Bioinformatics Advances, 2022, 1–8
https://doi.org/10.1093/bioadv/vbac059
Advance Access Publication Date: 17 August 2022
Original Paper

Graph Neural Networks

ProteinPrompt: a webserver for predicting protein–protein interactions

Sebastian Canzler [1,2,*,†], Markus Fischer [3,†], David Ulbricht[3], Nikola Ristic [3], Peter W. Hildebrand [3,4,5] and René Staritzbichler [1,3,*]

[1]Immuthera GmbH, 04275 Leipzig, Germany, [2]Department Computational Biology, Helmholtz Centre for Environmental Research—UFZ, 04318 Leipzig, Germany, [3]Institute of Medical Physics and Biophysics, University of Leipzig, 04107 Leipzig, Germany, [4]Charité—Universitätsmedizin Berlin, Corporate Member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Institute of Medical Physics and Biophysics, 10117 Berlin, Germany and [5]Berlin Institute of Health at Charité—Universitätsmedizin Berlin, 10117 Berlin, Germany

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Michael Gromiha

Received on December 21, 2021; revised on July 19, 2022; editorial decision on August 3, 2022; accepted on August 14, 2022

Abstract

Motivation: Protein–protein interactions (PPIs) play an essential role in a great variety of cellular processes and are therefore of significant interest for the design of new therapeutic compounds as well as the identification of side effects due to unexpected binding. Here, we present ProteinPrompt, a webserver that uses machine learning algorithms to calculate specific, currently unknown PPIs. Our tool is designed to quickly and reliably predict contact propensities based on an input sequence in order to scan large sequence libraries for potential binding partners, with the goal to accelerate and assure the quality of the laborious process of drug target identification.

Results: We collected and thoroughly filtered a comprehensive database of known binders from several sources, which is available as download. ProteinPrompt provides two complementary search methods of similar accuracy for comparison and consensus building. The default method is a random forest (RF) algorithm that uses the autocorrelations of seven amino acid scales. Alternatively, a graph neural network (GNN) implementation can be selected. Additionally, a consensus prediction is available. For each query sequence, potential binding partners are identified from a protein sequence database. The proteom of several organisms are available and can be searched for binders. To evaluate the predictive power of the algorithms, we prepared a test dataset that was rigorously filtered for redundancy. No sequence pairs similar to the ones used for training were included in this dataset. With this challenging dataset, the RF method achieved an accuracy rate of 0.88 and an area under the curve of 0.95. The GNN achieved an accuracy rate of 0.86 using the same dataset. Since the underlying learning approaches are unrelated, comparing the results of RF and GNNs reduces the likelihood of errors. The consensus reached an accuracy of 0.89.

Availability and implementation: ProteinPrompt is available online at: http://proteinformatics.org/ProteinPrompt, where training and test data used to optimize the methods are also available. The server makes it possible to scan the human proteome for potential binding partners of an input sequence within minutes. For local offline usage, we furthermore created a ProteinPrompt Docker image which allows for batch submission: https://gitlab.hzdr.de/proteinprompt/ProteinPrompt. In conclusion, we offer a fast, accurate, easy-to-use online service for predicting binding partners from an input sequence.

Contact: rene.staritzbichler@medizin.uni-leipzig.de or sebastian.canzler@ufz.de

1 Introduction

Protein interactions are key to the complex molecular interplays of cellular processes. The driving forces of these molecular networks are protein interactions rather than the individual functions of single protein components (Pawson, 2004). Biological processes, such as cellular organization, communication, immune responses and the regulation of transcription and translation, function appropriately only when various proteins interact and work together properly.

Laboratory identification and validation of protein interactions often relies on expensive, time-consuming biochemical and biophysical assays, including ELISA, western blot, immunoprecipitation, Förster resonance energy transfer or cross-linking approaches, florescence anisotropy, microscale thermophoresis, surface plasmon resonance spectroscopy, high-throughput screening methods (e.g. phage display) or combinations thereof.

A reliable *in silico* method for predicting protein–protein interactions (PPIs) would therefore shed more light on the details of biological pathways and pharmacological responses. Computations may complement and guide biochemical assays. However, explicit molecular dynamics or docking approaches require structural detail, which is often unavailable. Even if the structure of a protein is known, these methods are computationally expensive and therefore impractical for scanning huge libraries of candidates. Therefore, when structural insight is lacking or when speed is crucial, other methods are needed. Non-structure-based computational approaches for identifying potential PPIs generally use an extensive dataset of known PPIs, combined with information about cellular localization, amino acid sequences or secondary structures.

These methods may include phylogenetic trees (Pazos and Valencia, 2001), phylogenetic profiles (Barker and Pagel, 2005; Hamp and Rost, 2015), graph-based approaches (Yang *et al.*, 2020), support vector machines (Li *et al.*, 2019) or network-based approaches (Clauset *et al.*, 2008; Yook *et al.*, 2004), stacked autoencoders (Sun *et al.*, 2017) as well as (recurrent) convolutional neural networks (CNNs) (Chen *et al.*, 2019; Hashemifar *et al.*, 2018). In recent years, distinct prediction methods have been combined, e.g. CNNs and feature-selecting rotation forests (Wang *et al.*, 2019). Algorithms from language encoding (Yao *et al.*, 2019) and principle component analysis (Kong *et al.*, 2020) were used to derive feature vectors. Structural features were exploited (Das and Chakrabarti, 2021; Singh *et al.*, 2010). Overall, the field of biology has recently seen a massive increase in applications using deep learning (Ching *et al.*, 2018). Nevertheless, different proteome-wide prediction methods have demonstrated that knowledge of the amino acid sequence alone may be sufficient to identify novel, functional PPIs (Martin *et al.*, 2005; Shen *et al.*, 2007). These methods usually rely on statistical learning algorithms. Due to its significant advantages, which include simplicity, rapidity and generality, this method of prediction has become more and more common in recent years (Betel *et al.*, 2007; Chen *et al.*, 2020; Das and Chakrabarti, 2021; Liu *et al.*, 2012; Ofran and Rost, 2003; Pan *et al.*, 2010; Perovic *et al.*, 2017). Precalculated databases are available online: `PrePPI` (Zhang *et al.*, 2013), `ProfPPI` (Tran *et al.*, 2018), `STRING` (Szklarczyk *et al.*, 2011) or `PIPS` (McDowall *et al.*, 2009). Several webservices offer a limited number of pairwise predictions: `PSOPIA` (Murakami and Mizuguchi, 2014), `iLoops` (Planas-Iglesias *et al.*, 2013) and `iFrag` (Garcia-Garcia *et al.*, 2017). To our knowledge, no webserver currently offers scanning entire proteoms.

In this article, we present an online sequence-based approach to predicting PPIs. The tool's predictive power was boosted through rigorous fine-tuning of the key elements of machine learning, including dataset generation and feature vector design. Auto-correlation (AC) of hydrophobicities combined with a random forest (RF) machine-learning algorithm led to maximum accuracy. The quality and speed of this system make it a suitable high-throughput method for scanning sequence libraries. Furthermore, we achieved a comparable accuracy rate using a graph neural network (GNN). Since this approach has a completely different mathematical structure, we provide it as an option for the user on the server. Additionally, a consensus method is available.

Therefore, `ProteinPrompt` (**protein pr**ediction **o**f **m**atching **p**ar**t**ners) may serve as a reliable tool for identifying potential interaction partners from an entire proteom. It can thus be used to help identify the yet-unknown biological roles of many proteins and may contribute to identifying new therapeutic targets.

2 Materials and methods

To maximize the system's predictive power, it is pivotal to optimize all key elements, including the collection of training and testing data, the calculation of feature vectors and the selection and fine-tuning of the machine learning algorithm. Many varied approaches to all these steps were tested. Here, we focus on the approaches that resulted in the highest accuracy rates.

2.1 Collecting data points

In order to create comprehensive training and testing data, we tried to collect as many trustworthy PPI annotations as possible. We included data from various sources, such as the `Database of Human Interacting Proteins` (http://dip.doe-mbi.ucla.edu/) (DIP) (Salwinski *et al.*, 2004), the `Human Protein Reference Database` (http://www.hprd.org/) (HPRD) (Keshava Prasad *et al.*, 2009), the `Protein Database` (www.rcsb.org) (PDB) (Berman *et al.*, 2000) and the `Negatome Database` (http://mips.helmholtz-muenchen.de/proj/ppi/negatome/) (Blohm *et al.*, 2014). We also included annotations retrieved from the `KUPS` (http://www.ittc.ku.edu/chenlab/kups/) server (Chen *et al.*, 2011), which mainly incorporates PPIs from `MINT` (https://mint.bio.uniroma2.it/) (Licata *et al.*, 2012) and `IntAct` (https://www.ebi.ac.uk/intact/) (Orchard *et al.*, 2014). In addition to the negative annotations collected from the `Negatome Database`, the `KUPS` server generates negative data points based on the following criteria: (i) the proteins are functionally dissimilar, (ii) the proteins are located in different cellular compartments and (iii) the proteins are part of non-interacting domains.

After intense manual curation and after mapping the different names used to describe the same proteins, we derived a total list of 31 867 non-identical human proteins with a distinct UniProt identifier. For this set, 73 681 positive PPIs were collected from the databases mentioned above. We then used `CD-hit` (Fu *et al.*, 2012; Li and Godzik, 2006) on concatenated sequence pairs to reduce this set to 41 482 positive protein–protein pairs with at most 50% sequence identity. For the negative pairs, we collected over 1.5 million unique protein–protein pairs, from which we randomly selected a comparable number of PPIs to the size of the positive dataset, while still maintaining at most 50% sequence identity. In Supplementary Section S2.1.1, we investigated the effects of using a 40% cutoff in `CD-hit` and found that this reduction had only a negligible influence on the diversity of the feature vectors.

We separated the data into test and training data to estimate the quality of the optimized prediction model with an independent dataset, which was not involved in the training or similar to the data used for training. Our final training dataset contained 36 423 positive and 34 640 negative PPIs, while our testing dataset contained 5059 positive and 4817 negative data points.

Figure 2 illustrates the collection of data and its general usage. A detailed description of the number of individual proteins in each dataset can be found in Supplementary Section S2.1.

To assess the prediction quality of `ProteinPrompt`, we compared our system to several other publicly available programs. As some of them have limited speed or upload capacity, it was not feasible to perform this test using our entire test dataset, which includes more than 10k data points. Instead, we randomly selected 470 positive and 470 negative pairs from our test set.

2.2 Analysis of datasets

In machine learning, data are often considered equal. Under this assumption, data can be divided into training and test data using random selection. This is clearly not the case for protein sequences, as the level of similarity among sequences may vary dramatically. (A comparison may be drawn from face recognition: Protein sequences can differ from one another as much as human faces differ from those of cats or whales.) Park and Marcotte (2012) point out several combinatorial issues, which were further explored by Hamp and Rost (2015). These issues are due to the pairwise nature of the input and should be considered when building the test dataset. It should be noted that our input data are sequence pairs that are identical in nature and therefore symmetric. This section addresses the relation between data points and their separation into test and training sets.

Obviously, it is significantly easier to predict binding partners for a query sequence that is very similar to one of the sequences in the training dataset than to predict binding partners for a sequence that has no similarity to any sequence in the training set. The latter case requires the algorithm to have 'understood' some of the principles that control the binding, while the former case only requires it to interpolate from known cases. The more predictions are based on actual understanding, the more general the results will be. However, it is very difficult to understand the complex, 3D interactions of macromolecules based on patterns in their sequences. A full understanding of these interactions requires some level of understanding of the folding of the individual proteins as well as their preferences regarding relative orientation.

Our goal was to train a method that was as general as possible, in the sense that it should not specialize in a certain class of human protein sequences. This meant that we needed to minimize redundancies in the datasets. Furthermore, we intended to perform a very rigorous test by reducing similarities between the training and testing data. As pointed out by Park and Marcotte (2012), this approach may not lead to the best overall performance. However, it is the most rigorous way to test such a system.

Therefore, we analyzed the redundancies in our datasets by comparing similarities among data points. It should be noted that we were not looking for pairwise sequence similarity but the similarity of any given pair of sequences with any other pair of sequences. All datasets contain sequence pairs that are known to be binders or non-binders. First, we collected BLAST alignments for all the sequences in all datasets. We calculated similarity by dividing the number of identical positions by the length of the sequence. To compare the similarity of a pair A: (A_1, A_2) with pair B: (B_1, B_2) similarities were calculated in two possible combinations: Sim(A_1, B_1) with Sim(A_2, B_2) and Sim(A_1, B_2) with Sim(A_2, B_1). The combination resulting in a higher total similarity was selected. The 2D histogram in Figure 1 illustrates the low similarity between the binders in the training and test datasets. An analysis in Supplementary Section S2.2 illustrates how small the influence of the remaining similarities is. Thus, the selected test set represents a difficult challenge for the algorithm, and the accuracy rates reported here can be considered a worst-case estimate.

Equivalent analyses with very similar results were performed on both datasets separately; the results are provided in Supplementary Figs S1, S4 and S5. The values that are illustrated in the figures are provided in Supplementary Tables S1, S3 and S4.

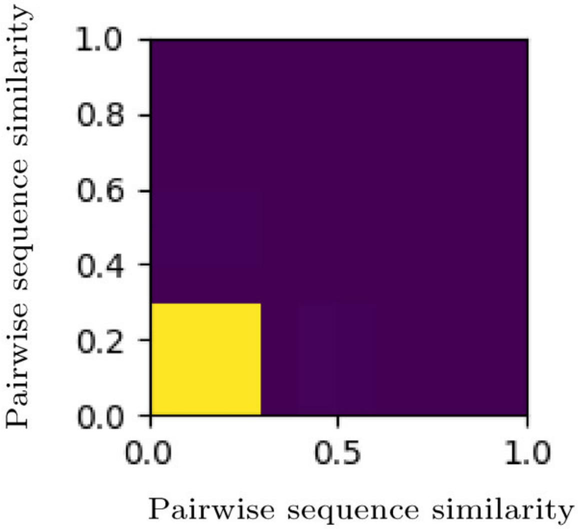

Fig. 1. Redundancy analysis of binders in the training and test datasets. Depicted is a 2D histogram of similarity between sequence pairs known to be binders in the training and test datasets. The bright square on the bottom left represents sequence pairs with no or very few similarities (below ~30%). It has a count of $1.8 \cdot 10^8$. The remaining dark area represents significantly smaller number of occurrences. A detailed description of this analysis is provided in section. The values of the histogram can be found in Supplementary Table S3. Note that this figure is identical to the left image in Supplementary Figure S4

2.3 Selection of learning method

Many machine learning algorithms are available. We used the `caret` R package (Kuhn, 2008) for fine-tuning and comparison. We tested several artificial neural network implementations, support vector machines and tree-based methods. After determining that the RF approach performed best on our extensive testing and training datasets, we moved to a Python-based implementation to improve performance. A significant advantage of RFs is that overfitting is not a major problem.

We also extensively tested several more recent machine-learning methods, such as CNNs and GNNs, which were implemented using the Tensorflow and PyTorch frameworks, respectively. CNNs were dismissed because of significant problems with overfitting. However, the GNN method resulted in comparable prediction quality to that achieved by the RF. Furthermore, the GNN method does not require any external preprocessing and can handle input data of varied sizes and structures. In our implementation, the GNN translates the data into fixed-sized feature vectors, followed by a multilayer perceptron (MLP) that is predicting the binding. Feature vector calculation and predictions are trained in a single optimization scheme. Therefore, the GNN-based implementation was trained on the raw data profiles.

2.4 Random forest classification

RF is a supervised learning algorithm that uses an ensemble of classification trees (Breiman, 2001). Each classification tree is built through bootstrap aggregation (bagging), a method of random sampling with replacement. From the original dataset, which contains n feature vectors, a random feature vector is selected n times and then copied to the dataset used to construct the tree. The copied feature vector is not removed from the original dataset. Thus, there will be multiple copies of some feature vectors. This means that different feature vectors are assigned varying degrees of importance. We defined a 420-dimension feature vector $F = (x_1, x_2, \ldots, x_{420})$ as the input for the RF model.

To construct an individual tree, a small random subset of a fixed size $m \ll 420$ is extracted from the feature vectors at each node. The best split between these m values, which is the split that leads to the highest predictive power, is selected as the condition at each node. Each tree is grown as large as possible without pruning, resulting in low-bias trees. While this results in overfitting for a single decision tree, RF uses a high number of decision trees, which means that the algorithm has low variance. The final number of trees in the forest was set to 750 to optimize run time.

To further enhance the prediction quality of the RF, we also incorporated inverse and reverse PPI representations in our training. Each protein–protein pair (A, B) is therefore also described in the reversed direction (B, A) as well as two combinations with one inversed protein sequence: (A, B_{inv}) and (B, A_{inv}). For example, A: 'ANLMK' and A_{inv}: 'KMLNA'. Among the eight possible protein

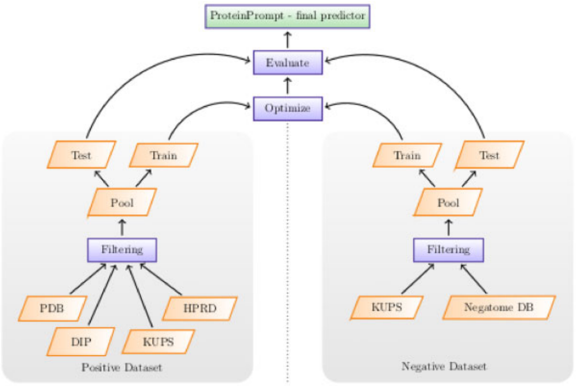

Fig. 2. Overview of the training procedure. Both positive (binder) as well as negative (non-binder) data are collected from all available sources, thoroughly filtered and then split into train and test data

pair representations, the combination of the four listed representations yielded the best prediction results and was therefore used in our RF model. This quadrupling of the data was not necessary for the GNN implementation.

2.4.1 Feature vector calculation

Feature vector calculation includes extracting and transforming sequence-based information into a numerical vector of a constant size. Therefore, it is essential to extract the properties that direct the PPIs.

Each amino acid sequence of a protein–protein complex was transformed into a sequence of numerical values representing seven sequence-derived physicochemical properties. These properties are hydrophobicity (Eisenberg *et al.*, 1984; Koehler *et al.*, 2009), hydropyhilicity (Hopp and Woods, 1981), the volume of the side chains of amino acids (Krigbaum and Komoriya, 1979), polarity (Grantham, 1974), polarizability (Charton and Charton, 1982), solvent-accessible surface area (Rose *et al.*, 1985) and the net charge index of the side chains of amino acids (Zhou *et al.*, 2006). The properties were calculated for each residue in the sequence. These scales are commonly used for protein recognition (Ding and Dubchak, 2001) and to predict protein interactions (Bock and Gough, 2001; 2003), protein alignment (Stamm *et al.*, 2013), protein structure (Durham *et al.*, 2009) or protein functional families (Cai *et al.*, 2003). These applications suggest that these properties significantly contribute to the stability of protein–protein complexes. Each amino acid scale was normalized as follows:

$$P'_i = (P_i - \overline{P})/\sigma_P \tag{1}$$

where $\overline{P}$ is the mean and σ is the standard deviation of the scale-based descriptor covering 20 amino acids, respectively:

$$\overline{P} = \sum_{i=1}^{20} P_i/20 \tag{2}$$

and

$$\sigma_P = \sqrt{\frac{1}{20}\sum_{i=1}^{20}(P_i - \overline{P})^2} \tag{3}$$

AC was then used to transform the data into appropriate feature vectors as follows:

$$AC_{lag,j} = \frac{\frac{1}{n-lag}\sum_{i=1}^{n-lag}(S_{i,j} - \overline{S_j})(S_{i+lag,j} - \overline{S_j})}{\sigma_{S_j} * \sigma_{S_j}} \tag{4}$$

where S_j is the translated amino acid sequence using the normalized scale-based descriptor P'_j with $j = 1, 2, \ldots, 7$, n is the length of sequence S, $lag = 1, 2, \ldots, 30$ is the shift for which the AC is calculated, $\overline{S_j}$ and σ_{S_j} are the mean and standard deviation of the translated sequence, respectively. Ding *et al.* (2016) showed that a maximum *lag* of <30 tends to lose useful information, while larger values may induce noise. Accordingly, the number of AC values for each of the seven scales is 30. The feature vector describing any individual amino acid sequence has $7 \cdot 30 = 210$ elements or dimensions. Thus, for a pair of sequences, the feature space has 420 dimensions.

2.5 Graph neural networks

GNNs are a relatively new type of neural networks that operate on graphs (Battaglia *et al.*, 2018; Scarselli *et al.*, 2009). Features can be assigned and predicted on a node, edge or graph level. The algorithm takes a graph as input, performs computations on the graph itself through a process called message passing, and returns a graph of identical structure but updated features.

In a message passing round, first, the edge features are updated by a learnable update function that takes connected nodes and current edge features into account. Afterwards, another learnable function calculates new node features based on the aggregation of all connected edges and current node features. In the final step, a graph-level target is constructed by a third learnable function that takes an aggregation of all the nodes and edges as input. Through this process, every node and edge collects information about its local region in the graph; these data are used to infer global features.

We used GNNs to condense the information in protein sequences of varying lengths to a fixed-sized vector representation of each sequence. One graph is constructed for each sequence; nodes represent amino acids, and edges are constructed between the nodes of adjacent amino acids. The node features were encoded using an MLP that transformed the values of the amino acid profiles into a 32-dimension vector. The edge features were encoded from a vector of same size that was initialized with ones. The update functions for the nodes and edges and for the graph were implemented by MLPs.

After five rounds of message passing, a final aggregation function is performed on the graph to calculate a 128-dimension graph-level feature from all nodes and edges. This output can be understood as an abstract representation of each protein. This is done for both sequences. Then, their feature vectors are concatenated to one 256-dimension vector, which is then used as the input for the MLP. The last network returns the prediction of the binding propensity of the two proteins.

The model was built using PyTorch and the PyTorch geometric library. The message passing graph net block was realized using its MetaLayer class with an additional edge update on the graph. Parameters were initialized as described by (Glorot and Bengio, 2010), and a leaky rectified linear unit (ReLU) (Maas *et al.*, 2013) was used for activations, except in the final layer of the last MLP, where a sigmoid activation was used. Training was done using binary cross-entropy loss and the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 1e−5 for 350 epochs. The iteration with the best performance on the test set was refined over an additional 50 epochs with a learning rate of 1e−7. See Supplementary Figure S7 for a graphical illustration of our implementation.

2.6 Consensus of RF and GNN

We trained a simple neural network with the outputs of RF and GNN for performing a consensus prediction. The scores returned by RF and GNN show rather different distributions. Thus, a linear combination is unlikely to result in the best possible consensus. We selected a NN with two inputs, two hidden layers with eight neurons and a single output neuron (2-8-8-1).

3 Results

3.1 Performance RF

Based on our test dataset of 5094 positive and 5094 negative cases (binders versus non-binders), we plotted the receiver operation characteristics (ROC) curve, shown in Figure 3. This allowed us to estimate the overall quality of the predictions and also provides a visual overview of the relationship between true and false positives. Each point on the curve shows how many falsely predicted binders should be expected for a given number of correctly predicted ones. As the datapoints are sorted by their predicted binding propensity, a reasonable threshold can be selected. This allows to fine-tune the balance between sensitivity and specificity.

Sensitivity is the ratio of correctly predicted binders (true positives) to actual binders. Specificity is the ratio of correctly predicted non-binders (true negatives) to all non-binders. For example, in Figure 3, a false positive rate of 0.03 will result in a true positive rate of ∼0.6. Thus, selecting this point (its associated score) as the threshold will result in a hit rate of 60% and only 3% false predictions should be expected.

An ideal signal would lead to a rectangular plot with an area under the curve (AUC) of 1. The other extreme, pure noise, would result in a diagonal line with an AUC of 0.5. Our method results in an AUC of 0.95, specificity of 0.88, sensitivity of 0.87 and an accuracy rate of 0.88. Currently, our method balances specificity and sensitivity to avoid unwanted bias in different applications.

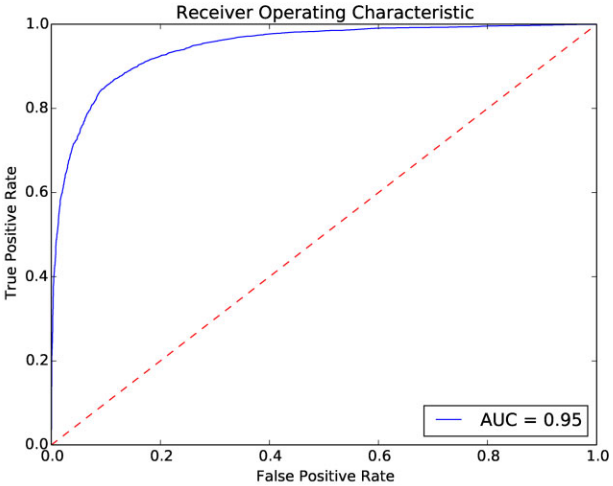

Fig. 3. Performance of `ProteinPrompt` on the entire test dataset of over 10 000 PPIs

Table 1. Quality measures of `ProteinPrompt` compared to other publicly available tools, such as `SPPS` (Liu *et al.*, 2012), `TRI_tool` (Perovic *et al.*, 2017) and `LR_PPI` (Pan *et al.*, 2010)

Tool	AUC	Spec.	Sens.	Acc.
`ProteinPrompt`	0.94	0.88	0.84	0.86
`SPPS`	0.77	0.34	0.96	0.66
`TRI_tool`	0.73	0.95	0.32	0.63
`LR_PPI`	0.60	0.16	0.91	0.53

Note: AUC, specificity, sensitivity and accuracy are listed.

3.1.1 RF compared to other tools

We compared our optimized RF system to publicly available tools, such as `SPPS` (Liu *et al.*, 2012), `TRI_tool` (Perovic *et al.*, 2017) and `LR_PPI` (Pan *et al.*, 2010). Due to limited access, we used the reduced test dataset, as described in the methods section. We also re-evaluated `ProteinPrompt` on the reduced test dataset to ensure comparable results. The results of our evaluation using the reduced test dataset are somewhat different from those we obtained using the entire test dataset.

The results shown in Table 1 and the ROC plot in Figure 4 suggest that `ProteinPrompt` outperforms the other three methods. Furthermore, `ProteinPrompt` balances sensitivity and specificity. `SPPS` and `LR_PPI` have excellent sensitivity of 0.96 and 0.91; however, their specificity is rather poor: 0.34 and 0.16, respectively. `TRI_tool`, on the other hand, shows a massive bias toward specificity, which is 0.95, compared to a sensitivity of 0.32. Furthermore, the overall prediction accuracy and the AUC of our tool are significantly higher. Accuracy: 0.86 versus 0.66 (`SPPS`), 0.63 (`TRI_tool`) or 0.53 (`LR_PPI`). AUC: 0.94 versus 0.77 (`SPPS`), 0.73 (`TRI_tool`) and 0.60 (`LR_PPI`), respectively.

3.1.2 Individual test cases

We further tested the detection rate of `ProteinPrompt` using experimentally verified protein interaction partners (EV PPIs). We compared the output of `ProteinPrompt` to that of the STRING database (https://string-db.org), which has been shown to include a very high number of experimentally proven PPIs (Kumar Bajpai *et al.*, 2020). Here, the EV PPIs of five different prominent proteins with various cellular functions (Supplementary Table S9) with high confidence (score > 0.7) were investigated. The PPI predictions of `ProteinPrompt` were compared to those of the STRING database. `ProteinPrompt` found all but one of the EV PPIs output by the STRING Database; for the SRC gene one of nine binding partners was not identified (Supplementary Table S9). The scores given by the STRING database and the scores of `ProteinPrompt` were then statistically analyzed and plotted as boxplots to enable direct comparison (Fig. 5). On average, `ProteinPrompt` predicted all identified EV PPIs with comparable or better accuracy than the STRING database (Fig. 5, Supplementary Table S9). However, due to the complexity of PPIs in nature, STRING scores do not necessarily reflect real protein–protein affinities. On our test dataset, `ProteinPrompt` was able to rapidly predict EV PPIs with an average accuracy rate of 0.89 and an SD of 0.09. From a user's perspective, `ProteinPrompt` identified 98% of all EV PPIs with a predicted score above 0.7, 84% of all EV PPIs with a predicted score above 0.8 and 46% of all EV PPIs with a predicted score above 0.9.

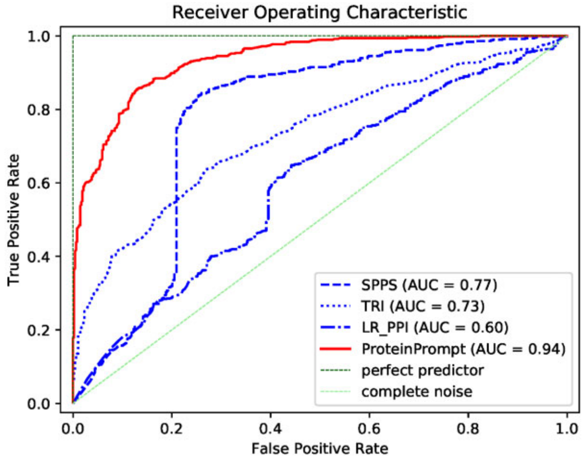

Fig. 4. ROC curves for `ProteinPrompt`, `SPPS`, `TRI_tool`, `LR_PPI` using the reduced test dataset of 968 protein–protein pairs

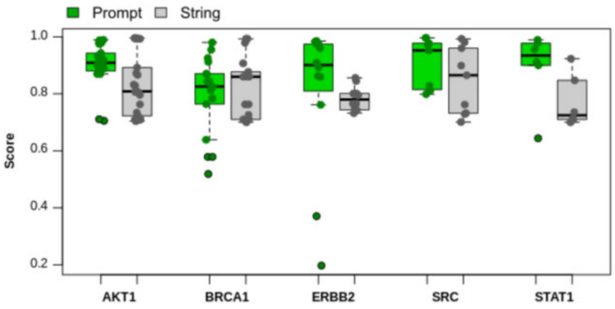

Fig. 5. EV PPI scores from `ProteinPrompt` (green) and from STRING database (grey) for a test set of five different proteins, indicated by their gene names in the boxplot

In Supplementary Section S3.1.2, a further 20 individual test cases are presented, taken from Dervishi *et al.* (2018), following the same approach.

3.1.3 Gold standard reference datasets

3.1.3.1 `PrePPI` *high confidence dataset.* As the first gold standard dataset of reliable PPIs, we chose the high-confidence human dataset (http://honiglab.c2b2.columbia.edu/PrePPI/ref/data/human.db.hc.201008.intm) published in the `prePPI` database (Zhang *et al.*, 2012). This dataset contains 7410 binding protein–proteins pairs that have been mentioned in at least two independent publications. After removing proteins contained in our training dataset and those containing the amino acid selenocysteine, we obtained a set of 7318 PPI candidates.

The RF classified a total of 95.8% of these protein pairs as binders. After filtering out protein pairs already present in our training data, we still achieved an accuracy of 90.23% for the remaining 3102 PPIs. The GNN predicted 85% of the total dataset as binders and 83.9% of the filtered dataset.

3.1.3.2 Dataset of Park and Marcotte. As an additional test, we used the (Park and Marcotte, 2012) dataset on which a number of methods were tested, allowing a direct comparison (Park and Marcotte, 2012) tested seven methods (Ding *et al.*, 2016), extended this with an additional three methods. ProteinPrompt in its current form performs similarly on this dataset as it does on our own test data. For all human subsets, ProteinPrompt achieved significantly higher accuracy than all other methods. When ProteinPrompt was trained with this data, the performance is similar to the other top-ranking tools. For most human subsets, the retrained version of ProteinPrompt ranked second. First, this confirms that combining RF with autocorrelation has the potential to achieve high accuracy. Second, this test highlights the critical impact of the dataset on the final performance. For the yeast data, ProteinPrompt does not perform as well, indicating limited transferability to other organisms, a fact also observed with other learning methods. Most learning methods are only valid within the domain of their training data. A detailed description can be found in Supplementary Materials.

3.2 Performance GNN

In the first test, we used the same seven amino acid scales that were used for the RF. We compared this to a graph topology in which 13 and 15 scales were assigned to each amino acid. The resulting accuracy rates are listed in Table 2.

Adding as many residue scales as possible to the graph does not improve the accuracy. Currently, the maximum accuracy is 86% for 13 residue scales. Extensive future research would be needed to further maximize the performance of the GNN.

In cross-validations, in which 20% of the training data were used as validation set, the accuracies for the validation data were nearly identical to the one of the test data. This reflects the low level of redundancy within the training dataset, which is shown in Supplementary Figure S5. We therefore considered it acceptable to perform a final optimization with 95% of the test dataset added to the training data. The resulting model is the one uploaded to the server. This strategy should lead to the most generally applicable model.

3.3 Performance consensus RF and GNN

The consensus using the RF and GNN as input was leading to a slight improvement with a final accuracy of 89.4%.

3.4 Analysis of feature importance

We first calculated the correlations of all combinations of the amino acid scales. A principle component analysis yielded very similar results.

To analyze which of the features was most important to the performance of the model, *permutation feature importance* (Breiman 2001) was performed by permuting the input values of one feature at a time and then calculating the loss of performance of the trained model on the test set, measured in accuracy, sensitivity, specificity, and AUC ROC. The detailed results can be found in Supplementary Section S3.4.

The GNN is not robust to such permutations, and loss of information from even one feature results in accuracy losses of up to 30% (see also Supplementary Fig. S12), with an even greater loss in sensitivity and specificity and AUC ROC. Unfortunately, this does not provide any insight into the importance of the features, but merely shows the lack of robustness of the network to such permutations.

Table 2. Accuracies of GNNs trained on different numbers of amino acid profiles

# profiles	Accuracy
7	83.5%
13	86.0%
15	83.8%

A different result emerged when analyzing the feature importance of the RF, as shown in Supplementary Figure S11. The RF, most likely due to the fact that it is an ensemble model, was much less perturbed by the permutation of a feature and ‘only’ lost a maximum of 15% in accuracy and about twice as much in sensitivity. Removing the importance of a feature also increased specificity.

For the RF, the most important features (according to the permutation analysis) are polarity, hydrophilicity and hydrophobicity. Although polarity, hydrophilicity and hydrophobicity are strongly correlated (up to 90%), the loss of one feature still had a strong effect on model performance (up to the mentioned 15%). This indicates that correlation alone is a poor criterion for feature reduction.

The detailed results can be found in Supplementary Section S3.4.

3.5 Webserver implementation

ProteinPrompt is available online as a webserver at: http://proteinformatics.org/ProteinPrompt.

Basic usage is as simple as providing an input sequence. By default, ProteinPrompt will search our manually curated database of human proteins, which contains 27 223 sequences. Scanning the entire human database takes ~1 min for the default method RF. For GNN, the search is even faster, but only the model trained on 15 scales is provided on the server. The consensus has to execute both methods and therefore requires correspondingly more time. Other, more extensive databases are also provided, including mammalian, vertebrate and metazoan protein sets. Searching these databases takes considerably longer. For example, the vertebrate database has 91 592 entries; therefore, searching it takes approximately three times longer than searching the human protein database.

The server is free for academic users. Providing an email address is optional. ProteinPrompt was originally optimized for sequences with a minimum length of 16 AA. The sequence length of the uploaded proteins is detected automatically. When the user provides a sequence shorter than 16 AA, a warning appears, but the calculation is still performed. As output, a ranked and scored list of proteins from the database is returned; this list can be downloaded.

4 Discussion

ProteinPrompt offers a reliable, fast way to predict protein interaction partners based on protein sequences. It is available as an easy-to-use online tool and is thus accessible to non-expert users. In order to develop this fast, reliable service, we optimized the learning algorithm, the binding database, and the representations of the sequences. We determined that the RF algorithm combined with autocorrelation on seven amino acid scales resulted in the highest accuracy.

We also determined that the GNN method performed nearly as well as the RF algorithm. To support consensus building, we offer both implementations on our server. It is remarkable that the RF algorithm, which is conceptually comparably simple, performs so well on such a complex task. This is even more remarkable considering that the RF approach, unlike the GNN, does not include simultaneous optimization of feature vector creation and model building.

An extensive database with limited redundancy was essential to reliably test the tool. This database was obtained through several iterations of manual curation of the test and training datasets. Despite the strict separation of the training and test datasets, which posed a significant learning challenge, ProteinPrompt turned out to perform very well compared to other available servers and methods. ProteinPrompt is reasonably accurate and can scan the entire human proteome within approximately one minute. To our knowledge, ProteinPrompt is currently the fastest online service available for scanning different proteoms to identify potential binding partners based on a sequence level. It is reasonable to assume that expanding the training data would lead to higher accuracy rates. Based on our extensive tests, we expect ProteinPrompt to support a better understanding of the complex networks of PPIs, which are the basis for a broad range of biological mechanisms.

Acknowledgements

We extend our thanks to Johanna Tiemann, Alexander Vogel, Thorsten Kaiser and Daniel Wiegreffe for proofreading.

Funding

This work was supported by the Sächsische Aufbaubank (SAB) [100256197] and Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through CRC 1423 [421152132; sub-project Z04] as well as through the Open Access Publishing Fund of Leipzig University within the DFG program Open Access Publication Funding.

Conflict of Interest: none declared.

Data availability

The data underlying this article are available on: http://proteinformatics.org/ProteinPrompt.

References

Barker,D. and Pagel,M. (2005) Predicting functional gene links from phylogenetic-statistical analyses of whole genomes. *PLoS Comput. Biol.*, **1**, e3.

Battaglia,P.W. *et al.* (2018) *Relational Inductive Biases, Deep Learning, and Graph Networks. arXiv* abs/1806.01261.

Berman,H.M. *et al.* (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.

Betel,D. *et al.* (2007) Structure-templated predictions of novel protein interactions from sequence information. *PLoS Comput. Biol.*, **3**, 1783–1789.

Blohm,P. *et al.* (2014) Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic Acids Res.*, **42**, D396–D400.

Bock,J.R. and Gough,D.A. (2001) Predicting protein–protein interactions from primary structure. *Bioinformatics*, **17**, 455–460.

Bock,J.R. and Gough,D.A. (2003) Whole-proteome interaction mining. *Bioinformatics*, **19**, 125–134.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Cai,C.Z. *et al.* (2003) SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res.*, **31**, 3692–3697.

Charton,M. and Charton,B.I. (1982) The structural dependence of amino acid hydrophobicity parameters. *J. Theor. Biol.*, **99**, 629–644.

Chen,M. *et al.* (2019) Multifaceted protein-protein interaction prediction based on siamese residual RCNN. *Bioinformatics*, **35**, i305–i314.

Chen,X.W. *et al.* (2011) KUPS: constructing datasets of interacting and non-interacting protein pairs with associated attributions. *Nucleic Acids Res.*, **39**, D750–D754.

Chen,Y. *et al.* (2020) Protein interface complementarity and gene duplication improve link prediction of protein-protein interaction network. *Front. Genet.*, **11**, 291.

Ching,T. *et al.* (2018) Opportunities and obstacles for deep learning in biology and medicine. *J R Soc Interface*, **15, 20170387**.

Clauset,A. *et al.* (2008) Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**, 98–101.

Das,S. and Chakrabarti,S. (2021) Classification and prediction of protein-protein interaction interface using machine learning algorithm. *Sci. Rep.*, **11**, 1761.

Dervishi,I. *et al.* (2018) Protein-protein interactions reveal key canonical pathways, upstream regulators, interactome domains, and novel targets in ALS. *Sci. Rep.*, **8**, 14732–14710.

Ding,C.H. and Dubchak,I. (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, **17**, 349–358.

Ding,Y. *et al.* (2016) Predicting protein-protein interactions via multivariate mutual information of protein sequences. *BMC Bioinformatics*, **17**, 398.

Durham,E. *et al.* (2009) Solvent accessible surface area approximations for rapid and accurate protein structure prediction. *J. Mol. Model.*, **15**, 1093–1108.

Eisenberg,D. *et al.* (1984) Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *J. Mol. Biol.*, **179**, 125–142.

Fu,L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

Garcia-Garcia,J. *et al.* (2017) iFrag: a protein-protein interface prediction server based on sequence fragments. *J. Mol. Biol.*, **429**, 382–389.

Glorot,X. and Bengio,Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In: Yee Whye, T. and Mike, T. (eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Volume 9 of Proceedings of Machine Learning Research*. PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256.

Grantham,R. (1974) Amino acid difference formula to help explain protein evolution. *Science*, **185**, 862–864.

Hamp,T. and Rost,B. (2015) Evolutionary profiles improve protein-protein interaction prediction from sequence. *Bioinformatics*, **31**, 1945–1950.

Hamp,T. and Rost,B. (2015) More challenges for machine-learning protein interactions. *Bioinformatics*, **31**, 1521–1525.

Hashemifar,S. *et al.* (2018) Predicting protein-protein interactions through sequence-based deep learning. *Bioinformatics*, **34**, i802–i810.

Hopp,T.P. and Woods,K.R. (1981) Prediction of protein antigenic determinants from amino acid sequences. *Proc. Natl. Acad. Sci. USA*, **78**, 3824–3828.

Keshava Prasad,T.S. *et al.* (2009) Human protein reference database–2009 update. *Nucleic Acids Res.*, **37**, D767–D772.

Koehler,J. *et al.* (2009) A unified hydrophobicity scale for multispan membrane proteins. *Proteins*, **76**, 13–29.

Kong,M. *et al.* (2020) Weighted sparse representation based classification. *Front. Genet.*, **11**, 18.

Krigbaum,W.R. and Komoriya,A. (1979) Local interactions as a structure determinant for protein molecules: II. *Biochim. Biophys. Acta*, **576**, 204–248.

Kuhn,M. (2008) Building predictive models in r using the caret package. *J. Stat. Soft.*, **28**, 1–26.

Kumar Bajpai,A. *et al.* (2020) Systematic comparison of the protein-protein interaction databases from a user's perspective. *J. Biomed. Inform.*, **103**, 103380.

Li,W. and Godzik,A. (2006) CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

Li,X. *et al.* (2019) Prediction of protein-protein interactions based on domain. *Comput. Math. Methods Med.*, **2019**, 5238406.

Licata,L. *et al.* (2012) Mint, the molecular interaction database: 2012 update. *Nucleic Acids Res.*, **40**, D857–D861.

Liu,X. *et al.* (2012) SPPS: a sequence-based method for predicting probability of protein-protein interaction partners. *PLoS One*, **7**, e30938.

Loshchilov,I. and Hutter,F. (2017) Decoupled weight decay regularization. *arXiv preprint* https://doi.org/10.48550/arXiv.1711.05101.

Maas,A.L. *et al.* (2013) Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Vol. **30**. Citeseer, p. 3.

Martin,S. *et al.* (2005) Predicting protein-protein interactions using signature products. *Bioinformatics*, **21**, 218–226.

McDowall,M.D. *et al.* (2009) PIPs: human protein-protein interaction prediction database. *Nucleic Acids Res.*, **37**, D651–D656.

Murakami,Y. and Mizuguchi,K. (2014) Homology-based prediction of interactions between proteins using averaged one-dependence estimators. *BMC Bioinformatics*, **15**, 213.

Ofran,Y. and Rost,B. (2003) Predicted protein-protein interaction sites from local sequence information. *FEBS Lett.*, **544**, 236–239.

Orchard,S. *et al.* (2014) The mintact project–intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.*, **42**, D358–D363.

Pan,X.Y. *et al.* (2010) Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *J. Proteome Res.*, **9**, 4992–5001.

Park,Y. and Marcotte,E.M. (2012) Flaws in evaluation schemes for pair-input computational predictions. *Nat. Methods*, **9**, 1134–1136.

Pawson,T. (2004) Specificity in signal transduction: from phosphotyrosine-SH2 domain interactions to complex cellular systems. *Cell*, **116**, 191–203.

Pazos,F. and Valencia,A. (2001) Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Eng.*, **14**, 609–614.

Perovic,V. *et al.* (2017) Tri_tool: a web-tool for prediction of protein-protein interactions in human transcriptional regulation. *Bioinformatics*, **33**, 289–291.

Planas-Iglesias,J. *et al.* (2013) iLoops: a protein-protein interaction prediction server based on structural features. *Bioinformatics*, **29**, 2360–2362.

Rose,G.D. *et al.* (1985) Hydrophobicity of amino acid residues in globular proteins. *Science*, **229**, 834–838.

Salwinski,L. *et al.* (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, **32**, D449–D451.

Scarselli,F. *et al.* (2009) The graph neural network model. *IEEE Trans. Neural Netw.*, **20**, 61–80.

Shen,J. *et al.* (2007) Predicting protein-protein interactions based only on sequences information. *Proc. Natl. Acad. Sci. USA*, **104**, 4337–4341.

Singh,R. *et al.* (2010) Struct2Net: a web service to predict protein-protein interactions using a structure-based approach. *Nucleic Acids Res.*, **38**, W508–W515.

Stamm,M. *et al.* (2013) Alignment of helical membrane protein sequences using alignme. *PLoS One*, **8**, e57731.

Sun,T. *et al.* (2017) Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, **18**, 277.

Szklarczyk,D. *et al.* (2011) The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.*, **39**, D561–D568.

Tran,L. *et al.* (2018) ProfPPIdb: pairs of physical protein-protein interactions predicted for entire proteomes. *PLoS One*, **13**, e0199988.

Wang,L. *et al.* (2019) Predicting protein-protein interactions from matrix-based protein sequence using convolution neural network and feature-selective rotation forest. *Sci. Rep.*, **9**, 9848.

Yang,F. *et al.* (2020) Graph-based prediction of protein-protein interactions with attributed signed graph embedding. *BMC Bioinformatics*, **21**, 323.

Yao,Y. *et al.* (2019) An integration of deep learning with feature embedding for protein-protein interaction prediction. *PeerJ*, **7**, e7126.

Yook,S.H. *et al.* (2004) Functional and topological characterization of protein interaction networks. *Proteomics*, **4**, 928–942.

Zhang,Q.C. *et al.* (2012) Structure-based prediction of protein-protein interactions on a genome-wide scale. *Nature*, **490**, 556–560.

Zhang,Q.C. *et al.* (2013) PrePPI: a structure-informed database of protein-protein interactions. *Nucleic Acids Res.*, **41**, D828–D833.

Zhou,P. *et al.* (2006) Genetic algorithm-based virtual screening of combinative mode for peptide/protein. Acta Chimica Sinica., **64**, 691–697.