res., vol. 4, no. 6, pp. 2236–2242, 2005.
[6] Y. Ho et al., “Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry,” Nature, vol. 415, no. 6868, pp. 180–183, 2002.
[7] Z.-H. You, K. C. Chan, and P. Hu, “Predicting protein–protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest,” PLoS One, vol. 10, no. 5, 2015, Art. no. e0125811.
[8] D.-S. Han, H.-S. Kim, W.-T. Lee, and J.-K. Suh, “PreSPI: A domain combination based prediction system for protein–protein interaction,” Nucleic Acids Res., vol. 32, no. 21, pp. 6312–6320, 2004.
[9] K. K. Wan, J. Park, and J. K. Suh, “Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair,” Genome Inform., vol. 13, pp. 42–50, 2002.
[10] E. Sprinzak and H. Margalit, “Correlated sequence-signatures as monitors of protein-protein interaction,” J. Mol. Biol., vol. 311, no. 4, pp. 681–692, 2001.
[11] U. Ogmen, O. Keskin, A. S. Aytown, R. Nussinov, and A. Gursoy, “PRISM: Protein interactions by structural matching,” Nucleic Acids Res., vol. 33, no. suppl_2, pp. W331–W336, 2005.
[12] P. Aloy and R. B. Russell, “InterPreTS: Protein interaction prediction through tertiary structure,” Bioinformatics, vol. 19, no. 1, pp. 161–162, 2003.
[13] P. Aloy and R. B. Russell, “Interrogating protein interaction networks through structural biology,” Proc. Nat. Acad. Sci. USA, vol. 99, no. 9, pp. 5896–5901, 2002.
[14] S. R. Maetschke, M. Simonens, M. J. Davis, and M. A. Ragan, “Gene ontology-driven inference of protein–protein interactions using inducers,” Bioinformatics, vol. 28, no. 1, pp. 69–75, 2012.
[15] S. Bandypopadhyay and K. Mallick, “A new feature vector based on gene ontology terms for protein-protein interaction prediction,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 14, no. 4, pp. 762–770, Jul./Aug. 2017.
[16] Y. Guo, L. Yu, Z. Wen, and M. Li, “Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences,” Nucleic Acids Res., vol. 36, no. 9, pp. 3025–3030, 2008.
[17] Y. Z. Zhou, Y. Gao, and Y. Y. Zheng, “Prediction of protein–protein interactions using local description of amino acid sequence,” in Proc. Advances Comput. Sci. Educ. Appl., 2011, pp. 254–262.
[18] L. Yang, J.-F. Xia, and J. Gui, “Prediction of protein–protein interactions from protein sequence using local descriptors,” Protein Peptide Lett., vol. 17, no. 9, pp. 1085–1090, 2010.
[19] E. Asgari and M. R. K. Mofrad, “Continuous distributed representation of biological sequences for deep proteomics and genomics,” PLoS One, vol. 10, no. 11, Nov. 2015, Art. no. e0141287. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0141287.
[20] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, “Distributed representations for biological sequence analysis,” 2016, arXiv:1608.05949.
[21] P. Ng, “DNA2vec: Consistent vector representations of variable-length k-mers,” 2017, arXiv:1701.06279.
[22] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, “Learned protein embeddings for machine learning,” Bioinformatics, vol. 1, 2018, Art. no. 7.
[23] A. S. Schwartz et al., “Deep semantic protein representation for annotation, discovery, and engineering,” BioRxiv, 2018, Art. no. 365965.
[24] E. Asgari, A. C. McHardy, and M. R. Mofrad, “Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX),” Sci. Rep., vol. 9, no. 1, 2019, Art. no. 3577.
[25] Y. Park and E. M. Marcotte, “Flaws in evaluation schemes for pair-input computational predictions,” Nature Methods, vol. 9, no. 12, 2012, Art. no. 1134.
et al., “Predicting protein–protein interactions based only on sequences information,” Proc. Nat. Acad. Sci. USA, vol. 104, no. 11, pp. 4337–4341, 2007.

Z.-H. You, L. Zhu, C.-H. Zheng, H.-J. Yu, S.-P. Deng, and Z. Ji, “Prediction of protein–protein interactions from amino acid sequences using a novel multi-scale Art. no. 44.

L. Wong, Z.-H. You, S. Li, Y.-A. Huang, and G. Liu, “Detection of protein–protein interactions from amino acid sequences using a rotation forest model with a novel PR-LPQ descriptor,” in Proc. Int. Conf. Infell. Comput., 2015, pp. 713–720.

Y.-A. Huang, Z.-H. You, X. Gao, L. Wong, and L. Wang, “Using weighted sparse representation model combined with discrete cosine transformation to predict protein–protein interactions from protein sequence,” BioMed Res. Int., vol. 2015, pp. 1–10, 2015.

S. Henikoff and J. G. Henikoff, “Amino acid substitution matrices from protein blocks,” Proc. Nat. Acad. Sci. USA, vol. 89, no. 22, pp. 10 915–10 919, 1992.

T. Sun, B. Zhou, L. Lai, and J. Pei, “Sequence-based prediction of protein interaction using a deep-learning algorithm,” BMC Bioinf., vol. 18, no. 1, Art. no. 485.

S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, “Predicting protein–protein interactions through sequence-based deep learning,” Bioinformatics, vol. 34, no. 17, pp. i802–i810, 2018.

H. Li, X.-J. Gong, H. Yu, and C. Zhou, “Deep neural network based predictions of protein interactions using primary sequences,” Molecules, vol. 23, no. 8, 2018, Art. no. 1923.

J.-P. Vert et al., “Multifaceted protein–protein interaction prediction based on siamese residual RCNN,” Bioinformatics, vol. 35, no. 14, pp. i305–i314, 2019.

T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient estimation of word representations in vector space,” in Proc. Int. Conf. Learn. Representations, 2013, pp. 1–12. [Online]. Available: http://arxiv.org/pdf/1301.3781v3.pdf

D. Kimothi, P. Biyani, J. M. Hogan, A. Soni, and W. Kelly, “Learning supervised embeddings for large scale sequence comparisons,” PLoS one, Public Library of Science: San Francisco, CA, USA, vol. 15, no. 3, p. e0216636, 2019.

L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

J. Wu, T. Vallenius, K. Ovaska, J. Westermarck, G. Corrado, and K. Chen, “Efficient estimation of word representations in vector space,” in Proc. Int. Conf. Learn. Representations, 2013, pp. 1–12. [Online]. Available: http://arxiv.org/pdf/1301.3781v3.pdf

D. Kimothi, P. Biyani, J. M. Hogan, A. Soni, and W. Kelly, “Learning supervised embeddings for large scale sequence comparisons,” PLoS one, Public Library of Science: San Francisco, CA, USA, vol. 15, no. 3, p. e0216636, 2019.

L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

J. Wu, T. Vallenius, K. Ovaska, J. Westermarck, T. P. Mäkelä, and S. Hautaniemi, “Integrated network analysis platform for protein interaction networks,” Nature Methods, vol. 6, no. 1, 2009, Art. no. 75.

I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg, “DIP, The database of interacting proteins: A research tool for studying cellular networks of protein interactions,” Nucleic Acids Res., vol. 30, no. 1, pp. 303–305, 2002.

M. H. Schaeffer, J.-F. Fontaine, A. Vinayagam, P. Porras, E. Wanker, and M. A. Andrade-Navarro, “HIPPIE: Integrating protein interaction networks with experimental-based quality scores,” PLoS One, vol. 7, no. 2, 2012, Art. no. e31826.

S. Martin, D. Roe, and J.-L. Faulon, “Predicting protein–protein interactions using signature products,” Bioinformatics, vol. 21, no. 2, pp. 218–226, 2004.

J.-P. Vert, J. Qi, and W. S. Noble, “A new pairwise kernel for biological network inference with support vector machines,” BMC Bioinf., vol. 8, no. 10, 2007, Art. no. 58.

S. Pitre et al., “Global investigation of protein–protein interactions in yeast saccharomyces cerevisiae using re-occurring short polypeptide sequences,” Nucleic Acids Res., vol. 36, no. 13, pp. 4286–4249, 2008.

M. Bellucci, F. Agostini, M. Masin, and G. G. Tartaglia, “Predicting protein associations with long noncoding RNAs,” Nature Methods, vol. 8, no. 6, 2011, Art. no. 444.

Y. Ding, J. Tang, and F. Guo, “Predicting protein–protein interactions via multivariate mutual information of protein sequences,” BMC Bioinf., vol. 17, no. 16, 2016, Art. no. 398.

Y. Li and L. Ilie, “SPRINT: Ultrafast protein–protein interaction prediction of the entire human interactome,” BMC Bioinf., vol. 18, no. 1, 2017, Art. no. 485.

Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, “Prediction of protein–protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis,” BMC Bioinf., vol. 14, no. 8, 2013, Art. no. S10.

X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, “DeepPPI: Boosting prediction of protein–protein interactions with deep neural networks,” J. Chem. Inf. Model., vol. 57, no. 6, pp. 1499–1510, 2017.

M. Heinzenber, Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, “Prediction of protein–protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis,” BMC Bioinf., vol. 14, no. 8, 2013, Art. no. S10.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.