Journal of Cheminformatics

RESEARCH ARTICLE Open Access

Deep learning integration of molecular and interactome data for protein–compound interaction prediction

Narumi Watanabe, Yuuto Ohnuki and Yasubumi Sakakibara*

Abstract

Motivation: Virtual screening, which can computationally predict the presence or absence of protein–compound interactions, has attracted attention as a large-scale, low-cost, and short-term search method for seed compounds. Existing machine learning methods for predicting protein–compound interactions are largely divided into those based on molecular structure data and those based on network data. The former utilize information on proteins and compounds, such as amino acid sequences and chemical structures; the latter rely on interaction network data, such as protein–protein interactions and compound–compound interactions. However, there have been few attempts to combine both types of data in molecular information and interaction networks.

Results: We developed a deep learning-based method that integrates protein features, compound features, and multiple types of interactome data to predict protein–compound interactions. We designed three benchmark datasets with different difficulties and applied them to evaluate the prediction method. The performance evaluations show that our deep learning framework for integrating molecular structure data and interactome data outperforms state-of-the-art machine learning methods for protein–compound interaction prediction tasks. The performance improvement is statistically significant according to the Wilcoxon signed-rank test. This finding reveals that the multi-interactome data captures perspectives other than amino acid sequence homology and chemical structure similarity and that both types of data synergistically improve the prediction accuracy. Furthermore, experiments on the three benchmark datasets show that our method is more robust than existing methods in accurately predicting interactions between proteins and compounds that are unseen in training samples.

Keywords: Protein–compound interaction, Deep learning, Heterogeneous interaction network, Integration

Introduction

Most compounds that act as drugs bind to target proteins that can cause disease, and these compounds can control their functions. Therefore, it is necessary when developing new drugs to search for compounds that can interact with the target protein, and this process must be performed efficiently. However, determining the interaction of a large number of protein–compound pairs via experiments is expensive in terms of time and cost. Virtual screening that can computationally classify the presence or absence of protein–compound interactions has attracted attention as a large-scale, low-cost, short-term search method for seed compounds. In particular, machine learning for virtual screening is considered to be applicable to a wide variety of proteins and compounds.

Machine learning-based methods for predicting protein–compound interactions are largely divided into those based on molecular structure data and those based on network data. The former use protein and compound data represented in amino acid sequences and chemical

*Correspondence: yasu@bio.keio.ac.jp
Department of Biosciences and Informatics, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

structure formulas, and they can be applied to proteins when a docking simulation cannot be performed because the 3D structure is unknown. In our previous studies [1–3], we performed binary classification for predicting protein–compound interactions using a support vector machine (SVM) on an interaction dataset downloaded from DrugBank (a database that contains information on existing drug compounds) [4]. A prediction accuracy of 85.1% was achieved. Based on this result, we developed COPICAT, a comprehensive prediction system for protein–compound interactions, which enabled us to search for lead compounds from a large compound database, PubChem [5], consisting of tens of millions of compounds.

Deep learning, a method developed in the field of machine learning, has been applied in a variety of fields in recent years because it achieves high prediction accuracy in fields such as image recognition, speech recognition, and compound activity prediction [6]. Deep learning-based protein–compound interaction prediction methods have been developed based on molecular structure data [7–10]. However, as these existing deep learning-based methods utilize information based on only amino acid sequences and chemical structures, the functional properties of proteins and compounds have not yet been incorporated into prediction models.

The other type of machine learning approach for protein–compound interaction prediction is based on network data. An interaction network is commonly used to comprehensively represent interactions between molecules. For example, the protein–protein interaction network represents the relationships among physically interacting proteins. In the protein–protein interaction network, a node is a protein, and an edge is drawn between a pair of proteins that interact with each other.

Some previous studies incorporated data from multiple interaction networks to predict molecular interactions. For instance, multi-modal graphs to handle three types of interactions have been proposed: protein–protein, protein–drug, and polypharmacy side effects. A deep learning method for multi-modal graphs, Decagon [11], was proposed to predict polypharmacy side effects. DTINet [12] and NeoDTI [13] were also designed and developed as graph-based deep learning frameworks to integrate heterogeneous networks for drug–target interaction (DTI) predictions and drug repositioning. In particular, NeoDTI exhibits substantial improvement in performance over other state-of-the-art prediction methods based on multiple interaction network data.

In addition to predicting protein–compound interactions, several studies have predicted other types of molecular interactions. Protein–protein interactions induce many biological processes within a cell, and experiential and computational methods have been developed to identify various protein–protein interactions. High-throughput experimental methods, such as yeast two-hybrid screening, were developed to discover and validate protein–protein interactions on a large scale. Computational methods for protein–protein interaction predictions employ various machine learning methods, such as SVM with feature extraction engineering [14]. The recurrent convolutional neural network (CNN), which is a deep learning method, was applied to sequence-based prediction for protein–protein interactions [15]. Compounds that can interact with each other are often represented as compound–compound interactions (also known as chemical–chemical interactions), and interactive compounds tend to share similar functions. Compound–compound interactions, called drug–drug interactions, can be used to predict side effects based on the assumption that interacting compounds are more likely to have similar toxicity [16]. A computational approach to compound–compound interaction predictions has been studied with various machine learning methods, including end-to-end learning with a CNN based on the SMILES representation [17].

The purpose of this study was to improve prediction accuracy by integrating molecular structure data and heterogeneous interactome data into a deep learning method for predicting protein–compound interactions. In addition to the molecular information (amino acid sequence and chemical structure) itself, protein–protein interaction network data with similar reaction pathways or physical direct binding and compound network data linking compounds with similar molecular activities are incorporated into the deep learning model as multi-interactome data. To the best of our knowledge, there are no deep learning-based solutions for predicting protein–compound interactions that integrate multiple heterogeneous interactome data along with the direct input of amino acid sequences and chemical structures.

This study proposes a method for predicting protein–compound (drug-target) interactions by combining protein features, compound features, and network context for both proteins and compounds. The network context is in the form of protein–protein interactions from the STRING database [18], and the compound–compound interactions are derived from the STITCH database [19]. The protein–protein interaction network and compound–compound interaction network are processed using node2vec [20] to generate feature vectors for each protein node and each compound node in the interaction networks in an unsupervised manner. Each network-based representation is then combined with additional features extracted from a CNN applied to the amino acid sequence of a protein and from the extended-connectivity

fingerprint (ECFP) of a compound. The final combined protein and compound representations are used to make a protein–compound interaction prediction with an additional fully connected layer. The overall learning architecture is illustrated in Fig. 1.

We designed three benchmark datasets with different difficulties and evaluated the performance of the model using these datasets. In these performance evaluations, we demonstrated that integrating the molecular structure data and multiple heterogeneous interactome data synergistically improves the accuracy of protein–compound interaction prediction. Furthermore, performance comparisons with state-of-the-art deep learning methods based on molecular information [10] and those based on interaction network data [13], as well as the traditional machine learning methods (SVM and random forest), showed that our model yields significant performance improvements for the most important evaluation measures: area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPRC), F-measure and accuracy. Other methods have low values for these measures. The improvement was verified as statistically significant according to the Wilcoxon signed-rank test. Finally, we analysed whether protein–protein interactions capture a different perspective than amino acid sequence homology and whether compound–compound interactions capture a different perspective than chemical structure similarity.

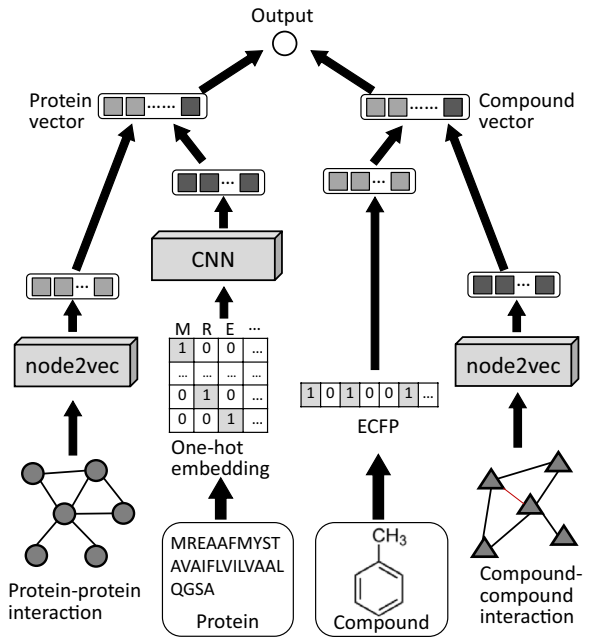

Fig. 1 Deep learning architecture that integrates molecular structure data and interactome data to predict protein–compound interactions. It integrates graph- and sequence-based representations for the target protein and compound. The amino acid sequence of the protein input was embedded into a one-hot vector of 20 dimensions in height. The ECFP representation of the compound input was embedded into a 1024-dimensional vector. The feature vectors were also extracted from the protein–protein and compound–compound interaction network using node2vec, a feature representation learning method for graphs. These feature vectors were combined as a protein vector and a compound vector. The interaction was predicted in the output unit

Methods

1D-CNN for encoding protein data

First, the protein data were applied to a one-dimensional convolutional neural network (1D-CNN). For the protein input, a one-hot vector was used for the distributed representation of an amino acid sequence of 20 dimensions at a height and width of 5762 dimensions with the maximum length of amino acid sequences.

An amino acid sequence is a linear structure (1-D grid). In this study, a filter (kernel) with a 1D convolution operation was applied to the linear structure. Here, a "1D" convolutional operation for linear structures is interpreted as scanning the input structure in only one direction along the linear structure with a filter of the same height (dimension) as that of the distributed representation of the input.

One-dimensional (1D) convolutional layer

We denote an input vector sequence that corresponds to the one-hot vector representation of an amino acid sequence by $A = [\boldsymbol{a}_1^{(1)}, \boldsymbol{a}_2^{(1)}, \ldots, \boldsymbol{a}_q^{(1)}]$ (as illustrated in Fig. 1). For a filter function in the l-th hidden layer of the CNN, the input is the set of feature maps in the (l-1)-th hidden layer $\boldsymbol{x}_{i:i+r-1,j}^{(l-1)} = c_{i,j}^{(l-1)} \in \mathbb{R}^{m \times n}$, where r is the size of the filter, m is the size of the feature map, and n is the number of feature maps. The output of the k-th filter is a feature map of the l-th layer $c_i^{(l,k)} \in \mathbb{R}^m$, which is defined as follows:

$$c_i^{(l,k)} = f\left(\mathbf{W}^{(l,k)} c_{i,j}^{(l-1)} + \mathbf{b}^{(l,k)}\right),$$

where f is an activation function (Leaky-ReLU), $\mathbf{W}^{(l,k)} \in \mathbb{R}^{m \times n \times d}$ is the weight matrix of the k-th filter in the l-th convolutional layer, and $\mathbf{b}^{(l,k)}$ is the bias vector. The average-pooling mechanism is applied to every convolution output. To obtain the final output $\boldsymbol{y} = \{y^{(t,1)}, y^{(t,2)}, \ldots, y^{(t,s)}\}$, global max-pooling is used as follows:

$$y^{(t,k)} = \max_i (c_i^{(t,k)}),$$

where t represents the last layer of the CNN, and s represents the number of filters in the last layer.

Extended-connectivity fingerprint (ECFP) for compound data

The extended-connectivity fingerprint (ECFP, also known as the circular fingerprint or Morgan fingerprint) [21] is the most commonly used feature representation for representing a property of the chemical structure of a compound. This algorithm first searches the partial structures around each atom recurrently, assigns an integer identifier to each partial structure and then expresses this as a binary vector using a hash function. Potentially, an infinite number of structures exist in the chemical space; consequently, the ECFP requires vectors with a large number of bits (usually 1024−2048 bits). In this study, we employed an ECFP with 1024 bits as the feature representation for the chemical structure of a compound.

Feature representation learning for protein–protein and compound–compound interactions

A protein–protein interaction network that connects physically interacting proteins and a compound–compound interaction network that connects compounds with similar molecular activities were input as multi-interactome data. First, each network was represented as a graph. A node is a protein in the protein–protein network and a compound in the compound–compound network. An edge is drawn between a pair of proteins (compounds) that interact with each other. By applying this graph to "node2vec" [20], the feature vector of each node was obtained in an unsupervised manner; node2vec is a deep learning method that learns the feature representation of nodes in a graph and obtains a feature vector for each node. Node2vec is a graph-embedding algorithm that can be applied to any type of graph, and it can learn a feature vector such that nodes that are nearby on the graph are also close in the embedded feature space. In other words, the inner product of the feature vectors of the nearby nodes is high. It is known that the accuracy of the node classification task and the link prediction task using the obtained feature representations of nodes is higher than that of the existing methods.

The node2vec algorithm was applied to the protein–protein interaction network and the compound–compound interaction network. Using a protein and a compound as vertices, the interaction networks were converted into graphs with edge weights based on the reliability of the experimental data and the similarity in molecular activity. Node2vec (version 0.2.2) from the Python library, which implemented the node2vec algorithm, was applied to the converted graph. The node2vec parameters used the default values (embedding dimensions: 128; number of nodes searched in one random walk: walk_length = 80; number of random walks per node: num_walk = 10; control of probability of revisiting a walk node: p = 1; control of the search speed and range: r = 1; whether to reflect the graph weight: weight_key = weight).

Let a protein–protein interaction network be expressed by a weighted graph $G_{protein} = \left(V_{protein}, E_{protein}\right)$ and a compound–compound interaction network be expressed by a weighted graph $G_{compound} = \left(V_{compound}, E_{compound}\right)$. By applying node2vec to these graphs, the feature representations can be obtained and are denoted as $\boldsymbol{N}_{protein} = \mathrm{node2vec}\left(G_{protein}\right) \in \mathbb{R}^d$ and $\boldsymbol{N}_{compound} = \mathrm{node2vec}\left(G_{compound}\right) \in \mathbb{R}^d$ for a dimension of d.

Deep learning model structure for integrating molecular information and the interaction network

The feature vectors obtained from the 1D-CNN for the amino acid sequence and node2vec for the protein–protein interaction network were concatenated and fed to the final output layer. Similarly, the feature vectors from the ECFP for the chemical structure and node2vec for the compound–compound interaction network were concatenated and fed to the final output layer.

We designed an output layer consisting of an element-wise product calculation followed by a fully connected layer, which is an extension of cosine similarity. The architecture is illustrated in Fig. 2. First, the feature vectors for the proteins and compounds were mapped onto the same latent space with a fixed dimension d by applying fully connected layers. The similarity between the vector for proteins and the vector for compounds on the latent space was calculated by the element-wise product calculation method followed by a fully connected layer. When a pair of proteins and compounds was input, it was predicted that there was an interaction between the input pair if the similarity was higher than a predefined threshold (where the default was 0.5); if the similarity was lower, no interaction was predicted. This model is denoted as the "integrated model".

More precisely, let $\boldsymbol{a}_{protein}$ denote the feature vector output by the 1D-CNN for an amino acid sequence, and let $\boldsymbol{b}_{compound}$ denote the feature vector of the ECFP for the chemical structure of a compound. Let $\boldsymbol{N}_{protein}$ and $\boldsymbol{N}_{compound}$ denote the feature representations obtained from node2vec for the protein–protein interaction network and the compound–compound interaction network. Two feature vectors $\boldsymbol{a}_{protein}$ and $\boldsymbol{N}_{protein}$ were concatenated as one vector $\boldsymbol{v}_{protein}$ for the protein multi-modal feature. Two feature vectors $\boldsymbol{b}_{compound}$ and $\boldsymbol{N}_{compound}$ were concatenated as one vector $\boldsymbol{v}_{compound}$ for the compound multi-modal feature. The concatenated feature vectors $\boldsymbol{v}_{protein}$ and $\boldsymbol{v}_{compound}$ were mapped onto the same latent space with a fixed dimension d by

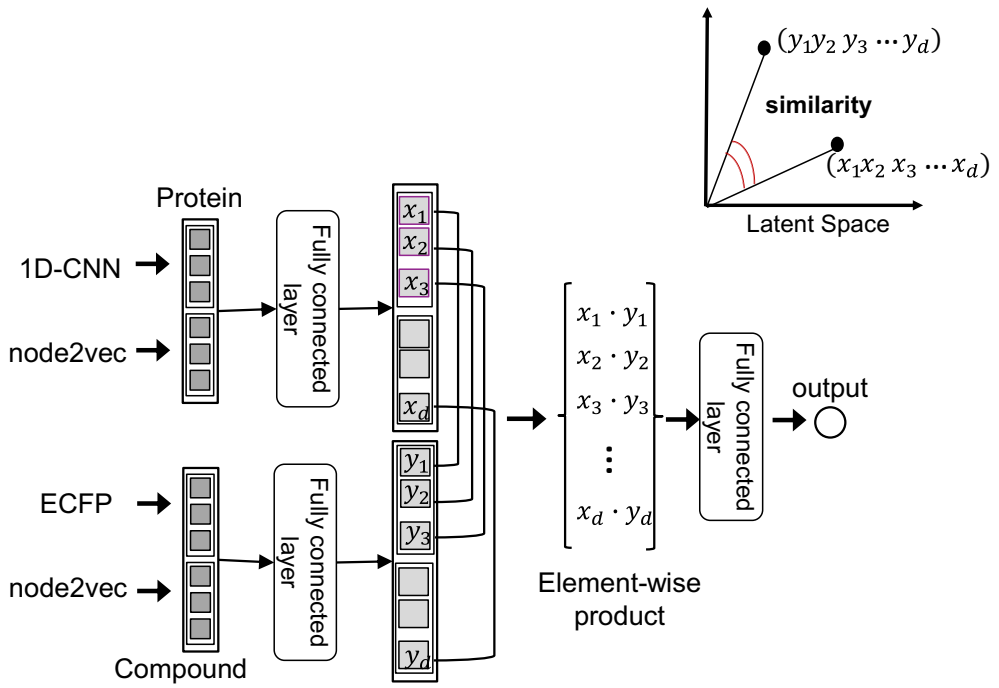

Fig. 2 The output layer architecture. The integrated model predicts the protein–compound interactions by embedding the protein and compound data from different modalities into a common latent space. The feature vectors for the proteins and compounds are mapped onto the same latent space by applying a fully connected layer. Then, their similarity in the latent space is calculated with an element-wise product calculation followed by a fully connected layer

applying the fully connected layers f and g. From this, the similarity between the two vectors for the latent space was calculated.

$$\boldsymbol{v}_{protein} = \mathrm{contact}\left(\boldsymbol{a}_{protein}, \boldsymbol{N}_{protein}\right),$$

$$\boldsymbol{v}_{compound} = \mathrm{contact}\left(\boldsymbol{b}_{compound}, \boldsymbol{N}_{compound}\right),$$

$$(x_1, x_2, \ldots, x_d) = f\left(\boldsymbol{v}_{protein}\right),$$

$$(y_1, y_2, \ldots, y_d) = g\left(\boldsymbol{v}_{compound}\right),$$

$$\mathrm{output}_{integrated} = h(x_1 \cdot y_1, x_2 \cdot y_2, \ldots, x_d \cdot y_d).$$

As described above, to handle data from different modalities, such as proteins and compounds, we adopted a method of embedding data of different modalities into a common latent space. Defining the similarity in the obtained latent space enables the measurement of the similarity between the data for different modalities. Visual semantic embedding (VSE) is a typical example of a method that handles data from different modalities and can associate images with text data in acquiring these multi-modal representations [22]. VSE was developed to generate captions from images (image captioning). The image feature and the sentence feature are linearly transformed and embedded into a common latent space.

Single-modality models

To see the effect of integrating multi-modal features, two baseline models were constructed for the performance comparison. One was based on molecular structure data and used only amino acid sequence and chemical structure information, and the other was based on interaction network data and used only protein–protein interaction and compound–compound interaction information. The single-modality model based on molecular structure data, denoted the "single-modality model (molecular)", is defined as follows:

$$(x_1, x_2, \ldots, x_n) = f\left(\boldsymbol{a}_{protein}\right),$$

$$(y_1, y_2, \ldots, y_n) = g\left(\boldsymbol{b}_{compound}\right),$$

$$\mathrm{output}_{molecule} = h\left(x_1 \cdot y_1, x_2 \cdot y_2, \ldots, x_n \cdot y_n\right),$$

and the single-modality model based on interaction network data, denoted the "single-modality model (network)", is defined as follows:

$$(x_1, x_2, \ldots, x_n) = f\left(\boldsymbol{N}_{protein}\right),$$

$$(y_1, y_2, \ldots, y_n) = g\left(\boldsymbol{N}_{compound}\right),$$

$$\text{output}_{network} = h(x_1 \cdot y_1, x_2 \cdot y_2, \ldots, x_n \cdot y_n).$$

Loss function

For the similarity output x of the model, the output value was restricted to the range 0 to 1 by the sigmoid function, and cross entropy was applied as the loss function $L(\theta)$ to calculate the training error.

Hyperparameter optimization

The hyperparameters, the number and size of the filters in the convolutional layers in the 1D-CNN, and the number of units in the fully connected output layers were optimized by the Bayesian optimization tool Optuna [23], which is an automatic hyperparameter optimization software framework specifically designed for machine learning. For the hyperparameter optimization, the validation dataset was obtained by dividing the training samples into a set for training and a set for validation.

Regularization

Regularization is important for avoiding overfitting and improving the prediction accuracy in deep learning for complex model architectures with a large number of parameters. Regularization is especially important in our deep learning model, which integrates multiple datasets of different modalities; hence, we employed several regularization methods.

We employed batch normalization [24], which allowed us to use much higher learning rates and be less careful about initialization, after each convolutional layer. We also inserted dropout [25] after the fully connected layers. Furthermore, we added an L2 regularization term to the training-loss function $L(\theta)$. When incorporating weight decay, the objective function to be optimized is as follows:

$$L(\theta) + \lambda \frac{1}{2} \sum_{w} \|w\|^2,$$

where w refers to the parameters of the entire model, and the second term of the above equation is the sum of the squared values of all the parameters divided by 2. λ is a parameter that controls the strength of regularization. Adding this term to the objective function has the effect of preventing the absolute value of the network weight from becoming too large, which helps prevent overfitting.

Comparison with existing state-of-the-art methods

The prediction performance of the proposed models was compared with that of state-of-the-art deep learning methods based on molecular structure data and interaction network data. The first method was based on a graph CNN for protein–compound prediction [10]. It employed a graph CNN for encoding chemical structures and a CNN for n-grams of amino acid sequences. The second method was NeoDTI [13], which demonstrated superior performance over other previous methods based on multiple-interaction-network data. We also compared our method with the traditional machine learning methods, namely, SVM and random forest [26], as the baseline prediction methods. These traditional methods require structured data as input. For the protein information, the 3-mer (3-residue) frequency in the amino acid sequence was used as the feature vector for 8000 dimensions. For the compound information, an ECFP with a length of 1024 and a radius of 2 was used. The radial basis function (RBF) was used as the kernel function of SVM, and all other parameters of SVM and random forest used the default values. In implementing these machine learning methods, scikit-learn (version 0.19.1) and chainer (version 5.0.0) were used.

Datasets

The protein–compound interaction data and compound–compound networks were retrieved from the database STITCH [19], and the protein–protein networks were retrieved from the database STRING [18].

Protein–compound interaction data

Protein–compound interaction data were obtained from the STITCH database [19]. STITCH contains data on the interaction of 430,000 compounds with 9.6 million proteins from 2031 species. The STITCH data sources consist of (1) structure-based prediction results, such as the genome context and co-expression; (2) high-throughput experimental data; (3) automatic text mining; and (4) information from existing databases. When a protein–compound dataset is downloaded from STITCH, a score based on the reliability is created for each of the above four items for each protein–compound pair. For the protein–compound interaction data used in this study (as a "positive" example), the threshold value for the reliability score of item (2) was set to 700, and the data with a reliability score of 700 or higher were extracted from STITCH such that interologs were eliminated and the data were composed of only experimentally reliable interactions; data

that did not meet this threshold were removed. For the STITCH data, interactions with a confidence score of 700 or more were determined based on the criterion that they were at least highly reliable [27]. Of the combinations of proteins and compounds, only pairs not stored in the STITCH database were taken as "negative" examples. In general, protein–compound pairs that are not stored in STITCH have very low confidence, with a score of 150 or less for their interaction [28]; these are thus considered to be non-interacting negative examples. The ratio of the positive and negative examples was 1 to 2.

Protein–protein interaction data

The protein–protein interaction information was obtained from the STRING database [18], which contains data for protein–protein interactions covering 24.6 million proteins from 5090 species. The STRING data sources consist of (1) experimental data; (2) pathway databases; (3) automatic text mining; (4) co-expression information; (5) neighbouring gene information; (6) gene fusion information; and (7) co-occurrence-based information. In particular, item (1) is interaction data obtained from actual experiments, which include biochemical, biophysical, and genetic experiments. These are extracted from databases organized by the BioGRID database [29] and the International Molecular Exchange (IMEx) consortium [30]. When the protein–protein interaction data from STRING were downloaded, a score based on the reliability was created for each of the above seven items for each protein–protein pair. Regarding the protein–protein interaction network, the threshold value for the reliability score of item (1) was set to 150. Data that did not satisfy this criterion were removed.

Compound–compound interaction data

The compound–compound interaction data were also obtained from the STITCH database. The compound–compound interaction data in STITCH are based on (1) the chemical reactions obtained from the pathway databases; (2) structural similarity; (3) association with previous literature; and (4) correspondence between the compounds based on molecular activity similarity. For the similarity of the molecular activities in item (4), the activity data obtained by screening the model cell line NCI60 were used. When the compound–compound interaction data were downloaded from STITCH, a score based on the reliability was created for each of the above four items for each compound pair. For the compound–compound interaction data used in this study, the threshold value for the reliability score in item (4) was set to 150. Data that did not satisfy this criterion were removed.

Construction of the baseline, unseen compound-test, and hard datasets for evaluation

From the STITCH and STRING databases, a total of 22,881 protein–compound interactions, 175,452 protein–protein interactions and 69,231 compound–compound interactions were downloaded. Using the downloaded dataset in which the protein–protein interaction, compound–compound interaction and protein–compound interaction data were all available, the three types of datasets below were constructed to perform five-fold cross-validation.

In typical k-fold cross-validation, all positive and negative examples are randomly split into k folds. One of them is used as a test sample, and the remaining $k-1$ folds are used as training samples; then, the k results obtained are averaged. We call the cross-validation dataset the *baseline dataset*. As more difficult and more practical tasks, we constructed two more cross-validation datasets, called the *unseen compound-test dataset* and the *hard dataset*. In the unseen compound-test dataset, we split the data into k folds such that none of the folds contained the same compounds as the others. In the unseen compound-test dataset, the compounds in the test sample did not appear in the training sample. In other words, the interaction of new (unseen) candidate compounds with the target proteins must be accurately predicted. In the hard dataset, we split the data into k folds such that none of the folds contain the same proteins and compounds as the others. In the hard dataset, neither the proteins nor the compounds in the test sample appear in the training sample. In other words, interactions in which neither the proteins nor the compounds are found in the training sample must be accurately predicted.

Results

The following measures were used for the performance evaluation criteria: AUROC, AUPRC, F-measure, and accuracy.

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}},$$

$$\text{Accuracuy} = \frac{TP + TN}{TP + FP + FN + TN},$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives; the recall is

defined by $TP/(TP+FN)$, and the precision is defined by $TP/(TP+FP)$.

The experiment was performed on a computer equipped with an Intel(R) Xeon CPU (E5-2698-v4, 2.20 GHz, 512 GB memory) with one NVIDIA Tesla V100 GPU. The computational time of our deep learning algorithm was approximately 27 s (CPU time) per epoch in the process of constructing the integrated model. The number of epochs required for the learning process to converge varied greatly from tens to hundreds.

Effectiveness of integrating molecular structure data and interaction network data

The performance of our three models was evaluated to determine the effectiveness of integrating the molecular structure data and the interaction network data. The results derived from the three datasets are shown in Tables 1, 2 and 3. In the tables, the mean and standard deviation (SD) for the five folds are shown. Furthermore, the symbol "*" indicates that there was a significant difference compared with the integrated model based on the Wilcoxon signed-rank test, with a p-value $p < 0.05$.

Compared with the two single-modality models, the integrated model significantly improved the prediction accuracy in all evaluation measures. For example, in terms of AUPRC, which is a more informative evaluation index in a dataset that is imbalanced between positive and negative samples, the integrated model showed significant improvements of 3.0%, 7.1 and 8.3% over the single-modality model (molecular) and 3.7%, 10.9 and

Table 1 Performance comparison of three proposed models with existing methods on the baseline dataset

	AUROC	AUPRC	F-measure	Accuracy
Integrated model (molecular + network)	0.972 ± 0.004	0.954 ± 0.005	0.900 ± 0.006	0.933 ± 0.004
Single-modality model (molecular)	0.956 ± 0.004*	0.927 ± 0.006*	0.868 ± 0.009*	0.911 ± 0.006*
Single-modality model (network)	0.947 ± 0.008*	0.920 ± 0.010*	0.853 ± 0.015*	0.904 ± 0.009*
Graph CNN-based method [10]	0.917 ± 0.006*	0.850 ± 0.006*	0.794 ± 0.014*	0.864 ± 0.008*
NeoDTI [13]	0.956 ± 0.005*	0.905 ± 0.016*	0.872 ± 0.006*	0.917 ± 0.004*
SVM	0.805 ± 0.009*	0.651 ±0.012*	0.743 ± 0.012*	0.837 ± 0.006*
Random forest	0.873 ± 0.009*	0.767 ± 0.015*	0.837 ± 0.012*	0.895 ± 0.007*

Table 2 Performance comparison on the unseen compound-test dataset

	AUROC	AUPRC	F-measure	Accuracy
Integrated model (molecular + network)	0.890 ± 0.039	0.842 ± 0.050	0.727 ± 0.085	0.843 ± 0.038
Single-modality model (molecular)	0.869 ± 0.027	0.786 ± 0.023*	0.657 ± 0.053	0.802 ± 0.017
Single-modality model (network)	0.831 ± 0.053	0.759 ± 0.055*	0.661 ± 0.073*	0.809 ± 0.030*
Graph CNN-based method [10]	0.804 ± 0.037*	0.679 ± 0.031*	0.637 ± 0.027	0.773 ± 0.009*
NeoDTI [13]	0.823 ± 0.067	0.773 ± 0.064*	0.621 ± 0.062*	0.805 ± 0.024*
SVM	0.765 ± 0.020*	0.603 ± 0.029*	0.689 ± 0.029	0.810 ± 0.016
Random forest	0.770 ± 0.023*	0.635 ± 0.026*	0.697 ± 0.036	0.828 ± 0.014

Table 3 Performance comparison on the hard dataset

	AUROC	AUPRC	F-measure	Accuracy
Integrated model (molecular + network)	0.882 ± 0.035	0.834 ± 0.041	0.714 ± 0.064	0.836 ± 0.030
Single-modality model (molecular)	0.851 ± 0.023	0.770 ± 0.023*	0.662 ± 0.038*	0.806 ± 0.020*
Single-modality model (network)	0.780 ± 0.051*	0.706 ± 0.040*	0.601 ± 0.057*	0.784 ± 0.023*
Graph CNN-based method [10]	0.707 ± 0.038*	0.563 ± 0.083*	0.427 ± 0.132*	0.719 ± 0.043*
NeoDTI [13]	0.790 ± 0.039*	0.715 ± 0.046*	0.297 ± 0.084*	0.719 ± 0.018*
SVM	0.652 ± 0.019*	0.500 ± 0.023*	0.481 ± 0.044*	0.755 ± 0.012*
Random forest	0.605 ± 0.033*	0.452 ± 0.046*	0.364 ± 0.075*	0.728 ± 0.026*

18.1% over the single-modality model (network) in the baseline dataset, unseen compound-test dataset and hard dataset, respectively. These results demonstrate that integrating multiple heterogeneous interactome data with molecular structure data brought a synergistic effect in improving the accuracy of protein–compound interaction prediction.

Performance comparison with other existing methods

The prediction performance of our three models was compared with that of state-of-the-art deep learning methods and traditional machine learning methods based on molecular structure data and interaction network data. The results for the three datasets are shown in Tables 1, 2 and 3.

The integrated model yielded superior prediction performance compared with the other existing methods. For the baseline dataset, the integrated model achieved significant improvements compared with the graph CNN-based method [10], NeoDTI [13] and the traditional machine learning methods (SVM and random forest) (Table 1). In fact, Wilcoxon signed-rank test [31] verification showed that the difference in performance was statistically significant, with a p-value $p < 0.05$, thereby proving the superiority of the integrated model.

For the unseen compound-test and hard datasets, a more marked difference in the performance of the integrated model was confirmed. We compared the integrated model with the graph CNN-based method and NeoDTI in terms of AUROC, AUPRC and F-measure. The integrated model greatly outperformed the others, with significant improvements (10.7% in terms of AUROC, 24.0% in terms of AUPRC and 14.1% in terms of F-measure for the unseen compound-test dataset, and 24.8% in terms of AUROC, 48.1% in terms of AUPRC and 67.2% in terms of F-measure for the hard dataset) over the graph CNN-based method. Compared with NeoDTI, significant improvements were also confirmed: 8.1% in terms of AUROC, 8.9% in terms of AUPRC and 17.1% in terms of F-measure for the unseen compound-test dataset, and 11.6% in terms of AUROC, 16.6% in terms of AUPRC and 140.4% in terms of F-measure for the hard dataset. Based on the above results, the integrated model can predict protein–compound interactions with stable accuracy, regardless of the difficulty of the dataset and the types of proteins and compounds that constitute the test data, compared with other existing methods. This is due to the use of features based on sequence information and compound structure information by the integrated model and features obtained from the interaction network, as well as the effect of using the element-wise product of the protein and compound feature vectors in the output layer.

The single-modality model also yielded superior prediction performance compared with that of the existing methods using the same-modality input data. The graph CNN-based method [10] yields a compound feature vector by converting the chemical structure into a graph and applying it to the graph CNN, and it generates a protein feature vector by splitting the amino acid sequence into *n*-grams and applying it to the CNN. Therefore, the graph CNN-based method can be defined as having the same molecular structure data-based prediction model as the single-modality model (molecular). For the baseline dataset, the unseen compound-test dataset and the hard dataset, the single-modality model (molecular) outperformed the graph CNN-based method. As an example, the single-modality model (molecular) achieved an improvement of 20.4% in terms of AUROC, 36.8% in terms of AUPRC and 55.0% in terms of F-measure for the hard dataset over the graph CNN-based method (Table 3). Based on this result, in protein–compound interaction prediction, it is sufficient to use the ECFP as a feature representation for the compound structure; in contrast, in the graph CNN-based method, the compound structure is converted into a graph structure, and a graph CNN is applied.

NeoDTI takes protein–protein interaction and compound–compound interaction information as input and predicts whether an edge is drawn between the compound and protein nodes by learning to reconstruct the network. Therefore, NeoDTI can be defined as an interaction network-based prediction model, which is the same as the single-modality model (network). The difference is that the single-modality model (network) first uses unsupervised deep learning (node2vec) to automatically learn feature representations for nodes in the given heterogeneous interaction networks and then applies supervised learning to predict protein–compound interactions based on the learned features, while NeoDTI simultaneously learns the feature representations of nodes and protein–compound interactions in a supervised manner. In the three datasets, the prediction performance of the single-modality model (network) was comparable to that of NeoDTI.

Discussion

To interpret the accuracy improvement obtained by integrating multi-interactome data with molecular structure data, which was shown in the previous section, we analysed whether the protein–protein interaction captured a different perspective than amino acid sequence homology and whether the compound–compound interaction captured a different perspective than chemical structure similarity. More concretely, we investigated the relationship between the amino acid sequence homology and similarity of proteins in the protein–protein interaction

network, as well as the relationship between the chemical structure similarity and the similarity in the compound–compound interaction network.

For every pair of proteins in the dataset used in the experiments, the amino acid sequence similarity was calculated using DIAMOND, and the cosine similarity between two vectors of the pair output by node2vec using the protein–protein interaction network was calculated. All of the protein pairs were plotted with the amino acid sequence similarity on the x-axis and the cosine similarity in the protein–protein interaction network on the y-axis. The scatter plot is shown in Fig. 3 (top). Similarly, for every pair of compounds, the Jaccard coefficient of the ECFPs of the two compounds and the cosine similarity between the two vectors output by node2vec using a compound–compound interaction network were calculated. All of the compound pairs were plotted with the Jaccard coefficient on the x-axis and the cosine similarity in the compound–compound interaction network on the y-axis, as depicted in Fig. 3 (bottom). However, no clear correlation was observed these scatter plots. In fact, the correlation coefficients for each scatter plot were 0.127 and 0.0346, respectively. In other words, it was confirmed that the amino acid sequence similarity and the similarity in the protein–protein interaction network were not proportional; it was also confirmed that the chemical structure similarity and the similarity in the compound-compound interaction network were not proportional. Therefore, we concluded that the protein–protein interaction network captured a different perspective than the amino acid sequence homology and compensated for it. The compound–compound interactions captured a different perspective than the chemical structure similarity and compensated for it.

For example, the protein "5-hydroxytryptamine (serotonin) receptor 6, G protein-coupled (HTR6)" and the compound "Mesulergine" in the test sample in the "hard dataset" have a positive interaction [32], and our model succeeded in correctly predicting it. Nevertheless, the single-modality model (molecular) and graph CNN-based method failed to predict a positive interaction; that is, both predicted that the pair would not interact. The most similar protein–compound pair in the training sample to the pair HTR6 and Mesulergine was the protein "adrenoceptor alpha 2A (ADRA2A)" and the compound "Pergolide" [33]. The protein ADRA2A and the compound Pergolide exhibited a positive interaction in the training sample. The sequence similarity score between HTR6 and ADRA2A was rather low at 100.5, but the similarity of the two proteins in the protein–protein interaction network was relatively high at 0.805. Part of the protein–protein interaction network around HTR6 and ADRA2A is displayed in Fig. 4 (left). Similarly, the Jaccard coefficient of the ECFPs between Mesulergine and Pergolide is relatively low at 0.273 (in general, compound pairs with Jaccard coefficients for ECFPs below 0.25 are considered not to have chemically similar structures [34]), but the cosine similarity of the two compounds in the compound–compound interaction network is high at 0.735. Part of the compound-compound interaction network around Mesulergine and Pergolide is displayed in Fig. 4 (right).

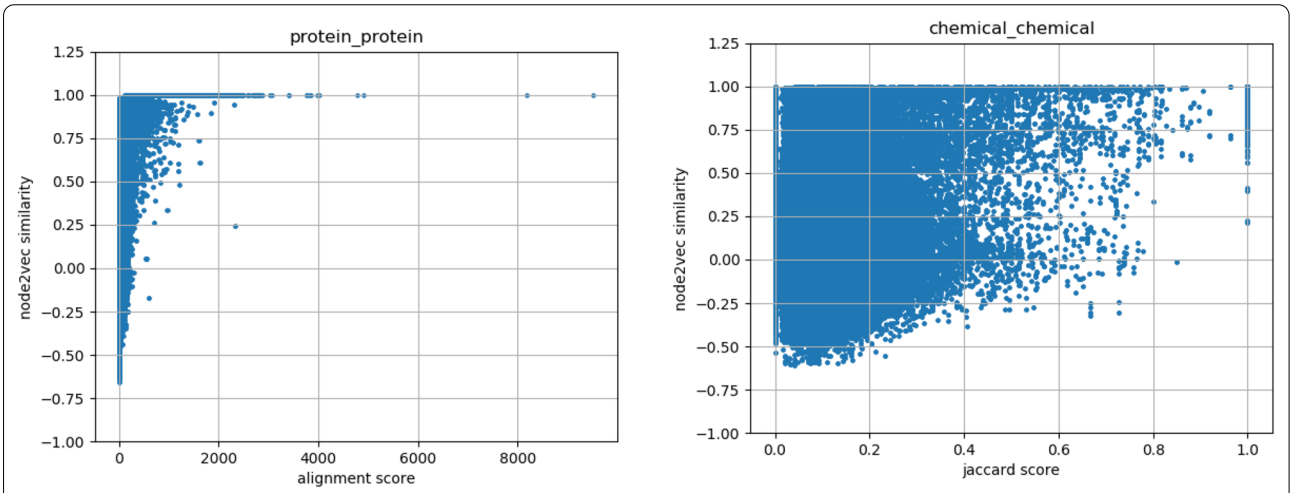

Fig. 3 (Left) Relationship between the amino acid sequence similarity and the similarity in the protein–protein interaction network. (Right) Relationship between the chemical-structure similarity and the similarity in the compound–compound interaction network. The amino acid sequence similarity was calculated using DIAMOND, and the chemical structure similarity was calculated as the Jaccard coefficient of the ECFPs of the two compounds. The correlation coefficients are 0.127 and 0.0346, respectively

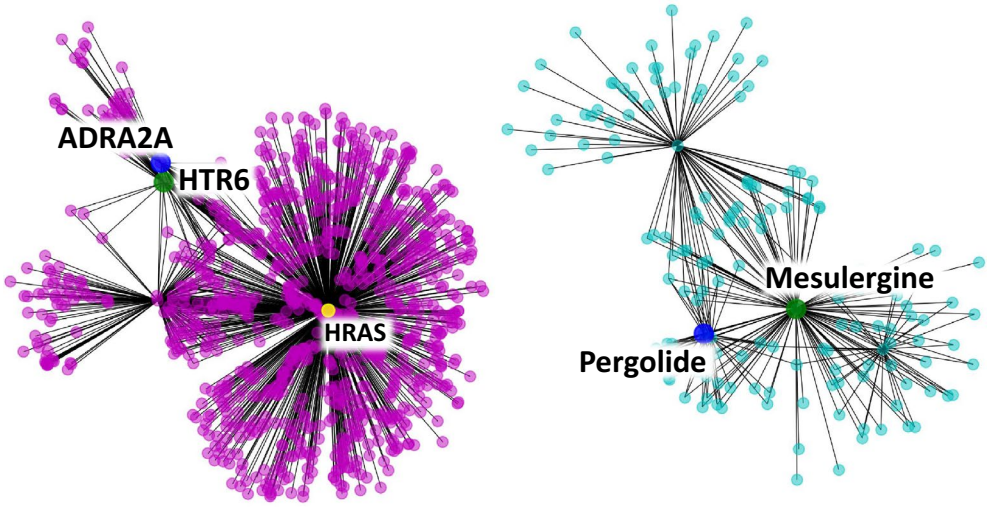

Fig. 4 (Left) Part of the protein–protein interaction network around HTR6 and ADRA2A. (Right) Part of the compound-compound interaction network around Mesulergine and Pergolide

Conclusions

This study aimed to improve the performance of protein–compound interaction prediction by integrating molecular structure and interactome data, which was achieved by integrating multiple heterogeneous interactome data into predictions of protein–compound interactions. An end-to-end learning method was developed that combines a 1D-CNN for amino acid sequences, an ECFP representation for compounds, and feature representation learning with node2vec for protein–protein and compound–compound interaction networks. The proposed integrated model exhibited significant performance differences with respect to accuracy measures compared with the current state-of-the-art deep learning methods. This improvement in performance was verified as statistically significant by the Wilcoxon signed-rank test. The results indicated that the proposed model is able to more accurately predict protein–compound interactions even in a hard dataset, whereby neither the proteins nor compounds in the test sample are included in the training sample.

An important future task is to integrate the gene regulatory network as additional interactome data to further improve protein–compound interaction prediction. A large number of gene expression profiles for various tissues and cell lines are available in public databases, and gene regulatory networks can be effectively inferred from gene expression profiles. For example, the effectiveness of utilizing gene expression data for drug repositioning was reported in a summary review [35]. Gene expression profiles can be effective in restoring connections between genes, drugs, and diseases involved in the same biological process. We have obtained a promising preliminary result of integrating gene expression data reposited in DrugBank into our deep learning model to improve the prediction accuracy for protein–compound interactions.

Abbreviations

SVM: Support vector machines; CNN: Convolutional neural network; ECFP: Extended-connectivity fingerprint; VSE: Visual semantic embedding; AUROC: Area under the receiver operating characteristic curve; AUPRC: Area under the precision-recall curve; SD: Standard deviation.

Acknowledgements

Not applicable.

Authors' contributions

NW: Implemented the software, analysed the data, and co-wrote the paper. YO: analysed the data and performed comparisons with the existing methods. YS: designed and supervised the research, analysed the data, and co-wrote the paper. All authors read and approved the final manuscript.

Funding

This work was supported by a Grant-in-Aid for Scientific Research on Innovative Areas "Frontier Research on Chemical Communications" [No. 17H06410] from the Ministry of Education, Culture, Sports, Science and Technology of Japan and a Grant-in-Aid for Scientific Research (A) (KAKENHI) [No. 18H04127] from the JSPS.

Availability of data and materials

All the source programs, including the main deep learning program and some other tools for the data formatting process, all the data used in this study, and instructions (README file) on how to use the program are available at our GitHub site: https://github.com/Njk-901aru/multi_DTI.git. A brief summary of how to use our programs is as follows:

Requirements: Python Anaconda 3 (RDKit and Chainer installed), PubChemPy (for converting the PubChem ID into SMILES), node2vec, networkx (for applying the node2vec algorithm), and pyensembl (for converting the Ensembl protein ID into amino acid sequences).

Usage: (1) data pre-processing, (2) application of node2vec to multi-interactome data, (3) learning and prediction by the integrated model.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 12 February 2021 Accepted: 21 April 2021
Published online: 01 May 2021

References

1. Nagamine N, Sakakibara Y (2007) Statistical prediction of protein–chemical interactions based on chemical structure and mass spectrometry data. Bioinformatics 23:2004–2012
2. Nagamine N, Shirakawa T, Minato Y, Torii K, Kobayashi H, Imoto M, Sakakibara Y (2009) Integrating statistical predictions and experimental verifications for enhancing protein-chemical interaction predictions in virtual screening. PLoS Comput Biol 5:e1000397
3. Sakakibara Y, Hachiya T, Uchida M, Nagamine N, Sugawara Y, Yokota M, Nakamura M, Popendorf K, Komori T, Sato K (2012) COPICAT: a software system for predicting interactions between proteins and chemical compounds. Bioinformatics 28:745–746
4. Wishart DS, Knox C, Guo AC, Cheng D, Shrivastava S, Tzur D, Gautam B, Hassanali M (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res 36:D901–D906
5. Kim S, Thiessen PA, Bolton EE, Chen J, Fu G, Gindulyte A, Han L, He J, He S, Shoemaker BA, Wang J, Yu B, Zhang J, Bryant SH (2016) PubChem substance and compound databases. Nucleic Acids Res 44:D1202–D1213
6. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
7. Tian K, Shao M, Wang Y, Guan J, Zhou S (2016) Boosting compound-protein interaction prediction by deep learning. Methods 110:64–72
8. Öztürk H, Özgür A, Ozkirimli E (2018) DeepDTA: deep drug-target binding affinity prediction. Bioinformatics 34:i821–i829
9. Lee I, Keum J, Nam H (2019) DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput Biol 15:e1007129
10. Tsubaki M, Tomii K, Sese J (2019) Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics 35:309–318
11. Zitnik M, Agrawal M, Leskovec J (2018) Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 34:i457–i466
12. Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J (2017) A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. Nat Commun 8:573
13. Wan F, Hong L, Xiao A, Jiang T, Zeng J (2019) NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. Bioinformatics 35:104–111
14. Hosur R, Peng J, Vinayagam A, Stelzl U, Xu J, Perrimon N, Bienkowska J, Berger B (2012) A computational framework for boosting confidence in high-throughput protein-protein interaction datasets. Genome Biol 13:R76
15. Chen M, Ju CJT, Zhou G, Chen X, Zhang T, Chang KW, Zaniolo C, Wang W (2019) Multifaceted protein-protein interaction prediction based on Siamese residual RCNN. Bioinformatics 35:i305–i314
16. Chen L, Lu J, Zhang J, Feng KR, Zheng MY, Cai YD (2013) Predicting chemical toxicity effects based on chemical-chemical interactions. PLoS ONE 8:e56517
17. Kwon S, Yoon S (2019) End-to-end representation learning for chemical-chemical interaction prediction. IEEE/ACM Trans Comput Biol Bioinform 16:1436–1447
18. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, Jensen LJ, von Mering C (2017) The STRING database in 2017: quality-controlled protein-protein association networks, made broadly accessible. Nucleic Acids Res 45:D362–D368
19. Kuhn M, von Mering C, Campillos M, Jensen LJ, Bork P (2016) STITCH: interaction networks of chemicals and proteins. Nucleic Acids Res 36:D684–D688
20. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks. In: Proceedings of KDD '16 (22nd ACM SIGKDD international conference on knowledge discovery and data mining). ACM, New York, NY, USA, p 855–864
21. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50:742–754
22. Kiros R, Salakhutdinov R, Zemel RS (2014) Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539
23. Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of 25th ACM SIGKDD international conference on knowledge discovery & data mining. ACM, New York, NY, USA, pp 2623–2631
24. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167
25. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
26. Bishop C (2006) Pattern recognition and machine learning. Springer, Berlin
27. Liu R, Hameed MDMA, Kumar K, Yu X, Wallqvist A, Reifman J (2017) Data-driven prediction of adverse drug reactions induced by drug-drug interactions. BMC Pharmacol Toxicol 18:44
28. Kuhn M, Szklarczyk D, Franceschini A, von Mering C, Jensen LJ, Bork P (2011) STITCH 3: zooming in on protein-chemical interactions. Nucleic Acids Res 40:D876–D880
29. Chatr-Aryamontri A, Oughtred R, Boucher L, Rust J, Chang C, Kolas NK, O'Donnell L, Oster S, Theesfeld C, Sellam A, Stark C, Breitkreutz BJ, Dolinski K, Tyers M (2019) The BioGRID interaction database: 2017 update. Nucleic Acids Res 45:D369–D379
30. Orchard S, Kerrien S, Abbani S, Aranda B, Bhate J, Bidwell S, Bridge A, Briganti L, Brinkman FSL, Cesareni G, Chatr-Aryamontri A, Chautard E, Chen C, Dumousseau M, Goll J, Hancock REW, Hannick LI, Jurisica I, Khadake J, Lynn DJ, Mahadevan U, Perfetto L, Raghunath A, Ricard-Blum S, Roechert B, Salwinski L, Stümpflen V, Tyers M, Uetz P, Xenarios I, Hermjakob H (2012) Protein interaction data curation: the International Molecular Exchange (IMEx) consortium. Nat Methods 9:345–350
31. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1:80–83
32. Krogsgaard-Larsen N, Jensen AA, Schroder TJ, Christoffersen CT, Kehler J (2014) Novel aza-analogous ergoline derived scaffolds as potent serotonin 5-HT6 and dopamine D2 receptor ligands. J Med Chem 57:5823–5828
33. Millan MJ, Maiofiss L, Cussac D, Audinot V, Boutin JA, Newman-Tancredi A (2002) Differential actions of antiparkinson agents at multiple classes of monoaminergic receptor. I. A multivariate analysis of the binding profiles of 14 drugs at 21 native and cloned human receptor subtypes. J Pharmacol Exp Ther 303:791–804
34. Childs-Disney JL, Tran T, Vummidi BR, Velagapudi SP, Haniff HS, Matsumoto Y, Crynen G, Southern MR, Biswas A, Wang ZF, Tellinghuisen TL, Disney MD (2018) A massively parallel selection of small molecule-RNA motif binding partners informs design of an antiviral from sequence. Chemistry 4:2384–2404
35. Iorio F, Rittman T, Ge H, Menden M, Saez-Rodriguez J (2013) Transcriptional data: a new gateway to drug repositioning? Drug Discov Today 18:350–357

Publisher's note