



# Capsule Networks Showed Excellent Performance in the Classification of hERG Blockers/Nonblockers

Yiwei Wang<sup>1,2†</sup>, Lei Huang<sup>3,4†</sup>, Siwen Jiang<sup>3</sup>, Yifei Wang<sup>1</sup>, Jun Zou<sup>1</sup>, Hongguang Fu<sup>3</sup> and Shengyong Yang<sup>1\*</sup>

<sup>1</sup> State Key Laboratory of Biotherapy and Cancer Center, West China Hospital, Sichuan University, Chengdu, China, <sup>2</sup> College of Preclinical Medicine, Southwest Medical University, Luzhou, China, <sup>3</sup> School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, <sup>4</sup> Basic Teaching Department, Sichuan College of Architectural Technology, Deyang, China

## OPEN ACCESS

### Edited by:

Jianfeng Pei,  
Peking University, China

### Reviewed by:

Mingyue Zheng,  
Chinese Academy of Sciences,  
China  
Youjun Xu,  
Peking University, China

### \*Correspondence:

Shengyong Yang  
yangsy@scu.edu.cn

<sup>†</sup>These authors have contributed  
equally to this work

### Specialty section:

This article was submitted to  
Translational Pharmacology,  
a section of the journal  
Frontiers in Pharmacology

**Received:** 27 September 2019

**Accepted:** 13 December 2019

**Published:** 28 January 2020

### Citation:

Wang Y, Huang L, Jiang S, Wang Y,  
Zou J, Fu H and Yang S (2020)  
Capsule Networks Showed Excellent  
Performance in the Classification of  
hERG Blockers/Nonblockers.  
*Front. Pharmacol.* 10:1631.  
doi: 10.3389/fphar.2019.01631

Capsule networks (CapsNets), a new class of deep neural network architectures proposed recently by Hinton et al., have shown a great performance in many fields, particularly in image recognition and natural language processing. However, CapsNets have not yet been applied to drug discovery-related studies. As the first attempt, we in this investigation adopted CapsNets to develop classification models of hERG blockers/nonblockers; drugs with hERG blockade activity are thought to have a potential risk of cardiotoxicity. Two capsule network architectures were established: convolution-capsule network (Conv-CapsNet) and restricted Boltzmann machine-capsule networks (RBM-CapsNet), in which convolution and a restricted Boltzmann machine (RBM) were used as feature extractors, respectively. Two prediction models of hERG blockers/nonblockers were then developed by Conv-CapsNet and RBM-CapsNet with the Doddareddy's training set composed of 2,389 compounds. The established models showed excellent performance in an independent test set comprising 255 compounds, with prediction accuracies of 91.8 and 92.2% for Conv-CapsNet and RBM-CapsNet models, respectively. Various comparisons were also made between our models and those developed by other machine learning methods including deep belief network (DBN), convolutional neural network (CNN), multilayer perceptron (MLP), support vector machine (SVM), k-nearest neighbors (kNN), logistic regression (LR), and LightGBM, and with different training sets. All the results showed that the models by Conv-CapsNet and RBM-CapsNet are among the best classification models. Overall, the excellent performance of capsule networks achieved in this investigation highlights their potential in drug discovery-related studies.

**Keywords:** deep learning, hERG, classification model, Capsule network, convolution-capsule network, restricted Boltzmann machine-capsule networks

## INTRODUCTION

The human ether-a-go-go-related gene (hERG) encodes a potassium channel protein, which is important for cardiac electrical activity and the coordination of heartbeat. Blockade of the hERG potassium channel can result in a potentially fatal disorder called long QT syndrome, as well as serious cardiotoxicity, which has led to the withdrawal of several marketed drugs and the failure of many drug research and development projects (Fermini and Fossa, 2003; Recanatini et al., 2005; Sanguinetti and Tristani-Firouzi, 2006; Bowes et al., 2012; Nachimuthu et al., 2012; Zhang et al., 2012; Shah, 2013; Kalyaanamoorthy and Barakat, 2018; Mladenka et al., 2018). Therefore, drug candidates that can bind with hERG should be eliminated as early as possible in drug discovery studies. At present, various *in vitro* experimental assays, such as fluorescent measurements (Dorn et al., 2005), radioligand binding assay (Yu et al., 2014), and patch-clamp electrophysiology (Stoelzle et al., 2011; Gillie et al., 2013; Danker and Moller, 2014), have been developed to measure the hERG binding affinity of chemicals. Nevertheless, these assays are often expensive and time-consuming, implying that they are not suitable for the evaluation of hERG binding affinity for a large number of chemicals in the early stage of drug discovery. Furthermore, the preconditions for the use of these analytical techniques are that the chemical compounds have been synthesized and are available in hand, which are usually not applicable in the era of virtual high-throughput screening. An alternative strategy is to use *in silico* methods; compared with experimental assays, *in silico* methods are cheaper and faster, and also do not involve any of the aforementioned preconditions.

To date, various *in silico* prediction models have been developed for hERG channel blockade. These models can be classified into structure-based and ligand-based models. Structure-based models utilize molecular docking to predict the binding mode and binding affinity of compounds to hERG. However, the structure-based methods often have some limitations such as protein flexibility, inaccurate scoring function, and solvent effect (Jia et al., 2008; Li et al., 2013). Ligand-based models can further be classified into several categories based on structural and functional features (Zolotoy et al., 2003; Aronov, 2005), quantitative structure-activity relationship (QSAR) models (Perry et al., 2006; Yoshida and Niwa, 2006; Tan et al., 2012), pharmacophore models (Cavalli et al., 2002; Aronov, 2006; Durdagi et al., 2011; Yamakawa et al., 2012; Kratz et al., 2014; Wang et al., 2016), and machine learning models (Wang et al., 2008; Klon, 2010; Wacker and Noskov, 2018). Compared with other models, machine learning models have attracted more attention in recent years due to the remarkable performance of machine learning methods in the handling of classification issues. For example, Wang et al. (2012) established binary classification models using Naïve Bayes (NB) classification and recursive partitioning (RP) methods, which achieved prediction accuracies of 85–89% in their test sets. Zhang and coworkers (Zhang et al., 2016) used five machine learning methods to develop models that can discriminate hERG

blockers from nonblockers, and they found that k-nearest neighbors (kNN) and support vector machine (SVM) methods showed a better performance than others. Broccatelli et al. (2012) derived several classification models of hERG blocker/nonblocker by using random forests (RF), SVM, and kNN algorithms with descriptor selections *via* genetic algorithm (GA) methods, and their prediction accuracies ranged from 83 to 86%. Didziapetris and Lanevskij (2016) employed a gradient-boosting machine (GBM) statistical technique to classify hERG blockers/nonblockers, and this offered overall prediction accuracies of 72–78% against different test sets. Very recently, Siramshetty et al. (2018) employed three methods (kNN, RF, and SVM) with different molecular descriptors, activity thresholds, and training set compositions to develop predictive models of hERG blockers/nonblockers, and their models showed better performance than previously reported ones.

There have been remarkable advances in deep learning methods since a fast learning algorithm for deep belief nets was proposed by Hinton in 2006 (Hinton et al., 2006a). They have widely been applied to fields particularly computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, and various games (Collobert and Weston, 2008; Bengio, 2009; Dahl et al., 2012; Hinton et al., 2012; LeCun et al., 2015; Defferrard et al., 2016; Mamoshina et al., 2016), where they have produced results comparable to or in some cases superior to human experts. In recent years, deep learning has also been applied to drug discovery, and it has demonstrated its potentials (Lusci et al., 2013; Ma et al., 2015; Xu et al., 2015; Aliper et al., 2016; Mayr et al., 2016; Pereira et al., 2016; Subramanian et al., 2016; Kadurin et al., 2017; Ragoza et al., 2017; Ramsundar et al., 2017; Xu et al., 2017; Ghasemi et al., 2018; Harel and Radinsky, 2018; Hu et al., 2018; Popova et al., 2018; Preuer et al., 2018; Russo et al., 2018; Segler et al., 2018; Shin et al., 2018; Cai et al., 2019; Wang et al., 2019a; Yang et al., 2019). However, there are still some issues that limit the application of deep learning in drug discovery. For example, deep learning usually requires a large number of samples for model training. Unfortunately, there are often a very limited number of agents (usually hundreds or thousands) in drug discovery-related studies due to high cost and the lengthy process involved in obtaining samples and their associated properties. In addition, commonly used deep learning algorithms or networks, such as convolutional neural network (CNN), are primarily designed for two-dimensional (2D) image recognition. In these networks, some special algorithms, such as the pooling algorithm in CNN, are adopted to reduce the dimensionality of the representation, which might lead to a loss of information.

To overcome the shortcomings of traditional deep learning networks, Hinton group (Sabour et al., 2017) proposed new deep learning architectures known as capsule networks (CapsNets), which introduced a novel building block that is used in deep learning to improve the model hierarchical relationships inside the internal knowledge representation of a neural network. CapsNets have shown great potential in some fields (Xi et al., 2017; Afshar et al., 2018; Lalonde and Bagci, 2018; Qiao et al.,

2018; Vesperini et al., 2018; Zhao et al., 2018; Wang et al., 2019b; Peng et al., 2019). However, CapsNets have not yet been applied to drug discovery-related studies. As the first attempt, in this study, we established two classification models of hERG blockers/nonblockers by using modified capsule network architectures. The models were evaluated using a test set and an external validation set, which are independent of the training set. Furthermore, our models were also compared with others.

The rest of this paper is organized as follows. The *Materials and Methods* section describes the implementation of the two capsule networks [convolution-capsule network (Conv-CapsNet) and RBM-CapsNet] developed in this study, as well as the data sets used and computational modeling details. The modeling, evaluation, and comparison with other models are presented in the *Results* section. The strengths of the capsule networks are analyzed in the *Discussion* section which is followed by a final summary.

## MATERIALS AND METHODS

### Convolution-Capsule Network Architecture

The architecture of Conv-CapsNet is schematically shown in **Figure 1**, which is similar in nature to that of the Hinton's original Capsule Network, except for one additional hidden feature layer. Apparently, Conv-CapsNet contains four layers: a convolutional layer, a hidden feature layer, a PrimaryCaps layer, and a DigitCaps layer. It is composed of 179 nodes for input, which are based on the feature vector size of the molecules. With mapping from the input vector, the hidden feature layer with 128 dimensional nodes was generated by one convolutional operation and one fully connected operation. The PrimaryCaps layer comprises eight capsules ( $u_i$ ), and each capsule in this layer includes eight-dimensional features. Furthermore, we computed

the contribution ( $\hat{u}_{j|i}$ ) of each capsule ( $u_i$ ) in PrimaryCaps to that ( $v_j$ ) in DigitCaps by using Eq. 1.

$$\hat{u}_{j|i} = W_{ij} \cdot u_i \quad (1)$$

The final layer (DigitCaps) has a two-dimensional capsule ( $v_j$ ) per digit class (two classes in this investigation). Each of these capsules received input from all the capsules in the PrimaryCaps layer through Eq. 2-1, Eq. 2-2, and Eq. 2-3.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (2-1)$$

$$s_j = \sum_k c_{kj} \hat{u}_{j|i} \quad (2-2)$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (2-3)$$

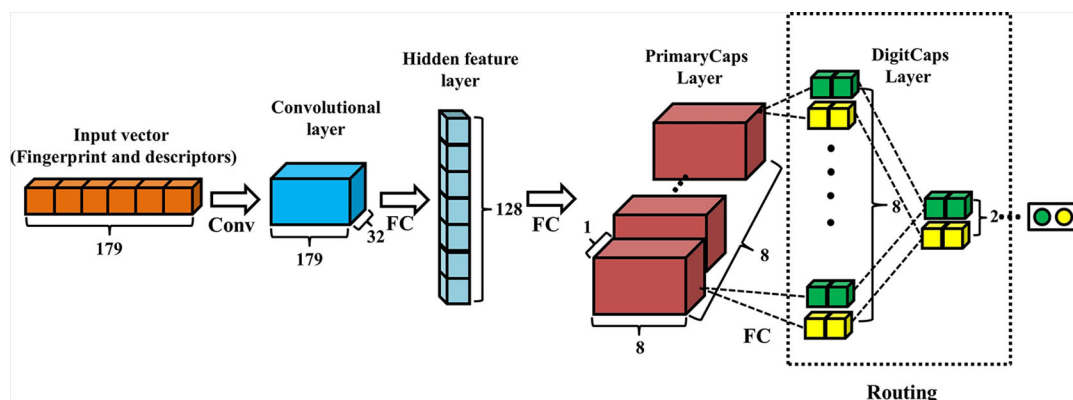
Finally, we computed the length of each digit capsule to predict the class of chemical molecules from Eq 3.

$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \times \max(0, \|v_k\| - m^-)^2 \quad (3)$$

In view of the small size of the dataset in this account, we added the L2 regularization behind the convolutional operation to prevent the network from overfitting (Ng, 2004).

### Hyperparameter Optimization

For the hyperparameter optimization of the Conv-CapsNet architecture, the different numbers of filters in the convolutional layer, nodes in the hidden feature layer, and dimensions in PrimaryCaps were explored. Additionally, the dynamic routing iterations between two capsule layers were tested from 1 to 3 with an increment of 1. For each group of the parameter settings, the performance of the model was evaluated by five-fold cross-validation based on the training



**FIGURE 1** | Architecture of convolution-capsule networks (Conv-CapsNet). The input is one-dimensional vector containing 179 components. The convolution layer has 32 filters of size 1×3. The hidden feature layer and PrimaryCaps layer consist of 128 and 64 nodes, respectively. The weight matrix between PrimaryCaps layer and DigitCaps layer is 8×8×2×2, and two dynamic routing iterations were adopted.

set. Once the highest accuracy was achieved with all the candidate settings, the best setting was subsequently applied to the test set and external validation set. We employed early stopping to reduce the overfitting problem, which is a technique commonly used for the reduction of overfitting (Caruana et al., 2001). With the early stopping, original training set was randomly divided into a new training set and a validation set (4:1). When the error in the validation set was less than that from the previous iteration, the training was immediately stopped. The final optimal hyperparameters for Conv-CapsNet are listed in **Table 1**.

### Model Training of Conv-CapsNet

The Conv-CapsNet weights were randomly initialized using a truncated normal distribution with the standard deviation being set as 0.01 during training. Both the convolutional and hidden feature layers adopted the rectified linear unit (Relu) as the activation function. To reduce the internal-covariate-shift, we used batch normalization to normalize the input distribution of each layer to a standard Gaussian distribution (Hinton et al., 2011; Ioffe and Szegedy, 2015). The adaptive moment estimation (Adam) method was employed for optimization (Kingma and Ba, 2014).

**Table 2** summarizes the algorithm and training procedure for Conv-CapsNet. *CW*, *W1*, and *W2* represent the parameters in the convolutional, hidden feature, and PrimaryCaps layers, respectively. The convolutional and the first two fully connected operations are represented by *conv*, *fc1*, and *fc2*, respectively; *conv\_layer*, *hf\_layer*, and *pc\_layer* denote the output from the convolutional, hidden feature, and PrimaryCaps layers, respectively. Through a feature vector extraction process in the convolutional layer, the hidden feature layer, and the PrimaryCaps layer (lines 1–4), *pc\_layer* was packed as capsules *u* (line 5). Here,  $\hat{u}$  denotes the contribution of one layer to the next layer. Next, the routing algorithm was used to generate the digit capsules (lines 6–13). *Len* is the length of the output of DigitCaps layer (lines 14). Lines 15–20 are for the network parameter update using a gradient step.

**TABLE 1** | Hyperparameter settings of convolution-capsule networks (Conv-CapsNet).

Hyperparameter	Setting
L2 normalization term	0.001
Activation	Relu
Batch size	148
Iteration epoch	300
Learning rate of network	0.001
Optimizer	Adam
Filter	32
Kernel_size	3
Number of nodes in the hidden feature layer	128
Number of nodes in the PrimaryCaps layer	64
Routing time	2
Dimension of each capsule	8
Length of PrimaryCaps	2
Length of DigitCaps	2

**TABLE 2** | Algorithm and training procedure of convolution-capsule networks (Conv-CapsNet).

**Algorithm:** Conv-CapsNet training algorithm, using a mini-batch stochastic gradient descent (SGD) for simplicity.  
**Input:** mini batch feature vector (**x**);  
 Number of Conv-CapsNet training epoch (**S**);  
 Number of dynamic routing iterations (**iter**).  
**Output:** Length of each capsules (**Len**).  
 1: **For** n=1 to **S** **do**  
 2: *conv\_layer* ← *conv*(*x*, *CW*)  
 3: *hf\_layer* ← *fc1*(*conv\_layer*, *W1*)  
 4: *pc\_layer* ← *fc2*(*hf\_layer*, *W2*)  
 5: *u* ← *Encapsule*(*pc\_layer*)  
 6: **For** all capsule *i* in PrimaryCaps layer:  $\hat{u}_{ji} \leftarrow W_{ij}u_i, \dots, \dots$  {contribution computes Eq. 1}  
 7: **For** all capsule *i* in PrimaryCaps layer and capsule *j* in DigitCaps layer:  $b_{ij} \leftarrow 0$   
 8: **For** *m*=1 to **iter** **do**  
 9: **For** all capsule *i* in PrimaryCaps layer:  $c_i \leftarrow \text{softmax}(b_i) \dots, \dots$  {softmax computes Eq. 2-1}  
 10: **For** all capsule *j* in DigitCaps layer:  $s_j \leftarrow \sum_i c_i \hat{u}_{ji} \dots$  {dynamic computes Eq. 2-2}  
 11: **For** all capsule *j* in DigitCaps layer:  $v_j \leftarrow \text{squash}(s_j) \dots, \dots$  {squash computes Eq. 2-3}  
 12: **For** all capsule *i* in PrimaryCaps layer and capsule *j* in DigitCaps layer:  $b_{ij} \leftarrow b_{ij} + \hat{u}_{ji} \cdot v_j$   
 13: **End for**  
 14: *Len* ← *Length of v*  
 15: *L* ← *loss of v* ..... {loss computes Eq. 3}  
 16: *W* ← *W* -  $\partial L / \partial W$   
 17: *CW* ← *CW* -  $\partial L / \partial CW$   
 18: *W1* ← *W1* -  $\partial L / \partial W1$   
 19: *W2* ← *W2* -  $\partial L / \partial W2$   
 20: **End for**

## Restricted Boltzmann Machine-Capsule Network Architecture

**Figure 2** displays the architecture of RBM-CapsNet, which consists of three layers: a hidden feature layer, a PrimaryCaps layer, and a DigitCaps layer. In RBM-CapsNet, two restricted Boltzmann machines (RBMs) replaced the convolutional and fully connected operations in Conv-CapsNet. The first RBM encodes the original vector (179-dimension) for the feature space (the hidden feature layer), which is subsequently used as the input for the next RBM. The RBMs used energy function (Eq. 4) as the loss function (Hinton and Salakhutdinov, 2006b).

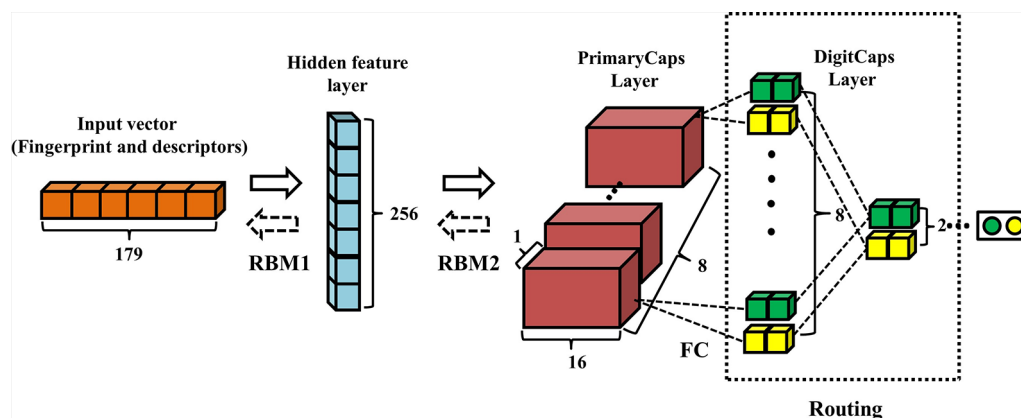
$$E(v, h) = -(a^T \cdot v + b^T \cdot h + v^T \cdot \omega \cdot h) \quad (4)$$

The capsule networks still consist of PrimaryCaps and DigitCaps, which are the same as in Conv-CapsNet. The detailed definitions of all the parameters in Eq. 1, 2, 3, and 4 are listed in the **Supplementary Material**.

### Hyperparameter Optimization

To optimize the hyperparameters in the RBM-CapsNet architecture, all the combinations of one to five RBM operations and 32, 64, 128, 256, and 512 nodes in each RBM were tested. The basic optimization procedure for the hyperparameters related to the capsules is very similar with that for Conv-CapsNet. The performance of each RBM-





**FIGURE 2 |** Architecture of restricted Boltzmann machine-capsule networks (RBM-CapsNet). The input is one-dimensional vector containing 179 components. The hidden feature layer and PrimaryCaps layer consist of 256 and 128 nodes, respectively. The weight matrix between PrimaryCaps layer and DigitCaps layer is  $8 \times 8 \times 2 \times 2$ , and two dynamic routing iterations were adopted.

CapsNet architecture was examined by five-fold cross-validation. The candidate RBM-CapsNet architecture that provided the highest accuracy was validated using the test set and external validation set. The detailed information on the optimized hyperparameters in RBM-CapsNet is summarized in **Table 3**.

### Model Training of Restricted Boltzmann Machine-Capsule Network

The training process was divided into two stages. First, two RBMs were pre-trained one by one with the loss function shown in Eq. 4. Second, the parameters of RBMs from pre-training were taken as initial values and the whole network was fine-tuned by back-propagation algorithm with end-to-end (Rumelhart et al., 1986).

**Table 4** summarizes the algorithm and training procedure for RBM-CapsNet.  $\theta_1$  and  $\theta_2$  represent the parameters of the hidden feature and PrimaryCaps layers, respectively.  $\phi_1$  and  $\phi_2$  represent the operations in RBM1 and RBM2, respectively. The *hf\_layer* and *pc\_layer* denote the output from the hidden feature and PrimaryCaps layers, respectively. After training RBM1 and

RBM2 individually (lines 1–6), the *pc\_layer* was packed as capsules  $u$  (line 10). The routing algorithm was then used to generate the digit capsules (lines 11–18). Len is the length of the output of DigitCaps layer (lines 19). Lines 20 to 24 are for a network parameter update using a gradient step ( $\partial L / \partial W$  represents the gradient of the contribution matrix, and  $\partial L / \partial \theta_1$  and  $\partial L / \partial \theta_2$  represent the gradients of the parameters for the hidden feature and PrimaryCaps layers, respectively).

### Data Sets

In this investigation, the Doddareddy's hERG blockade data set was used to establish our models (Doddareddy et al., 2010), which includes literature compounds tested on the hERG channel and Food and Drug Administration (FDA)-approved drugs. This data set contains a total of 2,644 compounds, including 1,112 positives (hERG blocker,  $IC_{50} < 10 \mu M$ ) and 1,532 negatives (hERG nonblocker,  $IC_{50} > 30 \mu M$ ). Doddareddy et al. partitioned this data set into a training set and a test set (Doddareddy et al., 2010). For comparison, the same partition scheme for the training and test sets as that by Doddareddy et al. was adopted in this investigation. Furthermore, we used Doddareddy's experimentally validated dataset (a total of 60 compounds: 50 agents from the Chembridge database and 10 from an in-house compound library) as an external validation set to assess the generalization ability of our models. In order to compare the performance of our models with others reported in the literature, we also used the same data sets as those in the literature, including Hou's (Wang et al., 2012; Wang et al., 2016), Zhang's (Zhang et al., 2016), Sun's (Sun et al., 2017), Siramshetty's (Siramshetty et al., 2018), and Cai's (Cai et al., 2019) data sets. Here, it is necessary to mention that an integrated data set of hERG blockade, which is the largest database to date, has been collected by Sato et al. (2018). However, we did not use this data set because it was not accessible. Another reason was that this data set has not been used to develop prediction models so far, and hence, a comparison study involving the data set was not feasible.

**TABLE 3 |** Hyperparameter settings of restricted Boltzmann machine-capsule networks (RBM-CapsNet).

Hyperparameter	Setting
Numbers of RBM	2
Number of nodes in the hidden feature layer	256
Number of nodes in the PrimaryCaps layer	128
Iteration of RBM	100
Iteration of network	200
Learning rate of RBM	0.001
Learning rate of network	0.005
Activation	Relu
Batch size	148
Optimizer	Adam
Routing time	2
Dimension of each capsule	8
Length of PrimaryCaps	2
Length of DigitCaps	2

**TABLE 4** | Algorithm and training procedure of restricted Boltzmann machine-capsule networks (RBM-CapsNet).

**Algorithm:** RBM-CapsNet training algorithm, using a mini-batch stochastic gradient descent (SGD) for simplicity.

**Input:** mini batch feature vector ( $\mathbf{x}$ );  
 Number of RBM training epoch (**S1**);  
 Number of Capsule training epoch (**S2**);  
 Number of dynamic routing iterations (**iter**).

**Output:** Length of each capsules (**Len**).

```

1: For n=1 to S1 do
2:   hf layer  $\leftarrow \phi_1(\mathbf{x}, \theta_1)$ .....{RBM1 training}
3: End for
4: For n=1 to S1 do
5:   pc layer  $\leftarrow \phi_2(\text{hf layer}, \theta_2)$ .....{RBM2 training}
6: End for
7: For n=1 to S2 do
8:   hf layer  $\leftarrow \phi_1(\mathbf{x}, \theta_1)$ 
9:   pc layer  $\leftarrow \phi_2(\text{hf layer}, \theta_2)$ 
10:  u  $\leftarrow$  Encapsule (pc layer)
11:  For all capsule i in PrimaryCaps layer:  $\hat{u}_{jji} \leftarrow W_{ij}u_i$ .....{contribution
  computes Eq. 1}
12:  For all capsule i in PrimaryCaps layer and capsule j in DigitCaps layer:  $b_{ij} \leftarrow$ 
13:  For m=1 to iter do
14:  For all capsule i in PrimaryCaps layer: .....{softmax computes Eq. 2-1}
15:  For all capsule j in DigitCaps layer:  $s_j \leftarrow \sum_i c_{ij} \hat{u}_{jji}$ .....{dynamic computes
  Eq. 2-2}
16:  For all capsule j in DigitCaps layer:  $v_j \leftarrow \text{squash}(s_j)$ .....{squash
  computes Eq. 2-3}
17:  For all capsule i in PrimaryCaps layer and capsule j in DigitCaps layer:  $b_{ij}$ 
 $\leftarrow b_{ij} + \hat{u}_{jji} \cdot v_j$ 
18: End for
19: Len  $\leftarrow$  Length of v
20: L  $\leftarrow$  loss of v.....{loss computes Eq. 3}
21: W  $\leftarrow W - \partial L / \partial W$ 
22:  $\theta_1 \leftarrow \theta_1 - \partial L / \partial \theta_1$ 
23:  $\theta_2 \leftarrow \theta_2 - \partial L / \partial \theta_2$ 
24: End for

```

## Molecular Characterization

In this investigation, a combination of MACCS (MDL Molecular Access) molecular fingerprints (166 bits) and 13 molecular descriptors was utilized to characterize the chemical compounds, which has been used by Zhang et al. and showed a good predictive performance in hERG blockade classification modeling (Zhang et al., 2016). Another reason why we adopted this characterization method (MACCS+13 descriptors, a total of 179 features) is because of their short length which is important for the reduction of the number of parameters in the modeling and the training time. By the way, the 13 molecular descriptors were selected because they are thought to be very related to ADMET properties and have been widely used in the modeling of various ADMET properties (Hou and Wang, 2008; Hou et al., 2009; Wang et al., 2012; Zhang et al., 2016). A detailed list of these descriptors are given as follows: the octanol-water partitioning coefficient, apparent partition coefficient at pH = 7.4, molecular solubility, molecular weight, number of hydrogen bond donors, number of hydrogen bond acceptors, number of rotatable bonds, number of rings, number of aromatic rings, sum of the oxygen and nitrogen atoms, polar surface area, molecular fractional polar surface area, and molecular surface area.

All the molecular fingerprints and molecular descriptors were computed with RDKit (Landrum, 2018) and PaDEL-Descriptor (Yap, 2011), respectively. Because the values of the different descriptors might span significantly different numerical ranges, their values were scaled to the same range (0, 1) by using the following formula:

$$x^* = \frac{x - \min}{\max - \min} \quad (5)$$

where  $x$  is the original value,  $x^*$  is the scaled value, and  $\max$  and  $\min$  are the maximum and minimum values of a descriptor, respectively.

## Model Assessment

All the models were assessed based on their accuracy (Q), sensitivity (SE), and specificity (SP). Q reflects the total prediction effect of a classifier. SE and SP represent the predictive power for positives and negatives, respectively. The definitions are given as follows (TP, true positive/blocker; TN, true negative/nonblocker; FP, false positive/blocker; and FN, false negative/nonblocker):

$$Q = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)$$

$$SE = \frac{TP}{TP + FN} \quad (7)$$

$$SP = \frac{TN}{FP + TN} \quad (8)$$

The classification capability of models was measured by area under the receive operating characteristic curve (AUC), which is an important indicator to illustrate the classification performance by changing its discrimination threshold.

Another measurement of the quality of binary (two-class) classifications is the Matthew's correlation coefficient (MCC). The MCC considers the balance ratios of the four confusion matrix categories (TP, TN, FP, and FN), and reflects the predictive power of models objectively without the influence of the disproportionate ratio of positives to negatives in the dataset. The MCC was calculated by using the following equation:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(FP + TN)(FP + TP)(FN + TN)(FN + TP)}} \quad (9)$$

## Computations

All the calculations were carried out with a single dual-processor, 16-core 2.1 GHz Intel® Xeon® E5-2683 v4 CPU with 126 GB memory and two NVIDIA Tesla P100 GPU accelerators. The software modules that were used to implement this project included Scikit-learn 0.18.1, Python 3.6.4, Anaconda 5.1.0 (64-bit), and TensorFlow 1.4.0.

## RESULTS

### Selection of the Optimal Capsule Network Architectures and Model Development

Hinton et al raised the concept of capsule network and proposed the first capsule network architecture prototype (Sabour et al., 2017). To

find the optimal capsule network architectures for the modeling of hERG blockade, we tried to construct a number of capsule networks with different architectures following Hinton's principle. Here, the Doddareddy's training set (positives: 1,004; negatives: 1,385) was adopted to train all the models, and the five-fold cross-validation method was used to monitor the training processes. In the five-fold cross-validation, the training set was randomly divided into five subsets. Of the five subsets, four subsets were used as the training data, and the remaining subset was used as the validation data for testing the model. The cross-validation process was repeated five times, with each of the five subsets used exactly once as the validation data. The average of the results from the five runs was calculated to produce a single estimation. The five-fold cross-validation results for the training set are given in **Table 5**. According to these results, Conv-CapsNet and RBM-CapsNet showed the best performance. For the Conv-CapsNet model, the prediction accuracies for the hERG blockers (SE) and the hERG nonblockers (SP) were 88.6 and 89.1%, respectively, and the overall prediction accuracy (Q) was 88.9%. For the RBM-CapsNet model, the prediction accuracies for hERG blockers and nonblockers were 84.3 and 89%, respectively and the overall prediction accuracy was 87.0%. Importantly, the MCC values of Conv-CapsNet and RBM-CapsNet were 0.774 and 0.734, respectively, which were also the highest among all the MCC values (**Table 5**); a higher MCC value often indicates a better prediction power of model. Therefore, the architectures of Conv-CapsNet and RBM-CapsNet were chosen as our capsule network architectures, and a detailed description for these architectures was given in the *Materials and Methods* section.

## Validation of Our Models' Prediction Ability Against hERG Blockers/Nonblockers by Doddareddy's Test Set and External Validation Set

In the above subsection, we obtained the optimal architectures of capsule networks. With these capsule network architectures, two classification models of hERG blockers/nonblockers, Conv-

CapsNet and RBM-CapsNet models have been developed. To verify the predictive ability of these models, two test sets that are independent of the training set were used: Doddareddy's test set (positives: 108; negatives: 147) and external validation set (positives: 18; negatives: 42).

**Table 6** summarizes the prediction results of the Conv-CapsNet and RBM-CapsNet models. From **Table 6**, we can see that both models show excellent prediction ability to the Doddareddy's test set and external validation set. With the Conv-CapsNet model, of the 108 blockers in the test set, 102 were correctly predicted, indicating a prediction accuracy of 94.4% for the blockers (SE). For the 147 nonblockers, 132 (TN) were properly predicted. The accuracy for the prediction of nonblockers (SP) was 89.8%. Of all the 255 agents (blockers and nonblockers), 234 were correctly predicted and 20 were wrongly predicted (see **Table 6**). The overall prediction accuracy (Q) and AUC measure were 91.8% and 0.940 (see **Figure 3**), respectively. For the external validation set, of the 18 blockers, 16 (TP) were correctly discriminated from nonblockers. The prediction accuracy for the blockers (SE) was 88.9%. Of the 42 nonblockers, 30 (TN) were correctly predicted, indicating a prediction accuracy of 71.4% for the nonblockers (SP). Totally, 46 out of 60 compounds were correctly predicted. The overall prediction accuracy (Q) and AUC measure were 76.7% and 0.806, respectively. With the RBM-CapsNet model, in the test set, 99 (TP) out of 108 blockers were correctly predicted, indicating a prediction accuracy of 91.7%. Out of 147 nonblockers, 136 (TN) were correctly predicted, indicating a prediction accuracy of 92.5% for nonblockers. This model achieved an overall prediction accuracy of 92.2%. For the external validation set, the prediction accuracies for blockers (SE) and nonblockers (SP) were 94.4 and 71.4%, respectively.

**TABLE 5** | Prediction results of hERG blockers/nonblockers classification models developed by capsule networks with different architectures.

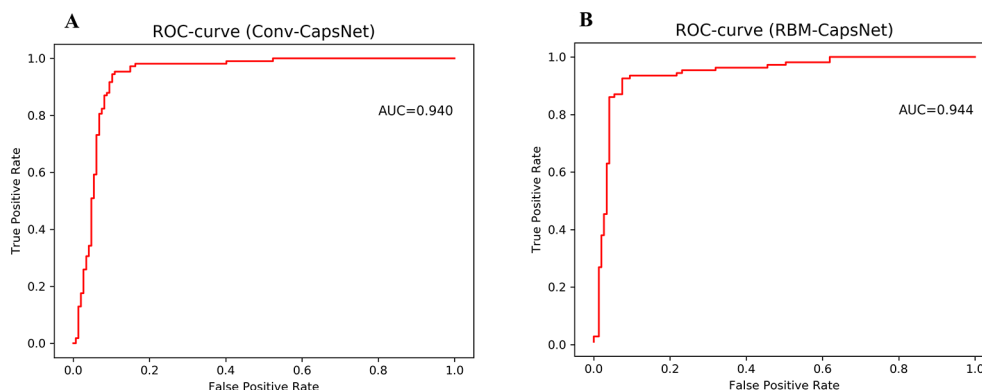
Capsule network architecture	SE	SP	MCC	SD	Q (%)
Original CapsNet	80.4%	86.7%	0.673	0.0141	84.1%
FC+FC	82.6%	86.7%	0.694	0.0195	85.0%
Conv+FC	82.2%	86.4%	0.687	0.0166	84.6%
Conv+FC+FC ( <b>Conv-CapsNet</b> )	<b>88.6%</b>	<b>89.1%</b>	<b>0.774</b>	<b>0.0109</b>	<b>88.9%</b>
Conv+Conv+FC+FC	84.5%	85.3%	0.693	0.0142	84.9%
Conv+Conv+Conv+FC+FC	81.9%	86.9%	0.685	0.0173	84.9%
One RBM	83.1%	86.5%	0.694	0.0182	84.9%
Two RBMs ( <b>RBM-CapsNet</b> )	<b>84.3%</b>	<b>89.0%</b>	<b>0.734</b>	<b>0.0160</b>	<b>87.0%</b>
Three RBMs	84.5%	85.5%	0.696	0.0160	85.0%
Four RBMs	81.2%	86.0%	0.673	0.0108	83.9%
Five RBMs	84.1%	86.4%	0.701	0.0156	85.4%

\*Conv, convolutional operation; FC, fully connected operation; RBM, restricted Boltzmann machine; Conv-CapsNet, convolution-capsule network; RBM-CapsNet, restricted Boltzmann machine-capsule network (The training set used was the Doddareddy's training set, and five-fold cross-validation was used to monitor the training performance. SE (%), sensitivity; SP (%), specificity; MCC, Matthew's correlation coefficient; SD, standard deviation; Q (%), overall accuracy). Conv-CapsNet and RBM-CapsNet showed the best performance.

**TABLE 6** | Prediction accuracies of hERG blockade classification models developed by different methods with the same Doddareddy's training set.

Model	SE	SP	MCC	Q (%)	AUC
Doddareddy's test set (255/P:108, N:147)					
Conv-CapsNet	94.4%	89.8%	0.835	91.8%	0.940
RBM-CapsNet	91.7%	92.5%	0.840	92.2%	0.944
CNN	87.0%	85.0%	0.715	85.9%	0.933
MLP	82.4%	86.4%	0.687	84.7%	0.920
DBN	72.2%	80.8%	0.533	80.8%	0.903
SVM	90.7%	84.4%	0.743	87.1%	0.933
kNN	69.4%	96.6%	0.703	85.1%	0.928
Logistic regression	88.8%	83.7%	0.710	85.5%	0.858
LightGBM	79.6%	82.3%	0.617	81.2%	0.810
Doddareddy's external validation (60/P:18, N:42)					
Conv-CapsNet	88.9%	71.4%	0.554	76.7%	0.806
RBM-CapsNet	94.4%	71.4%	0.604	78.7%	0.811
CNN	94.4%	52.4%	0.441	65.0%	0.725
MLP	88.9%	57.1%	0.426	66.7%	0.707
DBN	88.9%	52.4%	0.386	63.3%	0.683
SVM	88.9%	52.4%	0.386	63.3%	0.660
kNN	77.8%	52.4%	0.279	60.0%	0.624
Logistic regression	83.3%	52.4%	0.332	61.7%	0.623
LightGBM	61.1%	59.5%	0.190	60.0%	0.609

(TP, true positive; TN, true negative; FP, false positive; FN, false negative; SE (%), sensitivity,  $SE = TP / (TP + FN)$ ; SP (%), specificity,  $SP = TN / (TN + FP)$ ; Q (%), overall accuracy,  $Q = [TP + TN] / (TP + TN + FP + FN)$ ).



**FIGURE 3** | Receiver operating characteristic (ROC) curves for Doddareddy's test set by **(A)** convolution-capsule networks (Conv-CapsNet) and **(B)** restricted Boltzmann machine-capsule networks (RBM-CapsNet), respectively.

The overall prediction accuracy (Q) and the MCC values were 78.7% and 0.604, respectively. AUC for the test and external validation sets were 0.944 (see **Figure 3**) and 0.811, respectively. All of these results clearly demonstrate that the established Conv-CapsNet and RBM-CapsNet models can not only correctly classify the training set compounds but also have an outstanding predictability for external agents outside of the training set.

### Comparison of Our Models With Other Models Developed With the Same Doddareddy's Training Set

To compare the performance of our models with that of others, we adopted commonly used machine learning methods to develop prediction models of hERG blockers/nonblockers with the same Doddareddy's training set. These machine learning methods include deep belief network (DBN), CNN, multilayer perceptron (MLP), SVM, kNN, logistic regression (LR), and LightGBM. Hyperparameters for these methods were optimized by five-fold cross-validation in advance, and the optimal hyperparameter values are listed in **Tables S1–S4**, respectively. The prediction results to the Doddareddy's test set and external validation set are also given in **Table 6**. From **Table 6**, we can see that the prediction accuracies of the seven models are obviously lower than those of our Conv-CapsNet and RBM-CapsNet models.

### Comparison of Our Models With Other Models Developed With Training Sets Different From Doddareddy's Training Set

It has been well known that the performance of a prediction model is often strongly dependent on the training set used. Therefore, to make a more objective comparison, we collected various hERG blockade classification models developed with training sets different from Doddareddy's training set. With these training sets, we established a series of new prediction models by the Conv-CapsNet and RBM-CapsNet methods. To avoid a possible influence of molecular features, the same

molecular features used in the literature were used. **Table 7** summarizes the prediction accuracies of various models reported in the literature together with those of models by Conv-CapsNet and RBM-CapsNet.

Entry 1–3 of **Table 7** list models developed with Hou's training set 1 (positives: 283; negatives: 109), training set 2 (positives: 272; negatives: 120), and training set 3 (positive: 314; negative: 306), respectively. In Hou's training sets 1 and 2, a threshold of 40  $\mu\text{M}$  was used to distinguish hERG blockers and nonblockers (blockers:  $\text{IC}_{50} < 40 \mu\text{M}$ ; nonblockers:  $\text{IC}_{50} \geq 40 \mu\text{M}$ ). With training sets 1 and 2, Hou et al. established three models by RP, NB, and SVM methods, and the SVM models showed the best performance on their test sets. In Hou's training set 3, a threshold of 30  $\mu\text{M}$  was used to define hERG blockers and nonblockers. A Bayesian classification model developed by Hou et al. with Hou's training set 3 gave a prediction accuracy of 85% on their test set. With Hou's training sets 1–3, we also separately established models by Conv-CapsNet and RBM-CapsNet methods. As shown in **Table 7**, our models showed comparable or superior performance compared with Hou's models. Entry 4 in **Table 7** shows models established by Zhang's training set (positives: 717; negatives: 210), in which a threshold of 30  $\mu\text{M}$  was used to define hERG blockers and nonblockers. With the training set, Zhang et al. built two models by using SVM and kNN methods, which gave prediction accuracies of 83.5 and 82.2%, respectively, on their test set. Our models, developed by Conv-CapsNet and RBM-CapsNet, exhibited a better performance on the same test set (prediction accuracies: 84.5 and 85.2%, respectively). Entry 5 in **Table 7** displays models developed with Sun's training set, which is a big data set consisting of 3,024 agents (positives: 483; negatives: 2,541) with a threshold of 30  $\mu\text{M}$  for defining hERG blockers and nonblockers. With the training set, Siramshetty et al. established two models by using LibSVM and RF methods, and their prediction accuracies on the test set were 71.0 and 74.0%, respectively. Our models offered much higher prediction accuracies (Conv-CapsNet: 83.3%; RBM-CapsNet: 86.3%). Entry 6 in **Table 7** shows models built with Siramshetty's



**TABLE 7** | Prediction results of various hERG blockade classification models developed with training sets different from Doddareddy's training set.

Entry	Model	Training set	Test set	SE	SP	Q
1	RP (Wang et al., 2016)	Hou's training set 1 (P: 283; N: 109)	Hou's test set 1 (P: 129; N: 66)	79.8%	75.8%	78.5%
	NB (Wang et al., 2016)			82.2%	75.8%	80.0%
	SVM (Wang et al., 2016)			90.7%	65.2%	82.1%
	Conv-CapsNet			85.7%	78.8%	82.0%
	RBM-CapsNet			84.1%	80.3%	82.0%
2	RP (Wang et al., 2016)	Hou's training set 2 (P: 272; N: 120)	Hou's test set 2 (P: 140; N: 55)	80.0%	74.5%	78.5%
	NB (Wang et al., 2016)			81.4%	80.0%	81.0%
	SVM (Wang et al., 2016)			85.0%	74.5%	82.1%
	Conv-CapsNet			82.1%	81.8%	82.0%
	RBM-CapsNet			81.4%	83.6%	82.0%
3	Bayesian (Wang et al., 2012)	Hou's training set 3 (P: 314; N: 306)	Hou's test set 3 (P: 63; N: 57)	86.9%	83.1%	85.0%
	Conv-CapsNet			87.3%	86.0%	86.8%
	RBM-CapsNet			88.9%	84.2%	86.8%
4	SVM (Zhang et al., 2016)	Zhang's training set (P: 717; N: 210)	Zhang's test set (P: 188; N: 48)	95.8%	34.0%	83.5%
	kNN (Zhang et al., 2016)			92.6%	40.4%	82.2%
	Conv-CapsNet			88.8%	66.7%	84.5%
	RBM-CapsNet			90.4%	64.6%	85.2%
5	LibSVM (Siramshetty et al., 2018)	Sun's training set (P: 483; N: 2541)	Sun's test set (P: 53; N: 13)	68.0%	85.0%	71.0%
	RF (Siramshetty et al., 2018)			72.0%	85.0%	74.0%
	Conv-CapsNet			83.0%	84.6%	83.3%
	RBM-CapsNet			86.8%	84.6%	86.3%
6	LibSVM (Siramshetty et al., 2018)	Siramshetty's training set T3 (P: 1406; N: 1708)	Doddareddy's test set (P: 108; N: 147)	64.0%	89.0%	78.0%
	RF (Siramshetty et al., 2018)			68.0%	91.0%	81.0%
	Conv-CapsNet			85.2%	88.4%	87.1%
	RBM-CapsNet			83.3%	91.2%	87.8%

training set T3 which were extracted from the ChEMBL database. In this training set, agents with a binding affinity of less than 1  $\mu\text{M}$  were defined as hERG blockers, and those with a binding affinity of greater than 10  $\mu\text{M}$  were defined as hERG nonblockers. With the training set, Siramshetty et al. established two models by using LibSVM and RF methods, and their prediction accuracies on their test set were 78.0 and 81.0%, respectively. Our Conv-CapsNet and RBM-CapsNet models gave prediction accuracies of 87.1 and 87.8%, respectively, which are obviously higher than those of LibSVM and RF models. Very recently, Cai et al. developed a deep learning model, termed deepHERG, to predict hERG blockers with a large dataset containing 7,889 compounds (Cai et al., 2019). To make a comparison, we also used the same datasets to train and test hERG blocker prediction models. With the same validation set and evaluation method as those in Cai's work, our Conv-CapsNet (AUC = 0.974) and RBM-CapsNet (AUC = 0.978) showed a better performance than their deepHERG (AUC = 0.967) (see **Table S5**). Collectively, for different training sets given here, the models developed by Conv-CapsNet and RBM-CapsNet were among the best models established by various machine learning methods.

## DISCUSSION

Since the first capsule networks were proposed by Hinton's group in 2017 (Sabour et al., 2017), they have attracted considerable attention because of their performance. For example, despite the simple three-layer architecture of the original capsule networks, they have achieved state-of-the-art results with 0.25% test error on Mixed National Institute of Standards and Technology

database (MNIST) without data augmentation, which is better than the previous baseline of 0.39% (Sabour et al., 2017). The excellent performance of capsule networks is mainly due to the introduction of the capsules and dynamic routing algorithms. A capsule is a set of neurons that forms a vector. These vectors contain information including the magnitude/prevalence, spatial orientation, and other attributes of the extracted feature. In the capsule networks, capsules are "routed" to any capsule in the next layer *via* a dynamic routing algorithm, which takes into account the agreement between these capsule vectors, thus forming meaningful part-to-whole relationships not found in standard CNNs. In other words, capsule networks are capable of catching and holding more fine information than traditional deep neuron networks, one benefit of which is that the amount of input data can be significantly reduced.

Although CapsNets were just proposed very recently, they have already been successfully applied in many fields (Afshar et al., 2018; Kumar, 2018; Lalonde and Bagci, 2018; Li et al., 2018; Liu et al., 2018; Mobiny and Van Nguyen, 2018; Qiao et al., 2018; Zhao et al., 2018; Peng et al., 2019). Among these applications, majorities are related to image recognition. For example, Afshar et al. (2018) established a CapsNet for brain tumor classification by recognizing brain magnetic resonance imaging (MRI) images and proved that it could successfully overcome the defects of CNNs. Kumar (2018) proposed a novel method for traffic sign detection using a CapsNet that achieved outstanding performance, the input of which was traffic sign images. Li et al. (2018) built a CapsNet to recognize rice composites from unmanned aerial vehicle (UAV) images. This is understandable because CapsNets were originally developed to overcome the defects associated with image recognition in the traditional deep learning networks.

In image recognition, the input data is a two-dimensional array. In this two-dimensional array data, adjacent data points are often highly correlated. Small changes in any points generally do not affect image recognition in traditional deep learning methods. However, in issues related to drug discovery, such as the evaluation of ADMET properties (like the prediction of hERG blockers), one-dimensional vectors that describe small molecular structures and properties are usually used as the network input, for example, molecular fingerprints and descriptors. Generally, there is no direct logical relationship between the components in each vector for this kind of input. Importantly small changes in vector components might have a significant impact on the entire molecular structure and its associated properties. Nevertheless, these small changes in vector components are often overlooked in traditional deep learning methods. In addition, the relative positions of vector components are often critical though there is no direct logical relationship between them because a vector component represents a substructure or property. In this situation, capsule networks, which adopt vector neurons, are expected to have a better performance in handling this kind of issue (like the hERG blocker modeling) than other deep scalar neuron networks.

As expected, the two established capsule networks, Conv-CapsNet and RBM-CapsNet, showed excellent performance in the classification of hERG blockade. Although this is the first application of capsule networks in the classification of hERG blockers/nonblockers, the established models are still among the best classification models for hERG blockers/nonblockers. There can be no doubt that the use of capsules or vector neurons is one of the main reasons that contribute to the excellent performance of our models. Here each capsule represents a combination of substructures and/or properties. Analogy to the case in image recognition, the length of each capsule is the probability that the combination of substructures or properties exists in a molecule, and the orientation may represent the relative position of the combination of substructures in a compound. Obviously, our capsule networks can learn some combinations of substructures and/or properties that are important for the hERG blockers or nonblockers. Even so, we have to acknowledge that the prediction models of hERG blockers/nonblockers developed by the new capsule networks are still like a black box. Some

questions regarding the models are difficult to answer. For example, we can't exactly know what the combination of substructures and/or properties is, and which features are important to the model and which samples are hard to classify. Overall, the application of capsule networks in drug discovery is still in its infancy. Further improvement of capsule networks and applications in drug discovery are necessary in future studies.

## DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

## AUTHOR CONTRIBUTIONS

SY designed the study. LH designed the algorithms. YwW and SJ executed the experiment and performed the data analysis. YwW mainly wrote the manuscript. SY, JZ, YfW and HF revised the manuscript. All authors discussed and commented on the manuscript.

## FUNDING

This work was supported by the National Natural Science Foundation of China (61876034, 81573349, 81773633, 21772130, and 81930125), National Science and Technology Major Project (2018ZX09711003-003-006; 2018ZX09711002-014-002, 2018ZX09201018, 2019ZX09301-135 and 2018ZX09711002-011-019), and 1.3.5 Project for Disciplines of Excellence, West China Hospital, Sichuan University.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fphar.2019.01631/full#supplementary-material>

## REFERENCES

- Afshar, P., Mohammadi, A., and Plataniotis, K. N. (2018). "Brain tumor type classification via capsule networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)* (Athens: IEEE), 3129–3133. doi: 10.1109/icip.2018.8451379
- Aliper, A., Plis, S., Artemov, A., Ulloa, A., Mamoshina, P., and Zhavoronkov, A. (2016). Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Mol. Pharm.* 13, 2524–2530. doi: 10.1021/acs.molpharmaceut.6b00248
- Aronov, A. (2005). Predictive in silico modeling for hERG channel blockers. *Drug Discov. Today* 10, 149–155. doi: 10.1016/s1359-6446(04)03278-7
- Aronov, A. M. (2006). Common pharmacophores for uncharged human ether-a-go-go-related gene (hERG) blockers. *J. Chem. Inf. Model.* 49, 6917–6921. doi: 10.1021/jm0605000
- Bengio, Y. (2009). Learning deep architectures for AI. *Found. Trends Mach. Learn.* 2, 1–127. doi: 10.1561/2200000006
- Bowes, J., Brown, A. J., Hamon, J., Jarolimek, W., Sridhar, A., Waldron, G., et al. (2012). Reducing safety-related drug attrition: the use of in vitro pharmacological profiling. *Nat. Rev. Drug Discov.* 11, 909–922. doi: 10.1038/nrd3845
- Broccatelli, F., Mannhold, R., Moriconi, A., Giuli, S., and Carosati, E. (2012). QSAR modeling and data mining link Torsades de Pointes risk to the interplay of extent of metabolism, active transport, and HERG liability. *Mol. Pharm.* 9, 2290–2301. doi: 10.1021/mp300156r
- Cai, C., Guo, P., Zhou, Y., Zhou, J., Wang, Q., Zhang, F., et al. (2019). Deep learning-based prediction of drug-induced cardiotoxicity. *J. Chem. Inf. Model.* 59, 1073–1084. doi: 10.1021/acs.jcim.8b00769
- Caruana, R., Lawrence, S., and Giles, C. L. (2001). "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping". In *Proceedings of the 13th Conference on Neural Information Processing Systems (NIPS 2000)*, Denver, CO, USA. MIT Press, 402–408. doi: 10.5555/3008751.3008807
- Cavalli, A., Poluzzi, E., De Ponti, F., and Recanatini, M. (2002). Toward a pharmacophore for drugs inducing the long QT syndrome: insights from a

- CoMFA study of HERG K<sup>+</sup> channel blockers. *J. Med. Chem.* 45, 3844–3853. doi: 10.1021/jm0208875
- Collobert, R., and Weston, J. (2008). “A unified architecture for natural language processing: deep neural networks with multitask learning.” in *Proceedings of the 25th international conference on Machine learning (Helsinki: ACM)*, 160–167. doi: 10.1145/1390156.1390177
- Dahl, G. E., Dong, Y., Li, D., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* 20, 30–42. doi: 10.1109/taasl.2011.2134090
- Danker, T., and Moller, C. (2014). Early identification of hERG liability in drug discovery programs by automated patch clamp. *Front. Pharmacol.* 5, 203. doi: 10.3389/fphar.2014.00203
- Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). “Convolutional neural networks on graphs with fast localized spectral filtering.” In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain. 3844–3852. doi: 10.5555/3157382.3157527
- Didziapetris, R., and Lanevskij, K. (2016). Compilation and physicochemical classification analysis of a diverse hERG inhibition database. *J. Comput. Aided Mol. Des.* 30, 1175–1188. doi: 10.1007/s10822-016-9986-0
- Doddareddy, M. R., Klaasse, E. C., Shagufta, Ijzerman, A. P., and Bender, A. (2010). Prospective validation of a comprehensive in silico hERG model and its applications to commercial compound and drug databases. *Chem. Med. Chem.* 5, 716–729. doi: 10.1002/cmdc.201000024
- Dorn, A., Hermann, F., Ebnet, A., Bothmann, H., Trube, G., Christensen, K., et al. (2005). Evaluation of a high-throughput fluorescence assay method for HERG potassium channel inhibition. *J. Biomol. Screen* 10, 339–347. doi: 10.1177/1087057104272045
- Durdagi, S., Duff, H. J., and Noskov, S. Y. (2011). Combined receptor and ligand-based approach to the universal pharmacophore model development for studies of drug blockade to the hERG1 pore domain. *J. Chem. Inf. Model.* 51, 463–474. doi: 10.1021/ci100409y
- Fermini, B., and Fossa, A. A. (2003). The impact of drug-induced QT interval prolongation on drug discovery and development. *Nat. Rev. Drug. Discov.* 2, 439–447. doi: 10.1038/nrd1108
- Ghasemi, F., Mehridehnavi, A., Perez-Garrido, A., and Perez-Sanchez, H. (2018). Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. *Drug Discov. Today* 23, 1784–1790. doi: 10.1016/j.drudis.2018.06.016
- Gillie, D. J., Novick, S. J., Donovan, B. T., Payne, L. A., and Townsend, C. (2013). Development of a high-throughput electrophysiological assay for the human ether-a-go-go related potassium channel hERG. *J. Pharmacol. Toxicol. Methods* 67, 33–44. doi: 10.1016/j.vascn.2012.10.002
- Harel, S., and Radinsky, K. (2018). Prototype-based compound discovery using deep generative models. *Mol. Pharm.* 15, 4406–4416. doi: 10.1021/acs.molpharmaceut.8b00474
- Hinton, G. E., and Salakhutdinov, R. R. (2006b). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647
- Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006a). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527
- Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). “Transforming auto-encoders,” in *International Conference on Artificial Neural Networks (Berlin, Heidelberg: Springer)*, 44–51.
- Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* 29, 82–97. doi: 10.1109/MSP.2012.2205597
- Hou, T., and Wang, J. (2008). Structure-ADME relationship: still a long way to go? *Expert Opin. Drug Metab. Toxicol.* 4, 759–770. doi: 10.1517/17425255.4.6.759
- Hou, T., Li, Y., Zhang, W., and Wang, J. (2009). Recent developments of in silico predictions of intestinal absorption and oral bioavailability. *Comb. Chem. High T. Scr.* 12, 497–506. doi: 10.2174/138620709788489082
- Hu, Q., Feng, M., Lai, L., and Pei, J. (2018). Prediction of drug-likeness using deep autoencoder neural networks. *Front. Genet.* 9, 585. doi: 10.3389/fgene.2018.00585
- Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. doi: arXiv:1502.03167v3
- Jia, R., Yang, L. J., and Yang, S. Y. (2008). Binding energy contributions of the conserved bridging water molecules in CDK2-inhibitor complexes: a combined QM/MM study. *Chem. Phys. Lett.* 460, 300–305. doi: 10.1016/j.cplett.2008.06.002
- Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A., and Zhavoronkov, A. (2017). druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol. Pharm.* 14, 3098–3104. doi: 10.1021/acs.molpharmaceut.7b00346
- Kalyaanamoorthy, S., and Barakat, K. H. (2018). Development of safe drugs: the hERG challenge. *Med. Res. Rev.* 38, 525–555. doi: 10.1002/med.21445
- Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: arXiv:1412.6980v9
- Klon, A. E. (2010). Machine learning algorithms for the prediction of hERG and CYP450 binding in drug development. *Expert Opin. Drug Metab. Toxicol.* 6, 821–833. doi: 10.1517/17425255.2010.489550
- Kratz, J. M., Schuster, D., Edtbauer, M., Saxena, P., Mair, C. E., Kirchebner, J., et al. (2014). Experimentally validated HERG pharmacophore models as cardiotoxicity prediction tools. *J. Chem. Inf. Model.* 54, 2887–2901. doi: 10.1021/ci5001955
- Kumar, A. D. (2018). Novel deep learning model for traffic sign detection using capsule networks. *arXiv preprint arXiv:1805.04424*. doi: arXiv:1805.04424v1
- Lalonde, R., and Bagci, U. (2018). Capsules for object segmentation. *arXiv preprint arXiv:1804.04241*. doi: arXiv:1804.04241v1
- Landrum, G. (2018). Provided by GitHub and SourceForge. *RDKit: Open-Source Cheminformatics Software [Online]*. Available: <http://www.rdkit.org>.
- Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539
- Li, G. B., Yang, L. L., Wang, W. J., Li, L. L., and Yang, S. Y. (2013). ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein-ligand interactions. *J. Chem. Inf. Model.* 53, 592–600. doi: 10.1021/ci300493w
- Li, Y., Qian, M., Liu, P., Cai, Q., Li, X., Guo, J., et al. (2018). The recognition of rice images by UAV based on capsule network. *Cluster Comput.* 6, 1–10. doi: 10.1007/s10586-018-2482-7
- Liu, Y., Tang, J., Song, Y., and Dai, L. (2018). A capsule based approach for polyphonic sound event detection. *arXiv preprint arXiv:1807.07436*. doi: arXiv:1807.07436v2
- Lusci, A., Pollastri, G., and Baldi, P. (2013). Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* 53, 1563–1575. doi: 10.1021/ci400187y
- Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., and Svetnik, V. (2015). Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* 55, 263–274. doi: 10.1021/ci500747n
- Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Mol. Pharm.* 13, 1445–1454. doi: 10.1021/acs.molpharmaceut.5b00982
- Mayr, A., Klambauer, G., Unterthiner, T., and Hochreiter, S. (2016). DeepTox: toxicity prediction using deep Learning. *Front. Environ. Sci.* 3, 8. doi: 10.3389/fenvs.2015.00080
- Mladenka, P., Applova, L., Patocka, J., Costa, V. M., Remiao, F., Pourouva, J., et al. (2018). Comprehensive review of cardiovascular toxicity of drugs and related agents. *Med. Res. Rev.* 38, 1332–1403. doi: 10.1002/med.21476
- Mobiny, A., and Van Nguyen, H. (2018). “Fast capsNet for lung cancer screening,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention (Cham: Springer)*, 741–749.
- Nachimuthu, S., Assar, M. D., and Schussler, J. M. (2012). Drug-induced QT interval prolongation: mechanisms and clinical management. *Ther. Adv. Drug Saf.* 3, 241–253. doi: 10.1177/2042098612454283
- Ng, A. Y. (2004). “Feature selection, L 1 vs. L 2 regularization, and rotational invariance.” In *Proceedings of the 21th international conference on Machine learning (ACM)*, Banff. 78. doi: 10.1145/1015330.1015435
- Peng, C., Zheng, Y., and Huang, D. S. (2019). Capsule network-based modeling of multi-omics data for discovery of breast cancer-related genes. *IEEE/ACM Trans. Comput. Biol. Bioinform.* doi: 10.1109/TCBB.2019.2909905
- Pereira, J. C., Caffarena, E. R., and Dos Santos, C. N. (2016). Boosting docking-based virtual screening with deep learning. *J. Chem. Inf. Model.* 56, 2495–2506. doi: 10.1021/acs.jcim.6b00355
- Perry, M., Stansfeld, P. J., Leaney, J., Wood, C., De Groot, M. J., Leishman, D., et al. (2006). Drug binding interactions in the inner cavity of HERG channels:



- molecular insights from structure-activity relationships of clofilium and ibutilide analogs. *Mol. Pharm.* 69, 509–519. doi: 10.1124/mol.105.016741
- Popova, M., Isayev, O., and Tropsha, A. (2018). Deep reinforcement learning for de novo drug design. *Sci. Adv.* 4, eaap7885. doi: 10.1126/sciadv.aap7885
- Preuer, K., Lewis, R. P. I., Hochreiter, S., Bender, A., Bulusu, K. C., and Klambauer, G. (2018). DeepSynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics* 34, 1538–1546. doi: 10.1093/bioinformatics/btx806
- Qiao, K., Zhang, C., Wang, L., Yan, B., Chen, J., et al. (2018). Accurate reconstruction of image stimuli from human fMRI based on the decoding model with capsule network architecture. *arXiv preprint arXiv:1801.00602*. doi: arXiv:1801.00602v1
- Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., and Koes, D. R. (2017). Protein-ligand scoring with convolutional neural networks. *J. Chem. Inf. Model.* 57, 942–957. doi: 10.1021/acs.jcim.6b00740
- Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P., et al. (2017). Is multitask deep learning practical for pharma? *J. Chem. Inf. Model.* 57, 2068–2076. doi: 10.1021/acs.jcim.7b00146
- Recanatini, M., Poluzzi, E., Masetti, M., Cavalli, A., and De Ponti, F. (2005). QT prolongation through hERG K(+) channel blockade: current knowledge and strategies for the early prediction during drug development. *Med. Res. Rev.* 25, 133–166. doi: 10.1002/med.20019
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0
- Russo, D. P., Zorn, K. M., Clark, A. M., Zhu, H., and Ekins, S. (2018). Comparing multiple machine learning algorithms and metrics for estrogen receptor binding prediction. *Mol. Pharm.* 15, 4361–4370. doi: 10.1021/acs.molpharmaceut.8b00546
- Sabour, S., Frosst, N., and Hinton, G. E. (2017). “Dynamic Routing Between Capsules.” In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 3859–3869. doi: 10.5555/3294996.3295142.
- Sanguinetti, M. C., and Tristani-Firouzi, M. (2006). hERG potassium channels and cardiac arrhythmia. *Nature* 440, 463–469. doi: 10.1038/nature04710
- Sato, T., Yuki, H., Ogura, K., and Honma, T. (2018). Construction of an integrated database for hERG blocking small molecules. *PLoS One* 13, e0199348. doi: 10.1371/journal.pone.0199348
- Segler, M. H. S., Kogej, T., Tyrchan, C., and Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* 4, 120–131. doi: 10.1021/acscentsci.7b00512
- Shah, R. R. (2013). Drug-induced QT interval prolongation: does ethnicity of the thorough QT study population matter? *Br. J. Clin. Pharmacol.* 75, 347–358. doi: 10.1111/j.1365-2125.2012.04415.x
- Shin, M., Jang, D., Nam, H., Lee, K. H., and Lee, D. (2018). Predicting the absorption potential of chemical compounds through a deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 15, 432–440. doi: 10.1109/TCBB.2016.2535233
- Siramshetty, V. B., Chen, Q., Devarakonda, P., and Preissner, R. (2018). The Catch-22 of predicting hERG blockade using publicly accessible bioactivity data. *J. Chem. Inf. Model.* 58, 1224–1233. doi: 10.1021/acs.jcim.8b00150
- Stoelzle, S., Obergrussberger, A., Bruggemann, A., Haarmann, C., George, M., Kettchenhofen, R., et al. (2011). State-of-the-art automated patch clamp devices: heat activation, action potentials, and high Throughput in ion channel screening. *Front. Pharmacol.* 2, 76. doi: 10.3389/fphar.2011.00076
- Subramanian, G., Ramsundar, B., Pande, V., and Denny, R. A. (2016). Computational modeling of beta-secretase 1 (BACE-1) inhibitors using ligand based approaches. *J. Chem. Inf. Model.* 56, 1936–1949. doi: 10.1021/acs.jcim.6b00290
- Sun, H., Huang, R., Xia, M., Shahane, S., Southall, N., Wang, Y., et al. (2017). Prediction of hERG liability - Using SVM classification, bootstrapping and jackknifing. *Mol. Inform.* 36, 1600126. doi: 10.1002/minf.201600126
- Tan, Y., Chen, Y., You, Q., Sun, H., and Li, M. (2012). Predicting the potency of hERG K (+) channel inhibition by combining 3D-QSAR pharmacophore and 2D-QSAR models. *J. Mol. Model.* 18, 1023–1036. doi: 10.1007/s00894-011-1136-y
- Vesperini, F., Gabrielli, L., Principi, E., and Squartini, S. (2018). Polyphonic sound event detection by using capsule neural network. *arXiv preprint arXiv:1810.06325*. doi: arXiv:1810.06325
- Wacker, S., and Noskov, S. Y. (2018). Performance of machine learning algorithms for qualitative and quantitative prediction drug blockage of hERG1 channel. *Comput. Toxicol.* 6, 55–63. doi: 10.1016/j.comtox.2017.05.001
- Wang, M., Yang, X.-G., and Xue, Y. (2008). Identifying hERG potassium channel inhibitors by machine learning methods. *QSAR Comb. Sci.* 27, 1028–1035. doi: 10.1002/qsar.200810015
- Wang, S., Li, Y., Wang, J., Chen, L., Zhang, L., Yu, H., et al. (2012). ADMET evaluation in drug discovery. 12. Development of binary classification models for prediction of hERG potassium channel blockage. *Mol. Pharm.* 9, 996–1010. doi: 10.1021/mp300023x
- Wang, S., Sun, H., Liu, H., Li, D., Li, Y., and Hou, T. (2016). ADMET evaluation in drug discovery. 16. predicting hERG blockers by combining multiple pharmacophores and machine learning approaches. *Mol. Pharm.* 13, 2855–2866. doi: 10.1021/acs.molpharmaceut.6b00471
- Wang, D., Cui, C., Ding, X., Xiong, Z., Zheng, M., Luo, X., et al. (2019). Improving the virtual screening ability of target-specific scoring functions using deep learning methods. *Front. Pharmacol.* 10, 1–11. doi: 10.3389/fphar.2019.00924
- Wang, D., Liang, Y., and Xu, D. (2019). Capsule network for protein post-translational modification site prediction. *Bioinformatics* 35, 2386–2394. doi: 10.1093/bioinformatics/bty977
- Xi, E., Bing, S., and Jin, Y. (2017). Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*. doi: arXiv:1712.03480
- Xu, Y., Dai, Z., Chen, F., Gao, S., Pei, J., and Lai, L. (2015). Deep learning for drug-induced liver injury. *J. Chem. Inf. Model.* 55, 2085–2093. doi: 10.1021/acs.jcim.5b00238
- Xu, Y., Pei, J., and Lai, L. (2017). Deep learning based regression and multiclass models for acute oral toxicity prediction with automatic chemical feature extraction. *J. Chem. Inf. Model.* 57, 2672–2685. doi: 10.1021/acs.jcim.7b00244
- Yamakawa, Y., Furutani, K., Inanobe, A., Ohno, Y., and Kurachi, Y. (2012). Pharmacophore modeling for hERG channel facilitation. *Biochem. Biophys. Res. Commun.* 418, 161–166. doi: 10.1016/j.bbrc.2011.12.153
- Yang, X., Wang, Y., Byrne, R., Schneider, G., and Yang, S. (2019). Concepts of artificial intelligence for computer-assisted drug discovery. *Chem. Rev.* 119, 10520–10594. doi: 10.1021/acs.chemrev.8b00728
- Yap, C. W. (2011). PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. *J. Comput. Chem.* 32, 1466–1474. doi: 10.1002/jcc.21707
- Yoshida, K., and Niwa, T. (2006). Quantitative structure– activity relationship studies on inhibition of hERG potassium channels. *J. Chem. Inf. Model.* 46, 1371–1378. doi: 10.1021/ci050450g
- Yu, Z., Klaasse, E., Heitman, L. H., and Ijzerman, A. P. (2014). Allosteric modulators of the hERG K (+) channel: radioligand binding assays reveal allosteric characteristics of dofetilide analogs. *Toxicol. Appl. Pharmacol.* 274, 78–86. doi: 10.1016/j.taap.2013.10.024
- Zhang, W., Roederer, M. W., Chen, W.-Q., Fan, L., and Zhou, H.-H. (2012). Pharmacogenetics of drugs withdrawn from the market. *Pharmacogenomics* 13, 223–231. doi: 10.2217/pgs.11.137
- Zhang, C., Zhou, Y., Gu, S., Wu, Z., Wu, W., Liu, C., et al. (2016). In silico prediction of hERG potassium channel blockage by chemical category approaches. *Toxicol. Res. (Camb)* 5, 570–582. doi: 10.1039/c5tx00294j
- Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S., et al. (2018). Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*. doi: arXiv:1804.00538v4
- Zolotoy, A. B., Plouvier, B. P., Beatch, G. B., Hayes, E. S., Wall, R. A., Walker, M. J. A., et al. (2003). Physicochemical determinants for drug induced blockade of hERG potassium channels: effect of charge and charge shielding. *Curr. Med. Chem. Cardiovasc. Hematol. Agents* 1, 225–241. doi: 10.2174/1568016033477432

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Wang, Huang, Jiang, Wang, Zou, Fu and Yang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.