Briefings in Bioinformatics, 00(0), 2019, 1–13

doi: 10.1093/bib/bbz125
Advance Access Publication Date:
Problem solving protocol

A novel molecular representation with BiGRU neural networks for learning atom

Xuan Lin*, Zhe Quan*, Zhi-Jie Wang, Huang Huang and Xiangxiang Zeng

Corresponding author: Zhi-Jie Wang, School of Data and Computer Science, Sun Yat-sen University, Guangzhou, 510275, China.
E-mail: wangzhij5@mail.sysu.edu.cn
*The first two authors contribute equally to this paper.

Abstract

Molecular representations play critical roles in researching drug design and properties, and effective methods are beneficial to assisting in the calculation of molecules and solving related problem in drug discovery. In previous years, most of the traditional molecular representations are based on hand-crafted features and rely heavily on biological experimentations, which are often costly and time consuming. However, recent researches achieve promising results using machine learning on various domains. In this article, we present a novel method named Smi2Vec-BiGRU that is designed for learning atoms and solving the single- and multitask binary classification problems in the field of drug discovery, which are the basic and also key problems in this field. Specifically, our approach transforms the molecule data in the SMILES format into a set of sample vectors and then feeds them into the bidirectional gated recurrent unit neural networks for training, which learns low-dimensional vector representations for molecular drug. We conduct extensive experiments on several widely used benchmarks including Tox21, SIDER and ClinTox. The experimental results show that our approach can achieve state-of-the-art performance on these benchmarking datasets, demonstrating the feasibility and competitiveness of our proposed approach.

Key words: machine learning; molecular representation; recurrent neural networks; drug discovery

Introduction

Data-driven analysis plays a crucial role in many biological and chemical applications, such as molecule modeling and chemical property prediction [1, 2]. With the rapid development of machine learning techniques [3], in recent years researchers in the fields of bioinformatics and cheminformatics have attempted to utilize machine learning methods for molecule modeling, chemical property prediction, protein–protein interactions biology analysis and so on [4–7].

As we know,simplified molecular input line entry system [8] (SMILES) strings are usually used to represent and store molecule datasets, and they are in form of a single-line text consisting of molecular notations. In the real world, a molecule with an arbitrary size and shape could be hard to be represented and

Xuan Lin is a Ph.D candidate student in Hunan University. His research interests include machine learning, bioinformatics.
Zhe Quan is an associate professor in Hunan University. His main research interests include machine learning, artificial intelligence, parallel and high-performance computing.
Zhi-Jie Wang is a research associate professor at the Sun Yat-Sen University. His current research interests include data mining, artificial intelligence, databases, machine learning.
Huang Huang is a PhD candidate student in National University of Defense Technology. His research interests include parallel and high-performance computing.
Xiangxiang Zeng is a professor in Hunan University. His research interests include bio-computing and bioinformatics.
Submitted: 21 June 2019; **Received (in revised form):** 15 August 2019

used for machine learning tasks. Users usually need to transform them into other formats that are easy to be handled by machine learning algorithms. A widely adopted proposal is to use handcrafted features like extended connectivity fingerprints (ECFP) [9], Coulomb matrix [10], graph-like structure [11] and so on. Such a process is usually called 'featurization'. The transformed data (or featurization data) is usually used as the input and fed into the interface of machine learning methods, such as 'random forest' (RF), 'multilayer perception' and so on [12]. These featurization methods are also called '2D' molecular descriptors that try to extract relevant structural features derived from the molecular graph.

ECFP [9], as one of the most common representations in the above class, is referred to as the circular or Morgan fingerprints. In brief, each atom is firstly preprocessed to be assigned an integer identifier at the initial stage, and 'a bag of fragments' is constructed by iteratively expanding outward along bonds and then is hashed into a fixed-length representation or fingerprint after a duplicate identifier removal stage. The Coulomb Matrix proposed by [10] is another representation, which encodes information by use of the atomic self-energies and internuclear Coulomb repulsion operator. In addition, graph-like structure, state-of-the-art method appeared in recent years, computes an initial feature vector and a neighbor list for each atom. All methods mentioned above need a preprocessing of the chemical software named 'RDKit', and some of them are computationally complicated. Therefore, it is meaningful to develop better methods, in a novel prospective, for learning molecular features.

On the other hand, owing to the success of solving a wide range of machine learning problems by the artificial neural networks [13], recently, recurrent neural networks including long short-term memory (LSTM) [14], gated recurrent unit (GRU) [15] and their variants have emerged as powerful generative models in various domains including natural language processing (NLP) [16, 17] and image processing [18]. These lines of models regard the input data as sequential lists, and they are very suitable for solving time-dependent tasks like natural language understanding [19]. In the meanwhile, as shown in [20], the Atom2Vec [21] can learn the basic properties of atoms and is used to discover the periodic table of the elements.

Motivated by the remarkable achievements mentioned above, in this paper we attempt to develop a novel approach that merges the merits of the above techniques to learn low-dimensional vector representation for molecule and is designed for solving the single-task and multitask binary classification problems in the field of drug discovery. Such problems are the basic and also key problems in this field, since many other tasks [e.g. drug–target interactions (DTIs) and protein–protein interactions] significantly rely on the quality of the classification result. Generally, our approach first transforms the molecule data in the SMILES format into a set of sample vectors via a representation method named 'Smi2Vec'; meanwhile, it divides the molecule in SMILES format into atoms by spaces and extracts the atomic group, which may consists of symbols and numbers. These atoms are then initially encoded by one-hot encoding, which allows us to transform atoms into a specific vectors with some certain dimensions. Then it uses a manner similar to word2vec [21] to extract the sample vectors by training the specific vectors previously obtained. These vectors are then sliced together and fed into the embedding layer of our neural network. To extract high-dimensional features, in the embedding layer a large atomic matrix is constructed, which is convenient for model training in the latter steps. The extracted features are then trained, and finally the trained samples are sent to a classifier (e.g. 'sigmoid') for single- and multitask binary classification or property prediction. In summary, the main contributions of this paper are as follows:

1. We present a novel approach named Smi2Vec-BiGRU that is designed for learning atoms and solving the single- and multitask binary classification problems in the field of drug discovery. Unlike other representation methods, our method transforms molecules in the SMILES format into atom vectors, and it takes measures to extract the atomic groups from the molecules, which are used as the heavy atoms to better represent the structural information in molecules. Meanwhile, it leverages a powerful model, bidirectional GRU (BiGRU) neural network, which is initially developed for solving problems in NLP and image processing, to train the sample vectors embedded in the atomic matrix.
2. We conduct extensive experiments based on several widely used molecule datasets to evaluate the performance of our proposed method. The experimental results indicate that, for the single- and multitask binary classification problems in the field of drug discovery, our proposed approach can achieve competitive performance, compared against classic and state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 reviews prior works most related to ours. Section 3 presents our approach; for the completeness of this paper, the LSTM neural network-based model, appeared in the preliminary version [22], is also covered. Section 4 analyzes and discusses the experimental results. Section 5 concludes this paper.

Related work

Molecular representation

Molecular representation is diverse. The main molecular representation methods at present are ECFP [9], Coulomb matrix [10], graph-like structure [11] and so on. Recently, deep neural network (DNN) models have opened up new avenues for modeling SMILES strings as a language model. As shown in [20], Atom2Vec can learn the basic properties of atoms. Our work is inspired by theirs; yet, it is different from theirs. In that paper the methods and experiments are used to discover the periodic table of the elements. In addition, they mainly focused on the principle explanation of atom representation, while related model designs are not covered. Another unsupervised approach, named conditional diversity networks [23], also transformed SMILES strings into vectors, while the detailed steps are not covered. Moreover, there are several studies addressing deep learning-based embedding for representation of SMILES string. For example, Kushner *et al.* [24] proposed a variational autoencoder (VAE), in which data is represented as a parse tree from a context-free grammar. Further, Jin *et al.* [25] presented a junction tree VAE to generate molecular graphs that allows to incrementally expand molecules at every step. Meanwhile, Gomez *et al.* [26] reported an autoencoder to convert discrete representations of molecules to and from a multidimensional continuous representation. This line of works paid more attention on the generation of molecule and drug instead of the tasks mentioned in this paper.

Drug prediction and interaction

Our work is also related to drug discovery or prediction including adverse drug events (ADEs), drug reaction effects (DREs), drug–drug interactions (DDIs), DTIs, *in silico* drug repurposing and so

on. For example, Page *et al.* [27] identified the ADEs by relational learning. In addition, several methods proposed by [28] are for extracting DREs from forum posts and tweets. In [29], a network-based deep learning approach 'deepDR' was proposed specialized for *in silico* drug repurposing. Cheng *et al.* [30] developed a comprehensive source and free tool for assessment of chemical admet properties. Xiao *et al.* [31] provided the efficient solutions for the real-world DRE prediction and cast the DRE–drug relation structure into a three-layer hierarchical Bayesian model. Xiang *et al.* [32] proposed a naive Bayesian model that was constructed on a gene–adverse drug reaction (ADR) network for the rapid assessment of clinical ADRs with the frequency estimation. As for the adverse DDIs, most of methods focused on the binary prediction (with or without DDI). Cheng *et al.* [33] proposed a network-based method to identify clinically efficacious drug combinations for hypertension and cancer. Warmuth *et al.* [34] used the active learning techniques for selecting the successive batches and adopted three selection strategies in the drug discovery process. Ma *et al.* [35] made better trade-off between accuracy and interpretability. Ezzat *et al.* [36] proposed two kinds of matrix factorization approaches that adopt the regularization technique to learn the manifolds and developed a preprocessing step to enhance predictions. Besides, Cheng *et al.* [37] developed a inhibitor predicting models for five major CYP isoforms by using a combined classifier algorithm. Yu *et al.* [38] provided an integrative framework to predict drugs for hepatocellular carcinoma based on multi-source random walk (PD-MRW). And Khalid and Sezerman [39] combined both sequence and structural features for predicting HIV resistance by applying support vector machine (SVM) and random forests classifiers. Our work is different from this line of works in several points at least: (i) we use a different representation method for learning atoms in molecules; and (ii) we use the BiGRU neural networks to train the sample vectors, instead of some traditional methods such as SVM and RF.

DNN-based methods

In the past decade, DNNs have gained remarkable achievements in various research domains. DNNs can learn the potential regular pattern to obtain better performance analysis and prediction, by training a large number of datasets [40–42]. Different from other domains such as NLP and image processing, DNN in drug discovery depends heavily on molecular featurization [43, 44]. Moreover, it also relies on the prediction of molecular property and activity [11, 45–47]. Our work shares two common features with this line of works: (i) both (i.e. our method and this line of methods) discussed the issues related to drug discovery; and (ii) both are belonged to the DNN-based methods. Nevertheless, our work is different from theirs in two points at least: (i) they mainly focused on developing techniques for other tasks instead of single and multitask binary classification; and (ii) the BiGRU recurrent neural networks are not covered in their works.

This article provides a more complete understanding of learning atoms first presented in the conference version [22], providing additional insights, analysis and evaluation. Furthermore, we improve the original framework in two aspects. First, we propose the Smi2Vec, which replaces the format transformer in the original framework. Compared against format transformer, here Smi2Vec adds the notion of atomic group, which originates from the consideration of the structural information in molecules. Second, we propose using the BiGRU neural network to train sample vectors, instead of using the LSTM recurrent neural network presented in the original framework. The experimental results show that the refined method is much more efficient than that presented in the conference version.

Method

In this section, we first give an overview of our approach and then present each part of our approach in detail.

Overview

The architecture of our approach is as shown in Figure 1. The whole framework mainly consists of two parts: Smi2Vec and BiGRU neural network. Briefly, the molecule data in the format of SMILES is first processed by a representation method named 'Smi2Vec', which transforms the molecule data in the SMILES format into a set of sample vectors. These vectors are then sliced together based on some rules, which are used to construct the atomic matrix as the input of BiGRU neural network, and then the atomic matrix is to be trained by using the BiGRU neural networks. The output of the BiGRU neural networks shall be processed by a classifier, which is used to generate the output label for task classification.

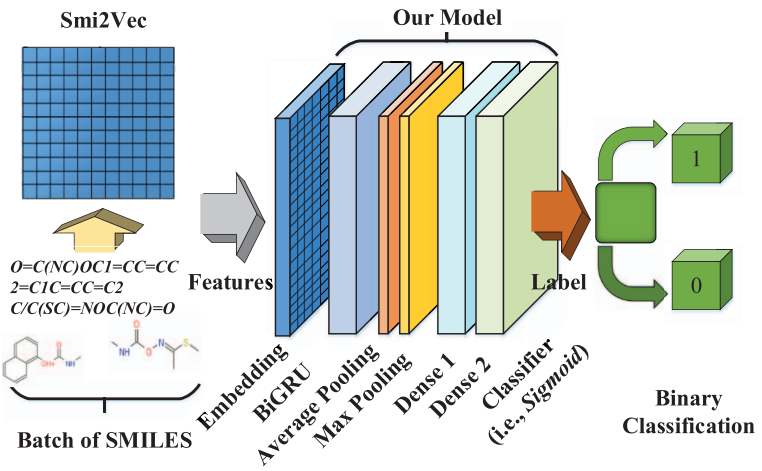

Figure 1. The specific process of Smi2Vec and the composition of BiGRU neural network.

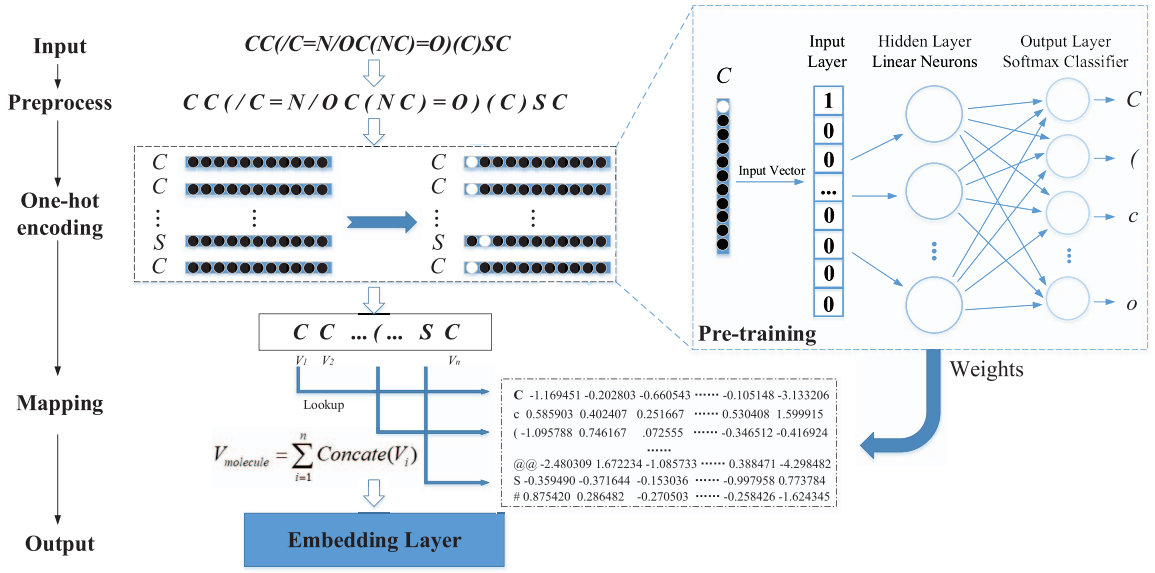

Figure 2. Sketch of 'Smi2Vec'.

Smi2Vec

The preliminaries

Choosing a proper molecular representation is at the heart of computer-based chemical analysis, and it is also very important for drug discovery and prediction, since one may need to analyze and predict properties of drug with the same or similar molecule representation. In the real world, most biological and chemical datasets are in the format of SMILES string. The SIMILES string of a unique molecule is a single-line text representation. For example, a molecule is encoded as a linearly arranged string $s_i = s_1, s_2, ...s_i (i = 1, 2, ..., n)$. The encoding rules of SMILES follows the strict grammars, which consist of symbols indicating element types, bond values, and the start and terminal locations for ring closures and branching components.

SMILES strings are powerful for representing and storing the molecule data. To apply machine learning methods for learning advanced features, we need to transform them into a new format suitable for utilization. Take 'Aldicarb' and its SMILES string $CC(/C = N/OC(NC) = O)(C)SC$ as an example; one can convert it by RDKit [48] software into a graph-structured representation, which can be later used for learning features via graph convolutions.

The proposed representation method

Sketch of Smi2Vec. Instead of using chemical software such as RDKit to transform SMILES strings, we adopt another manner that directly transforms them into atom vectors. Briefly, molecule in SMILES format is first preprocessed as an independent atom or symbol (also served as atom in *Smi2Vec*), and then they are expressed as a high-dimensional vectors, which are sample vectors and also machine-readable characters or strings. Figure 2 illustrates the sketch of this representation method. Next, we present the details.

The detailed steps of Smil2Vec. As we known, in NLP, the sentences are processed using word vectors. We observe that SMILES is a linguistic grammar that employs an alphabet of characters to describe the molecule, and each element or symbol has an associated definition in SMILES. In our preliminary version [22], we used the similar way in NLP to handle the SMILES strings. Specifically, we used the 'format transformer' to process it; more details can be found in [22]. In this article, we further revise it by adding the notion of 'atomic group'. This idea is originated from the consideration of the structural information of the molecule. Generally, for a patch of molecules $s_i = s_1, s_2, ... s_i (i = 1, 2, ..., n)$ in the format of SMILES, we divide them into a series of atoms by space, which is similar to the word segmentation process in NLP. Each single atom x_i may consist of different symbols and numbers, such as '(', ')', '[', ']', '@' and so on. We realize that some atomic groups in the molecule appear in a special way such as '[C@H]'. For the 'atomic groups' like the above, we 'treat' them and/or its variants as 'separate biological word' (i.e. atoms and symbols). Then, for all preprocessed atoms, we initially encode them by 'one-hot encoding' [49], which allows us to transform atoms into a specific vectors $v_i (i = 1, 2...n)$ with some certain dimensions. Since these specific vectors have less feature information, we need to construct a mapping from specific vectors to sample vectors. Here we use a manner similar to 'word2vec' [21] to 'extract' the sample vectors (by pretraining the specific vectors previously obtained).

Before the extract process (i.e. training the specific vectors), we suggest a technique, named 'simplified feature learning' (SFL), to accelerate the extraction of atom features. This technique is based on the follow observation: molecules in SMILES format consist of numbers and characters, while some symbols and numbers may represent repetitive information, which may cause the complicated training process. For example, Toluene is denoted as *Cc1ccccc1* in SMILES, a benzene ring is represented by number '1', while *c* and *C* denote aromatic and aliphatic carbon atoms, which essentially imply the existence of a benzene ring. Therefore, our SFL ignores the assigned numbers in SMILE string, which are redundant feature information, since they has been already expressed. Instead, our SFL adds the occurrence frequency of each element (as the additional information) to the specific vector. These strategies ensure the simplicity and integrity of the feature information.

After the sample vectors are extracted, they shall then be sliced together and used to construct the 'atomic matrix' as the input of the BiGRU neural network. Here the atomic matrix is

Algorithm 1 Smi2Vec

Input: a molecule s in the format of SMILES, dictionary D, atom vector s, fixed length m, vector dimension d.
Output: atom matrix A
1: atom set $\{x_j | 1 \le j < |s|\} \longleftarrow split(s)$
2: **for** j=1 to m **do**
3: **if** $x_j \notin dictionary$ **then**
4: embedding vector $a_j \longleftarrow$ randomly generated value $\in \Re^d$
5: **else**
6: $a_j \xleftarrow{map} x_j$ // **by using** D
7: **end for**
8: atom matrix $A \longleftarrow \sum_{j=1}^{m} a_j$
9: **return** $A \in \Re^{m \times d}$

convenient for model training in the later steps. The size of the matrix depends on the product of the batch size and the limited size (i.e. the dimension of an atom vector). Note that every vector v_i is encoded using an N-bit status register; each state has its own register bit, and at any time only one of them is valid. The construction process of an atomic matrix is shown in Figure 3. It is based on the following principle: the element v_i located in the center of window k will be output object, and the other elements are the input ones [for example, see Figure 2 again, if atoms $v_i^*(i = 1, 2, ..., 22)$ located in window k are the inputs, then the atom $v_i(i = 11)$ located in the central of window k will be the output]. In this way, it guarantees the output transmission channel unobstructed and ordered. In addition, each vector v_i initially encoded by one-hot encoding shall automatically find the corresponding vector in the pretrained atomic list, until the mapping process completes. For ease of reference, Algorithm 1 summarizes the detailed steps of the Smi2Vec. In general, A SMILES string with fixed length m is divided into a separate atom or symbol (Line 1). Then it 'maps' atom by looking up each of the atom embeddings from the pretrained dictionary, while it randomly generates values if it is not in dictionary (Line 2-7). Finally, it constructs an atom matrix A by aggregating embedding vectors (Line 8), where each line represents the pretrained vector of an atom.

Recurrent neural networks

LSTM neural network

Similar to the methods for dealing with semantics similarity in NLP, our preliminary version [22] adopts the LSTM recurrent neural network. The LSTM is an alternative RNN, it uses the so-called 'memory cell' (controlled by input, output and forget gates) to replace the 'conventional neuron' in order to overcome the vanishing gradient problem of traditional RNNs. In short, LSTM is a special class of RNN that is capable of capturing long sentence relationships.

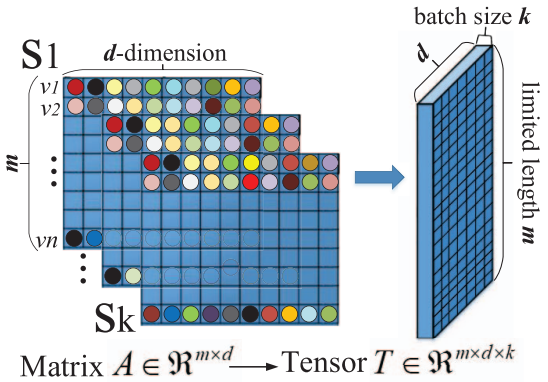

Figure 3. The workflow of atomic matrix.

Owing to the existence of the gate of the adopted model, we can learn and recognize the information needed to be retained or forgotten. Each atom including special symbols (e.g. = and ≡) has a corresponding time step x_t (t=1,2...n). The intermediate state associated with each time step is referred to as a hidden state vector h_t. This hidden state vector is used to encapsulate and summarize all the information appeared in the previous time step. The hidden state is a function of the current atom vector and the hidden state vector of the previous step. The hidden state vector h_t and the output gate vector o_t is computed as follows:

$$\begin{cases} h_t = \sigma(W^H h_{t-1} + W^X x_t + b_h) \\ o_t = \sigma(W^Y h_t + b_o), \end{cases} \tag{1}$$

where W^X represents the weight matrix between the input and hidden layer; W^H represents the recurrent weight matrix between the hidden layer and itself; W^Y represents the weight matrix between the hidden and output layers; b_h and b_o are bias in the hidden and output layers, respectively.

The value of W^H stays the same in all time steps, but the value of W^X changes in every input. The size of these values is not only affected by the current vector, but also affected by the previously hidden layer. It is easily observed that the value of h_t will vary when other elements such as x_t changes, the current hidden state vector is more concerned with the sum of the previous atom vectors than the current one. See Figure 4(a) for an illustration of the LSTM unit.

BiGRU neural network

Although LSTM recurrent neural network and its variants have made great strides in NLP, especially text classification tasks, we cannot ignore its limitations, such as too many parameters, complicated internal calculations and so on. Besides LSTMs, another type of recurrent unit used to deal with variable-length sequences is the GRU, which was proposed in [50]. Next, we first briefly discuss this technique for ease of understanding the BiGRU adopted in our paper.

The main idea of GRU is to make every recurrent unit to adaptively capture dependencies of different time scales. The GRU model can keep the same performance of LSTM and has advantages of simpler structure, fewer parameters and better convergence. Similar to the LSTM unit, it only composes of an update gate and a reset gate, without having a separate memory cells.

Let $X = (x_1, x_2, ..., x_t)$ be the input sequence, where $x_t \in \Re^d$. The current hidden state h_t^i of each GRU unit at time t is a linear interpolation between the previous hidden state h_{t-1}^i and the candidate hidden state $\tilde{h}_t^i$. The current hidden state h_t^i, the candidate hidden state $\tilde{h}_t^i$, the update gate z_t^i and the reset gate r_t^i are computed as follows [51]:

$$\begin{cases} h_t^i = (1 - z_t^i) h_{t-1}^i + z_t^i \tilde{h}_t^i \\ \tilde{h}_t^i = \Phi(W x_t + U(r_t \odot h_{t-1}))^i \\ z_t^i = \sigma(W^Z x_t + U^Z h_{t-1})^i \\ r_t^i = \sigma(W^R x_t + U^R h_{t-1})^i, \end{cases} \tag{2}$$

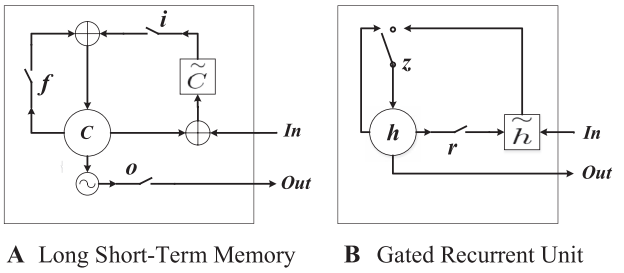

Figure 4. (**A**). i, f and o are the input, forget and output gate, respectively. c and $\tilde{c}$ represent the memory cell and the new memory cell content, respectively. (**B**). r and z are reset and update gates, and h and $\tilde{h}$ are the activation and the next activation.

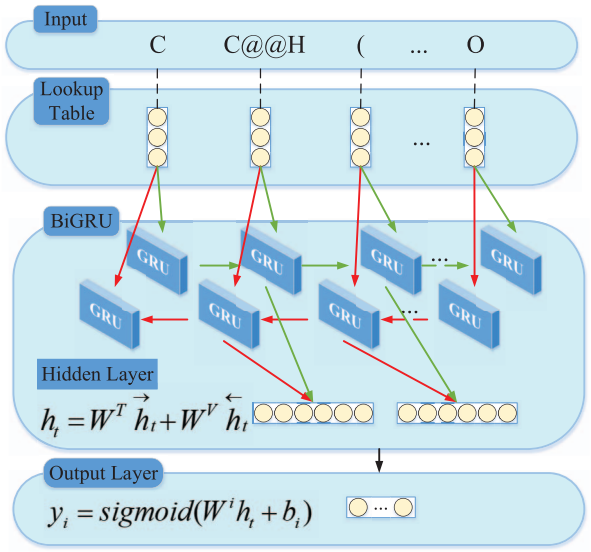

Figure 5. The main architecture of BiGRU neural network. Recurrent neural networks with GRU units pick up information along the forward and backward propagation. Take *C[C@@H](O)C(=O)O* as an example to illustrate the principle of our approach.

where Φ and σ represent different activation functions (i.e. 'tanh' and 'Sigmoid'), W, W^Z, W^R, U, U^Z and U^R denote the corresponding weight coefficients. An update gate z_t^i regulates how much the recurrent unit computes its hidden state. Similar to the LSTM unit, this computation procedure is involved with a linear sum between the current state and the following calculation state. And r_t represents a set of reset gates; $\odot$ is an element-wise multiplication. When r_t is close to 0, the reset gate makes the recurrent unit allow to forget the previously computation state. See Figure 4(b) for an illustration of the GRU.

Due to the unidirectional nature of GRU, it is impossible to encode information from the back to the front. This may affect the acquisition of partial structural information of molecular compound in the format of SMILES string. In the fine-grained classification problem, such as drug performance, biological activity and other classification tasks, it needs to pay more attention on the interaction between different atoms, atomic groups and inter-structural information. Therefore, in this paper we adopt a BiGRU [52], instead of using directly the GRU neural network mentioned earlier. See Figure 5 for an illustration of BiGRU neural network. The BiGRU processes the input sequences in both directions with two sublayers, in order to capture the full input molecule in the format of SMILES string. These two

Table 1. The setting of main parameters in our proposed model

Description	Value
The number of GRU cell in the hidden layer	200
'Loss' Function	binary_crossentropy
optimizer	adam
The fixed length of SMILES sequence	80
Spatial 1D version of dropout	0.2
Batch_size	128
Dimension of a vector	100
blackEpoch	black100
blackNumber of words for pretraining	black10 000
blackBatch size	black128

Note: other omitted parameters are set to default values.

sublayers compute forward and backward hidden sequences $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively. Then, they are combined to compute the current hidden state h_t and the output of BiGRU o_t. More specifically, we have the following:

$$\begin{cases} \overrightarrow{h_t} = GRU(x_t, \overrightarrow{h_{t-1}}) \\ \overleftarrow{h_t} = GRU(x_t, \overleftarrow{h_{t-1}}) \\ h_t = W^T \overrightarrow{h_t} + W^V \overleftarrow{h_t} \\ o_t = \Phi(W^O h_t) \end{cases} \tag{3}$$

where *GRU* function represents the nonlinear transformation of the input atom vectors, which are encoded into the corresponding hidden state of GRU; W^T and W^V represent the weight coefficients corresponding to the forward hidden state $\overrightarrow{h_t}$ and the reverse hidden state $\overleftarrow{h_t}$ in the bidirectional GRU, respectively; W^O is the weight coefficient between the hidden and output layers. Notice that, even if the directions are inconsistent, there is no impact on the value between $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$.

Finally, the output of BiGRU neural network is sent to a classifier for task classification or property prediction. This way, the designed approach with learnable ability can predict whether the tested molecule is toxic or not. More specifically, this paper uses the 'sigmoid' function to calculate the resulting probability of the classification y_i and compares it with the original label $\tilde{y_i}$. The objective function *Loss* and y_i are computed as follows:

$$\begin{cases} Loss = -\frac{1}{N} \sum_{i=0}^{N} (y_i log y_i + (1 - \tilde{y_i}) log(1 - y_i) \\ y_i = sigmoid(W^i h_t + b_i), \end{cases} \tag{4}$$

where W^i and b_i denote the weight coefficient and bias in the output layer, respectively; N is the number of batch size.

Experiment

In this section, we compare our proposed method with the competitors based on several commonly used molecule datasets from the MoleculeNet Benchmark [53]. Note that some datasets are single-task datasets while others are multitask datasets (more detailed descriptions shall be discussed later). In our experiments, the output of the final layer is changed according to the number of tasks, while the other steps in our approach are applied to all tested datasets. Our system was trained based on Tensorflow [54], and we used the Adam algorithm [55] to optimize all the parameters of the adopted neural networks. The specific parameter settings related to our model are shown in Table 1.

Data description

The datasets we used are BACE, Blood–Brain Barrier Penetration (BBBP), Toxicology in the 21st Century (Tox21), Side Effect Resource (SIDER), ToxCast and ClinTox. These datasets consist of a mix of physical and nonphysical properties, used for single-task and/or multitask binary classification problems. The type of objects/items/entities of these datasets are shown in Table 2. In these datasets, SMILES strings are used to encode input molecules. The details of these data are as follows (https://github.com/deepchem/deepchem):

- ***BACE***. The BACE dataset [56] provides quantitative and qualitative binding results and is a collection of 1522 compounds with their 2D structures and binary labels.
- ***BBBP***. The BBBP dataset [57] concentrates on the modeling and prediction of the barrier permeability. This dataset includes binary labels for over 2000 compounds on their permeability properties.
- ***HIV***. The HIV dataset [58] was provided by the Drug Therapeutics Program AIDS Antiviral Screen, which tested the ability to inhibit HIV replication for nearly 41 913 compounds. In general, screening results were evaluated and divided into three subsets; they are confirmed inactive (CI), confirmed active (CA) and confirmed moderately active (CM). Here, the latter two labels were combined as active subset to make a classification task between inactive (CI) and active (CA and CM) subsets.
- ***Tox21***. The Tox21 dataset [59] contains qualitative toxicity measurements for 8014 compounds on 12 different targets, including nuclear receptors and stress response pathways.
- ***SIDER***. The SIDER [60, 61] is a dataset of marked drugs and ADRs; it contains 1427 compounds on 27 system organ classes.
- ***ToxCast***. ToxCast [62] is another data collection that provides toxicology data for a large library of compounds, based on *in vitro* high-throughput screening. It contains 617 classification tasks for 8615 compounds.
- ***ClinTox***. The ClinTox dataset [63, 64] contains 2 classification tasks for 1491 drug compounds with known chemical structures: (i) clinical trial toxicity (or absence of toxicity) and (ii) Food and Drug Administration approval status.

It is well known that conventional machine learning methods require datasets to be split into training/validating/testing subsets for benchmark. Usually, models are trained by training sets; hyperparameters are tuned through validating sets; and testing sets are used for evaluating models/approaches/systems. In our experiments we used two split methods mentioned in [53], i.e. 'random splitting' and 'stratified random splitting'. In brief, random splitting method randomly splits dataset into the train/validate/test subsets. In contrast, the stratified random splitting method first sorts data points in the increasing order according to their label values, and then it splits the dataset into train/validate/test subsets by ensuring the same proportion of positive and negative samples in each subset. Typically, when there is some bias, random splitting is used for data splitting. In contrast, stratified random splitting is used. Typically, when there is some bias, random splitting is used for data splitting. In contrast, stratified random splitting ensures that each subset contains the full range of labels.

To keep the same benchmark, in our experiments random and stratified (random) split methods with an 8/1/1 ratio [here the 8/1/1 ratio means that 8/10, 1/10 and 1/10 of the corresponding dataset (e.g. 'BACE') are used as the train, validate and test set, respectively] was adopted for BBBP and BACE datasets, while only stratified (random) split method was adopted for Tox21, ClinTox, SIDER and ToxCast datasets (since random split exhibits poor AUC-ROC performance for all these methods). When positive and negative samples are severely unbalanced, oversampling was used to maintain an appropriate ratio in the training set.

Evaluation metrics

We report two standard evaluation measures commonly used in classification or prediction tasks. The first metric is the 'accuracy', which is used to measure the correct proportion of classification. It is computed as

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP, TN, FP and FN denote the true positive, true negative, false positive and false negative value, respectively. The second metric is the area under the curve score of receiver operating characteristic curve (ROC-AUC) [53], which is extracted based on the validation and test sets (larger is better). We used 5-fold cross-validation for the evaluation on all these datasets.

Baselines

We compared our method with the following solutions (including the one presented in the preliminary version [22]):

1. ***Basic Models***: these models include LR (logistic regression) [65], RF [66], SVM [67], DT (decision tree) [68] and KNN (K nearest neighbor) [69].
2. ***ECFP***: extended-connectivity fingerprints, known as ECFPs [9], are a class of topological fingerprints for molecular characterization. Here, ECFP_4 was adopted in the experiments.

Table 2. The detailed description of selected datasets

Dataset	Category	Data type	Tasks	Task type	Compound	Metric
BACE	Biophysics	SMILES	1	Binary classification	1522	ROC-AUC
BBBP	Physiology	SMILES	1	Binary classification	2053	ROC-AUC
HIV	Biophysics	SMILES	1	Binary classification	41 913	ROC-AUC
Tox21	Physiology	SMILES	12	Binary classification	8014	ROC-AUC
SIDER	Physiology	SMILES	27	Binary classification	1427	ROC-AUC
ToxCast	Physiology	SMILES	617	Binary classification	8615	ROC-AUC
ClinTox	Physiology	SMILES	2	Binary classification	1491	ROC-AUC

Table 3. ROC-AUC scores of various approaches in BACE, BBBP and HIV datasets

ModelDataset	BACE		BBBP		HIV							
	Random		Stratified		Random		Stratified		Random		Stratified	
	Test	Validate	Test	Validate	Test	VValidate	Test	Validate	Test	Validate	Test	Validate
LR	0.6888	0.6040	0.6412	0.6603	0.7374	0.6819	0.7277	0.8004	0.5000	0.5000	0.5000	0.4999
RF (n=10)	0.7662	0.7634	0.7488	0.7521	0.8106	0.7141	0.7358	0.8074	0.5650	0.5475	0.5524	0.5575
SVM	0.6045	0.5187	0.5279	0.5400	0.6700	0.5573	0.5874	0.6193	0.5000	0.5000	0.5000	0.5000
KNN	0.7598	0.7498	0.7603	0.7335	0.7456	0.8049	0.8142	0.6718	0.6023	0.5678	0.5736	0.5635
DT	0.6863	0.7272	0.7320	0.7416	0.7652	0.6973	0.7552	0.7692	0.5945	0.5839	0.5893	0.5894
Smi2Vec*+LSTM	0.8144	**0.8619**	0.7628	0.7330	0.8320	**0.8855**	0.8759	0.9356	0.7892	0.8103	0.8828	0.8820
Smi2Vec+BiGRU	**0.8440**	0.8584	**0.8539**	**0.8506**	**0.8886**	0.8584	**0.9457**	**0.9484**	**0.8955**	**0.9163**	**0.9117**	**0.9225**

Table 4. ROC-AUC scores of each task in Tox21. The results are based on the stratified split method

TaskModel	RF		SVM		Smi2Vec*-LSTM		**Smi2Vec-BiGRU**	
	Test	Validate	Test	Validate	Test	Validate	Test	Validate
NR-AR	0.6732	0.6730	0.4951	0.5098	0.6914	0.6909	**0.7114**	**0.7713**
NR-AR-LBD	0.6384	0.5825	0.5216	0.5208	0.7477	0.7228	**0.8243**	**0.8442**
NR-AhR	0.5980	0.6076	0.6396	0.6160	0.6780	0.6698	**0.8793**	**0.8751**
NR-Aromatase	0.5500	0.5798	0.5458	0.5486	0.4964	0.4991	**0.6985**	**0.8241**
NR-ER	0.5507	0.5433	0.5000	0.4992	0.6231	0.5546	**0.7360**	**0.7085**
NR-ER-LBD	0.5170	0.5931	0.5216	0.5436	0.5308	0.5256	**0.8675**	**0.7922**
NR-PPAR-gamma	0.5263	0.4984	0.5074	0.4944	0.5659	0.5000	**0.7494**	**0.7085**
SR-ARE	0.5568	0.5562	0.6355	0.5804	0.6414	0.5901	**0.7611**	**0.7945**
SR-ATAD5	0.5348	0.5356	0.4931	0.4982	0.5000	0.5171	**0.7632**	**0.7909**
SR-HSE	0.5124	0.5107	0.5161	0.4986	0.6120	0.6381	**0.7845**	**0.7540**
SR-MMP	0.6862	0.6809	0.6489	0.6596	0.7425	0.7438	**0.8599**	**0.8846**
SR-p53	0.5138	0.5310	0.4931	0.4982	0.5180	0.5149	**0.7321**	**0.7896**

3. ***graphconv***: it was proposed in [70]. This method implemented convolution with a spectral filter formed by linear Bspline interpolation. It is the first work to formulate an analogy of CNN on graph structure, instead of grid or others in Euclidean domains.
4. ***NFP***: neural fingerprint, known as NFP, which was proposed in [11]. It implemented convolutional nets that can take molecular graphs of arbitrary size as the input and were designed to be a drop-in replacement for Morgan or ECFP fingerprints.
5. ***GCN***: we referred to the graph CNN with K-localized spectral filter as GCN from [71], which makes great contributions to the development of graph-like structure neural networks.
6. ***AGCN***: a novel spectral graph convolution layer, named spectral graph convolution layer with graph Laplacian learning [72], which works with adaptive graphs. This method achieved the state-of-the-art results on many molecular datasets.
7. ***Smi2Vec*-LSTM***: the method presented in our preliminary version [22]. Here the notation $*$ means Smi2Vec* is different from the following Smi2Vec. Specifically, Smil2Vec* denotes a simple processing method that divides atoms into vectors without considering the positional and structural properties in molecules.
8. ***Smi2Vec-BiGRU***: this is the refined method presented in this article. Compared with 'Smi2Vec*-LSTM', the key differences are as follows: (i) it considers the structure information among candidate molecule; and (ii) it uses the BiGRU neural network, instead of LSTM.

Single- and multitask classification on molecular datasets

Single-task classification. For single-task classification (i.e. on the BACE, BBBP and HIV datasets), we trained several machine learning models as baselines on an identical single-task datasets. Also, for datasets used for testing these baselines, we used random and stratified splitting methods with an 8/1/1 ratio. The comparison results are shown in Table 3. It can be seen that our approach generally outperforms these compared methods on single-task datasets (i.e. BACE, BBBP and HIV). In particular, compared with the method 'Smi2Vec*-LSTM' presented in the preliminary version, on average our approach achieves the best results on these three datasets, demonstrating the competitiveness of our approach. In addition, we observe that, for these two data splitting methods (i.e. random and stratified), there is no significant impact on our approach. This indicates the stability of our approach.

Multi-task classification. In order to comprehensively analyze the performance of our approach for multi-task classification, we conducted experiments on each task in Tox21 and SIDER datasets. In this set of experiments, we mainly show the results of RF and SVM methods, since they presented better performance than other baselines for multiple task classification. Specifically, there are 12 and 27 tasks for Tox21 and SIDER, respectively. The compared results on Tox21 are shown in Table 4. It can be seen that on the whole our approaches including the previous method (i.e. 'Smi2Vec*-LSTM') show competitive performance on the Tox21 dataset. Particularly, on all tasks the improved method (i.e. Smi2Vec-BiGRU) achieves the

best performance, compared against the method presented in the preliminary version and also other competitors.

In addition, the results on the SIDER dataset are shown in Table 5. We can see that, on 25 out of 27 tasks, our approaches including 'Smi2Vec*-LSTM' achieve the best performance. Note that, on some tasks the Smi2Vec-GRU is better than Smi2Vec*-LSTM method, while it is inferior to Smi2Vec*-LSTM on other tasks. Nevertheless, on most of these tasks (about 80% of the whole tasks), the Smi2Vec-GRU obtains much more leading scores, demonstrating its competitiveness.

It is worth mentioning that, although the proposed method is much more complicate than classic methods (e.g. SVM and LR) and it generally outperforms these classical methods by about 0.1 to 0.2, this is a nontrivial performance improvement in drug discovery community. As will be shown later, other state-of-the-art methods are also much more complicate than classic methods, our method achieved similar or even better performance.

Comparison with graph-like structure methods

We also compared our approach with several state-of-the-art graph-like structure models proposed recently, including 'graphconv', 'GCN', 'NFP' and 'AGCN'; see Section 4.3 for their descriptions. Table 6 shows the comparison results. We observed that our approach can outperform the former several graph-like structured models including 'graphconv', 'NFP' and 'GCN' on these tested datasets. Particularly, compared with 'AGCN', the strongest graph-like structured model in the literature, our method nearly reached the same results on Tox21 and SIDER, and even improved the ROC-AUC score on ClinTox. Yet, we have to admit that the performance on ToxCast is inferior to AGCN. It could be that the ToxCast dataset includes qualitative results over 600 experiments on 8615 compounds; as for a binary classification task, it is difficult for Smi2Vec-GRU to construct an embedding matrix, due to the severe sparsity.

Table 5. ROC-AUC scores of every task in SIDER. The results are based on the stratified split method

Task	RF		SVM		Smi2Vec*-LSTM		**Smi2Vec-BiGRU**	
	Test	Validate	Test	Validate	Test	Validate	Test	Validate
Hepatobiliary disorders	0.5654	0.5696	0.5553	0.5733	0.5843	**0.6916**	**0.6504**	0.6640
Metabolism and nutrition disorders	0.5490	0.5337	0.5083	0.5091	0.5345	0.5382	**0.5998**	**0.5867**
Product issues	0.4906	0.5087	0.5032	0.5000	0.5048	**0.5120**	**0.5662**	0.4951
Eye disorders	0.5034	0.4984	0.5071	0.5179	0.5087	0.5260	**0.6044**	**0.6101**
Investigations	0.4877	0.5185	0.5014	0.5179	0.5045	0.4917	**0.6674**	**0.6316**
Musculoskeletal and connective tissue disorders	0.5180	0.5234	0.5116	0.5111	**0.5620**	0.5355	0.5533	**0.6108**
Gastrointestinal disorders	0.5551	0.4962	0.4926	0.5385	0.5564	0.5686	**0.6629**	**0.6295**
Social circumstances	0.4828	0.5185	0.4918	0.4958	0.5170	0.5204	**0.6089**	**0.5885**
Immune system disorders	0.5558	0.5271	0.5024	0.4950	0.5248	0.5465	**0.5770**	**0.6040**
Reproductive system and breast disorders	0.5436	0.5863	0.5601	0.6101	0.5956	0.5689	**0.6061**	**0.6265**
Neoplasms benign, malignant and unspecified	0.5363	**0.5744**	0.5388	0.5676	0.5396	0.5073	**0.5818**	0.5361
General disorders and administration site conditions	0.4922	0.4962	0.4826	0.4955	0.4929	0.5061	**0.5884**	**0.5750**
Endocrine disorders	0.5245	0.5015	0.4920	0.4959	0.5323	0.5301	**0.5873**	**0.6222**
Surgical and medical procedures	0.5228	0.5609	0.4955	0.4858	0.4960	0.5000	**0.5642**	**0.5699**
Vascular disorders	0.4976	0.4981	0.5036	0.5129	0.5011	0.5132	**0.5303**	**0.6090**
Blood and lymphatic system disorders	0.5373	0.5248	0.5036	0.5833	0.5408	0.5869	**0.6000**	**0.6933**
Skin and subcutaneous tissue disorders	0.5417	0.5335	0.4959	0.4915	0.5642	0.5407	**0.6683**	**0.6367**
Congenital, familial and genetic disorders	0.4713	0.5015	0.4971	0.4950	0.5000	0.5059	**0.6084**	**0.5860**
Infections and infestations	0.5139	0.5005	0.4989	0.5135	0.5148	0.5224	**0.6621**	**0.6040**
Respiratory, thoracic and mediastinal disorders	0.4893	0.4952	0.5122	0.5238	0.4954	0.5335	**0.6250**	**0.5533**
Psychiatric disorders	0.5282	0.5126	0.5073	0.5070	0.5378	0.5168	**0.5861**	**0.5983**
Renal and urinary disorders	0.5632	0.5514	0.5137	0.5234	0.5767	**0.6394**	**0.6173**	0.5938
Pregnancy, puerperium and perinatal conditions	0.4769	0.4885	0.4962	0.5288	0.4961	0.4885	**0.5164**	**0.5386**
Ear and labyrinth disorders	0.5617	0.5781	0.5000	0.4938	0.5012	0.4851	**0.6395**	**0.6158**
Cardiac disorders	0.5530	0.5871	0.4899	0.5000	0.5734	0.5104	**0.5918**	**0.6565**
Nervous system disorders	0.4890	0.5346	0.5417	0.5147	0.5147	0.5522	**0.7350**	**0.7079**
Injury, poisoning and procedural complications	0.5262	0.5333	0.4947	0.5145	**0.5315**	0.5546	**0.5943**	0.5461

Table 6. Task-averaged ROC-AUC scores on Tox21, ClinTox, SIDER and ToxCast datasets

Datasets	Tox21		ClinTox		SIDER		ToxCast	
	Validation	Testing	Validation	Testing	Validation	Testing	Validation	Testing
graphconv	0.7105	0.7023	0.7896	0.7069	0.5806	0.5642	0.6497	0.6496
NFP	0.7502	0.7341	0.7356	0.7469	0.6049	0.5525	0.6561	0.6384
GCN	0.7540	0.7481	0.8303	0.7573	0.6085	0.5914	0.6914	0.6739
AGCN	0.7947	**0.8016**	0.9267	0.8678	**0.6112**	0.5921	**0.7227**	**0.7033**
Ours	**0.7948**	0.7806	**0.9557**	**0.9779**	0.6033	**0.6071**	0.6621	0.6449

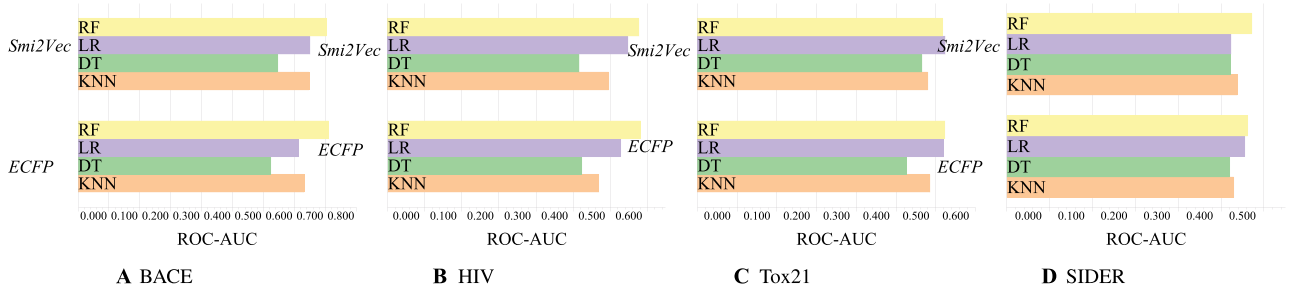

Figure 6. Performance analysis of 'Smi2Vec' and 'ECFP' for single and multiple classification tasks: 'BACE', four models are evaluated by ROC-AUC score on random split; 'HIV', four models are evaluated by ROC-AUC score on scaffold split; 'Tox21' and 'SIDER', four models are evaluated by averaged ROC-AUC score on random split. For ROC-AUC, higher value indicates better performance (to the right). Better viewed in color print.

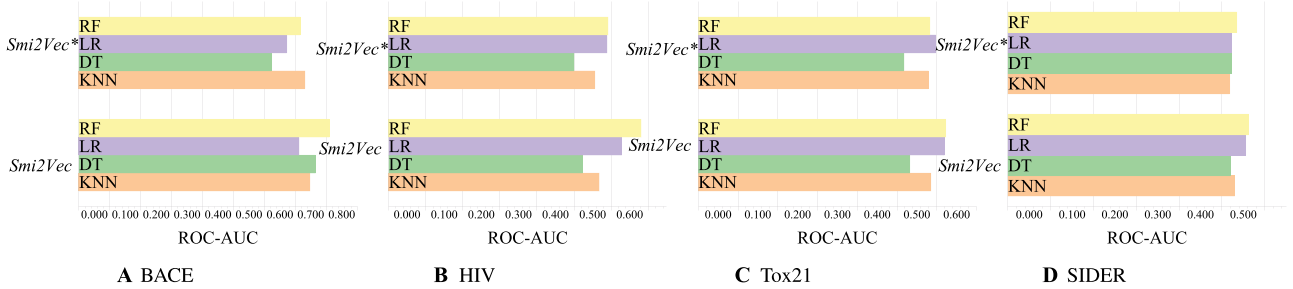

Figure 7. Performance analysis of 'Smi2Vec*' and 'Smi2Vec' for single and multiple classification tasks: 'BACE', four models are evaluated by ROC-AUC score on random split; 'HIV', four models are evaluated by ROC-AUC score on scaffold split; 'Tox21' and 'SIDER', four models are evaluated by averaged ROC-AUC score on random split. For ROC-AUC, higher value indicates better performance (to the right). Better viewed in color print.

Performance of Smi2Vec

To evaluate the performance of the proposed representation method, we compared Smi2Vec with 'ECFP' and also Smi2Vec* trained on the same conventional machine learning model; see Section 4.3 for the description of 'ECFP'. In the meanwhile, we tested different tasks including single- and multitask classification. The averaged ROC-AUC scores are shown in Figures 6 and 7. From these figures, we found that 'Smi2Vec' can nearly achieve the same performance on all these datasets, compared against 'ECFP'. This indicates that our approach, like ECFP, can also learn the structural relationship between molecules. As for the HIV dataset, it contains the single task for 41913 compounds; we used the scaffold splitting that separated structurally different molecules into different subsets. Intrinsically, the scaffold splitting is suitable for 'ECFP'. Interestingly, we observed that there is no apparent difference in the experimental results when 'Smi2Vec' was used. This essentially implies the good learning ability of our proposed method. In the meanwhile, compared with the Smi2Vec* method presented in the preliminary version, 'Smi2Vec' keeps almost the same performance, and particularly it obtained higher ROC-AUC scores for RF and LF on the HIV and SIDER datasets, respectively.

Summary

We found that Smi2Vec-GRU and the method presented in our preliminary version are feasible and competitive. Specially, (i) for the single-task classification, these two methods outperformed other classic methods such as KNN, DT and SVM on all the testing datasets; and the improved method, i.e. Smi2Vec-GUR, performed the best on most of cases. (ii) For the multi-task classification, the improved method Smi2Vec-GRU outperformed the best on all the tasks of the Tox21 dataset, compared against the classic methods and the method presented in our preliminary version; as for the SIDER dataset, Smi2Vec-GRU and the method presented in our preliminary version outperformed other classic methods on 26 out of 27 tasks; particularly, the improved method Smi2Vec performed the best on about 80% of the whole tasks. (iii) Our method outperformed most of state-of-the-art graph-like structure models such as 'graphconv', 'NFP' and 'GCN'. Furthermore, we found that the representation method, Smil2Vec, is feasible for capturing the fine-grained structural properties of molecule. Specifically, (i) it can nearly achieved the same performance on all these datasets, compared against the ECFP, which is a widely-used topological fingerprints for molecular characterization. (ii) Compared against the Smil2Vec* presented in our preliminary version, it kept almost the same performance, and particularly it obtained higher ROC-AUC scores for RF and LF on the HIV and SIDER datasets, respectively. On the other hand, we also realized the limitation of our method, i.e. Smi2Vec-GRU. Specifically, compared against the strongest graph-like structured model in the literature, i.e. AGCN, although our method nearly reached the same results on Tox21 and SIDER, and even improved the ROC-AUC score on ClinTox. However, it is obviously inferior to AGCN on the ToxCast dataset. This may imply that our method could be not suitable for the datasets that appear the severe sparsity.

Discussion and conclusion

In this paper, we have presented an approach, Smi2Vec-BiGRU, for learning atoms and solving the problem of single- and multi-task classification in the field of drug discovery. We have conducted extensive experiments based on several widely used molecule datasets. The experimental results demonstrated the feasibility and competitiveness of our proposed method. In the future, we may attempt to (i) further improve our approach by exploiting some other techniques such as attention mecha-

nisms, and (ii) solve other related problems in the fields of drug discovery and bioinformatics.

Key Points

- We present a novel approach named Smi2Vec-BiGRU that is designed for learning atoms and solving the single-task and multi-task binary classification problems in the field of drug discovery. Such problems are the basic and also key problems in this field, since many other tasks (e.g.,Drug-target interactions, Protein-protein interactions) significantly rely on the quality of the classification result.
- Our method leverages a powerful model, Bidirectional Gated Recurrent Unit (BiGRU) neural network, which is initially developed for solving problems in NLP and image processing, to train the sample vectors embedded in the atomic matrix.
- The experimental results show that, for the problem of single- and multi-task binary classification in the field of drug discovery, our proposed approach can achieve competitive performance, compared against classic and state-of-the-art graph-structured methods.

Acknowledgments

We thank the editors and anonymous reviewers very much for their insightful comments, which significantly improve the quality of the paper.

Funding

National Key R&D Program of China (2018YFB0204100); National Natural Science Foundation of China (U1811264, 61972425, 61602166, 61772183, U1401256, U1501252, U1611264, U1711261 and U1711262).

References

1. Chen HM, Engkvist O, Wang YH, *et al*. The rise of deep learning in drug discovery. *Drug Discov Today* 2018;**23**(6):1241–50.
2. Ding P, Yin R, Luo J, *et al*. Ensemble prediction of synergistic drug combinations incorporating biological, chemical, pharmacological, and network knowledge. *IEEE J Biomed Health Inform* 2018;**23**(3):1336–45.
3. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;**521**:436–44.
4. Duan MX, Li KL, Liao XK, *et al*. A parallel multiclassification algorithm for big data using an extreme learning machine. *IEEE Trans Neural Netw Learn Syst* 2017;**29**(6):2337–51.
5. Liu Y, Luo J, Ding P. Inferring microrna targets based on restricted Boltzmann machines. *IEEE J Biomed Health Inform* 2018;**23**(1):427–36.
6. Chen J, Li K, Bilal K, *et al*. Parallel protein community detection in large-scale ppi networks based on multi-source learning. *IEEE/ACM Trans Comput Biol Bioinform* 2018. doi: 10.1109/TCBB.2018.2868088.
7. Li C, Li K, Chen T, *et al*. SW-tandem: a highly efficient tool for large-scale peptide sequencing with parallel spectrum dot product on sunway taihulight. *Bioinformatics* 2019. doi: 10.1093/bioinformatics/btz147.
8. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988;**28**(1):31–6.
9. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model* 2010;**50**(5):742–54.
10. Montavon G, Hansen K, Fazli S, *et al*. Learning invariant representations of molecules for atomization energy prediction. In: *NIPS 2012, Advances in Neural Information Processing Systems*. 2012, 440–8. Lake Tahoe, Nevada, United States.
11. Duvenaud DK, Maclaurin D, Iparraguirre J, *et al*. Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems*. 2015, 2224–32. Montreal, Quebec, Canada.
12. Luo JW, Huang W, Cao BW. A novel approach to identify the miRNA–mRNA causal regulatory modules in cancer. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**15**(1):309–15.
13. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. 2012, 1097–105. Lake Tahoe, Nevada, United States.
14. Liu PF, Qiu XP, Chen XC, *et al*. Multi-timescale long short-term memory neural network for modelling sentences and documents. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, 2326–35. Lisbon, Portugal.
15. Chung JY, Gulcehre C, Cho K, *et al*. Gated feedback recurrent neural networks. In: *Proceedings of the 32nd International Conference on Machine Learning*, 2015, 2067–75. Lille, France.
16. Yao KS, Zweig G, Hwang MY, *et al*. Recurrent neural networks for language understanding. In: *Interspeech*. 2013, 2524–8. Lyon, France.
17. Quan Z, Wang Z-J, Le Y, *et al*. An efficient framework for sentence similarity modeling. *IEEE/ACM Trans Audio, Speech Language Process* 2019;**27**(4):853–65.
18. Duan MX, Li KL, Li KQ. An ensemble cnn2elm for age estimation. *IEEE Trans Inf Forensics Secur* 2017;**13**(3):758–72.
19. Fernández S, Graves A, Schmidhuber J. An application of recurrent neural networks to discriminative keyword spotting. In: *Artificial Neural Networks-ICANN 2007, International Conference*, September 9–13, 2007, 220–9. Porto, Portugal
20. Zhou Q, Tang PZ, Liu SX, *et al*. Learning atoms for materials discovery. *Proc Natl Acad Sci U S A* 2018;**115**(28):E6411–7.
21. Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. *International Conference on Learning Representations*. 2013, 1–12. Scottsdale, Arizona, USA.
22. Quan Z, Lin X, Wang ZJ, *et al*. A system for learning atoms based on long short-term memory recurrent neural networks. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2018, 728–33. Madrid, Spain.
23. Harel S, Radinsky K. Accelerating prototype-based drug discovery using conditional diversity networks. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, 331–9. London, United Kingdom.
24. Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, 1945–54. Sydney, NSW, Australia.
25. Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. In: *Proceedings*

of the 34th International Conference on Machine Learning. Vol. **80**, 2018, 1–17. Stockholmsmässan, Stockholm, Sweden.
26. Gómez-Bombarelli R, Wei J, Duvenaud D, *et al*. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent Sci* 2018;**4**(2):268–76.
27. Page D, Costa VS, Natarajan S, *et al*. Identifying adverse drug events by relational learning. In: *Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012, 790–3. Toronto, Ontario, Canada.
28. Yates A, Goharian N, Frieder O. Extracting adverse drug reactions from social media. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015, 2460–7. Austin, Texas, USA.
29. Zeng X, Zhu S, Liu X, *et al*. Deepdr: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics*. doi: 10.1093/bioinformatics/btz418.
30. Cheng F, Li W, Zhou Y, *et al*. Admetsar: a comprehensive source and free tool for assessment of chemical admet properties. *J Chem Inf Model* 2012;**52**:3099–105.
31. Xiao C, Zhang P, Chaovalitwongse W, *et al*. Adverse drug reaction prediction with symbolic latent dirichlet allocation. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017, 1590–6. San Francisco, California, USA.
32. Xiang YP, Liu K, Cheng XY, *et al*. Rapid assessment of adverse drug reactions by statistical solution of gene association network. *IEEE/ACM Trans Comput Biol Bioinform* 2015;**12**(4): 844–50.
33. Cheng F, Kovács I, Barabási A-L. Network-based prediction of drug combinations. *Nat Commun* 2019;**10**(1):1197.
34. Warmuth MK, Rätsch G, Mathieson M, *et al*. Active learning in the drug discovery process. In: *Advances in Neural Information Processing Systems*, 2002, 1449–56. Vancouver, British Columbia, Canada.
35. Ma TF, Xiao C, Zhou JY, *et al*. Drug similarity integration through attentive multi-view graph auto-encoders. In: *IJCAI 2018, International Joint Conference on Artificial Intelligence*. 2018, 3477–83. Stockholm, Sweden.
36. Ezzat A, Zhao PL, Wu M, *et al*. Drug–target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**14**(3):646–56.
37. Cheng F, Yu Y, Shen J, *et al*. Classification of cytochrome p450 inhibitors and noninhibitors using combined classifiers. *J Chem Inf Model* 2011;**51**(5):996–1011.
38. Yu L, Su RD, Wang B, *et al*. Prediction of novel drugs for hepatocellular carcinoma based on multi-source random walk. *IEEE/ACM Trans Comput Biol Bioinform* 2016;**14**(4): 966–77.
39. Khalid Z, Sezerman OU. Prediction of HIV drug resistance by combining sequence and structural properties. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**15**(3):966–73.
40. Ma JS, Sheridan RP, Liaw A, *et al*. Deep neural nets as a method for quantitative structure–activity relationships. *J Chem Inf Model* 2015;**55**(2):263–74.
41. Yu B, Yin HT, Zhu ZX. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 2018, 3634–40. Stockholm, Sweden.
42. Zhang MH, Cui ZC, Neumann M, *et al*. An end-to-end deep learning architecture for graph classification. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, February 2–7, 2018, 4438–45. New Orleans, Louisiana, USA.
43. Jin B, Yang HY, Xiao C, *et al*. Multitask dyadic prediction and its application in prediction of adverse drug–drug interaction. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017, 331–9. San Francisco, California, USA.
44. Kearnes S, McCloskey K, Berndl M, *et al*. Molecular graph convolutions: moving beyond fingerprints. *J Comput Aided Mol Des* 2016;**30**(8):595–608.
45. Lei T, Jin W, Barzilay R, *et al*. Deriving neural architectures from sequence and graph kernels. In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, 2024–33. Sydney, NSW, Australia.
46. Gilmer J, Schoenholz S, Riley P, *et al*. Neural message passing for quantum chemistry. In: *Proceedings of the 34th International Conference on Machine Learning*, 2017, 1263–72. Sydney, NSW, Australia.
47. Altae-Tran H, Ramsundar B, Pappu A, *et al*. Low data drug discovery with one-shot learning. *ACS Cent Sci* 2017;**3**(4): 283–93.
48. Landrum G, *et al*. *RDKit: open-source cheminformatics*, http://www.rdkit.org/.
49. Rodríguez P, Bautista MA, Gonzalez J, *et al*. Beyond one-hot encoding: lower dimensional target embedding. *Image Vis Comput* 2018;**75**:21–31.
50. Cho K, Van MB, Bahdanau D, *et al*. On the properties of neural machine translation: encoder–decoder approaches. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, October 25, 2014, 103–111. Doha, Qatar.
51. Chung J, Gulcehre C, Cho K, *et al*. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* 2014;**abs/1412.3555**:1–9.
52. Chakrabarty A, Pandit OA, Garain U. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, 1481–91. Vancouver, Canada.
53. Wu ZQ, Ramsundar B, Feinberg E, *et al*. Moleculenet: a benchmark for molecular machine learning. *Chem Sci* 2018; **9**(2):513–30.
54. Abadi M, Barham P, Chen JM, *et al*. Tensorflow: a system for large-scale machine learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016, 265–83. Savannah, GA, USA.
55. Kingma DP, Ba J. Adam: a method for stochastic optimization. *International Conference on Learning Representations*. 2015, 1–9. San Diego, CA, USA.
56. Subramanian G, Ramsundar B, Pande V, *et al*. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *J Chem Inf Model* 2016;**56**(10): 1936–49.
57. Martins I, Teixeira A, Pinheiro L, *et al*. A bayesian approach to in silico blood–brain barrier penetration modeling. *J Chem Inf Model* 2012;**52**(6):1686–97.
58. Zaharevd. *Aids Antiviral Screen Data*. https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data, (20 June 2018, date last accessed).
59. *Tox21 Challenge*. https://tripod.nih.gov/tox21/challenge/, (12 July 2018, date last accessed).
60. *Medical Dictionary for Regulatory Activities*. http://www.meddra.org/, (15 July 2018, date last accessed).
61. Gayvert K, Madhukar N, Elemento O. A data-driven approach to predicting successes and failures of clinical trials. *Cell Chem Biol* 2016;**23**(10):1294–301.
62. Richard A, Judson R, Houck K, *et al*. Toxcast chemical landscape: paving the road to 21st century toxicology. *Chem Res Toxicol* 2016;**29**(8):1225–51.

63. Novick P, Ortiz O, Poelman J, *et al*. Sweetlead: an *in silico* database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery. *PLoS One* 2013;**8**(11):e79568.
64. *Aggregate Analysis of clincaltrials.gov (AACT) Database*. https://www.ctti-clinicaltrials.org/aact-database, (28 June 2018, date last accessed).
65. Park H, Shiraishi Y, Imoto S, *et al*. A novel adaptive penalized logistic regression for uncovering biomarker associated with anti-cancer drug sensitivity. *IEEE/ACM Trans Comput Biol Bioinform* 2016;**14**(4):771–82.
66. Fabris F, Doherty A, Palmer D, *et al*. A new approach for interpreting random forest models and its application to the biology of ageing. *Bioinformatics* 2018;**34**(14):2449–56.
67. Bao Y, Hayashida M, Akutsu T. Lbsizecleav: improved support vector machine (svm)-based prediction of dicer cleavage sites using loop/bulge length. *BMC Bioinformatics* 2016;**17**(1):487.
68. Yamada KD, Omori S, Nishi H, *et al*. Identification of the sequence determinants of protein n-terminal acetylation through a decision tree approach. *BMC Bioinformatics* 2017;**18**(1):289.
69. Deng ZY, Zhu XS, Cheng DB, *et al*. Efficient knn classification algorithm for big data. *Neurocomputing* 2016;**195**:143–8.
70. Bruna J, Zaremba W, Szlam A, *et al*. Spectral networks and locally connected networks on graphs. In: *International Conference on Learning Representations, ICLR 2014*, Apr 14–16, 2014, Banff, Canada, 2014, 1–14.
71. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: *NIPS 2016, Advances in Neural Information Processing Systems*. 2016, 3844–52. Barcelona, Spain.
72. Li RY, Wang S, Zhu FY, *et al*. Adaptive graph convolutional neural networks. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, February 2–7, 2018, 3546–53. New Orleans, Louisiana, USA.