RSC Advances

PAPER

View Article Online
View Journal | View Issue

Cite this: *RSC Adv.*, 2022, **12**, 29525

Graph–sequence attention and transformer for predicting drug–target affinity

Xiangfeng Yan and Yong Liu*

Drug–target binding affinity (DTA) prediction has drawn increasing interest due to its substantial position in the drug discovery process. The development of new drugs is costly, time-consuming, and often accompanied by safety issues. Drug repurposing can avoid the expensive and lengthy process of drug development by finding new uses for already approved drugs. Therefore, it is of great significance to develop effective computational methods to predict DTAs. The attention mechanisms allow the computational method to focus on the most relevant parts of the input and have been proven to be useful for various tasks. In this study, we proposed a novel model based on self-attention, called GSATDTA, to predict the binding affinity between drugs and targets. For the representation of drugs, we use Bi-directional Gated Recurrent Units (BiGRU) to extract the SMILES representation from SMILES sequences, and graph neural networks to extract the graph representation of the molecular graphs. Then we utilize an attention mechanism to fuse the two representations of the drug. For the target/protein, we utilized an efficient transformer to learn the representation of the protein, which can capture the long-distance relationships in the sequence of amino acids. We conduct extensive experiments to compare our model with state-of-the-art models. Experimental results show that our model outperforms the current state-of-the-art methods on two independent datasets.

Received 4th September 2022
Accepted 4th October 2022

DOI: 10.1039/d2ra05566j

rsc.li/rsc-advances

1 Introduction

Drug discovery is a complicated process, which has the risks of long research cycles, high costs, and low success rates. It takes billions of dollars and more than ten years to develop a new drug from development to approval.[1,2] The effective prediction of drug–target binding affinity (DTA) is one of the significant issues in drug discovery.[3–5] Drugs are usually represented as a string obtained from the simplified molecular-input line-entry system (SMILES)[6] or represented by a molecule graph with atoms as nodes and chemical bonds as edges. Targets (or proteins) are sequences of amino acids. Binding affinity indicates the strength of drug–target pair interaction. Through binding, drugs can have a positive or negative influence on functions carried out by proteins, affecting the disease conditions.[7] By understanding drug–target binding affinity, it is possible to find out candidate drugs that can inhibit the target/protein and benefit many other bioinformatics applications.[8,9]

Early computational attempts were focused on biologically intuitive methods, such as ligand-similarity based approaches and docking simulations.[10] Ligand-similarity based methods predict interactions by comparing a new ligand to known ligands of proteins. However, ligand-similarity methods perform poorly when the number of known ligands is insufficient. Docking simulation methods require the 3D structures of the target proteins hence becoming inapplicable when there are numerous proteins with unavailable 3D structures.[11] In the past few years, some research has begun predicting DTAs from a network perspective.[12–14] However, the prediction qualities of network-based approaches are strongly limited by available linked information. Thus, these methods do not perform very well on association predictions for new drugs or targets with scarce linked information. Additionally, some useful information, such as drug and target feature information, cannot be fully utilized to improve prediction accuracy for these methods.

With the development of artificial intelligence, deep learning approaches for DTA prediction have become popular and can be categorized into two main groups according to the input data: sequence-based and graph-based methods. The sequence-based methods learned the representations from sequential data, which are SMILES sequences of drugs and amino acid sequences of proteins. The graph-based methods represented drugs as molecular graphs, which learned the representations from molecular graphs and amino acid sequences of proteins. Although deep learning models show excellent performance improvement in DTA prediction, two main challenges remain to study. First, these methods consider either SMILES sequences

School of Computer Science and Technology, Heilongjiang University, Harbin, China. E-mail: 2010023@hlju.edu.cn

or molecular graphs, which failed to capture comprehensive representations of drugs. A SMILES sequence can offer the following features as a representation: (i) ionic groups and atomic groups are represented in the canonical way, which avoids confusion with their surrounding atomic groups. For instance, ammonium is denoted as [NH4+] rather than HHNHH; (ii) some specially defined symbols are used to preserve chemical properties such as chemical valence, isotopes, *etc.* However, merely taking drugs with sophisticated internal connectivity as simple sequential data lacks sufficient interpretable and expressive capabilities. Molecular graph brings two unique benefits as compared to SMILES sequence: (i) molecular graph can capture the spatial connectivity of different atoms, especially for star structures and ring structures (*e.g.*, alkyl and benzene ring); (ii) chemical molecular bonds are well preserved, which might influence the molecular properties. For instance, carbon dioxide has divalent bonds between carbon and oxygen. However, similar to sequence modeling in SMILES, simply using molecular graphs to model molecules cannot enable methods to comprehensively learn molecular representations. It is difficult to capture information on some specific molecular properties, such as atoms' chirality, using molecular graphs. Second, most existing methods utilized convolutional neural networks (CNNs) to learn low-dimensional feature representations of proteins from the sequence of amino acids, which ignored the long-distance relationships in the protein sequences.

To overcome the mentioned challenges of current methods for DTA prediction, we propose a novel triple-channel model, named Graph–Sequence Attention and Transformer for Predicting Drug–Target Affinity (GSATDTA), to predict the binding affinity between drugs and targets. Recently, Guo *et al.*[15] proposed that integrating the capabilities of both molecular graphs and SMILES sequences can further enhance molecule representation expressive power. Enlightened by this work, we use a graph neural network to learn the graph representation from molecular graphs and a BiGRU to learn the SMILES representation from SMILES sequences. Then, we propose a graph–sequence attention mechanism to capture significant information from both the SMILES sequence and molecular graph. For the protein representations, we replace CNN with an efficient transformer to learn the representation of the protein, which can capture the long-distance relationships in the sequence of amino acids.[16] The main contributions of this paper are summarized as follows:

- We leveraged both the graph and sequence information and proposed a graph–sequence attention mechanism to learn effective drug representations for DTA prediction.
- We utilized an efficient transformer to learn the representation of the protein, which can capture the long-distance relationships in the sequence of amino acids.
- We conduct extensive experiments on two benchmark datasets to investigate the performance of our proposed model. The experimental results show that our proposed model achieves the best performance in the drug–target binding affinity prediction task.

2 Related work

2.1 Simulation methods

Most previous works have focused on simulation-based methods (*i.e.*, molecular docking and descriptors). For example, Li *et al.*[17] proposed a docking method based on random forest (RF). The RF model was also adopted in KronRLS,[18] which uses the similarity score obtained by the Kronecker product of the similarity matrix to improve the predictive performance. To alleviate the limitation of linear dependence in KronRLS, a gradient boosting method is proposed in SimBoost[19] to construct the similarity between drugs and targets. While classical methods have shown rational performance in DTA prediction, they are usually computationally expensive or rely on external expert knowledge or the 3D structure of the target/protein.

2.2 Sequence-based methods

The sequence-based methods learned the representations from sequential data, which are SMILES sequences of drugs and amino acid sequences of proteins. For example, DeepDTA[20] uses the 1D representation of the drug and protein, and uses convolutional neural networks (CNNs) to learn representations from the raw protein sequences and SMILES strings. Then, they combined these representations to feed into a fully connected layer to predict the drug–target affinity scores. Similarly, WideDTA[21] also relies only on the 1D representation, but different from DeepDTA, the SMILES and protein sequence were represented as words (instead of characters) that correspond to an eight-character sequence and a three-residual sequence, respectively. In addition, WideDTA utilized the ligand maximum common substructure (LMCS)[22] of drugs and motifs and domains of proteins (PDM),[23] which formed a four-branch architecture together with the ligand SMILES and protein sequence branches. The WideDTA model is an extension of DeepDTA, which also used CNN to learn the representations of drugs and proteins. GANsDTA[24] utilized a generative adversarial networks (GAN) to learn beneficial patterns within labeled and unlabeled sequences and used convolutional regression to forecast binding affinity scores. MATT_DTI[3] also utilizes the 1D representation, it proposed a relation-aware self-attention block to model the relative position between atoms in drugs, considering the correlation between atoms. As for the protein, it utilizes three convolutional layers as the feature extractor, followed by a max pooling layer. Then, a multi-head attention block is built to model the similarity of drug–target pairs as the interaction information for DTA prediction.

2.3 Graph-based methods

The graph-based methods represented drugs as molecular graphs, and learned the representations from molecular graphs and amino acid sequences of proteins. Tsubaki *et al.*[25] proposed the application of the graph neural network (GNN) to DTA (or compound–protein interaction, CPI) prediction. In their model, the chemical structures of drugs (provided in SMILES notation) are represented as graphs. Therefore, they propose the use of

GNN and CNN, which can learn low-dimensional real-valued vector representations of molecular graphs and protein sequences. Similarly, Gao *et al.*[26] utilized the GNN for drug representation, whereas the protein descriptors were obtained using long short-term memory (LSTM). GraphDTA[5] also introduced graph representation to take advantage of the topological structure information of the molecular graph. GraphDTA used a three-layer GCN as an alternative for drug representation while utilizing the CNN to learn protein representation as in DeepDTA. Among the research on deep learning for drug discovery, DeepGS[27] is the most relevant to our work. DeepGS considers both the molecular graphs and SMILES sequences of drugs, however, they directly contacted the two representations, which failed to capture comprehensive information about drugs. Furthermore, DeepGS and other deep learning methods utilized CNN to learn local representations of protein sequences, which ignored the long-distance relationships in the protein sequences.

Compared with these deep learning models, we designed a graph–sequence attention mechanism, which can capture significant information from both SMILES sequences and molecular graphs. As for the protein, we replace CNN with an efficient transformer to extract the long-distance relationships in the sequence of amino acids.

3 Materials and methods

We regard DTA prediction as a regression task to predict the binding affinity value between a drug–target pair. We denote the SMILES sequence set as $D = \{D_i\}_{i=1}^{|D|}$ and the protein sequence set as $P = \{P_j\}_{j=1}^{|P|}$. We use RDKit to convert the SMILES set D into a molecular graph set $G = \{G_i\}_{i=1}^{|G|}$, where $G_i = (V_i, E_i)$ denotes a molecular graph. Specifically, a node on a molecular graph represents an atom, and an edge represents a chemical bond between two atoms. Formally, the problem of drug–target binding affinity prediction is defined as follows.

Given a SMILES set D and a protein set P, and their interaction labels $Y = \{Y_{i,j} | 1 \leq i \leq |D|, 1 \leq j \leq |P|, Y_{i,j} \in R\}$, the binding affinity prediction problem is to learn a function f: $D \times P \rightarrow Y$ such that $f(D_i, P_j) \rightarrow Y_{i,j}$.

3.1 Overview

We proposed a triple-channel model, called GSATDTA; the overall architecture of the model is shown in Fig. 1. It takes the symbolic sequences of the target/protein and drug as inputs, as well as the molecular structure of the drug, and outputs the binding affinity of the drug to the target. For the drug representations, we use a graph neural network to learn the topological structure information from molecular graphs and BiGRU to learn contextual information about drugs represented by SMILES sequences. Then, we utilized the graph–sequence attention mechanism to capture significant information from both the SMILES sequence and molecular graph. For the protein representations, we employed an efficient transformer to capture the long-distance relationships in the sequence of amino acids. Thus, we obtained the target representation and the fusion drug representation. The connection of two representations is inputted into several dense layers and ends with

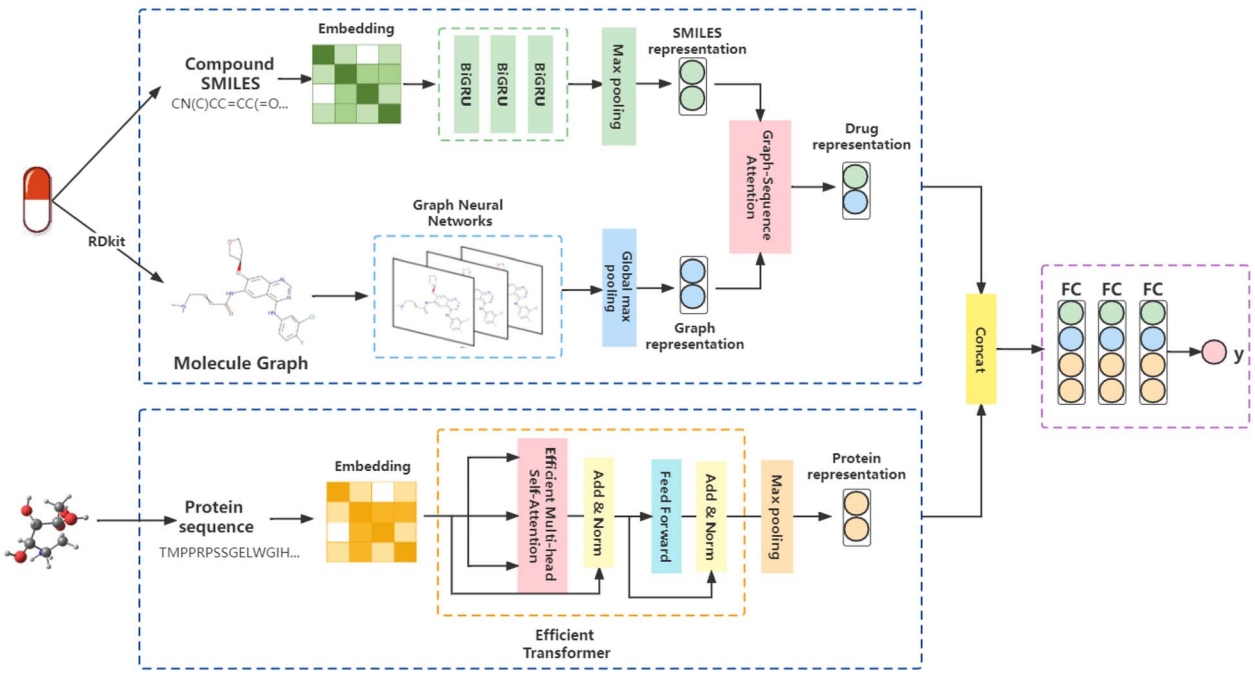

Fig. 1 Illustration of the proposed GSATDTA. We leveraged both the graph and sequence information and utilized the graph–sequence attention mechanism to learn effective drug representations. For the protein, we employed an efficient transformer to learn the representation of the protein sequence. Finally, the two representations were concatenated and passed through several dense fully connected layers to estimate the output as the drug–target affinity value.

a regression layer to predict the drug–target binding affinity value. Next, we will present the details of our model.

3.2 Representation learning of drug

3.2.1 Representation learning of sequence. Simplified Molecular-Input Line-Entry System (SMILES) is a non-unique representation that encodes the molecular graph into a string of ASCII characters. For example, the SMILES "CN(C)CC=CC(=O)…" for a drug in Fig. 1 is a sequence of atoms and chemical bonds. SMILES rules can cover atoms, ions, chemical bonds, valences, and chemical reactions, which can accurately express branched, cyclic, tetrahedral, aromatic, and chiral structures, as well as the expression of isomers and isotopes.

Most previous studies adopted one-hot encoding to encode the symbols in the SMILES sequence. However, one-hot encoding ignores the contextual value of the symbol, and thus cannot reveal the functionality of the symbol within the context.[28,29] To address this problem, we utilized Smi2Vec,[30] a method similar to Word2Vec,[31] to encode the tokens in the SMILES sequences. According to DeepGS,[27] we fixed maximum lengths of 100 for SMILES sequences. We cut the SMILES sequence if the length of the SMILES sequence is longer than 100. Otherwise, we use zero-padding at the end of the SMILES sequence. In a typical case, a fixed-length SMILES string, for example, C, is partitioned into individual atoms or symbols. It is then mapped to atoms by seeking out each atom embedded from a pretrained dictionary, and if not in the dictionary, it is obtained from randomly generated values.[32,33] Then, atom embedding vectors are aggregated to form the final embedding matrix. Enlightened by the gate function in GRU,[34] we applied the three layers technique of BiGRU to the generated matrix to obtain a latent representation of the drug, which allows us to model the local chemical context. Finally, we obtain the SMILES representation $S_i \in R^N$ through the max-pooling layer and the fully connected layer, where N is the output dimensional of the fully connected layer.

3.2.2 Representation learning of graph. A vital indication for the estimation of DTA is to effectively exploit molecular structure information to reveal the interconnections between atoms in the drug.[5,25] To achieve this, we transformed the SMILES sequence into the molecule graph through the RDKit. Graph Isomorphic Network (GIN)[35] has shown its superiority for modeling graph representation in many studies and supposedly achieves maximum discriminative power among graph neural networks. Our model consists of five GIN layers, each GIN layer is followed by a batch normalization layer, activated by a ReLU function. Specifically, GIN uses a multi-layer perceptron (MLP) model to update the node features as formula (1):

$$\mathrm{MLP}\left((1+\varepsilon)X_i^j + \sum_{k\in N(j)} X_i^k\right) \tag{1}$$

where ε is either a learnable parameter or fixed scalar, $X_i^j \in R^F$ is the feature vector of node j in the molecular graph i, and $N(j)$ is the set of nodes adjacent to node j. Finally, a global max-pooling layer is added to aggregate the entire graph representation $H_i \in R^N$.

3.2.3 Graph–sequence attention. The attention mechanisms allow the network to focus on the most relevant parts of the input and have been proven to be useful for various tasks.[4,36] To capture significant information from both the SMILES sequence and molecular graph, we designed an attention mechanism, called Graph–Sequence Attention. Specifically, given the SMILES representation $S_i \in R^N$ and the graph representation $H_i \in R^N$, we transform S_i and H_i into the vectors d_{S_i} and d_{H_i} through formula (2) for feature extraction and attention modeling:

$$\begin{aligned} d_{S_i} &= \mathrm{ReLU}(W_1 \cdot S_i + b) \\ d_{H_i} &= \mathrm{ReLU}(W_2 \cdot H_i + b) \end{aligned} \tag{2}$$

where $W_1 \in R^N$ and $W_2 \in R^N$ are trainable parameters, and b represents the bias vector. Then, we combine the output d_{S_i} with d_{H_i} through a dimensional-wise fusion gate F. F is accomplished by the *sigmoid* activation function to encode two parts of the representation:

$$F = sigmoid(W_G \cdot d_{H_i} + W_S \cdot d_{S_i}) \tag{3}$$

where $W_G \in R^N$ and $W_S \in R^N$ are trainable parameters of the fusion gate. Finally, the final vector representation output of a specific drug is generated through F:

$$M_\mathrm{d} = F \odot H_i + (1 - F) \odot S_i \tag{4}$$

where $\odot$ is the element-wise product.

3.3 Representation learning of protein

The protein sequence is a string of ASCII characters, which represents amino acids. Mathematically, a protein sequence is expressed as $P_j = \{p_1, p_2, \ldots, p_i, \ldots\}$, where $p_i \in N^*$ and the length of P_j depends on the proteins. We fix the length of the input protein sequence as L_p to ensure the same size of inputs. According to the token embedding and position embedding in the transformer,[16] the input of the efficient transformer is the sum of token embedding and position embedding of protein sequences. The token embedding $T^p_{\mathrm{tok}} \in R^{L_p \times M}$ has a trainable weight $W_m \in R^{v_p \times M}$, where v_p is the vocabulary size of proteins and M is the embedding size of proteins. The position embedding $T^p_{\mathrm{pos}} \in R^{L_p \times M}$ has a trainable weight $W_p \in R^{L_p \times M}$. The output of the embedding operations is

$$T^p = T^p_{\mathrm{tok}} + T^p_{\mathrm{pos}}, \tag{5}$$

where $T^p \in R^{L_p \times M}$.

3.3.1 Efficient transformer. Most protein sequences are long sequences, so simple CNN cannot capture the long-distance relationships in protein sequences well. Therefore, current DTA prediction models are limited due to that CNN cannot capture the long-distance relationships in protein sequences. The transformer can alleviate this limitation, which offers a more flexible mechanism to model the long-distance relationships. The multi-head self-attention (MSA) in the transformer encoder layer has been modeled and integrated to improve the model's ability to learn long-distance

relationships. Multi-head attention can jointly attend to information from the extracted features at different sequence positions. The self-attention layer takes three inputs, the keys, **K**, the values, **V**, and the queries, **Q**, and calculates the attention as follows:

$$\text{attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V \quad (6)$$

where $\sqrt{d_K}$ is a scaling factor depending on the layer size.

However, the memory and computation for multi-head self-attention in the traditional transformer scale quadratically with spatial or embedding dimensions (*i.e.*, the number of channels), causing vast overheads for training and inference. Thus, we replace the multi-head self-attention with Efficient Multi-head Self-Attention (EMSA).[37] The architecture of the efficient multi-head self-attention is shown in Fig. 2.

Similar to MSA, EMSA first adopts a set of projections to obtain query **Q**. To compress memory, the 2D input $T^p \in R^{L_p \times M}$ is reshaped to a 3D one along the spatial dimension (*i.e.*, $\widehat{T^p} \in R^{M \times h \times w}$) and then fed to a depth-wise convolution operation to reduce the height and width dimension by a factor s. To make it simple, s is an adaptive set by the feature map size or the stage number. The kernel size, stride and padding are $s+1$, s, and $s/2$, respectively. The new token map after spatial reduction $\widehat{T^p} \in R^{M \times h/s \times w/s}$ is then reshaped to a 2D one, *i.e.*, $\widehat{T^p} \in R^{L'_p \times M}$, $L'_p = h/s \times w/s$. Then $\widehat{T^p}$ is fed to two sets of projection to get key **K** and value **V**. After that, we adopt eqn (7) to compute the attention function on query **Q**, **K** and value **V**.

$$\text{EMSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{IN}\left(\text{softmax}\left(\text{conv}\left(\frac{QK^T}{\sqrt{d_K}}\right)\right)\right)V \quad (7)$$

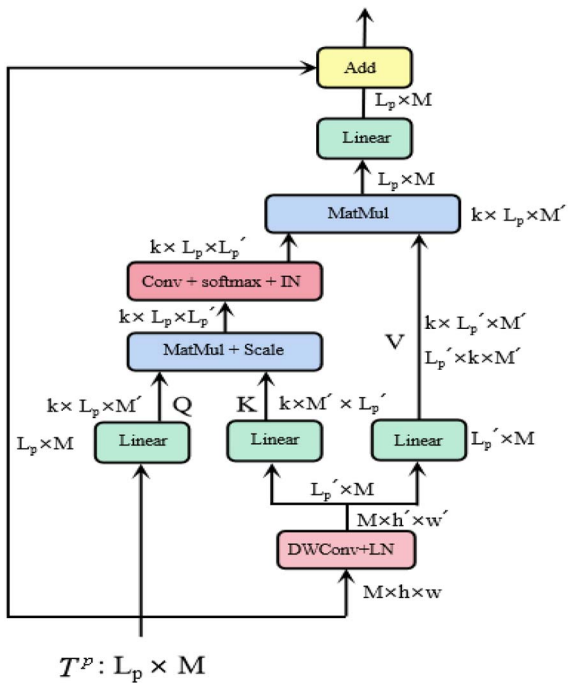

Fig. 2 Efficient multi-head self-attention.

Here, conv(·) is a standard 1 × 1 convolutional operation, which models the interactions among different heads. As a result, the attention function of each head can depend on all of the keys and queries. However, this will impair the ability of MSA to jointly attend to information from different representation subsets at different positions. To restore this diversity ability, we add an Instance Normalization[38] (*i.e.*, IN(·)) for the dot product matrix (after softmax). Then, the output values of each head are concatenated and linearly projected to form the final output. Finally, the protein representation is obtained through the max-pooling layer and the fully connected layer, which we denote as T_j.

3.4 Drug–target binding affinity prediction

In this paper, we treat the drug–target binding affinity prediction task as a regression task. With the representation learned from the previous sections, we can integrate all the information about the drug and target to predict the binding affinity value. Firstly, we concatenated the drug representation M_d and the protein representation T_j. Secondly, we feed it into three dense fully connected layers to predict the binding affinity value. Besides, we use ReLU as the activation function for increasing the nonlinear relationship. Given the set of drug–target pairs and the ground-truth labels, we use the mean squared error (MSE) as the loss function.

4 Experiments and results

4.1 Datasets

Following previous works, we employ two widely used datasets dedicated to DTA prediction:

- Davis:[20] the Davis dataset, which contains 68 drugs and 442 targets, with 30 056 drug–target interactions. Affinity values range from 5.0 to 10.8.
- Kiba:[20] the Kiba dataset, which contains 2111 drugs and 229 targets, with 118 254 drug–target interactions. Affinity values range from 0.0 to 17.2.

For both datasets, we use the same training/testing data ratio as MATT_DTA,[3] DeepDTA,[20] and GraphDTA[5] in our experiments, making the comparison as fair as possible. That is, 80% of the data is used for training and the remaining 20% is used for testing the model. For the same purpose, we use the same evaluation metrics as MATT_DTA, GraphDTA, and DeepDTA for evaluating model performance: the mean squared error (MSE, the smaller the better), r_m^2 (the larger the better), and the concordance index (CI, the larger the better). Table 1 summarizes the details of the Davis and Kiba datasets.

Table 1 Summary of the benchmark datasets

	Davis	Kiba
Compounds	68	2111
Proteins	442	229
Interactions	30 056	118 254
Training data	25 046	98 545
Test data	5010	19 709

4.2 Evaluation metrics

MSE is a common metric to measure the difference between the predicted value and the real value:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{8}$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, and N is the number of drug–target pairs.

CI is used to measure whether the predicted binding affinity values of two random drug–target pairs were predicted in the same order as their true values:

$$\mathrm{CI} = \frac{1}{Z}\sum_{d_i > d_j} h(b_i - b_j) \tag{9}$$

where b_i is the prediction value for the larger affinity d_i, b_j is the prediction value for the smaller affinity d_j, and h is the step function, as shown in eqn (10), and Z is a normalization constant that equals the number of drug–target pairs with different binding affinity values.

$$h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{10}$$

The metric ${r_\mathrm{m}}^2$ is used to evaluate the external prediction performance of QSAR (Quantitative Structure–Activity Relationship) models. A model is acceptable if and only if ${r_\mathrm{m}}^2 = r^2 \times (1 - \sqrt{r^2 - {r_0}^2})$, where r^2 and ${r_0}^2$ are the squared correlation coefficient values between the observed and predicted values with and without intercept, respectively.

4.3 Baseline methods

We compare our model with the following state-of-the-art models:

• KronRLS:[18] this baseline formulates the problem of learning a prediction function f as finding a minimizer of the following objective function:

$$J(f) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda {\|f\|_k}^2 \tag{11}$$

where N is the number of drug–target pairs, ${\|f\|_k}^2$ is the norm of f, which is related to the kernel function k, and $\lambda > 0$ is a regularization hyper-parameter defined by the user.

• SimBoost:[19] this baseline is a gradient-boosting machine-based method that constructs features of drug, target, and drug–target pairs. These features are fed into a supervised learning method named gradient boosting regression trees, which is derived from the gradient boosting machine model. Using a gradient regression tree, for a given drug–target pair d_{t_i}, the binding affinity score $\hat{y}_i$ is calculated as follows:

$$\hat{y}_i = \theta(d_{t_i}) = \sum_{m=1}^{M} f_m(d_{t_i}), \quad f_m \in F \tag{12}$$

where M represents the number of regression trees, f_m is a regression tree and F represents the space of all possible trees. The objective function with regularization in the regression tree set is described in the following form:

$$R(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{m=1}^{M} \alpha(f_m) \tag{13}$$

where N is the number of drug–target pairs, l is the loss function, y_i is the true value, $\hat{y}_i$ is the predicted value, and α is a tuning parameter that controls the complexity of the model.

• DeepDTA:[20] this baseline trains two 3-layer CNNs using label/one-hot encoding to encode drug and protein sequences for DTA prediction. The CNN model contains two independent CNN blocks that capture features from SMILES sequences and protein sequences, respectively. The drug and target representations are concatenated and passed to a fully connected layer for DTA prediction.

• WideDTA:[21] this baseline represents SMILES strings and protein sequences as word sequences and represents the corresponding drugs and proteins through the most common subsequence. In particular, drugs are described by the most common subsequences as Ligand Maximum Common Substructures (LMCS); proteins are represented by the most conserved subsequences that are Protein Domain profiles or Motifs (PDM) retrieved from the PROSTE database. WideDTA contains four independent CNN blocks that learn features from SMILES sequences, LMCS, protein sequences, and PDM. The drug and target representations are all concatenated and passed to a fully connected layer for DTA prediction.

• GANsDTA:[24] this baseline proposes a semi-supervised generative adversarial networks (GANs)-based method to predict binding affinity. This method comprises two types of networks, two partial GANs for the feature extraction from the raw protein sequences and SMILES strings separately, and a regression network using convolutional neural networks for prediction.

• DeepGS:[27] DeepGS considers both the molecular graphs and SMILES sequences of drugs, and uses BiGRU to extract the local chemical context of SMILES sequences and GAT to capture the topological structure of molecular graphs. For the protein sequences, the CNN module is utilized to learn protein representations from the sequences of amino acids. Then, the representations of drugs and targets are concatenated and passed to a fully connected layer for DTA prediction.

• GraphDTA:[5] this baseline converts drugs from SMILES sequences to molecular graphs. GraphDTA consists of two separate modules, a GNN module for modeling molecular graphs to obtain drug representations, and a CNN module for modeling protein sequences to obtain target representations. The drug and target representations are concatenated and passed to a fully connected layer for DTA prediction.

• MATT_DTI:[3] this baseline use SMILES sequences and protein sequences as inputs. Unlike DeepDTA, MATT_DTI proposes a relation-aware self-attention module to model SMILES sequences. The relative self-attention module can enhance the relative position information between atoms in

a compound while considering the relationship between elements. After the drug and target representations are obtained, a multi-head attention mechanism is used to model the interaction of drug representations and protein representations for DTA prediction.

4.4 Results and discussion

To examine the competitiveness of our proposed model, we compared the model with the current state-of-the-art models on the DTA prediction task. Tables 2 and 3 show the performance of different models on the Davis and Kiba datasets based on MSE, CI, and r_m^2 metrics. As shown in Tables 2 and 3, those classical methods such as SimBoost[19] perform worse than deep learning methods. This is because classical methods rely heavily on manually annotated features provided by a domain expert and drug–target similarity matrices. In comparison, deep learning methods can capture more hidden features by using the automatic feature extraction ability of CNN and GNN.

First, we consider a few recent textual representation approaches such as: DeepDTA,[20] WideDTA,[21] GANsDTA,[24] and MATT_DTI.[3] Among these approaches, MATT_DTI achieved the best results in terms of CI and MSE. MATT_DTI achieved CI of 0.890, and MSE of 0.229 on the Davis dataset; also, on the Kiba dataset, it achieved CI of 0.889 and MSE of 0.150. This explains the effectiveness of the attention mechanism in learning drug information in the case of MATT_DTI.

Second, we also consider some graph network approaches, GraphDTA and DeepGS. These graph representation approaches can effectively capture topological relationships of drug molecules, which enable further performance improvement. Amongst them, the GraphDTA shows a higher CI value of 0.893 and a lower MSE of 0.229 on the Davis dataset; and on the Kiba dataset, GraphDTA achieved 0.889 in terms of CI and 0.147 in terms of MSE. From Tables 2 and 3, we can find that although DeepGS considers both the molecular graphs and SMILES sequences of drugs, it performs worse than GraphDTA in terms of CI, MSE, and r_m^2. The reason is that DeepGS directly contacted the SMILES representation and graph representation, which failed to capture comprehensive information about drugs.

As shown in Tables 2 and 3, our proposed GSATDTA has a robust performance on both datasets. For the Davis dataset, our model achieved 0.906 (0.013 improvement), 0.200 (reduced by 0.029), and 0.732 (0.046 improvement) for CI, MSE, and r_m^2, respectively. For the Kiba dataset, we achieved 0.902 for CI (0.013 improvement), 0.126 for MSE (reduced by 0.021), and 0.790 for r_m^2 (0.034 improvement). We observe that our model outperforms existing deep-learning methods on three measures, which can be explained due to two factors:

(1) Compared with these models, we replaced CNN with an efficient transformer to learn the representation of the protein, which can capture the long-distance relationships in the sequence of amino acids.

(2) We adopted the GIN architecture to learn the structural information of the molecular graphs and employed BiGRU to obtain extra contextual information for the SMILES sequences. Then, we utilized the graph–sequence attention mechanism to capture significant information from both SMILES representation and graph representation.

Table 2 Prediction performance on Davis dataset

Method	CI	MSE	r_m^2
KronRLS	0.871	0.379	0.407
SimBoost	0.872	0.282	0.644
Sequence-based approaches			
DeepDTA	0.878	0.261	0.630
WideDTA	0.886	0.262	0.633
GANsDTA	0.881	0.276	0.653
MATT_DTI	0.890	0.229	0.682
Graph-based approaches			
DeepGS	0.882	0.252	0.686
GraphDTA	0.893	0.229	0.649
GSATDTA (ours)	**0.906**	**0.200**	**0.732**

Table 3 Prediction performance on Kiba dataset

Method	CI	MSE	r_m^2
KronRLS	0.782	0.411	0.342
SimBoost	0.836	0.222	0.629
Sequence-based approaches			
DeepDTA	0.863	0.194	0.673
WideDTA	0.875	0.179	0.675
GANsDTA	0.866	0.224	0.675
MATT_DTI	0.889	0.150	0.756
Graph-based approaches			
DeepGS	0.860	0.193	0.684
GraphDTA	0.889	0.147	0.674
GSATDTA (ours)	**0.902**	**0.126**	**0.790**

4.5 Ablation experiment

To further validate the effect of the different components in GSATDTA, we designed two variants: GSATDTA-a and GSATDTA-b. GSATDTA-a is mainly for investigating the graph–sequence attention, while GSATDTA-b is used to demonstrate the effectiveness of the efficient transformer.

- GSATDTA-a used BiGRU to learn the SMILES representation and GIN to learn the graph representation. Then, it directly contacted the two representations without graph–sequence attention. For the protein, GSATDTA-a utilized the efficient transformer to learn the protein representation.
- GSATDTA-b used BiGRU to learn the SMILES representation and GIN to learn the graph representation. Then, it utilized the graph–sequence attention to fuse the SMILES representation and the graph representation. For the protein, GSATDTA-b replaced the efficient transformer with CNN to learn the protein representation.

From Table 4, we can find that in the DTA prediction task, the performance of GSATDTA-a is worse than GSATDTA. These results demonstrate that the graph–sequence attention applied

Table 4 Ablation experiments on Davis and Kiba

	Davis			Kiba		
Method	CI	MSE	r_m^2	CI	MSE	r_m^2
GSATDTA	0.906	0.200	0.732	0.902	0.126	0.790
GSATDTA-a	0.899	0.211	0.712	0.894	0.133	0.764
GSATDTA-b	0.894	0.216	0.697	0.890	0.136	0.752

in GSATDTA is beneficial to learning a comprehensive representation of the drugs. Furthermore, we can also observe that GSATDTA-b performs worse than GSATDTA. The result of ablation experiments indicates that the efficient transformer is efficient and effective to learn a good representation of proteins.

Further, in an attempt to verify our hypothesis about the effectiveness of BiGRU in capturing contextual information from input SMILES, we evaluated the performance of the proposed GSATDTA on the Davis dataset using different versions of RNNs as presented in Fig. 3. We find that simple RNN attains the highest MSE value with 0.224, and there is 0.006 improvement achieved when using GRU. Also, BiLSTM achieved an extra improvement of 3.7%, while BiGRU obtained the lowest MSE value with 0.200, which outperformed the BiLSTM performance by 4.8%. This experiment demonstrates the effectiveness of using BiGRU for modeling the SMILES sequence input.

In the molecular property prediction task, Guo *et al.*[15] proposed that integrating the capabilities of both molecular graphs and SMILES sequences can further enhance the model performance. To verify this work in the DTA prediction task, we also conduct an ablation experiment to investigate whether integrating the capabilities of both molecular graphs and SMILES sequences can further enhance the model performance in the drug–target binding affinity prediction task. We designed another two variants: GSATDTA-G and GSATDTA-S. GSATDTA-G uses GIN to learn the graph representation while GSATDTA-S utilizes BiGRU to learn the SMILES representation. For the protein, these two variants both employ the efficient transformer to learn the protein representation. To make the comparison as fair as possible, we chose GSATDTA-a to compare with them. Furthermore, for the evaluation metrics of GSATDTA-G and GSATDTA-S, we use the same metrics as our proposed method, which are the MSE, r_m^2, and CI.

Fig. 4 and 5 illustrate the predicted against measured (actual) binding affinity values for the Kiba dataset. A perfect model is expected to provide a $p = y$ line where predictions (p) are equal to the measured (y) values. From Fig. 4 and 5, we can observe that compared with GSATDTA-S and GSATDTA-G, GSATDTA-a is denser around the $p = y$ line. More specifically, in regions ① and ④, GSATDTA-S performed better than GSATDTA-G, while in regions ②, ③, and ⑤, GSATDTA-G performed better than GSATDTA-S. For GSATDTA-a, only a few points are spread in these areas. From Fig. 4 and 5, we can also find that the overall trend of GSATDTA-a is more similar to GSATDTA-G, but in regions ① and ④, GSATDTA-a is more similar to GSATDTA-S and performs better than GSATDTA-G. We believe that the topological structure information of the molecular graph is critical to the DTA prediction task, but the local context features in SMILES sequences can be used as supplemental information to predict drug–target binding affinity. In conclusion, the results of visualization indicate that integrating the capabilities of both molecular graphs and SMILES sequences indeed can further enhance the model performance in the drug–target binding affinity prediction task, which is consistent with the viewpoint of Guo *et al.*[15]

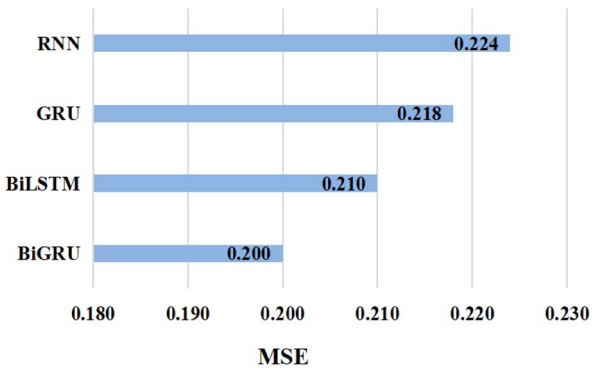

Fig. 3 The MSE value attained by implementing GSATDTA using different types of RNN on the Davis dataset.

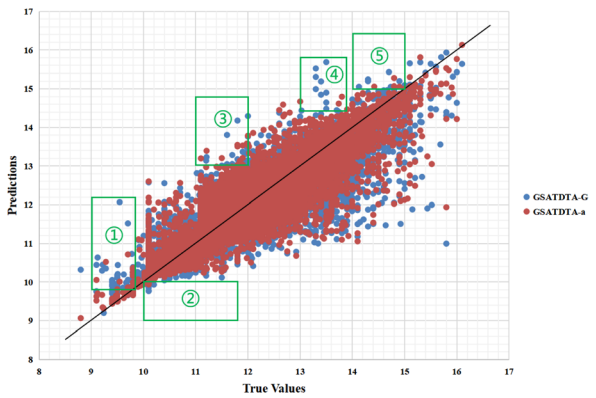

Fig. 4 Predictions of the GSATDTA-a and GSATDTA-G model against measured (real) binding affinity values for the Kiba dataset.

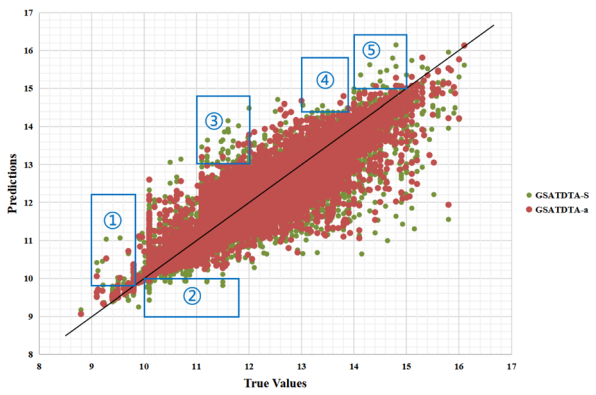

Fig. 5 Predictions of the GSATDTA-a and GSATDTA-S model against measured (real) binding affinity values for the Kiba dataset.

Fig. 6 and 7 show the visualization of predicted against measured (actual) binding affinity values for the Davis dataset. We can find similar performance in Fig. 6 and 7. In region ①, GSATDTA-G performed better than GSATDTA-S, while in region ②, GSATDTA-S performed better than GSATDTA-G. GSATDTA-a performed better than GSATDTA-G and GSATDTA-S in region ③ and there are only a few points spread in these areas for GSATDTA-a.

Furthermore, we also represent the prediction performance of our proposed model based on the predicted value and measured (actual) value (Fig. 8 and 9).

In the Kiba dataset, we analyzed the samples with large errors on the test set and found that when the protein sequence length exceeds 1000, there will be large errors. This is because we fixed the length of the protein sequence to 1000 according to DeepDTA[20] and DeepGS,[27] which leads to information loss when extracting protein features. There are 4086 samples in the test set with protein sequence lengths longer than 1000. We calculated that the MSE of these samples is 0.174, which is higher than the overall MSE (0.126) of the test set. The MSE of the remaining 15 623 samples in the test set is 0.120, which is lower than the overall MSE of the test set. Therefore, when the sequence length of the protein is longer than 1000, our model suffers from large prediction errors.

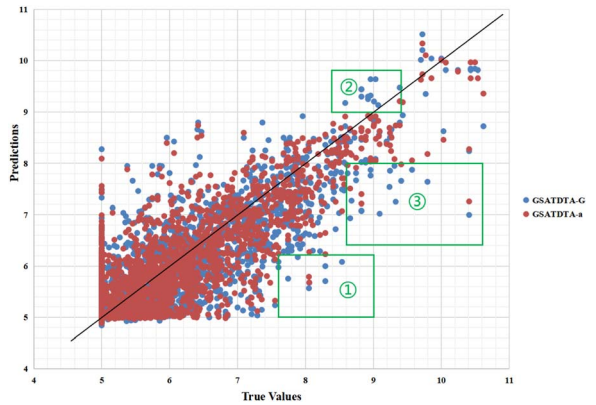

Fig. 6 Predictions of the GSATDTA-a and GSATDTA-G model against measured (real) binding affinity values for the Davis dataset.

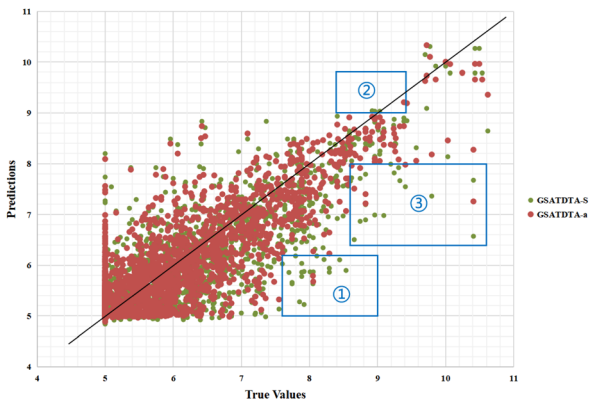

Fig. 7 Predictions of the GSATDTA-a and GSATDTA-S model against measured (real) binding affinity values for the Davis dataset.

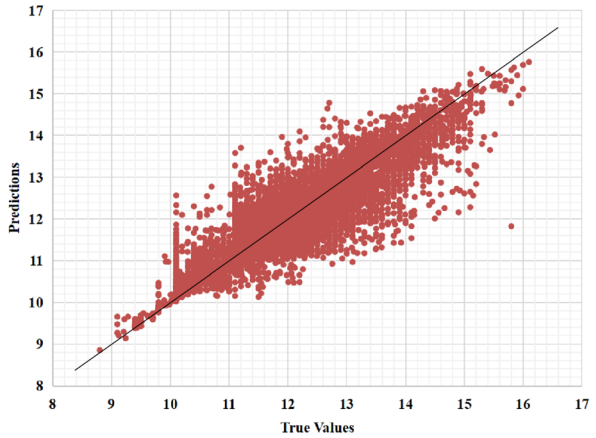

Fig. 8 Predictions of the GSATDTA model against measured (real) binding affinity values for the Kiba dataset.

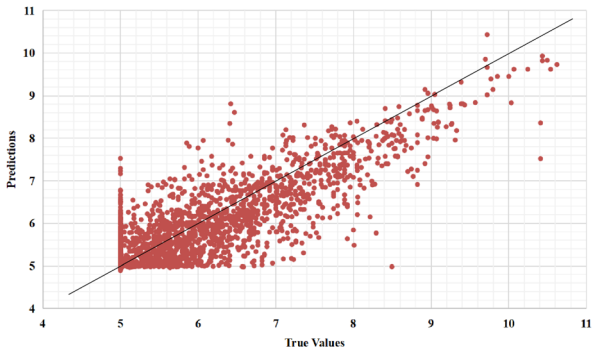

Fig. 9 Predictions of the GSATDTA model against measured (real) binding affinity values for the Davis dataset.

We can find the same performance in the Davis dataset. In the Davis test set, there are 1333 samples with protein sequence lengths longer than 1000. We calculate that the MSE of these samples is 0.214, which is higher than the overall MSE (0.200) of the test set. The MSE of the remaining 3677 samples in the test set is 0.192, which is lower than the overall MSE of the test set.

5 Conclusions

In this paper, we proposed a novel model based on self-attention, called GSATDTA, to predict the binding affinity between drugs and targets. We leveraged both the graph and sequence information and proposed a graph–sequence attention mechanism to learn effective drug representations for DTA prediction. Furthermore, we utilized an efficient transformer to learn the representation of proteins, which can capture the long-distance relationships in the sequence of amino acids. Extensive experimental results show that our model outperforms the state-of-the-art models in terms of MSE, CI, and r_m^2 on two independent datasets. Since the transformer has achieved great success in various tasks, in the future, we will

consider investigating the graph transformer to learn the representations of the drug to predict drug–target binding affinity.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61972135), the Natural Science Foundation of Heilongjiang Province in China (No. LH2020F043).

Notes and references

1 J. A. DiMasi, H. G. Grabowski and R. W. Hansen, *J. Health Econ.*, 2016, **47**, 20–33.
2 A. D. Roses, *Nat. Rev. Drug Discovery*, 2008, **7**, 807–817.
3 Y. Zeng, X. Chen, Y. Luo, X. Li and D. Peng, *Briefings Bioinf.*, 2021, **22**, bbab117.
4 Q. Zhao, H. Zhao, K. Zheng and J. Wang, *Bioinformatics*, 2022, **38**, 655–662.
5 T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le and S. Venkatesh, *Bioinformatics*, 2021, **37**, 1140–1147.
6 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.
7 J. You, B. Liu, Z. Ying, V. Pande and J. Leskovec, *Adv. Neural Inf. Process. Syst.*, 2018, **31**, 6412–6422.
8 X. Lin, Z. Quan, Z.-J. Wang, H. Huang and X. Zeng, *Briefings Bioinf.*, 2020, **21**, 2099–2111.
9 Z. Quan, Y. Guo, X. Lin, Z.-J. Wang and X. Zeng, *2019 IEEE International Conference on Bioinformatics and Biomedicine*, BIBM, 2019, pp. 717–722.
10 A. C. Cheng, R. G. Coleman, K. T. Smyth, Q. Cao, P. Soulard, D. R. Caffrey, A. C. Salzberg and E. S. Huang, *Nat. Biotechnol.*, 2007, **25**, 71–75.
11 G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell and A. J. Olson, *J. Comput. Chem.*, 2009, **30**, 2785–2791.
12 X. Chen, M.-X. Liu and G.-Y. Yan, *Mol. BioSyst.*, 2012, **8**, 1970–1978.
13 F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang and Y. Tang, *PLoS Comput. Biol.*, 2012, **8**, e1002503.
14 X.-Y. Yan, S.-W. Zhang and S.-Y. Zhang, *Mol. BioSyst.*, 2016, **12**, 520–531.
15 Z. Guo, W. Yu, C. Zhang, M. Jiang and N. V. Chawla, *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 435–443.
16 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, 5998–6008.
17 H. Li, K.-S. Leung, M.-H. Wong and P. J. Ballester, *Molecules*, 2015, **20**, 10947–10962.
18 T. Pahikkala, A. Airola, S. Pietilä, S. Shakyawar, A. Szwajda, J. Tang and T. Aittokallio, *Briefings Bioinf.*, 2015, **16**, 325–337.
19 T. He, M. Heidemeyer, F. Ban, A. Cherkasov and M. Ester, SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines, *J. Cheminf.*, 2017, **9**, 24.
20 H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, i821–i829.
21 H. Öztürk, E. Ozkirimli and A. Özgür, *arXiv*, 2019, preprint, arXiv:1902.04166, DOI: 10.48550/arXiv.1902.04166.
22 M. Woźniak, A. Wołos, U. Modrzyk, R. L. Górski, J. Winkowski, M. Bajczyk, S. Szymkuć, B. A. Grzybowski and M. Eder, *Sci. Rep.*, 2018, **8**, 1–10.
23 C. J. Sigrist, L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, V. Bulliard, A. Bairoch and N. Hulo, *Nucleic Acids Res.*, 2010, **38**, D161–D166.
24 L. Zhao, J. Wang, L. Pang, Y. Liu and J. Zhang, *Front. Genet.*, 2020, **10**, 1243.
25 M. Tsubaki, K. Tomii and J. Sese, *Bioinformatics*, 2019, **35**, 309–318.
26 K. Y. Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, P. Zhang, *et al.*, *Int. Joint Conf. Artif. Intell.*, 2018, 3371–3377.
27 X. Lin, K. Zhao, T. Xiao, Z. Quan, Z.-J. Wang and P. S. Yu, *ECAI 2020*, IOS Press, 2020, pp. 1301–1308.
28 T. Song, J. Jiang, W. Li and D. Xu, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 2020, **13**, 2853–2860.
29 T. Song, F. Meng, A. Rodriguez-Paton, P. Li, P. Zheng and X. Wang, *IEEE Access*, 2019, **7**, 166823–166832.
30 Z. Quan, X. Lin, Z.-J. Wang, Y. Liu, F. Wang and K. Li, *2018 IEEE International Conference on Bioinformatics and Biomedicine*, BIBM, 2018, pp. 728–733.
31 T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, *Adv. Neural Inf. Process. Syst.*, 2013, **26**, 3111–3119.
32 T. Song, S. Pang, S. Hao, A. Rodríguez-Patón and P. Zheng, *Neural Process. Lett.*, 2019, **50**, 1485–1502.
33 F. Gong, C. Li, W. Gong, X. Li, X. Yuan, Y. Ma and T. Song, *Comput. Intell. Neurosci.*, 2019, **2019**, 1939171.
34 J. Chung, C. Gulcehre, K. Cho and Y. Bengio, *International conference on machine learning*, 2015, pp. 2067–2075.
35 K. Xu, W. Hu, J. Leskovec and S. Jegelka, *7th International Conference on Learning Representations*, ICLR, 2019, p. 2019.
36 Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, *et al.*, *J. Med. Chem.*, 2020, **63**, 8749–8760.
37 Q. Zhang and Y.-B. Yang, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 15475–15485.
38 D. Ulyanov, A. Vedaldi and V. Lempitsky, *arXiv*, 2016, preprint, arXiv:1607.08022, DOI: 10.48550/arXiv.1607.08022.