Quantitative Toxicity Prediction via Ensembling of Heterogeneous Predictors

Abdul Karim (abdul.karim@griffithuni.edu.au)
Griffith University https://orcid.org/0000-0002-4431-7507
Vahid Riahi
Griffith University
Avinash Mishra
Indian Institute of Technologies
Abdollah Dehzangi
Morgan State University
M. A. Hakim Newton
Griffith University
Abdul Sattar
Griffith University

Research article

Keywords: Deep Learning, Ensembling, Quantitative Toxicity

Posted Date: December 19th, 2019

DOI: https://doi.org/10.21203/rs.2.19338/v1

RESEARCH

Quantitative Toxicity Prediction via Ensembling of Heterogeneous Predictors

Abdul Karim[1,4*]
, Vahid Riahi[5]
, Avinash Mishra[2]
, Abdollah Dehzangi[3]
, MA Hakim Newton[4]
and Abdul Sattar[4]

*Correspondence:
abdul.karim@griffithuni.edu.au
[1]School of Information Communication Technology, Griffith University, Nathan, 4111 Brisbane, Australia
Full list of author information is available at the end of the article

Abstract

Background: Representing molecules in the form of only one type of features and using those features to predict their activities is one of the most important approaches for machine-learning-based chemical-activity-prediction. For molecular activities like quantitative toxicity prediction, the performance depends on the type of features extracted and the machine learning approach used. For such cases, using one type of features and machine learning model restricts the prediction performance to specific representation and model used.

Results: In this paper, we study quantitative toxicity prediction and propose a machine learning model for the same. Our model uses an ensemble of heterogeneous predictors instead of typically using homogeneous predictors. The predictors that we use vary either on the type of features used or on the deep learning architecture employed. Each of these predictors presumably has its own strengths and weaknesses in terms of toxicity prediction. Our motivation is to make a combined model that utilizes different types of features and architectures to obtain better collective performance that could go beyond the performance of each individual predictor. We use six predictors in our model and test the model on four standard quantitative toxicity benchmark datasets. Experimental results show that our model outperforms the state-of-the-art toxicity prediction models in 8 out of 12 accuracy measures.

Conclusion: Our experiments show that ensembling heterogeneous predictor improves the performance over single predictors and homogeneous ensembling of single predictors.The results show that each data representation or deep learning based predictor has its own strengths and weaknesses, thus employing a model ensembling multiple heterogeneous predictors could go beyond individual performance of each data representation or each predictor type.

Code Availability: Our implementation of the proposed model is freely available from our GitHub repository https://github.com/Abdulk084/HPE

Keywords: Deep Learning; Ensembling; Quantitative Toxicity

Background

Every year a great number of chemical compounds are produced. A large number of them are suspected to be toxic and many of them are eventually proved so. Toxicity is the degree to which a chemical compound can harm humans or animals. The

main metric employed to measure the toxicity of chemical compounds is the concentration of the compounds and the time of their exposure to the organism[1]. The concentration of compounds is measured by experiments known as *endpoints* measuring experiments. Toxicity endpoints mainly are either qualitative or quantitative. The qualitative endpoints categorize the chemical compounds in two groups: toxic and nontoxic. On the other hand, the quantitative endpoints record the minimal amount of chemical compounds that can reach given lethal effects. In this paper, we study quantitative toxicity prediction. Toxicity predictions, similar to the prediction of various characteristics of chemical compounds, are traditionally performed by *in-vivo* or *in-vitro* techniques. However, these techniques are very time-taking and cost-intensive. They also raise ethical concerns because of the involvement of animals. To address these issues, *in-silico* methods (computer-aided methods) have recently attracted much attention because they are efficient in time and cost and they do not compromise with accuracy much. There exist many *in-silico* methods, but the quantitative structure activity relationship (QSAR) method is one of the most successful ones. The main intuition behind the QSAR method is that chemicals molecules that are similar in structure should have similar activities and properties. Therefore, studying the relationships between chemical structures and biological activities of existing chemicals enables prediction of the activities and properties of new chemicals.

QSAR modelling using deep learning techniques have become very acceptable in recent years [2]. Most of these methods work on the features generated from the text format of the molecules. The text format is in a chemical language named the simplified molecular-input line-entry system (SMILES), which is used to describe the chemical structure of a molecule as a string of characters[3]. There is a special grammar for SMILES strings and different characters represent atoms or bonds among them. Such SMILES strings are utilized to obtain various types of numerical features (e.g. physicochemical descriptors) and molecular graphs by using different featurization methods[4, 5]. Traditional machine learning approaches such as K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest (RF), and Fully Connected Neural Networks (FCNN) are based on numerical features, particularly when used to predict activity or toxicity of a chemical compound[6]. Besides, numerical features, SMILES strings can also be used to generate molecular graphs or images, which then can be used in various types of convolutional neural network (CNN) to predict molecular activities[7]. Using CNN for molecular graphs or images needs relatively less domain expertise. It should be noted that SMILES strings can also be transformed into a vector representation or their respective fingerprints (fingerprints are bit strings composed of 0's and 1's) to be used in Recurrent Neural Networks (RNN) for molecular activity prediction [8].

Recently in the area of quantitative toxicity prediction, specialized type of features called element-specific topological descriptors (ESTDs) are used in deep neural networks and consensus models by TopTox to predict quantitative toxicity activity level[9]. Another recent work named AdmetSAR used molecular fingerprints to predict toxicity values by RF, SVM, and KNN models[10]. Yet there is another research with a name Hybrid2D, which used joint optimization of shallow neural networks and decision trees on 2D features only to predict toxicity measurement levels[11].

The performance of all these quantitative prediction methods is restricted by the specific type of features or model used in prediction.

In this paper, we propose a model comprising an ensemble of heterogeneous predictors (HPE). HPE uses six different deep learning methods, thus called predictors in the paper hereafter, to predict the regression values of four bench-mark quantitative toxicity data sets. These predictors are: (1) fully connected physico-chemical (FCPC) (2) fully connected physico-chemical extended (FCPCe) (3) convolution 1D SMILES (C1DS) (4) convolution 2D fingerprints (C2DF) (5) molecular graph convolution (MGC) and (6) molecular weave convolution (MWC). FCPC and FCFCe are fully connected neural networks, C1DS and C2DF are two types of convolutional neural networks, and MGC and MWC are two types of graph convolutional networks. In our HPE model, we ensembled the outputs of these predictors to achieve the overall performance. It should be noted that these predictors varies (heterogeneity) on either class, architecture or feature levels as shown in the Table 3. For instance, FCPC and FCPCe vary on feature level only. They both use numerical features (different in number only) but share the same architecture. C1DS and C2DF vary on the architecture and the feature level both. C1DS uses SMILES directly as input while C2DF converts SMILES into fingerprints first. MGC and MWC also vary on the architecture and the feature level. The details of these predictors are given in the methods section. Thus by introducing heterogeneity in each predictor with respect to the others, we were able to make a single model that utilizes different types of features and architectures to obtain collective performance that could go beyond individual performance of single predictor type. On four bench-mark quantitative toxicity based datasets, our proposed method obtains significantly better accuracy levels in 8 out of 12 metrices than that obtained by the state-of-the-art quantitative toxicity prediction methods. In terms of datasets, we outperform in IGC_{50}, LD_{50} and LC_{50}-DM than all other methods. Moreover, our experiments also show that HPE model significantly improves the performance over individual predictors and their homogeneous ensembling for all four quantitative toxicity datasets.

Results

We report the prediction results of the proposed HPE model. We have compared the proposed model with each of the single predictor (i.e., FCPC, FCPCe, C1DS, C2DF, MGC and MWC) used in our HPE model, and also with their homogeneous ensembles. The homogeneous ensembles (Hom) of each predictors are obtained by ensembling each individual predictor with itself six times. We also have compared the proposed model against known best-performing models in the literature: TopTox[9] and the methods used in the development of TEST software[12] which are based on hierarchical method nearest neighbor methods.

Comparison of HPE against Individual Predictors and their Homogeneous Ensembles

Table 1 presents the prediction results of individual predictors, their homogeneous ensembles (Hom), and our final model HPE in four datasets using three metrics. It should be noted that HPE is the ensemble of all six predictors. Comparing columns Ind and Hom in each dataset, in each metric, we see that each Hom obtains better

performance compared to the corresponding individual predictor (Ind). These are expected results and we include to reaffirm the strength of homogeneous ensembles. Our main results come from the ensembling heterogeneous predictors or HPE. Comparing columns Hom and HPE, we see that the HPE outperforms the homogeneous ensembles in all metrics in all datasets. The difference is in the range of 0.018–0.084 with an average of 0.03825 in a scale of 1.00. This clearly demonstrates the strength of the HPE over the homogeneous ensembles.

As can be seen from Table 1, in all four data sets, and in all three metrics, the proposed HPE model outperforms all six predictors. These results confirm that using a heterogeneous predictors ensembling (HPE) model using 6 different predictors is better than using just a single predictor. The results show that each data representation or neural network type has its own strengths and weaknesses, thus employing a model ensembling multiple predictors could go beyond individual performance of each data representation or each neural network type. For further clarification, the results are discussed below for each data set in detail.

- **IGC_{50}:** the proposed HPE obtained a correlation coefficient (R^2) of 0.831, RMSE of 0.426 $log(mol/L)$, and MAE of 0.182 $log(mol/L)$. However, among those 6 various individual predictors, MGC obtained the best R^2 value with of 0.782. FCPC obtained the best RMSE and MAE values with of 0.472 $log(mol/L)$ and 0.223 $log(mol/L)$, respectively. HPE improves the R^2 by 6.26% and 4.52%, RMSE by 9.74% and 7.59% , MAE by 9.03% and 8.73% from the best Ind and best Hom respectively.
- **LD_{50}:** the proposed HPE model obtains better results in all three metrics with R^2 of 0.680, RMSE of 0.536 $log(mol/L)$, and MAE of 0.407 $log(mol/L)$. HPE improves the R^2 by 7.59% and 4.61%, RMSE by 10.96% and 4.79% , MAE by 8.94% and 4.23% from the best Ind and best Hom respectively.
- **LC_{50}-DM :** as table shows, for this data set, the proposed HPE model obtains better results in all three metrics as well. It obtains R^2 of 0.811, RMSE of 0.787 $log(mol/L)$, and MAE of 0.620 $log(mol/L)$. HPE improves the R^2 by 8.13% and 6.29%, RMSE by 3.14% and 2.95% , MAE by 8.01% and 5.05% from the best Ind and best Hom respectively.
- **LC_{50}:** for this data set, the proposed HPE obtained R^2 of 0.742, RMSE of 0.788 $log(mol/L)$, and MAE of 0.621 $log(mol/L)$. HPE improves the R^2 by 7.53% and 4.50%, RMSE by 8.26% and 6.52% , MAE by 15.85% and 11.91% from the best Ind and best Hom respectively.

Evaluation of HPE model against several best-performing models

After finding the effectiveness of the proposed HPE model over various individual predictors and Hom, here we are to examine its performance against the state-of-the-art algorithms in the literature; the models used in the development of TEST software[12], TopTox[9] and Hybrid2d[11]. The results are shown in Table 2. As can be seen, from total 12 metrics, the proposed HPE model obtain the best results in 8 of them, especially in two of the data sets, it dominates other algorithms with obtaining better results in all three metrics. The detailed results are discussed below.

- **IGC_{50}:** As can be seen, for this data set, TEST consensus obtained the highest R^2 among different models in TEST software with of 0.764, while TopTox

model achieved R^2 of 0.802. However, the proposed model obtained R^2 of 0.831 which is better than all 6 models compared including TopTox. The proposed model also obtained better RMSE and MAE values with of 0.426 $log(mol/L)$ and 0.282 $log(mol/L)$, respectively.

- **LD_{50}:** for this data set, the proposed model dominates other algorithms in all three metrics with R^2 of 0.680, RMSE of 0.536 $log(mol/L)$, and MAE of 0.407 $log(mol/L)$. The results of TopTox model, in all three metrics, was better than TEST software models but worse than the proposed model in this paper.
- **LC_{50}-DM:** For R^2 and MRSE, the proposed model obtained 0.811 and 0.787 $log(mol/L)$ which was the better than all other models compared. However, for MAE, the proposed model obtained 0.620 $log(mol/L)$ which was better than all other models but TopTox with of 0.592 $log(mol/L)$.
- **LC_{50}:** As this table indicates, the proposed model obtained better R^2 results than 6 comparing models yet TopTox[9] with R^2 of 0.788. The TopTox[9] also obtained better results in terms of RMSE and MAE with of 0.677 $log(mol/L)$ and 0.446 $log(mol/L)$ respectively.

Discussion

Representing molecules in single type of representation and then using homogeneous modeling techniques might not help to capture the whole information about that molecule. For instance, basic molecular graph representation does not capture the quantum mechanical structure of molecules or necessarily express the information. Similarly the models which uses molecular graphs as input like graph convolution will not be able to distinguish between chiral molecules (molecules having same graph structure with a mirror image to each other). In case of fingerprints as an input, it is also possible that different molecules may have identical fingerprints which will make it difficult for a model to distinguish if it only takes fingerprints as input. There is also some information loss when one type of features are converted into another type of features.

In our experiments on the quantitative toxicity datasets, HPE obtains the highest performance followed by Hom and then individual predictors. The percentage improvement of HPE over Hom and Ind in all four datasets indicates that various predictors might be learning different knowledge from the same dataset. As it can be seen in Table 1, graph based predictors like MGC and MWC achieves better performances in most of the metrices and datasets. Specifically in maximizing R^2 for IGC_{50}, LD_{50}, and LC_{50}, MGC produces the best results whereas for LC_{50}-DM, MWC produces best results. The quantitative toxicity datasets considered in this study contain relatively smaller molecules which makes them more suitable for graph based predictors. The second highest performers on the average are FCPCe and FCPC which uses the features based on physico-chemical properties. These features have proved to have high predictive power in literature. It can be noticed that predictors like C1DS and C2DF struggle to perform as compared to other predictors. Yet, when all of them are ensembled to form an HPE model, they help in improving the results.

Even though various heterogeneous predictors ensembling enhance the overall accuracy. Yet it would be interesting to see the commonality between the learnt

representation of various individual predictors and to what degree one predictor's captured knowledge differ to the others.

Conclusion

Toxicity prediction methods of chemical compounds recently achieved enhanced performance in terms of accuracy after the introduction of various deep learning models in this space. Usually, molecules are represented in a fixed representation which are then used as features with a specific machine learning method to predict the toxicity. Among various other types of compounds toxicity, quantitative toxicity measurement has a paramount importance in pharmaceuticals. The performance of any quantitative toxicity prediction method depends upon the specific features and model used. This restricts the overall performance to single type of features and a model. In this paper, we propose a method which uses various heterogeneous predictors ensembling (HPE) to achieve better performance in quantitative toxicity prediction of four bench mark data-sets. Thus eliminating the restriction of model and data representation bound approach, each of our model's predictor vary either on features level, deep learning architecture level or both. These predictors include FCPC, FCPCe, C1DS, C2DF, MGC and MWC. FCPC and FCPCe vary on feature level only. They both use numerical features (different in number only) but share the same architecture. C1DS and C2DF vary on architecture and feature level both. C1DS uses SMILES directly as input while C2DF first converts SMILES into fingerprints. Molecular graph convolution (MGC) and molecular weave convolution (MWC) also vary on architecture and feature level both. Our motivation is to make a single model that utilizes different types of features and architectures to obtain collective performance that could go beyond individual performance of single predictor type. We also performed experiments which showed that heterogeneous ensembling method performs better than ensembling the homogeneous predictors. We achieved better performance in 8 out of 12 accuracy metrics for four quantitative toxicity data sets.

Abbreviations

Quantitative Structure Activity Relationship (QSAR), Simplified Molecular-input Line-entry System (SMILES), K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest (RF), Fully Connected Neural Networks (FCNN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), Element-Specific Topological Descriptors (ESTDs), Heterogeneous Predictors Ensembling (HPE), Fully Connected Physico-chemical (FCPC), Fully Connected Physico-chemical Extended (FCPCe), Convolution 1D SMILES (C1DS), Convolution 2D Fingerprints (C2DF), Molecular Graph Con- volution (MGC), Molecular Weave Convolution (MWC).

Methods

In this section, we first overview four data sets used in this paper. Then, we discuss the evaluation criteria for the individual predictors and their ensembles. Lastly, we present the implementation details of the individual predictors of our HPE model.

Data Sets

Mathematical representation of toxicity is the simplest way to understand the unwanted effect of a given compound on human health. These mathematical formulas for toxicity are based on two factors: (i) dose and (ii) time. These two factors combine and formulate quantitative toxicity of a compound. Quantitative toxicity, due to its mathematical characteristics, is not only easy to understand but is also proven to be compatible with prediction algorithms. This study considered four toxicity datasets. The datasets have different with endpoints indicators that shows a quantitative number to represent toxicity. LC_{50}, IGC_{50} and LD_{50} are the three different endpoints used for all four datasets. Here, two datasets (LC_{50} and LC_{50}-DM) share the same endpoint i.e. LC_{50} however they are tested with different animals. All these three measures are being used in toxicology for estimating the toxicity behaviour of any given chemical compound. LC_{50} and LD_{50} are the concentration and dose of the compound that kills half members of the tested population. LC_{50} is mostly used as an endpoint for testing the compounds on aquatic animals while LD_{50} is used for indicative toxicity on laboratory mice/rat. Here, LD_{50} depends on the route of administration: oral administration could cause less toxicity than intravenous route. IGC_{50} is the third measure that is also used on aquatic animals, but instead of showing the completely neutralizing effect, it shows the growth inhibition effect on tested population. In addition to concentration, these measures are also dependent on the duration of exposure of a given organism to the compound. Thus, each data set shows not only the type of endpoint but also the exposure time. Here, the first dataset shows the 96 hr LC_{50} record on fathead minnow, the second dataset shows 46 hr LC_{50}-DM record on *Daphnia magna*, the third dataset shows 40 hr IGC_{50} record on *Tetrahymena pyriformis* and the fourth dataset shows oral LD_{50} record on rat population. The concentration of compounds used in LC50 was in milligrams. Datasets were obtained from Wu et. al. work [9] while the original repository is available at http://cfpub.epa.gov/ecotox/ and http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp. These datasets have different sizes ranging from hundreds to thousands. For instance, LC_{50}-DM contains 353, LC_{50} contains 823, IGC_{50} contains 1792 and LD_{50} contains 7413 molecules.

Evaluation Criteria

In order to evaluate our predictors, we used K-fold cross validation with $K = 10$. Data was split into 10 equal random parts. One part was kept for testing while the 9 other parts were used for training. This process was repeated for 10 times. All the results shown later in the section represent an average value of 10 fold cross validation.

We have used three evaluation metrics for reporting the performance of our HPE and individual predictors. The first metric is mean absolute error (MAE) which is calculated as below. This metric calculates the average absolute error (i.e., differences between the prediction y_j and the actual observation $\hat{y}_j$) over the test data set. In this metric, all errors have equal weights.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{1}$$

The second metric is Root mean squared error (RMSE), which calculates the square root of the average of squared errors as follows. In this metric, the errors are squared, so large errors will have higher weights. This metric is more useful when large errors are undesirable.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{2}$$

Both of the above-mentioned metrics range from 0 to ∞, and the lower the MAE and $RMSE$ values, the better the model performance.

The third metric used in the paper is correlation coefficient R^2 which is calculated as below. In the equation below, $\bar{y}$ is the average of the actual observations. This metric explains the relationship between the prediction and the actual observation. It varies between 0 and 1, and the higher the value of R^2, the better the model's performance.

$$R^2 = 1 - \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)}{\sum_{j=1}^{n}(y_j - \bar{y}_j)} \tag{3}$$

Methods

In our HPE model, we ensembled six various deep learning based predictors to achieve the overall performance. It should be noted that these predictors vary (heterogeneity) on either class, architecture or feature levels as shown in the Table 3. We used an ensemble averaging method to combine the output of each individual predictor and to compute the final output of our model.

We refer the reader to [13] for the concepts and mathematics of deep learning and neural networks. In the rest of this section, we explain these predictors in terms of their classes, architectures and features. FCPC and FCPCe vary on feature levels only, C1DS and C2DF vary on architectures and feature levels both. MGC and MWC also vary on architectures and feature levels.

Fully Connected Physico-Chem (FCPC) and Fully Connected Physico-Chem_ext (FCPCe)

The first challenge in any machine learning algorithm is selecting a specific representation of the training data. The most common type of representation is numerical value based features. Usually for numerical features, a standard fully connected neural network is used. A neural network that has each unit of each layer connected to all the units of the next layer is termed as a *fully connected neural network* (FCNN). FCNN operates on a fixed shape input by passing information through multiple non-linear transformations. The first two predictors of our method (FCPC and FCPCe) use standard fully connected neural networks as shown in Figure 1 . FCNN in both FCPC and FCPCe predictors consist of 10 layers with 1000 neurons in each layer. The final layer consists of single unit with a linear function. Non linear activation function of sigmoid is used after each layer except the final layer. A dropout value of 0.5 is used after each layer. The learning rate was kept $5e^{-6}$ with a batch size of 32. Optimization was performed using the ADAM optimizer[14]. Both

of these predictors are built using a Keras deep learning framework on a system with Nvidia Tesla K40 GPU[15].

In FCPC component of our model, we used only 2D pysio-chemical features. These 2D pysio-chemical features are numerical in nature and are computed using Padel descriptor[4]. Out of total 1444 features, we computed 1148 features because Padel fails to compute features for large molecules due to time and memory constraint. For FCPCe component, we extended the feature set to 3D (total of 1826) using Modred descriptor [16]. It should be noted that 580 features are common between these two sets of physico-chemical features given in the supplementary as additional information.

Convolution 1D SMILES (C1DS) and Convolution 2D Fingerprint (C2DF)

A *convolutional neural network* (CNN) is a special type of neural network for the image data. CNNs can extract low level features from images and compute more complex features as we go deeper in the networks [17]. Variants of CNN like Inception, Alexnet and Resnet have been developed and employed as highly accurate image classification models [18]. 1D convolution is a special type of convolution which uses convolution operation over one dimension such as sequence or time series data as opposed to 2D convolution which works for 2 dimensional data such as images. It should be noted that there is another type of specialized neural network called recurrent neural network (RNN) which also works for sequential data but suffers from high computational cost as compared to 1D convolutional neural network [19].

We developed 1D convolutional neural network (C1DS) as a third predictor of our model. C1DS was trained directly on SMILES strings of the molecules. SMILES is a chemical language that describes the chemical structure of a molecule in a string of characters[3]. There is a special grammar for SMILES strings. Different characters represent atoms or bonds between the atoms. For instance, a small c represents aromatic carbon whereas capital C represents aliphatic carbon. To represent a single or double bond between atoms, special characters like = and - are used between the atom characters. An example of a SMILES string is COc(c1)cccc1C#N, which represents 3-cyanoanisole.

The architecture of CIDS predictor is shown in Figure 2a. The SMILES strings of molecules are of different lengths. We pad each smile with "0" and make them all equal to the length of longest smile in particular data set. The longest SMILES string is 52, 103, 75 and 181 for IGC50, LC50-DM, LC50 and LD50 respectively. Each character of the SMILES is encoded into a numerical value. Thus we obtain equal length vectors of each SMILE to be used in convolution 1D predictor. This fixed dimensional feature vector goes into the embedding layer of convolution 1D predictor. Each integer value of the fixed sized vector is embedded into 400 dimensional vector, thus creating a matrix of the shape *[maximum length of a SMILES string, 400]*. This matrix is trained along with the rest of the model training. After the embedding layer, we applied three 1D convolution layers, each with 192 filters with size of 10, 5 and 3 respectively. A ReLu activation function and batch normalization is used after each convolution layer. After flattening out, a fully connected dense layer with 100 units followed by a ReLu activation and dropout of 0.5 is

applied. The output layer is a single neuron with a linear activation function. It should be noted that learning rate, batch size and optimization algorithm are kept the same as of the FCPC and FCPCe components. We used Keras with NVidia Tesla K40 GPU for building convolution 1D SMILES predictor[15].

As described before, 2D convolutional neural network is a special type of neural network used for 2 dimensional data such as images. This predictor (C2DF) of our model (HPE) is based on 2D covolutional neural network inspired by FP2VEC model[20] as shown in Figure 2b. Each SMILES string of a molecule in all 4 data sets is first converted into their respective fingerprint. We used RDKit to convert the SMILES strings into 1024 bit Morgan fingerprints of a radius 2[21]. Fingerprints are bit strings composed of 0's and 1's. The position at which there is 1 represents a chemical feature defined by a specific design of fingerprint[22]. We computed a fingerprint indices vector by only taking those indices of the fingerprints with the value "1". The length of the fingerprint indices vector is computed to be 92 as the maximum number of "1s" in 1024 bit fingerprint of any molecule is 92 for all the four data sets under consideration. Those molecules with less than 92 "1s" in their 1024 bit fingerprint were padded with zero at the end. Thus we obtain fixed length vectors called fingerprint indices vector of each 1024 bit size fingerprint. This fixed length fingerprint indices vector goes into the embedding layer of C2DF predictor. Similar to C1DF, each integer value of the fingerprint index vector is embedded into 400 dimensional vector, thus creating a matrix of the shape *[92, 400]*. This matrix is trained along with the rest of the model training as similar to C1DS.

Unlike C1DS, in C2DF we used 2D convolution layer followed by maxpool layer. The output of the embedding layer in C2DF is fed into a 2D convolutional layer. The number of filters in this layer is chosen to be 2024 each with a size of *[4, 400]*. A maxpool layer with a kernal size of 89 followed by a dense layer with 100 units in it is applied. Rest of the hyper-parameters were kept same as that of C1DS. It should be noted that parameters like embedding size (which is chosen to be 400 for both C1DS and C2DF), filter/kernal sizes, number of filters, learning rate, batch size and optimizer type are chosen to be inspired from the previous published research[20, 11, 23, 24, 25] and initial experimentation.

Molecular Graph Convolution (MGC) and Molecular Weave Convolution (MWC)
Molecular Graph Convolution (MGC) and Molecular Weave Convolution (MWC) belong to the third category of our developed predictors. They use similar features and classes but different architectures as given in the Table 3. As the name suggests, Graph Convolution Networks (GCN) are inspired from the convolutional neural network by redefining them for graphs instead of typical pixel based images[26]. Typical neural networks like fully connected, recurrent and convolution neural networks extract latent representation from Euclidean space but they fail to work efficiently on graph data applications[27]. For instance, in the space of chemistry, a molecule can be represented in the form of a molecular graph, where the nodes represent the atoms and the bonds are represented by edges in the graph as shown in Figure3.

MGC and MWC are graph convolution neural networks trained on molecular graphs as input data. Conceptually, MGC only requires the structure or graph of a molecule and a vector of features for every atom (A) that describes the surrounding local chemical environment whereas MWC requires pair features (P) as well.

SMILES of each molecule is converted into their respective molecular graphs using RDKit[21]. Atom features such as atom type, chirality, formal charge, partial charge, ring sizes, hybridization, hydrogen bonding and aromaticity are computed using deepchem library [28, 29]. Pair features include bond type, graph distance, and same ring as described in previous papers [28, 29]. MGC predictor applies convolution layers to the central and its surrounding atoms, thus capturing the local chemical environment. As opposed to MGC, MWC predictor applies global convolutions to central atom along with all other atoms in a molecule while taking into account their corresponding atoms pair features as well. We used MGC and MWC as our two predictors for our HPE model from deepchem library with default settings[30]. The specific architecture details of both MGC and MWC can be found in the original molecular graph convolution paper by Google and deepchem open source library[28, 29, 30].

Declarations:

Competing interests

The authors declare that they have no competing interests.

Author's contributions

A.K. conceived and conducted the experiment(s). V.R., A.M., A.D., M.A.H.N., and A.S. analysed the results. All authors reviewed the manuscript. . . .

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Funding

This research is partially supported by Australian Research Council Discovery Grant DP180102727. The funding body did not play any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript. . .

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan XP GPU used for this research. . . .

Additional information

Supplementary

- **FCPC_Features_names.xlsx:** shows the 2D features names for FCPC predictor of our model computed using padel descriptor.
- **FCPCe_Features_names.xlsx:** shows the 2D+3D features names for FCPCe predictor of our model computed using Modred descriptor.
- **FCPC_FCPCe_Commen_Features.xlsx:** shows the commen features between FCPC and FCPCe

Availability of data and materials

- **Data:** data for this study can be obtained directly from Wu et. al. [9]. It can also be obtained from the original repository at http://cfpub.epa.gov/ecotox/ and http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp.
- **Code:** code for our model is available on our GitHub repository as follows.
  - **Name:** HPE (Heterogeneous Predictors Ensembling)
  - **Home page:** `https://github.com/Abdulk084/HPE`
  - **Operating system:** Ubuntu 18.04.3 LTS
  - **Programming language:** Python 3.7.3

Author details

[1]School of Information Communication Technology, Griffith University, Nathan, 4111 Brisbane, Australia. [2]Department of Chemical Engineering Indian Institute of Technology Hauz Khas New Delhi, India. [3]Department of Computer Science, Morgan State University Baltimore, USA. [4]Institute of Integrated and Intelligent Systems, Griffith University, Nathan, 4111 Brisbane, Australia. [5]Commonwealth Scientific and Industrial Research Organisation , Australia.

References

1. McFarland, J.W.: Parabolic relation between drug potency and hydrophobicity. Journal of medicinal chemistry **13**(6), 1192–1196 (1970)
2. Kato, Y., Hamada, S., Goto, H.: Molecular activity prediction using deep learning software library. In: 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), pp. 1–6 (2016). IEEE
3. Weininger, D.: Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. Journal of chemical information and computer sciences **28**(1), 31–36 (1988)
4. Yap, C.W.: Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. Journal of computational chemistry **32**(7), 1466–1474 (2011)
5. Ramsundar, B., Eastman, P., Walters, P., Pande, V.: Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More. " O'Reilly Media, Inc.", ??? (2019)
6. Lima, A.N., Philot, E.A., Trossini, G.H.G., Scott, L.P.B., Maltarollo, V.G., Honorio, K.M.: Use of machine learning approaches for novel drug discovery. Expert opinion on drug discovery **11**(3), 225–239 (2016)
7. Goh, G.B., Siegel, C., Vishnu, A., Hodas, N., Baker, N.: How much chemistry does a deep neural network need to know to make accurate predictions? In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1340–1349 (2018). IEEE
8. Goh, G.B., Hodas, N., Siegel, C., Vishnu, A.: Smiles2vec: Predicting chemical properties from text representations (2018)
9. Wu, K., Wei, G.-W.: Quantitative toxicity prediction using topology based multitask deep neural networks. Journal of chemical information and modeling **58**(2), 520–531 (2018)
10. Yang, H., Lou, C., Sun, L., Li, J., Cai, Y., Wang, Z., Li, W., Liu, G., Tang, Y.: admetsar 2.0: web-service for prediction and optimization of chemical admet properties. Bioinformatics **35**(6), 1067–1069 (2018)
11. Karim, A., Mishra, A., Newton, M.H., Sattar, A.: Efficient toxicity prediction via simple features using shallow neural networks and decision trees. ACS Omega **4**(1), 1874–1888 (2019)
12. Martin, T.: User's guide for test (version 4.2)(toxicity estimation software tool) a program to estimate toxicity from molecular structure. us epa office of research and development, washington, dc. Technical report, EPA/600/R-16/058 (2016)
13. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Chollet, F., et al.: Keras. https://keras.io (2015)
16. Moriwaki, H., Tian, Y.-S., Kawashita, N., Takagi, T.: Mordred: a molecular descriptor calculator. Journal of cheminformatics **10**(1), 4 (2018)
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
20. Jeon, W., Kim, D.: Fp2vec: a new molecular featurizer for learning molecular properties. Bioinformatics (2019)
21. Landrum, G.: RDKit: Open-source Cheminformatics. http://www.rdkit.org
22. Sánchez-Cruz, N., Medina-Franco, J.L.: Statistical-based database fingerprint: chemical space dependent representation of compound databases. Journal of cheminformatics **10**(1), 55 (2018)
23. Karim, A., Singh, J., Mishra, A., Dehzangi, A., Newton, M.H., Sattar, A.: Toxicity prediction by multimodal deep learning. In: Pacific Rim Knowledge Acquisition Workshop, pp. 142–152 (2019). Springer
24. Goh, G.B., Hodas, N.O., Siegel, C., Vishnu, A.: Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. arXiv preprint arXiv:1712.02034 (2017)
25. Goh, G.B., Siegel, C., Vishnu, A., Hodas, N.O., Baker, N.: Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. arXiv preprint arXiv:1706.06689 (2017)
26. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
27. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596 (2019)
28. Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., Pande, V.: Moleculenet: a benchmark for molecular machine learning. Chemical science **9**(2), 513–530 (2018)
29. Kearnes, S., McCloskey, K., Berndl, M., Pande, V., Riley, P.: Molecular graph convolutions: moving beyond fingerprints. Journal of computer-aided molecular design **30**(8), 595–608 (2016)
30. Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., Wu, Z.: Deep Learning for the Life Sciences. O'Reilly Media, ??? (2019). https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837

Figures

Tables

Table 1 Comparison of prediction results of individual predictors, their homogeneous ensembles, and our proposed HPE model on four datasets. In the table, columns Ind, Hom, and HPE are respectively for individual predictors, their homogeneous ensembles, and the heterogeneous ensemble. For each metric, the bold numbers are the best ones in the respective columns, and the underlined number is the best among all.

Metric	Predictor	IGC_{50} Dataset			LD_{50} Dataset			LC_{50}-DM Dataset			LC_{50} Dataset		
		Ind	Hom	HPE	Ind	Hom	HPE	Ind	Hom	HPE	Ind	Hom	HPE
R^2 maximise	FCPC	0.781	0.785		0.564	0.572		0.740	0.751		0.671	0.685	
	FCPCe	0.683	0.698		0.563	0.581		0.642	0.658		0.675	0.689	
	C1DS	0.699	0.715		0.538	0.539		0.702	0.713		0.646	0.653	
	C2DF	0.632	0.645		0.557	0.564		0.665	0.671		0.601	0.615	
	MGC	**0.782**	**0.795**		**0.632**	**0.650**		0.669	0.675		**0.690**	**0.710**	
	MWC	0.771	0.785		0.586	0.623		**0.750**	**0.763**		0.687	0.693	
				0.831			0.680			0.811			0.742
RMSE minimise	FCPC	**0.472**	0.471		0.621	0.610		0.864	0.850		0.874	0.860	
	FCPCe	0.564	0.550		0.617	0.604		1.085	1.055		0.872	0.859	
	C1DS	0.544	0.542		0.659	0.643		1.036	1.026		0.926	0.910	
	C2DF	0.605	0.602		0.623	0.616		0.985	0.961		0.967	0.962	
	MGC	0.480	0.476		**0.602**	**0.563**		0.969	0.962		0.986	0.967	
	MWC	0.478	**0.461**		0.625	0.589		**0.820**	**0.811**		**0.859**	**0.843**	
				0.426			0.536			0.787			0.788
MAE minimise	FCPC	0.315	0.311		0.461	0.458		0.747	0.736		0.764	0.743	
	FCPCe	0.318	0.310		0.473	0.443		1.177	1.160		0.761	0.740	
	C1DS	0.353	0.334		0.514	0.497		1.074	1.070		0.857	0.834	
	C2DF	0.366	0.351		0.467	0.462		0.972	0.960		0.935	0.920	
	MGC	**0.310**	**0.309**		**0.447**	**0.425**		0.939	0.913		0.972	0.961	
	MWC	0.313	0.310		0.469	0.442		**0.674**	**0.653**		**0.738**	**0.705**	
				0.282			0.407			0.620			0.621

Table 2 Comparison of prediction results For HPE model vs. the Sate-of-the-art models on four datasets

Model_Name	R^2	RMSE	MAE	R^2	RMSE	MAE
		IGC_{50}			LD_{50}	
HPE	**0.831**	**0.426**	**0.282**	**0.680**	**0.536**	**0.407**
hierarchical[12]	0.719	0.539	0.358	0.578	0.650	0.460
FDA[12]	0.747	0.489	0.337	0.557	0.657	0.474
group contribution[12]	0.682	0.575	0.411	–	–	–
nearest neighbor[12]	0.6	0.638	0.451	0.557	0.656	0.477
TEST consensus[12]	0.764	0.475	0.332	0.626	0.594	0.431
TopTox[9]	0.802	0.438	0.305	0.653	0.568	0.421
Hybrid2D[11]	0.810	–	–	0.629	–	–
		LC_{50}-DM			LC_{50}	
HPE	**0.811**	**0.787**	0.62	0.742	0.788	0.621
hierarchical[12]	0.695	0.979	0.757	0.71	0.801	0.574
single model[12]	0.697	0.993	0.772	0.704	0.803	0.605
FDA[12]	0.565	1.19	0.909	0.626	0.915	0.656
group contribution[12]	0.671	0.803	0.62	0.686	0.81	0.578
nearest neighbor[12]	0.733	0.975	0.745	0.667	0.876	0.649
TEST consensus[12]	0.739	0.911	0.727	0.728	0.768	0.545
TopTox[9]	0.788	0.805	**0.592**	**0.788**	**0.677**	**0.446**
Hybrid2D[11]	0.616	–	–	0.678	–	–

Table 3 Predictors with their attributes.

Predictor Name	Class	Architecture	Features
FCPC	Fully Connected Deep Neural Network	Standard feed-forward	2D physico-chemical features [4]
FCPCe	Fully Connected Deep Neural Network	Standard feed-forward	2D+3D physico-chemical features [16]
C1DS	Convolutional Neural Network	1D Convolution	SMILE Strings
C2DF	Convolutional Neural Network	2D Convolution	Fingerprints
MGC	Geometric Neural Network	Graph Convolution	Molecular Graph Coordinates (Atom Features)
MWC	Geometric Neural Network	Weave	Molecular Graph Coordinates (Atom and Pair Features)

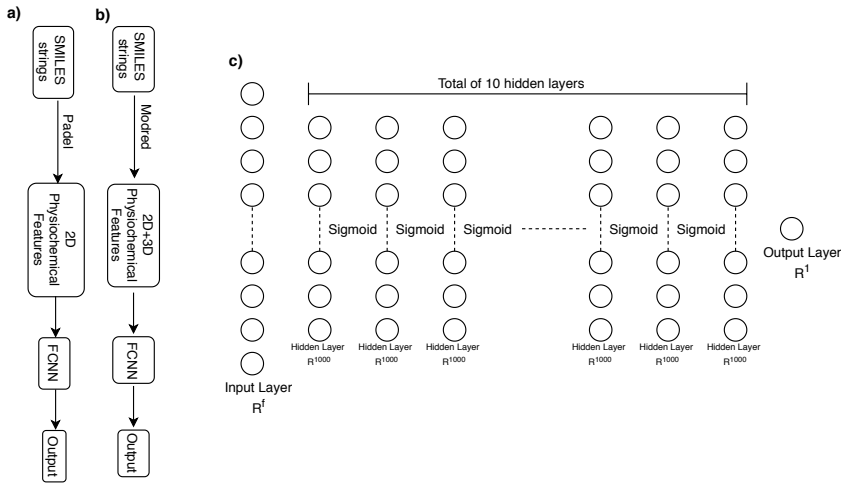

Figure 1 a) FCPC predictor which takes only 2D features computed using Padel descriptor. b) FCPCe predictor which takes both 2D and 3D features computed using Modred descriptor. c) FCNN used with Physico-Chem and physico-Chem Extended Features. Each unit of one layer is connected to all units of the succeeding layer. R^f shows the input features where f takes values of 1148 for FCPC and 1826 for FCPCe.

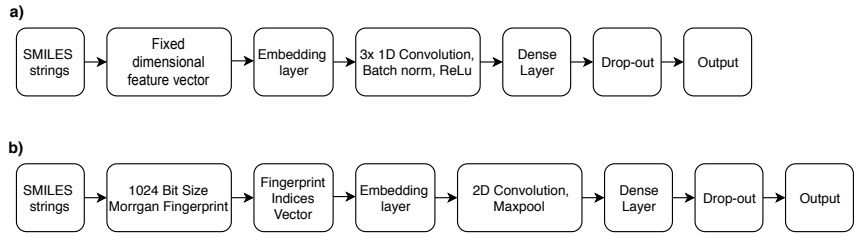

Figure 2 a) Architecture of convolution 1D SMILES (C1DS) predictor. b) Architecture of convolution 2D Fingerprints (C2DF) predictor.

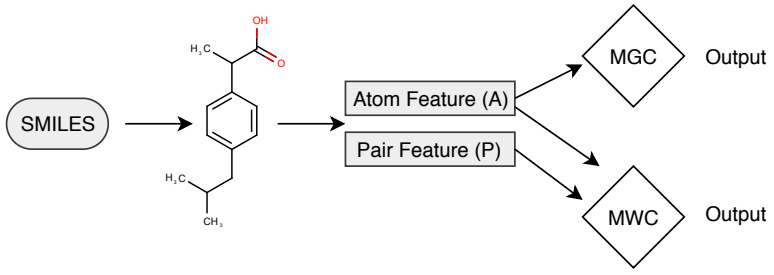

Figure 3 SMILES string is converted into 2D Molecular graph of Ibuprofen. The unmarked vertices/nodes represent carbon atoms whereas the edges represent bonds in the graph. Atom (A) and pair (P) features are computed from the molecular graph. MGC is trained only with the atom features whereas MWC takes both atom and pair features together.

Figures

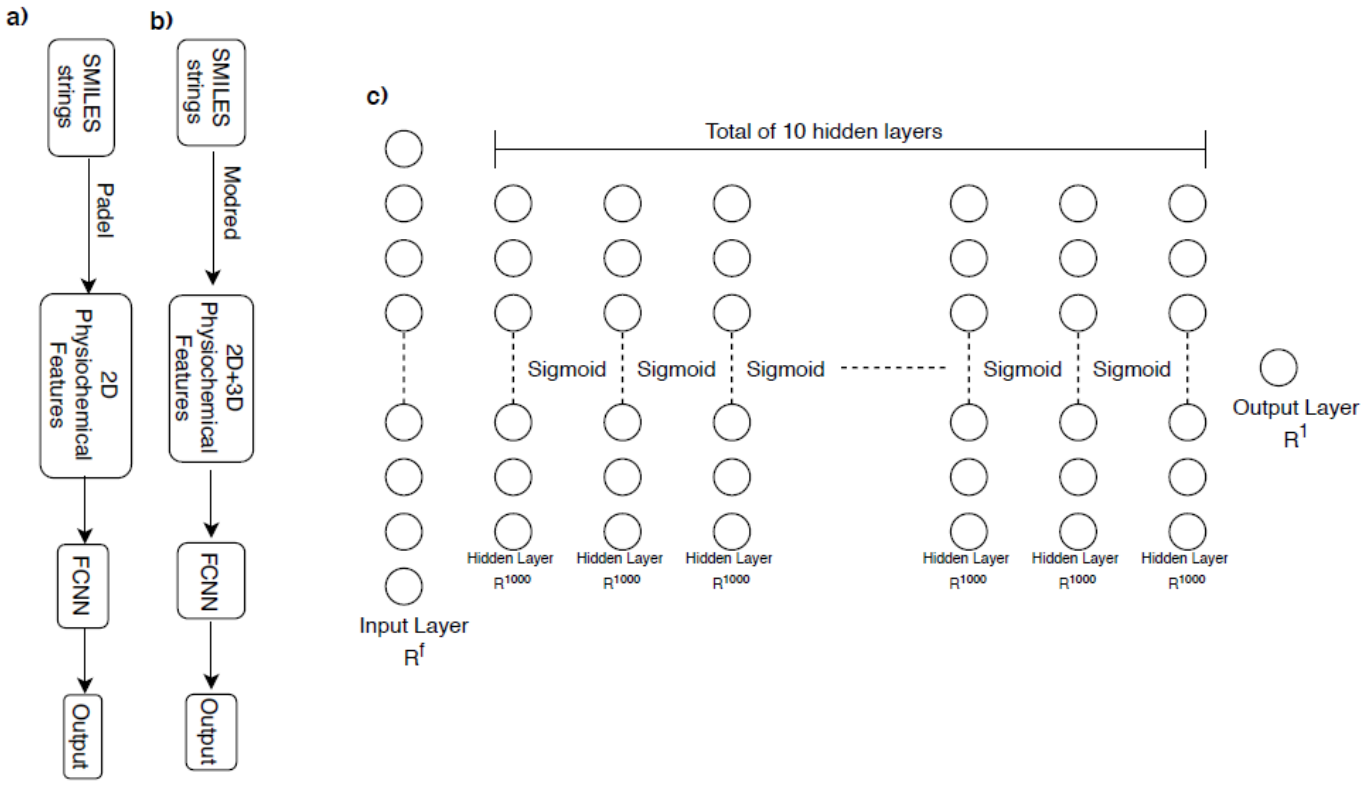

Figure 1

a) FCPC predictor which takes only 2D features computed using Padel descriptor. b) FCPCe predictor which takes both 2D and 3D features computed using Modred descriptor. c) FCNN used with Physico-Chem and physico-Chem Extended Features. Each unit of one layer is connected to all units of the succeeding layer. Rf shows the input features where f takes values of 1148 for FCPC and 1826 for FCPCe.

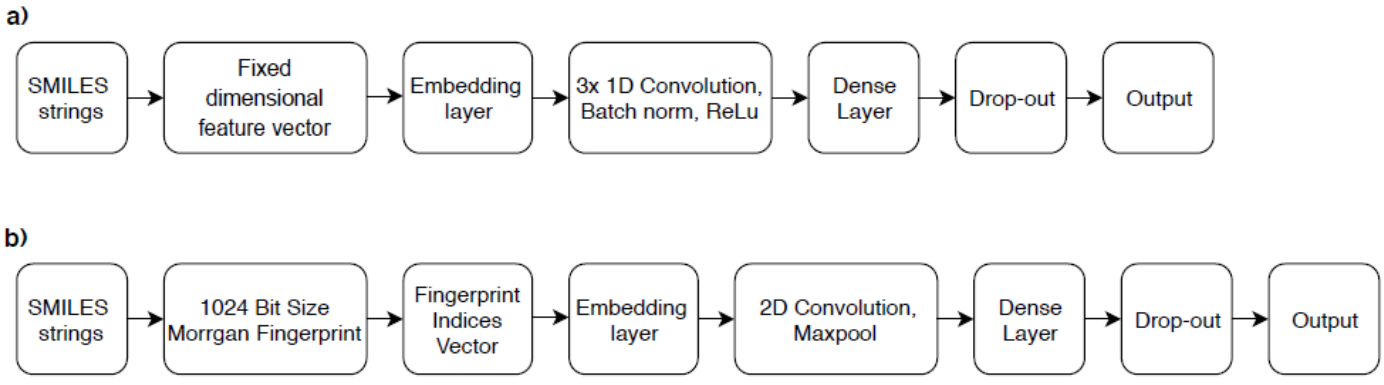

Figure 2

a) Architecture of convolution 1D SMILES (C1DS) predictor. b) Architecture of convolution 2D Fingerprints (C2DF) predictor.

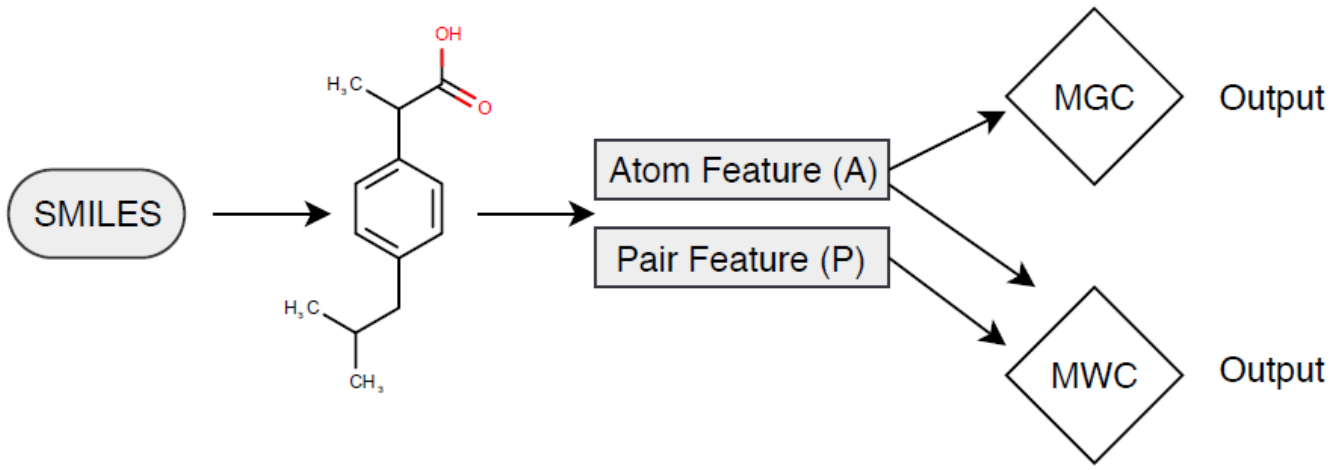

Figure 3

SMILES string is converted into 2D Molecular graph of Ibuprofen. The unmarked vertices/nodes represent carbon atoms whereas the edges represent bonds in the graph. Atom (A) and pair (P) features are computed from the molecular graph. MGC is trained only with the atom features whereas MWC takes both atom and pair features together.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- FCPEFCPCeCommenFeatures.xlsx
- FCPCFeaturesnames.xlsx
- FCPCeFeaturesnames.xlsx