Bioinformatics, 37(20), 2021, 3428–3435
doi: 10.1093/bioinformatics/btab346
Advance Access Publication Date: 12 May 2021
Original Paper

OXFORD

Sequence analysis

Bayesian neural network with pretrained protein embedding enhances prediction accuracy of drug-protein interaction

QHwan Kim, Joon-Hyuk Ko, Sunghoon Kim , Nojun Park and Wonho Jhe *

Department of Physics and Astronomy, Institute of Applied Physics, Seoul National University, Gwanak-gu, Seoul 08826, Republic of Korea

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli
Received on December 16, 2020; revised on April 26, 2021; editorial decision on April 30, 2021; accepted on May 5, 2021

Abstract

Motivation: Characterizing drug–protein interactions (DPIs) is crucial to the high-throughput screening for drug discovery. The deep learning-based approaches have attracted attention because they can predict DPIs without human trial and error. However, because data labeling requires significant resources, the available protein data size is relatively small, which consequently decreases model performance. Here, we propose two methods to construct a deep learning framework that exhibits superior performance with a small labeled dataset.

Results: At first, we use transfer learning in encoding protein sequences with a pretrained model, which trains general sequence representations in an unsupervised manner. Second, we use a Bayesian neural network to make a robust model by estimating the data uncertainty. Our resulting model performs better than the previous baselines at predicting interactions between molecules and proteins. We also show that the quantified uncertainty from the Bayesian inference is related to confidence and can be used for screening DPI data points.

Availability and implementation: The code is available at https://github.com/QHwan/PretrainDPI.

Contact: whjhe@snu.ac.kr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Identifying novel drug–protein interactions (DPIs) has been studied broadly for the prediction of potential side effects (Mizutani *et al.*, 2012), toxicities (Liebler and Guengerich, 2005) and repositioning of drugs (Pushpakom *et al.*, 2019; Xue *et al.*, 2018). However, quantifying the DPI of every possible drug–protein pairs is prohibitively time-consuming and expensive since it requires individual experiments or simulations for each and every pairs.

With the development of public datasets for protein sequences and molecule–protein interactions (Liu *et al.*, 2007, 2015), machine learning-based methods (Fokoue *et al.*, 2016; He *et al.*, 2017; Vamathevan *et al.*, 2019; Wen *et al.*, 2017) have emerged as candidates for fast DPI identification. Recently, deep neural networks (DNNs) have attracted attention because they outperform other machine learning-based methods in various tasks, such as computer vision (He *et al.*, 2015) and natural language processing (Devlin *et al.*, 2019; Vaswani *et al.*, 2017).

In usual DPI task, a protein is represented as a one-dimensional long sequence of amino acid characters. Thus, deep learning models for natural language processing have been broadly used to obtain useful protein features from the sequences. Previous studies in this approach include using recurrent neural networks with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) or gated recurrent unit (Cho *et al.*, 2014) layers for their ability to identify long-term dependencies in sequential data (Gao *et al.*, 2018; Karimi *et al.*, 2019; Wang *et al.*, 2020). Other studies have used convolutional neural networks (CNNs) (Lee *et al.*, 2019; Öztürk *et al.*, 2018; Shin *et al.*, 2019; Tsubaki *et al.*, 2019; Zhang *et al.*, 2019) to extract hidden local patterns in sequences. Different representations of proteins, such as two-dimensional contact maps (Jiang *et al.*, 2020; Zheng *et al.*, 2020) or three-dimensional atom coordinates (Lim *et al.*, 2019; Morrone *et al.*, 2020), in addition to one-dimensional sequences, have also been used to increase model performance.

Supervised training of high-capacity DNN models from scratch requires a large amount of labeled training data points. For example, Mahajan *et al.* (2018) showed that more labeled data is required to increase accuracy after training 10^9 images. However, currently available DPI datasets usually contain thousands of labeled protein sequences, a small number compared to the >195 M unrevealed interaction information in UniProtKB (UniProt Consortium, 2015). The lack of qualified labeled data points suppresses the usage of

more elaborated deep learning architectures, which could potentially increase performance and reliability (Brigato and Iocchi, 2020). In particular, the scarcity of labeled data of biology- and chemistry-related tasks has been suggested consistently (Ryu *et al.*, 2019; Vamathevan *et al.*, 2019) although the labeling requires expensive and time-consuming experiments.

To overcome the difficulties of learning with limited data, several studies have proposed methods to increase the expressiveness of deep learning models without additional endeavor to label generation. Of those, transfer learning uses a model pretrained with a large corpus of data on different tasks. This pretrained model is then transferred to the target tasks by adding classification layers and fine-tuning with the original small dataset. Transfer learning approaches have shown substantial performance improvement in computer vision (Kornblith *et al.*, 2019), natural language processing (Devlin *et al.*, 2019) and structure-property prediction of molecules (Hu *et al.*, 2020; Winter *et al.*, 2019). In cases where labeled data is expensive, such as in scientific problems, the pretrained model can be prepared in an unsupervised manner, using large but unlabeled datasets. Winter *et al.* (2019) trained an autoencoder model with a huge corpus of chemical structures and used it to predict molecular properties. Villegas-Morcillo *et al.* (2020) showed that supervised classification tasks with a pretrained protein sequence model could achieve competitive performance with other complicated models. On the other hand, in the study of protein–drug interactions where encoding long protein sequence is important, previous works used small protein–drug interaction datasets which only contain few tens of thousands of protein sequences. Adopting a pretrained model trained on a vast amount of protein sequences could be used to construct a more robust protein–drug interaction classification model.

Another method to obtain a more robust and reliable model with a small dataset is the Bayesian neural network (BNN) (Gal and Ghahramani, 2015). Compared to a conventional DNN, which gives a definite point prediction for each given input, a BNN returns a distribution of predictions, which qualitatively corresponds to the aggregate prediction of an ensemble of different neural networks trained on the same dataset. Direct implementation of BNN is infeasible because training an ensemble of neural networks requires enormous computing power. Monte-Carlo dropout (MC-dropout) approach (Gal and Ghahramani, 2016; Kendall and Gal, 2017) enables training BNNs in reasonable time by approximating the posterior distribution of network weights by a product of Bernoulli distributions using dropout layers.

Here, we propose an end-to-end deep learning framework for highly accurate DPI prediction with transfer learning and BNN. The transfer learning method is used to obtain protein-level representations from the pretrained model. We choose the pretrained model as a stacked transformer architecture trained with 250 million unlabeled protein sequences in an unsupervised manner (Rives *et al.*, 2019). The protein embeddings extracted from the pretrained model are prepared with a large corpus of sequences and are expected to have a large expression capacity. The molecules are represented by molecular graphs and are encoded through the graph interaction network layers. We use three public DPI datasets, and the estimation of the model performance shows that our proposed model outperforms previous baseline approaches. Further study shows that the choice of the pretrained model and the GraphNet is essential to the increase of prediction accuracy. From the BNN, we can estimate the prediction uncertainty by sampling outputs. The proposed model correctly decomposes estimated uncertainty into model-based and data-based elements. These uncertainties can further be used to virtually screen data points, which excludes data points with high uncertainty to increase model prediction. In summary, the main contributions of our work are as follows.

1. We propose the first approach to predict DPI with the BNN framework and the pretrained protein sequence model;
2. our method demonstrates highly accurate predictions on three public DPI datasets; and
3. the output of the BNN can estimate the confidence of the data points.

2 Materials and methods

2.1 Datasets

We evaluate our model and other baseline models on three public DPI datasets: the BindingDB dataset (Gao *et al.*, 2018), the Human dataset (Liu *et al.*, 2015) and the *C. elegans* dataset (Liu *et al.*, 2015).

2.1.1 BindingDB

BindingDB is a public database of experimentally measured binding affinities between small molecules and proteins (Liu *et al.*, 2007). The original dataset contains 1.3 million interaction labels with quantitative measurements of IC_{50}, EC50 and Ki. We use the binarized version of the BindingDB dataset constructed by Gao *et al.* (2018), which contains 39 747 positive interactions and 31 218 negative interactions. The training/validation/testing split is already defined in the prepared dataset and no cross-validation is adopted. The training set contains 28 240 positive and 21 915 negative interactions. The validation set includes 2831 positive and 2776 negative interactions. The test set contains 2706 positive and 2802 negative interactions.

In the BindingDB dataset, some molecule/protein data points exist in both train and test datasets. Following suggestions from previous works (Gao *et al.*, 2018), we further split the test dataset into four sub-test sets that the model can be learned and applied to predict the label between a molecule and protein target. The binary interaction test data is divided by 'seen' and 'unseen' depending on whether the protein and molecule are observed in the training dataset. The combination of seen and unseen can be applied to a specific task. For example, one can use the seen drug and unseen protein pair for the drug repositioning task.

2.1.2 Human and *Caenorhabditis elegans*

Created by Liu *et al.* (2015), these datasets include highly credible negative samples of the compound-protein pairs obtained using a systematic screening framework. Following Tsubaki *et al.* (2019), we use the balanced and the unbalanced dataset, where the ratios of the positive to negative samples are 1:1 and 1:3, respectively. The human dataset contains 3369 positive interactions between 1052 unique molecules and 852 unique proteins; the *C.elegans* dataset contains 4000 positive interactions between 1434 unique molecules and 2504 unique proteins. Also, we use an 80%/10%/10% training/validation/testing random split with a five-fold cross-validation strategy. The ratio of classes (1:1 and 1:3) in the training/validation/testing sets is preserved.

2.2 Proposed model

In this study, the DPI is defined as a binary label representing the presence of an interaction. Figure 1a shows the schematic of the proposed model. The input data is a pair of strings consisting of a protein sequence and a drug SMILES string. The input data passes embedding layers to be encoded as a pair of representation vectors. These protein and drug representation vectors are then concatenated and passed through fully connected layers, resulting in a binary prediction for interaction. In each training cycle, this prediction is compared with the ground truth, and model parameters are tuned to decrease the difference between the two using the backpropagation algorithm. To implement BNNs, we apply dropout layers in every layer except the pretrained layer, the concatenation layer, and the final fully-connected layer. Detailed descriptions of the model are given below.

2.2.1 Feature extraction of proteins

A protein sequence is represented as a list of amino acids provided in the raw training data. Note that we do not use a set of gene

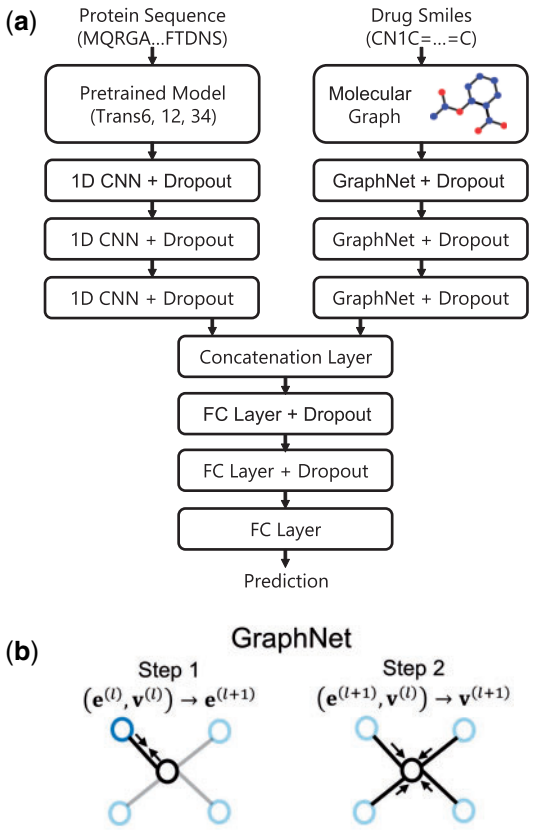

Fig. 1. An overview of the proposed neural network architecture schematic. (**a**) The protein and molecule representations are obtained by passing through the pretrained transformer model and GraphNet layers, respectively. The protein and molecule representation vectors are then concatenated and fed into a classifier consisting of fully connected layers. (**b**) Mechanism of the message passing in GraphNet. The GraphNet performs message passing on the molecular graph, recursively updating graph edges $\mathbf{e}^{(l)}$ and nodes $\mathbf{v}^{(l)}$

ontology annotations that provides high-level information on the protein functions. To extract protein-level embeddings, we use the pretrained models from Rives *et al.* (2019), which were trained with 250 million protein sequences in an unsupervised manner. Rives *et al.* (2019) used an attention-based transformer architecture (Vaswani *et al.*, 2017) and found that their model outperforms other recurrent network-based methods for predicting protein functionality. We select three models, Trans6, Trans12 and Trans34, which are pretrained with 6, 12 and 34 transformer layers, respectively.

For each protein sequence of length L_p, the pretrained models output an embedding matrix $\mathbf{X}_\mathrm{p} \in \mathbb{R}^{L\times d}$, where $d = 768$ for Trans6, Trans12 and $d = 1,280$ for Trans34 model. From amino-acid level feature $\mathbf{X}_\mathrm{p}$, we obtain the protein level feature $\mathbf{x}_\mathrm{p}^{(0)} \in \mathbb{R}^d$ by averaging over the L amino acids features.

With the protein-level embedding $\mathbf{x}_\mathrm{p}^{(0)}$, we use three 1-dimensional convolutional neural networks (1D-CNN) to smooth patterns in protein features. Note that the 1D-CNN gives slightly better performance than the fully-connected layers.

2.2.2 Feature extraction of drugs

The raw training data of drugs is in the SMILES (Simplified Molecular Input Line Entry System) format (Weininger, 1988). For each input SMILES string, we construct a corresponding molecular graph that contains connectivity and structure information of the compound.

In the molecular graph, atoms and bonds are represented with vectors with structural features that characterize the surrounding chemical environment. The details of the attributes are shown in Supplementary Table S1, which follow the feature design from DeepChem (Wu *et al.*, 2018). The graph construction and corresponding feature extraction processes are conducted using RDKit (Landrum, 2006)—an open-source chemical informatics software. Initial encodings of the i-th atom and bond between the i- and j-th atoms are denoted as vectors, $\mathbf{v}_i^{(0)}$ and $\mathbf{e}_{ij}^{(0)}$, respectively. These atom and bond features are updated by a message passing-based graph network during model inference.

The message passing framework of graph data has been used broadly to predict the properties of crystal (Xie and Grossman, 2018), organic molecules (Ryu *et al.*, 2019), ice (Kim *et al.*, 2020) and glasses (Bapst *et al.*, 2020). To extract the drug molecule features, we use the graph interaction network (GraphNet) model (Battaglia *et al.*, 2016). Figure 1b shows the schematic of the GraphNet mechanism. First proposed by Battaglia *et al.* (2016) to infer interactions between objects, the GraphNet exchanges information between graph edges and nodes and recursively updates them.

The GraphNet first updates an edge between the i- and j-th nodes as,

$$\mathbf{e}_{ij}^{(l+1)} = \mathrm{ReLU}\left[\left(\mathbf{e}_{ij}^{(l)} \oplus \mathbf{v}_i^{(l)} \oplus \mathbf{v}_j^{(l)}\right)\mathbf{W}_\mathrm{e}^{(l)} + \mathbf{b}_\mathrm{e}^{(l)}\right], \tag{1}$$

where $\oplus$ is the concatenation operator, $\mathbf{W}_\mathrm{e}^{(l)}$ is the weight matrix of the edge update, and $\mathbf{b}_\mathrm{e}^{(l)}$ is the bias. Then the update of the i-th node is carried out using the features of the node and the sum of its linked edge features as,

$$\mathbf{v}_i^{(l+1)} = \mathrm{ReLU}\left[\left(\mathbf{v}_i^{(l)} \oplus \sum_{j\in N(i)} \mathbf{e}_{ij}^{(l+1)}\right)\mathbf{W}_\mathrm{v}^{(l)} + \mathbf{b}_\mathrm{v}^{(l)}\right], \tag{2}$$

where $\mathbf{W}_\mathrm{v}^{(l)}$ is the weight matrix of node update, and $\mathbf{b}_\mathrm{v}^{(l)}$ is the bias. After the updates of node and edge states are finalized, we obtain a graph feature (molecular feature) by gathering all the node and edge states. As a gathering function, we choose the most typical readout function, which is an average of every atom and bond states processed by,

$$\mathbf{x}_\mathrm{d} = \left(\frac{1}{N_\mathrm{v}}\sum_i \mathbf{v}_i\right) \oplus \left(\frac{1}{N_\mathrm{e}}\sum_{i,j} \mathbf{e}_{ij}\right), \tag{3}$$

where N_v and N_e are the numbers of nodes and edges in the molecular graph, respectively.

2.2.3 Classifier

We prepare the drug-protein feature vector $\mathbf{x}$ by concatenating $\mathbf{x}_\mathrm{p}$ and $\mathbf{x}_\mathrm{d}$,

$$\mathbf{x} = \mathbf{x}_\mathrm{p} \oplus \mathbf{x}_\mathrm{d}. \tag{4}$$

In the classifier block, the feature vector $\mathbf{x}$ passes fully connected layers with ReLU activation to output the final prediction value. The dimension of the last layer is 2, corresponding to the one-hot encoding of the binary classification labels.

2.2.4 Bayesian neural network

For a given training set $\{\mathbf{X}, \mathbf{Y}\}$, let $p(\mathbf{Y}|\mathbf{X}, \mathbf{w})$ and $p(\mathbf{w})$ be model likelihood and a prior distribution for a vector of model parameters $\mathbf{w} = \{\mathbf{W}_1, \ldots, \mathbf{W}_k\}$, where k is the number of layers. In a Bayesian framework, model parameters are considered as random variables and the output is defined as

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int_\Omega p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathbf{X}, \mathbf{Y})\mathrm{d}\mathbf{w} \tag{5}$$

for a new input $\mathbf{x}^*$ and a new output $\mathbf{y}^*$.

The direct computation of Equation (5) in the neural network is often infeasible because of the heavy computational cost required to train an ensemble of weights. Here, we use variational inference, approximating the posterior distribution with a distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{Y}) \sim q_\theta(\mathbf{w})$ parameterized by a low-dimensional variational parameter θ.

The quality of the variational distribution $q_\theta(\mathbf{w})$ is crucial to the implementation of the BNN. The recently proposed MC-dropout approach attaches dropout layers to every neural network layer to approximate the posterior distribution with a product of Bernoulli distributions (Gal and Ghahramani, 2016). The MC-dropout method is practical because it does not need a model ensemble directly to obtain the variational posterior distribution. Also, the expectation and the variance of an output can be easily obtained with the collection of outputs sampled by the repeated inference of a new input $\mathbf{x}^*$ while the dropout layers are turned on. Thus, we adopt MC-dropout in this work.

Performing variational inference with the variational distribution $q_\theta(\mathbf{w})$ results in the variational predictive distribution of a new output $\mathbf{y}^*$ given a new input $\mathbf{x}^*$ as

$$q_\theta^*(\mathbf{y}^*|\mathbf{x}^*) = \int_\Omega q_\theta(\mathbf{w}) p\left(\hat{\mathbf{y}}^*(\mathbf{w})|\mathbf{x}^*, \mathbf{w}\right) d\mathbf{w}, \tag{6}$$

where $\hat{\mathbf{y}}^*(\mathbf{w})$ is the output of input $\mathbf{x}^*$ for a given $\mathbf{w}$. In BNN, the integration in Equation (6) is replaced with a predictive mean over T times of MC sampling, which is estimated by

$$\hat{E}[\mathbf{y}^*|\mathbf{x}^*] = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{y}}_t^*. \tag{7}$$

where $\hat{\mathbf{y}}_t^*$ is t-th estimation of BNN with input $\mathbf{x}^*$.

In estimating the predictive variance of the model, we decompose the source of uncertainty into aleatoric and epistemic, which was first suggested by Kendall and Gal (2017) and optimized for classification tasks by Kwon *et al.* (2020). Aleatoric uncertainty originates from the inherent noise of data points, while epistemic uncertainty arises due to model prediction variability. Here, we use the method suggested by Kwon *et al.* (2020), which does not involve extra parameters.

The predictive variance is estimated by

$$\hat{\mathrm{Var}}[\mathbf{y}^*|\mathbf{x}^*] = \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{\mathbf{y}}_t^* - \bar{\mathbf{y}})(\hat{\mathbf{y}}_t^* - \bar{\mathbf{y}})^T}_{\text{epistemic}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\mathrm{diag}(\hat{\mathbf{y}}_t^*) - (\hat{\mathbf{y}}_t^*)(\hat{\mathbf{y}}_t^*)^T\right)}_{\text{aleatoric}}, \tag{8}$$

where $\bar{\mathbf{y}} = \sum_{t=1}^{T} \hat{\mathbf{y}}_t^*/T$ and $\hat{\mathbf{y}}_t^* = \mathrm{softmax}\left(\mathbf{f}^{\hat{\mathbf{w}}_t}(\mathbf{x}^*)\right)$.

2.3 Implementation and evaluation strategy

We implement our proposed model with Pytorch 1.5.1 (Paszke *et al.*, 2019). The training process takes at most 200 epochs on all the datasets using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001 and a batch size of 32. The hidden layer dimensions of GraphNet in the molecular feature extractor and MLP in the classifier are 256 and 512, respectively. The number of layers of both the protein and drug feature extractors is set to 3. The coefficient of L2 regularization is 0.001. These hyperparameters are searched in a wide range.

The training objective is to minimize the loss function $\mathcal{L}$, given by the sum of the cross-entropy loss and the regularization as follows

$$\mathcal{L}(\mathbf{w}) = -\sum_{i=1}^{N} y_i[\log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)] + \frac{\lambda}{2}||\mathbf{w}||_2^2, \tag{9}$$

where $\mathbf{w}$ is the set of model parameters, N is the number of interaction labels, and λ is the L2 regularization hyperparameter.

To implement MC-dropout sampling, we turn on dropout layers during inference on test datasets with $T = 30$ samplings. The mean performance and the decomposed uncertainties of the output are calculated with Equations (7) and (8), respectively.

The main performance metric was chosen to be the area under the receiver operating curve (ROC-AUC). ROC-AUC is defined as the area under the ROC curve whose x- and y-axis is a false positive rate and true positive rate, respectively. It is broadly used as the main metric of binary classification because it takes into account all classification thresholds from 0 to 1. We also report some additional performance metrics—accuracy for the BindingDB dataset, and precision and recall for the Human and *C.elegans* dataset in line with the original studies.

3 Results and discussions

To train DPI datasets, we prepare six models, Trans6, Trans12, Trans34, Trans6+Drop, Trans12+Drop and Trans34+Drop. The latter three models use the pretrained protein model and implement the BNN architecture with MC-Dropout (Fig. 1a), while the former three models only use the pretrained model. The numbers 6, 12 and 34 correspond to the number of transformer layers in the pretrained model.

3.1 Performance of the proposed model

With the BindingDB dataset, we compare our model against three baselines: Tiresias, DBN, and E2E. Tiresias uses similarity measures of drug and protein pairs (Fokoue *et al.*, 2016). DBN uses stacked restricted Boltzmann machines with the inputs as extended connectivity fingerprints (Wen *et al.*, 2017). E2E uses graph convolutional networks and LSTM to process drug-protein pair information with Gene Ontology annotations (Gao *et al.*, 2018).

As described in Section 2, we further split the test dataset into four sub-test sets with seen/unseen protein/drug. Figure 2 shows that the proposed method consistently performs well on all four sub-test sets. The tables for the performance evaluation with Figure 2 are included in Supplementary Table S2. The models with pretraining and MC-dropout give consistently high performance in all four categories. The sub-test dataset with unseen protein is difficult to classify, and only the E2E model shows comparable performance with our proposed model. Tiresias and DBN perform well on seen proteins and outperform E2E but have much worse performances on unseen proteins. The features used in these two models, similarity

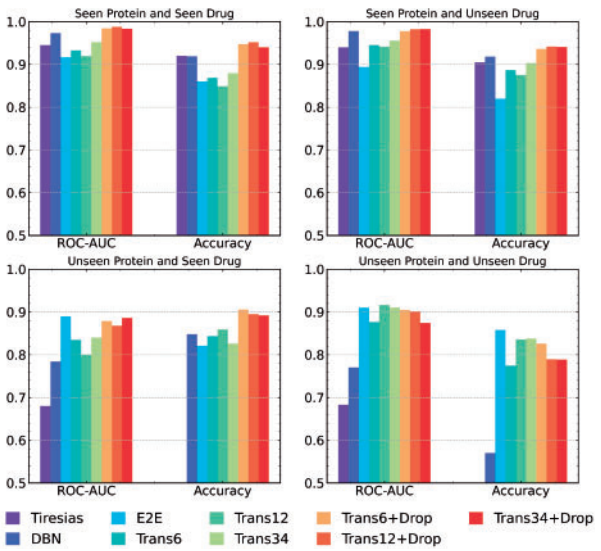

Fig. 2. Performance comparison of proposed models, similarity-based approach (Tiresias), stacked restricted Boltzmann layers (DBN) and graph convolutional networks—long short-term memory-based approach (E2E). For each model, two metrics are reported: area under receiver operating characteristic curve (ROC-AUC) and accuracy. The binary interaction test data is divided into 'seen' and 'unseen' depending on whether the protein and drug are observed in the training dataset. The accuracy scores of Tiresias are not seen in the bottom graphs because they are lower than the lower bound of the y-axis

score and predefined molecular fingerprints, do not generalize molecules well. E2E uses machine-based molecular features and performs better than the Tiresias and DBM on unseen proteins, but its overall performance is lower than Trans+Drop. The Trans+Drop models consistently perform better than the Trans models as well as other baselines. Only for the unseen protein and unseen drug, Trans+Drop shows similar performances with Trans and E2E.

The protein embeddings extracted from Trans+Drop can have a large expression capacity because the pretrained protein model is prepared with 250 M sequences (Rives *et al.*, 2019). It implies that the extraction of generalized protein embedding with a long sequence plays an essential role in DPI classification. If we measure scores by aggregating four test sub-datasets, the ROC-AUC of Trans6+Drop, which achieves the best score amongst the proposed models, is 0.943 while those of Tiresias, DBN and E2E are 0.818, 0.881 and 0.913, respectively. The overall ROC-AUC scores of other models are shown in Supplementary Table S3.

Also, we compare our proposed method with previous DPI approaches on the Human and the *C.elegans* datasets. The models used for comparison are the k-nearest neighbor (k-NN), random forest (RF), L2-logistic (L2), support vector machine (SVM) and graph neural network (GNN) models. The k-NN, RF, L2 and SVM models use similarity features of drug structures and protein sequences. The GNN model uses *n*-grams to encode protein sequences and molecular embeddings based on subgraphs defined within a given radius. We note that the baseline models of these datasets are different from those of BindingDB because we choose models from the previous studies of each dataset. For the Human and the *C.elegans*, we refer Tsubaki *et al.* (2019).

As shown in Table 1, our best performing model achieves the highest ROC-AUC, precision, and recall scores among the neural network-based methods. In the human dataset, SVM shows better performance for the Precision score, but our proposed model outperforms in the other metrics. In the *C.elegans* dataset, Trans6+Drop shows the best performance over all metrics, except for the recall score of the balanced dataset where Trans34+Drop performs best.

Our results show that models with transfer learning and BNN (Trans6+Drop, Trans12+Drop, Trans34+Drop) outperform other baseline models when evaluated with the three public DPI datasets. We note that only the pretrained protein sequence can train models (Trans6, Trans12, Trans34) competitive with the baselines, but an additional Bayesian frameworks further increase performance. The BNN model is also a good predictor for an unbalanced dataset, a common problem in real drug-protein interaction applications. It suggests that the role of BNN, training robust model is another key figure of performance enhancement.

To characterize the importance of the encoding methods we proposed, we compare ROC-AUC curves with different protein and drug representations. Figure 3a shows ROC-AUC curves of different protein embedding methods with (Trans34+Drop) and without (Drop) pretrained layer. In the Drop model, we use one-hot encoding for the protein sequence and use three 1D-CNN layers. The result shows that the extraction of protein level encoding obtained from the pretrained layer increases model performance. We also consider the importance of the molecular graph encoding method by using the graph convolutional network (GCN) (Kipf and Welling, 2017) and comparing it to GraphNet. Figure 3b shows that the choice of message passing algorithms also determines prediction accuracy. The GraphNet architecture, which uses node and edge features and updates them iteratively, shows relatively better results than the GCN, which uses the node feature alone.

The additional point is that the most complex model, Trans34+Drop, does not always give the best results. This is in agreement with the literature, where it was found that the prediction accuracy is not strictly proportional to the sequence model complexity (Rives *et al.*, 2019). We increase the number of 1D-CNN and GraphNet layers, respectively, and characterize the relation between model complexity and model performance. Supplementary Figure S1 shows that the validation ROC-AUC score of Trans34+Drop is maximized when the number of layers of both the protein and drug encoding layers is set to 3. If the architecture is larger than this size,

Table 1. ROC-AUC, Precision and Recall scores of human and *C.elegans* dataset with proposed models, k-nearest neighbor (k-NN), random forest (RF), L2 logistic (L2), support vector machine (SVM) and graph neural network (GNN) proposed by Tsubaki *et al.* (2019)

Human

Methods	Balanced Dataset (1: 1)			Unbalanced Dataset (1: 3)		
	ROC-AUC	Precision	Recall	ROC-AUC	Precision	Recall
KNN	0.860	0.798	0.927	0.904	0.716	0.882
RF	0.940	0.861	0.897	*0.954*	0.847	0.824
L2	0.911	0.891	0.913	0.920	0.837	0.773
SVM	0.910	*0.966*	*0.950*	0.942	*0.969*	0.883
GNN	*0.970*	0.923	0.918	0.950	0.949	*0.913*
Trans6	0.968	0.902	0.901	0.971	0.915	0.910
Trans12	0.960	0.881	**0.949**	0.969	**0.958**	0.863
Trans34	0.973	0.914	0.925	0.971	0.930	0.863
Trans6+Drop	**0.975**	0.932	0.922	**0.976**	0.939	0.902
Trans12+Drop	0.971	0.914	0.924	0.963	0.932	0.902
Trans34+Drop	**0.975**	**0.945**	0.935	0.970	0.925	**0.923**

C.elegans

Methods	Balanced Dataset (1: 1)			Unbalanced Dataset (1: 3)		
	ROC-AUC	Precision	Recall	ROC-AUC	Precision	Recall
KNN	0.858	0.801	0.827	0.892	0.787	0.743
RF	0.902	0.821	0.844	0.926	0.836	0.705
L2	0.892	0.890	0.877	0.896	0.875	0.681
SVM	0.894	0.785	0.818	0.901	0.837	0.576
GNN	*0.978*	*0.938*	*0.929*	*0.971*	*0.916*	*0.921*
Trans6	0.981	0.937	0.949	0.977	0.871	0.917
Trans12	0.975	0.949	0.910	0.967	0.876	0.861
Trans34	0.973	0.914	0.925	0.969	0.900	0.915
Trans6+Drop	**0.986**	**0.955**	0.933	**0.983**	**0.923**	**0.944**
Trans12+Drop	0.980	0.946	0.928	0.981	0.890	0.940
Trans34+Drop	0.981	0.946	**0.940**	0.980	0.914	0.937

Note: The best scores for each of the proposed models are emphasized in bold. The italicized scores correspond to the best scores for the baseline models.

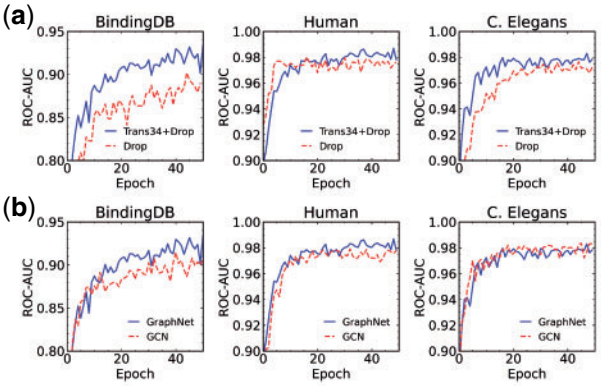

Fig. 3. Comparison of ROC-AUC curves on the validation set as a function of epoch with different embedding methods. (**a**) Performance comparison of protein embedding methods with (Trans34+Drop) and without (Drop) pretrained layer. (**b**) Performance comparison of drug molecule embedding methods with GraphNet and graph convolutional network (GCN) in Trans34+Drop architecture

the ROC-AUC score saturates or even decreases because an over-smoothing occurs. Therefore, when using transfer learning, we recommend preparing several pretrained models and comparing their results before making the final choice.

3.2 Robustness of proposed model

In this section, we test the robustness of the Bayesian models by varying the protein data quality. The robustness is estimated by tracking the degradation of the model performance as more and more external noise is added to the dataset. The type of noise for the experiment is chosen to be the Gaussian noise $\mathcal{N}(0, \sigma^2)$, where 0 is the mean and σ is the standard deviation of the distribution.

Figure 4 shows the ROC-AUC scores of the two models Trans6 and Trans6+Drop applied to three DPI datasets as a function of the noise level σ. As the noise level increases, the ROC-AUC of Trans6+Drop remains more robust to the additive noise than Trans6. In the BindingDB dataset, the ROC-AUC score of Bayesian Trans6+Drop does not fall under 0.8 when noise standard deviation increases until 0.5, whereas Trans6 loses its predictability. For Human and *C.elegans* datasets, the models maintain relatively good performance regardless of the additive noise, but the Bayesian model consistently outperforms the other. It indicates that the BNN architecture trains model more robust to noise, a point we attribute to the overall enhanced performance of our proposed model.

Note that the predictions on the BindingDB dataset are more vulnerable to external noise than those for the other two datasets. We relate this behavior with the 'classification difficulty' of the datasets. Because the datasets are curated in different sample pools, some datasets could contain more points near the classification boundary than other datasets. The dataset with a large subset of data points lying on the classification boundary can be more obfuscated by noise. One can indirectly estimate the classification difficulty of the datasets by comparing the classification scores without the noise. When we consider the Trans6+Drop model, the ROC-AUC score of BindingDB (0.943) is smaller than those of the other two datasets (0.975, 0.986). It indicates that the BindingDB is more challenging to classify and therefore more vulnerable to external noise.

3.3 Quality of estimated uncertainties

We first test whether the uncertainties obtained from the proposed BNN model are correctly estimated. This is accomplished by reducing the training set sizes and observing the resulting changes in the uncertainties. When dataset size is decreased, aleatoric uncertainty, which is related to the inherent noise of the data, should stay constant. In contrast, the model error-related epistemic noise should increase due to the lack of sufficient training data.

Table 2 shows the uncertainties obtained from the reduced training set sizes (1, 1/2, 1/4) and the entire test set. The uncertainties are obtained via Eq. (8). It shows that the epistemic uncertainty increases as the training size gets smaller, while the aleatoric uncertainty remains relatively constant. It indicates that our proposed model reliably estimates uncertainties.

Because the model successfully estimates uncertainties, we can plot confidence-accuracy graphs, as shown in Figure 5. We use three uncertainties, epistemic uncertainty, aleatoric uncertainty and the sum of the two. Here confidence percentile means that we only consider the top n percent of data points in the test set ranked by the

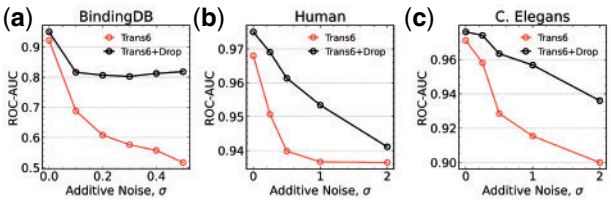

Fig. 4. ROC-AUC scores on the test set as a function of the standard deviation of the additive noise on (**a**) BindingDB, (**b**) Human and (**c**) *C.elegans* dataset. The additive noise is sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$

Table 2. Epistemic and aleatoric uncertainties for a range of different training dataset sizes (1, 1/2, 1/4 of the original training dataset size)

Dataset	Epistemic	Aleatoric
BindingDB/4	0.018	0.036
BindingDB/2	0.013	0.037
BindingDB	0.011	0.037
Human/4	0.0128	0.020
Human/2	0.0096	0.018
Human	0.0082	0.019
C.elegans/4	0.0137	0.0155
C.elegans/2	0.0098	0.0153
C.elegans	0.0053	0.0143

Note: The results show that the aleatoric uncertainty remains constant, whereas the epistemic uncertainty increases when the training size decreases.

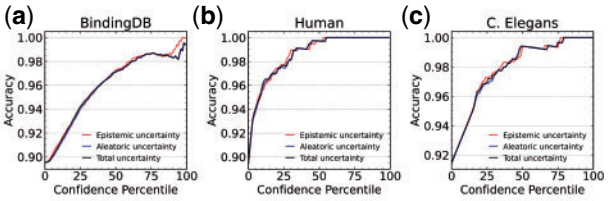

Fig. 5. Model accuracies on the test set as a function of confidence percentile of (**a**) BindingDB, (**b**) Human and (**c**) *C.elegans* dataset. The confidence is estimated based on the epistemic uncertainty (red line), aleatoric uncertainty (blue line), and the sum of the two (black line)

confidence, which is defined as the inverse of uncertainty. The plots show how the test set accuracy varies as a function of the confidence percentile. In every dataset, the accuracy is an increasing function of model confidence. Thus the data points with low confidence can be interpreted as the outlier and can be screened in DPI datasets in drug development applications. For example, if we delete 50% of the lowest confident points of the Human dataset, we can achieve nearly 100% accuracy. Note that there is no consistent trend regarding which uncertainty is more important, and the two uncertainties should be treated equally to achieve an accurate estimation.

For BindingDB, the test dataset is divided into four categories with the 'seen' and 'unseen' proteins and drugs. The sub-test datasets of the 'unseen' categories include data points out of training data distributions and which are expected to be biased. We plot the probability density distributions of predicted variance (uncertainty) of four test sub-datasets of BindingDB in Figure 6. The result shows that the biased level of a sub-dataset is related to its predicted variance. The most biased dataset, unseen protein and unseen drug, shows the highest variances. It indicates that when we screen the test dataset using the confidence percentile (Fig. 5), the most biased data points are initially screened. The BNN architecture we proposed can thus be useful to overcome dataset bias in predicting protein–drug interactions.

3.4 Case study

To verify the effectiveness of the proposed architecture in practical problems, we test interactions between antiviral drugs being used and SARS-CoV-2 proteins. We use the amino acid sequences of 3C-like protease (PDB ID: 6WQF) and RNA-dependent RNA polymerase (NCBI: YP_009725307.1) of the SARS-CoV-2 replication complex from the Protein Data Bank (PDB) database and the National Center for Biotechnology Information (NCBI). We prepare five drug candidates for SARS-CoV-2 proteins.

Tables 3 and 4 show the drug-protein interaction prediction list for 3C-like protease and RNA polymerase proteins. Table 3 shows that 3C-like protease can bind with Remdesivir (Elfiky, 2020), Ritonavir (Stower, 2020), Lopinavir (Stower, 2020), Quercetin

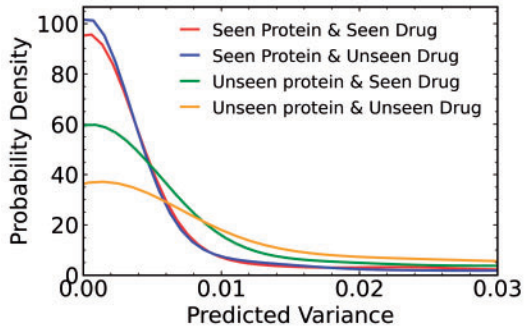

Fig. 6. Probability distributions of predicted variance of BindingDB dataset. The binary test data is divided into 'seen' and 'unseen' depending on whether the protein and drug are observed in the training dataset

Table 3. Drug–protein interaction prediction results of antiviral drugs and 3C-like protease of SARS-CoV-2

Molecules	Predicted probability	Clinical approved
Baricitnib	0.9826	Favalli *et al.* (2020)
Quercetin	0.9449	Sargiacomo *et al.* (2020)
Ritonavir	0.8778	Stower (2020)
Remdesivir	0.8027	Elfiky (2020)
Lopinavir	0.7159	Stower (2020)
Aspirin	0.1335	

Table 4. Drug–protein interaction prediction results of antiviral drugs and RNA-dependent RNA polymerase of SARS-CoV-2

Molecules	Predicted probability	Clinical approved
Ivermectin	0.9956	Caly *et al.* (2020)
Ritonavir	0.9884	Stower (2020)
Remdesivir	0.9650	Elfiky (2020)
Lopinavir	0.8469	Stower (2020)
Daclatasvir	0.6054	Lythgoe and Middleton (2020)
Aspirin	0.0708	

(Sargiacomo *et al.*, 2020) and Baricitnib (Favalli *et al.*, 2020). Table 4 shows that RNA polymerase can bind with Remdesivir (Elfiky, 2020), Ritonavir (Stower, 2020), Lopinavir (Stower, 2020), Daclatasvir (Lythgoe and Middleton, 2020) and Ivermectin (Caly *et al.*, 2020). These drug molecules have been estimated as the potential drugs for SARS-Cov-2 through clinical trials (Caly *et al.*, 2020; Elfiky, 2020; Favalli *et al.*, 2020; Lythgoe and Middleton, 2020; Sargiacomo *et al.*, 2020; Stower, 2020). On the other hand, if we study weakly related drugs such as aspirin, the result shows the small interaction score between protein. These prediction results from proposed model, which correspond with the experimental results, verify the validity of our proposed model in predicting the new drugs in the drug discovery pipeline.

4 Conclusion

In this study, we present a novel Bayesian deep learning framework with a pretrained protein sequence model to predict drug-protein interactions. Experiments on three public datasets demonstrate that our proposed model consistently outputs increased prediction accuracies. Our estimation of model performance shows that BNNs are highly robust to additive noise, which explains the superior performances of the proposed model. Furthermore, from the prediction uncertainty of our model outputs, one can evaluate the confidence level, which can then be used to screen the dataset for unreliable data points.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No. 2016R1A3B1908660).

Author contributions

All authors contributed to construct the concept and initialize the project. Q.K. and W.J. made the program. All authors participated in the discussion of the results. Q.K. and W.J. wrote the manuscript. All authors reviewed the manuscript.

Conflict of Interest: The authors declare no competing interests.

Data availability

The code is available at https://github.com/QHwan/PretrainDPI.

References

Bapst,V. *et al.* (2020) Unveiling the predictive power of static structure in glassy systems. *Nat. Phys.*, **16**, 448–454.

Battaglia,P. *et al.* (2016) Interaction networks for learning about objects, relations and physics. In: *Advances in Neural Information Processing Systems*. pp. 4502–4510.

Brigato,L. and Iocchi,L. (2020) A close look at deep learning with small data. https://arxiv.org/abs/2003.12843.

Caly,L. *et al.* (2020) The FDA-approved drug ivermectin inhibits the replication of SARS-CoV-2 *in vitro*. *Antiviral Res.*, **178**, 104787.

Cho,K. *et al.* (2014) On the properties of neural machine translation: encoder-decoder approaches. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. pp. 103–111.

Devlin,J. *et al.* (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04085.

Elfiky,A.A. (2020) Ribavirin, remdesivir, sofosbuvir, galidesivir, and tenofovir against SARS-CoV-2 RNA dependent RNA polymerase (RdRp): a molecular docking study. *Life Sci*, **253**, 117592.

Favalli,E.G. *et al.* (2020) Baricitinib for COVID-19: a suitable treatment? *Lancet Infect. Dis.*, **20**, 1012–1013.

Fokoue,A. *et al.* (2016) Predicting drug-drug interactions through large-scale similarity-based link prediction. In: *International Semantic Web Conference*. pp. 774–789.

Gal,Y. and Ghahramani,Z. (2015) Bayesian convolutional neural networks with bernoulli approximate variational inference. https://arxiv.org/abs/1506.02158.

Gal,Y. and Ghahramani,Z. (2016) Dropout as a bayesian approximation: Representating model uncertainty in deep learning. https://arxiv.org/abs/1506.02142.

Gao,K.Y. *et al.* (2018) Interpretable drug target prediction using deep neural representation. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.

He,K. *et al.* (2015) Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1017–1024.

He,T. *et al.* (2017) SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines. *J. Chemiinform.*, **9**, 24.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Hu,W. *et al.* (2020) Strategies for pre-training graph neural networks. https://arxiv.org/abs/1905.12265.

Jiang,M. *et al.* (2020) Drug-target affinity prediction using graph neural network and contact maps. *RSC Adv*, **10**, 20701–20712.

Karimi,M. *et al.* (2019) DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, **35**, 3329–3338.

Kendall,A. and Gal,Y. (2017) What uncertainties do we need in bayesian deep learning for computer vision? https://arxiv.org/abs/1703.04977.

Kim,Q. *et al.* (2020) GCIceNet: a graph convolutional network for accurate classification of water phases. *Phys. Chem. Chem. Phys.*, **22**, 26340–26350.

Kingma,D.P. and Ba,J. (2014) Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980.

Kipf,T.N. and Welling,M. (2017) Semi-Supervised Classification with Graph Convolutional Networks. https://arxiv.org/abs/1609.02907.

Kornblith,S. *et al.* (2019) Do better imagenet models transfer better? https://arxiv.org/abs/1805.08974.

Kwon,Y. *et al.* (2020) Uncertainty quantification using Bayesian neural networks in classification: application to ischemic stroke lesion segmentation. *Comput. Stat. Data Anal.*, **142**, 106816.

Landrum,G. (2006) RDKit: Open-source cheminformatics. http://rdkit.org.

Lee,I. *et al.* (2019) DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Comput. Biol.*, **15**, e1007129.

Liebler,D.C., and Guengerich,F.P. (2005) Elucidating mechanisms of drug-induced toxicity. *Nat. Rev. Drug Discov.*, **4**, 410–420.

Lim,J. *et al.* (2019) Predicting drug-target interaction using a novel graph neural network with 3d structure-embedded graph representation. *J. Chem. Inf. Model.*, **59**, 3981–3988.

Liu,H. *et al.* (2015) Improving compound-protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, **31**, i221–i229.

Liu,T. *et al.* (2007) BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.*, **35**, D198–D201.

Lythgoe,M.P. and Middleton,P. (2020) Ongoing clinical trials for the management of the COVID-19 pandemic. *Trends Pharmacol. Sci.*, **41**, 363–382.

Mahajan,D. *et al.* (2018) Exploring the limits of weakly supervised pretraining. https://arxiv.org/abs/1805.00932.

Mizutani,S. *et al.* (2012) Relating drug-protein interaction network with drug side effects. *Bioinformatics*, **28**, i522–i528.

Morrone,J.A. *et al.* (2020) Combining docking pose rank and structure with deep learning improves protein-ligand binding mode prediction over a baseline docking approach. *J. Chem. Inf. Model.*, **60**, 4170–4179.

Öztürk,H. *et al.* (2018) DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics*, **34**, i821–i829.

Paszke,A. *et al.* (2019) PyTorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.*, **32**, 8024–8035.

Pushpakom,S. *et al.* (2019) Drug repurposing: progress, challenges and recommendations. *Nat. Rev. Drug Discov.*, **18**, 41–58.

Rives,A. *et al.* (2019) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. https://www.biorxiv.org/content/10.1101/622803v3.

Ryu,S. *et al.* (2019) A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chem. Sci.*, **10**, 8438–8446.

Sargiacomo,C. *et al.* (2020) COVID-19 and chronological aging: senolytics and other anti-aging drugs for the treatment or prevention of corona virus infection? *Aging*, **12**, 6511–6517.

Shin,B. *et al.* (2019) Self-attention based molecule representation for predicting drug-target interaction. *Proc. Mach. Learn. Res.*, **106**, 230–248.

Stower,H. (2020) Lopinavir-Ritonavir in severe COVID-19. *Nat. Med.*, **26**, 465.

Tsubaki,M. *et al.* (2019) Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**, 309–318.

UniProt Consortium (2015) UniProt: a hub for protein information. *Nucleic Acids Res.*, **43**, D204–D212.

Vamathevan,J. *et al.* (2019) Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.*, **18**, 463–477.

Vaswani,A. *et al.* (2017) Attention is all you need. *Adv. Neural Inf. Process. Syst.*, **301**.

Villegas-Morcillo,A. *et al.* (2020) Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, btaa701, 162–170.

Wang,Y.-B. *et al.* (2020) A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network. *BMC Med. Inform. Decis. Mak.*, **20**, 49.

Weininger,D. (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, **28**, 31–36.

Wen,M. *et al.* (2017) Deep-learning-based drug-target interaction prediction. *J. Proteome Res.*, **16**, 1401–1409.

Winter,R. *et al.* (2019) Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.*, **10**, 1692–1701.

Wu,Z. *et al.* (2018) MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, **9**, 513–530.

Xie,T. and Grossman,J.C. (2018) Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, **120**, 145301.

Xue,H. *et al.* (2018) Review of drug repositioning approached and resources. *Int. J. Biol. Sci.*, **14**, 1232–1244.

Zhang,H. *et al.* (2019) DeepBindRG: a deep learning based method for estimating effective protein-ligand affinity. *PeerJ*, **7**, e7362.

Zheng,S. *et al.* (2020) Predicting drug protein interaction using quasi-visual question answering system. *Nat. Mach. Intell.*, **2**, 134–140.