Artificial Intelligence in the Life Sciences 1 (2021) 100014

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Research Article

Maxsmi: Maximizing molecular property prediction performance with confidence estimation using SMILES augmentation and deep learning

Talia B. Kimber[a,*], Maxime Gagnebin[b], Andrea Volkamer[a,*]

[a] In silico Toxicology and Structural Bioinformatics, Institute of Physiology, Charité-Universitätsmedizin Berlin, Charitéplatz 1, 10117, Berlin, Germany
[b] Berlin, Germany

ARTICLE INFO

Keywords:
Deep learning
Molecular property prediction
Data augmentation
Open-source
Confidence assessment
SMILES

ABSTRACT

Accurate molecular property or activity prediction is one of the main goals in computer-aided drug design. Quantitative structure-activity relationship (QSAR) modeling and machine learning, more recently deep learning, have become an integral part of this process. Such algorithms require lots of data for training which, in the case of physico-chemical and bioactivity data sets, remains scarce. To address the lack of data, augmentation techniques are increasingly applied in deep learning. Here, we exploit that one compound can be represented by various SMILES strings as means of data augmentation and we explore several augmentation techniques. Convolutional and recurrent neural networks are trained on four data sets, including experimental solubility, lipophilicity, and bioactivity measurements. Moreover, the uncertainty of the models is assessed by applying augmentation on the test set. Our results show that data augmentation improves the accuracy independently of the deep learning model and of the size of the data. The best strategies lead to the Maxsmi models, the models that **max**imize the performance in **SMI**LES augmentation. Our findings show that the standard deviation of the per SMILES prediction correlates with the accuracy of the associated compound prediction. In addition, our systematic testing of different augmentation strategies provides an extensive guideline to SMILES augmentation. A prediction tool using the Maxsmi models for novel compounds on the aforementioned physico-chemical and bioactivity tasks is made available at https://github.com/volkamerlab/maxsmi.

1. Introduction

Drug design is a time-consuming and costly process [1,2] with high attrition rates [3]. It can be supported with *in silico* methods by guiding the design process, optimizing compounds, and discarding those with undesired properties at an early stage of development. In this context, computer-aided drug design (CADD) has become central in the drug discovery pipeline and is widely adopted in research and development in both academia and pharmaceutical companies.

Over the last few decades, there has been a keen interest in machine learning (ML) and more specifically deep learning (DL), which have been applied to a variety of areas, covering computer vision [4], speech recognition [5], as well as the life sciences. Only to name a few, AlphaFold 2 from DeepMind which predicts protein folding [6], PotentialNet which focuses on protein-ligand binding affinity [7], *de novo* molecular design suitable for compound optimization [8], and cytotoxicity prediction as in the work by Webel et al. [9]. Such excitement in DL may be explained by the main three following factors [10].

1. The gain of computational power through graphics processing units (GPUs) and tensor processing units (TPUs). Platforms such as Google Colaboratory [11] allow any user to exploit high performance computing resources without any cost and such free and easy access is unprecedented.
2. The ever growing amount of available data. More data are created and stored in databases every day in various fields. Many processes are automatized making data more accessible and usable either internally, as in pharmaceutical companies, or publicly. For example in academic research, in competitions, such as Kaggle [12], or in challenges, such as D3R – the Drug Design Data Resource challenge [13] or the Tox21 challenge [14].
3. The advances in algorithms, making models perform better than ever before. Deep learning algorithms may surpass human performance if trained on a data set containing over 10 million data points, as suggested by Goodfellow et al. [10].

With the rise of ML/DL research, many applications have been extended to the field of molecular property and affinity prediction, even though insufficient data remains a challenge in the field.

* Corresponding author.
E-mail addresses: talia.kimber@gmail.com (T.B. Kimber), andrea.volkamer@charite.de (A. Volkamer).

https://doi.org/10.1016/j.ailsci.2021.100014
Received 4 October 2021; Received in revised form 13 November 2021; Accepted 15 November 2021
Available online 18 November 2021

Encoding molecular compounds in both human- and computer-readable formats is a necessary step in CADD. A convenient encoding is SMILES, or simplified molecular-input line-entry system [15]. As the name suggests, SMILES is a linear notation of a molecule based on atom and bond enumeration, as well as branch, ring closure, and disconnection specification. Several advantages arise from this compact representation.

1. The printable characters make SMILES easily readable by computers and decipherable by humans.
2. Being a single line, SMILES resemble words and are therefore cheap to store.
3. Such a notation is very popular and many open-source databases store compounds in SMILES.

However, there is a trade-off between readability and specification: having a compact encoding means losing detailed information about the molecule such as 2D or 3D features. Moreover, subtle chemistry rules such as aromaticity do not have a standard way of being handled [16].

The implementation of a SMILES given a compound can be described as follows: from any starting atom in the molecule, enumerate the atoms and bonds following a path in the molecular graph. Two aspects of this construction lead to the non-uniqueness of SMILES: 1. the atom to start the enumeration from, and 2. the path to follow along the graph.

Therefore, one molecule can have many different valid SMILES, simply by starting the enumeration from a different atom or by choosing a different path. Nevertheless, in some settings, having a bijection between a molecule and its SMILES notation may be sought. For example, when determining the overlapping molecules from two data sets. In this context, most cheminformatics tools have their own algorithm implemented allowing them to always retrieve the same SMILES given a molecular graph, such a SMILES is called canonical [17].

As mentioned previously, deep learning being data greedy and both physico-chemical and bioactivity databases being meager, elaborate techniques have to be integrated to unleash the full potential of deep neural networks. In this context, data augmentation in general [18,19], and more specifically SMILES augmentation [20–23], is a powerful assistance in molecular prediction. From a machine learning perspective, data augmentation allows the model to see the same object through different angles and has been successfully applied in image classification [4,24], where images undergo transformations such as flipping, coloring, cropping, rotating, and translating. From a computational perspective, SMILES augmentation is advantageous because generating random SMILES is fast and memory efficient, and even though training a model may be more computationally expensive, it remains cheap to evaluate.

The first occurrence of SMILES augmentation in QSAR modeling was developed by Bjerrum [20], where affinity against dihydrofolate reductase (DHFR) is predicted on a small data set of 756 compounds. The model consists of long short-term memory (LSTM) layers and a fully connected layer for the normalized $\log IC_{50}$ value. Each molecule in the data set is augmented on average 130 times. The model with SMILES augmentation reaches a test correlation coefficient of 0.68, a 0.12 increase with respect to the canonical model. From then on, several studies have built on the same idea, applying SMILES augmentation in QSAR modeling [21]. Moreover, convolutional neural networks have successfully been applied in the context of SMILES augmentation, outperforming models using traditional molecular descriptors [22,23]. Such augmentation techniques have also emerged in related fields, such as retrosynthesis [25,26] and generative modeling [27,28]. While all these studies show the benefit of augmenting the data, none of them focus, to the best of our knowledge, on a systematic analysis on how to augment the data set best, and most decide *a priori* on an augmentation number. This study aims at filling this gap by offering a systematic augmentation approach, both in the augmentation strategies and by how much the data should be augmented. Moreover, a command-line interface is available for users interested in the prediction of physico-chemical properties for novel molecules and assessing the uncertainty of the prediction. To this end, all code is made freely available at https://github.com/volkamerlab/maxsmi.

2. Methods

In this section, we first describe different augmentation strategies which can be used for data augmentation when dealing with SMILES. Second, we illustrate how SMILES augmentation can be viewed as an ensemble learning technique when it comes to prediction. We then examine the deep learning models that are trained in this study.

2.1. Augmentation strategies

As discussed in the Introduction, one compound can have several valid SMILES, since both the starting atom and the path along the molecular graph used to generate the SMILES can differ. Here, the way a user can explore such random SMILES is detailed and five strategies to augment a single SMILES to multiple SMILES are described: no augmentation, augmentation with duplication, augmentation without duplication, augmentation with reduced duplication, and augmentation with estimated maximum. For the following sections, we assume that we are given a data set D, containing N pairs of {compound, label}. Label refers to the measured property, such as lipophilicity or solubility. The implementation of these strategies is based on the open-source cheminformatics software RDKit [29].

2.1.1. No augmentation

The level zero to augmentation is having no augmentation or, in other terms, augmentation of zero. This means that given a data set D with N compounds, the "no augmentation" version of D also contains N SMILES. More specifically, in this setting, the SMILES associated with each compound is the canonical SMILES.

2.1.2. Augmentation with duplication

Generating random SMILES implies picking at random an initial atom and following a random path along the molecular graph. Augmenting the data set D by m means that for each of the N compounds in D, m instances of random SMILES are drawn and the associated labels are matched for each compound. In this case, augmenting D by m would result in the augmented data set containing $N \times m$ data points. In this scenario, all molecules in D are multiplied by the same factor m. Consequently, smaller molecules, with fewer SMILES variations, will contain more duplicates whereas larger molecules are more likely to cover a diverse set of random SMILES. A disadvantage of such an augmentation strategy is that SMILES corresponding to small molecules will be over-represented in the data set and could create a bias in model training.

2.1.3. Augmentation without duplication

Removing duplicated entries is common in data wrangling [30]. In the context of SMILES augmentation, this translates to discarding duplicates after having generated a number of random SMILES. For data set D, the final number of data points after augmentation varies according to the augmentation number, i.e. the number of times a sample is drawn from the valid SMILES space, and the size of the molecules in the data set. A disadvantage of such an augmentation strategy is that small molecules, which presumably possess fewer unique SMILES representatives, will be under-represented in the data set and could create a bias in model training.

2.1.4. Augmentation with reduced duplication

In order to find a compromise between keeping or removing all duplicates, the notion of augmentation with reduced duplication is introduced. In this setting, only a fraction of the number of duplicates is kept. Mathematically speaking, if the data set D is augmented by m, then a

function $f(m)$ which grows slower than linear is used to control the number of replicas kept for each SMILES. Sensible functions would be the squared root function $f(m) = \sqrt{m}$ or the natural logarithm $f(m) = \ln(m)$, the former being used for the experiments in this study.

A corner case of the former three augmentation strategies by augmentation number m is when $m = 1$. In this instance, a random SMILES will be generated and the number of data points would still be N, as in the "no augmentation" case, with the difference that "no augmentation" contains canonical SMILES only.

2.1.5. Augmentation with estimated maximum

The final strategy described is augmentation with estimated maximum, which aims to cover a wide range of the valid SMILES space for a given compound, or in other words, to generate a number of unique SMILES that depends on the compound. In our study, the implementation of this augmentation strategy randomly samples SMILES corresponding to a compound, and the sampling process is stopped once the same SMILES string has been generated a pre-defined number of times. The experiments of this study set 10 generations of the same SMILES as a stopping criterion. It is noteworthy that the number of SMILES this method generates is highly dependent on the size of the compound, unlike the previous methods which always generate a number of SMILES bounded by m. For example, our implementation of this augmentation strategy generated 50 659 unique SMILES variations for the compound given by the canonical SMILES CC(=O)C1(C)CCC2C3C=C(C)C4=CC(=O)CCC4(C)C3CCC21C, whereas only three were generated for the canonical SMILES C=CC=C, namely C(=C)C=C, C(C=C)=C, and C=CC=C.

2.2. SMILES augmentation as ensemble learning for compound prediction and confidence measure

The application of data augmentation strategies during training has proven to be successful, as shown in previous works [4,24]. In QSAR modeling particularly, SMILES augmentation is not only beneficial on the training set [22], but there are also advantages of augmenting the test set, or more generally, an unlabeled data set, as explained in this section.

Let us assume that a model M with a set of parameters Θ was trained for a certain number of epochs. Let us consider an unlabeled data set containing N compounds for which we want to make predictions. Each compound C can be augmented using random SMILES: $S_1(C), S_2(C), \ldots, S_k(C)$, where k depends on the strategy. The model M_Θ produces a prediction for each of those SMILES, i.e. for $i \in \{1, \ldots, k\}$

$$\hat{y}_i(C) = M_\Theta(S_i(C)),$$

leading to a *per SMILES* prediction rather than a *per compound* prediction. Using an aggregation function $A : \mathbb{R}^k \to \mathbb{R}$, such as the mean, a prediction for compound C can be computed as

$$\hat{y}(C) = A\big(\hat{y}_1(C), \ldots, \hat{y}_k(C)\big).$$

Such aggregation can be viewed as a consensus among the SMILES prediction and interpreted as ensemble learning for a given compound.

Additionally, if the standard deviations of the predictions are computed, they can be interpreted as a confidence in the molecular property or activity prediction. If the standard deviation is large, then there is a high variation in the per SMILES prediction, and the model is uncertain in its per compound prediction. An illustration of such a molecular prediction is shown in Fig. 1. Following the rationale by Tagasovska and Lopez-Paz [31], the aleatoric and epistemic uncertainties are often intertwined; the uncertainty computed in our work rather falls into the aleatoric category, a type of uncertainty linked to the model predictions and the randomness in the input data [31–33].

2.3. Deep learning models

Neural networks are powerful algorithms that allow accurate predictions on various tasks. In the case of QSAR/ML/DL modeling and more specifically the use of SMILES representation, two types of models can be applied, convolutional [10, Chapter 9] and recurrent [10, Chapter 10] neural networks.

In this study, comparing deep learning models and how they perform with respect to data augmentation is one of the key focuses. To this end, three types of models are architectured and trained, namely 1D and 2D convolutional neural networks (CONV1D, CONV2D) as well as a recurrent neural network (RNN). The architecture of the recurrent network consists of an LSTM layer, followed by two fully connected layers of 128 and 64 units, respectively. It was inspired by Bjerrum [20], in which an LSTM layer is followed by a single 64 unit fully connected layer. Using a similar approach, a single 1D convolutional layer of kernel size 10 and stride 1 is applied in the CONV1D model. Two fully connected layers follow the convolution. The CONV2D adheres to the same pattern but instead of using a 1D convolution, a 2D convolution operation is performed using one single channel. Finally, all three architectured models stay consistent in the depth of the network and remain shallow.

In this study, all deep learning models are trained for 250 epochs, using mini-batches of size 16, where the mean squared error is the considered loss. Optimization is done with stochastic gradient descent and a learning rate of 0.001. Note that a fixed number of epochs is used in this study, but for sake of completeness, three sample models with early stopping were also run. The results with and without early stopping did not change significantly (data not shown). Moreover, some models were trained by adapting the number of epochs with respect to the augmentation number, but this only proved to overfit the training set and yielded the same results on the test set as training with 250 epochs (data not shown).

3. Data and experimental setup

This section introduces the data sets used in this study, namely their provenance as well as the required preprocessing. Furthermore, a step by step instruction for efficient SMILES augmentation is described. Finally, the evaluation design and experimental setup are covered.

3.1. Provenance

The data in this research come from two sources: MoleculeNet [34], and the ChEMBL database [35], chosen for two main reasons.

1. They are freely available and easily downloadable or retrievable.
2. They are often used as benchmarks for study comparison [22,36,37].

For tasks in MoleculeNet, we focus on physico-chemical prediction tasks and retrieve the data from the three following sets of varying sizes, all available as part of DeepChem [38] at https://deepchem.readthedocs.io/en/latest.

1. Measured water solubility is referred to as the ESOL data set [39]. The raw data contains 1 128 data points. This data set is further processed to only include molecules with at most 25 heavy atoms for experimental setup and is referred to as ESOL_small.
2. The FreeSolv [40] data set consists of 642 pairs of SMILES and experimental hydration free energy of small molecules in water (kcal/mol).
3. The lipophilicity data set originates from ChEMBL [35] and contains 4 200 pairs of SMILES and experimental values of octanol/water distribution coefficient.

Bioactivity data can be found in large quantities in ChEMBL. To date, over 18 million activities are stored in the database, covering

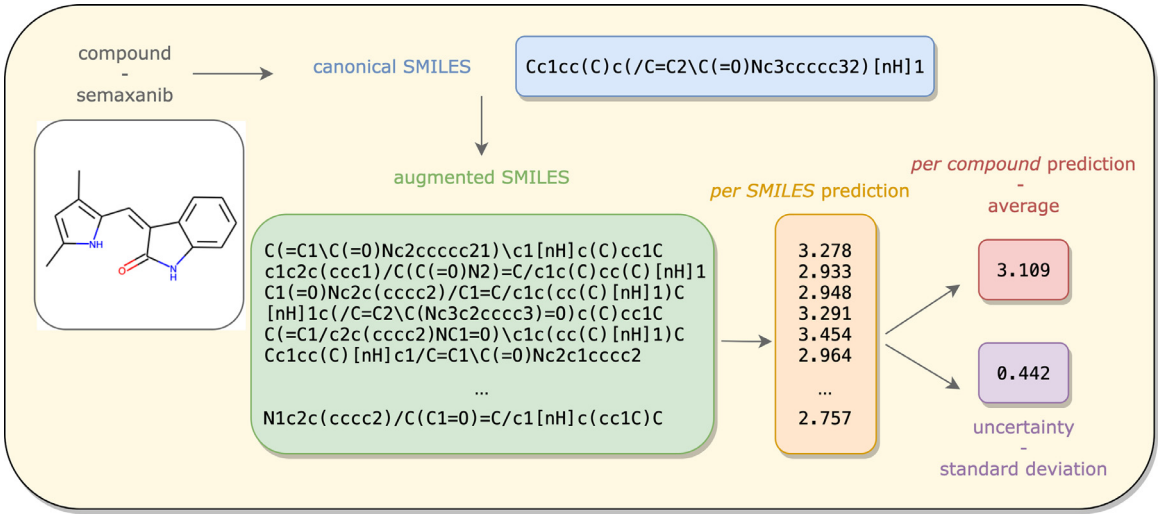

Fig. 1. Compound prediction and confidence measure thanks to SMILES augmentation. Given a compound represented by its canonical SMILES, a set of random SMILES are generated. The trained machine learning model produces a prediction for each of the SMILES variations. Aggregating these values leads to a per compound prediction and computing the standard deviation is interpreted as an uncertainty in the prediction.

Table 1
Data sets for this study. Size of the data sets before and after preprocessing, as well as the size of the training and test sets before applying an augmentation strategy, and the provenance of the data.

Dataset	Size before preprocessing	Size after preprocessing	Train set 80%, before augmention	Test set 20%, before augmention	Provenance
ESOL	1 128	1 128	902	226	MoleculeNet[a]
ESOL_small	1 128	1 068	854	214	MoleculeNet[a]
FreeSolv	642	642	513	129	MoleculeNet[a]
Lipophilicity	4 200	4 199	3 359	840	MoleculeNet[a]
Affinity (EGFR)	6 026	5 849	4 679	1 170	Kinodata[b]

[a] https://deepchem.readthedocs.io/en/latest.
[b] https://github.com/openkinome/kinodata.

more than 14 000 targets and two million compounds [41]. Among targets, kinases are a well studied protein family due to their involvement, among others, in cancer and inflammatory diseases [42]. Kinodata, from the Openkinome organization [43], provides an already curated data set of human kinase bioactivities, retrieved from one of the latest versions to date of ChEMBL (version 28) and is freely available at https://github.com/openkinome/kinodata. Moreover, for this study, Kinodata is further filtered for the epidermal growth factor receptor (EGFR) kinase [44], since it is known to be an important drug target. Its UniProt identifier is given by P00533 [45]. Affinity towards the EGFR kinase is quantified using pIC_{50} values, the negative base 10 logarithm of IC_{50} [46]. Information about the data set provenance and size are detailed in Table 1.

3.2. Data preprocessing and input featurization

In order to train a deep learning neural network on data containing molecular compounds, the data set undergoes preprocessing and compounds encoding.

Once the data sets are retrieved from their original source, invalid SMILES, detected by RDKit [29], not available (NA) values and disconnected compounds, marked by a dot in a SMILES, are removed. Molecules are transformed to the canonical SMILES representation, using RDKit functionalities.

For model training, the SMILES are one-hot encoded, based on a dictionary of unique symbols constructed from the SMILES in the data. Atoms represented by two letters, such as Br for bromine or Cl for chlorine, as well as @@ for chirality specification, are treated as if single symbols. Finally, all inputs are padded up to the length of the longest SMILES. The reader is kindly referred to the work by Kimber et al. [47] for further details on one-hot encoding and padding.

3.3. Important steps in SMILES augmentation

When processing SMILES for augmentation, some technical aspects are essential. This section assumes a training and test split, but the rationale is the same in the presence of a validation set.

Firstly, it is important that the data are first split and then augmented, rather than augmented and split. In the latter case, one compound could have SMILES appearing in both training and testing leading to most probably excellent performance, but yet statistically incorrect.

Secondly, storing values such as the length of the longest SMILES or the dictionary of characters should be done not before but after augmenting the data. Indeed, augmentation may lead to the extension of the dictionary as well as the lengthening of SMILES. For example, the canonical SMILES CCCC consists of the letter C solely and contains four characters. However, one of its possible random variations is C(C)CC, which not only introduces new characters, such as the opening "(" and closing ")" of branches but is composed of six characters. Therefore, critical values such as length and dictionary should be retained after augmentation.

Finally, these same values should be computed on the union of the training and the test set for the smooth training and evaluation of the model. Indeed, if the dictionary of characters is only built on the basis of the SMILES in the training data, there might be additional atoms, or characters in the test set that the model will not recognize and will not

be able to one-hot encode. Moreover, if the length of the longest SMILES is taken from the training set and not the union of the training and test sets, augmentation on the test set could produce a longer SMILES than the longest one in the training set, leading to dimensionality errors.

For all the above reasons, it is important for machine learning engineers to abide by the steps as described in this section for statistically correct results, as well as programmatic error-free model training and evaluation.

3.4. Experimental setup and model evaluation

In order to draw a conclusion on the efficiency of data augmentation, three data sets of varying sizes are considered, namely ESOL, FreeSolv, and lipophilicity (see the Provenance section). For each of these sets, the data are split once into 80% training and 20% test set, with a fixed random seed for testing to be consistent with the augmentation schemes. Given all possible combinations between the five augmentation strategies and different augmentation numbers, the three deep learning models, and the various data sets, including cross-validation would have added considerable computational costs and has therefore not been implemented in this study.

For model evaluation, the root mean squared error (RMSE) [48] on the test set is reported, so that the lower the RMSE value, the better the model. However additional information such as the measure of goodness of fit, also known as the R2 value [49], on both training and test sets, as well as the time required for model training and evaluation are also stored.

Five augmentations strategies are studied: No augmentation, which considers the canonical SMILES representation. The augmentations with, without, and with reduced duplication, for numerous augmentation numbers: a finer grid from 1 to 20 with a step size of 1, and a coarser grid from 20 to 100 with a step size of 10. Finally, the estimated maximum strategy where a SMILES representation has to be generated 10 times for the process to stop. For this last strategy, the ESOL_small data set (see Table 1) is used to keep the augmentation to a reasonable time-scale. For the same reason, the same augmentation strategy is not run on the lipophilicity data set.

The augmentation strategies are applied to both the training set and the test set, so that for example, if the FreeSolv training data set is augmented 20 times without duplication, then so would the FreeSolv test set.

Ensemble learning is applied on each test set and the mean is used as aggregation. However, a user could easily adapt it to another function, such as the median. The standard deviation is stored for each compound in the test set.

Moreover, a Random Forest (RF) model [50] is used as a baseline, with all default parameters from Scikit-learn [51]. The inputs to the model are the Morgan fingerprints of radius 2 and length 1024. Augmentation strategies as discussed above are not applicable in the context of fingerprints.

Simulations are run on a GeForce GTX 1080 Ti, provided by the central HPC cluster of the Freie Universität Berlin [52].

3.5. Code and documentation

All code is written in Python 3 [53] following PEP8 style guide [54] and is freely available at https://github.com/volkamerlab/maxsmi. Results of this study can be found at the same link. Examples and documentation, generated via Read the Docs [55], can be found at https://maxsmi.readthedocs.io/en/latest/.

Package management is done with Anaconda [56]. RDKit [29] is used for cheminformatics, PyTorch [57] for deep learning, and other popular packages such as Scikit-learn [51], NumPy [58], and Pandas [59] for general purposes. Continuous integration is deployed with Github actions [60] ensuring runs on Linux, Mac, and Windows operating systems. Unit tests are done with Pytest [61], and code coverage is measured via Codecov [62].

4. Results and discussion

This section gives a thorough analysis of the results that are obtained using the experimental setup described in the previous section and provides the reader with guidelines on data augmentation applicable to new data and exemplified with affinity measurements towards the EGFR kinase. An example of the user prediction for compounds through a simple command-line interface is described.

4.1. SMILES augmentation improves model performance

As mentioned previously, deep learning models are data greedy and the findings of our study reinforce this statement by a systematic analysis of performance differences when augmenting the input data. Feeding a neural network with different SMILES representations of the same compound leads to better performing models, as shown in Figs. 2, A1, and A2. Improvements are also visible with respect to the baseline model. These observations are made on all three physico-chemical data sets, namely ESOL, FreeSolv, and lipophilicity, independently of the data set size that ranges between approximately 600 and 4 000 compounds (see Table 1). For example, the ESOL performance with no augmentation has an RMSE value of 0.839 for the CONV1D model, whereas the performance of the same model with reduced augmentation and $m = 70$ achieves an RMSE as low as 0.569, see Fig. 2. As the number of augmentation increases, the RMSE values become smaller, indicated by lighter shades of purple in Figs. 2, A1, and A2. Note that at first, as the augmentation number increases in the single digits, there is a clear increase in the performance of the models. For example, for the lipophilicity data set augmented with duplication and the CONV2D model, the single random SMILES model has an RMSE value of 1.309 and reaches values below 1 as of an augmentation number of 4 (see Fig. A2). On the ESOL data set, the RNN model without duplication starts at an RMSE of 1.016 and reaches values below 0.8 after only an augmentation of 5 (see Fig. 2). However, the performance steadily reaches a plateau. For example, the RMSE of the CONV1D model trained on FreeSolv is slightly above 1 as of 20 number of augmentation and fluctuates around this value thereafter, as shown in Fig. 3. Similar observations can be made for ESOL and lipophilicity. Using the same model, the RMSE on ESOL reaches a plateau around 0.60 at 40 augmentation steps (see Fig. A4) and lipophilicity around 0.60 at 60 (see Fig. A5). This result suggests the following:

1. There does not seem to be one optimal value that particularly stands out.
2. A trade-off between performance and computation time must be found. As expected, the computation time increases as the number of data points increases, as shown in Fig. A6.

4.1.1. Deep learning model performance by architecture

Not only does augmenting the data set overall help the learning for all three considered tasks, but so is the case for all three deep learning architectures. This leads to the observation that augmentation improves performance independently of the deep learning model, suggesting that for any future QSAR study for molecular property prediction using SMILES and deep learning, SMILES augmentation should be the method of choice. However, in this particular study and these particular deep learning architectures, results point to the fact that the CONV1D model tends to outperform the RNN model, which itself seems to outperform CONV2D. As shown in Fig. 4, on the ESOL data using augmentation with reduced duplication, as of an augmentation number of 40, the RMSE value of the CONV2D model fluctuates around 0.7, the RNN model around 0.65 and the CONV1D around 0.6, promoting the latter model to best performing model. This exhibits the power of convolutions

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	0.964	1.009	1.016	0.975	0.978	1.020	0.964	0.986	1.022
2	0.785	0.787	0.964	0.786	0.768	0.935	0.785	0.769	0.943
3	0.785	0.726	0.896	0.784	0.962	0.899	0.785	0.728	0.805
4	0.732	0.761	0.881	0.718	0.761	0.847	0.739	0.774	0.817
5	0.716	0.748	0.791	0.716	0.723	0.789	0.712	0.730	0.793
6	0.666	0.743	0.788	0.679	0.695	0.771	0.673	0.714	0.760
7	0.660	0.676	0.773	0.667	0.670	0.775	0.658	0.683	0.764
8	0.712	0.692	0.743	0.666	0.700	0.744	0.672	0.721	0.733
9	0.642	0.761	0.727	0.642	0.655	0.718	0.641	0.671	0.715
10	0.646	0.689	0.729	0.663	0.706	0.696	0.647	0.692	0.752
11	0.638	0.668	0.696	0.620	0.720	0.760	0.639	0.653	0.696
12	0.606	0.666	0.715	0.615	0.652	0.673	0.613	0.678	0.670
13	0.621	0.692	0.698	0.613	0.664	0.682	0.615	0.669	0.720
14	0.633	0.656	0.702	0.631	0.645	0.670	0.628	0.631	0.687
15	0.622	0.640	0.676	0.596	0.652	0.672	0.626	0.650	0.681
16	0.624	0.680	0.726	0.615	0.662	0.638	0.624	0.660	0.670
17	0.608	0.675	0.689	0.615	0.661	0.667	0.626	0.657	0.658
18	0.615	0.671	0.705	0.606	0.633	0.660	0.592	0.697	0.745
19	0.605	0.664	0.666	0.604	0.658	0.642	0.599	0.749	0.682
20	0.615	0.654	0.656	0.604	0.649	0.630	0.605	0.666	0.632
30	0.612	0.664	0.626	0.592	0.668	0.637	0.619	0.653	0.609
40	0.583	0.684	0.626	0.595	0.656	0.620	0.589	0.674	0.605
50	0.593	0.688	0.641	0.583	0.662	0.601	0.599	0.661	0.617
60	0.589	0.666	0.616	0.584	0.659	0.608	0.584	0.692	0.638
70	0.588	0.660	0.607	0.577	0.675	0.589	0.569	0.659	0.592
80	0.582	0.647	0.642	0.573	0.675	0.632	0.587	0.651	0.622
90	0.589	0.676	0.600	0.598	0.664	0.633	0.579	0.654	0.632
100	0.580	0.650	0.591	0.596	0.658	0.646	0.582	0.684	0.623

no augmentation			estimated maximum			baseline
CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	RF
0.839	0.895	0.930	0.576	0.657	0.683	1.102

Fig. 2. Test RMSE using data augmentation on the ESOL data set. The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the ESOL data set is augmenting the data set 70 times using a reduced number of duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

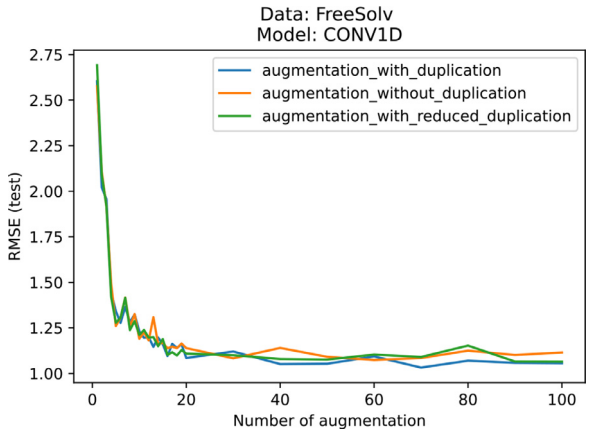

Fig. 3. Performance reaches a plateau independently of the augmentation strategy. The performance of the CONV1D model trained and evaluated on the FreeSolv data set reaches a test RMSE value slightly above 1 as of 20 augmentation steps and fluctuates around this value thereafter, for all augmentation strategies: with, without, and with reduced duplication. For the ESOL and lipophilicity data, see Figs. A4 and A5.

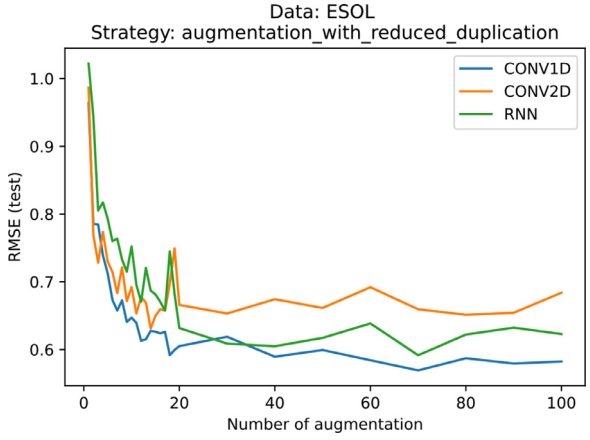

Fig. 4. The 1D convolutional (CONV1D) model outperforms the recurrent (RNN) and 2D convolution (CONV2D) models. The figure shows the evolution of the root mean squared error (RMSE) on the test set with respect to the number of augmentation using reduced duplication on the ESOL data. CONV1D outperforms RNN, which outperforms CONV2D.

and their ability to extract relevant features in compounds based on one-hot encoded SMILES input. This also implies that although applying 2D convolutions to SMILES is programmatically feasible, 1D convolutions are better suited than 2D convolutions, the latter having shown great success in image classification. Indeed, when considering the one-hot encoded matrix, SMILES are more similar to words, in which the position of the atoms is important, rather than to images.

4.1.2. *There is no* best *augmentation strategy applying to all data sets*

From an augmentation strategy point of view, conclusions are not straightforward. The three augmentation strategies, namely with, without, and with reduced duplication, all perform similarly well, without one standing out. For example, the test RMSE on the FreeSolv data set trained using the CONV1D model reaches values just above 1 for all three strategies, as shown in Fig. 3.

Moreover, generating a large portion of the SMILES space using the strategy with estimated maximum surprisingly does not lead to the best results. On the ESOL data set, this strategy reaches a test RMSE of 0.683 using RNN, whereas the same model but using strategies with, without, and with reduced duplication already outperforms the estimated maximum as of an augmentation number of 19 and onward, as shown in

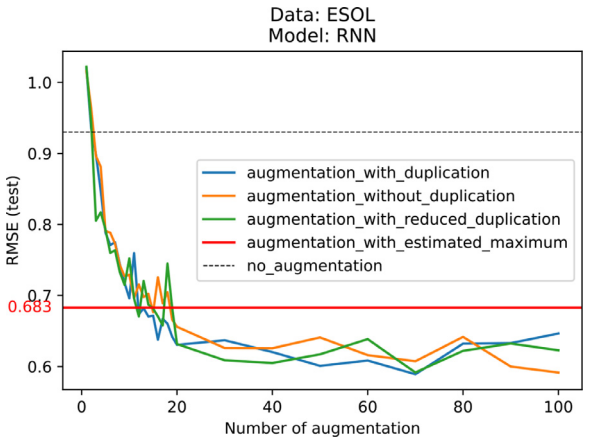

Fig. 5. Generating a large portion of the SMILES space does not necessarily lead to the best performance. Even though the RNN model is presented with SMILES variations that cover a large portion of the SMILES space using the augmentations strategy with estimated maximum, on the ESOL data set, this strategy does not achieve the best results.

Fig. 5. Although less obvious than in the ESOL case, a similar conclusion can be made on the FreeSolv data set and for example the CONV1D model, as shown in Fig. A3. This suggests that there might be a point of saturation, where the neural network stops learning, even though being fed more data.

4.1.3. *Maxsmi models: Best performing model per data set*

From the results of the experiments, as mentioned previously, there does not seem to be one augmentation strategy that fully stands out, neither does a particular model. However, from a purely numerical standpoint, there is an optimal performance value and this value is highlighted in yellow in Figs. 2, A1, and A2. For the ESOL data set, the tuple of (model, augmentation number, augmentation strategy) that yields best performance is the CONV1D model, an augmentation number of 70 and keeping a reduced number of duplicates. For the FreeSolv data set, the same model but generating 70 random SMILES keeping all duplicates is the best setting. Finally, for lipophilicity, generating 80 random SMILES and removing duplicates leads to the best performance. Given these three best models, we select them for further analysis, henceforth calling them Maxsmi models and summarized in Table 2.

4.1.4. *Performance comparison between canonical and random SMILES*

One interesting observation from this study is the performance comparison between training a model with canonical SMILES versus training a model using one random SMILES representation, in other terms, augmentation of 1. The canonical model systemically outperforms the model that uses a random SMILES. More specifically, for the ESOL data set, the canonical model reaches an RMSE value of 0.839 using CONV1D, whereas the random version 0.964 with the same model. In the FreeSolv and lipophilicity cases, the canonical model yields an RMSE value of 1.963 and 0.994, versus 2.577 and 1.268 for random SMILES. A possible explanation for such an outcome is the simplicity in the canonical SMILES representation. The algorithm in RDKit produces the more readable SMILES representation, one that avoids branches, as well as nested branches. Table 3 shows some of these differences. For example, a random version might add brackets, where the canonical version has none (see the first row in Table 3), it might add sets of brackets, where the canonical version keeps them to a minimum (see the second row in Table 3) and the random version even allows nested brackets where the canonical version avoids them (see the last row in Table 3).

To conclude with this observation, if SMILES augmentation cannot be applied for future studies for any reason, practitioners are highly

Table 2
The best augmentation strategies define the Maxsmi models. After training three data sets (ESOL, FreeSolv, and lipophilicity) on various deep learning models (CONV1D, CONV2D, and RNN), using different augmentation numbers and strategies, the setting that yields the best performance, or lowest root mean squared error (RMSE) on the test set is selected and named the Maxsmi model.

Data	Model	Augmentation number	Augmentation strategy	Test RMSE
ESOL	CONV1D	70	With reduced duplication	0.569
FreeSolv	CONV1D	70	With duplication	1.032
Lipophilicity	CONV1D	80	Without duplication	0.593

Table 3
Models based on the canonical SMILES outperform the ones based on a single random SMILES. The test prediction for a model trained and evaluated on the RDKit canonical SMILES systematically performs better than the same model trained and evaluated on a single random SMILES. ESOL is the prediction task leading to the values in the table.

Canonical SMILES	Random SMILES	True value	Canonical SMILES prediction (&error)	Random SMILES prediction (&error)
CCCCCC	C(C)CCCC	−3.84	−2.87 (0.97)	−2.77 (1.07)
CCCC(=O)CC	C(=O)(CCC)CC	−0.83	−1.37 (0.54)	−1.65 (0.82)
CCCC(=O)OCC	C(OC(CCC)=O)C	−1.36	−1.14 (0.22)	−0.55 (0.81)

recommended to consider the canonical SMILES representation rather than a random one.

4.2. Ensemble learning for compound prediction and confidence measure

Using the Maxsmi models established above, we look into more details at the information gained from ensemble learning for molecular prediction, and more specifically at the average and standard deviation computed from the per SMILES prediction. Feeding different SMILES representations to the model and aggregating the prediction for each SMILES variation to obtain a single prediction per compound is valuable not only from a practical point of view where molecular prediction is more informative than a SMILES prediction, but it also allows the model to merge information coming from different perspectives of the same compound. Moreover, the standard deviation associated with the SMILES predictions allows to quantify the uncertainty of the prediction of the model toward a given compound. The higher the standard deviation for a molecule, the less concurrent are the predictions by the model, and thus, less confident.

4.2.1. Difference between canonical vs. averaged prediction

Considering the Maxsmi models trained with their respective augmentations, we analyze the difference in prediction on the test set when using the canonical or the averaged prediction. More specifically, we compare the prediction error of the Maxsmi models when evaluated on the test set twice: once using the canonical SMILES for compound prediction and a second time averaging the per SMILES prediction using the same augmentation number and strategy which was used for training. For both evaluations, the error between the prediction and the true value is computed. Fig. 6 shows the histogram of these errors on the ESOL data. As shown in the figure, more compounds have an error close to zero using the ensemble learning evaluation rather than the canonical, which incentives the use of ensemble learning for future studies. However, this gain is marginal and the canonical prediction performs similarly well compared to the averaged prediction. In light of the overall gain in the accuracy of the models, this indicates that augmentation during training is the more crucial step. As discussed in the following paragraph, an advantage of using augmentation on the test set is to estimate the confidence of the model in its prediction.

4.2.2. More confident model implies smaller prediction error

As mentioned in the SMILES augmentation as ensemble learning for compound prediction and confidence measure section, computing the standard deviation of the per SMILES prediction provides a confidence measure in the compound prediction. In this section, we analyze the

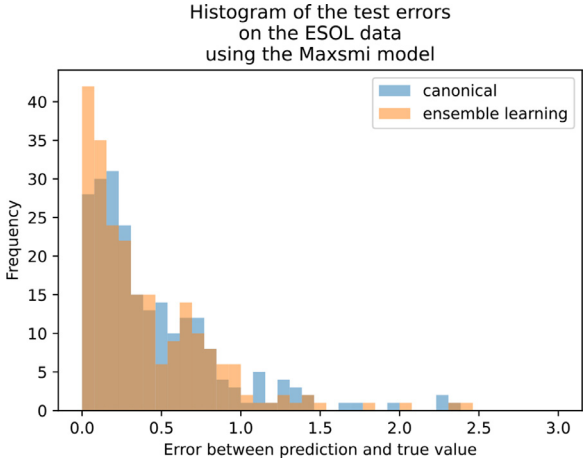

Fig. 6. Lower errors when evaluating the Maxsmi model using ensemble learning. There are fewer errors in the evaluation of the trained Maxsmi models when using ensemble learning (i.e. the averaged per SMILES prediction) vs. the canonical prediction.

relationship between high confidence and small prediction error on the test set for the Maxsmi models. A way of visually evaluating uncertainty is to plot the confidence curve [32], which displays how the error varies with the sequential removal of compounds from lowest to highest confidence. Fig. 7 shows the confidence curve of the Maxsmi model used on the FreeSolv data. As shown in the figure, as molecules with low confidence are sequentially removed, the mean prediction error decreases. In other words, the error vanishes as only compounds with the highest certainty predictions are kept, demonstrating a relationship between high confidence and small prediction error. Fig. A7 shows the confidence curves of the Maxsmi models for the ESOL and lipophilicity data. The general trend of the curve is decreasing in the ESOL case. Once the 10% of compounds with the highest confidence are kept, the error is below 0.25. However, in the lipophilicity case, although the general trend is also decreasing, even when keeping the 10% of compounds with the highest confidence, the error is still above 0.3.

4.3. Comparison to other studies

Given the results of the Maxsmi models, their performance is compared to other studies, namely MoleculeNet [34], CNF [22], and MolPMoFiT [21], that are trained and evaluated on the same data sets as Maxsmi, see Table 4.

Table 4
The Maxsmi models reach state-of-the-art results. Comparison of four studies on the same data sets (ESOL, FreeSolv, and lipophilicity). The Maxsmi model outperforms most of the other models with a lower RMSE on a randomly split test set.

Study	Test RMSE (± std if available)			Split			Model
	ESOL	FreeSolv	Lipophilicity	Fold	Ratio % (train:valid:test)	Type	
Maxsmi	**0.569**	**1.032**	0.593	Single	80 : 0 : 20	Random	CNN
MoleculeNet[34]	0.58 ± 0.03	1.15 ± 0.12	0.655 ± 0.036	3	80 : 10 : 10	Random	GNN
CNF[22]	0.62	1.11	0.67	5-fold CV	NA	Random	CNN
MolPMoFiT[21]	NA	1.197 ± 0.127	$\mathbf{0.565 \pm 0.037}$	10	80 : 10 : 10	Random	RNN

Abbreviations: RMSE = root mean squared error, std = standard deviation, CNN = Convolutional Neural Network, GNN = Graph Neural Network, RNN = Recurrent Neural Network, NA = not available, CV = cross-validation.

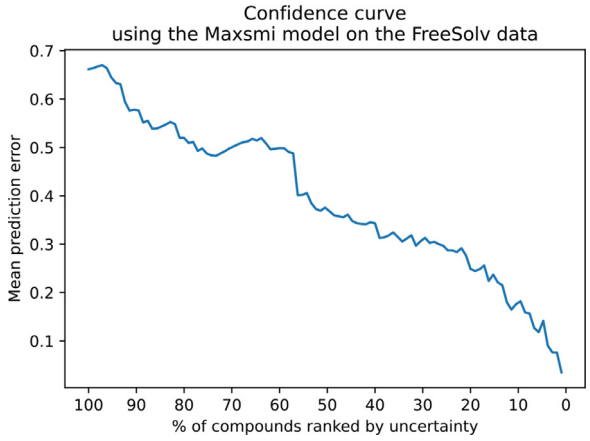

Fig. 7. More confident Maxsmi model on FreeSolv implies smaller prediction error. The general trend of the confidence curve is decreasing, showing that as compounds with high uncertainty are removed, the error becomes smaller.

The first considered study is MoleculeNet, where several molecular encodings and models are trained and evaluated, but where no augmentation is used. In MoleculeNet, the data is randomly split into training, validation, and test set, using an 80 : 10 : 10 ratio and run three times on different seeds. The best performing model on the test set for both ESOL and FreeSolv is a message passing neural network with an RMSE and standard deviation of 0.58 ± 0.03 and 1.15 ± 0.12, respectively [34, Table S5]. On the lipophilicity data set, a slightly different graph model performs best with 0.655 ± 0.036. On all three tasks, the Maxsmi results perform better than MoleculeNet (see Table 4), suggesting that SMILES augmentation with shallow neural networks could perform at least as well as, if not better than, graph neural networks (GNNs).

The second study we consider is the Convolutional Neural Fingerprint (CNF) model [22,23], in which SMILES augmentation is applied, generating unique representations for each compound, i.e. augmentation without duplication. The CNF model is evaluated using five-fold cross-validation (CV), however standard deviations are not reported. Test RMSE values are 0.62, 1.11 and 0.66 on the ESOL, FreeSolv, and lipophilicity data set, respectively, see [22, Table S1]. Similar to MoleculeNet, the Maxsmi model slightly outperforms these results on all three tasks. This suggests that augmenting the data set by greater factors, e.g. closer to 70 as in Maxsmi, yields better results than 10 times augmentation as in CNF.

Finally, the Molecular Prediction Model Fine-Tuning (MolPMoFiT) [21] study builds an RNN model based on LSTM layers using SMILES augmentation with duplication. The lipophilicity data is augmented 25 times whereas the FreeSolv data 50 times. MolPMoFiT is trained and evaluated using 10 splits of ratio 80 : 10 : 10 for the training, validation, and test sets. The model reaches an RMSE value (and standard deviation) of 1.197 ± 0.127 on the FreeSolv data and 0.565 ± 0.037 on the lipophilicity data, see [21, Figure 3, 4]. While Maxsmi leads on the FreeSolv prediction problem, MolPMoFiT slightly outperforms Maxsmi on the lipophilicity data (Table 4).

Lastly, study comparison should be treated with utmost attention, since results can not be compared blindly. For instance, if the data preprocessing is done differently in each study, or the splits are not identical, or the parameters of the experiments are not set to be the same, then the results are not fairly comparable.

4.4. Test case: EGFR affinity data following the guideline

Given the results of the Maxsmi models on the physico-chemical data sets, we now discuss guidelines to apply SMILES augmentation on a new data set. The EGFR affinity data, discussed in the Provenance section and henceforth simply referred to as affinity data, is used as a test case, but the idea can be applied to different data sets and broader use cases.

Since the affinity data set contains 5 849 data points after preprocessing (see Table 1), the lipophilicity and affinity data are of a similar order of magnitude, although the latter is somewhat larger. Therefore, a compromise between the size of the data set and the tuple that gives the best results for the FreeSolv, ESOL, and lipophilicity data (see Table 2) is found: for the affinity data, the CONV1D model is chosen (similarly to lipophilicity, ESOL, and FreeSolv), the number of augmentation is set to 70, as for ESOL and FreeSolv, and the augmentation strategy is set to augmentation with reduced duplication for a less computational intensive training than augmentation with duplication. As comparison, the Maxsmi, the canonical, and the baseline models on affinity are trained and evaluated. The same experimental setup for splitting and evaluation as mentioned in the Data and experimental setup section is applied. On the test set, the canonical model reaches an RMSE value and coefficient of correlation $R2$ value of 1.031 and 0.494, respectively. In comparison, the Maxsmi model shows great improvement with test RMSE and $R2$ values of 0.777 and 0.712, respectively (see Table 5). Surprisingly,

Table 5
The Maxsmi models strike again! The Maxsmi model developed for the affinity against the EGFR kinase and the Random Forest (RF) baseline model outperform the canonical model.

Name	Model	Augmentation number	Augmentation strategy	Test RMSE	Test $R2$
Maxsmi	CONV1D	70	Augmentation with reduced duplication	0.777	0.712
Canonical	CONV1D	0	No augmentation	1.031	0.494
Baseline	RF	0	No augmentation	0.758	0.726

the RF baseline model performs similarly to the Maxsmi model, with an RMSE of 0.758 and an $R2$ of 0.726.

4.5. Maxsmi models available for user predictions

Given the good performance of the Maxsmi models for all three physico-chemical tasks and for EGFR affinity, we retrained them on all points in the data set as a final product. The aim is to offer a single command-line interface for prediction. A user can provide a SMILES as input, choose a given task and they will receive an output file in the form of a CSV table with relevant information, such as 1. the user input SMILES itself, 2. whether the compound was in the training set, 3. the canonical SMILES, and its associated variations 4. the per SMILES predictions, 5. the per compound prediction, 6. and the standard deviation.

A PNG file of the 2D molecular graph associated with the input SMILES is also generated. For example, for the semaxanib drug, taken from the PKIDB database [63], and given by the SMILES `O=C2C(\1ccccc1N2)=C/c3c(cc([nH]3)C)C`, lipophilicity is predicted using the command-line

```
$ python maxsmi/prediction_unlabeled_data.py
--task="lipophilicity" --smiles_prediction="O=
C2C(\1ccccc1N2)=C/c3c(cc([nH]3)C)C"
```

The command above was used to generate the values in Fig. 1.

5. Conclusion

In this study, SMILES augmentation applied to deep learning molecular property and activity prediction is investigated. Five augmentation strategies that can be applied as SMILES augmentation are explored, together with three neural network architectures, and the performance thoroughly assessed on three molecular data sets: ESOL, FreeSolv, and lipophilicity.

Our findings show that augmentation improves the performance of deep learning models not only independently of the model, but also with respect to the size of the data set. This suggests that the choice of augmentation strategy can be viewed as hyper-parameter tuning.

The tuple consisting of (model, augmentation number, augmentation strategy) that maximizes the performance on the test set leads to the definition of the Maxsmi models. Our findings also show that the model using canonical SMILES outperforms the one using single random SMILES, thanks to the simplicity of the canonical notation.

Additionally, the Maxsmi models outperform, or perform at least as well as state-of-the-art models such as MoleculeNet, CNF, and MolPMoFiT, on the three physico-chemical data sets. This suggests that applying simple SMILES augmentation techniques can reach similar or even better performance as sophisticated models such as graph-based neural networks, as in the case of MoleculeNet. Moreover, we use our findings to guide the application of SMILES augmentation on a new data set and provide a test case with data on affinity against the EGFR kinase. Finally, we provide an easy to use framework for out-of-sample prediction on four tasks: ESOL, FreeSolv, lipophilicity, and affinity against EGFR, which should be helpful to assess properties of novel compounds. The open-source code allows to perform similar studies on different data sets with minor programmatic adjustments.

As an outlook, we observe that strategies that keep all, or a fraction of duplicates, may help the model to learn inherent symmetry in a compound. Indeed the same random SMILES representation will certainly be generated multiple times for a symmetric molecule even though the initial atom and the path along the graph are different. In this sense, SMILES duplication is not an artificial construction, and keeping replicas could retain important information about the underlying symmetry of a compound.

6. Abbreviations

List of abbreviations used in the paper.

QSAR	Quantitative Structure-Activity Relationship
SMILES	Simplified Molecular-Input Line-Entry System
CADD	Computer-Aided Drug Design
ML	Machine Learning
DL	Deep Learning
LSTM	Long Short-Term Memory
EGFR	Epidermal Growth Factor Receptor
NA	Not Available
RMSE	Root Mean Squared Error
CONV1D	1D Convolutional Neural Network
CONV2D	2D Convolutional Neural Network
RNN	Recurrent Neural Network
RF	Random Forest
GNN	Graph Neural Network
CNF	Convolutional Neural Fingerprint
CV	Cross-validation
MolPMoFiT	Molecular Prediction Model Fine-Tuning

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the HPC Service of ZEDAT, Freie Universität Berlin, for computing time. Talia B. Kimber received funding from the Stiftung Charité in the context of the Einstein BIH Visiting Fellow Project. The authors thank Greg Landrum for beneficial discussions, as well as Dominique Sydow for valuable advice on documentation.

Appendix

Figures

Figures in the appendix.

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	2.577	2.868	2.514	2.603	2.881	2.396	2.691	3.017	2.431
2	2.108	2.222	1.989	2.021	2.103	2.122	2.091	1.979	2.002
3	1.914	2.315	1.900	1.956	1.967	1.810	1.917	2.371	2.029
4	1.494	1.646	1.545	1.444	1.692	1.479	1.420	1.594	1.568
5	1.260	1.524	1.381	1.342	1.517	1.332	1.278	1.478	1.469
6	1.306	1.580	1.398	1.277	1.723	1.352	1.306	1.656	1.358
7	1.416	1.745	1.612	1.365	1.494	1.446	1.415	1.559	1.398
8	1.258	1.446	1.273	1.282	1.472	1.288	1.238	1.375	1.259
9	1.325	1.578	1.238	1.324	1.598	1.240	1.287	1.498	1.253
10	1.190	1.485	1.303	1.231	1.517	1.290	1.211	1.544	1.229
11	1.225	1.434	1.331	1.195	1.519	1.224	1.238	1.410	1.283
12	1.181	1.393	1.422	1.198	1.438	1.273	1.196	1.449	1.199
13	1.308	1.522	1.392	1.146	1.584	1.327	1.199	1.557	1.266
14	1.162	1.483	1.240	1.197	1.501	1.279	1.149	1.501	1.477
15	1.169	1.516	1.237	1.160	1.545	1.202	1.188	1.441	1.223
16	1.138	1.306	1.256	1.096	1.488	1.157	1.102	1.397	1.250
17	1.147	1.354	1.321	1.162	1.470	1.231	1.117	1.501	1.295
18	1.139	1.504	1.260	1.141	1.494	1.569	1.099	1.572	1.213
19	1.164	1.475	1.244	1.159	1.561	1.531	1.129	1.558	1.272
20	1.139	1.523	1.330	1.085	1.449	1.274	1.109	1.502	1.199
30	1.083	1.374	1.299	1.120	1.553	1.349	1.100	1.492	1.431
40	1.140	1.525	1.311	1.051	1.700	1.159	1.079	1.605	1.187
50	1.091	1.579	1.291	1.053	1.452	1.250	1.076	1.437	1.174
60	1.074	1.554	1.264	1.094	1.427	1.226	1.104	1.522	1.284
70	1.085	1.679	1.239	1.032	1.476	1.109	1.090	1.453	1.254
80	1.125	1.639	1.270	1.070	1.449	1.341	1.153	1.523	1.176
90	1.101	1.506	1.280	1.058	1.556	1.243	1.065	1.520	1.323
100	1.115	1.631	1.276	1.056	1.598	1.281	1.065	1.635	1.184

no augmentation			estimated maximum			baseline
CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	RF
1.963	2.090	2.373	1.124	1.585	1.289	2.563

Fig. A1. Test RMSE using data augmentation on the FreeSolv data set. The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the FreeSolv data set is augmenting the data set 70 times keeping all duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	1.268	1.286	1.144	1.208	1.309	1.145	1.211	1.260	1.144
2	1.042	1.115	1.073	1.048	1.140	1.078	1.033	1.126	1.083
3	0.933	1.004	1.077	0.949	1.010	1.015	0.921	1.016	1.026
4	0.874	0.956	0.918	0.886	0.967	0.942	0.859	0.921	0.967
5	0.861	0.944	0.931	0.849	0.939	0.908	0.857	0.961	0.920
6	0.815	0.903	0.862	0.804	0.911	0.875	0.815	0.890	0.881
7	0.783	0.886	0.828	0.807	0.871	0.848	0.797	0.892	0.848
8	0.782	0.880	0.785	0.787	0.871	0.797	0.779	0.864	0.838
9	0.769	0.846	0.793	0.783	0.854	0.785	0.791	0.852	0.795
10	0.755	0.852	0.781	0.751	0.844	0.789	0.752	0.839	0.797
11	0.754	0.832	0.745	0.737	0.838	0.744	0.738	0.837	0.745
12	0.730	0.837	0.737	0.724	0.816	0.746	0.721	0.843	0.753
13	0.717	0.811	0.723	0.722	0.829	0.719	0.725	0.816	0.730
14	0.715	0.818	0.721	0.713	0.819	0.772	0.716	0.808	0.715
15	0.712	0.808	0.724	0.707	0.804	0.715	0.716	0.810	0.701
16	0.704	0.804	0.694	0.702	0.789	0.720	0.711	0.798	0.701
17	0.706	0.808	0.707	0.699	0.791	0.686	0.709	0.797	0.698
18	0.686	0.807	0.679	0.691	0.801	0.690	0.678	0.791	0.699
19	0.681	0.809	0.703	0.672	0.785	0.700	0.686	0.793	0.688
20	0.671	0.787	0.673	0.657	0.777	0.672	0.671	0.779	0.673
30	0.640	0.752	0.680	0.645	0.747	0.684	0.655	0.754	0.679
40	0.638	0.723	0.660	0.637	0.742	0.662	0.634	0.746	0.651
50	0.614	0.744	0.691	0.623	0.727	0.692	0.617	0.724	0.680
60	0.604	0.735	0.671	0.606	0.725	0.654	0.609	0.738	0.723
70	0.597	0.717	0.670	0.602	0.716	0.673	0.608	0.737	0.692
80	0.593	0.723	0.687	0.603	0.720	0.695	0.606	0.718	0.687
90	0.606	0.735	0.694	0.609	0.730	0.649	0.600	0.718	0.693
100	0.596	0.715	0.706	0.600	0.715	0.688	0.596	0.723	0.720

no augmentation			baseline
CONV1D	CONV2D	RNN	RF
0.994	1.060	1.023	0.860

Fig. A2. Test RMSE using data augmentation on the lipophilicity data set. The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the lipophilicity data set is augmenting the data set 80 times removing duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

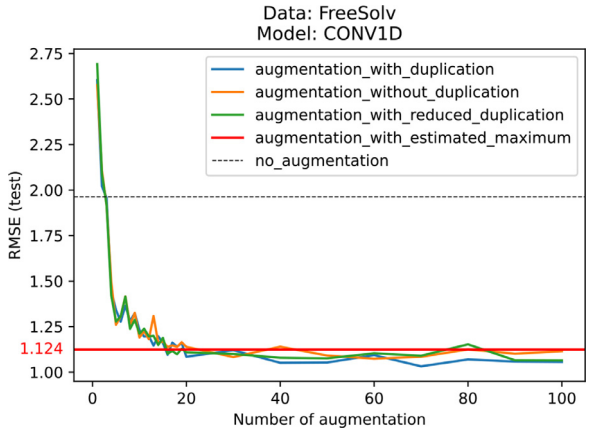

Fig. A3. Generating a large portion of the SMILES space does not lead to the best performance. Even though the CONV1D model is presented with SMILES variations that cover a large portion of the SMILES space using the augmentations strategy with estimated maximum, on the FreeSolv data set, this strategy does not achieve the best results.

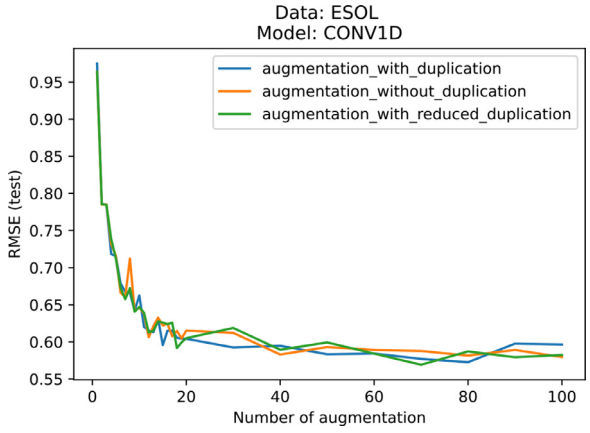

Fig. A4. Performance reaches a plateau independently of the augmentation strategy. The performance of the CONV1D model trained and evaluated on the ESOL data set reaches a test RMSE value slightly below 0.6 as of 40 augmentation steps and fluctuates below this value thereafter, for all augmentation strategies: with, without, and with reduced duplication.

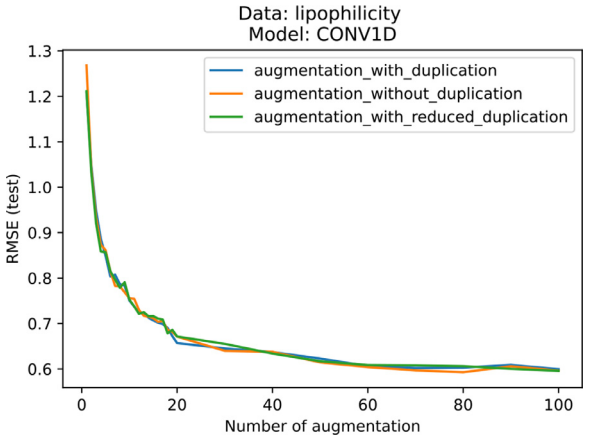

Fig. A5. Performance reaches a plateau independently of the augmentation strategy. The performance of the CONV1D model trained and evaluated on the lipophilicity data set reaches a test RMSE value slightly below 0.6 as of 60 augmentation steps and fluctuates below this value thereafter, for all augmentation strategies: with, without, and with reduced duplication.

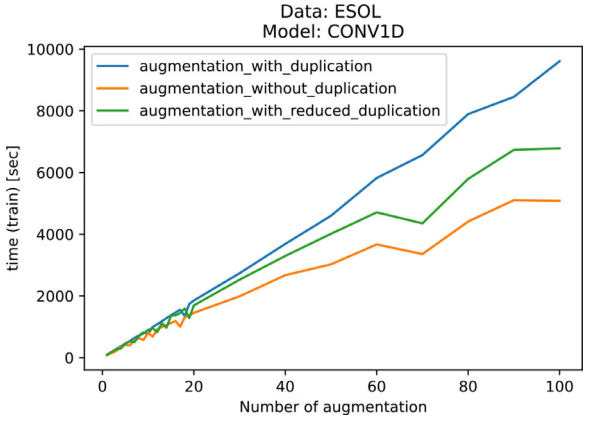

Fig. A6. Trade-off between performance and computation time. As expected, the training time of the CONV1D model on the ESOL data increases with the augmentation number for all augmentation strategies: with, without, and with reduced duplication. Augmenting the training set by 100 and keeping duplicate leads to 90 200 data points (see Table 1). Training the model on a GPU takes approximately three hours and reaches a test RMSE of 0.580 (see Figure 2). However, augmenting the data by just 19 leads to a test RMSE of 0.605 in less than 30 minutes.

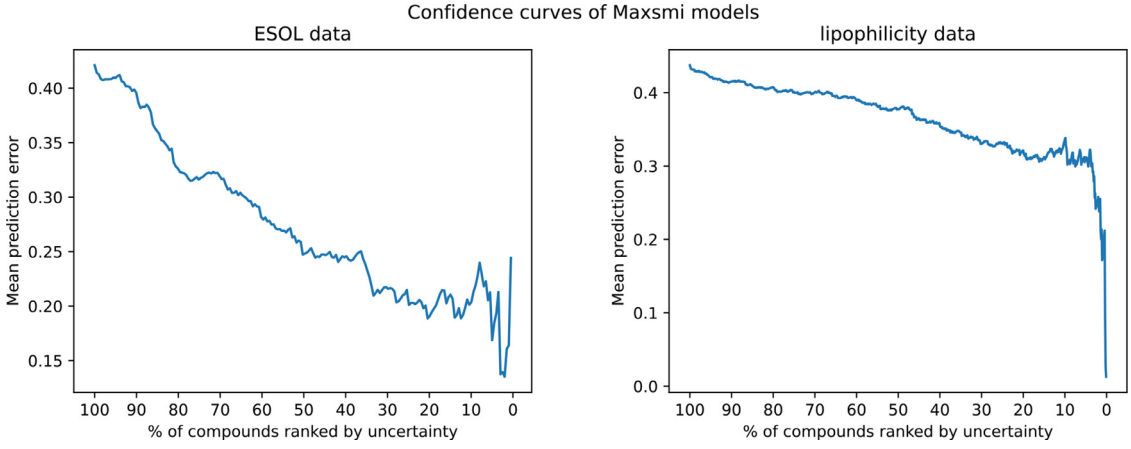

Fig. A7. Confidence curves of the Maxsmi models on the ESOL and lipophilicity data. The general trend of the curve in the left plot (the ESOL data) is decreasing, showing a relationship between high confidence and small mean prediction error. Although also generally decreasing, the mean prediction error in the right plot (the lipophilicity data) is still above 0.3 when only keeping the 10% of compounds with the highest confidence.

References

[1] Paul SM, Mytelka DS, Dunwiddie CT, Persinger CC, Munos BH, Lindborg SR, et al. How to improve r&d productivity: the pharmaceutical industry's grand challenge. Nat Rev Drug Discovery 2010;9(3):203–14. doi:10.1038/nrd3078.

[2] Scannell JW, Blanckley A, Boldon H, Warrington B. Diagnosing the decline in pharmaceutical r&d efficiency. Nat Rev Drug Discovery 2012;11(3):191–200. doi:10.1038/nrd3681.

[3] Waring MJ, Arrowsmith J, Leach AR, Leeson PD, Mandrell S, Owen RM, et al. An analysis of the attrition of drug candidates from four major pharmaceutical companies. Nat Rev Drug Discovery 2015;14(7):475–86. doi:10.1038/nrd4609.

[4] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Commun ACM 2017;60(6):84–90. doi:10.1145/3065386.

[5] Graves A, Mohamed A-r, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; 2013. p. 6645–9. doi:10.1109/ICASSP.2013.6638947.

[6] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with alphafold. Nature 2021;596:583–9. doi:10.1038/s41586-021-03819-2.

[7] Feinberg EN, Sur D, Wu Z, Husic BE, Mai H, Li Y, et al. Potentialnet for molecular property prediction. ACS Cent Sci 2018;4(11):1520–30. doi:10.1021/acscentsci.8b00507.

[8] Sattarov B, Baskin II, Horvath D, Marcou G, Bjerrum EJ, Varnek A. De novo molecular design by combining deep autoencoder recurrent neural networks with generative topographic mapping. J Chem Inf Model 2019;59(3):1182–96. doi:10.1021/acs.jcim.8b00751.

[9] Webel HE, Kimber TB, Radetzki S, Neuenschwander M, Nazaré M, Volkamer A. Revealing cytotoxic substructures in molecules using deep learning. J Comput Aided Mol Des 2020;34(7):731–46. doi:10.1007/s10822-020-00310-4.

[10] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016. http://www.deeplearningbook.org

[11] Bisong E. Google Colaboratory. Berkeley, CA: Apress; 2019. p. 59–64. ISBN 978-1-4842-4470-8

[12] Kaggle. https://www.kaggle.com/; 2021. [Online; accessed 27-August-2021].

[13] Parks CD, Gaieb Z, Chiu M, Yang H, Shao C, Walters WP, et al. D3R Grand challenge 4: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. J Comput Aided Mol Des 2020;34(2):99–119. doi:10.1007/s10822-020-00289-y.

[14] Huang R, Xia M. Editorial: Tox21 challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental toxicants and drugs. Front Environ Sci 2017;5:3. doi:10.3389/fenvs.2017.00003.

[15] Weininger D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. J Chem Inf Comput Sci 1988;28(1):31–6. doi:10.1021/ci00057a005.

[16] O'Boyle NM. Towards a universal SMILES representation - a standard method to generate canonical SMILES based on the inchi. J Cheminform 2012;4(1). doi:10.1186/1758-2946-4-22.

[17] Weininger D, Weininger A, Weininger JL. Smiles. 2. algorithm for generation of unique smiles notation. J Chem Inf Comput Sci 1989;29(2):97–101. doi:10.1021/ci00062a008.

[18] Hemmerich J, Asilar E, Ecker GF. COVER: Conformational oversampling as data augmentation for molecules. J Cheminform 2020;12(1). doi:10.1186/s13321-020-00420-z.

[19] Li Y, Rezaei MA, Li C, Li X. Deepatom: A framework for protein-ligand binding affinity prediction. In: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2019. p. 303–10. doi:10.1109/BIBM47256.2019.8982964.

[20] Bjerrum EJ. Smiles enumeration as data augmentation for neural network modeling of molecules. arXiv preprint arXiv:170307076 2017. https://arxiv.org/abs/1703.07076

[21] Li X, Fourches D. Inductive transfer learning for molecular activity prediction: next-gen QSAR models with molpmofit. J Cheminform 2020;12(1). doi:10.1186/s13321-020-00430-x.

[22] Kimber TB, Engelke S, Tetko IV, Bruno E, Godin G. Synergy effect between convolutional neural networks and the multiplicity of smiles for improvement of molecular prediction. arXiv preprint arXiv:181204439 2018. https://arxiv.org/abs/1812.04439

[23] Tetko IV, Karpov P, Bruno E, Kimber TB, Godin G. Augmentation is what you need!. In: Tetko IV, Kůrková V, Karpov P, Theis F, editors. Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions. Cham: Springer International Publishing; 2019. p. 831–5. doi:10.1007/978-3-030-30493-5_79. ISBN 978-3-030-30493-5

[24] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. J Big Data 2019;6(1). doi:10.1186/s40537-019-0197-0.

[25] Tetko IV, Karpov P, Deursen RV, Godin G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. Nat Commun 2020;11(1). doi:10.1038/s41467-020-19266-y.

[26] Sumner D, He J, Thakkar A, Engkvist O, Bjerrum EJ. Levenshtein augmentation improves performance of smiles based deep-learning synthesis prediction. ChemRxiv 2020. doi:10.26434/chemrxiv.12562121.v2.

[27] Arús-Pous J, Johansson SV, Prykhodko O, Bjerrum EJ, Tyrchan C, Reymond J-L, et al. Randomized SMILES strings improve the quality of molecular generative models. J Cheminform 2019;11(1). doi:10.1186/s13321-019-0393-0.

[28] van Deursen R, Ertl P, Tetko IV, Godin G. GEN: Highly efficient SMILES explorer using autodidactic generative examination networks. J Cheminform 2020;12(1). doi:10.1186/s13321-020-00425-8.

[29] RDKit: Open-source cheminformatics. http://www.rdkit.org [Online; accessed 01-July-2021]; 2021.

[30] Kazil J, Jarmul K. Data wrangling with python: tips and tools to make your life easier. 1st. O'Reilly Media, Inc.; 2016. ISBN 1491948817

[31] Tagasovska N, Lopez-Paz D. Single-model uncertainties for deep learning. arXiv preprint arXiv:181100908 2019. https://arxiv.org/abs/1811.00908

[32] Scalia G, Grambow CA, Pernici B, Li Y-P, Green WH. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. J Chem Inf Model 2020;60(6):2697–717. doi:10.1021/acs.jcim.9b00975.

[33] Ayhan M.S., Berens P. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. 2018. https://openreview.net/pdf?id=rJZz-knjz.

[34] Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V. Moleculenet: a benchmark for molecular machine learning. Chem Sci 2018;9:513–30. doi:10.1039/C7SC02664A.

[35] Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, et al. The ChEMBL database in 2017. Nucleic Acids Res 2016;45(D1):D945–54. doi:10.1093/nar/gkw1074.

[36] Mayr A, Klambauer G, Unterthiner T, Steijaert M, Wegner JK, Ceulemans H, Clevert D-A, Hochreiter S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. Chem Sci 2018;9:5441–51. doi:10.1039/C8SC00148K.

[37] Zhang Y, Lee AA. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. Chem Sci 2019;10:8154–63. doi:10.1039/C9SC00616H.

[38] Ramsundar B, Eastman P, Walters P, Pande V, Leswing K, Wu Z. Deep learning for the life sciences. O'Reilly Media; 2019. ISBN 9781492039839

[39] Delaney JS. Esol: estimating aqueous solubility directly from molecular structure. J Chem Inf Comput Sci 2004;44(3):1000–5. doi:10.1021/ci034243x.

[40] Mobley DL, Guthrie JP. FreeSolv: a database of experimental and calculated hydration free energies, with input files. J Comput Aided Mol Des 2014;28(7):711–20. doi:10.1007/s10822-014-9747-x.

[41] ChEMBL. https://www.ebi.ac.uk/chembl/ [Online; accessed 27-August-2021]; 2021.

[42] Kooistra AJ, Volkamer A. Kinase-centric computational drug development. In: Annual Reports in Medicinal Chemistry. Elsevier; 2017. p. 197–236. doi:10.1016/bs.armc.2017.08.001.

[43] OpenKinome. http://openkinome.org/ [Online; accessed 27-August-2021]; 2021.

[44] Herbst RS. Review of epidermal growth factor receptor biology. International Journal of Radiation Oncology*Biology*Physics 2004;59(2, Supplement):S21–6. doi:10.1016/j.ijrobp.2003.11.041.

[45] Consortium TU. Uniprot: the universal protein knowledgebase in 2021. Nucleic Acids Res 2020;49(D1):D480–9. doi:10.1093/nar/gkaa1100.

[46] IC50 Values. Offermanns S, Rosenthal W, editors. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. doi:10.1007/978-3-540-38918-7_5943. ISBN 978-3-540-38918-7

[47] Kimber TB, Chen Y, Volkamer A. Deep learning in virtual screening: recent applications and developments. Int J Mol Sci 2021;22(9). doi:10.3390/ijms22094435.

[48] Mean squared error. Sammut C, Webb GI, editors. Boston, MA: Springer US; 2010. doi:10.1007/978-0-387-30164-8_528. ISBN 978-0-387-30164-8

[49] Kvålseth TO. Cautionary note about r^2. Am Stat 1985;39(4):279–85. doi:10.1080/00031305.1985.10479448.

[50] Breiman L. Random forests. Mach Learn 2001;45(1):5–32. doi:10.1023/A:1010933404324.

[51] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, a Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. Journal of Machine Learning Research 2011;12:2825–30. http://jmlr.org/papers/v12/pedregosa11a.html

[52] Bennett L., Melchers B., Proppe B. Curta: A general-purpose high-performance computer at Zedat, Freie Universität Berlin. 2020. https://doi.org/10.17169/refubium-26754.

[53] Van Rossum G, Drake FL. Python 3 reference manual. Scotts Valley, CA: CreateSpace; 2009. ISBN 1441412697

[54] van Rossum G, Warsaw B, Coghlan N. Style guide for Python code. PEP; 2001. https://www.python.org/dev/peps/pep-0008/

[55] Read the Docs. https://readthedocs.io/en/stable/ [Online; accessed 30-July-2021]; 2021.

[56] Anaconda software distribution. 2020. https://anaconda.com/.

[57] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems 32. Curran Associates, Inc.; 2019. p. 8024–35. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[58] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with numpy. Nature 2020;585(7825):357–62. doi:10.1038/s41586-020-2649-2.

[59] The pandas development team. pandas-dev/pandas: Pandas. 2020. 10.5281/zenodo.3509134

[60] GitHub Actions. https://docs.github.com/en/actions [Online; accessed 30-July-2021]; 2021.

[61] Pytest. https://docs.pytest.org/ [Online; accessed 30-July-2021]; 2021.

[62] Codecov. https://docs.codecov.com/docs [Online; accessed 30-July-2021]; 2021.

[63] Carles F, Bourg S, Meyer C, Bonnet P. PKIDB: a curated, annotated and updated database of protein kinase inhibitors in clinical trials. Molecules 2018;23(4). doi:10.3390/molecules23040908.