Journal of
CHEMICAL INFORMATION
AND MODELING

Article

Cite This: *J. Chem. Inf. Model.* 2018, 58, 27−35

pubs.acs.org/jcim

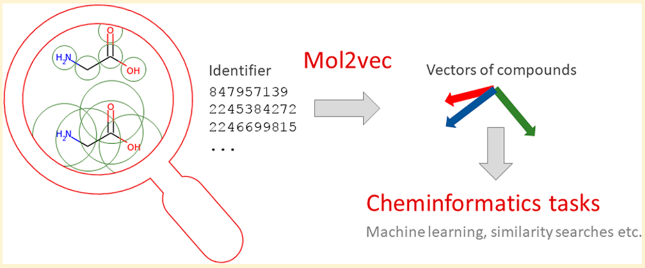
Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition

Sabrina Jaeger, Simone Fulle,* and Samo Turk*

BioMed X Innovation Center, Im Neuenheimer Feld 515, 69120 Heidelberg, Germany

Supporting Information

ABSTRACT: Inspired by natural language processing techniques, we here introduce Mol2vec, which is an unsupervised machine learning approach to learn vector representations of molecular substructures. Like the Word2vec models, where vectors of closely related words are in close proximity in the vector space, Mol2vec learns vector representations of molecular substructures that point in similar directions for chemically related substructures. Compounds can finally be encoded as vectors by summing the vectors of the individual substructures and, for instance, be fed into supervised machine learning approaches to predict compound properties. The underlying substructure vector embeddings are obtained by training an unsupervised machine learning approach on a so-called corpus of compounds that consists of all available chemical matter. The resulting Mol2vec model is pretrained once, yields dense vector representations, and overcomes drawbacks of common compound feature representations such as sparseness and bit collisions. The prediction capabilities are demonstrated on several compound property and bioactivity data sets and compared with results obtained for Morgan fingerprints as a reference compound representation. Mol2vec can be easily combined with ProtVec, which employs the same Word2vec concept on protein sequences, resulting in a proteochemometric approach that is alignment-independent and thus can also be easily used for proteins with low sequence similarities.



INTRODUCTION

As numeric representation of molecules is an essential part of cheminformatics, a variety of descriptors and molecular fingerprints (FP) exist that are either fed into machine learning (ML) models or form the basis for similarity searching and clustering approaches. The most commonly used representations include Morgan FPs (also known as extended-connectivity fingerprints (ECFPs)),[1] as they often outperform other types of FPs in similarity search and virtual screening tasks[2,3] and have also been successfully used for molecular activity predictions.[4−7] To generate a Morgan FP, all substructures around all heavy atoms of a molecule within a defined radius are generated and assigned to unique identifiers (called Morgan identifiers below). These identifiers are then usually hashed to a vector with a fixed length. However, the vectors obtained are not only very high dimensional and sparse but also might contain bit collisions introduced by the hashing step.

The recent rise in popularity of artificial neural networks brought several breakthroughs in ML, and ideas from various fields of data science are also spilling over to cheminformatics. Convolutional neural networks, originally developed for image recognition, were successfully applied on molecular graphs[8,9] and on 2D depictions of molecules.[10] In parallel, natural language processing (NLP) techniques were adopted to learn from classical features like molecular descriptors,[11] molecular FPs,[12] SMILES strings,[13] and graph representations of compounds.[8] Most worth noting, the NLP method “term frequency−inverse document frequency” (tf-idf) was applied on Morgan fingerprints for compound−protein prediction[12] and the “latent Dirichlet allocation” method for chemical topic modeling.[14] Another popular NLP approach is Word2vec,[15] which learns high-dimensional embeddings of words where vectors of similar words end up near in vector space. This concept has already been adapted to protein sequences (ProtVec) for the classification of protein families and disordered proteins[16] but has not been applied to molecules to date.

Here we introduce Mol2vec, which is an NLP-inspired technique that considers compound substructures derived from the Morgan algorithm as “words” and compounds as “sentences”. By application of the Word2vec algorithm on a corpus of compounds, high-dimensional embeddings of substructures are obtained, where the vectors for chemically related substructures occupy the same part of vector space. Mol2vec is an unsupervised method that is initially trained on unlabeled data to obtain feature vectors of substructures, which can be summed to obtain compound vectors. It should be noted that while the generation of a Mol2vec model is an unsupervised pretraining step, subsequent machine learning models for property predictions are supervised throughout this

Received: October 20, 2017
Published: December 22, 2017

DOI: 10.1021/acs.jcim.7b00616

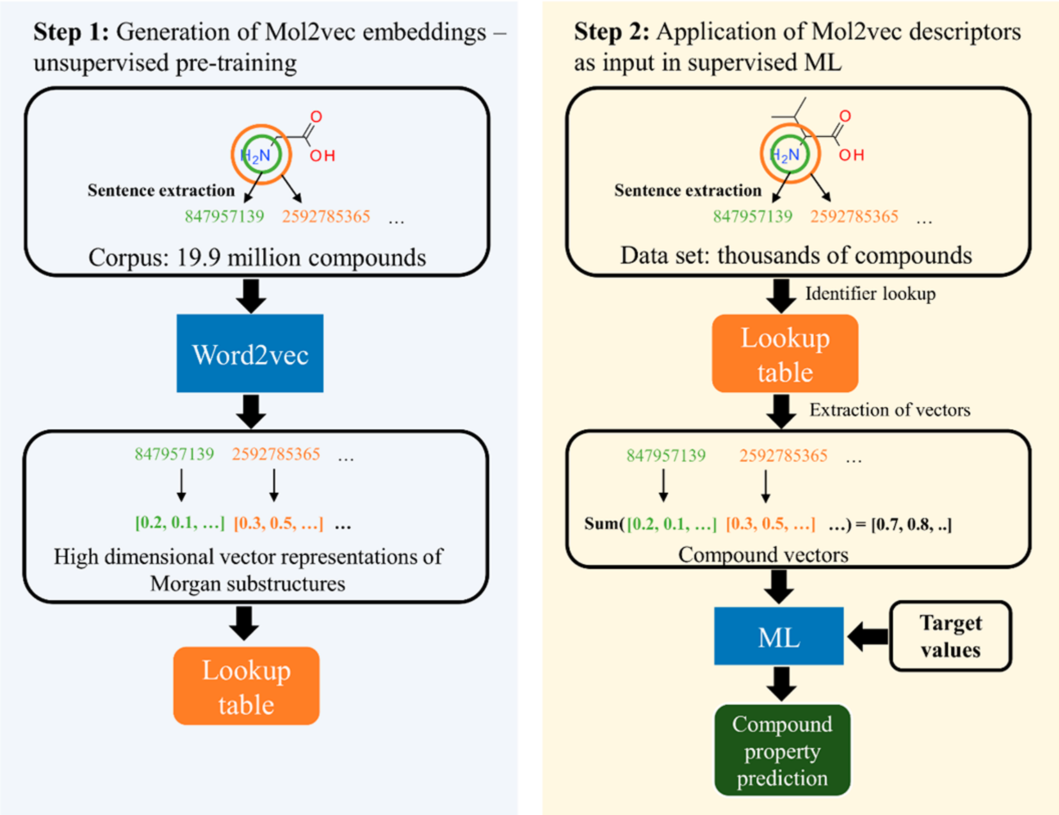

Step 1: Generation of Mol2vec embeddings – unsupervised pre-training
Sentence extraction
847957139 2592785365 …
Corpus: 19.9 million compounds
Word2vec
847957139 2592785365 …
[0.2, 0.1, …] [0.3, 0.5, …] …
High dimensional vector representations of Morgan substructures
Lookup table
Step 2: Application of Mol2vec descriptors as input in supervised ML
Sentence extraction
847957139 2592785365 …
Data set: thousands of compounds
Identifier lookup
Lookup table
Extraction of vectors
847957139 2592785365 …
Sum([0.2, 0.1, …] [0.3, 0.5, …] …) = [0.7, 0.8, ..]
Compound vectors
ML
Target values
Compound property prediction

Figure 1. Overview of the generation and usage steps of Mol2vec. Step 1: Mol2vec embeddings (i.e., vector representations of substructures) are generated via an unsupervised pretraining step. Step 2: Application of Mol2vec vectors requires that substructure vectors be retrieved and summed to obtain compound vectors, which can finally be used to train a supervised prediction model.

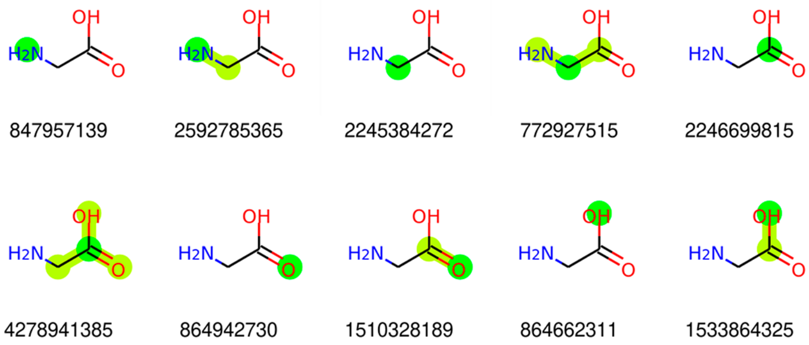


Figure 2. Depiction of identifiers obtained with the Morgan algorithm on the structure of glycine forming a molecular sentence. Identifiers are ordered in the same order as the atoms in the canonical SMILES representation for consistency reasons. If an atom has more than one identifier, the first identifier for that atom is the one for radius 0, followed by radius 1, etc.

article (Figure 1). Questions addressed below are how Mol2vec performs on different compound data sets, on regression and classification problems, and, combined with the ProtVec representation for proteins in proteochemometric (PCM) approaches, on proteins with different sequence similarity ranges.

■ METHODS

Mol2vec and ProtVec are unsupervised pretraining methods that can be used to obtain high-dimensional embeddings of molecular substructures or *n*-grams of protein sequences (i.e., they provide featurization of compounds and proteins). These vectors can then be further used in supervised ML tasks. In this section, we first describe the data sets used for pretraining of the Mol2vec and ProtVec models and the pretraining itself, followed by the data sets used for the evaluation of Mol2vec in supervised tasks and the machine learning methods employed for property predictions.

Pretraining Compound Data Set. The corpus of compounds was composed using the ZINC version 15[17] and ChEMBL version 23[18,19] databases as sources of compounds. The two databases were merged, and duplicates were removed; only compounds that could be processed by RDKit[20] (release 2017.03.3) were kept, and they were filtered using the following cutoffs and criteria: molecular weight between 12 and 600, heavy-atom count between 3 and 50, clogP[21] between −5 and 7, and only H, B, C, N, O, F, P, S, Cl, and Br atoms allowed. Additionally, all counterions and solvents were removed, and canonical SMILES representations were generated by RDKit.[22] This procedure yielded 19.9 million compounds.

Compound Encoding and Mol2vec Model. In a fashion analogous to NLP, molecules were considered as sentences and substructures as words. To obtain words for each molecule, the Morgan algorithm[1] was used to generate all atom identifiers at radii 0 and 1, resulting in 119 and 19 831 identifiers, respectively. Identifiers of each atom (radius 0 followed by radius 1 each) were then ordered into a "molecular sentence" with the same atom order as present in the canonical SMILES representation (Figure 2).

All rare words that occurred less than three times in the corpus were replaced with a string "UNSEEN" because (1) Word2vec is not able to get meaningful embeddings for rare words and (2) this enables the model to gracefully handle unseen (or unknown) words that might appear when predictions are performed on new data. The distribution of "UNSEEN" in the corpus is random, and hence, a vector close to zero is usually learned. All 19.9 million compounds from the corpus were processed to obtain molecular sentences, which were subsequently used as input for the training of the Word2vec algorithm to obtain a Mol2vec model. The training was done using the gensim[23] implementation of Word2vec, which is a shallow, two-layer neural network.

Although Word2vec is an unsupervised method, it still internally defines an auxiliary prediction task. Depending on the auxiliary task, Word2vec can be trained using one of the following two approaches: (1) continuous bag of words (CBOW) if the task is to predict a word from the context words, or (2) Skip-gram if the context is predicted on the basis of a word. In CBOW the order of words in the context is not important because of the bag-of-words assumption, while in Skip-gram adjacent words are assigned with higher weights. Furthermore, the two parameters "window size" and "dimensional embeddings" were explored to find the best settings for Mol2vec. The window size controls the size of the context and was set to the in NLP commonly used sizes of 5 and 10 in the case of CBOW and Skip-gram, respectively. Furthermore, to account for the fact that each atom is represented twice via Morgan identifiers (i.e., at radius 0 and 1), the effect of double window sizes (i.e., 10 for CBOW and 20 for Skip-gram) was also evaluated. Finally, 100- and 300-dimensional embeddings were generated for all combinations.

The vector for a molecule (featurization) is finally obtained by summing all of the vectors of the Morgan substructures of the molecule. If an unknown identifier occurs during featurization of the new data, the "UNSEEN" vector is employed.

A Mol2vec implementation that automates the above steps is available as a Python package via https://github.com/samoturk/mol2vec. The preparation of the database (i.e., converting molecules to sentences) takes ~8 h for 20 million compounds on a modern quad-core CPU, with an additional ~2 h to train the Mol2vec model.

ProtVec Model. The protein corpus of 554 241 sequences was collected from UniProt,[24] and the protein sequences were featurized using the ProtVec[16] approach. All possible words were generated by representing each sequence in the corpus as three sequence variants (i.e., sentences), each shifted by one amino acid, followed by the generation of all possible 3-grams (words) (Figure 3). This yielded 1 662 723 sentences for the protein corpus.

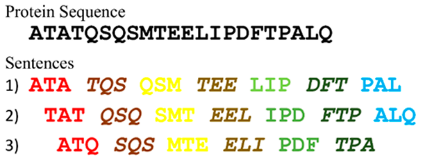

Protein Sequence
ATATQSQSMTEELIPDFTPALQ
Sentences
1) ATA TQS QSM TEE LIP DFT PAL
2) TAT QSQ SMT EEL IPD FTP ALQ
3) ATQ SQS MTE ELI PDF TPA

Figure 3. Protein sequence processing. Each sequence is represented as n sequences (i.e., sentences) with a shifted reading frame and split into n-grams (i.e., words). Here n = 3.

The ProtVec model was trained with the genism Word2vec implementation using recommended settings[16] (i.e., a Skip-gram architecture with a window size of 25), except for the output vector size, which was increased to 300 to match the dimensionality of the Mol2vec vectors. To handle potentially new 3-grams, the model was trained on "UNSEEN" words in a similar way as the Mol2vec models. The final model resulted in high-dimensional embeddings of 9154 unique 3-grams.

PCM Vectors. For the PCM approach, Mol2vec was combined with ProtVec (called PCM2vec below) by concatenating the two vectors. Baseline PCM vectors were concatenated with Morgan FPs (2048 bits) and z-scales,[25] which are sequence-based physicochemical protein descriptors. Since the use of z-scales relies on a sequence alignment, they were used only for the kinase data set. Following the study described in ref 7, the kinase sequences were aligned, and the z-scales (z_3) were calculated only for the 85 binding site residues defined in KLIFS.[26] The length of the target descriptor was adjusted to 2048 using a WTA-hash function to match the dimensionality of the Morgan FP.[27]

Benchmarking Data Sets. The performance of Mol2vec in subsequent ML models was evaluated using the ESOL, Ames, and Tox21 data sets as well as one curated kinase data set:

- The *ESOL solubility data set*[28] was chosen to evaluate the performance of Mol2vec in a regression task to predict the aqueous solubilities of 1144 compounds.
- The *Ames mutagenicity data set*[29] contains 6471 compounds that were determined to be either mutagenic (3481) or nonmutagenic (2990) and thus represents a balanced data set for classification.
- The *Tox21 data set*[30] consists of 12 targets that were associated with human toxicity and contains a total of 8192 compounds. Tox21 was retrieved as a part of the DeepChem package[31] to enable a comparison with established methods.
- A *kinase data set* was compiled using ChEMBL version 23 and evaluated with respect to classification tasks.[19] Bioactivities for 284 kinases (listed in the Supporting Information) were extracted and filtered to keep only IC_{50}, K_d, and K_i values from binding assays with a target confidence of at least 8. Bioactivities were converted to pIC_{50} values, and an activity threshold of 6.3 was employed.

Validation of Models Based on Mol2vec Vectors. All of the machine learning models using compound data were trained using 20× 5-fold cross-validation and compared using the Wilcoxon signed rank test. The following performance metrics were employed: coefficient of determination (R^2_{cv}), mean absolute error (MAE), and mean squared error (MSE) for the regression tasks and area under the receiver operating characteristic (ROC) curve (AUC), sensitivity (i.e., true-

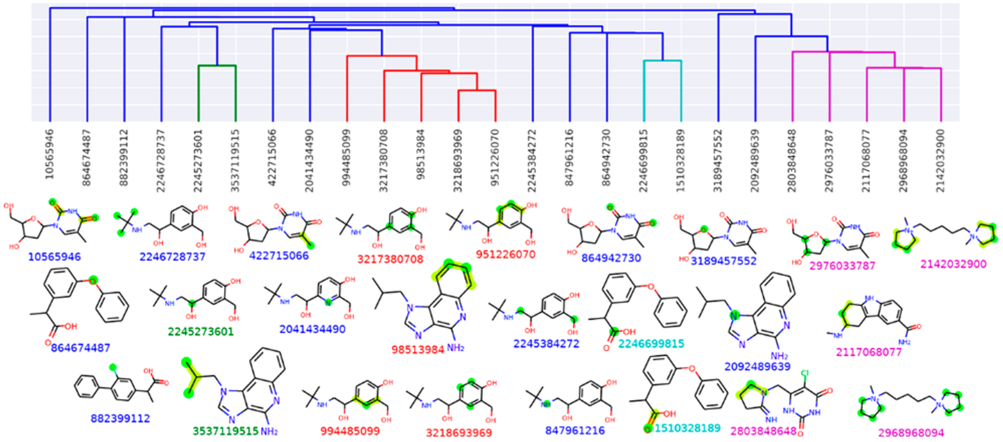


Figure 4. Dendrogram showing relationships among vectors representing the 25 most common substructures in the compound corpus. Substructures are depicted (central atoms in green and surrounding atoms in light green) on a representative compound from a pool of FDA-approved drugs.

positive rate), and specificity (i.e., true-negative rate) for the classification tasks. Compounds in all data sets were processed using RDKit to remove compounds with fewer than three heavy atoms, to remove all salts (i.e., counterions) and solvents, and to generate canonical SMILES representations. Compounds were encoded as vectors (featurization) by summing the vectors of Morgan substructures retrieved from the pretrained Mol2vec model. If a Morgan substructure was new to the Mol2vec model, it was represented with a vector for “UNSEEN”. Morgan FPs with radius 2 and hashed to 4096 bits (2048 bits for PCM experiments because of memory constraints) were used as baseline features to train fingerprint-based ML models.

A PCM approach was evaluated for the two data sets with several targets (i.e., Tox21 and kinase bioactivities) to assess the influence of adding protein information by concatenating compound descriptors (Morgan FP or Mol2vec) with protein descriptors (z-scales or ProtVec). ProtVec descriptors for the proteins in the Tox21 (Supporting Information List S1) and kinase (List S2) data sets were calculated on the basis of the entire protein and catalytic domain, respectively. The performance of the PCM models was evaluated by a rigorous four-level (CV1−CV4) validation scheme.[7] Briefly, CV1 tests the model performance on new compound−target pairs, CV2 on new targets, CV3 on new compounds, and CV4 on by the model new compounds and targets.

Machine Learning Methods. Three different machine learning methods (i.e., random forest (RF), gradient boosting machine (GBM), and deep neural network (DNN)) were evaluated using Mol2vec embeddings as compound features. The implementation of RF in scikit-learn[32] was used with 500 estimators, the square root of the number of features as the maximum number of features, and balanced class weight. The XGBoost implementation[33] of GBM was used with 2000 estimators, the maximum depth of trees set to 3, and the learning rate set to 0.1. For the GBM classifier, the weight of the positive samples was adjusted to reflect the actives/inactives ratio in the respective data set. Several feed-forward DNNs were built using Keras[34] with the TensorFlow[35] back end. After an initial benchmarking, variations of two different DNN architectures were optimized for different types of features (i.e., binary FPs and continuous Mol2vec vectors): (1) DNNs trained with Morgan FPs (radius 2) had one hidden layer with 512 neurons and an output layer with one neuron. All of the layers had normal initialization and employed the rectified linear unit (ReLU)[36] activation function, except for the output neuron in the case of classification tasks, which employed a sigmoid activation function. The Adam optimizer[37] was used to minimize the Poisson loss function for classification and to minimize the MSE for regression. (2) DNNs trained with Mol2vec embeddings had four hidden layers with 2000 neurons each and one output neuron. All of the layers had normal initialization and employed the ReLU activation function, except for the output neuron in the case of classification tasks, which again employed a sigmoid activation function. The Adamax optimizer[37] was used to minimize the binary cross entropy loss function for classification and to minimize the MSE for regression. All of the DNNs used a dropout value of 0.1 to avoid overfitting.[38] In the case of the DNN classifiers, the weight of the actives was adjusted to reflect the imbalance in the data.

RESULTS AND DISCUSSION

Mol2vec is an unsupervised pretraining method to generate an information-rich representation of molecular substructures. Since it is an unsupervised method, it does not require labeled data as input and can leverage from larger amounts like the here-employed 19.9 million compounds. The obtained embeddings from the pretraining can be used, for instance, to explore the relationships between different substructures, while the derived compound vectors can be used to assess compound similarity or as features for supervised ML predictions.

Mol2vec Training and Hyperparameter Evaluation. The evaluation of different parameters for Mol2vec revealed that the best settings overlap with those recommended for Word2vec in NLP on text data and comprise the Skip-gram model with a window size of 10 and 300-dimensional embeddings of Morgan substructures. The quality of the individual embeddings was assessed by using them as features in supervised ML tasks on the Tox21 data set (Table S1 in the Supporting Information). As was observed for NLP applications,[15] also in our case Skip-gram yielded higher-performing embeddings compared with CBOW, possibly because it captures spatial relationships better as a result of the weighting of words in the context. Higher dimensionality of embeddings also had a beneficial effect on the performance, while varying

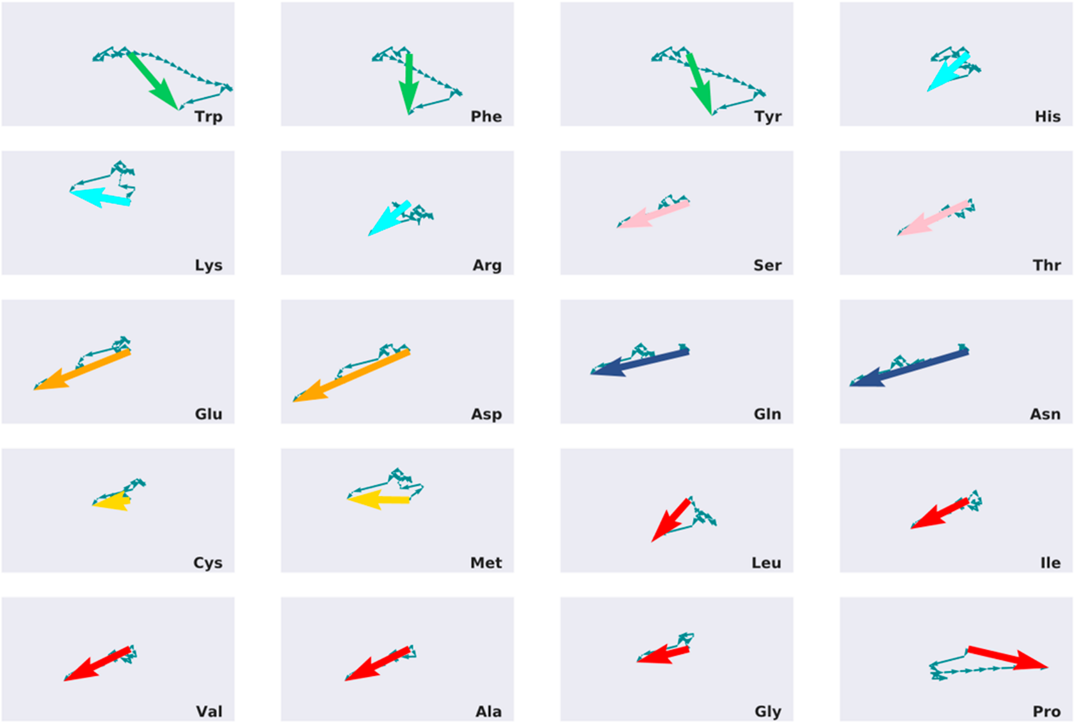


Figure 5. 2D projections (t-SNE) of Mol2vec vectors of amino acids (bold arrows). These vectors were obtained by summing the vectors of the Morgan substructures (small arrows) present in the respective molecules (amino acids in the present example). The directions of the vectors provide a visual representation of similarities. A quantitative assessment can be obtained via cosine angle calculations (Table S2).

Table 1. Performance of Mol2vec and Other Models on Regression Predictions of the ESOL Data Set

ML features	ML method	R^2_{cv}	MSE	MAE	ref
descriptors	MLR	0.81 ± 0.01	0.82	0.69	28
molecular graph	CNN	0.82	–	–	40
molecular graph	CNN	–	–	0.52 ± 0.07	41
molecular graph	CNN	0.93	0.31 ± 0.03	0.40 ± 0.00	9
molecular graph	RNN	0.92 ± 0.01	0.35	0.43	42
Morgan FPs	GBM	0.66 ± 0.00	1.43 ± 0.00	0.88 ± 0.00	this work
Mol2vec	GBM	0.86 ± 0.00	0.62 ± 0.00	0.60 ± 0.00	this work

the window size had almost no effect. The final Mol2vec model was trained on a corpus of 19.9 million molecules.

Chemical Intuition of Mol2vec Descriptors. A key assumption of the Mol2vec approach is that related functional groups and molecules are close in the generated vector space. This was visually investigated as well as quantified by extracting the 25 most common substructures from the compound corpus as well as featurizing standard amino acids via Mol2vec descriptors. Encouragingly, the Mol2vec vectors of the 25 most common substructures cluster in expected relationships (Figure 4). Aromatic carbon types are correctly identified to be chemically related (red identifiers), as are aliphatic carbons in ring systems (purple), nonring aliphatic carbons (green), and carbonyl carbon and oxygen (turquoise). Further interesting relationships can be explored by looking at more substructures concurrently. Readers are encouraged to visit https://github.com/samoturk/mol2vec_notebooks for an interactive visualization of the embeddings of the 300 most common identifiers.

Another consequence of using the Word2vec approach is that the underlying vector magnitude reflects the importance of a word,[39] or in this case a substructure. The magnitudes of vectors for very frequent words (e.g., "the", "a", "an", etc.) are usually smaller that those of vectors for less frequent and more meaningful words. This behavior is also observed in Mol2vec, where vectors of common substructures (e.g., different carbon types) have lower magnitudes. There is a slight negative correlation between substructure occurrence and vector magnitude ($R = -0.39$).

Similarly, 2D projections of vector representations obtained for the 20 proteinogenic amino acids also agree with chemical intuition and capture the similarity between related amino acids (Figure 5 and Table S2). For instance, Pro is an obvious outlier, while other amino acids are nicely grouped on the basis of their functional groups and properties. Also interesting is the observation that the transition distance between Glu and Gln is similar to that between Asp and Asn, which is line with the underlying change of the carboxylic acid group to an amide.

Next, the prediction capabilities of Mol2vec vectors are demonstrated on several compound property and activity data sets and compared with results obtained for the Morgan fingerprints as a reference compound representation.

Mol2vec Compound Vectors as Features for Supervised ML Tasks. Employing Mol2vec vectors as input for different ML tasks (Figure 1), such as classification and regression, and on a variety of data sets indicated overall that Mol2vec vectors yield state-of-the-art performance. The

Table 2. Performance of Mol2vec and Other Methods on Classification Prediction of the Ames Data Set

ML features	ML method	AUC	sensitivity	specificity	ref
descriptors	SVM	0.86 ± 0.01	–	–	29
descriptors and Morgan FPs	NBC	0.84 ± 0.01	0.74 ± 0.02	0.81 ± 0.01	43
Morgan FPs	RF	0.88 ± 0.00	0.82 ± 0.00	0.80 ± 0.01	this work
Mol2vec	RF	0.87 ± 0.00	0.80 ± 0.01	0.80 ± 0.01	this work

Table 3. Performance of Mol2vec and Other Methods on Classification Predictions of the Tox21 Data Set

ML features	ML method	AUC	sensitivity	specificity	ref
molecular graph	CNN	0.71 ± 0.13	–	–	9
molecular descriptors and FPs	SVM	0.71 ± 0.13	–	–	5
molecular descriptors and FPs	DNN	0.72 ± 0.13	–	–	5
Morgan FPs	RF	0.83 ± 0.05	0.28 ± 0.14	0.99 ± 0.01	this work
Mol2vec	RF	0.83 ± 0.05	0.20 ± 0.15	1.00 ± 0.01	this work

Table 4. Summary of the Prediction Performance of PCM Models in Comparison to Compound Features on Tox21[a]

validation level	ML features	AUC	sensitivity	specificity
CV1	Morgan FPs	0.79 ± 0.01	0.73 ± 0.01	0.74 ± 0.00
	Mol2vec	0.78 ± 0.01	0.73 ± 0.00	0.72 ± 0.02
	PCM2vec	**0.87 ± 0.01**	**0.80 ± 0.01**	**0.79 ± 0.01**
CV2	Morgan FPs	0.73 ± 0.07	0.63 ± 0.08	0.71 ± 0.03
	Mol2vec	0.72 ± 0.07	0.65 ± 0.09	0.68 ± 0.04
	PCM2vec	0.70 ± 0.04	0.55 ± 0.02	0.69 ± 0.04
CV3	Morgan FPs	0.78 ± 0.02	0.65 ± 0.03	0.77 ± 0.02
	Mol2vec	0.79 ± 0.01	0.70 ± 0.02	0.74 ± 0.01
	PCM2vec	**0.85 ± 0.01**	**0.75 ± 0.01**	**0.80 ± 0.01**
CV4	Morgan FPs	0.76 ± 0.03	0.59 ± 0.06	0.77 ± 0.05
	Mol2vec	0.73 ± 0.06	0.62 ± 0.10	0.74 ± 0.05
	PCM2vec	0.75 ± 0.02	0.61 ± 0.12	0.73 ± 0.11

[a] Validation levels: CV1, new compound−target pairs; CV2, new targets; CV3, new compounds; CV4, new compounds and targets. Highlighted cases mark the validation levels where PCM2vec outperforms the ML models based on compound features only.

combination with GBM seems to be very suitable for regression tasks (i.e., Ames data set), while the combination with RF can be recommended for classification tasks, including proteochemometric (PCM) approaches.

Although several DNN architectures were evaluated, they were still outperformed by the tree-based methods GBM and RF. However, we would like to note that further fine-tuning might improve the prediction power of the Mol2vec−DNN combination. Tables S3−S5 present detailed performance numbers for all of the employed methods. In the following, the best predictions obtained for the Mol2vec vectors are described more in detail and compared with results obtained for the Morgan FPs as baseline descriptors as well as results described in the literature for the employed data sets.

The ESOL solubility data set was selected to test Mol2vec in a regression task (Table 1 and Figure S1). Mol2vec yields better predictions (R^2_{cv} = 0.86) than the originally reported multiple linear regression (MLR) model[28] as well as a molecular graph convolution method,[40] and importantly, it outperforms our Morgan-FP-based baseline model (R^2_{cv} = 0.66). However, the best reported results on the ESOL data set to date were obtained by two molecular graph convolution methods[9,41] and one recurrent-network-based method[42] ($R^2_{\mathrm{cv}} \approx$ 0.93).

The Ames benchmarking data set is a classic toxicological data set used to benchmark various classification ML methods. Here Mol2vec and Morgan FPs result in equally good predictions and are in line with AUC results reported for an SVM model and a naïve Bayes classifier (NBC)[43] on this data set,[29] but the former two achieved higher sensitivity values (Table 2).

The Tox21 data set represents a challenging classification data set covering 12 targets and over 8000 compounds with unbalanced classes. Here, Mol2vec and Morgan FPs result in equally good predictions (i.e., both obtained average AUC values of 0.83, although Morgan FPs yielded a slightly more balanced model) and this time outperform existing literature results (Tables 3 and S6).

Overall, Mol2vec descriptors show competitive performance compared with the classic Morgan FPs on the employed benchmarking data sets for classification (i.e., Ames and Tox21) and outperformed the Morgan FPs on the regression predictions for the ESOL data set. Morgan FP and Mol2vec features are both based on identifiers calculated by the Morgan algorithm. While these identifiers are hashed to a binary fingerprint in the case of the Morgan FPs, they form a vector with continuous values in the case of the Mol2vec approach. Since the final Mol2vec vector is a sum of substructure vectors, it implicitly captures substructure counts as well as substructure importance via the vector amplitude. In addition, Mol2vec also has the advantage of lower dimensionality of the final vectors, which significantly speeds up the training and lowers the memory requirements. Further tuning of the Mol2vec approach, for example by using Morgan identifiers with bigger radii, might further improve the prediction performance.

Mol2vec Compound and ProtVec Protein Vectors as Features for PCM. Earlier studies using the PCM approach, where compound and protein descriptors are employed as

Table 5. Summary of Prediction Performance of PCM Models in Comparison to Compound Features on the Kinase Data Set[a]

validation level	ML features	AUC	sensitivity	specificity
CV1	Morgan FPs	0.91 ± 0.00	0.82 ± 0.00	0.85 ± 0.00
	Mol2vec	0.91 ± 0.00	0.83 ± 0.00	0.84 ± 0.00
	Morgan FPs + z-score	**0.96 ± 0.00**	**0.89 ± 0.00**	**0.90 ± 0.00**
	PCM2vec	**0.95 ± 0.00**	**0.89 ± 0.00**	**0.87 ± 0.00**
CV2	Morgan FPs	0.88 ± 0.00	0.76 ± 0.01	0.85 ± 0.01
	Mol2vec	0.89 ± 0.00	0.80 ± 0.01	0.83 ± 0.00
	Morgan FPs + z-score	0.89 ± 0.00	0.37 ± 0.03	0.96 ± 0.00
	PCM2vec	0.89 ± 0.00	0.65 ± 0.01	0.90 ± 0.01
CV3	Morgan FPs	0.82 ± 0.00	0.94 ± 0.00	0.41 ± 0.01
	Mol2vec	0.78 ± 0.01	0.97 ± 0.00	0.24 ± 0.01
	Morgan FPs + z-score	**0.94 ± 0.00**	**0.86 ± 0.01**	**0.89 ± 0.00**
	PCM2vec	**0.91 ± 0.00**	**0.92 ± 0.01**	**0.63 ± 0.02**
CV4	Morgan FPs	0.74 ± 0.02	0.87 ± 0.02	0.43 ± 0.02
	Mol2vec	0.73 ± 0.02	0.94 ± 0.01	0.26 ± 0.03
	Morgan FPs + z-score	**0.84 ± 0.02**	**0.35 ± 0.04**	**0.96 ± 0.01**
	PCM2vec	**0.77 ± 0.02**	**0.68 ± 0.04**	**0.72 ± 0.02**

[a] Highlighted cases mark the validation levels where PCM models (i.e., Morgan FPs + z-scores and PCM2vec) outperform the ML models based on compound features only (i.e., Morgan FPs and Mol2vec).

concatenated features, indicate that the additional use of protein information can improve the prediction quality.[44] To test the benefit of Mol2vec for PCM applications, Mol2vec vectors were coupled with ProtVec vectors. Such PCM2vec models (Tabled 4 and 5) were compared with results obtained for Morgan FPs, Mol2vec, and in the case of the kinase data set also with a classical PCM approach (i.e., Morgan FPs for compounds combined with z-scales for proteins). In each scenario, one model was trained on the entire set of training data for several targets, allowing the benefit of protein descriptors to be quantified. Among several ML methods evaluated (Tables S7 and S8), RF yielded the best results overall, and it was employed for the results below.

PCM models existing to date required sequence alignments and were thus built for conserved target classes such as kinases and G-protein-coupled receptors.[44] Thus, it is worth noting that ProtVec is alignment-independent and thus can be directly applied not only to the kinase data set (Table 5) but also to the Tox21 data set (Table 4), which consists of unrelated proteins with low sequence similarity. On both data sets, PCM2vec outperforms the compound features in CV1 and CV3, which indicates that the added protein information improves the extrapolation to new compound−target pairs and new compounds (Tables 4 and 5). In CV2, the here-employed PCM2vec approach performs slightly worse than Morgan FPs and Mol2vec on the Tox21 data set and performs equally well on the kinase data set. For CV4, PCM2vec achieves similar performance as the compound fingerprints on the Tox21 data set and better predictions on the kinase data.

Since kinases share high sequence similarity, Morgan FPs + z-scales was added as a baseline PCM approach when evaluating the impact of the employed features on the prediction of the kinase data set. The comparison of PCM2vec with a classical PCM model for kinases (i.e., Morgan FPs + z-scores) revealed that the latter yields equally good results in CV1 and CV2 (i.e., new compound−target pairs and new targets, respectively) and slightly better results in CV3 and CV4 (i.e., new compounds and new compounds and targets, respectively). One reason for the better prediction of the classical PCM model might be that it considers binding site residues only, while ProtVec was built on the basis of the entire kinase domain. However, although there is a performance difference between PCM2vec and Morgan FPs + z-scales, practically PCM2vec yields in the present case more balanced models with roughly equal sensitivity and specificity (i.e., in CV2 and CV4). Furthermore, it can be also directly applied on distant protein classes such as the Tox21 data set, in general resulting in better predictions for new compound−target pairs as well compounds compared with using only compound-based features.

CONCLUSION

Inspired by natural language processing (NLP) techniques, Mol2vec represents a novel way of embedding compound substructures as information-rich vectors. The substructures were extracted in the present study by employing the Morgan algorithm and, in the context of NLP, represent words while the complete molecules are sentences. New compounds can be described by summing the substructure vectors retrieved from a pretrained Mol2vec model.

The Mol2vec model itself is an unsupervised pretraining method that is trained on all available chemical structures, yielding high-quality embeddings of molecules. Results on common substructures as well as amino acids nicely illustrate that the derived substructure vectors of chemically related substructures and compounds occupy similar vector space. This result is not unexpected since Word2vec vectors representing similar words also end up near in vector space. Substructure vectors can simply be summed to obtain compound vectors, which can be used to calculate compound similarity or as features in supervised ML tasks. A thorough evaluation of Mol2vec on different chemical data sets showed that it can achieve state-of-the-art performance and compared with Morgan FPs seems to be especially suited for regression tasks. Additionally, Mol2vec combined with ProtVec (i.e., PCM2vec) performs well in proteochemometrics approaches and can be directly applied to data sets with unrelated targets with low sequence similarities.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.7b00616.

> Evaluation of different Word2vec hyperparameters (Table S1); pairwise cosine distances of Mol2vec vectors of amino acids (Table S2); evaluation of different features and machine learning methods on the ESOL data set (Table S3); evaluation of different features and machine learning methods on the Ames data set (Table S4); evaluation of different features and machine learning methods on the Tox21 data set (Table S5); results of machine learning predictions on individual Tox21 targets (Table S6); evaluation of different machine learning methods in the PCM approach on the Tox21 data set (Table S7); evaluation of different machine learning methods in the PCM approach on the kinase data set (Table S8); experimental versus predicted solubilities obtained for models trained with Morgan fingerprints and Mol2vec features (Figure S1); Tox21 assays used for PCM models (List S1); kinases in the kinase data set (List S2) (PDF)

AUTHOR INFORMATION

Corresponding Authors

*E-mail: fulle@bio.mx.
*E-mail: turk@bio.mx.

ORCID

Sabrina Jaeger: 0000-0003-1144-7468
Simone Fulle: 0000-0002-7646-5889
Samo Turk: 0000-0003-2044-7670

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank Benjamin Merget for fruitful discussions and Katra Kolšek for feedback on the manuscript.

ABBREVIATIONS

AUC, area under the curve; CNN, convolutional neural network; DNN, deep neural networks; FP, fingerprint; GBM, gradient boosting machine; MAE, mean absolute error; MLR, multiple linear regression; MSE, mean squared error; PCM, proteochemometrics; ReLU, rectified linear unit; RF, random forest; RNN, recurrent neural network; ROC, receiver operating characteristic; SVM, support vector machine; XGB, extreme gradient boosting

REFERENCES

(1) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(2) Riniker, S.; Landrum, G. A. Open-Source Platform to Benchmark Fingerprints for Ligand-Based Virtual Screening. *J. Cheminf.* **2013**, *5*, 26.

(3) O'Boyle, N. M.; Sayle, R. A. Comparing Structural Fingerprints Using a Literature-Based Similarity Benchmark. *J. Cheminf.* **2016**, *8*, 36.

(4) Riniker, S.; Fechner, N.; Landrum, G. A. Heterogeneous Classifier Fusion for Ligand-Based Virtual Screening: Or, How Decision Making by Committee Can Be a Good Thing. *J. Chem. Inf. Model.* **2013**, *53*, 2829−2836.

(5) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction Using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 80.

(6) Merget, B.; Turk, S.; Eid, S.; Rippmann, F.; Fulle, S. Profiling Prediction of Kinase Inhibitors: Towards the Virtual Assay. *J. Med. Chem.* **2017**, *60*, 474−485.

(7) Sorgenfrei, F. A.; Fulle, S.; Merget, B. Kinome-Wide Profiling Prediction of Small Molecules. *ChemMedChem* **2017**, DOI: 10.1002/cmdc.201700180.

(8) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.

(9) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *J. Chem. Inf. Model.* **2017**, *57*, 1757−1772.

(10) 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. 2017, arxiv:1706.06689 [stat.ML]. arXiv.org e-Print archive. https://arxiv.org/abs/1706.06689.

(11) Hull, R. D.; Singh, S. B.; Nachbar, R. B.; Sheridan, R. P.; Kearsley, S. K.; Fluder, E. M. Latent Semantic Structure Indexing (LaSSI) for Defining Chemical Similarity. *J. Med. Chem.* **2001**, *44*, 1177−1184.

(12) Wan, F.; Zeng, J. Deep Learning with Feature Embedding for Compound−Protein Interaction Prediction. 2016, bioRχiv preprint server. DOI: 10.1101/086033.

(13) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-Novo Design through Deep Reinforcement Learning. *J. Cheminf.* **2017**, *9*, 48.

(14) Schneider, N.; Fechner, N.; Landrum, G. A.; Stiefl, N. Chemical Topic Modeling: Exploring Molecular Data Sets Using a Common Text-Mining Approach. *J. Chem. Inf. Model.* **2017**, *57*, 1816−1831.

(15) Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. 2013, arxiv:1301.3781 [cs.CL]. arXiv.org e-Print archive. https://arxiv.org/abs/1301.3781.

(16) Asgari, E.; Mofrad, M. R. K. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One* **2015**, *10*, e0141287.

(17) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.

(18) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.

(19) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papadatos, G.; Santos, R.; Overington, J. P. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, D1083−D1090.

(20) RDKit. http://www.rdkit.org/ (accessed Aug 8, 2017).

(21) Wildman, S.; Crippen, G. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Model.* **1999**, *39*, 868−873.

(22) Schneider, N.; Sayle, R. A.; Landrum, G. A. Get Your Atoms in Order—An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm. *J. Chem. Inf. Model.* **2015**, *55*, 2111−2120.

(23) Řehuřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*; ELRA: Valletta, Malta, 2010; pp 45−50.

(24) UniProt Consortium. UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Res.* **2017**, *45*, D158−D169.

(25) Sandberg, M.; Eriksson, L.; Jonsson, J.; Sjöström, M.; Wold, S. New Chemical Descriptors Relevant for the Design of Biologically

Active Peptides. A Multivariate Characterization of 87 Amino Acids. *J. Med. Chem.* **1998**, *41*, 2481−2491.

(26) Kooistra, A. J.; Kanev, G. K.; van Linden, O. P. J.; Leurs, R.; de Esch, I. J. P.; de Graaf, C. KLIFS: A Structural Kinase-Ligand Interaction Database. *Nucleic Acids Res.* **2016**, *44*, D365−D371.

(27) Yagnik, J.; Strelow, D.; Ross, D. A.; Lin, R. S. The Power of Comparative Reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: New York, 2011; pp 2431−2438.

(28) Delaney, J. S. ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *J. Chem. Inf. Model.* **2004**, *44*, 1000−1005.

(29) Hansen, K.; Mika, S.; Schroeter, T.; Sutter, A.; ter Laak, A.; Steger-Hartmann, T.; Heinrich, N.; Muller, K. R. Benchmark Data Set for in Silico Prediction of Ames Mutagenicity. *J. Chem. Inf. Model.* **2009**, *49*, 2077−2081.

(30) Tox21 Data Challenge 2014. https://tripod.nih.gov/tox21/challenge/ (accessed Sept 27, 2017).

(31) Ramsundar, B.; Eastman, P.; Feinberg, E.; Gomes, J.; Leswing, K.; Pappu, A.; Wu, M.; Pande, V. DeepChem: Democratizing Deep-Learning for Drug Discovery, Quantum Chemistry, Materials Science and Biology. https://github.com/deepchem/deepchem (accessed Aug 8, 2017).

(32) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(33) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. 2016, arxiv:1603.02754 [cs.LG]. arXiv.org e-Print archive. https://arxiv.org/abs/1603.02754.

(34) Keras Team. Keras: Deep Learning for Python. https://github.com/fchollet/keras (accessed Aug 8, 2017).

(35) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. https://research.google.com/pubs/pub45166.html (accessed Sep 27, 2017).

(36) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*; Fürnkranz, J., Joachims, T., Eds.; Omnipress: Madison, WI, 2010; pp 807−814.

(37) Kingma, D. P.; Ba, J. L. Adam: A Method of Stochastic Optimization. 2014, arxiv:1412.6980 [cs.LG]. arXiv.org e-Print archive. https://arxiv.org/abs/1412.6980.

(38) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way To Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.

(39) Schakel, A. M. J.; Wilson, B. J. Measuring Word Significance Using Distributed Representations of Words. 2015, arxiv:1508.02297 [cs.CL]. arXiv.org e-Print archive. https://arxiv.org/abs/1508.02297.

(40) 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. 2017, arxiv:1703.00564 [cs.LG]. arXiv.org e-Print archive. https://arxiv.org/abs/1703.00564.

(41) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. 2015, arxiv:1509.09292 [cs.LG]. arXiv.org e-Print archive. https://arxiv.org/abs/1509.09292.

(42) Lusci, A.; Pollastri, G.; Baldi, P. Deep Architectures and Deep Learning in Chemoinformatics: The Prediction of Aqueous Solubility for Drug-like Molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1563−1575.

(43) Zhang, H.; Kang, Y.-L.; Zhu, Y.-Y.; Zhao, K.-X.; Liang, J.-Y.; Ding, L.; Zhang, T.-G.; Zhang, J. Novel Naïve Bayes Classification Models for Predicting the Chemical Ames Mutagenicity. *Toxicol. In Vitro* **2017**, *41*, 56−63.

(44) Cortés-Ciriano, I.; Ain, Q. U.; Subramanian, V.; Lenselink, E. B.; Méndez-Lucio, O.; IJzerman, A. P.; Wohlfahrt, G.; Prusis, P.; Malliavin, T. E.; van Westen, G. J. P.; Bender, A. Polypharmacology Modelling Using Proteochemometrics (PCM): Recent Methodological Developments, Applications to Target Families, and Future Prospects. *MedChemComm* **2015**, *6*, 24−50.