biomolecules

MDPI

Article

Predicting Potent Compounds Using a Conditional Variational Autoencoder Based upon a New Structure–Potency Fingerprint

Tiago Janela, Kosuke Takeuchi and Jürgen Bajorath *

Department of Life Science Informatics and Data Science, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Friedrich-Hirzebruch-Allee 5/6, D-53115 Bonn, Germany

* Correspondence: bajorath@bit.uni-bonn.de; Tel.: +49-228-73-69100

Abstract: Prediction of the potency of bioactive compounds generally relies on linear or nonlinear quantitative structure–activity relationship (QSAR) models. Nonlinear models are generated using machine learning methods. We introduce a novel approach for potency prediction that depends on a newly designed molecular fingerprint (FP) representation. This structure–potency fingerprint (SPFP) combines different modules accounting for the structural features of active compounds and their potency values in a single bit string, hence unifying structure and potency representation. This encoding enables the derivation of a conditional variational autoencoder (CVAE) using SPFPs of training compounds and apply the model to predict the SPFP potency module of test compounds using only their structure module as input. The SPFP–CVAE approach correctly predicts the potency values of compounds belonging to different activity classes with an accuracy comparable to support vector regression (SVR), representing the state-of-the-art in the field. In addition, highly potent compounds are predicted with very similar accuracy as SVR and deep neural networks.

Keywords: bioactive compounds; potency prediction; fingerprints; machine learning; conditional variational autoencoder

Citation: Janela, T.; Takeuchi, K.; Bajorath, J. Predicting Potent Compounds Using a Conditional Variational Autoencoder Based upon a New Structure–Potency Fingerprint. *Biomolecules* **2023**, *13*, 393. https://doi.org/10.3390/biom13020393

Academic Editors: Umesh Desai, Daniel Afosah and Mire Zloh

Received: 14 December 2022
Revised: 7 February 2023
Accepted: 16 February 2023
Published: 18 February 2023

1. Introduction

Compound potency prediction is a major task in chemoinformatics and computational medicinal chemistry. For potency prediction, both structure- and ligand-based approaches are available. Structure-based methods attempt to predict small molecule (ligand) potency on the basis of experimental (or modeled) three-dimensional (3D) structures of ligand–target complexes. Ideally, such predictions aim to calculate the free energy of binding [1,2], for example, by applying alchemical free energy perturbation methods [2]. These calculations are challenging due to their high computational costs and the need to achieve consistent accuracy across different targets and compound classes [1]. Alternatively, scoring functions of different levels of sophistication are used to approximate ligand binding energies [3–6].

At the other end of the methodological spectrum reside classical ligand-based approaches for 2D and 3D quantitative structure–activity relationship (QSAR) modeling, which derive linear descriptor-based models for predicting potency values of congeneric compounds (structural analogues) [7,8]. Furthermore, for ligand-based modeling of non-linear SARs and potency prediction, random forest (RF) regression [9] and, in particular, support vector regression (SVR) have become preferred machine learning approaches [10,11]. While SVR typically produces statistically sound prediction models, it also displays a tendency to under-predict the individual most potent compounds because they are often algorithmically classified as outliers [12].

The increasing popularity of deep machine learning in pharmaceutical research [13–17] is also influencing structure- and ligand-based potency prediction. One of the attractions of deep learning is the ability to derive new object representations from input data such

as molecular graphs, thereby alleviating the need to use pre-conceived molecular descriptors for prediction tasks. Suitable deep neural network (DNN) architectures have been adapted for developing scoring functions [6,7] or deriving ligand–target binding energy models [18–21]. Despite their apparent success, such models are in part controversially viewed due to the observed strong dependence of their performance on varying training set composition [22,23], resulting from the memorization of training data leading to apparently accurate predictions that do not depend on correctly accounting for ligand–target interactions [22–24]. Similar observations have also been made for deep compound classification models with limited generalization ability [25]. In addition to studying ligand–target interactions, DNNs are also intensely investigated for ligand-based molecular property predictions including potency [26–29]. To these ends, various DNN architectures and learning strategies have been adapted. However, on data sets from medicinal chemistry, which are often limited in size, DNN-based property prediction models often do often exceed—or even meet—the performance of simpler models [29,30]. Hence, for both compound property and potency prediction, no firm conclusion can currently be drawn concerning the potential superiority of DNNs over standard approaches. We have recently shown that k-nearest neighbor (kNN) analysis meets the accuracy of other ML methods in potency prediction [31]. Moreover, randomized predictions typically reproduce experimental potency values within an order of magnitude, which is a direct consequence of potency value distributions in compound activity classes commonly used for benchmarking [31]. Hence, the best ML models and random predictions are only distinguished by a small margin of maximally one order of magnitude, representing a general limitation associated with the benchmarking of potency prediction methods. This needs to be taken into consideration when evaluating these methods, calling for the inclusion of simple controls such as kNN.

In this work, we introduce a novel concept for compound potency prediction that combines a special fingerprint (FP), termed structure–potency FP (SPFP), with a deep learning approach. FPs accounting for chemical structure and topology are a mainstay for chemical similarity searching [32,33]. SPFP is the first FP representation specifically designed to combine compound structure and potency information in a modular format. Using SPFP, a conditional variational autoencoder (CVAE) [34,35] is trained to predict potency from chemical structure using the structural module of test compounds as input. Given the uniform structure–potency bit string encoding, SPFP–CVAE models do not depend on class labels or associated variables for learning.

2. Materials and Methods

2.1. Compound Activity Classes

Bioactive compounds were extracted from ChEMBL (version 28) [36]. The compounds with a reported direct target interaction (target confidence score: 9) and a numerically specified potency (pIC50) value (standard relation: "=") were initially retrieved. Then, the compounds with a molecular weight less than 1000 Da and potency values falling into the pIC50 range from 5 to 11 were selected. All the compounds with interactions labeled as "inactive", "not active", "inconclusive", "potential transcription error", or "pan assay interference compounds" (PAINS) [37] were discarded. Furthermore, the PAINS filter from RDKit, a filter based on liability rules from medicinal chemistry [38], and the aggregation advisor [39] were applied to remove compounds with potential assay interference characteristics. On the basis of these criteria, 132,175 unique compounds were obtained with activity against 1315 human targets. The qualifying compounds were organized into target-based activity classes (pharmaceutical anti-targets were omitted). A set of 10 activity classes was randomly selected from the large pool, comprising 18,231 unique compounds (Table 1) and used for activity class-based model building, hyperparameter optimization, and model evaluation.

Table 1. Activity classes. Ten activity classes used for deriving and evaluating activity class-based prediction models are reported.

Target Name	Target ID	# Compounds
Beta-secretase 1	4822	2270
11-beta-hydroxysteroid dehydrogenase 1	4235	2232
Phosphodiesterase 10A	4409	2109
Acetyl-CoA carboxylase 2	4829	1811
Dipeptidyl peptidase IV	284	1709
Sodium channel protein type IX alpha subunit	4296	1703
Tyrosine-protein kinase SYK	2599	1616
Vascular endothelial growth factor receptor 2	279	1614
Epidermal growth factor receptor erbB1	203	1606
Vanilloid receptor	4794	1562

2.2. Model Building and Evaluation

For each activity class, training and test sets were randomly assembled to yield a constant 90:10 compound partition. Across all models, the predictive performance was evaluated over 10 independent trials using different performance measures. For 80:20 compound data partitions used as a control, nearly identical results were obtained.

2.2.1. Conditional Variational Autoencoder

CVAE [40] is an adaptation of the variational autoencoder (VAE) [41], a supervised deep learning algorithm for generative modeling that constructs a conditioned data representation into a continuous latent variable (z). The probabilistic encoder $q(z \mid X, c)$ (recognition network) uses a condition vector (c) to map the input data to a Gaussian distribution, $p(z \mid c) \sim N(0, I)$ (prior network) into the latent space. The decoder $p(X \mid z, c)$ then reconstructs data samples from the conditioned latent space to obtain the original input representation (dimensionality). The encoder and decoder are trained with the objective of optimizing the evidence lower bound (ELBO) of the input data [42,43]. During training, the conditioned encoder learns to approximate a latent variable distribution by minimizing the Kullback–Leibler (KL) divergence [44] between data distributions in the original and latent space. The decoder is trained to minimize the reconstruction error of the data representation.

The CVAE encoder and decoder networks consisted of three hidden layers, with 512, 256, and 128 neurons, respectively. For hyper-parameter optimization, a grid search protocol was applied to determine the number of neurons for the latent layer (16, 32, and 64). Different learning rates (0.1, 0.01, and 0.001), dropout rates (0 and 0.5), and batch sizes (16, 32, and 64) were evaluated. Network training was performed with Adam [45] optimizer and the hyperbolic tangent (tanh) was used as the activation function. The parameters β (1 and 2) and σ (0.01, 0.1, and 1) were tested. The learning rate was steadily reduced, during training, to improve learning, and the models were run for a maximum of 150 epochs or until convergence was reached with the early stopping option to avoid network overfitting. The CVAE cost function was computed as the mean of the reconstruction (binary cross-entropy) loss and KL divergence loss.

2.2.2. Support Vector Regression

The support vector regression (SVR) is a variant of the supervised support vector machine algorithm that derives an ε-insensitive tube based on the training data for the prediction of numerical values, with the maximum permitted error provided by the width of the ε tube [10,11].

For SVR, the cost hyper-parameter C was optimized by testing (0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 10, 100, and 10000) values. The SVR models were built using the Tanimoto kernel [46].

2.2.3. Random Forest Regression

Random forest regression (RFR) is a machine-learning algorithm based on an ensemble of decision trees. During model training, each tree is created by splitting the respective node and bootstrapping aggregation is used to randomly select the training instances. The mean value across all decision trees is used to determine the final prediction [47].

In RFR parameter optimization, the number of decision trees (50, 100, and 200), the minimal number of samples for a split (2, 3, 5, and 10), and the minimum number of leaf-node samples (1, 2, 5, and 10) were used as the search parameter space.

2.2.4. Deep Neural Network

DNN is a deep learning method capable of mathematically modeling data using a non-linear activation function through the neurons of the network's fully connected layers. The network learning process consists of interactively determining the difference between the observed and predicted values, using a stochastic gradient descent algorithm to minimize the loss function until it converges to a specific minimum value [48,49].

The DNN models were trained using several network architectures by varying the different numbers of hidden layers (2 and 3) with hyperbolic tangent (tanh) activation, and the network neurons (100–500). Grid searches were performed for different batch sizes (16 and 64), dropout (0 and 0.5), and learning rates (0.1, 0.01, and 0.001). The networks were trained using an Adam optimizer for a maximum of 200 epochs with early termination.

2.2.5. k-Nearest Neighbor Ranking

kNN is a supervised learning method that ranks the training compounds based on increasing the fingerprint similarity (shortest distance). For the final prediction, the k-top training compounds potency value is accessed (e.g., 1-NN—potency value, and 3-NN—average potency) and assigned to the test compound [50]. For kNN optimization, the optimal k values were evaluated with 1, 3, and 5 top-rated compounds.

2.2.6. Mean Regression

The mean regressor (MR) approach is based on assigning the mean potency value of the training set to each compound present in the test set. This method was used as a control calculation to generate the random predictions.

2.2.7. Random Predictions

A y-randomization control was performed by the random reassignment of potency values across the compounds from each activity class (random shuffling) [51].

2.2.8. Hyperparamters and Implementation

The kNN, SVR, RF, and SPFP–CVAE model hyperparameters were optimized using an internal five-fold cross-validation, whereas the DNN parameter optimization was performed with an internal 90:10 training–validation split. The SVR, RFR, kNN, and MR models were generated using scikit-learn [52]. The CVAE and DNN models were implemented with Keras [53] and Tensorflow [54].

2.2.9. Evaluation Metrics

The performances of all the models were evaluated by calculating the mean absolute error (MAE) and root mean squared error (RMSE) for predicted and experimental test compound potency values using scikit-learn.

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (1)$$

$$RMSE(y, \hat{y}) = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \quad (2)$$

where n is the number of compounds, and y and ŷ are the experimental and predicted potency values, respectively.

An assessment of the statistical significance was performed for the value distributions from predictions based on MAE and RMSE values using the nonparametric Wilcoxon test [55]. The null hypothesis was either rejected or accepted, by setting alpha to 0.05 and comparing it to the respective *p*-value ($p \leq 0.05$).

2.3. Molecular Representation

The compounds were represented using a folded version of the extended connectivity fingerprint with bond diameter 4 (ECFP4) [56], which is a generally preferred topological descriptor for many chemoinformatics applications, consisting of layered atom environments, consisting of 2048 bits. The ECFP4 fingerprint was generated using RDKit [57].

The scripts for the reported calculations and the curated activity classes are available from the authors upon request.

3. Results and Discussion

3.1. Concept of Potency Prediction Based on Fingerprint-Based Potency Encoding

The introduction of a structure–potency fingerprint (SPFP) provided the basis for a new approach in potency prediction. The underlying idea was to unify structural and potency encodings in a modular fingerprint representation of a constant format such that the potency module representing a numerical value could be predicted from the structural module of test compounds using deep learning. This unified and intuitive modular encoding of compound structure and potency enabled the derivation of a chemical language model such as a CVAE using SPFPs of training compounds to predict the potency module of test compounds using only their structure module as input.

An extended connectivity fingerprint with a constant size of 2048 bits represented the structure module of SPFP that was combined with a newly designed potency module for representing compound potency values. We defined two principal requirements for the potency module. Hence, it was required to, first, represent the biologically relevant large (negative decadic logarithmic) potency range from 5 to 11 and, second, encode the potency values at a meaningful resolution such that accurate predictions could in principle be obtained. Therefore, alternative single value, value range, and cumulative coding schemes suitable for bit string representations were initially investigated and cumulative value range encoding was found to be the most robust approach (that is, yielding the most stable predictions across independent trials). Accordingly, contiguous segments of increasing numbers of bits were used to represent increasingly potent compounds populating the entire logarithmic potency range from 5 to 11. For example, Figure 1a,b show how a potency value of 5.2 and 8.0 was encoded by setting on the first four and 51 bits in the potency module, respectively. To meet the second requirement stated above, we set the size of the potency module to a minimum of 100-bit positions such that each individual bit position accounted for 0.06 log units via cumulative potency encoding. Accordingly, the resolution of the potency predictions was intrinsically limited to 6% of a log unit. This level was deemed acceptable for the approach because it fell within the typical range of experimental accuracy limitations. Smaller bit numbers for the potency module would lead to larger resolution limits while larger numbers would further increase the resolution. Therefore, we also tested larger versions of the potency module using the SPFP–CVAE models comprising 500 and 1000 bits, as reported in Figure 2. These control calculations produced very similar results to those obtained for the 100-bit potency module, hence showing that the prediction accuracy could not be further increased by decreasing the resolution limit of the potency encoding and supporting the choice of 100-bit positions for the potency module. Furthermore, for potency predictions using CVAE sampling, a bit

module with a constant format and meaningful size was required to assess the predictions in a meaningful way (see below).

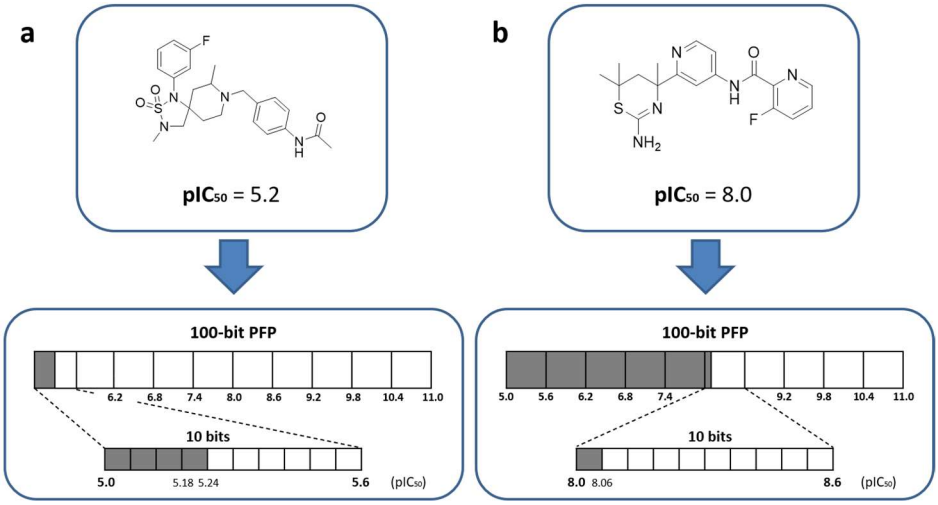

Figure 1. Cumulative potency encoding. (**a**,**b**) illustrate how logarithmic potency values of different magnitude are encoded in the potency module of SPFP.

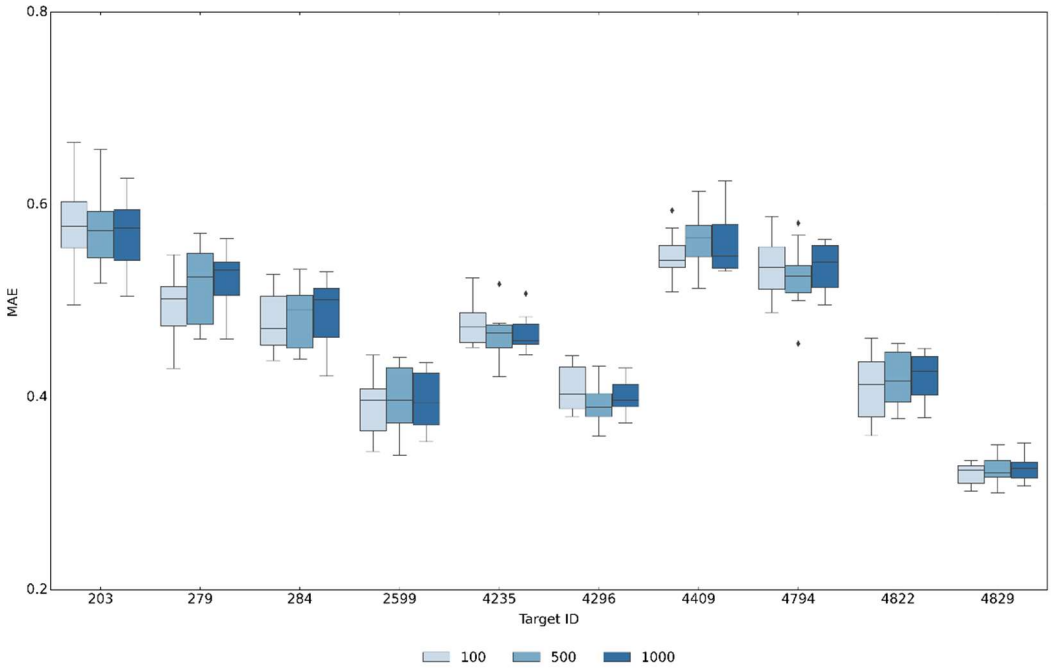

Figure 2. Prediction accuracy for SPSF with differently sized potency modules. Boxplots report mean absolute error (MAE) values of SPFP–CVAE models using alternative SPFP versions with potency modules comprising 100, 500, or 1000 bits evaluated across all activity classes according to Table 1. In boxplots, the upper and lower whiskers indicate maximum and minimum values, the boundaries of the box represent the upper and lower quartiles, values classified as statistical outliers are shown as diamonds, and the median value is indicated by a horizontal line.

3.2. Learning and Prediction Strategy

The CVAE model architecture used here consists of an encoder, latent space layer, and decoder, as illustrated in Figure 3.

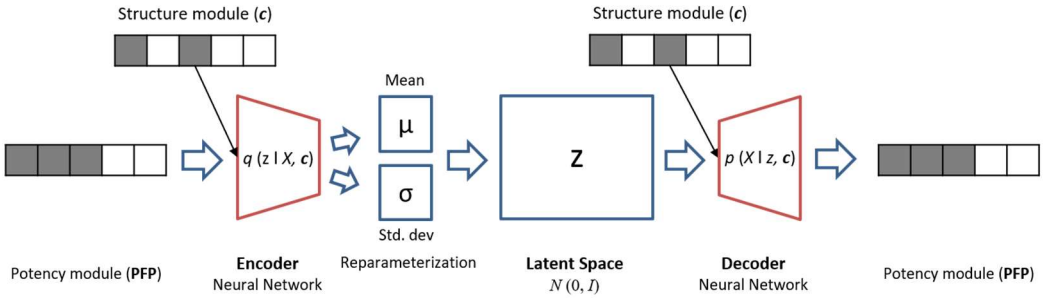

Figure 3. Architecture of the conditional variational autoencoder. The encoder network transforms the potency module (PFP), conditioned by the structure module (c), into a distribution of latent variables (z). The decoder samples a conditioned latent vector from a Gaussian distribution and reconstructs the potency module.

For each compound activity class, a CVAE model was trained to reproduce the complete bit patterns of the potency module, conditioned by the structure module, as illustrated in Figure 4a. Each CVAE model was then used to predict the bit settings of the potency module (PFP). Therefore, the potency values of the test compound were predicted by submitting the structure module (c) to the CVAE decoder to generate the corresponding potency module, as illustrated in Figure 4b. Since the CVAE predictions depended on the sampling of potency modules in latent space, the evaluation criteria for potency module variants were defined. Accordingly, for a given test compound, a sampled potency module was classified as valid if it contained a contiguous bit string in which all bits were set on. If this criterion was met, the predicted potency value was assigned to the center of the respective potency interval (e.g., 5.03 for the [5.0–5.06] interval), resulting in a constant standard deviation of ±0.03 log units for all predictions. By contrast, if the output bits were not contiguous, that is, if they were not consistent with the cumulative encoding of the potency module, the prediction was classified as invalid and the sampling was continued until a valid prediction was obtained, given a maximal number of permitted sampling steps.

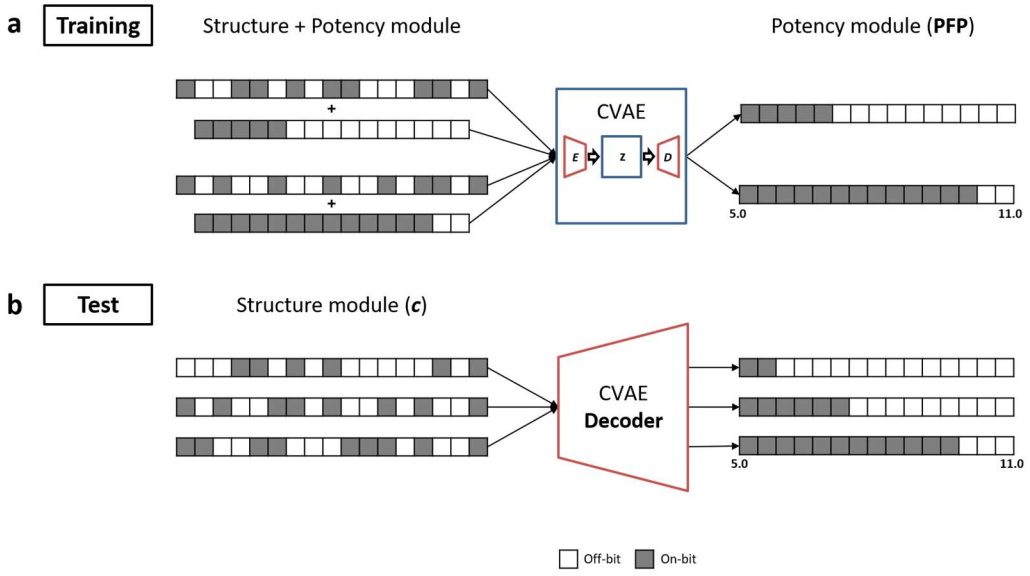

Figure 4. Conditional variational autoencoder modeling. In (**a**,**b**), the CVAE training and prediction strategies are illustrated, as discussed in the text.

3.3. Potency Predictions

For 10 randomly selected compound activity classes, different ML models were generated. Figure 5 shows that the compound potency value distributions of the activity classes were overlapping yet distinct, mostly yielding median potency values in the high

nanomolar range. The comparison also shows that logarithmic potency values below 5 (approaching experimental accuracy limitations) and above 10 (sub-nanomolar potency) were generally sparse.

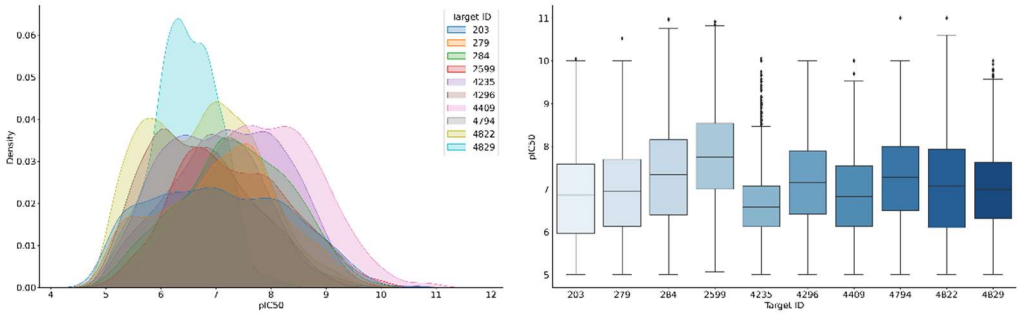

Figure 5. Potency value distribution of activity classes. Kernel density estimation plots (**left**) and boxplots (**right**) compare the potency values distributions of the 10 activity classes. Coloring of boxplots is arbitrary. The horizontal line indicates the median of the value distribution and diamond symbols represent statistical outliers.

Activity class-dependent potency prediction models were then generated for SPFP–CVAE, k-nearest neighbor (kNN) analysis, SVR, RFR, and DNN. These ML approaches currently represent the state of the art in compound potency prediction [31]. In addition, a mean regressor (MR) was applied as a control, which simply assigned the mean potency value of an activity class to all test compounds. The results are reported in Figure 6.

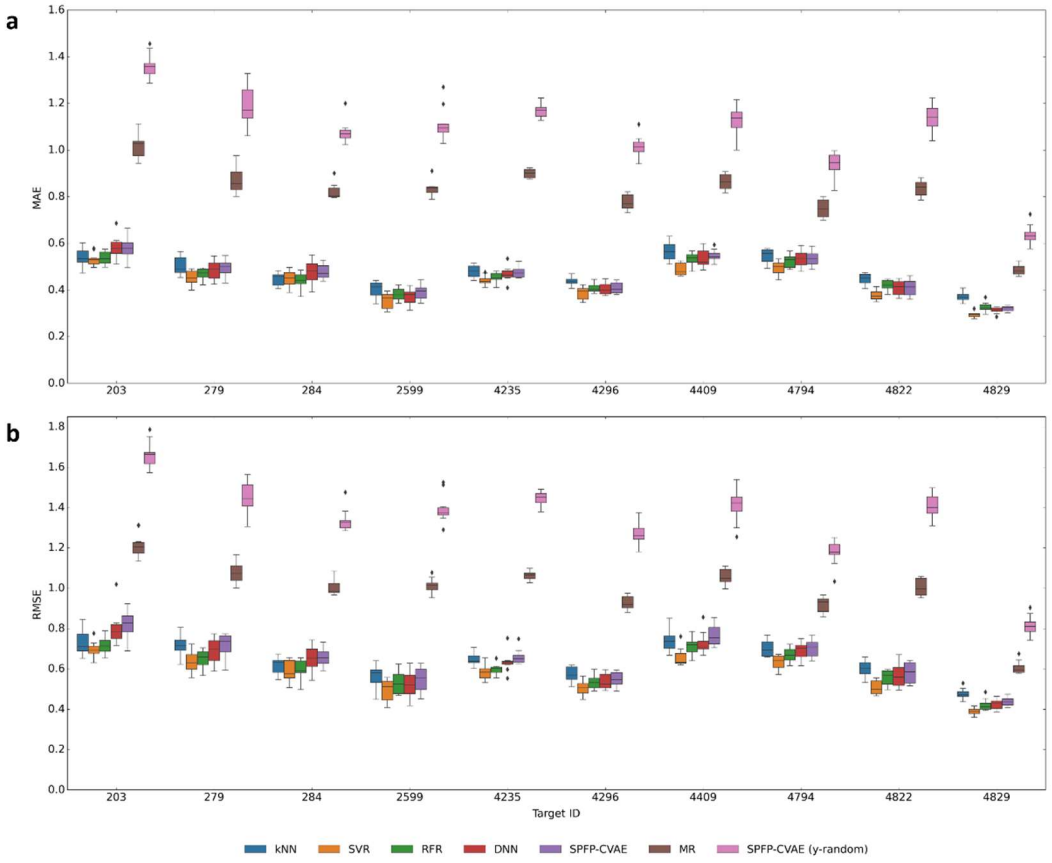

Figure 6. Prediction accuracy. Boxplots report the (**a**) MAE and (**b**) RMSE values for potency predictions using different ML models across all activity classes.

Overall, similar prediction accuracy was observed for the different ML models, regardless of their complexity, mostly with median MAE and RMSE of ~0.4–0.5 and ~0.6–0.7, respectively. As observed before [31], simple kNN-based potency assignments approached or exceeded the prediction accuracy of ML models and there was no advantage of deep learning approaches over other ML methods. Moreover, the MR yielded median MAE and RMSE values of ~0.8–0.9 and ~1.0–1.1, respectively. The performance of randomized SPFP–CVAE models was only slightly worse than MR, mostly with a median MAE value of ~1.0–1.2, owing to the dominance of compounds with potency values between 6 and 8 across all activity classes, as reported in Figure 5. These artificial predictions using MR or randomized models reproduced experimental values within about one order of magnitude, providing a limit for prediction accuracy, while most accurate ML models typically achieved mean MAE value of ~0.4. Hence, there was only a relatively small margin between best and artificial predictions, defining a window of less than one order of magnitude in which model performance must be evaluated [31]. In the previous study, equally curated versions of three activity classes (279, 284, 4822) from a different ChEMBL release were investigated using ML methods with different calculation protocols, yielding prediction accuracies very similar to the values reported herein [31].

Many of the small performance differences observed in Figure 6 were not statistically significant, as reported in Figure 7, while differences between SPFP–CVAE, SVR, and RFR were statistically significant for about half of the activity classes. However, the prediction accuracy of these three approaches was very similar, which was also reflected by the respective *p*-values. Overall, SVR was the preferred approach, but only by a very small margin compared to SPFP–CVAE and other ML methods. For example, the differences in the median MAE between SPFP–CVAE and SVR ranged max. ~0.01–0.02, depending on the activity class, which was marginal at most and would be considered irrelevant for all practical purposes.

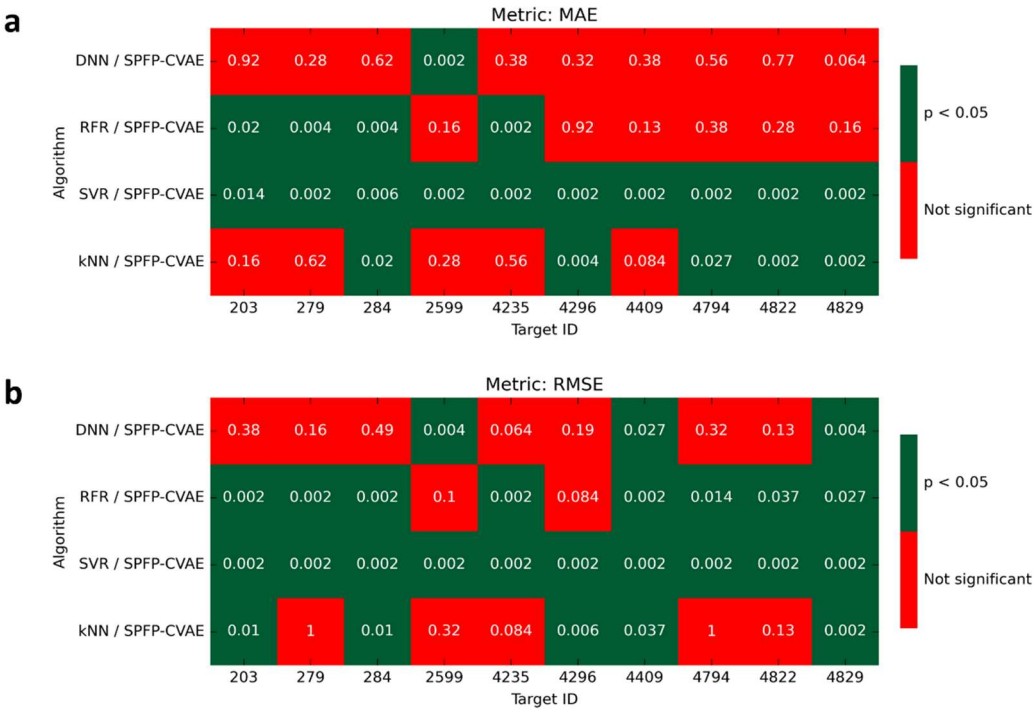

Figure 7. Statistical significance assessment. Statistical significant (Wilcoxon signed-rank) tests based on (**a**) MAE and (**b**) RMSE values were carried out for the performance differences observed between SPFP–CVAE and all other ML models (kNN, SVR, RFR, and DNN). Red cells indicate *p*-values above $\alpha = 0.05$ (no statistical significance) and green cells *p*-values below $\alpha = 0.05$ (statistical significance).

3.4. Predicting Highly Potent Compounds

We then investigated the ability of the different ML methods to predict the 10% most potent compounds in a test set (typically amounting to ~15–20 compounds) using models derived based on the original sets. The results for models derived from original training sets are shown in Figure 8. Due to the small test sample size of these predictions, the MAE and RMSE value distributions were broader than for the global predictions reported in Figure 6.

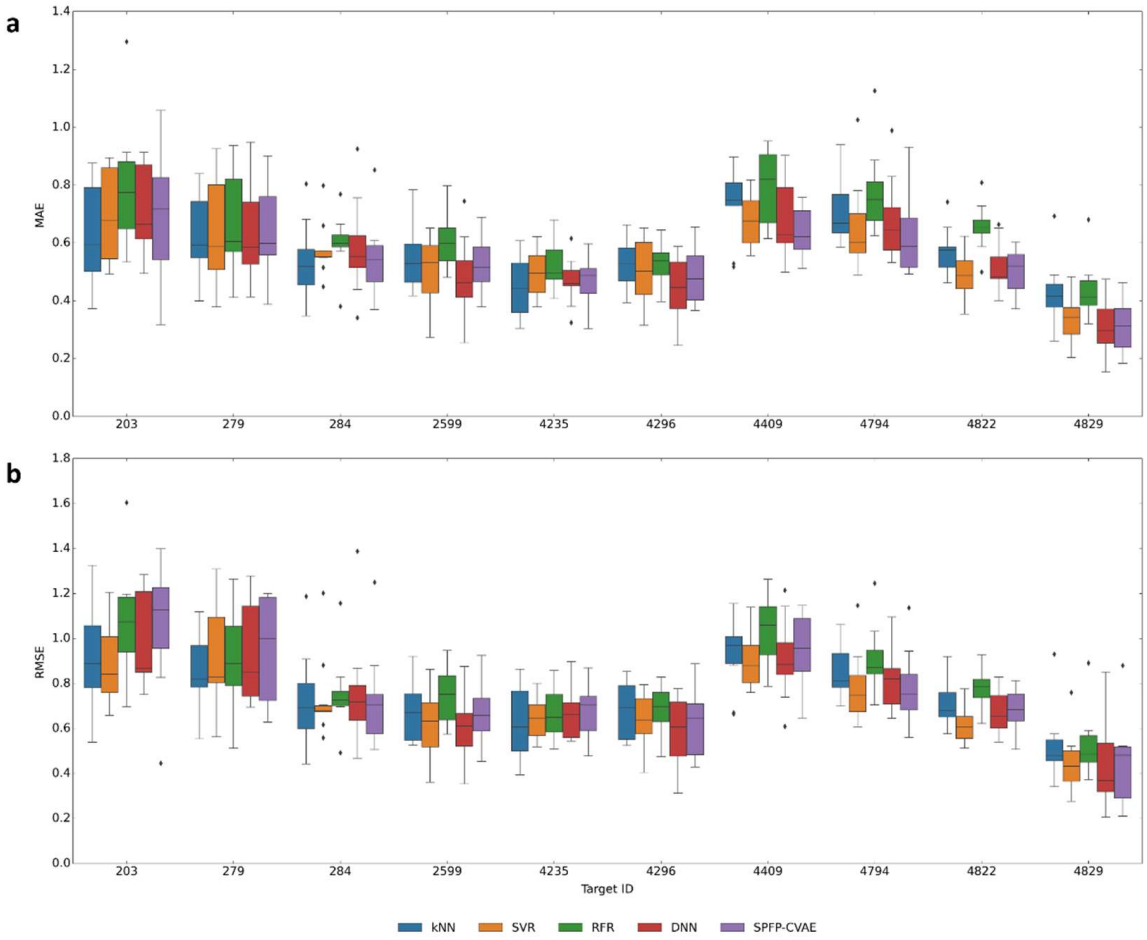

Figure 8. Prediction accuracy for the most potent test compounds. Boxplots report the median MAE (**a**) and RMSE (**b**) values for the 10% most potent test compounds from all classes and different ML models including kNN.

Compared to the global predictions, the median MAE and RMSE value for most potent compounds increased to ~0.6 and ~0.8 or greater, respectively, for about half of the activity classes while the prediction accuracy remained similar to before for the remaining classes. However, the performance of the different ML methods including kNN continued to be comparable (MR was omitted here because of the naturally large deviations for the small number of the most potent compounds). Overall, SVR, SPFP–CVAE, and DNN yielded best predictions with only small (and activity class-dependent) differences between these methods. The predictions for the exemplary compounds are shown in Figure 9.

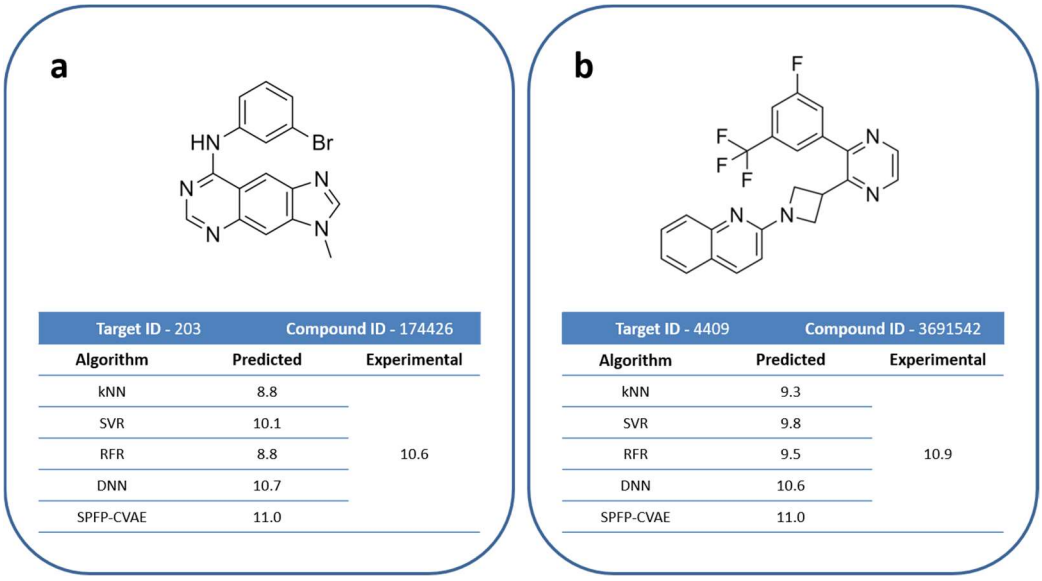

Figure 9. Highly potent compounds. (**a**,**b**) illustrate predictions for two exemplary highly potent test compounds using different methods.

4. Conclusions

Compound potency prediction is an important task in chemoinformatics and medicinal chemistry. For structure- and ligand-based predictions, different methods have been introduced. QSAR techniques including non-linear modeling using machine learning continue to play an important role. Herein, we have introduced a new methodological concept for compound potency prediction that depends on the newly designed SPFP format for structure–potency encoding and CVAE learning. The SPFP–CVAE concept was devised to enable the prediction of bit settings in SPFP potency modules from input structure modules, without learning correlations between structural representations and potency values used as a dependent variable. In activity class-dependent predictions, the SPFP–CVAE approach essentially met SVR performance, representing the current state of the art in the field. Given the general limitations associated with the potency predictions in benchmark settings, we consider the prediction of most potent compounds a particularly meaningful exercise. In this case, SVR, SPFP–CVAE, and DNN achieved comparable accuracy. Taken together, our results indicate that the SPFP–CVAE concept introduced herein provides a new methodological framework for compound potency prediction that can be further explored in various ways. Importantly, FP-based structure–potency encoding, as introduced herein, can be easily modified for different applications, providing a versatile input format for ML.

Author Contributions: T.J. and K.T.: Methodology, Resources, Investigation, and Formal analysis, Writing—review and editing; J.B.: Conceptualization, Methodology, Formal analysis, Writing—original draft, and Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Compound data sets are publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mobley, D.L.; Gilson, M.K. Predicting Binding Free Energies: Frontiers and Benchmarks. *Annu. Rev. Biophys.* **2017**, *46*, 531–558. [CrossRef]
2. Williams-Noonan, B.J.; Yuriev, E.; Chalmers, D.K. Free Energy Methods in Drug Design: Prospects of "Alchemical Perturbation" In Medicinal Chemistry. *J. Med. Chem.* **2018**, *61*, 61638–61649. [CrossRef]
3. Liu, J.; Wang, R. Classification of Current Scoring Functions. *J. Chem. Inf. Model.* **2015**, *55*, 475–482. [CrossRef]
4. Gleeson, M.P.; Gleeson, D. QM/MM Calculations in Drug Discovery: A Useful Method for Studying Binding Phenomena? *J. Chem. Inf. Model.* **2009**, *49*, 670–677. [CrossRef]
5. Guedes, I.A.; Pereira, F.S.S.; Dardenne, L.E. Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges. *Front. Pharmacol.* **2018**, *9*, e1089. [CrossRef]
6. Li, H.; Sze, K.H.; Lu, G.; Ballester, P.J. Machine-Learning Scoring Functions for Structure-Based Virtual Screening. *WIREs Comput. Mol. Sci.* **2021**, *11*, e1478. [CrossRef]
7. Lewis, R.A.; Wood, D. Modern 2D QSAR for Drug Discovery. *WIREs Comput. Mol. Sci.* **2014**, *4*, 505–522. [CrossRef]
8. Akamatsu, M. Current State and Perspectives of 3D-QSAR. *Curr. Top. Med. Chem.* **2002**, *2*, 1381–1394. [CrossRef]
9. Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958. [CrossRef]
10. Drucker, H.; Burges, C. Support Vector Regression Machines. *Adv. Neural. Inform. Proc. Syst.* **1997**, *9*, 155–161.
11. Smola, A.J.; Schölkopf, B. A Tutorial on Support Vector Regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
12. Balfer, J.; Bajorath, J. Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis. *PloS ONE* **2015**, *10*, e0119301. [CrossRef]
13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
14. Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; et al. Applications of Machine Learning in Drug Discovery and Development. *Nat. Rev. Drug Discov.* **2019**, *18*, 463–477. [CrossRef]
15. Lavecchia, A. Deep Learning in Drug Discovery: Opportunities, Challenges and Future Prospects. *Drug Discov. Today* **2019**, *24*, 2017–2032. [CrossRef]
16. Bajorath, J. Deep Machine Learning for Computer-Aided Drug Design. *Front. Drug Discov.* **2022**, *2*, e829043. [CrossRef]
17. Kim, J.; Park, S.; Min, D.; Kim, W.Y. Comprehensive Survey of Recent Drug Discovery Using Deep Learning. *Int. J. Mol. Sci.* **2019**, *22*, e9983. [CrossRef]
18. Jimenez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58*, 287–296. [CrossRef]
19. Torng, W.; Altman, R.B. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *J. Chem. Inf. Model.* **2019**, *59*, 4131–4149. [CrossRef]
20. Kwon, Y.; Shin, W.H.; Ko, J.; Lee, J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. *Int. J. Mol. Sci.* **2020**, *21*, e8424. [CrossRef]
21. Son, J.; Kim, D. Development of a Graph Convolutional Neural Network Model for Efficient Prediction of Protein-Ligand Binding Affinities. *PLoS ONE* **2021**, *16*, e0249404. [CrossRef]
22. Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C.J.; Duca, J.S.; Hornak, V.; Koes, D.R.; Kurtzman, T. Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening. *PLoS ONE* **2019**, *14*, e0220113. [CrossRef]
23. Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* **2020**, *11*, e69. [CrossRef]
24. Volkov, M.; Turk, J.A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein-Ligand Structures with Deep Neural Networks. *J. Med. Chem.* **2022**, *65*, 7946–7958. [CrossRef]
25. Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization rather than Generalization. *J. Chem. Inf. Model.* **2021**, *58*, 916–932. [CrossRef]
26. Hou, F.; Wu, Z.; Hu, Z.; Xiao, Z.; Wang, L.; Zhang, X.; Li, G. Comparison Study on the Prediction of Multiple Molecular Properties by Various Neural Networks. *J. Phys. Chem. A* **2018**, *122*, 9128–9134. [CrossRef]
27. Feinberg, E.N.; Sur, D.; Wu, Z.; Husic, B.E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V.S. PotentialNet for Molecular Property Prediction. *ACS Cent. Sci.* **2018**, *4*, 1520–1530. [CrossRef]
28. Shen, J.; Nicolaou, C.A. Molecular Property Prediction: Recent Trends in the Era of Artificial Intelligence. *Drug Discov. Today Technol.* **2019**, *32*, 29–36. [CrossRef]
29. Walters, W.P.; Barzilay, R. Applications of Deep Learning in Molecule Generation and Molecular Property Prediction. *Acc. Chem. Res.* **2020**, *54*, 263–270. [CrossRef]
30. Bajorath, J. State-of-the-Art of Artificial Intelligence in Medicinal Chemistry. *Future Sci. OA* **2021**, *7*, FSO702. [CrossRef]
31. Janela, T.; Bajorath, J. Simple Nearest Neighbor Analysis Meets the Accuracy of Compound Potency Predictions Using Complex Machine Learning Models. *Nat. Mach. Intell.* **2022**, *4*, 1246–1255. [CrossRef]
32. Willett, P. Similarity-Based Virtual Screening Using 2D Fingerprints. *Drug Discov. Today* **2006**, *11*, 1046–1053. [CrossRef]
33. Vogt, M.; Stumpfe, D.; Geppert, H.; Bajorath, J. Scaffold Hopping Using Two-Dimensional Fingerprints: True Potential, Black Magic, or a Hopeless Endeavor? Guidelines for Virtual Screening. *J. Med. Chem.* **2010**, *53*, 5707–5715. [CrossRef]

34. Gómez-Bombarelli, R.; Wei, J.N.; Duvenaud, D.; Hernández-Lobato, J.M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T.D.; Adams, R.P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. [CrossRef]
35. Blaschke, T.; Olivecrona, M.; Engkvist, O.; Bajorath, J.; Chen, H. Application of Generative Autoencoder in De Novo Molecular design. *Mol. Inform.* **2018**, *37*, e1700123. [CrossRef]
36. Bento, A.P.; Gaulton, A.; Hersey, A.; Bellis, L.J.; Chambers, J.; Davies, M.; Krüger, F.A.; Light, Y.; Mak, L.; McGlinchey, S.; et al. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, D1083–D1090. [CrossRef]
37. Baell, J.B.; Holloway, G.A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53*, 2719–2740. [CrossRef]
38. Bruns, R.F.; Watson, I.A. Rules for Identifying Potentially Reactive or Promiscuous Compounds. *J. Med. Chem.* **2012**, *55*, 9763–9772. [CrossRef]
39. Irwin, J.J.; Duan, D.; Torosyan, H.; Doak, A.K.; Ziebart, K.T.; Sterling, T.; Tumanian, G.; Shoichet, B.K. An Aggregation Advisor for Ligand Discovery. *J. Med. Chem.* **2015**, *58*, 7076–7087. [CrossRef]
40. Sohn, K.; Lee, H.; Yan, X. Learning Structured Output Representation Using Deep Conditional Generative Models. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, Canada, 7–12 December 2015; pp. 3483–3491.
41. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114. Available online: https://arxiv.org/abs/1312.6114 (accessed on 1 June 2022).
42. Doersch, C. Tutorial on Variational Autoencoders. *arXiv* **2016**, arXiv:1606.05908. Available online: https://arxiv.org/abs/1606.05908 (accessed on 1 May 2022).
43. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Proceedings of the 31st International Conference on Machine Learning (ICML), Beijing, China, 21–26 June 2014; pp. 3057–3070.
44. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
45. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
46. Ralaivola, L.; Swamidass, S.J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Netw.* **2005**, *18*, 1093–1110. [CrossRef]
47. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
48. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
49. Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2015.
50. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185.
51. Rücker, C.; Rücker, G.; Meringer, M. y-Randomization and its Variants in QSPR/QSAR. *J. Chem. Inf. Model.* **2007**, *47*, 2345–2357. [CrossRef]
52. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
53. Chollet, F.K. Keras. Available online: https://github.com/fchollet/keras (accessed on 30 July 2022).
54. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
55. Conover, W.J. On Methods of Handling Ties in the Wilcoxon Signed-Rank Test. *J. Am. Stat. Assoc.* **1973**, *68*, 985–988. [CrossRef]
56. Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754. [CrossRef]
57. RDKit: Cheminformatics and Machine Learning Software. 2013. Available online: http://www.rdkit.org (accessed on 1 July 2022).