

# Sequence-based prediction of protein-protein interactions: a structure-aware interpretable deep learning model

Samuel Sledzieski<sup>\*†</sup>   Rohit Singh<sup>\*†</sup>   Lenore Cowen<sup>‡¶</sup>   Bonnie Berger<sup>§¶</sup>

## Abstract

Protein-protein interaction (PPI) networks have proven to be a valuable tool in systems biology to facilitate the discovery and understanding of protein function. Unfortunately, experimental PPI data remains sparse in most model organisms and even more so in other species. Existing methods for computational prediction of PPIs seek to address this limitation, and while they perform well when sufficient within-species training data is available, they generalize poorly to new species or often require specific types and sizes of training data that may not be available in the species of interest. We therefore present D-SCRIPT, a deep learning method for predicting a physical interaction between two proteins given just their sequences. Compared to existing methods, D-SCRIPT generalizes better to new species and is robust to limitations in training data size. Our approach encodes the intuition that for two proteins to physically interact, a subset of amino acids from each protein should be in contact with the other. The intermediate stages of D-SCRIPT directly implement this intuition; the penultimate stage in D-SCRIPT is a rough estimate of the inter-protein contact map of the protein dimer. This structurally-motivated design enables interpretability of our model and, since structure is more conserved evolutionarily than sequence, improves generalizability across species. We show that a D-SCRIPT model trained on 38,345 human PPIs enables significantly improved functional characterization of fly proteins compared to the state-of-the-art approach. Evaluating the same D-SCRIPT model on protein complexes with known 3-D structure, we find that the inter-protein contact map output by D-SCRIPT has significant overlap with the ground truth. Our work suggests that recent advances in deep learning language modeling of protein structure can be leveraged for protein interaction prediction from sequence. D-SCRIPT is available at <http://dscrip.t.csail.mit.edu>.

---

\*These authors contributed equally to the work

<sup>†</sup>Computer Science and Artificial Intelligence Lab., Massachusetts Institute of Technology, Cambridge, MA 02139

<sup>‡</sup>Department of Computer Science, Tufts University, Medford, MA 02155

<sup>§</sup>Computer Science and Artificial Intelligence Lab. and Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139

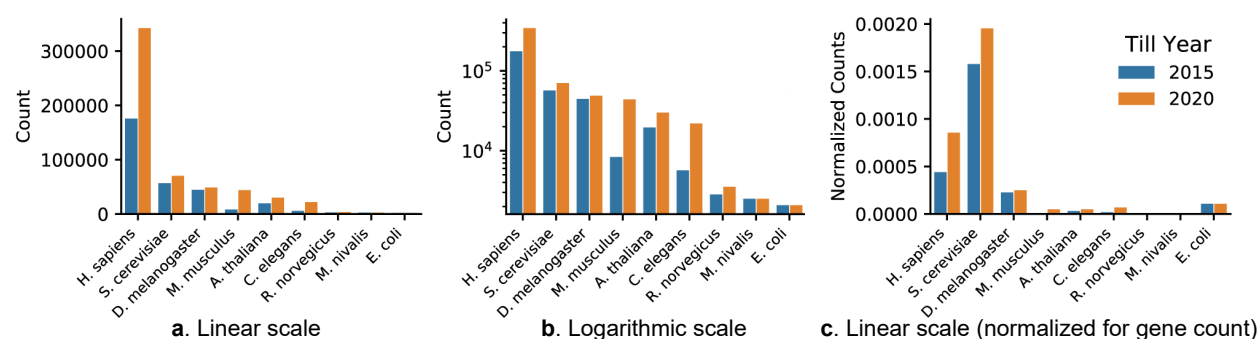
<sup>¶</sup>Co-corresponding authors: [cowen@cs.tufts.edu](mailto:cowen@cs.tufts.edu), [bab@mit.edu](mailto:bab@mit.edu)

# 1 Introduction

The systematic mapping of physical protein-protein interactions (PPIs) in the cell has proven extremely valuable in deepening our understanding of protein function and biology. In species such as yeast and humans where a large network of experimentally determined PPIs exists, this PPI network information has proven valuable for downstream inference tasks in understanding functional genomics and biological pathway analysis [1–5]. However, despite the introduction of high-throughput methods [6–10] to assay PPIs, the experimentally determined human PPIs to date are believed to represent only a small fraction of the true set of protein pairs that physically bind in the human cell [11]. In other species, the data is even sparser. As Figure 1 shows, in the case of most model organisms, the number of experimentally determined PPIs is far smaller than for human, and in the case of non-model organisms, it can be nearly non-existent. This motivates our study of computational methods to predict PPIs.

Here, we introduce a new deep learning method, D-SCRIPT (Deep Sequence Contact Residue Interaction Prediction Transfer), for determining if two proteins interact physically in the cell based on their amino acid sequences. Our key conceptual advance is that a well-matched combination of input featurization and model architecture allow for the model to be trained solely from sequence data, supervised only with a binary interaction label, and yet produce an intermediate representation that substantially captures the structural mechanism of interaction between the protein pair. Our feature construction takes advantage of recent advances in deep learning-based language modeling of single protein structures. Using Bepler and Berger’s [12] pre-trained language model, we construct informative protein embeddings that are endowed with structural information about each of the proteins. The internal representation of our model uses these features to explicitly encode the intuition that a physical interaction between two proteins requires that a subset of the residues in each protein be in contact with the other protein. We do so by having an inter-protein contact-map matrix as the penultimate stage in our model, which is then summarized into a single score (Figure 2). These innovations are combined with a novel pooling operation and a regularization loss function that successfully guide the model towards structurally-plausible contact maps during training.

Our design enables D-SCRIPT to offer a combination of advantages that have hitherto been unavailable simultaneously: broad applicability, interpretability, and high cross-species accuracy. It can be applied to arbitrary pairs of protein sequences, including between-species interactions. The intermediate contact-



**Figure 1: The corpus of experimental PPI data is limited.** We show the counts of PPIs in various organisms detected via direct binding experiments (sourced from BioGRID, see Appendix A.1 for details). For readability, we offer both (a) linear and (b) log-scale plots, as well as (c) PPI counts normalized by the square of the species gene-count, to adjust for the number of possible gene pairs. Note the relative lack of PPI data in species other than human and yeast. Furthermore, the growth of this corpus over the past five years has been modest (except in the case of human and mouse PPIs).

map representation offers a physical interpretation of the interaction. Since structure is more conserved than sequence over evolutionary time [13], this physical model of interaction should translate well across species, making D-SCRIPT more generalizable. We note that the use of Bepler and Berger’s pre-trained model allows us to indirectly benefit from the rich data on 3-D structures of individual proteins. In contrast, a PPI prediction method that was directly supervised with 3-D structures of protein complexes, in order to learn the physical mechanism of interaction, would need to contend with the relatively small size of that corpus [14–16].

We demonstrate D-SCRIPT’s success compared to competing methods in correctly predicting PPIs in organisms with sparse PPI networks. We evaluate D-SCRIPT in the cross-species prediction setting, where a method trained on human PPIs is used to predict PPIs in several less-studied model organisms. We show that in this context, D-SCRIPT substantially improves upon existing methods, including the state-of-the-art deep learning method PIPR [17]. To investigate intra-species performance, we compare D-SCRIPT’s prediction of PPI interactions in the human PPI network, when trained through cross-validation on human PPI interaction data. We find, as expected, that state-of-the-art PIPR substantially outperforms D-SCRIPT when predicting interactions between proteins that have many PPI examples in the training set, but the situation is reversed for proteins with a paucity of PPI interactions in the training set. A simple hybrid method that jointly incorporates the confidence of each method performs best of all.

To examine if the higher cross-species accuracy of D-SCRIPT translates to improved inference of protein function, we run an analysis pipeline to predict functional modules in *Drosophila* using networks predicted by D-SCRIPT and PIPR, and show that D-SCRIPT achieves more functionally coherent clusters. On evaluating the physical plausibility of the intermediate contact map representation, we remarkably find that the map partially discovers the structural mechanism of an interaction despite the model having been trained only on sequence data. Specifically, we evaluate our predictions on Vreven et al.’s [18] benchmark database of 3-D structures of docked protein-pairs and observe that our model’s predicted contact map is significantly similar to the ground-truth inter-protein contact map in cases where our model predicts an interaction. These contact map predictions could be used to augment methods which predict protein interaction from known contacts, such as Hosur et al. [19]. Furthermore, our modular design enables the investigation of model output at various intermediate stages; we demonstrate that each additional stage captures incremental biological intuition. We also present preliminary predictions for viral-host PPIs between SARS-CoV-2 & Human.

**Related Work.** D-SCRIPT, like other recent successful deep learning methods PIPR and DPPI [17, 20], belongs to the class of methods that perform PPI prediction from protein amino acid sequence alone, in contrast to a different class of highly successful PPI prediction methods based on network information. Network-based PPI prediction methods use “guilt by association” or more sophisticated network propagation or diffusion methods [4] to computationally predict pairs of proteins that physically interact by bootstrapping from the connectivity patterns of the known interactions [21–24]. Alternatively, other types of protein-protein association that can be more widely available, such as co-expression and co-localization, have also been shown to correlate with physical protein-protein interaction. In cases where there is a reasonably-sized training set and/or a rich set of input data modalities, these methods perform very well and can be powerful tools [3, 25–27]. Unfortunately, there has been limited success in adopting these methods beyond those few model organisms where there is sufficient known interaction (e.g. yeast and Human networks) or association data (e.g. yeast networks). Taking advantage of the abundant genetic-interaction coverage in yeast, Khurana et al. [28] described one of the few successful approaches to transfer them to human (where the data is sparse) to glean insights into neurodegenerative disease.

The advantage of a sequence-based approach like D-SCRIPT is that the input sequence data is almost

always available, due to the enormous advances in low-cost genome sequencing. Among sequence-based methods, D-SCRIPT’s strength is in its greater cross-species generalizability and more accurate predictions in cases where the existing training data is sparse. Moreover, our contact-map based approach parallels the recent work on deep learning prediction of single-structure contact maps from protein sequence [29–31]. While these methods are trained on 3-D structure data, our method is designed to be trained using only sequence data. Nonetheless, insights from our approach and these methods could potentially be combined in future work.

## 2 Methods

### 2.1 Model Architecture

Our deep learning model for predicting PPIs directly from protein sequences, similar to previous deep learning methods DPPI [20] and PIPR [17], can be composed into two stages:

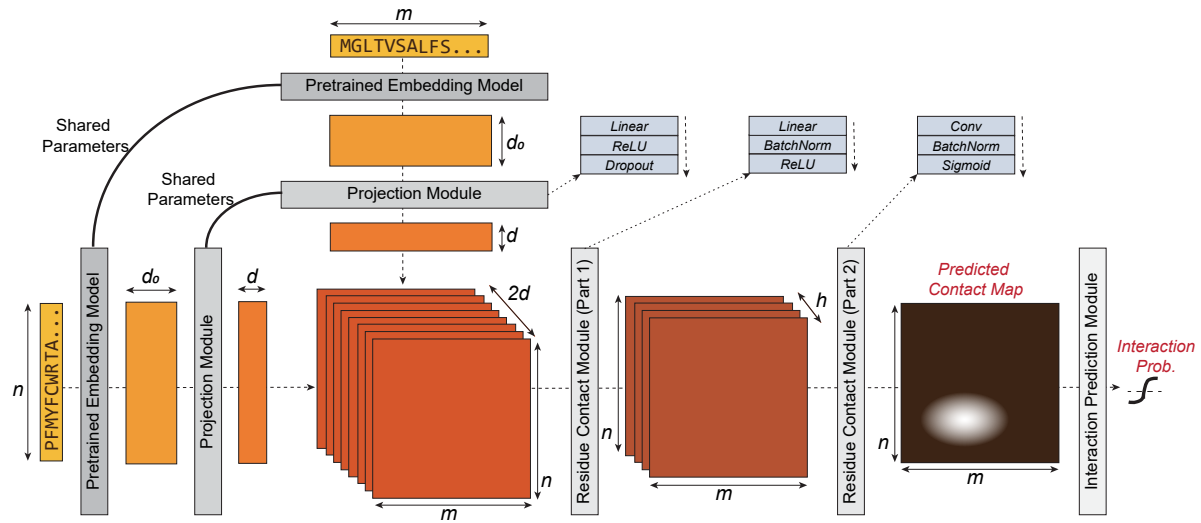
- Stage 1: Generate a rich feature representation for each protein separately
- Stage 2: Predict an interaction based on these features

where we note that the model is trained end-to-end across both stages. In both DPPI and PIPR, most of the complexity in the model is built into Stage 1. Once a feature representation has been computed, the interaction prediction for DPPI or PIPR is based on a direct element-wise product of the two protein feature vectors. A key innovation of D-SCRIPT is in our design of a more structurally-aware Stage 2. Stage 1 is accomplished by using the pre-trained protein sequence model from Bepler & Berger followed by a projection module (Section 2.1.2), where the model learns low-dimensional protein embeddings which can also be used as compact representations for downstream interaction and structural prediction tasks. For the second stage, we present a novel architecture that encodes a *physical* model of protein interaction: we predict two proteins to interact only if there exists a short sequence of residues in the first protein that is highly compatible with a sequence of residues in the second protein. In the contact module (Section 2.1.3), the low-dimensional embeddings are used to compute a sparse contact map which predicts the locations of contacts between protein residues. Finally, the interaction module (Section 2.1.4) uses a customized max-pooling operation on the contact map to predict the probability of interaction between the input proteins.

#### 2.1.1 Input & Structure-aware Embedding

The input to D-SCRIPT is a pair of protein sequences  $S_1, S_2$  with lengths  $(n, m)$  and it outputs an interaction probability  $\hat{p} \in [0, 1]$  and a predicted-contact matrix  $\hat{C} \in [0, 1]^{n \times m}$ .

We first generate embeddings  $E_1 \in \mathcal{R}^{n \times d_0}, E_2 \in \mathcal{R}^{m \times d_0}$  by embedding  $S_1$  and  $S_2$  with a pre-trained model from Bepler & Berger. Their model is a Bi-LSTM (bidirectional long short-term memory) neural network trained on three independent pieces of information: 1) the protein’s SCOP classification, indicating its general structure, 2) self-contact map of a protein’s 3-D structure, and 3) sequence alignment of similar proteins. These embeddings capture both local and global structural features of the protein sequences: the  $d_0$ -dimensional encoding of each amino acid contains information not just about the amino acid and its immediate context, but also the global structure of the protein. This is a key distinction from other approaches (e.g. Chen et al.’s in PIPR), where each amino acid’s embedding represents just its biochemical properties or a short-range context (e.g., 5-7 residues) around it. We note that alternative embeddings (e.g. Rives et al. [32] or Luo et al. [33]) can potentially be substituted here.



**Figure 2: D-SCRIPT architecture overview.** Left to right: The Pre-trained Embedding Model generates features for each individual protein. The Projection Module reduces them to  $d$  dimensions. Each low-dimensional single-protein embedding implicitly includes, among other features, an encoding that broadly captures the protein’s residue-contact map. The novel portion of the architecture combines these low-dimensional embeddings to compute a sparse *inter-protein* contact map through a two-step process which first computes a representation for each pair of residues, then incorporates local information using a convolutional filter. The last module (Interaction Prediction Module) uses a customized max-pooling operation (Section 2.1.4) to predict the probability of interaction between the input proteins.

### 2.1.2 Projection Module

In the projection module, we reduce  $E_1$  and  $E_2$  to  $d$ -dimensional representations using a fully-connected linear layer (multi-layer perceptron) with  $d_0$  input and  $d$  output nodes. Specifically, given an input embedding  $E \in \mathcal{R}^{n \times d_0}$ , we compute the embedding projection  $Z \in \mathcal{R}_{\geq 0}^{n \times d}$  as

$$Z_i = \text{Drop}(\text{ReLU}(E_i W_1 + b_1)) \quad \forall i \in 1 \dots n \quad (1)$$

with  $W_1 \in \mathcal{R}^{d_0 \times d}$ ,  $b_1 \in \mathcal{R}^d$  as learned weights and biases. The rectified linear unit (ReLU) is a non-linear operation which applies the transformation  $\text{ReLU}(x) = \max(0, x)$ . The dropout layer (Drop) randomly sets 50% of the weights to zero, helping prevent over-fitting in  $W_1$ .

### 2.1.3 Residue Contact Module

The residue contact model takes the  $d$ -dimensional embeddings  $Z_1, Z_2$  and models the interaction between the residues of each protein. First, for each pair of residue embeddings  $Z_{1_i}, Z_{2_j} \in \mathcal{R}^d, i \in 1, \dots, n, j \in 1, \dots, m$ , we compute a broadcast matrix with hidden dimension  $h$ ,  $B_{z_0, z_1} \in \mathcal{R}_{\geq 0}^{n \times m \times h}$

$$\text{diff}_{i,j} = |Z_{0_i} \ominus Z_{1_j}| \quad (2)$$

$$\text{mul}_{i,j} = Z_{0_i} \odot Z_{1_j} \quad (3)$$

$$B_{i,j} = \text{ReLU}(\text{Batch}([\text{diff}_{i,j}, \text{mul}_{i,j}]W_2 + b_2)) \quad (4)$$

where  $\ominus$  indicates the element-wise difference and  $\odot$  indicates the Hadamard product. This featurization is symmetric and has been previously used in NLP and protein sequence modeling tasks [12, 34].

$W_2 \in \mathcal{R}^{2d \times h}$ ,  $b_2 \in \mathcal{R}^h$  are the learned weights and biases. The batch normalization operation normalizes the mean and variance of the input features, thus stabilizing the learning process. Each element  $B_{i,j}$  captures the direct interaction between residues  $S_{1_i}$  and  $S_{2_j}$ . The broadcast matrix  $B$  is used to compute the contact prediction matrix  $\hat{C} \in [0, 1]^{n \times m}$ , where

$$\hat{C}_{i,j} = \sigma(\text{Batch}(\text{Conv}(B_{(i-w:i+w),(j-w:j+w)}))) \quad (5)$$

The two-dimensional convolution (Conv) operation with width  $2w + 1$  and  $h$  channels uses the  $h$ -dimensional representation of all residues within  $w$  of  $B_{i,j}$  to compute  $\hat{C}_{i,j}$ , and thus detects local patterns in two-dimensional residue contact space. The broadcast matrix is zero-padded to allow for the convolution operation to be performed at all indices. We again apply a batch normalization to stabilize learning. We apply the sigmoid operation  $\sigma$ , which restricts the output values of  $\hat{C}$  to be in the range  $[0, 1]$ , and thus they can be interpreted as the predicted probability that two residues are in contact.

### 2.1.4 Interaction Prediction Module

The interaction prediction module computes a single probability of interaction  $\hat{p}$  from the  $n \times m$  contact map  $\hat{C}$ . To do so, we perform two pooling operations. The first is a standard max-pool: an  $l$ -dimensional max-pool divides  $\hat{C}$  into  $\lceil \frac{n}{l} \rceil \times \lceil \frac{m}{l} \rceil$  non-overlapping regions and takes the maximum value of each region, with zero-padding applied where necessary. This max-pooled matrix  $P$  represents the probability of interaction in local regions of the proteins and maintains only the highest-probability residue contacts in each region for global prediction. The second pooling operation is a global pooling operation, calculated as

$$Q_{i,j} = \text{ReLU}(P_{i,j} - \mu - (\gamma * \sigma^2)) \quad (6)$$

$$\hat{p}_{raw} = \frac{\sum_{i,j} Q_{i,j}}{\sum_{i,j} (\text{sign}(Q_{i,j})) + 1} \quad (7)$$

where  $\mu, \sigma^2$  are the mean and variance of the  $P_{ij}$  values and  $\gamma$  is a learned parameter. The matrix  $Q$  sparsifies  $P$ , maintaining only those contacts which are  $\gamma\sigma^2$  greater than the mean, and setting all others to zero. We then predict that the proteins will interact with the average probability of interaction among these high-probability contacts. Together with the regularization that the contact matrix be sparse (see Section 2.2), this global pooling operation captures the intuition that a pair of interacting proteins will be characterized by a relatively small number of high-probability interacting residues or regions.

The final step of interaction prediction is designed to enhance the bimodality of the output distribution, so that the choice of a cutoff becomes less important in distinguishing positive and negative predictions. We apply the logistic activation function to compute the output probability  $\hat{p} = \sigma_{(x_0, \eta)}(\hat{p}_{raw})$  where

$$\sigma_{(x_0, \eta)}(x) = \frac{1}{1 + e^{-\eta(x - x_0)}} \quad (8)$$

This activation function, with  $x_0 = 0.5$ , takes our raw probability predictions and makes them more “extreme”, depressing values below  $x_0$  towards 0 and inflating values above  $x_0$  towards 1, with  $\eta$  controlling the rate at which this occurs. We return  $\hat{p}$  and  $\hat{C}$  as the model prediction, from which we calculate the loss and optimize the gradient as described in Section 2.2.

## 2.2 Training

**Training Objective.** Given the true labels, the predicted probabilities  $\hat{p}$ , and the contact maps  $\hat{C}$ , we compute the loss as  $\lambda L^{BCE} + (1 - \lambda)L^{MAG}$ ; here  $\lambda$  is a hyper-parameter that balances between  $L^{BCE}$ , the



binary cross-entropy (BCE) loss, and  $L^{MAG}$ , the contact-map magnitude loss (MAG). While the BCE loss is standard in a classification context, we introduce  $L^{MAG}$  as a novel regularization that enables us to learn realistically sparse contact maps.  $L^{MAG}$  for a single training example is calculated as the arithmetic mean value of the contact map  $\hat{C}$ . Jointly minimizing the total magnitude of contact maps with the BCE captures the intuition that interacting proteins are characterized by just a few high probability inter-protein contacts, while most residues will not be in contact.

**Implementation Details.** We implemented D-SCRIPT in PyTorch 1.2.0 and trained with a NVIDIA Tesla V100 with 32GB of memory. Embeddings from the pre-trained Bepler & Berger model were produced by concatenating the final values of the output and all hidden layers, so that  $d_0 = 6165$ . We used a projection dimension of  $d = 100$ , a hidden dimension of  $h = 50$ , a convolutional filter with width  $2w + 1 = 7$ , and a local max-pooling width of  $l = 9$ . We used  $x_0 = 0.5, \eta = 20$  for the custom logistic activation, and  $\lambda = 0.35$  for calculating the training loss. Weights were initialized using PyTorch defaults. We used a batch size of 25, the Adam optimizer with a learning rate of 0.001, and trained all models for 10 epochs.

### 3 Results

**PPI Data Set.** To evaluate the performance of D-SCRIPT in predicting protein-protein interactions, we use data from the STRING [26] database (version 11). STRING contains protein pairs corresponding to a variety of primary sources and interaction modalities (e.g., binding vs co-expression). In order to select only high-confidence physical protein interactions, we limited our positive examples to binding interactions associated with a positive experimental-evidence score. From this set, we removed PPIs involving very short proteins (shorter than 50 amino acids) and, due to GPU memory constraints, also excluded proteins longer than 800 amino acids. Next, we removed PPIs with high sequence redundancy to other PPIs, following the precedent of previous approaches [17, 20]. Specifically, we clustered proteins at the 40% similarity threshold using CD-HIT [35, 36], and a PPI (A-B) was considered sequence redundant (and excluded) if we had already selected another PPI (C-D) such that the protein pairs (A, C) and (B, D) each shared a CD-HIT cluster. Removing sequence redundant PPIs from the data set prevents the model from memorizing interactions based on sequence similarity alone. To generate negative examples of PPI, we followed [20] and randomly paired proteins from the non-redundant set, choosing a 10:1 negative-to-positive ratio to reflect the intuition that true positive PPIs are likely rare. Our human PPI data set contained 47,932 positive and 479,320 negative protein interactions, of which we set apart 80% (38,345) for training and 20% (9,587) for validation. For each of 5 model organisms (Table 1) we selected 5,000 positive interactions and 50,000 negative interactions using this procedure, with the exception of *E.coli* (2,000/20,000) where the available set of positive examples in STRING was limited.

**D-SCRIPT Excels at Cross-Species PPI Prediction.** We first sought to see how D-SCRIPT performed on the task of cross-species interaction prediction. We trained a model on human PPIs and evaluated it using PPI data sets from five other model organisms. We compared D-SCRIPT with PIPR, shown by Chen et al. [17] to be the best performing sequence-based PPI prediction method, trained on the same set of human PPIs as the D-SCRIPT model. We compare model complexity of D-SCRIPT and PIPR in Appendix A.2. We also compare to a recently released method by Richoux et al. [37] in Appendix A.3. Additionally, we compared with a hybrid method (PIPR + D-SCRIPT) where PIPR was used to adjust D-SCRIPT’s prediction: when PIPR is highly confident that an interaction *does* occur ( $\hat{p} > 0.9$ ), the predicted probability of D-SCRIPT is increased by 50%. In Table 1, we report the precision, recall, area under precision-recall curve (AUPR)

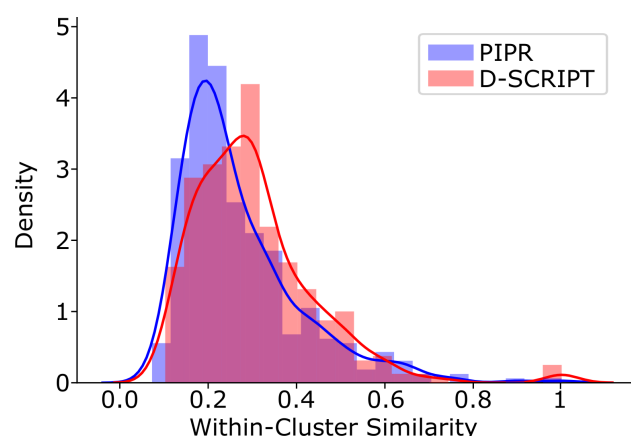
and area under ROC curve (AUROC) of each method in each of five species. For highly unbalanced data, as is the case here, we note that AUPR is generally considered a better metric than AUROC. D-SCRIPT significantly outperforms PIPR cross-species and maintains a high AUPR across all species, even those which are extremely evolutionary distant from human. In fact, its AUPR in these species remains comparable to that seen in human cross-validation. The hybrid method outperforms both D-SCRIPT and PIPR alone, but improves D-SCRIPT only modestly in cross-species analysis.

**D-SCRIPT Hybrid Performs Well on Human Cross Validation.** While our aim is enhanced cross-species PPI prediction, we sought to investigate how D-SCRIPT would perform at predicting within-species interactions in Human. We performed 5-fold cross validation and report here the average across all folds. Additionally, we evaluated a hybrid method (PIPR + D-SCRIPT) where we used D-SCRIPT to adjust PIPR’s prediction: when D-SCRIPT is highly confident that an interaction does *not* happen ( $\hat{p} < 0.01$ ), we reduce the predicted probability from PIPR by half. Table 1 shows that while PIPR significantly outperforms D-SCRIPT on human PPIs in cross-validation, a combination of the methods outperforms either one alone. Notably, the PIPR + D-SCRIPT hybrid achieves substantially higher precision, though at the expense of recall. This may be a desirable trade-off in certain contexts, e.g., when generating PPI candidates for experimental validation. We further investigated the performance of D-SCRIPT and PIPR on subsets of the human data, seeking to better understand their relative strengths. As we elaborate in Appendix A.4, D-SCRIPT performs better on interactions involving proteins that occur infrequently in the PPI network while PIPR performs better on proteins that occur frequently. This suggests that PIPR may perform better when a large amount of training data is available but D-SCRIPT may generalize better to new proteins (and species).

**Table 1: Evaluation of models trained on human PPIs.** *H. sapiens* results are average performance over 5-fold cross validation. All other species were evaluated using a model trained on human data. D-SCRIPT significantly outperforms PIPR cross-species, despite PIPR performing better in human cross-validation. In both cases, a hybrid model is able to achieve better performance than either model alone.

Species	Model	AUPR	Precision	Recall	AUROC
<i>M. musculus</i>	PIPR	0.526	0.734	0.331	<b>0.839</b>
	D-SCRIPT	0.580	0.818	0.346	0.833
	PIPR + D-SCRIPT	<b>0.609</b>	<b>0.820</b>	<b>0.355</b>	0.838
<i>D. melanogaster</i>	PIPR	0.278	0.521	0.121	0.728
	D-SCRIPT	0.552	<b>0.798</b>	0.359	<b>0.824</b>
	PIPR + D-SCRIPT	<b>0.562</b>	<b>0.798</b>	<b>0.361</b>	<b>0.824</b>
<i>C. elegans</i>	PIPR	0.346	0.673	0.142	0.757
	D-SCRIPT	0.548	0.840	0.306	0.813
	PIPR + D-SCRIPT	<b>0.559</b>	<b>0.841</b>	<b>0.308</b>	<b>0.814</b>
<i>S. cerevisiae</i>	PIPR	0.230	0.398	0.085	0.718
	D-SCRIPT	0.405	0.706	0.223	<b>0.789</b>
	PIPR + D-SCRIPT	<b>0.417</b>	<b>0.708</b>	<b>0.225</b>	<b>0.789</b>
<i>E. coli</i>	PIPR	0.308	0.629	0.131	0.675
	D-SCRIPT	0.571	0.791	<b>0.520</b>	<b>0.863</b>
	PIPR + D-SCRIPT	<b>0.588</b>	<b>0.793</b>	0.394	<b>0.863</b>
<i>H. sapiens</i> (5-fold cross validation)	PIPR	0.835	0.838	<b>0.701</b>	0.960
	D-SCRIPT	0.516	0.728	0.278	0.833
	PIPR + D-SCRIPT	<b>0.844</b>	<b>0.949</b>	0.400	<b>0.962</b>





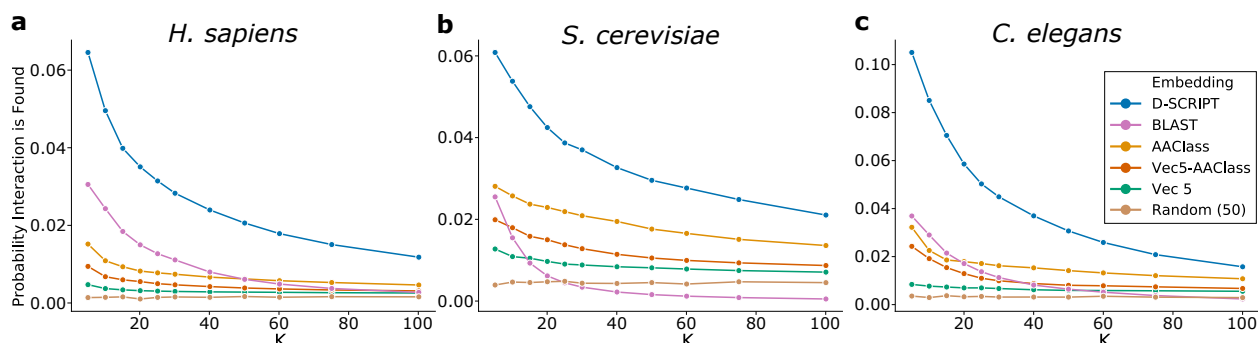
**Figure 3: Distribution of within-cluster similarity scores in *Drosophila melanogaster*.** D-SCRIPT recovers more functionally coherent clusters than PIPR ( $p = 0.000723$ , one-tailed  $t$ -test). Clusters were generated by predicting protein interactions using either D-SCRIPT or PIPR, computing the diffusion state distance (DSD) between proteins, and clustering the DSD matrix using spectral clustering. Within-cluster similarity was calculated as the average Jaccard similarity between GO Slim annotations of all pairs of proteins in the cluster.

**Functional Module Prediction.** The importance of PPI networks arises, in part, from the graph-theoretic analyses on them which enable the functional characterization of un-annotated proteins. We therefore sought to test if D-SCRIPT’s success at cross-species generalization would translate to better functional inference in new species. In particular, we hypothesized that, compared to PIPR, the D-SCRIPT model trained on human data should facilitate more accurate inference of protein functional modules in *Drosophila*. Towards this end, we generated a set of 10,475,595 candidate pairs from the set of *Drosophila* proteins in STRING. Using D-SCRIPT and PIPR’s human-trained models, we predicted interactions over this candidate set. Module detection on the resulting PPI networks was performed by spectral clustering, with pairwise distances between proteins assessed using Cao et al.’s [38, 39] diffusion state distance (DSD) metric. This module detection approach performed well in a recent DREAM challenge on functional module detection [5].

We tested the relative functional coherence of 374 (PIPR) and 384 (D-SCRIPT) clusters using available GO (Gene Ontology) annotations from FlyBase [40], filtering out electronically-inferred and homology-based annotations. All GO terms were mapped to a limited set of GO Slim terms using the *D. Melanogaster* species-specific list [41]. For each cluster, we computed the within-cluster functional similarity, calculated as the mean Jaccard similarity of the sets of GO Slim annotations for all pairs of proteins within the cluster. We also used a different, graph-theoretic measure of protein similarity based on GO terms [42]; it produced similar results (Appendix A.5). Figure 3 shows that the average within-cluster similarity when interactions are predicted using D-SCRIPT is significantly higher than when using PIPR ( $p = 0.000723$ , one-tailed  $t$ -test), and that D-SCRIPT results in 24% more highly-enriched (top 10% of scores) clusters as PIPR.

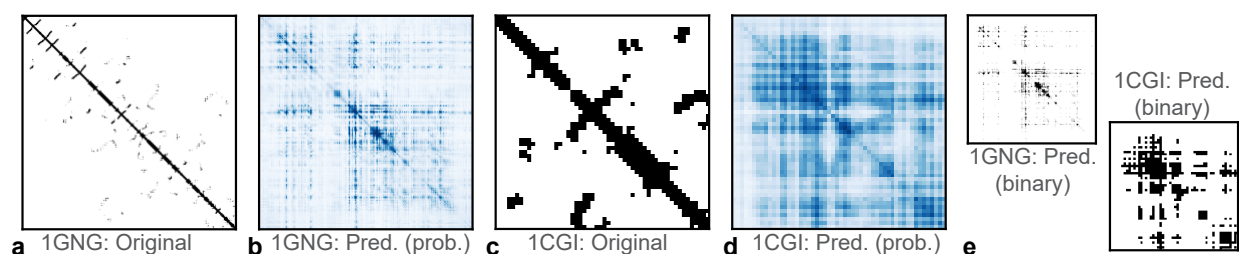
**Preliminary prediction of viral-host PPIs between SARS-Cov-2 & Human.** We applied D-SCRIPT and PIPR to computationally screen all SARS-Cov-2 proteins against 19,777 human proteins, predicting approximately 3,000 viral-host PPIs from each method (Appendix A.6). We then characterized each viral protein’s function by the GO annotations of its human interactors. Compared to the corresponding annotations derived from 332 experimentally-determined PPIs (Gordon et al. [43]), we found D-SCRIPT based annotations to overlap more with the experimental results than those from PIPR ( $p = 0.059$ , paired one-tailed  $t$ -test). We have made these predicted PPIs available for the community’s use (see A.6), but note that these are preliminary results and need further analysis.

**Comparison of D-SCRIPT Embedding with Other Embeddings.** To compare our sequence embedding method with other potential ones, we evaluated various protein sequence embeddings under the following framework: the Euclidean distance in an embedding space was used to define a distance measure between proteins. Given a true positive PPI ( $A, B$ ), we applied this distance measure to identify the  $k$ -nearest neighbors of  $A$  and  $B$  each, and computed how many of the  $k^2$  possible combinations of these neighbors corresponded to a positive PPI. For D-SCRIPT, we used the  $\mathcal{R}_{\geq 0}^{n \times d}$  output of the projection module (Section 2.1.2) and averaged the features across the length of the protein to obtain a  $d$ -dimensional embedding. We compared with a one-hot embedding categorizing each amino acid into one of seven classes based on biochemical properties (“AAClass”, from [44]), a 5-residue-context Skip-Gram embedding (“Vec5”, from [45]), a concatenation of Vec5 and AAClass (used in the input for PIPR), and a randomly generated 50-dimensional embedding with values drawn uniformly from the range  $[0, 1]$ . Additionally, we used BLAST [46] to select  $k$  neighbors. Figure 4 shows that D-SCRIPT significantly outperforms all other embeddings.



**Figure 4: Neighborhood interaction search using protein embeddings.** Performance comparison of D-SCRIPT projection module (Section 2.1.2) trained on human data with several other embedding methods on recovering true interacting protein pairs in the neighborhood of known PPIs in human (a), yeast (b), and roundworm (c). D-SCRIPT embeddings recover significantly more interacting proteins than any other embedding, regardless of species or number of neighbors checked. AAClass also performs well, likely because it characterizes biochemistry which is preserved at longer evolutionary distances. BLAST performs well at low values of  $k$  but has difficulty recovering interactions for larger values — likely due to network rewiring over longer evolutionary distances.

**Prediction of Intra-Protein Self Contacts.** One of our aims when designing D-SCRIPT was to capture the structural aspects of interaction — the per-protein embedding produced by the trained projection module (Section 2.1.2) should encode structural information. To examine this aspect, we randomly selected 300 proteins from the Protein Data Bank (PDB) and generated  $(n \times d)$ -dimensional embeddings of these proteins using the D-SCRIPT model trained on human PPIs. We assessed intra-protein contacts at 8 Å in the PDB structure and converted each protein’s contact map to a binary classification data set: a protein sequence of length  $n$  corresponded to  $\frac{n(n-1)}{2}$  observations, with the observation  $ij$  corresponding to a putative contact between residues  $i$  and  $j$ . We randomly split the 300 PDB structures into a training set of 100 and a test set of 200, evaluating how well a logistic regression that uses the D-SCRIPT embeddings as the input predicts contacts between residues. The regression was  $L_2$ -regularized and class balanced, with its input for observation  $ij$  being the concatenation of the  $d$ -dimensional embeddings  $Z_i$  and  $Z_j$  output by the projection module as well as their combinations  $\text{diff}_{ij}$ ,  $\text{mul}_{ij}$  as defined in Equations 2 & 3. Figure 5 shows that a linear combination of features in the projection module output is able to recapitulate a significant subset of the ground-truth contacts, achieving a median per-structure AUPR of 0.19 over the test data set. These results strongly suggest that the end-to-end training of D-SCRIPT — using only sequence data — results in

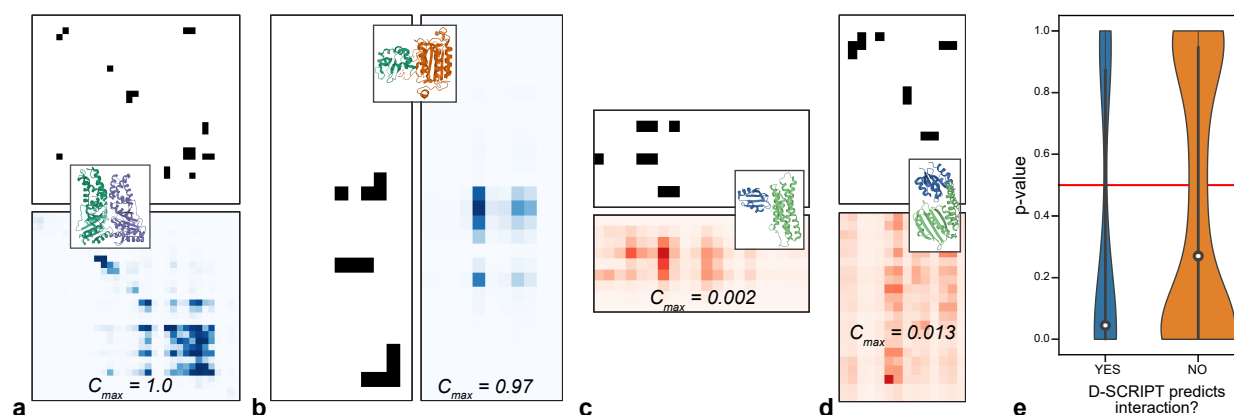


**Figure 5: Self-contact prediction using D-SCRIPT embeddings.** The PDB identifier 1GNG corresponds to a protein with 356 residues where the D-SCRIPT embedding’s self-contact prediction accuracy is near the median of cases we studied (AUPR=0.19), while 1CGI corresponds to a short protein (54 residues) in which the embedding achieves a higher accuracy (AUPR=0.38). The original contacts (a, c) were assessed at 8 Å. Contacts were predicted (b, d) using a logistic regression model trained on contacts seen in a held-out set of proteins, and the binarization thresholds for panel (e) were chosen so as to result in the same number of contacts as in the original maps.

an intermediate representation that captures structural information at the level of each protein.

**Prediction of Inter-Protein Docking Contacts.** We investigated whether the interpretability of our model could aid in predicting inter-protein docking contacts. The output of the Residue Contact Module (Section 2.1.3) of D-SCRIPT is an inter-protein contact map  $\hat{C}$  where  $\hat{C}_{ij} \in [0, 1]$  can be interpreted as the probability of residue  $i$  from protein  $S_1$  being in contact with residue  $j$  of protein  $S_2$ . We verified the contact maps produced after training were consistent with our design goal: the maps corresponding to negative predictions should have uniformly near-zero  $\hat{C}_{ij}$  scores, while those for positive predictions should be sparse but with isolated regions of high  $\hat{C}_{ij}$  scores. We found that this was indeed the case generally and show some examples in Figure 6: the maximum  $\hat{C}$  value is high for positive examples and low for negative examples.

We next sought to test if  $\hat{C}$  is physically representative of the actual docking mechanism of the interaction. We emphasize that this is a high bar given we do not provide any 3-D information to the model nor any guidance on docking and, in principle, the model could perform well on the classification task without  $\hat{C}$  being physically accurate. We performed this test using Hwang et al.’s benchmark data set of docked protein structures [18]. For every pair of chains in each PDB complex in the benchmark set, we generated a candidate PPI. We applied our human-data-trained model on 295 candidate PPIs and evaluated the predicted contact maps against the ground-truth contacts (assessed at 8 Å). In cases where our model predicted an interaction, we found  $\hat{C}$  to indeed recapitulate the ground-truth contacts substantially (Figure 6 a,b). Even in some of the cases where D-SCRIPT did not predict an interaction, the distribution of  $\hat{C}_{ij}$  scores was nevertheless consistent with the ground-truth (Figure 6 c). To systematically evaluate  $\hat{C}$ ’s accuracy, we interpreted ground truth and predicted contact maps as probability distributions over the  $n \times m$  matrix and measured the 2-D Earth mover’s distance between these distributions, computed by solving an optimal transport problem under the Euclidean metric [47]. We chose this metric to measure similarity between regions of the two contact maps rather than measuring per-residue matches (with a metric such as binary cross-entropy) because the convolutional and max-pooling layers in our model aggregate over neighboring residues, thus diffusing the signal. For each candidate PPI, we established random baselines by shuffling  $\hat{C}$  and recomputing the Earth mover’s distance. We estimated the p-value of the actual  $\hat{C}$  against 500 random trials, finding that in cases where D-SCRIPT predicted an interaction, the  $\hat{C}$  contact-maps were significantly similar to the ground-truth (median FDR-corrected  $p = 0.08$ ). Even in cases where D-SCRIPT did not predict an interaction,  $\hat{C}$ ’s similarity to the ground truth was higher than that of the random baselines.



**Figure 6: When D-ScripT correctly predicts an interaction, its contact maps  $\hat{C}$  are significantly similar to the ground truth.** We show inter-protein contact maps of protein structures known to dock together. Panels (a,b) correspond to pairs where D-ScripT correctly predicted an interaction, while panels (c,d) are cases where it incorrectly predicted no interaction. The black-and-white matrices correspond to the PDB ground truth while the colored matrices correspond to D-ScripT's  $\hat{C}$ ; for the latter, the color scales of (a,b) differ from (c,d). While  $\hat{C}$  contains large values for positive pairs, its maximum  $C_{max}$  is very low for negative pairs. Panel (e) shows a systematic evaluation of the 2-D Earth mover's distance-based similarity between  $\hat{C}$  and the ground truth. Not only are the correctly-predicted  $\hat{C}$ 's significantly similar to the ground truth, even when D-ScripT incorrectly predicts two proteins don't interact, its contact maps are still similar to ground truth.

**Performance.** D-ScripT took approximately 3 days to train for 10 epochs on 843,602 training pairs, and fits within a single 32GB GPU. Running time and GPU memory usage scales roughly quadratically,  $O(mn)$ , with the protein lengths  $m, n$ , since D-ScripT models the full  $n \times m$  contact map as an intermediate step. Prediction of new candidate pairs with a trained model is very fast, requiring on average 0.13 seconds/pair and less than 5GB of GPU memory. Since D-ScripT generalizes well cross-species, it only needs to be trained once on a large corpus of data, and can be used to make predictions in a variety of settings.

## 4 Discussion

We have introduced D-ScripT, an interpretable method for PPI interaction prediction from sequence. It joins the small but growing set of advances in interpretable deep learning methods in computational biology [48, 49]. We showed that its predictions generalize better than other methods to PPIs for which there are fewer training examples of interactions with the constituent proteins, and importantly, to a cross-species setting where the model is only trained on protein sequences from a different species. This is particularly exciting for organisms such as *C. elegans*, for which the known portion of the PPI network is still quite sparse. In organisms like *D. melanogaster* where some PPI data does exist, future work could productively combine that data with D-ScripT models trained on human data. We suspect that D-ScripT's relative success across species, but under-performance on a within-species evaluation, is due to the simplicity of the model and the extent to which it is regularized. These design choices enhance D-ScripT's generalizability, directing it to learn general structural aspects of the interaction, rather than using network structure or the frequency of any individual protein as an interaction partner. However, for certain tasks a balance between the cross-species generalizability of D-ScripT and the within-species specificity of other state-of-the-art methods may be desirable. A future research direction might be transfer learning to tune a pre-trained D-ScripT model to a target species, while another approach could be to integrate it with graph-theoretic PPI predictions. Insights from recent advances in the prediction of contact maps and structures of individual

proteins could also be incorporated into our model architecture. D-SCRIPT illustrates that learning the language of individual proteins, a remarkably successful deep learning effort, also helps decode the language of protein interactions.

**Acknowledgements** We thank Tristan Bepler for helpful discussions and technical assistance. SS, RS and BB were supported by the NIH grant R01 GM081871. LC was supported by NSF HDR grant 1939263.

## References

- [1] Sharan, R., Ulitsky, I. & Shamir, R. Network-based prediction of protein function. *Molecular systems biology* **3**, 88 (2007).
- [2] Navlakha, S. & Kingsford, C. The power of protein interaction networks for associating genes with diseases. *Bioinformatics* **26**, 1057–1063 (2010).
- [3] Cho, H., Berger, B. & Peng, J. Compact integration of multi-network topology for functional analysis of genes. *Cell systems* **3**, 540–548 (2016).
- [4] Cowen, L., Ideker, T., Raphael, B. J. & Sharan, R. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics* **18**, 551 (2017).
- [5] Choobdar, S. *et al.* Assessment of network module identification across complex diseases. *Nature methods* **16**, 843–852 (2019).
- [6] Fields, S. & Song, O.-k. A novel genetic system to detect protein–protein interactions. *Nature* **340**, 245–246 (1989).
- [7] Krogan, N. J. *et al.* Global landscape of protein complexes in the yeast *saccharomyces cerevisiae*. *Nature* **440**, 637–643 (2006).
- [8] Kumar, A. & Snyder, M. Protein complexes take the bait. *Nature* **415**, 123–124 (2002).
- [9] Sahni, N. *et al.* Widespread macromolecular interaction perturbations in human genetic disorders. *Cell* **161**, 647–660 (2015).
- [10] Taipale, M. *et al.* A quantitative chaperone interaction network reveals the architecture of cellular protein homeostasis pathways. *Cell* **158**, 434–448 (2014).
- [11] Menche, J. *et al.* Uncovering disease-disease relationships through the incomplete interactome. *Science* **347** (2015).
- [12] Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. In *7th International Conference on Learning Representations, ICLR 2019* (2019).
- [13] Ingles-Prieto, A. *et al.* Conservation of protein structure over four billion years. *Structure* **21**, 1690–1697 (2013).
- [14] Singh, R., Xu, J. & Berger, B. Struct2net: integrating structure into protein-protein interaction prediction. In *Biocomputing 2006*, 403–414 (World Scientific, 2006).



- [15] Singh, R., Park, D., Xu, J., Hosur, R. & Berger, B. Struct2net: a web service to predict protein–protein interactions using a structure-based approach. *Nucleic acids research* **38**, W508–W515 (2010).
- [16] Hosur, R., Xu, J., Bienkowska, J. & Berger, B. iwrap: an interface threading approach with application to prediction of cancer-related protein–protein interactions. *Journal of molecular biology* **405**, 1295–1310 (2011).
- [17] Chen, M. *et al.* Multifaceted protein-protein interaction prediction based on Siamese residual RCNN. In *Bioinformatics*, vol. 35, i305–i314 (Oxford University Press, 2019). URL <https://academic.oup.com/bioinformatics/article/35/14/i305/5529260>.
- [18] Hwang, H., Vreven, T., Janin, J. & Weng, Z. Protein–protein docking benchmark version 4.0. *Proteins: Structure, Function, and Bioinformatics* **78**, 3111–3114 (2010).
- [19] Hosur, R. *et al.* A computational framework for boosting confidence in high-throughput protein-protein interaction datasets. *Genome biology* **13**, 1–14 (2012).
- [20] Hashemifar, S., Neyshabur, B., Khan, A. A. & Xu, J. Predicting protein-protein interactions through sequence-based deep learning. In *Bioinformatics*, vol. 34, i802–i810 (Oxford University Press, 2018). URL <https://academic.oup.com/bioinformatics/article/34/17/i802/5093239>.
- [21] Lei, C. & Ruan, J. A novel link prediction algorithm for reconstructing protein–protein interaction networks by topological similarity. *Bioinformatics* **29**, 355–364 (2013).
- [22] Hulovatyy, Y., Solava, R. W. & Milenković, T. Revealing missing parts of the interactome via link prediction. *PloS one* **9**, e90073 (2014).
- [23] Kovács, I. A. *et al.* Network-based prediction of protein interactions. *Nature communications* **10**, 1–8 (2019).
- [24] Devkota, K., Murphy, J. M. & Cowen, L. J. Glide: combining local methods and diffusion state embeddings to predict missing interactions in biological networks. *Bioinformatics* **36**, i464–i473 (2020).
- [25] Franz, M. *et al.* Genemania update 2018. *Nucleic acids research* **46**, W60–W64 (2018).
- [26] Szklarczyk, D. *et al.* String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research* **47**, D607–D613 (2019).
- [27] Wang, B. *et al.* Network enhancement as a general method to denoise weighted biological networks. *Nature communications* **9**, 1–8 (2018).
- [28] Khurana, V. *et al.* Genome-scale networks link neurodegenerative disease genes to  $\alpha$ -synuclein through specific molecular pathways. *Cell systems* **4**, 157–170 (2017).
- [29] Liu, Y., Palmedo, P., Ye, Q., Berger, B. & Peng, J. Enhancing evolutionary couplings with deep convolutional neural networks. *Cell systems* **6**, 65–74 (2018).
- [30] Jones, D. T. & Kandathil, S. M. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics* **34**, 3308–3315 (2018).



- [31] Xu, J. Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences* **116**, 16856–16865 (2019).
- [32] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv* (2020).
- [33] Luo, Y. *et al.* Evolutionary context-integrated deep sequence modeling for protein engineering. In *International Conference on Research in Computational Molecular Biology*, 261–263 (Springer, 2020).
- [34] Tai, K. S., Socher, R. & Manning, C. D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* (2015).
- [35] Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658–1659 (2006).
- [36] Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).
- [37] Richoux, F., Servantie, C., Borès, C. & Téletchéa, S. Comparing two deep learning sequence-based models for protein-protein interaction prediction. *bioRxiv* (2019). URL <http://arxiv.org/abs/1901.06268>. 1901.06268.
- [38] Cao, M. *et al.* Going the distance for protein function prediction: a new distance metric for protein interaction networks. *PloS one* **8**, e76339 (2013).
- [39] Cao, M. *et al.* New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. *Bioinformatics* **30**, i219–i227 (2014).
- [40] Thurmond, J. *et al.* Flybase 2.0: the next generation. *Nucleic acids research* **47**, D759–D765 (2019).
- [41] Adams, M. D. *et al.* The genome sequence of drosophila melanogaster. *Science* **287**, 2185–2195 (2000).
- [42] Zhao, C. & Wang, Z. Gogo: An improved algorithm to measure the semantic similarity between gene ontology terms. *Scientific reports* **8**, 1–10 (2018).
- [43] Gordon, D. E. *et al.* A sars-cov-2 protein interaction map reveals targets for drug repurposing. *Nature* 1–13 (2020).
- [44] Shen, J. *et al.* Predicting protein–protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences* **104**, 4337–4341 (2007).
- [45] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119 (2013).
- [46] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of molecular biology* **215**, 403–410 (1990).
- [47] Flamary, R. & Courty, N. Pot python optimal transport library (2017). URL <https://pythonot.github.io/>.

- [48] Luo, Y., Peng, J. & Ma, J. When causal inference meets deep learning. *Nature Machine Intelligence* **2**, 426–427 (2020).
- [49] Hie, B., Bryson, B. D. & Berger, B. A. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Systems* (2020).
- [50] Yu, G., Wang, L.-G., Han, Y. & He, Q.-Y. clusterprofiler: an r package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology* **16**, 284–287 (2012).
- [51] Mutowo, P. *et al.* A drug target slim: using gene ontology and gene ontology annotations to navigate protein-ligand target space in chembl. *Journal of biomedical semantics* **7**, 59 (2016).

## A Appendix

### A.1 The Corpus of Experimental PPI Data is Limited

We sourced the PPI data in Figure 1 from BioGRID. While we have sourced PPI data from the STRING database everywhere else in this paper, here we chose to use BioGRID because the publication date of a PPI is easily accessible in BioGRID, allowing us to estimate the number of PPIs assayed in the last five years. While the BioGRID selection may not precisely match the STRING selection due to curation differences between the two databases, our primary aim here is conveying the relative data availability across species; this estimate should not be significantly impacted by differences in curation.

### A.2 Comparison of Model Complexities of D-SCRIPT and PIPR

The version of PIPR that we compare to has 72,500 trainable parameters. The full D-SCRIPT model has 629,207 trainable parameters, but the vast majority of those (616,600) are in the projection module, which is a simple linear combination of all concatenated hidden states of the Bepler & Berger language model (which are themselves already redundant). The remaining stages of the model together have 12,707 trainable parameters. Practically, we prevent over-fitting through a high rate of dropout (50%) in the Projection module (Section 2.1.2), combined with a very simple but structurally informed architecture in the Contact (Section 2.1.3) and Interaction (Section 2.1.4) prediction modules. Additionally, the contact map magnitude loss (Section 2.2) acts as a regularization by requiring that the predicted contact maps be sparse. Empirically, D-SCRIPT generalizes much better than PIPR, and does not seem to overfit to the training data.

### A.3 Comparison with Another Sequence-based PPI Prediction Method

We also compared our method to a model described by Richoux et al. [37]. In it, they trained a multi-layer fully connected network to predict protein interaction given two input sequences. The results comparing this model to D-SCRIPT, presented in Table A.1, are organized like those presented in Table 1 in the main text. Similarly to Chen et al.’s PIPR model, the Richoux et al. model performs better than D-SCRIPT when trained and tested on human PPIs while D-SCRIPT generalizes better across species, with substantially higher AUPRs than the Richoux model when the human-data-trained model is evaluated on other species. “Richoux + D-SCRIPT” refers to the same hybrid combination described in Section 3.

### A.4 Deeper Investigation of D-SCRIPT and PIPR’s Relative Strengths

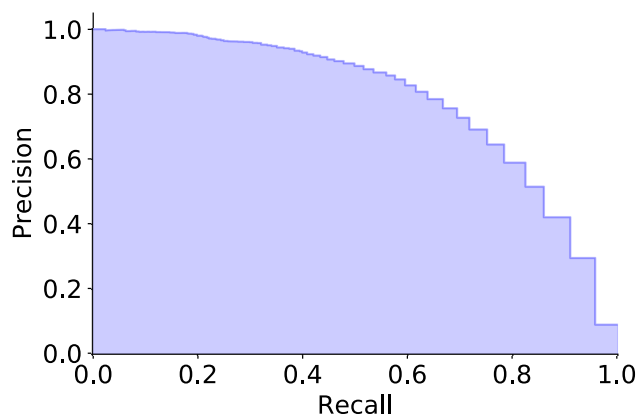
We investigated the relative performance of D-SCRIPT and PIPR on subsets of human proteins, seeking to better understand their relative strengths.

#### Using protein frequency counts as a predictor

PPI networks are dominated by a dense and highly connected core of nodes, where the core includes many hub nodes. A method that does relatively well on these hub proteins will have an advantage on within-species evaluations. For our cross-validation setup, we followed the precedent of previous work and split training and test data by edges [17, 20]. Consequently, a protein’s relative frequency will be roughly similar in the training and test data sets. In this context, a PPI prediction heuristic that simply remembers frequency counts does strikingly well (see figure below). Here, we scored the likelihood of a candidate PPI ( $A, B$ ) being true as the minimum of frequency counts, in training data, of  $A$  and  $B$ .

**Table A.1: Evaluation of models trained on human PPIs.** We compare D-SCRIPT to the fully connected model described in Richoux et al. by a 5-fold cross validation in human, and the application of a human-data trained model to PPIs from 5 other species. As with the comparison to PIPR (see Table 1), while Richoux outperforms PIPR in the cross validation setting, D-SCRIPT generalizes significantly better to unseen proteins cross-species, and a hybrid model outperforms either model alone.

Species	Model	Precision	Recall	AUPR	AUROC
<i>M. musculus</i>	Richoux	0.749	0.321	0.518	<b>0.816</b>
	D-SCRIPT	0.818	0.346	0.580	0.833
	Richoux + D-SCRIPT	<b>0.821</b>	<b>0.354</b>	<b>0.608</b>	0.837
<i>D. melanogaster</i>	Richoux	0.582	0.126	0.231	0.659
	D-SCRIPT	0.798	0.359	0.552	0.824
	Richoux + D-SCRIPT	<b>0.799</b>	<b>0.360</b>	<b>0.559</b>	<b>0.824</b>
<i>C. elegans</i>	Richoux	0.672	0.079	0.252	0.671
	D-SCRIPT	0.840	0.306	0.548	0.813
	Richoux + D-SCRIPT	<b>0.841</b>	<b>0.307</b>	<b>0.554</b>	<b>0.814</b>
<i>S. cerevisiae</i>	Richoux	0.501	0.059	0.201	0.652
	D-SCRIPT	0.706	0.223	0.405	0.789
	Richoux + D-SCRIPT	<b>0.707</b>	<b>0.224</b>	<b>0.413</b>	<b>0.789</b>
<i>E. coli</i>	Richoux	0.746	0.093	0.303	0.687
	D-SCRIPT	0.791	0.520	0.571	0.863
	Richoux + D-SCRIPT	<b>0.792</b>	<b>0.391</b>	<b>0.581</b>	<b>0.863</b>
<i>H. sapiens</i> (5-fold cross validation)	Richoux	0.815	<b>0.737</b>	0.834	0.964
	D-SCRIPT	0.728	0.278	0.516	0.833
	Richoux + D-SCRIPT	<b>0.944</b>	0.405	<b>0.848</b>	<b>0.966</b>



**Figure A.1: Performance of a training-counts-only classifier:** As part of the investigation of the impact of a protein's frequency in the training set on classifier performance, we created a naive classifier which simply predicts that a pair interacts with probability corresponding to the minimum number of times either of a pair exists in a positive interaction in the training set, normalized over the maximum number of times any such protein appears. We find that this naive classifier actually performs rather well in within-species cross-validation, achieving an AUPR of 0.784. However, such a classifier would output a probability of 0 for all interactions in a data set which did not share any proteins with the training set, as is the case in the cross-species setting.

## D-SCRIPT performs better on infrequently occurring proteins

To further investigate the finding above, we ranked proteins in the human PPI network by their frequency of occurrence. For a set of quantiles  $q \in [0, 1]$ , we evaluated out-of-sample D-SCRIPT and PIPR predictions on the human PPI sub-network consisting only of proteins of rank  $q$  or lower; here, lower  $q$  corresponds to

a lower frequency of occurrence. In absolute terms, as the table below indicates, both D-SCRIPT and PIPR become more accurate at higher  $q$ . However, D-SCRIPT has a relative advantage at lower  $q$  (i.e., infrequently occurring proteins) while PIPR performs better at higher  $q$ . In other words, PIPR’s better within-species performance can be traced to it being more accurate on proteins that occur frequently. This also suggests an explanation for PIPR’s lower cross-species generalizability than D-SCRIPT: when making predictions on an entirely new set of proteins in a different species, knowing the relative frequencies of proteins in the training data might not be particularly useful.

The difference between D-SCRIPT and PIPR might stem from their respective architectures. The protein representation learned by D-SCRIPT is constrained to be a linear projection of the Bepler & Berger pre-trained embedding, albeit with ReLU and dropout layers. This regularizes how much frequency information can be incorporated into the model; we note that the Bepler and Berger model was trained with data on individual proteins and would not reflect PPI frequency information. In contrast, PIPR’s design allows for a lot more leeway in training each protein’s representation. This flexibility may allow PIPR to better incorporate the occurrence frequencies into its representation, helping its within-species performance but potentially hurting its cross-species generalizability.

**Table A.2: Area under precision-recall curve (AUPR) of PPI prediction performance for interactions where proteins appear rarely in the training set.** We created increasingly restrictive subsets of the test data, where no protein in the subset may appear more than some cutoff number of times in the training set. While PIPR outperforms D-SCRIPT when frequently trained on proteins are present, at lower quantile values  $q$ , D-SCRIPT performs relatively better than PIPR - although the absolute performance of both classifiers drops off as the data set is more challenging.

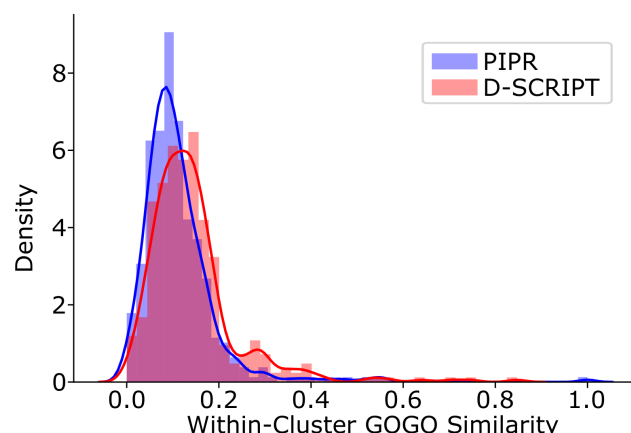
Cutoff Quantile ( $q$ )	Cutoff Value	% of Test Pairs	PIPR	D-SCRIPT
1.0	209	100	<b>0.830503</b>	0.591353
0.9	12	75.31	<b>0.206161</b>	0.118704
0.8	5	61.12	0.061128	<b>0.068445</b>
0.7	2	46.85	0.014476	<b>0.024299</b>
0.6	1	38.03	0.005838	<b>0.011789</b>
0.5	0	25.06	0.001373	<b>0.006673</b>

## A.5 Functional Enrichment Using GOGO Distance as the GO Term Similarity Metric

To compute within-cluster functional similarity in the main text, we took the simple but robust approach of mapping GO terms to a limited set of GO Slim terms and then computing Jaccard similarity across the latter (Section 3). In Figure A.2, we used a recently introduced approach from the literature, Zhao et al.’s GOGO algorithm [42], to compute the similarity between pairs of GO terms; we then aggregate these to get within-cluster similarity scores. Our evaluation here yields similar results as presented in the main text: D-SCRIPT produces clusters with higher functional similarity than PIPR.

## A.6 GO Enrichment of SARS-CoV-2 Proteins

We performed a preliminary study to predict viral-host interactions between SARS-CoV-2 and human proteins wherein we compared the sets of over-represented GO terms for human interactors of SARS-CoV-2 proteins, as predicted by D-SCRIPT or PIPR, with those over-represented in the experimentally-determined human interactors (Gordon et al. [43]). Figure A.3 shows the relative similarity of computationally predicted annotations to the experimentally-determined annotations for each SARS-CoV-2 protein. Overall, we found that sets of enriched terms computed using the D-SCRIPT network overlap slightly more with the true network than those computed using the PIPR network ( $p = 0.059$ ). Among the putative accessory



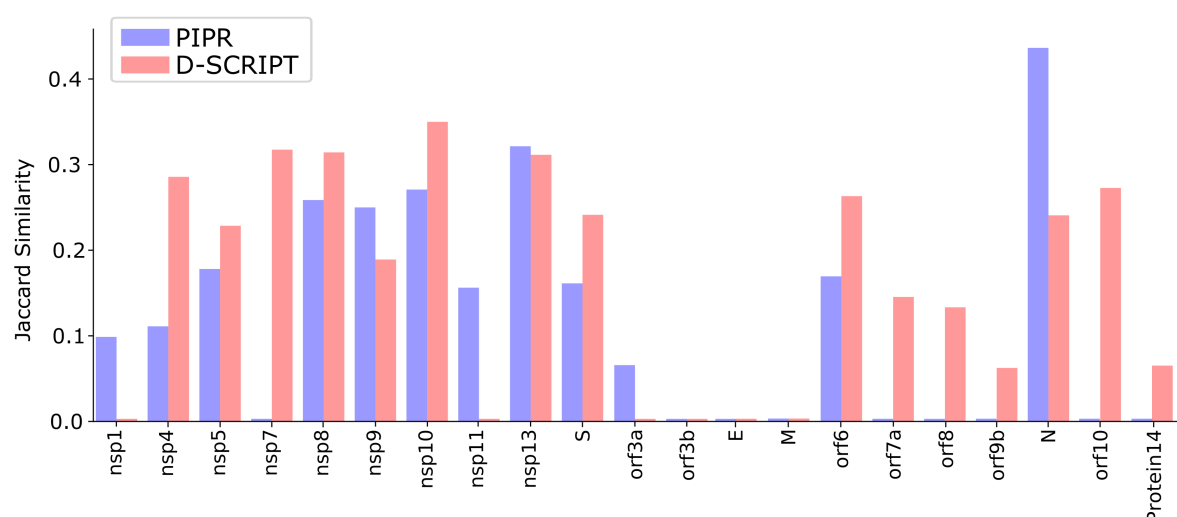
**Figure A.2: Distribution of within-cluster similarity scores using GOGO Protein Distances.** D-SCRIPT recovers significantly more functionally coherent clusters ( $p = 3.94e - 06$ ) than PIPR when similarity between proteins is measured using GOGO [42]. Clusters of interacting proteins were constructed from the PIPR and D-SCRIPT networks using spectral clustering, where distances between proteins were computed using the diffusion state distance (DSD) metric [38]. We compute the average similarity between pairs of proteins in each cluster. Similarity between proteins is calculated as GOGO Biological Processes similarity using all FlyBase GO term annotations.

factors (ORF\* and Protein 14), D-SCRIPT performs significantly better (mean Jaccard similarity 0.029 vs. 0.118,  $p = 0.022$ , paired one-tailed  $t$ -test). Visually, PIPR seems to be somewhat better at predicting interaction partners for the non-structural proteins (NSP\*), although D-SCRIPT still has a slightly larger mean similarity (0.183 vs. 0.222,  $p = 0.221$ ). While D-SCRIPT performs better on the intensively studied spike (S) protein, PIPR shows a higher overlap for the nucleocapsid (N). Neither method predicts enriched terms for the other structural proteins encoding the envelope (E) and membrane (M) (0.149 vs. 0.121,  $p = 0.672$  across the four proteins).

**Methods.** Candidate pairs were generated using the viral sequences from Gordon et al. [43] and 19,777 human sequences from the STRING database (Section 3), and predicted edges using D-SCRIPT and PIPR. We predicted 3,273 edges using D-SCRIPT and 2,922 edges using PIPR. 332 putative true viral-host interactions were taken from Gordon et al. Human sequences were mapped to UniProt sequences identifiers from [43] with sequence similarity  $\geq 95\%$  using BLAST [46], and UniProt identifiers were used to identify a set of Gene Ontology terms for the human interactors of each viral protein. Following [43], we identified over-represented GO terms using the clusterProfiler R package (version 3.14.3) [50] with a 1% false discovery rate (FDR). Over-represented GO terms were mapped to a common set of terms taken from the ChEMBL Drug Target GO Slim Subset [51]. For each viral protein, we computed the Jaccard similarity between the set of GO Slim terms enriched in the putative true network and each of the computationally predicted methods. We computed a paired one-tailed  $t$ -test to statistically compare the relative similarities of D-SCRIPT and PIPR.

**Availability.** Virus-host edges predicted using D-SCRIPT or PIPR are available for use by the community at <http://dscrip.mit.edu>.





**Figure A.3: Similarity of Enriched GO Terms to True Network.** Interactions between SARS-CoV-2 proteins and human proteins were predicted using D-SCRIPT and PIPR. Viral proteins were then annotated with the enriched ChEMBL GO Slim terms linked to their human interactors. Compared to PIPR, interactions computed with D-SCRIPT show a greater annotation overlap ( $p = 0.059$ ) with those estimated from putative true interactions from Gordon et al. [43]. D-SCRIPT performs especially well at predicting functional enrichments for putative accessory factor proteins, where PIPR recovers none of the enriched functional terms in several cases.