Archives of Biochemistry and Biophysics 698 (2021) 108730

Contents lists available at ScienceDirect

Archives of Biochemistry and Biophysics

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

Review article

Artificial intelligence in the early stages of drug discovery

Claudio N. Cavasotto [a,b,c,*], Juan I. Di Filippo [a,c]

[a] Computational Drug Design and Biomedical Informatics Laboratory, Translational Medicine Research Institute (IIMT), CONICET-Universidad Austral, Pilar, Buenos Aires, Argentina

[b] Facultad de Ciencias Biomédicas, and Facultad de Ingeniería, Universidad Austral, Pilar, Buenos Aires, Argentina

[c] Austral Institute for Applied Artificial Intelligence, Universidad Austral, Pilar, Buenos Aires, Argentina

ARTICLE INFO

Keywords:
Artificial Intelligence
Drug discovery
Machine learning
Deep learning
Target identification
Hit and lead identification
Property prediction

ABSTRACT

Although the use of computational methods within the pharmaceutical industry is well established, there is an urgent need for new approaches that can improve and optimize the pipeline of drug discovery and development. In spite of the fact that there is no unique solution for this need for innovation, there has recently been a strong interest in the use of Artificial Intelligence for this purpose. As a matter of fact, not only there have been major contributions from the scientific community in this respect, but there has also been a growing partnership between the pharmaceutical industry and Artificial Intelligence companies. Beyond these contributions and efforts there is an underlying question, which we intend to discuss in this review: can the intrinsic difficulties within the drug discovery process be overcome with the implementation of Artificial Intelligence? While this is an open question, in this work we will focus on the advantages that these algorithms provide over the traditional methods in the context of early drug discovery.

1. Introduction

It takes on average at least 10 years to get a new drug to the market, with an associated cost which can top U$S 3 billion [1]. This long and expensive process comprises a chain of complex procedures, which can be roughly divided into four major steps: (i) Early stage: target identification and validation, hit discovery and lead optimization; (ii) preclinical studies; (iii) clinical trials: phases I, II and III; (iv) FDA review, and post-approval research and monitoring.

The task of finding a new drug is often described as finding a needle in a haystack. This is mainly because of the large size of the chemical space, which is estimated to be $\sim 10^{60}$ drug-like molecules [2]. For practical purposes, this space could be considered as infinite, since even exploring millions of molecules per second, the age of the universe would not be enough to search the totality of chemical entities.

Initially, hit and lead identification were mainly dominated by high-throughput screening (HTS). This is an experimental and time-intensive technique, which exhibits several drawbacks, such as stagnant success rates coupled with false positives and negatives [3]. The availability of high quality three-dimensional (3D) structures of protein–ligand (PL) complexes opened the door to structure-based approaches, which was complemented by the development of computational methods to rapidly screen *in silico* chemical libraries against a given target to prioritize compounds for bioevaluation. These structure-based virtual screening (SBVS) methods have since evolved in terms of algorithm development, computational efficiency, and applications to drug design [4,5]. Similar advances have been observed for ligand-based virtual screening (LBVS) approaches [6,7]. Today, computer-aided drug discovery is an established and consolidated tool in the drug development process [5,8,9].

Artificial Intelligence (AI) is perceived as one of the main disruptive technologies of our decade. It encompasses a set of computational algorithms that allow machines and computers to simulate human cognitive abilities such as learning from experience and solving problems. Traditional computational methods rely on manually programmed logical operations to process information, and are used for optimizing tasks that are difficult or time consuming for the human intelligence, such as performing a linear regression on a set of data. AI, on the contrary, is associated with tasks where is highly non-trivial to describe the solution of a problem with handcrafted rules; for example, recognizing dogs in pictures.

Two sub-fields of AI are Machine Learning (ML) and Deep Learning (DL). Both are useful to automatically map a set of inputs to a set of outputs (supervised learning), and to learn underlying relationships in a given set of data (unsupervised learning). Moreover, DL is a subfield

* Corresponding author at: Computational Drug Design and Biomedical Informatics Laboratory, Translational Medicine Research Institute (IIMT), CONICET-Universidad Austral, Pilar, Buenos Aires, Argentina.

E-mail addresses: CCavasotto@austral.edu.ar, cnc@cavasotto-lab.net (C.N. Cavasotto).

https://doi.org/10.1016/j.abb.2020.108730
Received 5 November 2020; Received in revised form 11 December 2020; Accepted 14 December 2020
Available online 19 December 2020

of ML, which involves the use of models with a greater amount of learnable parameters, in comparison to ML models, like deep neural networks or convolutional neural networks.

A variety of fields have shown a strong and increasing interest in AI; in fact, some tasks that are performed with AI nowadays would have been impossible a few years ago. Since recently, these techniques are also being incorporated in the drug discovery pipeline [10–12], in versatile ways, and promising results are showing that "AI has enormous potential to revolutionize drug discovery" [13]. This fact is also highlighted by the growing relationship between AI companies and the Pharmaceutical Industry [14]. Considering that, from a study using 21,143 compounds, the overall success rate for all drug design programs (from Phase I to drug approval) is ~5.2%, down from 11.2% in 2005 [15], it is thus clear that the use of AI is mainly driven by the urgent need to reduce attrition and costs.

In this work, we review key applications of AI to different aspects of early drug discovery [16], specifically, target identification and validation, hit and lead identification (virtual screening, drug repurposing, generative models and *de novo* design), and property prediction. The Reader may refer to excellent reviews about the use of AI in other stages which encompass applications to biological imaging analysis [10], selection of a population for clinical trials [14], planning and automating chemical synthesis, predicting biomarkers, and computational pathology [13].

2. Commonly used methods in AI

In this section, we present a basic description of some common AI methods used in drug discovery, focusing particularly on DL techniques developed in the last years. The goal is to offer a conceptual insight of these algorithms. Special focus is given to the neural network description because its understanding is fundamental to comprehend more complex methods. Plenty of resources are available that offer fully detailed descriptions of AI methods [11,17–19].

Neural Network (NN). A NN defines a mapping between inputs X and outcomes Y. This mapping depends on parameters called weights and the use of a function, called activation function, which can be chosen to give a non-linear character to the mapping. It is a learning algorithm in the sense that it modifies the value of its weights so that the application of NN to inputs X, results in the best possible approximation of Y. Given an input matrix X, where each row represents a data sample and the columns represent the features used to describe these samples, and a set of weights given by a matrix W, the NN "feedforwards" this input matrix as shown in Eq. (1), resulting in the hidden layer matrix H,

$$H = f(X \cdot W + b) \tag{1}$$

where f is the activation function and b, known as the bias, is another learnable weight. This result is again forwarded with another set of weights to give the final outcome $\hat{Y}$, a matrix containing in its rows the predicted outputs for each sample of the input matrix:

$$\hat{Y} = f'(H \cdot W' + b') \tag{2}$$

Fig. 1 presents a scheme of this process, in which one sample (represented by three features x_1, x_2 and x_3) is passed through the NN to give an outcome $\hat{y}$. Weights are proportional to edge widths.

Initially, weights can be assigned at random. To approximate predicted outcomes $\hat{Y}$ to the real outputs Y, a loss function $\mathcal{L}$ is defined, which takes into account the difference between these two matrices. A trivial implementation can be, for example, the following:

$$\mathcal{L} = \sum_i \sum_j \left| Y_{ij} - \hat{Y}_{ij} \right| \tag{3}$$

If the activation functions f and f' are differentiable, then $\mathcal{L} = \mathcal{L}(W, W', b, b')$ is differentiable and a minimum could be searched using, for example, a gradient descent algorithm, which relies on the partial derivatives of $\mathcal{L}$ with respect to the weights to update the weights in each step.

Deep Neural Network (DNN). A DNN is simply a NN with more than one hidden layer. Given an input matrix X, the first hidden layer, $H^{(1)}$, is computed as in the case of NN:

$$H^{(1)} = f^{(0)}(X \cdot W^{(0)} + b^{(0)}) \tag{4}$$

where $f^{(0)}$, $W^{(0)}$ and $b^{(0)}$ are, respectively, the activation function, the set of weights and the bias used to feedforward the input X and compute $H^{(1)}$. The following hidden layers are calculated with different sets of learnable parameters and activation functions in the same manner:

$$H^{(i)} = f^{(i-1)}(H^{(i-1)} \cdot W^{(i-1)} + b^{(i-1)}) \tag{5}$$

where i = 2,..N, being N the amount of hidden layers. Finally, the predicted outcome $\hat{Y}$ is given by:

$$\hat{Y} = f^{(N)}(H^{(N)} \cdot W^{(N)} + b^{(N)}) \tag{6}$$

A basic scheme of this process is shown in Fig. 2.

Recurrent Neural Network (RNN). When NN are extended to include feedback connections (Fig. 3), they are called recurrent neural networks (RNN). This type of NN is optimal for sequence data problems, like for example, time series and natural language processing (NLP).

Convolutional Neural Network (CNN). This is a kind of neural network for processing data that has a grid-like topology, like for example, images. Unlike NNs, instead of performing a common matrix multiplication between the input matrix and weights matrices (called filters in this case), matrices are element-wise multiplicated. Given two matrices X and W, the element-wise multiplication consists in the following operation:

$$X * W = \sum_i \sum_j X_{ij} \cdot W_{ij} \tag{7}$$

This operation is obviously defined between two matrices that have the same dimensions, but usually, filters are defined with a lower dimension than the input matrix. In this case, the filters are element-wise multiplicated with sub-matrices from the input matrix, which is known as the convolution operation, from which the methodology receives its name (see Fig. 4).

The filters on the first layer, for example, detect local information around the input image and compute high values when the specific pattern they are looking for is in the convoluted region. The type of patterns that the filters are designed to detect are automatically learned in the training process. In the case of images, filters from the first layers generally detect basic shapes, like edges, and going deeper into the network, they can detect objects, like noses or ears.

Several filters may be applied per layer, and the outcoming results stacked in high-dimensional tensors. This high dimensional data can be manipulated by performing some operations like pooling or flattening. The CNN is generally used as a feature extractor, and its outputs are generally merged with another model to make predictions, like for example, a fully connected neural network. Fig. 5 shows an example in which a CNN architecture is used to predict which number is written in an image.

Multi-task learning (MTL). In some scenarios one could be interested in predicting, for a series of inputs X, not one outcome Y, but several outcomes $Y_1, Y_2, \ldots, Y_N$. One approach, considered as single-task learning, would consist in training N models, each one for each task. Another option, MTL, consists in training one unique model that shares information between tasks. In general, the use of related tasks enables the model to improve its performance. One way of implementing this method is training a unique DNN responsible of predicting all outcomes Y_1, Y_2, …, Y_N, in which the input matrix is feedforwarded through some hidden layers, but at some point the architecture is split, as shown in Fig. 6, such that each task has assigned specific hidden layers to predict the corresponding output values.

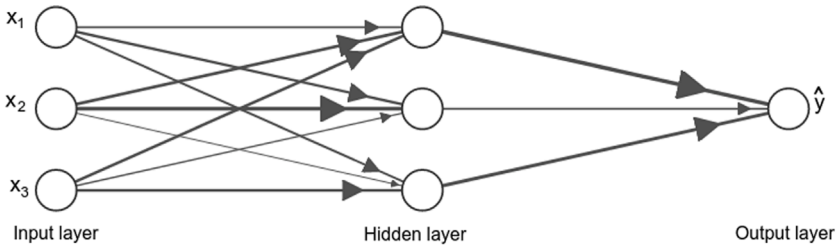

Fig. 1. Neural network architecture.

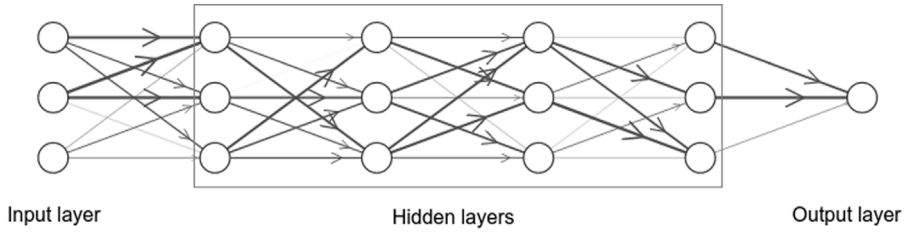

Fig. 2. Deep neural network architecture.

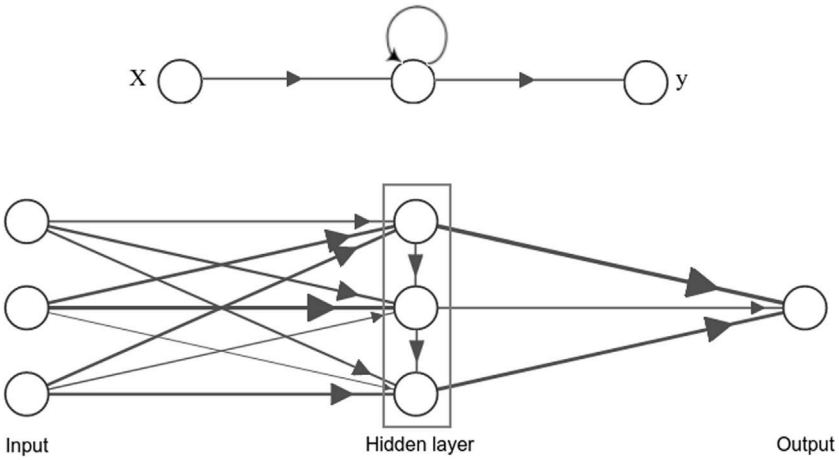

Fig. 3. Recurrent neural network architecture. Top: Reduced scheme. Bottom: Unfolded scheme.

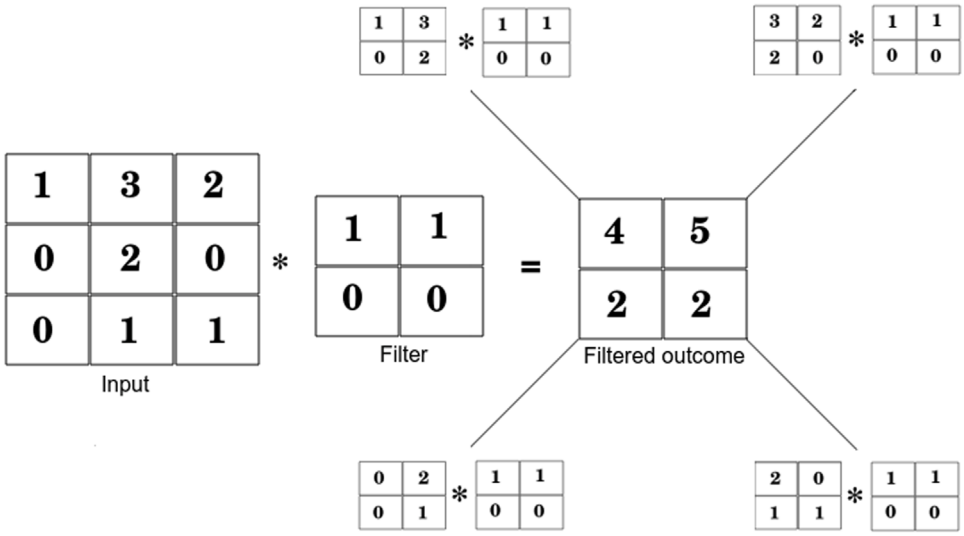

Fig. 4. The convolution operation.

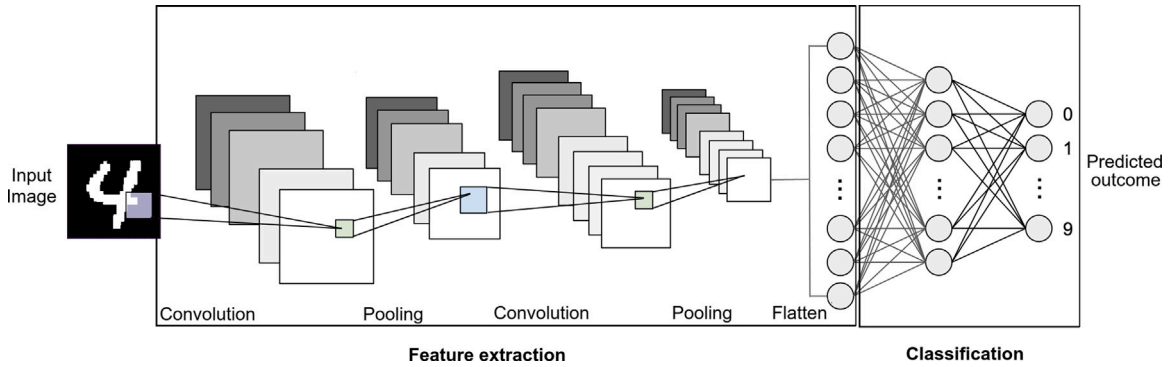

Fig. 5. Convolutional neural network example.

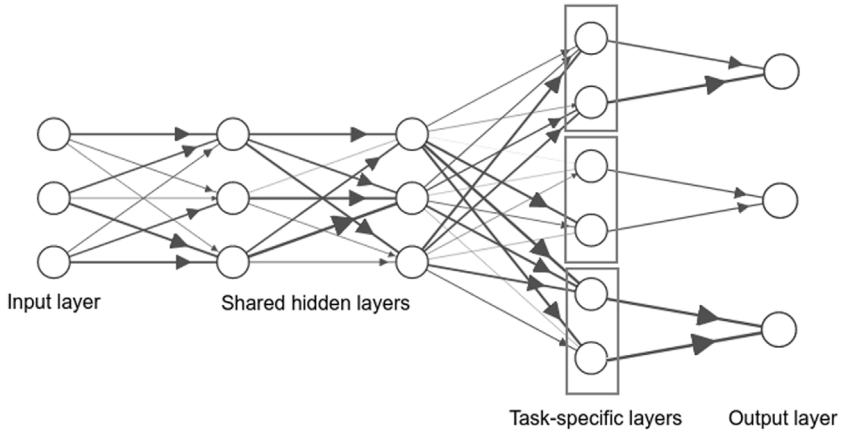

Fig. 6. Multi-task learning scheme.

Generative Modeling. Given a set of data $\mathcal{D}$, for example, a set of images containing handwritten numbers (Fig. 7), generative models are methodologies which can generate new $\mathcal{D}$-like samples after being trained with the dataset $\mathcal{D}$. Thus, a generative model trained with the handwritten images dataset could be used to generate images that look like handwritten numbers but that are not exactly the same as the ones present in the training dataset.

Generative Adversarial Networks (GAN). It is a method used to perform generative modeling where the general architecture is composed of two models that compete with each other. As shown in Fig. 8, one of them generates data (the generator) and the other discriminates if this data is real or fake based on a set of samples that has already been seen (the discriminator). This architecture can reach a state, after the training procedure, in which the discriminator cannot distinguish any more between real data and generated one.

AutoEncoder (AE). It is a method used to represent data in a low dimensional space called the latent space. It is composed by an encoder, that maps inputs onto the latent space, and a decoder, that maps points from the latent space back to the input format. A straightforward way of implementing an AE (Fig. 9) is to train a NN to reproduce the input data with a low dimensional hidden layer.

Variational AutoEncoder (VAE). Introducing certain modifications to the AE, it is possible to sample new points from the latent space to generate data. Basically, VAEs are generative AE.

Reinforcement Learning (RL). A RL setup is composed by an agent and an environment that interact, in a finite loop, as shown in Fig. 10 . The environment gives the agent the current state, and in return, the agent takes an action. The environment gives the agent a reward, based on the state and the action the agent took, and the new state. The objective is to maximize the sum of obtained rewards. For this purpose, a function that specifies which action must be taken in each state (policy) is optimized.

3. AI methods in early drug discovery

Given a disease of interest, the first fundamental step in drug development is the identification of an associated molecular target, whose modulation with a drug may affect the disease state (target identification). Generally speaking, the next step is the screening of small-molecules which could bind to and modulate the target (hit and lead identification), in order to later progress towards preclinical and clinical stages, and finally become a commercialized drug.

3.1. Target identification and validation

Considering the amount of publicly available biological data, AI is an ideal candidate to identify molecular targets, provided of course that these data-driven approaches are validated experimentally. In the following text we present some applications with focus on how data was incorporated into the AI benchmark.

In a recent study, Ferrero et al. [20] worked on the identification of therapeutic targets based on gene-disease association data extracted from the Open Targets platform [21]. Particularly, the authors utilized a set of 18,104 genes, each one related to a series of diseases, with the objective of predicting the probability that a given gene could be a drug–target using a ML model.

For each gene-disease pair, five features were provided in terms of a gene-disease association score (rather than in binary format), reflecting the presence of any of the following: (i) an animal model with a knockout-gene that manifests a phenotype concordant with the human disease, (ii) a germline mutation in the gene associated with the disease, (iii) a significant gene expression change in the disease, (iv) a somatic mutation in the gene associated with the disease, (v) if the gene is part of a pathway that is affected in the disease. For each

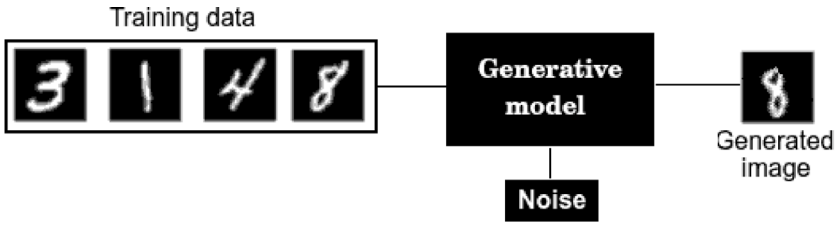

Fig. 7. Generative modeling example.

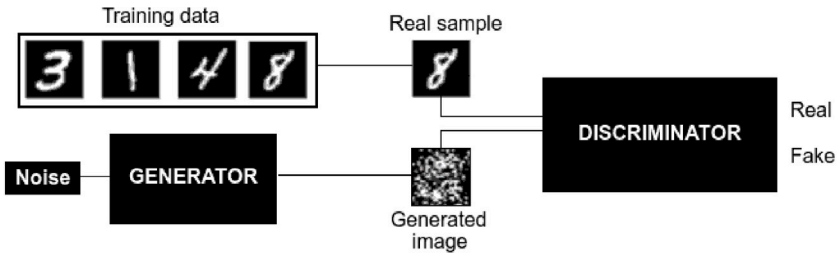

Fig. 8. Generative Adversarial Network example.

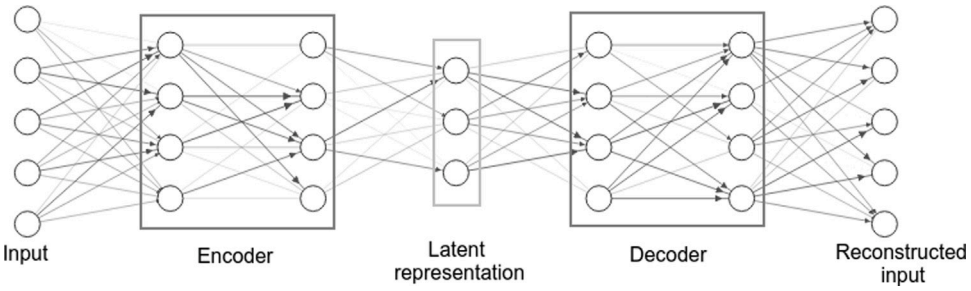

Fig. 9. Autoencoder architecture.

gene, the scores for each data type were averaged across all associated diseases, which resulted in five averaged scores per gene that were used as input features. For model outputs, genes were labeled as "target" if they were found in any of the following categories from the Informa Pharmaprojects data [22]: preclinical, clinical trial, phase I, II, and III clinical trials, pre-registration, registered, launched. This resulted in 1421 genes with the "target" label, and 16,683 unlabeled genes. A dataset was then built comprising the genes labeled as "target" and a random sample of 1421 of the unlabeled cases, that were labeled as "non-target". This smaller set was split into a training set (80%) and test set (20%). Four models – RF, SVM, NN and a gradient boosting machine – were trained using the training set data and then evaluated with the test set. The best performing model over the test set was NN, displaying a Receiver Operating Characteristics (ROC) Area Under the Curve (AUC) of 0.76, accuracy above 0.71, a 0.74 precision, and a 0.64 recall. This NN was later used in a prospective fashion to predict the outcome of the remaining 15,262 genes that were not included either in the training or test sets. First, genes with a probability greater than 0.9 of being labeled as "target" led to a subset of 1431 genes. To validate these predictions, the authors collected from the scientific literature genes or proteins that were flagged as potential therapeutic targets in titles and abstracts on MEDLINE [23], what led to a set of 25,603 matches, which corresponded to 4413 unique genes. It was found that 590 out of the 1431 genes labeled as "targets" were found in this literature extracted set.

Wang et al. [24] studied the prediction of drug target proteins via a binary classification task. In this study, data was collected from the DrugBank database [25], building a dataset of 517 drug target proteins (DTPS), and 5376 putative non-drug target proteins (NDTPs). To describe model inputs, sequence information statistics and amino

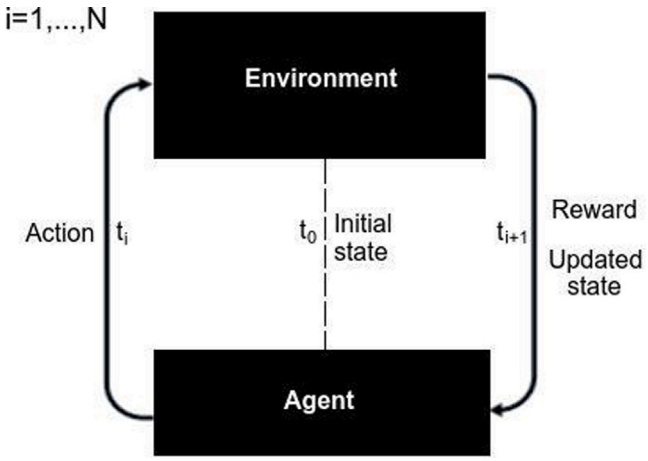

Fig. 10. Reinforcement learning scheme.

acid properties were calculated with pepstats (EMBOSS) [26] for every protein in the data set. These properties included the number of small, aromatic, aliphatic, polar, non-polar, charged, and basic amino acids, and include descriptors such as single peptide cleavages [27], transmembrane helices [28], low complexity regions [29], N-glycosylation sites [30], and O-glycosylation sites [31], totaling a set of 39 input features per protein. An AE was trained using the whole set and then the latent representations were used as inputs for an SVM classificator. To train this last model, the data was split into training (70%) and test

(30%) sets. For comparison, two additional SVM models were trained, one with the original descriptors of the protein and the other with a selected subset of features selected by a wrapper method. The SVM model trained with the original descriptors suffered from severe overfitting, achieving null recall in the test set. The SVM trained with latent representations was superior to the SVM trained with selected features. The performance of these models was assessed between two metrics, the recall value, and the F1 score, which is the harmonic mean between the recall and precision values. Specifically, the SVM trained with latent representations achieved an F1 score of 0.23 and a recall of 0.71 in the test set. Based on the results, the authors suggested that non drug target proteins misclassified as drug target proteins should be further studied experimentally. Including both the training and test set, the amount of these misclassifications was approximately 23% of the total samples. Interestingly, this percentage is in agreement with a previous work, which also utilized an SVM classificator and a smaller dataset [32].

Jeon et al. [33] performed a study which also involved the use of SVM models for the identification of drug targets. In this case, the objective was to classify proteins into drug- and non-drug–targets for breast, pancreatic and ovarian cancer. The dataset used contained 5169 proteins which were not labeled as drug–targets nor associated with cancer pathogenesis, 62 breast cancer drug targets, 69 pancreatic cancer targets, and 45 ovarian cancer targets. Drug targets were identified from DrugBank [25] and the Therapeutic Target Database [34]. Five descriptors were selected for model inputs: (i) From a large-scale screening against 29 breast, 28 pancreatic and 15 ovarian cancer cell lines [35], the average gene activity ranking profile score across all cell lines was computed for each type of cancer; (ii) from the Cancer Cell Line Encyclopedia [36], which contains information of 58 breast, 44 pancreatic and 50 ovarian cancer cell lines, two features were extracted: the average probe intensity and the average DNA copy number; both averages were calculated for all cancer cell lines in the database for each type of cancer; (iii) from the COSMIC database [37], the number of all mutations observed in DNA sequence was computed; (iv) from the Protein–Protein interaction (PPI) network [38], a topological feature was extracted, the closeness centrality, which is a reciprocal of the average distance to all other nodes from the given protein. Three SVM models were constructed using the mentioned properties in order to distinguish drug targets from non-drug targets for each cancer type. Using a 10-fold cross-validation, the models displayed good performance. Approximately, accuracy ranged from 90% to 93%, sensitivity from 62% to 89%, and specificity from 90% to 93%. These models were used to make predictions over a separate set of 15,663 cancer-associated human proteins, which was obtained by mining existing literature on experimental applications of cancer pathogenesis. Of these predicted drug targets, 266 are specific to breast cancer, 462 to pancreatic cancer, and 355 to ovarian cancer; 122 targets were found in the intersection of the three cancer types, and 69 of these overlap with a set of 116 known cancer targets extracted from a cancer drug resistance database [39]. Results were further validated by designing inhibitors for some of the predicted targets that displayed anti-proliferative effects.

An approach to finding potential drug targets that differs considerably from the works described before is the one performed by Bakkar et al. [40]. In this study, the authors proposed to identify RNA-binding proteins (RBPs) that are altered in Amyothropic Lateral Sclerosis (ALS), a known neurodegenerative disease. It is accepted that RBP dysregulation is a contributing factor in ALS pathobiology, but there are no effective treatments for this disease. Starting from the base of 11 known RBPs that mutate in ALS, the objective of this study was to propose and validate new protein candidates, from a known set of 1478 RBPs in the human genome. For this purpose, the authors used the AI platform IBM Watson to determine semantic similarities between RPBs published in the literature. The main hypothesis behind this approach was that proteins discussed in similar textual contexts may have similar functions. The algorithm considers two proteins to be similar if the words and phrases used in documents that mention them are similar. Once the semantic similarity is calculated between every pair of RBPs, a network is constructed with this information, and a score is assigned to each RBP by using a network analysis algorithm, namely graph diffusion. This algorithm relies on a given a set of positive RBPs and a given set of candidate RBPs and rank orders the candidate RBPs by similarity to the known positive set, assigning a score from 0 to 1 corresponding to how closely related each of them is to the positive set. Initially, the methodology was assessed retrospectively by using different subsets of the mentioned 11 RBPs as the positive set and retrieving the rank of the RPBs that were left out from this set, which were incorporated with the candidate samples. RBPs were ranked in terms of the score assigned by the graph diffusion method, which ultimately gives a probability of an RBP of being altered in ALS. In all studied cases, the rank of the left-out RBPs ranged between the top score and the top 11% of the ranked list. Once the methodology was validated for identifying RBPs involved in ALS, the set of 11 RBPs known to mutate in ALS were used as a positive set. The top ten proteins predicted by the algorithm included three that were previously associated with ALS. From the remaining seven proteins unlinked to ALS, alterations in five RBPs were validated, i.e., RBPs exhibited statistically significant differences between ALS and controls by at least two of the following methods: protein subcellular distribution using immunohistochemistry, measures of protein levels by immunoblot, RNA levels by total tissue extracts and laser-capture microdissection, and RNA analysis of motor neurons generated from patient-derived iPS cells.

Madhukar et al. [41] implemented a Bayesian ML approach to predict drug binding targets. This approach integrates multiple data types: growth inhibition data from the National Cancer Institute's Development Therapeutics Program [42], post-treatment gene expression data from the Broad Connectivity Map project [43,44], side effects downloaded from the Side Effect Resource (SIDER) database [45], bioassay results and chemical structures extracted from PubChem [46,47], and known drug targets extracted from the DrugBank database [48,49]. The constructed database contained, approximately, 2000 different drugs with 1670 different known targets and over 100,000 unique compounds with no known targets, which are described as "orphan compounds". For every drug pair, the methodology outputs a measure of the probability that the two drugs share a given target based on the available data types of each drug. Two main implementations arise from this methodology: finding modulators for a specific protein, and identifying targets for a given small molecule. For the first task, the authors predicted targets from a set of orphan compounds and prioritized a list of small molecules predicted to target microtubules, obtaining a set of 24 compounds; in an experimental validation with human breast cancer MDA-MB-231 cells, 14 of the 24 orphan small molecules exhibited significant effects on microtubules.

For the second task, the authors used their method to predict a related target to ONC201, a small orphan molecule currently in multiple phase II clinical trials for advanced cancers. The most likely targets of ONC201 were the dopamine receptor D2 (Fig. 11) and the alpha adrenergic receptor. Experimental validation of these predictions showed that ONC201 selectively antagonizes the D2-like subfamily of dopamine receptors.

3.2. Hit and lead identification

3.2.1. Virtual screening

Virtual Screening (VS) refers to a series of computational techniques whose objective is to prioritize from a large chemical library, a list of compounds for biological evaluation. This approach emerged as a complementary approach to HTS. Two main types of VS can be identified: in LBVS, the topological, and/or physico-chemical and/or pharmacophoric properties of known ligands are taken into account to search for molecules that might bind to the target (cf. excellent reviews on this subject [6,7,50,51]). In SBVS, the 3D structure of the

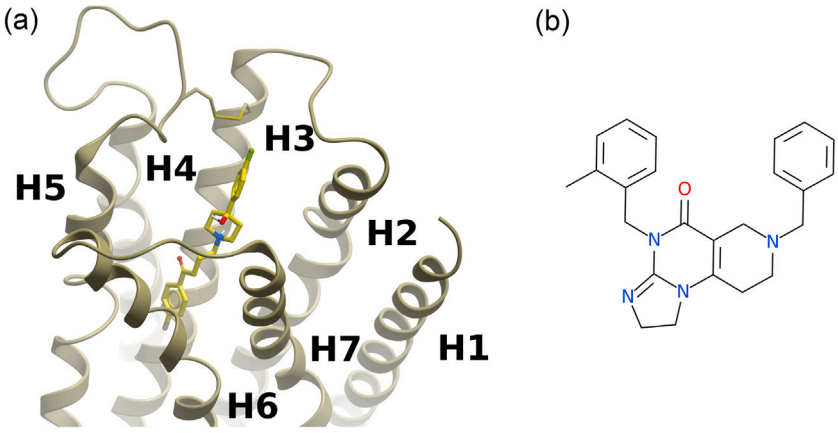

Fig. 11. (a) Dopamine receptor D2 complexed with haloperidol within the binding site defined by helices 3, 5, 6 and 7. (b) Structure of the orphan compound ONC201, for which the Dopamine receptor D2 was predicted as a target. Figure prepared with ICM (Molsoft LLC, San Diego, CA).

target of interest is used to screen large chemical libraries using either high-throughput docking [9,52–55] or receptor-based pharmacophore screening [56]. Needless to say, VS has proven a suitable method for finding lead compounds that have the potential to become effective drugs, but there are two main aspects in which the application of AI could enhance VS: efficiency and accuracy. Moreover, ML and DL in SBVS could be an effective way to account for non-linearity into free binding energy calculations, which ideally could lead to better results.

The implementation of machine learning methods for VS is not novel. In 2006, Plewczynski et al. [57] compared several classification methods in terms of their capability to predict active compounds for different biological targets, namely, HIV-reverse transcriptase, COX2, dihydrofolate reductase, estrogen receptor (ER), and thrombin. The set of compounds used in this study was extracted from the 2004 edition of the MDDR (MDL Drug Data Report) [58]. For each target, two sets of ligands were used. A larger set contained all ligands for each target according to the annotation in the MDDR; on average, each target had associated 11,400 molecules, of which less than 5% were active. A smaller set only included ligands which had gone beyond Phase I according to the annotations in the MDDR; on average, each receptor had associated 10,800 molecules, with the number of actives being less than 1%. Model inputs were generated using the regular atom pair (AP) descriptors [59]. Outcomes were binary, indicating whether the compound was active or inactive on the corresponding target. Seven models were implemented: SVM, NN, naive Bayesian classification, k-nearest-neighbors (kNN), random forest (RF), decision trees (DT), and trend vectors (TV). Each methodology was evaluated in terms of the enrichment factor, recall and precision. The authors performed an extensive comparison between methods and targets using different splits of training and test sets. Their analysis showed that different algorithms have different strengths and concluded that the methodology of choice should be based on the priorities of the problem itself. For example, methods like TV and RF provided particularly high enrichments, and others like SVM exhibit high recall values. Finally, the results from the seven methods were combined within different consensus approaches, where for each target, a compound was considered to be predicted as active if it was predicted active by at least one, two, …, seven of the methods. Based on this analysis, the authors showed that, in the second set, where the number of active molecules is lower, a consensus approach could lead to a substantial improvement in comparison to single methods, suggesting that this approach is more relevant for unbalanced data sets.

Kinnings et al. [60] performed a study in which the principal objective was to improve docking scoring functions with ML. As a frame of reference, the authors used the docking program eHiTS [61]. Two datasets were constructed using the Mycobacterium tuberculosis enoyl acyl carrier protein reductase, InhA, as a protein target. The first set was built using 80 InhA inhibitors with IC_{50} data extracted from the Binding database [62], and the second one included these 80 inhibitors and 36 decoy molecules per inhibitor, extracted from the Directory of Useful Decoys (DUD) [63]. Datasets containing actives and decoys extracted from DUD were constructed for 12 additional targets; angiotensin converting enzyme, acetylcholinesterase, cyclin-dependent kinase 2, cyclooxygenase-2, epidermal growth factor receptor, factor Xa, HIV reverse transcriptase, P38 mitogen activated protein kinase, phosphodiesterase 5, platelet-derived growth factor receptor kinase, tyrosine kinase SRC and vascular endothelial growth factor receptor. Model inputs consisted in a set of 11 selected energy terms calculated with eHiTS, which included energy terms like hydrogen bonding, $\pi-\pi$ stacking, van der Waals interaction, electrostatic interaction, hydrophobic effect, and entropy. Using the first dataset, the authors constructed a regression SVM model to predict IC_{50} values, which was evaluated with the Spearman's rank correlation coefficient ρ between the predicted IC_{50} values and the experimental data using a 5-cross validation. The SVM model achieved a correlation coefficient of $\rho = 0.67$, outperforming the eHiTS scoring function, which in contrast had a correlation coefficient of $\rho = 0.12$. Using the second set, the authors developed an active-inactive compound classificator with a multiplanar-SVM model. Based on the ROC curve, the SVM model exhibited higher enrichment factors consistently at different percentages of the database than eHiTS. For example, the SVM achieved enrichment factors of 6.0 and 8.8 at 2% and 5% levels, respectively. This methodology was further assessed by implementing multiplanar SVM classification models for each of the mentioned 12 targets. ROC curves showed that the developed model outperformed eHiTS in every single case.

Taking advantage of the emergent techniques associated with NNs to control over-fitting or to perform hyperparameter tuning, Dahl et al. [64] showed that NNs performed better than other baseline methods in quantitative structure–activity relationships (QSAR) predictions. Specifically, the aim of this work was to classify a series of compounds as active or inactive for 19 cellular and biochemical assays. The dataset was built based on publicly available assay results extracted from PubChem [65], and included a set of multiple assays on different families of Cytochrome P450 enzymes, and a set associated to the inhibition of Sentrin-specific proteases. The number of datapoints varied over each assay, ranging from ~2000 to ~14,000 data points. The percentage of active molecules in each assay varied from 20% to 80%. As inputs, the compounds were described with a set of 3764 molecular descriptors generated with Dragon [66]. For each assay, three quarters of the available data were selected at random as the training set, and the other quarter as the test set. Single-task NNs and a Multi-task NN were trained with all the assays, and at the same time were compared with a selection of baseline methods, namely, RF, gradient boosted decision tree ensembles and Logistic Regression

(LR), where the classification's performance was measured with the ROC AUC. In 14 of the 19 assays, NNs achieved an AUC exceeding the best baseline result by a statistically significant margin. In 12 of these 14 cases, the best AUC was achieved by the Multi-task NN. The corresponding NN AUCs in these 14 cases ranged from 0.65 to 0.94.

Unterthiner et al. [67] studied the performance of DL in VS testing and assessed whether these methods can scale to in-house datasets of pharmaceutical companies. Specifically, the objective of this study consisted in predicting, simultaneously, if a given compound is active on several targets, which was performed in terms of a classification task. A large dataset was extracted from ChEMBL [68], which contained more than 1200 targets and 1.3 M compounds. To feed the algorithms, the compounds were represented with Extended Connectivity Fingerprints, which yielded a total of 13,558,545 sparse features. Outputs were binary, indicating if a compound is active or inactive in each of the corresponding targets. DNNs were compared to seven other methods, namely, SVM, LR, kNN, Pipeline Pilot Bayesian Classifier, Parzen Rosenblatt kernel density estimator, Binary Kernel Discrimination and Similarity Ensemble Approach. The performance of each model was measured with the mean ROC AUC across all targets. The DNN significantly outperformed the other algorithms, including two commercial methods. Specifically, the NN achieved an AUC $\geq$ 0.8 on 813 out of the 1230 targets (66%), and on 12 targets achieved a perfect score (AUC=1.0), with the median AUC laying at 0.86. On the other hand, the median AUC for commercial solutions was below 0.8.

A recent study performed by Lenselink et al. [69] also assessed the performance of DL models to classify compounds as active or inactive for a set of several proteins. The dataset was extracted from the ChEMBL database, and was composed of 1227 targets, which, on average, had 257 tested compounds associated. The distribution between active and inactive compounds was $\sim$ 50% each. To describe inputs, the authors calculated, for every compound, several physicochemical descriptors – partition coefficient (AlogP), molecular weight (MW), hydrogen bond acceptors (HBA) and donors (HBD), fractional polar surface area (fractional PSA), and rotatable bonds (RB) – as well as the RDKit Morgan fingerprints. These fingerprints were calculated as bit vectors (256 bits), rather than in a fingerprint count format. To generate Morgan fingerprints, also known as circular fingerprints, a fingerprint radius must be specified, which in this work, was set to 3 bonds. Another approach was also explored, in which information of the target was incorporated into the input combined with compound data, so that activity predictions were made over protein/compound pairs. To this aim, 169 protein descriptors were incorporated; protein sequences were divided into 20 equal parts, and for each part, eight properties were calculated for each aminoacid and then averaged, resulting in eight descriptors per sequence partition. These properties included the number of stereo atoms, logP, charge, HBA and HBD, rigidity, aromatic bonds and MW. The overall average of the mentioned amino acids properties was also used, as well as the sequence length. Models were trained on 70% of a random split set and then validated on the remaining 30%. The authors also performed a temporal validation, which consisted in grouping the data by publication year rather than by random partitioning. As an evaluating metric, the authors used the Matthews correlation coefficient (MCC) and the Boltzmann-Enhanced Discrimination of the Receiver Operating Characteristic (BEDROC). Several algorithms were used as a baseline for comparison; RF, Naive Bayes, LR, and SVM. The two best methods overall were a multiclass DNN, and a DNN that incorporated protein descriptors. In the random split, the mean MCC and BEDROC over all the methodologies were 0.49 and 0.85, respectively. The MCC and BEDROC of the multitask NN were 0.57 and 0.92, and in the case of incorporating protein descriptors, the scores of the DNN were 0.55 and 0.93, respectively. The performance of all the methods had a significant drop when evaluated in the temporal split. Based on this, DNNs were found to be the best algorithm.

A remarkable example of transference between a successful application of AI in a certain field and drug discovery is the case of AtomNet, implemented by Wallach et al. [70]. Computer vision has been enormously benefited from the use of Convolutional Neural Networks (CNN), and AtomNet was the first application of deep CNN for SBVS. The methodology was designed to classify compounds as actives or inactives. Four sets were used to evaluate the model's performance. The first set was extracted from the Directory of Useful Decoys Enhanced (DUDE) [71], and consisted of 102 targets, 22,886 actives – with an average of 224 actives per target – and 50 property matched decoys (PMDs) per active. The second set was built from the first set, but selecting 30 targets at random as a test set, with the remaining 72 targets forming the training set. The third set was designed by the authors, ensuring that there was no overlap between the training and test molecules. In this case, the dataset was extracted from ChEMBL, and consisted of 290 targets, 78,904 actives and 2,367,120 PMDs extracted from the ZINC database. The data was splitted into training, validation and test sets via clustering techniques. The last dataset was built in a similar fashion to the previous case, but the PMDs were replaced with experimentally verified inactive molecules. This fourth set consisted of 290 targets, 78,904 active compounds and 363,187 inactive compounds. This scenario represents a more challenging task because, unlike the previous cases, it includes structurally similar molecules that have different activities. As input, the algorithm receives 1D vectors, the result of unfolding 3D grids within the target's binding site. This representation includes information about some basic structural features, ranging from simple descriptors such as the enumeration of atom types to more complex protein–ligand descriptors such as structural protein–ligand interaction fingerprints (SPLIF) [72], Structural interaction fingerprint (SIFt) [73], or atom-pairs-based interaction fingerprint (APIF) [74]. The docking program Smina was used as a baseline method for comparison with AtomNet. Performance was measured in terms of AUC and logAUC, which gives more importance to the early enrichment. On every evaluation data set, AtomNet outperformed Smina. For example, in the DUDE benchmark, restricted to the held-out 30 target subset, AtomNet achieved a mean AUC of 0.86 compared to 0.70 achieved by Smina. Specifically, AtomNet achieved an AUC of 0.90 on 14 targets (46.7%), and 0.80 on 22 targets (73.3%). On the other hand, Smina achieved an AUC of 0.90 for 1 target (3.3%) and 0.80 for 5 targets (16.7%). In the most challenging set, although both Atomnet and Smina performed worse than on the previous benchmarks, AtomNet still significantly outperforms Smina with respect to overall and early enrichment performances. AtomNet achieved mean AUC and mean logAUC of 0.75 and 0.15, respectively, compared to 0.61 and 0.05 achieved by Smina. Also, the authors presented a comparison of their methodology with three other commercial docking algorithms reported in the literature; Gabel et al. evaluated Surflex-Dock [75,76] on a set of 10 targets from DUDE, Coleman et al. evaluated DOCK3.7 [77,78] on all targets from DUDE, and Allen et al. evaluated Dock6.7 [79] on 5 targets from DUDE. Based on the corresponding targets from the DUDE benchmark, AtomNet outperformed the other methods in terms of AUC. The mean AUCs reported were 0.76 using Surflex-Dock, 0.70 using DOCK3.7 and 0.72 using Dock6.7. Atomnet's mean AUCs were, respectively, 0.93, 0.90 and 0.85. When dealing with images, filters from CNN learn to automatically detect relevant features at different levels of abstraction, going from lines and edges to, for example, noses and eyes. As the authors explain, in this case, due to data representation and model architecture, it is not possible to visualize the filter detection directly. However, they managed to visualize what the CNN is learning in a more indirect way, applying the filters to input data. Interestingly, the method detected relevant chemical functions with their autonomously trained convolutional filters.

Later studies utilizing CNNs have also been published [80–82]. For example, Pereira et al. [82] developed a DL methodology which included a convolutional layer to classify compounds as actives or non actives with the objective of improving docking-based VS. The data used in this study was extracted from the DUD, and was composed

of 40 receptors, 2950 annotated ligands and 36 decoys per ligand. Protein-molecule complexes were generated for each target, with the corresponding set of ligands and decoys, with two docking programs, Autodock Vina 1.1.2 [83] and Dock 6.6 [84]. The proposed model takes as inputs structural data of the protein-molecule complex, involving for each compound atom, neighbor atom types, atomic partial charges, and associated residues. The model outputs a score for differentiating ligands from decoys. The authors implemented a leave one out cross-validation, shown in Fig. 12, consisting in leaving only one receptor in the test set while the model was trained with the rest of the receptors, in an iterative manner so that every target was present in the test set once. In every cross validation iteration, all receptors similar to the one used in the test set were removed from the training set. Similar receptors were defined as those sharing the same biological class or those with reported positive cross enrichment [63,85].

The methodology trained with the protein-molecule complex generated with Auto Dock Vina exhibited robust results. In particular, it achieved an average AUC, across all targets, of 0.81, surpassing several docking results on the DUD database reported in the literature [85–88], including commercial docking softwares ICM [89] and Glide SP [90].

In a recent study [91], Adeshina et al. built a binary classifier to distinguish between ligands and non-ligands with the objective of performing SBVS. Major importance was given to the training set development, which included a small number of very challenging decoy complexes per ligand. Specifically, the dataset consisted of 1383 ligand–target complexes with three decoys per ligand; 39 of these decoy compounds included chemical features that could not be processed by the programs the authors used to extract structural features so they were removed, leading to a total of 4110 decoy complexes. The authors used top scoring functions reported in the literature to this constructed data set to distinguish between ligands and decoys, namely, nnscore [92], RF-Score v1 [93], RF-Score v2 [94], RF-Score v3 [95], RF-Score-VS [96], PLEClinear, PLECnn and PLECrf [97]. While these scoring functions are of continuous nature, their values were converted through a threshold into an active/inactive binary output. The best-performing scoring function achieved an MCC of only 0.39, indicating that the task represented a challenging classification problem. Then, the authors built a model based on a decision tree algorithm, to distinguish between actives and decoys in their constructed set. Model inputs consisted of 68 features extracted from different sources: energy terms and structural quantifiers from the Rosetta energy function [98], multiple distance-dependent atom counts from RF-Score [92], analysis of intermolecular contacts from BINANA [99], ligand-specific molecular descriptors from ChemAxon's cxcalc [100], a program used to perform chemical calculations, and a term intended to capture ligand conformational entropy lost upon binding from OpenEye's SZYBKI tool [101]. Using the binary outputs described above, they achieved a MCC of 0.74. Also, this methodology and the mentioned scoring functions were compared using two external sets. One was extracted from the DEKOIS project [102,103], and contained 23 proteins, and between 30 to 40 actives and 800 to 1200 decoys per protein. The other one was extracted from a previous study concerning the inhibition of PPI [104], and was composed of 10 protein targets. Each target had associated one active compound and approximately 2000 decoys. In both cases, only the RF-Score-VS was not outperformed by the authors' method. However, it was pointed out that in the first mentioned set, about half of the 23 targets in that benchmark were included in the data used to train RF-Score-VS, while the authors methodology was trained with none of them. Finally, their methodology was applied in a prospective fashion against acetylcholinesterase. Testing in biochemical assays the top-scoring compounds, the authors found that the majority of the selected compounds showed detectable enzyme inhibition and high potency. This highlights the fact that the performance exhibited in the initial benchmark could be extended satisfactorily to other cases, and established a promising method for real-world applications.

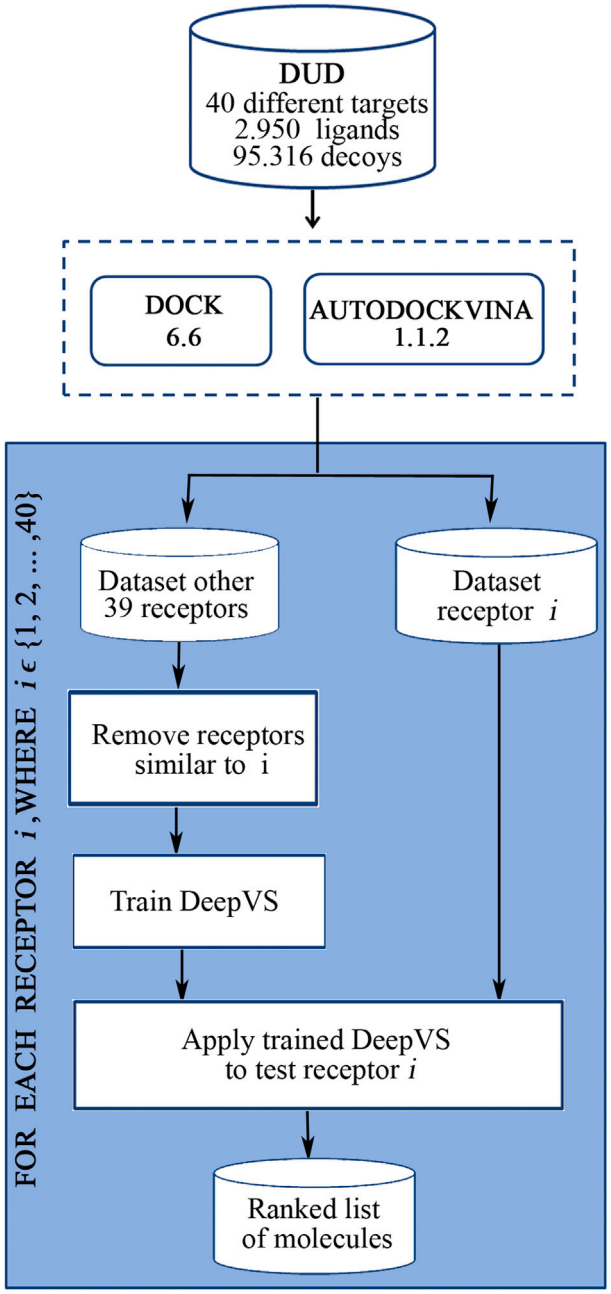

Fig. 12. Training methodology of DeepVS using a leave one out cross validation. Reprinted with permission from Ref. [82]. Copyright (2016) American Chemical Society.

Gentile et al. [105] proposed a DL-based pipeline to reduce an ultralarge docking database of billions of entries to a manageable few million-molecules subset. The proposed method was implemented in a docking campaign comprising the 1.36 billion molecules from ZINC15 database and 12 protein targets: androgen receptor, estrogen receptor-alpha, peroxisome proliferator-activated γ receptor (PPARγ), calcium/calmodulin-dependent protein kinase kinase 2 (CDK2), cyclin-dependent kinase 6 (CDK6), vascular endothelial growth factor receptor 2 (VEFR2), adenosine A_{2A} receptor, thromboxane A2 receptor, angiotensin II receptor type 1, Nav1.7 sodium channel, gloeobacter ligand-gated ion channel, and gamma-aminobutyric acid receptor type A. Protein crystal structures were extracted from the Protein Data Bank [106]; For each target, 5 million compounds selected at random were docked using the FRED docking program [107]. These 5 million

compounds were divided into training set (3 million compounds), validation set (1 million compounds) and test set (1 million compounds). Then, using a docking score cutoff, a DNN model was trained to predict the probability of a compound being a virtual hit or a non-hit. For model inputs, compounds were described with Morgan fingerprints with a size of 1024 bits and a radius of 2, generated with the RDKit package. Then, the following process was carried out in an iterative manner. The trained DL model was used to predict docking outcomes of the unprocessed entries of the database. One million molecules predicted to be virtual hits were randomly sampled and docked with FRED. The docking outcomes were then used for training augmentation. This last step was repeated 10 times, so, for each target, 13 million molecular structures were docked, representing less than 1% of the ZINC15 database. For each target, the validation set and the test set remained unchanged. The first one was used to determine the docking score cutoff while the test set was used to assess the model's performance. The number of remaining molecules predicted as virtual hits after the last iteration, ranged, for the 12 targets, between 1% and 12% of the ZINC15 database. In the best scenario, corresponding to the PPARγ protein, the database was reduced to 1% of its size. Thus, with the proposed pipeline, screening of ZINC15 against this target requires docking of 50 times fewer molecules than conventional VS. Enrichment values for the top 10, 100, and 1000 predicted virtual hits identified in the test sets after the final iteration ranged from 240 to 6000. Such enrichments decreased consistently in all cases when evaluating larger portions of top ranked structures, suggesting that true hits are highly concentrated at the top of the ranked molecules, and molecules at the bottom of the rank are mostly false positives.

A study performed by Jimenez-Luna et al. [108] implemented a DL model to rank congeneric series of ligands by predicting the relative binding affinities for close analogues. For the training set, 495 congeneric series with their corresponding IC_{50} values were extracted from the BindingDB protein–ligand set [62]. Model inputs consisted in pairs of ligands which belong to a same congeneric series, and model outputs consisted in the difference in affinity between the ligands. For the inputs, 3D grids were utilized within the corresponding binding sites and 18 properties were used to describe each grid element. These included eight properties associated with pharmacophoric-like descriptors of the target, and ten properties associated with ligand descriptors based on atom types. The implemented model was a two-legged 3D convolutional neural network, where each leg corresponded to each of the ligands input to the model. The performance was measured in terms of the Root Mean Square Error (RMSE), and either the Pearson's or the Spearman's correlation coefficient between the experimental and predicted affinity differences. To explore how the model would work in a real lead optimization scenario, the model was initially trained with only one binding-energy difference per congeneric series, achieving an average correlation coefficient above 0.4 and RMSE below 1.25 (pIC_{50} units). This performance increased as more ligands were added to each congeneric series, reaching a plateau after the addition of five extra ligands. With 4 additional ligands in each congeneric series, the model achieved a correlation coefficient above 0.62 and a RMSE value below 1.05. After training the model with all the BindingDB data, it was tested with two publicly available sets, one extracted from the literature [109], and the free-available BRD4 bromodomain inhibitor dataset [110]. The model was fine-tuned with some of these new data points and tested in the remaining samples. In the majority of the cases, the authors' methodology surpassed the performance of a baseline free binding energy model [109] that was used for comparison. The model was also tested using several in-house pharma databases composed of series of congeneric ligands for different targets (number of series in parenthesis): (i) from Janssen R&D phosphodiesterase 2 (3), phosphodiesterase 3 (3) and phosphodiesterase 10 (3), proto-oncogene tyrosine kinase (1), and beta-secretase 1 (1); (ii) from Pfizer, a kinase (3), an enzyme (1), a phosphodiesterase (1), and an activator of transcription (1); from Biogen, a tyrosine-protein kinase (1) and a receptor-associated kinase (1). As occurred in the previous scenario, in the majority of these cases, the model outperformed several baseline models trained with low amount of new data. Finally, using the data of five congeneric series, an interesting retrospective analysis was performed. For each series, the order in which ligands were experimentally assayed was compared to the order in which the DL method suggested to assay it. In four of the five series, the analogue with the highest associated affinity would have been assayed earlier using the DL suggestions.

Recently, with the promising results that AI displayed in VS, more general studies appeared in the literature, discussing, among other topics, how to increase the performance of ML scoring functions with the inclusion of ligand-based features [111], or which ML scoring function is the most suitable for prospective use on a given target [112].

In spite of these implementations, it is clear that a ML scoring function may not always be the best choice. In the study performed by Singh et al. [113], the authors explored different alternatives to improve the performance of SBVS on protein–protein interfaces, and they found that a knowledge-based scoring function, DLIGAND2 [114], outperformed the results from RF-Score-VS v2.

3.2.2. Drug repurposing

An appealing alternative for drug discovery is to find a new use for an old drug, an approach known as drug repurposing which represents a notorious advantage in terms of time and costs [115]. In fact, this strategy could be extended to promising drug candidates that have passed clinical phase I of clinical. When no treatments are available, a repurposing strategy would be ideal. There are some examples of applications of ML approaches in the field that are linked to a repurposing practice, in some cases in a more direct way than others.

A direct approach is found in the work performed by Aliper et al. [116], in which a set of drugs were classified into several therapeutic categories. The dataset used in this study consisted of perturbation samples of 678 drugs across A549, MCF-7 and PC-3 cell lines, extracted from the LINCS Project [117]. Samples included different drug concentrations, time of perturbation and cell line parameters. These samples were linked to 12 therapeutic categories derived from the medical subject headings (MeSH) [118]: Anti-inflammatory, hematologic, cardiovascular, central nervous system, urological, respiratory system, reproductive control, dermatological, gastrointestinal, anti-infective, antineoplastic, and lipid regulation. Three alternatives were explored to describe the samples for model inputs. Initially, samples were described with gene expression data, consisting of 12,797 genes in total, but two approaches were studied to select the biologically relevant features: (i) the first approach was implemented with OncoFinder [119], which performs a quantitative estimation of signaling pathway activation strength, indicating how significantly a pathway is up- or down-regulated; in total, 271 pathway activation profiles were calculated for each sample; (ii) The second approach consisted in using a normalized gene expression of 977 so-called "landmark genes", defined in the LINCS Project. Outputs were the mentioned therapeutic categories, where there was only one category per drug. Two models were implemented, SVM and DNN, and compared in terms of the F1 score with a 10-cross validation scheme. An initial mean F1 score of 0.24 was obtained with a DNN trained with all the gene expression data. In regard to the feature selection approaches, the best performance was obtained using the signaling pathway description. After removing samples in which the pathway activation score was zero over the mentioned signaling pathways, the final set consisted of 308, 454 and 433 drugs for the A549, MCF7, and PC3 cell lines, respectively, corresponding to 9352 samples. Using the 12 categories, the mean F1 score of the DNN and the SVM classifier were 0.55 and 0.37, respectively. In the case of using the most abundant classes – antineoplasic, cardiovascular and nervous system – the mean F1 score of the DNN was 0.70 in contrast to 0.53 of the SVM. Finally, the authors constructed a confusion matrix which illustrates in how many cases a category is misclassified into another class, and suggested that this information could be useful to

perform drug repurposing. For example, drugs corresponding to the cardiovascular category were often misclassified as central nervous system and antineoplasic classes. One of the drugs used in the dataset, Otenzepad, a known muscarinic receptor antagonist used for cardiac arrhythmia, was misclassified as central nervous system, and remarkably as the authors pointed out, this compound has a clear role in brain function.

A more indirect approach to drug repurposing was performed by Kinnings et al. [60] by using the SVM model discussed in Section 3.2.1. To identify potential lead compounds from existing drugs that can inhibit InhA, 962 protein structures co-crystallized with 274 FDA approved drugs were extracted from the Protein Data Bank (PDB) [120] and a binding site similarity comparison with InhA was performed with SMAP [121–123]. Based on the SMAP values, strong connections were found between InhA and the phosphodiesterase type 5 (PDE5) inhibitors. In consequence, 303 PDE inhibitors, extracted from the Binding database were docked into the InhA binding site and the developed SVM models were used to rank these inhibitors. There was a considerable overlap between the best scored compounds and those with the best IC_{50} values predicted by the regression model. Particularly, six PDE4 inhibitors were in the top ten ranked compounds of both methods. Another significant connection to InhA was Estradiol, which binds the ERα/β, and Raloxifene, a selective ER modulator. Consequently, molecules experimentally confirmed to bind ERα, ERβ, the estrogen related receptor (ERR) or ERRγ were extracted from the Binding database, totaling 223 compounds. Again, a strong correlation was observed between the regression SVM model and the classification SVM model, and the most highly ranked compounds were analogs of 4-hydroxytamoxifen.

A remarkable work performed by Stokes et al. [124] led to the discovery of a new antibiotic using a DL classifier trained to predict antibacterial activity against *E. coli*. This case is a good example of how AI can surpass obstacles in problems such as the discovery of new compounds against emergent antibiotic-resistance bacteria strains. Four datasets were used in this study. The first set, the only one used for the training process, contained 2335 molecules, which included FDA-approved drugs and natural products screened for growth inhibition against *E. coli*, where 120 molecules were identified as active. The second set contained 6111 molecules extracted from the Drug Repurposing hub [125]. The third dataset was composed of 9997 molecules extracted from the WuXi anti-tuberculosis library, housed at the Broad Institute [126]. Finally, the fourth set contained ~107 million antibiotic-like compounds extracted from the ZINC15 database [127]. The first set was used to train a binary classification model (active-inactive) based on a direct message passing NN, which has as inputs the graph structures of the molecules. Nodes and edges, representing atoms and bonds, were initialized with different features. Atomic features included atomic number, number of bonds, formal charge, chirality, number of bonded hydrogens, hybridization, aromaticity and atomic mass. Bond features include bond type, conjugation, ring membership and stereochemistry. Also, the authors mentioned that the learned molecular representation of each compound was complemented with a set of 200 features calculated with RDKit. The rest of the mentioned sets were used to search for antibacterials based on the predictions of the classifier. With respect to the second set, empirical testing of the top 99 molecules predicted by the model revealed that 51 of them showed activity against *E. coli*. These 51 compounds were prioritized based on several criteria: being at pre-clinical stage or in phase 1, 2 or 3; displaying low toxicity based on the predictions of a DNN trained with the ClinTox database [128,129], and showing low structural similarity to the molecules present in the training set. The c-Jun N-terminal kinase inhibitor SU33217 [130,131], which the authors rename Halicin, satisfied the specified criteria. As was explored in depth in the study, this compound exhibited several desired attributes, such as showing high *in vivo* efficacy, displaying a broad-spectrum bactericidal activity, and being structurally distant from known antibiotics. Prior to predicting activities in the third dataset, the model was re-trained with the empirical information gathered in the aforementioned assays. The highest predicted probability of being active in the third set was 0.37, a low probability in comparison to the one obtained in the second set (~0.98). From the third set, 300 structures were assayed for growth inhibition against *E. coli*, which corresponded to 200 compounds with higher score and 100 structures with lower score, but none of them displayed antibacterial activity.

The model was re-trained with this new empirical data. Then, activity was predicted for the fourth set; compounds with a score greater than 0.8 and a Tanimoto similarity to the nearest antibiotic neighbor lower than 0.4 were prioritized. It was found that 23 compounds met this criteria and were assayed for growth inhibition against *E. coli*, *S. aureus*, *Klebsiella pneumoniae*, *A. baumannii* and *P. aeruginosa*. Interestingly, 8 of the 23 compounds showed activity against one of the mentioned targets. Also, two of these compounds exhibited potent broad-spectrum activity (Fig. 13).

3.2.3. Generative models and de novo design

A particular area that has driven a lot of attention in recent years is generative molecular design based on DL. AI in *de novo* design plays two fundamental roles: (i) it provides an algorithm to effectively generate molecular structures, and (ii) it evaluates the generated compounds via property prediction, prioritizing compounds with suited pharmacological and physicochemical properties. In what follows we present some examples of the most common techniques of generative modeling that were successfully adapted to the drug discovery pipeline, such as RNN, VAE and GAN, among others.

Gomez et al. [132] followed a VAE approach used in the context of NLP [133] and extrapolated the model to use it with Simplified Molecular Input Line Entry Specifications (SMILES), showing that it is feasible to use variational autoencoders to generate molecular structures. Two datasets were used in this study. The first set, consisted of 250,000 drug-like commercially available molecules extracted at random from the ZINC database [134], and the second one was composed of 108,000 molecules with fewer than nine heavy atoms extracted from the QM9 dataset [135]. Two VAEs were studied, each one trained with a different dataset. Inputs, and outputs as well, were molecular SMILES, and a genetic algorithm (GA) was used as a comparison method to generate structures. In the first place, the generative capacity of the VAE methodology was evaluated. Bearing in mind that the decoding process is of a non-deterministic nature, the authors showed that for most latent points a particular molecule was decoded most frequently, as well as some other molecules with slight variations with lower frequencies. They also estimated that 30 molecules can be generated in the vicinity of a given reference molecule. So, for example, the VAE trained with the ZINC dataset, could generate ~7.5 million molecules. Secondly, the VAE models were compared with a baseline GA. Particularly, the study showed that molecules generated by the VAE exhibit chemical properties – octanol-water logP, quantitative estimation of drug-likeness (QED) and synthetic accessibility score (SAS) – that are more similar to the training set than the ones generated by the GA. Also, the molecules generated by the GA tended to be more complex and with lower drug-likeness than the ones generated from the VAE. Finally, the authors evaluated the model's capability to generate structures with specific properties. For this purpose, they extended the model's architecture by adding a property predictor model – a fully connected NN – that uses as inputs encoded structures, i.e., points from the latent space, and as outputs specified property values such as logP, QED and SAS. Interestingly, when the VAEs were trained jointly with the property prediction task, a principal component analysis (PCA) of the latent space revealed that molecules were separated in distinct regions according to their respective property values. This is a key feature considering that guiding the generation of molecular structures towards a particular trait is translated to an optimization problem in the latent

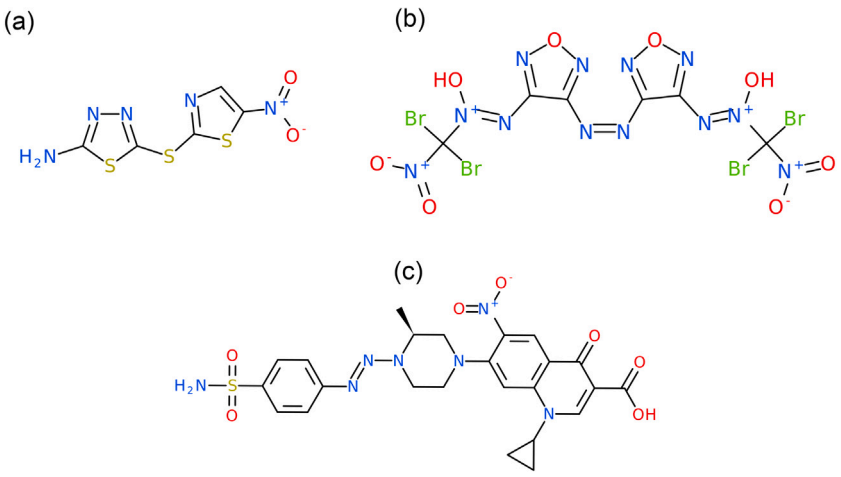

Fig. 13. Antibiotic discovery using deep learning. (a) Halicin, found in the drug repurposing hub. (b) ZINC000225434673 and (c) ZINC000100032716, two potent broad-spectrum antibiotics found in the ZINC15 database. Figure prepared with ICM (Molsoft LLC, San Diego, CA).

space. Indeed, the authors defined a score function based on QED and SAS values, with the aim of finding the most drug-like molecule that is also easy to synthesize, and using a Gaussian Process (GP) to navigate the latent space, retrieved molecules with higher score than compounds sampled at random or structures generated by the baseline genetic algorithm.

Segler et al. [136] used a RNN model to generate active compounds against three specific targets, namely, the 5-HT_{2A} receptor, *Plasmodium Falciparum* and *Staphylococcus aureus*. The data used in this study consisted of a set of 1.4 million molecules, with measured biological activity derived from the ChEMBL database. Also, three additional sets containing active tested compounds against the mentioned targets were also extracted from ChEMBL, including 732 molecules for the 5-HT_{2A} receptor, 2479 compounds for *Plasmodium Falciparum* and 7051 structures for *Staphylococcus aureus*; both inputs and outputs of the generative model were SMILES. To assess if the generated compounds were active against a given target, the authors implemented a gradient boosting trees classifier on each target; each one was trained with extended connectivity fingerprints with a diameter of 4 (ECFP4) [137,138]. The outputs of these classifiers were of binary type. After training the RNN model with the 1.4 million compounds, the models' performance at generating SMILES was assessed. From ~1 million generated SMILES, 97.7% corresponded to valid structures, out of which ~865 thousand compounds were not present in the training set. After removing duplicates from this generated set, ~850,000 molecular structures were obtained. Based on AstraZeneca filters [139], 75% of these generated molecules were cataloged as suitable for a high-throughput screening (HTS) campaign. This percentage also coincides with the one calculated for the training set. Also, the authors compared generated structures with the ones used in the training set in base of several chemical properties like MW, HBD, HBA, RB, logP, and total PSA, showing that both sets of properties overlapped almost completely. In summary, the model was able to generate novel structures that resemble the training set characteristics. In order to modify the initial distribution of the generated molecules towards active compounds against a given target, the authors used a transfer learning methodology, which consisted in retraining the generative model with the small data set of the active compounds for the corresponding target. For the 5-HT_{2A} receptor, sampling structures in each fine-tuning step showed that the number of generated molecules predicted to be active increased with the re-training process. A similarity analysis between the generated structures and their nearest active ligand revealed that the generated molecules were close analogues to tested actives. Also, some of these compounds, which were predicted to be active, presented new scaffolds. In the case of *P. Falciparum*, the initial RNN model was re-trained, with the corresponding active compounds, and then used to generate ~128,000 unique compounds. From this set of generated molecules, 28% were present in the hold-out test set. In the last case, corresponding to *S. aureus*, the model reproduced 14% of the test set molecules. As the results show, the algorithm not only generated valid screening molecules conserving the properties of the training data, but was also able to transfer the modifications imposed by the structures of a smaller data set in the fine-tuning process.

Olivecrona et al. [140] implemented a DL model based on RNNs to generate molecules with certain specified properties. Specifically, they worked on three distinct cases: these consisted in generating molecules that did not contain sulfur atoms, generating analogues to Celecoxib, and generating compounds active against a biological target, the dopamine type 2 receptor (DRD2). To train the generative model, a set of ~1.5 million structures were extracted from ChEMBL containing between 10 and 50 heavy atoms and the following elements: H, B, C, N, O, F, Si, P, S, Cl, Br and I. Inputs and outputs of this model were SMILES. On the other hand, another set of molecules was used to train a ML predictor for activity against DRD2. The corresponding bio-activity data was extracted from the ExCAPE database [141] and contained 7218 actives (pIC50 > 5) and 100,000 inactives (pIC50 < 5). Inputs for this activity model were the extended connectivity fingerprints with a diameter of 6 (ECFP6) and outputs were binary. The implemented generative model consisted of a combination of two RNNs, called the Prior and the Agent, and a reinforcement learning method. The Prior was trained to learn both the syntax of SMILES and the conditional probability distributions of the training set. The Agent, whose architecture is identical to the Prior, was trained with a reinforcement learning algorithm that modifies the Agent's probability distribution based on the desired properties of the structures to be generated. After training the Prior with the ChEMBL dataset, the authors assessed the generative capability of the Prior and showed that 94% of the sequences generated by it corresponded to valid molecular structures, out of which 90% were novel structures outside of the training set. The first scenario, where the Agent was trained to generate molecules excluding sulfur atoms, was used as a starting point to check the methodology. Effectively, the fraction of generated molecules without sulfur atoms increased from 0.66, corresponding to the Prior, to 0.98, corresponding to the Agent. To generate analogues to Celecoxib, the similarity between molecules was calculated via the Jaccard index within the FCFP4 fingerprints and the reward function was defined taking into account the degree of overlap between the fingerprints of the generated structures and the desired ones. Before generating analogues to the query structure, the authors checked that the model could effectively generate Celecoxib, which was accomplished even in the case where

the Prior was trained with a reduced set that include neither Celecoxib nor molecules with a similarity measure with it larger than 0.5 (1804 structures). Secondly, analogues to Celecoxib were generated. This was done by rewarding generated structures with a high Jaccard index with the query structure but strictly lower than 1.0. By sampling structures in the intermediate steps of the reinforcement procedure, the authors also suggested that this methodology could be also used for scaffold hopping. For the last case of study, the ExCAPE data was divided into training and test sets and an SVM model was trained to predict whether a structure is active or inactive against DRD2 with the training set. The Prior was trained on a subset of the ChEMBL dataset, where all DRD2 actives had been removed. After the reinforcement learning procedure, the fraction of predicted actives increased from 0.02, for structures generated by the Prior, to 0.96, for compounds generated by the corresponding Agent network. Remarkably, 7% of actives present in the ExCAPE test set used to train the SVM model were recovered by the Agent. Thus, the model generated structures that are experimentally confirmed active molecules which were not used at all in any training procedure. In a similar fashion to the previous cited work, Blaschke et al. [142] used DL to generate analogues to Celecoxib and generate molecules predicted to be active against DRD2. The main difference lies in the implemented methodologies to perform these tasks, which consisted in a series of different AE models.

These AE models were trained with a set of SMILES (both inputs and outputs), corresponding to ~1.3 million compounds extracted from ChEMBL, all of which had more than 10 heavy atoms. In particular, four types of AEs were used, two VAEs and two adversarial autoencoders (AAEs), which are, essentially, a combination of an AE with a GAN. Celecoxib was encoded into the corresponding latent space for each model and random latent vectors were sampled within a region around it. In each of the four latent spaces, the ECFP6 Tanimoto Similarity between the decoded structures and Celecoxib decreased with the distance (in the latent space). This indicated that the similarity principle is preserved for the encoded structures and represents an advantageous aspect for performing *de novo* design or compound optimization in the case of having a query structure. Also, Celecoxib and close analogues could be retrieved even in the case when all molecules with a FCFP4 Tanimoto similarity index greater than 0.5 to Celecoxib (1788 molecules) were excluded from the training set. To find novel compounds that were predicted to be active against DRD2, the authors used one of the AAEs. For this task, no active compounds were present in the training set of the generative model. The activity predictor model used was the same SVM classifier as mentioned in the previous work [140]. A Bayesian Optimization method was used to search structures with high probabilities of being active in the latent space. From ~370,000 sampled compounds, the average probability of being active was greater than 0.95. Although known actives were not retrieved, 11.5% of the generated compounds had a Tanimoto similarity greater than 0.35 to their closest active compound, and examples were shown of generated molecules that had the same scaffolds as validated actives. This indicated that the model could effectively generate novel compounds. In this regard, the authors concluded that AEs are a useful approach for tackling inverse QSAR problems.

An interesting study that also implemented an AAE was performed by Kadurin et al. [143], in which the generation process was guided towards anticancer compounds. The data used to train the model was extracted from the NCI-60 cancer cell line assay full dose response data [42], corresponding to 6252 compounds profiled on the MCF-7 cell line. As inputs and outputs, the model used a set of 166-bit Molecular ACCess System (MACCS) calculated with the Open Babel chemistry toolbox [144] and the concentration of the compounds in the corresponding cell line, respectively. The model was trained to encode and reconstruct both the binary fingerprints and the corresponding drug concentration of each molecule. Additionally, the latent layer had a reserved neuron responsible for the corresponding growth inhibition percentage value (GI), which indicated the increase or reduction in the number of tumor cells after drug treatment. These GI values were also included in the training procedure. Once the model was trained, 640 fingerprints were generated with their corresponding drug concentrations. After retrieving vectors with log concentration < -5.0 M and screening them against ~72 million compounds derived from Pubchem, 69 unique compounds were obtained. To validate these predictions, compounds were searched in the Pubchem BioAssay database [145] and several of them were identified as known or suspected anticancer agents of various kinds. Although further experimental validation is needed to assess whether the remaining predicted compounds actually display anticancer activity or not, this work showed that this methodology can lead to biologically relevant predictions.

It is evident that these DL models provide a great versatility to tackle different kind of problems, since the architectures of the models, or the combination of different methodologies, or the management of the input data may vary. Méndez-Lucio et al. [146] implemented a DL architecture with the objective of generating novel compounds that induce a specific gene expression profile. The dataset used in this work was derived from the L1000 Cmap database [147], which contains gene expression profiles for more than 25,000 perturbagens, where each signature reports the expression of 978 genes. From this database, perturbagens tested at 5 or 10 μM either on MCF7 or VCAP cell lines after 24 h of exposure were extracted, totaling ~32 thousand gene expressions, corresponding to ~20 thousand single compounds. As model inputs both SMILES and transcriptomic data were used, and SMILES were used as model outputs. Particularly, the model consists of an AE and two stacked conditional GANs; the AE was trained on a separate dataset of ~1.3 million compounds extracted from ChEMBL to encode and decode structures effectively, taking molecular SMILES as both inputs and outputs. The GANs were trained with the L1000 Cmap data. The first GAN received random noise and a gene expression signature, and generated a molecular representation which could be decoded into SMILES with the AE decoder. The second GAN worked in the same way, but instead of receiving noise, had the molecular representation generated by the first GAN as input. The condition imposed on these architectures was to generate molecules that induce the inputted gene expression profile when exposed to a cell. This was measured by a NN in the form of a classification score, which was a real value between 0 and 1 which indicated the probability of a molecule of inducing the desired gene expression. These NN inputs were both the latent representation and the specified gene expression. The described methodology was used to design inhibitor-like molecules. Under the hypothesis that the gene expression profiles from a knock-out protein and the inhibited protein would be similar, the authors utilized 148 gene expressions from the L1000 database, which correspond to the knock-out of 10 target proteins in MCF7. Also, more than 1000 known actives per protein were extracted from the Excape database. For each signature, 1000 molecular representations were generated and compared to the known inhibitors not used in the training set, via the Fraggle and Tanimoto similarity coefficients, using MACCS and Morgan Fingerprints. Generated molecules were more similar to known actives than the ones generated by a similarity search approach restricted to the data in the training set. Finally, the authors used their model to perform scaffold optimization. In particular, they optimized the benzene ring towards active-like compounds for the same ten targets as before, shown in Fig. 14. For this purpose, the model was fed with 148 pairs of encoded benzene ring-gene expression signature. Overall, the generated molecules showed similar molecular fragments to their nearest known active molecule (not used in the training set). In particular, 46% of the resulting molecules kept a benzene ring with the appropriate side chains added by the generative model.

Several works have been reported in the literature in which a graph representation of molecules is used as a model input [148–151]. In a recent study, Imrie et al. [152] used a graph-based deep generative model that integrated 3D structural information with the objective of performing fragment linking or scaffold hopping. Particularly, the

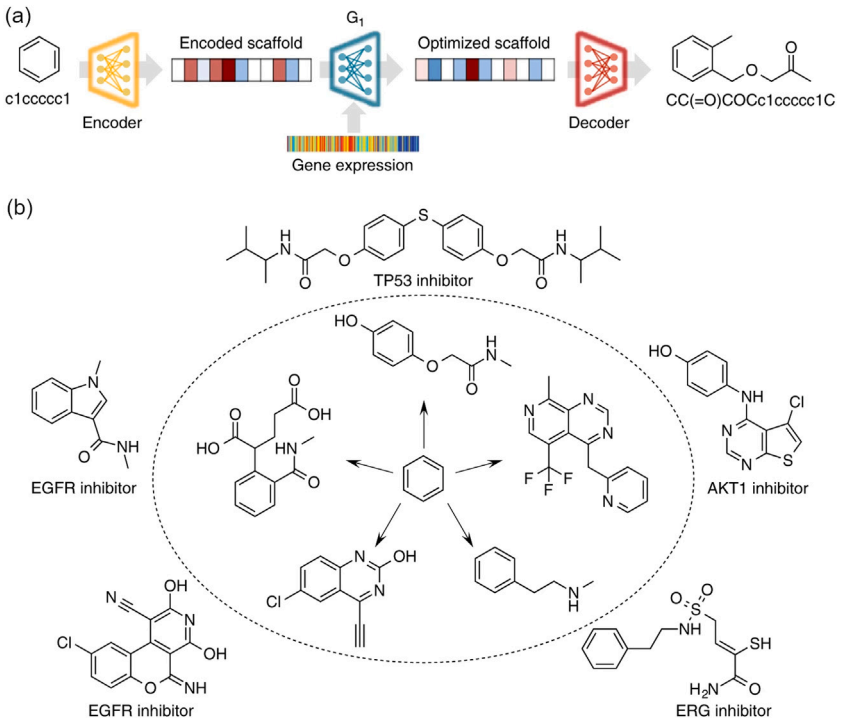

Fig. 14. Benzene ring scaffold optimization towards different targets using gene expression signatures. (a) The encoder transforms the SMILES of the scaffold into a latent representation that is fed into the generator (in blue) together with the desired gene expression signature. The output is the latent representation of an optimized molecule that can be decoded into a compound with a high probability to produce the gene expression signature. (b) Molecules generated by optimizing the benzene ring using the knock-out gene expression of AKT1, EGFR, ERG, and TP53 are shown inside the dotted circle and their closest active nearest neighbor outside the circle. Reprinted with permission from Ref. [146], licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

method took two partial structures and designed a molecule incorporating both fragments. A set of 250,000 compounds were extracted at random from the ZINC database and possible fragmentations of each molecule were constructed. This set of fragment–molecule pairs was filtered for specific 2D properties, including synthetic accessibility, ring aromaticity and pan-assay interference. Filtering these pairs led to ~419 thousand samples, with linkers between 3 and 12 atoms; 800 samples were selected at random to use as a test set, ensuring no overlap with the training set. To incorporate structural information, 3D conformers were generated for all the samples, and the lowest-energy conformation was taken as the reference 3D structure in each case. The authors used an external validation set composed of experimentally determined conformers; 285 protein–ligand complexes were extracted from the PDBbind core set [153]. After assessing possible fragmentations and applying the same 2D filters as before, 309 samples were obtained. The method was defined in a VAE benchmark which used as inputs two graphs containing information about distance and relative orientation between two fragments, and a latent vector representing the linker between the fragments, and outputs a graph in which the linker between the fragments was reproduced or replaced. The method was trained with the ZINC pairs of fragment–molecule, for which the latent vectors were, in each sample, the encoded linker of the corresponding molecule. To generate novel linkers between two fragments, random latent vectors can be fed to the model. Once trained, 250 molecules were generated for each pair of fragments, both for the test set and the external validation set. The authors evaluated if the generated compounds satisfied the 2D filters applied initially to the database and some 3D constraints, which encompassed the overlap between pharmacophoric features, a volumetric comparison between the generated and original molecule, and the RMSE between the coordinates of atoms in the starting fragments and the generated molecule. Not only the generated molecules resembled the constraints imposed on the training set, but also the method was able to design novel linkers. As it was mentioned in this study, several computational methods published for fragment linking or scaffold hopping [154–159] rely exclusively on a database of candidate fragments from which to select a linker. The authors compared their methodology with a more classical method which samples linkers from the training set data to joining two fragments. Designed linkers are exemplified in Fig. 15.

The generative model produced compounds with a higher 3D similarity, both to the initial fragments and the original molecules. Finally, the model was evaluated in three scenarios where a different study was used for comparison in each of them. Firstly, the method was used to generate inhibitors against inosine 5-monophosphate dehydrogenase. In comparison to a study performed by Trapero et al. [160], in which the authors identified promising inhibitors of the mentioned target, all reported potent molecules could be reproduced. Also, many of the generated molecules were scored higher than the original hits in a docking-based evaluation using AutoDock Vina [83]. Secondly, starting with an indazole-based inhibitor, the authors explored the ability of the method to change molecular scaffolds, particularly, towards the aminopyrazole-based inhibitor. This was made with the aim of comparing results with the work performed by Kamenecka et al. [161] in which the authors designed c-Jun N-terminal kinase 3 (JNK3) inhibitors with high selectivity over p38, another closely related mitogen-activated protein kinase. The method not only reproduced both the starting and final molecules reported in the baseline study, but also suggested many other scaffolds with high 3D similarity to the initial crystal data. Finally, the method was used in a proteolysis targeting chimera (PROTAC) case study, where was used the work performed by Farnaby et al. [162] for comparison, and showed that the method could generate novel linkers with similar geometries to PROTACS reported in the mentioned baseline work.

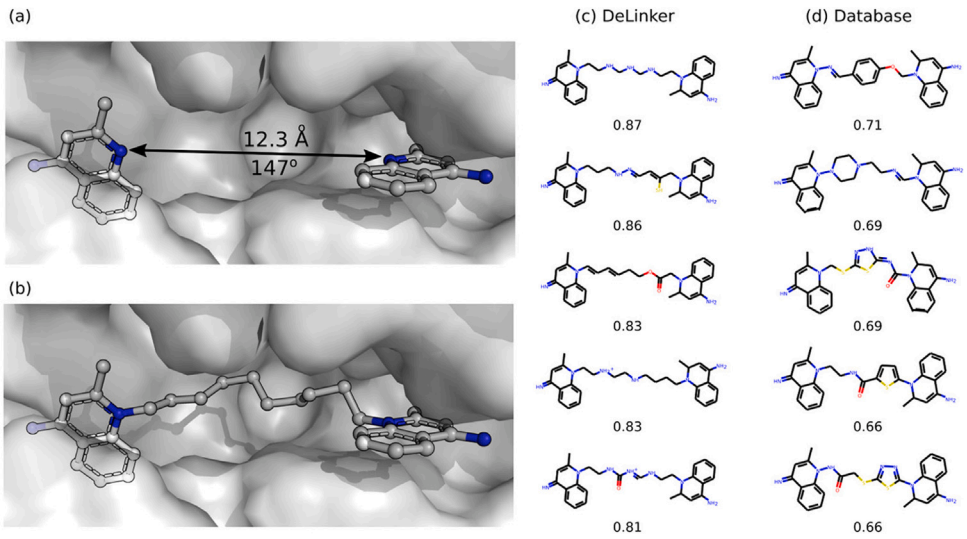

Fig. 15. Comparison of the author's methodology (DeLinker) with a database search method. (a) Fragmention of dequalinium [PDB ID: 3ARP, (b)]. The most 3D similar molecules, proposed by DeLinker and the database method are shown in (c) and (d), respectively, together with the 3D similarity score. Reprinted with permission from Ref. [152], licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/). Further permissions related to the material excerpted should be directed to the American Chemical Society (https://pubs.acs.org/doi/abs/10.1021/acs.jcim.9b01120).

3.3. Property prediction

Once a potent series of compounds that bind to a target has been identified, these should be optimized in terms of the properties that regulate their behavior within the organism. Thus, the pharmacokinetic profile, such as their absorption, distribution, metabolism and excretion (ADME) properties, and their toxicological effects should be improved. In this scenario, property prediction driven by AI methods is very appealing, as seen in several examples in the literature [163,164]. In what follows we comment on some implemented methods, including both ML and DL.

Lapins et al. [165], motivated by the fact that lipophilicity plays a crucial role in the pharmacokinetic profile of a drug candidate, used ML to predict water-octanol distribution coefficient (logD). The dataset used in this study was composed of ~1.6 million chemical compounds extracted from ChEMBL with logD annotations from the ACD/logD algorithm of Advanced Chemistry Development, Inc. [166]. For model inputs, these compounds where described with ~1.1 million molecular signature descriptors [167]: signature molecular descriptor constitutes a vector of occurrences of all atom signatures in the dataset, where an atom signature is a canonical representation of the atom's environment (i.e., neighboring and next-to neighboring atoms). Signatures distinguish between different atom and bond types, as well as between aromatic and aliphatic atoms in the atom's environment. The training set was composed of ~1.5 million compounds, and the test set using the remaining ~100,000. A SVM model was trained with the mentioned data. After optimizing the model with the test set, the trained SVM was evaluated with two external validation sets extracted from the literature. One of these was composed by 29 molecules [168] and the other by 72 compounds [169]; the correlation values between the predicted and actual values were $R^2 = 0.93$ and $R^2 = 0.98$, respectively. In addition to the SVM model, a conformal prediction algorithm was utilized, which outputs intervals around the predicted values of the SVM to satisfy a required confidence level. With respect to the results obtained with the validation sets, the median prediction intervals were ±0.4 log units at 80% validity (i.e., when the 80% of the real values lie within the predicted values and their corresponding intervals) and ±0.6 log units at 90% validity, which the authors considered reasonable to be useful.

Schyman et al. [170] implemented a modified kNN algorithm, the variable nearest neighbor (vNN), to predict several ADMET properties. In the following, we list the predicted properties and, in parenthesis, the number of compounds of the training set, as well as the source from which they were extracted: blood–brain barrier permeability (353) [171,172], mitochondrial toxicity (6261) [173]; cytotoxicity (6097) [174]; drug-induced liver injury (1427) [175]; cytochrome p450 enzymes (CYPs) inhibition, corresponding to the following CYPs, 1A2 (7558), 2D6 (7805), 2C9(8072), 2C19 (8155), and 3A4 (10,373) [174]; blockade of the human ether-à-go-go-related gene function (685) [174,176]; P-glycoprotein substrates and inhibitors (822 and 2304) [177–179]; chemical mutagenicity (6512) [180]; maximum recommended therapeutic dose (1184) [181]; and human liver microsomal stability (3219) [174]. For model inputs, compounds were described with extended-connectivity fingerprints with a diameter of four chemical bonds (ECFP4) [137], and the similarity between them was calculated using the Tanimoto distance. A vNN model was constructed for each of the mentioned sets; the kNN method relies on the premise that compounds with similar structures have similar properties. Given a similarity metric, one way of predicting a property value for a given compound using kNN is taking the weighted average property values of the k-nearest neighbors of that compound (which belong to the training set), where k is a fixed integer, in which closer neighbors contribute more to the predicted value. The problem with this method is that it does not take into account how structurally dissimilar the k nearest neighbors could be from the given compound. An alternative approach is to use a predetermined similarity threshold. In this study, the implemented vNN used all nearest neighbors that met a structural similarity criterion. When no nearest neighbor met the criterion, the vNN method made no prediction. The method performance was assessed with a 10-fold cross validation and several metrics. Overall, the algorithm obtained accuracy between 0.71 and 0.91, sensitivity values between 0.61 and 0.94, and specificity values ranging from 0.73 to 0.96, displaying a good performance across several tasks.

Recently, Wenzel et al. [182] applied a DL methodology to the prediction of ADMET properties. Data was extracted from public and private sources; as before, we specify in parenthesis the total number of molecules available in each case. Two properties were extracted from ChEMBL: metabolic clearance for three species; human (5348), mouse (2166) and rat (790); and passive permeability in Caco-2 cells (2582). The data from Sanofi contributed three properties: metabolic lability from eight species: human (57,635), rat (51,355), mouse (48,242), guinea pig (1533), dog (1056), macaque (588), rabbit (553) and monkey (246), passive permeability (46,440), and distribution coefficient

(81,309). Compounds were described using atom pairs (APs) [59] and pharmacophoric donor–acceptor pairs (DPs) [183] calculated with RDKit [184]. APs and DPs were combined as suggested by Ma et al. [185]; the number of these AP-DP descriptors varied across the mentioned sets from ~3500 to ~9900. In each of the mentioned sets, the data was split into a training set, a test set (used as a control set to avoid overfitting), and an external validation set. Model performance was measured in terms of the squared correlation coefficient R^2. The performance of single-task DNN and multi-task DNN was compared. In the first scenario, using the ChEMBL dataset, four single DNNs were trained, each one corresponding to the prediction of one property, concerning human, rat, and mouse metabolic clearance values, and permeability in Caco-2 cells. It was shown that the use of human, rat and mouse data with one multi-task DNN improved the R^2 value of the model, in comparison to its single DNN counterparts. These improvements ranged from 2.7% to 19.3%. Only in one of the validation sets there was a decrease of the performance of −3.4%. Adding the permeability information further improved these results, obtaining higher values in the whole test and validation sets related to the metabolic clearance. In the case of predicting the Caco-2 permeability, the single DNN performed better. The decrease in R^2 when using multi-task learning was of −13.2%. This indicated, in one hand, that highly correlated data improves performance in a multi-task learning scenario, and in the other hand, that the use of non-related data may be useful in some cases but not in others. This was repeated with the Sanofi dataset, using the metabolic lability instead of the metabolic clearance, and similar results were obtained. Also, this analysis was extended using the remaining species, and it was shown that relevant features from the larger datasets could be transferred to the smaller ones in some cases. For example, an improvement of 30% was obtained in the dog, macaque and rabbit models. The authors later tested their models to predict passive permeability, metabolic lability and lipophilicity of two congeneric series of ligands with the aim of evaluating whether the methodology could effectively detect local changes in structures (which are translated into ADME property differences). One series consisted of 199 polar CXCR3 antagonists [186,187], while the other was composed by 48 Renin inhibitors [188–190]. As before, the data was split into training, test and validation sets. As the models achieved an excellent correlation with experimental data, the authors suggested that predicted outliers should be tested and experimentally validated. Finally, a visualization method, called Response Map, was proposed for model interpretation, which consists, essentially, in depicting property changes following the fragmentation or derivatization of a parent structure. As the authors showed with several samples of the CXCR3 antagonists and the Renin inhibitors, this approach provides useful information for compound design and optimization. For example, from the response map of the CXCR3 antagonist Nefazodone, it was concluded that a peripheral decoration of the molecule with a –CN substituent would be an effective way to reduce its metabolic lability, both in human and mouse species.

Toxicity optimization is a demanding step in the preclinical stage of a drug discovery campaign, being an expensive and time-consuming activity [191]. We have already mentioned some ML applications in this regard (cf. for example Ref. [192]), but DL methodologies also excelled in the task. This fact was shown in the Tox21 [193] data challenge, where a comparison of computational methods for toxicity prediction was performed in terms of a binary classification problem. The data set used in the competition consisted of 12,707 compounds with 12 different toxic effects measured, although not all the compounds were assayed for each of them. Approximately, 54% of the compounds were assayed for 10 or more effects, and only 500 were measured in only one assay. Toxic effects were divided in two main categories: one associated with the activation of several bio-molecular targets (nuclear receptor panel), which were related to the disruption of the endocrine system function [194,195]; and the other related to the activation of stress response pathways (stress response panel), which can lead to liver injury or cancer [196–198]. For the participants, the data was divided in 11,764 compounds for the training set, 296 molecules for the leaderboard set and the remaining 647 structures were used as a test set. Performance was measured with the ROC-AUC.

Mayr et al. [199], the winners of this competition, implemented a DL architecture, DeepTox, which consisted of an ensemble of methods, such as DNN, RF, SVM and elastic net, in which high priority was given to the DNN. For model inputs the authors calculated as many features as possible. These included, for example, MW, Van der Waals volume, partial charge, atom counts, surface areas, presence or absence of predefined toxicophore features, and MACCS binary fingerprints. Finally, multi-task learning DNNs were also implemented, showing that in 10 of the 12 tasks, the multi-task NNs outperformed single task NN. The described DL pipeline outperformed all other competitors, which included other ML methods like RF, SVM, kNN and Naive Bayes classifiers. DeepTox exhibited a high-level performance across all the tasks, not ranking below 5th place in any sub-challenge. It ranked first in 6 of the 12 tasks, and won both on the nuclear receptor and the stress response panels, as well as the overall competition. Finally, the authors trained a multitask DNN only with ECFP4 fingerprints, which encode substructures around each atom in a compound. Interestingly, by analyzing the activation of the neurons over the set of compounds, these were associated with several toxicophores, showing that the DL method encodes the last ones in their hidden layers. Some examples of the detected toxicophores are shown in Fig. 16

Some examples illustrated in Fig. 16 show that the first layers detect small toxicophores, such as sulfonic acid groups, and the higher layers tend to correlate with larger toxicophores. This is an important result in the sense that the authors provided a way of interpreting the DL model, and based on this interpretation, proposed that novel toxicophores may also be encoded in the hidden layers.

Xu et al. [175] implemented a DL application to predict drug-induced liver injury (DILI) in terms of a binary classification task. Four publicly available data sets with compounds annotated as DILI-positive or DILI-negative were extracted from the literature containing each a total amount of 375 [200], 1184 [201], 320 [202] and 236 [203] compounds. The last two sets were only used as validation data, while the first two were split into training and validation sets. Also, another set was built by combining three of these datasets, containing 673 compounds, which was also split for training and validation. The authors implemented a recursive NN that takes as inputs undirected graphs constructed from the corresponding molecular structures. Both in the training set and the corresponding validation set, this methodology achieved better results in comparison to the original method from which the data was obtained, which included, a decision forest algorithm trained on Mold^2 descriptors [204], and an ensemble of methods concerning SVM, kNN, and Naive Bayes classifier, using PaDEL descriptors [205]. Performance metrics varied according to the baseline study, but included accuracy, sensitivity, specificity, MCC, and geometric mean. In the case of using the set composed of 375 compounds, of which 190 samples were used for training, a significant decrease in the corresponding metrics was observed in the validation set, indicating possible overfitting of the model. With respect to the combined dataset, the model achieved solid results. Specifically, using 60 random splits of training and validation sets, the mean performance in the validation was as follows: accuracy 84.3, sensitivity 79.4, specificity 90.9, MCC 0.70, and geometric mean 85.0. Finally, the authors trained both a NN and a DNN with the combined dataset, using PaDEL and Mold^2 descriptors to describe model inputs, and showed that their original methodology outperformed them, stating that the used DL architecture had an aggregated value for this classification task.

4. Discussion

A recurrent discussion related to the implementation of AI in drug discovery is the "hype versus reality" topic. While expectations are

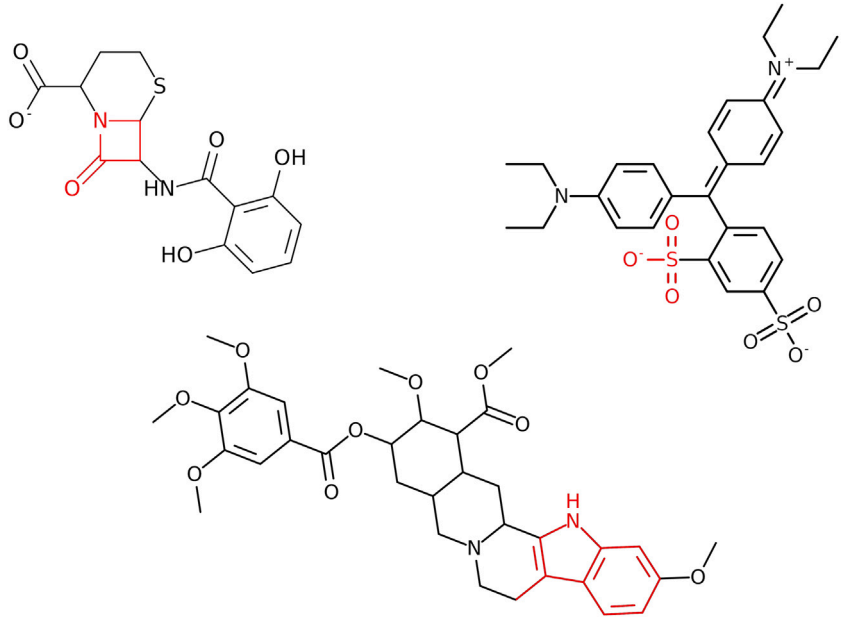

Fig. 16. Examples of toxicophore features (red) detected by the trained NN. Top compounds are related to high activation of neurons in the first layer. The bottom compound is related to high activated neurons in a deeper layer. Figure prepared with ICM (Molsoft LLC, San Diego, CA). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

high, advances have been notable. The growing progress made by scientific research has been matched by an increasing partnership between the pharma industry and AI companies. Evidence indicates that these technologies have come to stay. We do not know yet how far we are from a new era of AI-driven drug discovery, and to which extent this complex process can be further optimized, but no doubt great advances have been made.

In the years to come, we do expect an increasing number of AI applications in drug discovery, basically, for two main reasons. In the first place, it is reasonable to anticipate more DL implementations due to the growing amount of available data and increasing computational power. In the second place, having witnessed how ML applications transitioned to DL implementations, as well as how the latest techniques developed in the last years were consistently implemented in drug discovery, it is almost guaranteed that, in the future, the state-of-the-art AI methodologies will be adapted to the drug discovery pipeline.

Moreover, the development of AI methodologies has, objectively, a major advantage compared to improvements in other techniques: AI tools are not drug discovery-specific. The fact that AI encompasses a wider spectrum of areas in which to be applied enhances the possibilities of new applications. Indeed, the implementation of novel AI techniques directly into drug discovery is a fact, and we count on the possibility of adapting new successful AI algorithms from other areas.

Upon the discussed implementations of AI in different steps of the early drug discovery process, we do not see a strong relationship between a given drug discovery stage and a specific AI methodology. Certainly, NNs play a fundamental role, but the intrinsic versatility of this method makes each NN implementation different from the other. For example, in the case of generative modeling, we discussed implementations of RNNs, VAEs, GANs, AAEs, and even their combinations.

Plenty of examples have been discussed in which the results predicted by an AI methodology were validated in a retrospective fashion. Hopefully, AI applications will tend to be validated prospectively. In this respect, there is a race, particularly in generative modeling, to place an AI discovered drug into the market. A particular work that has had considerable media resonance is the one performed by Zhavoronkov et al. [206]. The main objective of this study was to design, synthesize and test inhibitors of a kinase, the discoidin domain receptor family member 1 (DDR1). To design these compounds, the authors implemented a generative model based on AEs and reinforcement learning. One of the synthesized structures exhibited an IC_{50} of 10 nM in an enzymatic kinase assay, an IC_{50} of 10.3 nM measured by autophosporylation in U2OS cells, a suitable half-life and clearance values measured in human, rat mouse and dog liver microsomes, and showed a favorable pharmacokinetic profile in a rodent model. As pointed out later by Walters and Murcko [207,208], this designed compound was very similar to a molecule used in the training set, Potinatib, which is a marketed multi-kinase inhibitor which exhibits an IC_{50} of 9 nM in an enzymatic kinase assay.

Particularly, one of the main aspects that Walters and Murcko suggested is the use of standardized criteria to evaluate the compounds generated with a generative model. This not only complements the aforementioned study, but would enrich the area of generative modeling in general. In summary, despite the intrinsic versatility and value of DL, which provides a wide range of possibilities to explore, appropriate measures must be taken into account to evaluate these new methodologies appropriately.

Concerning the successful implementations of AI in drug discovery there is a key aspect to bear into account, namely, the interpretability of the models. While this obviously belongs in general to AI, it takes on a particular relevance in the medicinal chemistry context. Complex and non-linear connections between inputs and their corresponding outputs are not always easily understood, and with the growing number of applications in the field, it is becoming important to assess and understand what decisions the AI method is taking, and why it is taking them. Evidently, this is a hard task, but cooperation between medicinal chemists and data scientists may lead to important steps towards more easily interpretable models. An excellent discussion on this topic, coupled with insights of the mathematical models, is given by Jiménez-Luna et al. [209].

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

This work was supported by the National Agency for the Promotion of Science and Technology, Argentina (ANPCyT) (PICT-2017-3767). CNC thanks Molsoft LLC (San Diego, CA) for providing an academic license for the ICM program.

References

[1] J.A. DiMasi, H.G. Grabowski, R.W. Hansen, J. Health Econ. 47 (2016) 20–33.
[2] J.-L. Reymond, L. Ruddigkeit, L. Blum, R. van Deursen, Wiley Interdiscip. Rev. Comput. Mol. Sci. 2 (2012) 717–733.
[3] S.S. Phatak, C.C. Stephan, C.N. Cavasotto, Expert Opin. Drug Discovery 4 (2009) 947–959.
[4] C.N. Cavasotto, M.G. Aucar, N.S. Adler, Int. J. Quantum Chem. 119 (2019) e25678.
[5] F. Spyrakis, C.N. Cavasotto, Arch. Biochem. Biophys. 583 (2015) 105–119.
[6] K. Heikamp, J. Bajorath, Chem. Biol. Drug Des. 81 (2013) 33–40.
[7] J. Vázquez, M. López, E. Gibert, E. Herrero, F.J. Luque, Molecules 25 (2020) 4723.
[8] W.L. Jorgensen, Angew. Chem. Int. Ed. Engl. 51 (2012) 11680–4.
[9] C.N. Cavasotto, A.J. Orry, Curr. Top. Med. Chem. 7 (2007) 1006–1014.
[10] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, T. Blaschke, Drug Discov. Today 23 (2018) 1241–1250.
[11] P.B. Jørgensen, M.N. Schmidt, O. Winther, Mol. Inform. 37 (2018) 1700133.
[12] K.A. Carpenter, D.S. Cohen, J.T. Jarrell, X. Huang, Future Med. Chem. 10 (2018) 2557–2567.
[13] H.S. Chan, H. Shan, T. Dahoun, H. Vogel, S. Yuan, Trends Pharmacol. Sci. (2019).
[14] K.-K. Mak, M.R. Pichika, Drug Discov. Today 24 (2019) 773–780.
[15] C.H. Wong, K.W. Siah, A.W. Lo, Biostatistics 20 (2018) 273–286.
[16] J. Hughes, S. Rees, S. Kalindjian, K. Philpott, Br. J. Pharmacol. 162 (2011) 1239–1249.
[17] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[18] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, MIT Press, Cambridge, 2016.
[19] A. Ng, Machine Learning Yearning, Stanford Press, 2017.
[20] E. Ferrero, I. Dunham, P. Sanseau, J. Transl. Med. 15 (2017) 182.
[21] G. Koscielny, P. An, D. Carvalho-Silva, J.A. Cham, L. Fumis, R. Gasparyan, S. Hasan, N. Karamanis, M. Maguire, E. Papa, et al., Nucl. Acids Res. 45 (2017) D985–D994.
[22] Informa Pharmaprojects, https://pharmaintelligence.informa.com/products-and-services/data-and-analysis/pharmaprojects. (Accessed September 2020).
[23] MEDLINE, https://www.nlm.nih.gov/bsd/medline.html. (Accessed September 2020).
[24] Q. Wang, Y. Feng, J. Huang, T. Wang, G. Cheng, PLoS One 12 (2017) e0176486.
[25] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, et al., Nucl. Acids Res. 39 (2010) D1035–D1041.
[26] P. Rice, I. Longden, A. Bleasby, Emboss: the European molecular biology open software suite, 2000.
[27] J.D. Bendtsen, H. Nielsen, G. Von Heijne, S. Brunak, J. Mol. Biol. 340 (2004) 783–795.
[28] A. Krogh, B. Larsson, G. Von Heijne, E.L. Sonnhammer, J. Mol. Biol. 305 (2001) 567–580.
[29] J.C. Wootton, S. Federhen, Comput. Chem. 17 (1993) 149–163.
[30] L.J. Jensen, R. Gupta, H.-H. Staerfeldt, S. Brunak, Bioinformatics 19 (2003) 635–642.
[31] K. Julenius, A. Mølgaard, R. Gupta, S. Brunak, Glycobiology 15 (2005) 153–164.
[32] T.M. Bakheet, A.J. Doig, Bioinformatics 25 (2009) 451–457.
[33] J. Jeon, S. Nim, J. Teyra, A. Datti, J.L. Wrana, S.S. Sidhu, J. Moffat, P.M. Kim, Genome Med. 6 (2014) 1–18.
[34] X. Chen, Z.L. Ji, Y.Z. Chen, Nucl. Acids Res. 30 (2002) 412–415.
[35] R. Marcotte, K.R. Brown, F. Suarez, A. Sayad, K. Karamboulas, P.M. Krzyzanowski, F. Sircoulomb, M. Medrano, Y. Fedyshyn, J.L. Koh, et al., Cancer Discov. 2 (2012) 172–189.
[36] J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A.A. Margolin, S. Kim, C.J. Wilson, J. Lehár, G.V. Kryukov, D. Sonkin, et al., Nature 483 (2012) 603–607.
[37] S.A. Forbes, N. Bindal, S. Bamford, C. Cole, C.Y. Kok, D. Beare, M. Jia, R. Shepherd, K. Leung, A. Menzies, et al., Nucl. Acids Res. 39 (2010) D945–D950.
[38] A. Bossi, B. Lehner, Mol. Syst. Biol. 5 (2009) 260.
[39] R. Kumar, K. Chaudhary, S. Gupta, H. Singh, S. Kumar, A. Gautam, P. Kapoor, G.P. Raghava, Sci. Rep. 3 (2013) 1445.
[40] N. Bakkar, T. Kovalik, I. Lorenzini, S. Spangler, A. Lacoste, K. Sponaugle, P. Ferrante, E. Argentinis, R. Sattler, R. Bowser, Acta Neuropathol. 135 (2018) 227–247.
[41] N.S. Madhukar, P.K. Khade, L. Huang, K. Gayvert, G. Galletti, M. Stogniew, J.E. Allen, P. Giannakakou, O. Elemento, Nature Commun. 10 (2019) 1–14.
[42] R.H. Shoemaker, Nat. Rev. Cancer 6 (2006) 813–823.
[43] J. Lamb, E.D. Crawford, D. Peck, J.W. Modell, I.C. Blat, M.J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K.N. Ross, et al., Science 313 (2006) 1929–1935.
[44] J. Lamb, Nature Rev. Cancer 7 (2007) 54–60.
[45] M. Kuhn, M. Campillos, I. Letunic, L.J. Jensen, P. Bork, Mol. Syst. Biol. 6 (2010) 343.
[46] Q. Li, T. Cheng, Y. Wang, S.H. Bryant, Drug Discov. Today 15 (2010) 1052–1057.
[47] B. Chen, D.J. Wild, J. Mol. Graph. Model. 28 (2010) 420–426.
[48] V. Law, C. Knox, Y. Djoumbou, T. Jewison, A.C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu, et al., Nucl. Acids Res. 42 (2014) D1091–D1097.
[49] D.S. Wishart, C. Knox, A.C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, M. Hassanali, Nucl. Acids Res. 36 (2008) D901–D906.
[50] T.I. Oprea, H. Matter, Curr. Opin. Chem. Biol. 8 (2004) 349–358.
[51] T. Lengauer, C. Lemmen, M. Rarey, M. Zimmermann, Drug Discov. Today 9 (2004) 27–34.
[52] C. Sotriffer, in: C.N. Cavasotto (Ed.), In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications, CRC Press, Taylor & Francis Group, Boca Raton, FL, 2015, pp. 155–188.
[53] A. Ciancetta, S. Moro, in: C.N. Cavasotto (Ed.), In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications, CRC Press, Taylor & Francis Group, Boca Raton, FL, 2015, pp. 189–213.
[54] D. Rognan, in: C. Sotriffer (Ed.), Virtual Screening. Principles, Challenges and Practical Guidelines, in: Methods and Principles in Medicinal Chemistry, vol. 48, Wiley-VCH Verlag, Weinheim, Germany, 2011, pp. 153–176.
[55] C.N. Cavasotto, in: F. Luque, X. Barril (Eds.), Physico-Chemical and Computational Approaches to Drug Discovery, Royal Society of Chemistry, London, 2012, pp. 195–222.
[56] T. Kaserer, K. Beck, M. Akram, A. Odermatt, D. Schuster, Molecules 20 (2015) 22799–22832, http://dx.doi.org/10.3390/molecules201219880.
[57] D. Plewczynski, S.A. Spieser, U. Koch, J. Chem. Inf. Model. 46 (2006) 1098–1106.
[58] MDL Information Systems Inc., San Leandro, CA, MACCS Drug Data Report, 2006, http://www.mdli.com. (Accessed March 2006).
[59] R.E. Carhart, D.H. Smith, R. Venkataraghavan, J. Chem. Inf. Comput. Sci. 25 (1985) 64–73.
[60] S.L. Kinnings, N. Liu, P.J. Tonge, R.M. Jackson, L. Xie, P.E. Bourne, J. Chem. Inf. Model. 51 (2011) 408–419.
[61] Z. Zsoldos, D. Reid, A. Simon, S.B. Sadjad, A.P. Johnson, J. Mol. Graph. Model. 26 (2007) 198–212.
[62] T. Liu, Y. Lin, X. Wen, R.N. Jorissen, M.K. Gilson, Nucl. Acids Res. 35 (2007) D198–D201.
[63] N. Huang, B.K. Shoichet, J.J. Irwin, J. Med. Chem. 49 (2006) 6789–6801.
[64] G.E. Dahl, N. Jaitly, R. Salakhutdinov, 2014, arXiv preprint arXiv:1406.1231.
[65] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B.A. Shoemaker, P.A. Thiessen, B. Yu, et al., Nucl. Acids Res. 47 (2019) D1102–D1109.
[66] A. Mauri, V. Consonni, M. Pavan, R. Todeschini, Match 56 (2006) 237–248.
[67] T. Unterthiner, A. Mayr, G. Klambauer, M. Steijaert, J.K. Wegner, H. Ceulemans, S. Hochreiter, Proceedings of the Deep Learning Workshop at NIPS, Vol. 27, pp. 1–9.
[68] D. Mendez, A. Gaulton, A.P. Bento, J. Chambers, M. De Veij, E. Félix, M.P. Magariños, J.F. Mosquera, P. Mutowo, M. Nowotka, et al., Nucl. Acids Res. 47 (2019) D930–D940.
[69] E.B. Lenselink, N. Ten Dijke, B. Bongers, G. Papadatos, H.W. Van Vlijmen, W. Kowalczyk, A.P. IJzerman, G.J. Van Westen, J. Cheminform. 9 (2017) 1–14.
[70] I. Wallach, M. Dzamba, A. Heifets, 2015, arXiv preprint arXiv:1510.02855.
[71] M.M. Mysinger, M. Carchia, J.J. Irwin, B.K. Shoichet, J. Med. Chem. 55 (2012) 6582–6594.
[72] C. Da, D. Kireev, J. Chem. Inf. Model. 54 (2014) 2555–2561.
[73] Z. Deng, C. Chuaqui, J. Singh, J. Med. Chem. 47 (2004) 337–344.
[74] V.I. Pérez-Nueno, O. Rabal, J.I. Borrell, J. Teixidó, J. Chem. Inf. Model. 49 (2009) 1245–1260.
[75] J. Gabel, J. Desaphy, D. Rognan, J. Chem. Inf. Model. 54 (2014) 2807–2815.
[76] R. Spitzer, A.N. Jain, J. Comput. Aided Mol. Des. 26 (2012) 687–699.
[77] R.G. Coleman, M. Carchia, T. Sterling, J.J. Irwin, B.K. Shoichet, PLoS One 8 (2013) e75992.
[78] R.G. Coleman, T. Sterling, D.R. Weiss, J. Comput. Aided Mol. Des. 28 (2014) 201–209.
[79] W.J. Allen, T.E. Balius, S. Mukherjee, S.R. Brozell, D.T. Moustakas, P.T. Lang, D.A. Case, I.D. Kuntz, R.C. Rizzo, J. Comput. Chem. 36 (2015) 1132–1156.
[80] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, D.R. Koes, J. Chem. Inf. Model. 57 (2017) 942–957.
[81] J. Gomes, B. Ramsundar, E.N. Feinberg, V.S. Pande, 2017, arXiv preprint arXiv:1703.10603.
[82] J.C. Pereira, E.R. Caffarena, C.N. dos Santos, J. Chem. Inf. Model. 56 (2016) 2495–2506.
[83] O. Trott, A.J. Olson, J. Comput. Chem. 31 (2010) 455–461.
[84] P.T. Lang, S.R. Brozell, S. Mukherjee, E.F. Pettersen, E.C. Meng, V. Thomas, R.C. Rizzo, D.A. Case, T.L. James, I.D. Kuntz, Rna 15 (2009) 1219–1230.
[85] M. Arciniega, O.F. Lange, J. Chem. Inf. Model. 54 (2014) 1401–1411.
[86] J.D. Durrant, A.J. Friedman, K.E. Rogers, J.A. McCammon, J. Chem. Inf. Model. 53 (2013) 1726–1735.

[87] M.A. Neves, M. Totrov, R. Abagyan, J. Comput. Aided Mol. Des. 26 (2012) 675–686.
[88] J.B. Cross, D.C. Thompson, B.K. Rai, J.C. Baber, K.Y. Fan, Y. Hu, C. Humblet, J. Chem. Inf. Model. 49 (2009) 1455–1474.
[89] R. Abagyan, M. Totrov, D. Kuznetsov, J. Comput. Chem. 15 (1994) 488–506.
[90] R.A. Friesner, J.L. Banks, R.B. Murphy, T.A. Halgren, J.J. Klicic, D.T. Mainz, M.P. Repasky, E.H. Knoll, M. Shelley, J.K. Perry, et al., J. Med. Chem. 47 (2004) 1739–1749.
[91] Y.O. Adeshina, E.J. Deeds, J. Karanicolas, Proc. Natl. Acad. Sci. 117 (2020) 18477–18488.
[92] J.D. Durrant, J.A. McCammon, J. Chem. Inf. Model. 51 (2011) 2897–2903.
[93] P.J. Ballester, J.B. Mitchell, Bioinformatics 26 (2010) 1169–1175.
[94] P.J. Ballester, A. Schreyer, T.L. Blundell, J. Chem. Inf. Model. 54 (2014) 944–955.
[95] H. Li, K.-S. Leung, M.-H. Wong, P.J. Ballester, Mol. Inf. 34 (2015) 115–126.
[96] M. Wójcikowski, P.J. Ballester, P. Siedlecki, Sci. Rep. 7 (2017) 46710.
[97] M. Wójcikowski, M. Kukiełka, M.M. Stepniewska-Dziubinska, P. Siedlecki, Bioinformatics 35 (2019) 1334–1341.
[98] R.F. Alford, A. Leaver-Fay, J.R. Jeliazkov, M.J. O'Meara, F.P. DiMaio, H. Park, M.V. Shapovalov, P.D. Renfrew, V.K. Mulligan, K. Kappel, et al., J. Chem. Theory Comput. 13 (2017) 3031–3048.
[99] J.D. Durrant, J.A. McCammon, J. Mol. Graph. Model. 29 (2011) 888–893.
[100] ChemAxon, https://chemaxon.com/. (Accessed September 2020).
[101] Szybki, https://www.eyesopen.com/szybki. (Accessed September 2020).
[102] S.M. Vogel, M.R. Bauer, F.M. Boeckler, J. Chem. Inf. Model. 51 (2011) 2650–2665.
[103] M.R. Bauer, T.M. Ibrahim, S.M. Vogel, F.M. Boeckler, J. Chem. Inf. Model. 53 (2013) 1447–1462.
[104] A. Bazzoli, S.P. Kelow, J. Karanicolas, PLoS One 10 (2015) e0140359.
[105] F. Gentile, V. Agrawal, M. Hsing, A.-T. Ton, F. Ban, U. Norinder, M.E. Gleave, A. Cherkasov, ACS Cent. Sci. (2020).
[106] J.L. Sussman, D. Lin, J. Jiang, N.O. Manning, J. Prilusky, O. Ritter, E.E. Abola, Acta Crystallogr. D 54 (1998) 1078–1084.
[107] M. McGann, J. Comput. Aided Mol. Des. 26 (2012) 897–906.
[108] J. Jiménez, L. Pérez-Benito, G. Martínez-Rosell, S. Sciabola, R. Torella, G. Tresadern, G. De Fabritiis, Chem. Sci. 10 (2019) 10911–10918.
[109] L. Wang, Y. Wu, Y. Deng, B. Kim, L. Pierce, G. Krilov, D. Lupyan, S. Robinson, M.K. Dahlgren, J. Greenwood, et al., J. Am. Chem. Soc. 137 (2015) 2695–2703.
[110] D.L. Mobley, M.K. Gilson, Annu. Rev. Biophys. 46 (2017) 531–558.
[111] F. Boyles, C.M. Deane, G.M. Morris, Bioinformatics 36 (2020) 758–764.
[112] P.J. Ballester, Drug Discov. Today Technol. (2020).
[113] N. Singh, L. Chaput, B.O. Villoutreix, J. Chem. Inf. Model. 60 (2020) 3910–3934.
[114] P. Chen, Y. Ke, Y. Lu, Y. Du, J. Li, H. Yan, H. Zhao, Y. Zhou, Y. Yang, J. Cheminform. 11 (2019) 52.
[115] C.N. Cavasotto, J.I. Di Filippo, Mol. Inform. (2020) 2000115, http://dx.doi.org/10.1002/minf.202000115, in press.
[116] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, A. Zhavoronkov, Mol. Pharmaceut. 13 (2016) 2524–2530.
[117] NIH LINCS Program, http://www.lincsproject.org/. (Accessed September 2020).
[118] MeSH, https://www.nlm.nih.gov/mesh/. (Accessed September 2020).
[119] A.A. Buzdin, A.A. Zhavoronkov, M.B. Korzinkin, L.S. Venkova, A.A. Zenin, P.Y. Smirnov, N.M. Borisov, Front. Genet. 5 (2014) 55.
[120] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, Nucleic Acids Res. 28 (2000) 235–242.
[121] L. Xie, P.E. Bourne, BMC Bioinform. 8 (2007) S9.
[122] L. Xie, P.E. Bourne, Proc. Natl. Acad. Sci. 105 (2008) 5441–5446.
[123] L. Xie, L. Xie, P.E. Bourne, Bioinformatics 25 (2009) i305–i312.
[124] J.M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N.M. Donghia, C.R. MacNair, S. French, L.A. Carfrae, Z. Bloom-Ackerman, et al., Cell 180 (2020) 688–702.
[125] S.M. Corsello, J.A. Bittker, Z. Liu, J. Gould, P. McCarren, J.E. Hirschman, S.E. Johnston, A. Vrcic, B. Wong, M. Khan, et al., Nature Med. 23 (2017) 405–408.
[126] Broad Institute of MIT and Harvard, https://www.broadinstitute.org/. (Accessed September 2020).
[127] T. Sterling, J.J. Irwin, J. Chem. Inf. Model. 55 (2015) 2324–2337.
[128] K.M. Gayvert, N.S. Madhukar, O. Elemento, Cell Chem. Biol. 23 (2016) 1294–1301.
[129] Z. Wu, B. Ramsundar, E.N. Feinberg, J. Gomes, C. Geniesse, A.S. Pappu, K. Leswing, V. Pande, Chem. Sci. 9 (2018) 513–530.
[130] S. Jang, L.-R. Yu, M.A. Abdelmegeed, Y. Gao, A. Banerjee, B.-J. Song, Redox Biol. 6 (2015) 552–564.
[131] S.K. De, J.L. Stebbins, L.-H. Chen, M. Riel-Mehan, T. Machleidt, R. Dahl, H. Yuan, A. Emdadi, E. Barile, V. Chen, et al., J. Med. Chem. 52 (2009) 1943–1952.
[132] R. Gómez-Bombarelli, J.N. Wei, D. Duvenaud, J.M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T.D. Hirzel, R.P. Adams, A. Aspuru-Guzik, ACS Cent. Sci. 4 (2018) 268–276.
[133] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Jozefowicz, S. Bengio, 2015, arXiv preprint arXiv:1511.06349.
[134] J.J. Irwin, T. Sterling, M.M. Mysinger, E.S. Bolstad, R.G. Coleman, J. Chem. Inf. Model. 52 (2012) 1757–1768.
[135] R. Ramakrishnan, P.O. Dral, M. Rupp, O.A. Von Lilienfeld, Sci. Data 1 (2014) 1–7.
[136] M.H. Segler, T. Kogej, C. Tyrchan, M.P. Waller, ACS Cent. Sci. 4 (2018) 120–131.
[137] D. Rogers, M. Hahn, J. Chem. Inf. Model. 50 (2010) 742–754.
[138] S. Riniker, G.A. Landrum, J. Cheminform. 5 (2013) 26.
[139] J.G. Cumming, A.M. Davis, S. Muresan, M. Haeberlein, H. Chen, Nature Rev. Drug Discov. 12 (2013) 948–962.
[140] M. Olivecrona, T. Blaschke, O. Engkvist, H. Chen, J. Cheminform. 9 (2017) 48.
[141] J. Sun, N. Jeliazkova, V. Chupakhin, J.-F. Golib-Dzib, O. Engkvist, L. Carlsson, J. Wegner, H. Ceulemans, I. Georgiev, V. Jeliazkov, et al., J. Cheminform. 9 (2017) 17.
[142] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, H. Chen, Mol. Inform. 37 (2018) 1700123.
[143] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, A. Zhavoronkov, Oncotarget 8 (2017) 10883.
[144] N.M. O'Boyle, M. Banck, C.A. James, C. Morley, T. Vandermeersch, G.R. Hutchison, J. Cheminform. 3 (2011) 33.
[145] Y. Wang, T. Suzek, J. Zhang, J. Wang, S. He, T. Cheng, B.A. Shoemaker, A. Gindulyte, S.H. Bryant, Nucl. Acids Res. 42 (2014) D1075–D1082.
[146] O. Méndez-Lucio, B. Baillif, D.-A. Clevert, D. Rouquié, J. Wichard, Nature Commun. 11 (2020) 1–10.
[147] A. Subramanian, R. Narayan, S.M. Corsello, D.D. Peck, T.E. Natoli, X. Lu, J. Gould, J.F. Davis, A.A. Tubelli, J.K. Asiedu, et al., Cell 171 (2017) 1437–1452.
[148] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, P. Battaglia, 2018, arXiv preprint arXiv:1803.03324.
[149] J. You, R. Ying, X. Ren, W.L. Hamilton, J. Leskovec, 2018, arXiv preprint arXiv:1802.08773.
[150] Y. Li, L. Zhang, Z. Liu, J. Cheminform. 10 (2018) 33.
[151] Q. Liu, M. Allamanis, M. Brockschmidt, A. Gaunt, Advances in Neural Information Processing Systems, pp. 7795–7804.
[152] F. Imrie, A.R. Bradley, M. van der Schaar, C.M. Deane, J. Chem. Inf. Model. 60 (2020) 1983–1995.
[153] M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li, R. Wang, J. Chem. Inf. Model. 59 (2018) 895–913.
[154] H.-J. Böhm, J. Comput. Aided Mol. Des. 6 (1992) 61–78.
[155] H.-J. Böhm, J. Comput. Aided Mol. Des. 6 (1992) 593–606.
[156] P. Maass, T. Schulz-Gasch, M. Stahl, M. Rarey, J. Chem. Inf. Model. 47 (2007) 390–399.
[157] D.C. Thompson, R.A. Denny, R. Nilakantan, C. Humblet, D. Joseph-McCarth, E. Feyfant, J. Comput. Aided Mol. Des. 22 (2008) 761.
[158] F. Dey, A. Caflisch, J. Chem. Inf. Model. 48 (2008) 679–690.
[159] M.J. Vainio, T. Kogej, F. Raubacher, J. Sadowski, Scaffold hopping by fragment replacement, 2013.
[160] A. Trapero, A. Pacitto, V. Singh, M. Sabbah, A.G. Coyne, V. Mizrahi, T.L. Blundell, D.B. Ascher, C. Abell, J. Med. Chem. 61 (2018) 2806–2822.
[161] T. Kamenecka, J. Habel, D. Duckett, W. Chen, Y.Y. Ling, B. Frackowiak, R. Jiang, Y. Shin, X. Song, P. LoGrasso, J. Biol. Chem. 284 (2009) 12853–12861.
[162] W. Farnaby, M. Koegl, M.J. Roy, C. Whitworth, E. Diers, N. Trainor, D. Zollman, S. Steurer, J. Karolyi-Oezguer, C. Riedmueller, et al., Nature Chem. Biol. 15 (2019) 672–680.
[163] V.G. Maltarollo, J.C. Gertrudes, P.R. Oliveira, K.M. Honorio, Expert Opin. Drug Metabol. Toxicol. 11 (2015) 259–271.
[164] L. Tao, P. Zhang, C. Qin, S. Chen, C. Zhang, Z. Chen, F. Zhu, S. Yang, Y. Wei, Y. Chen, Adv. Drug Deliv. Rev. 86 (2015) 83–100.
[165] M. Lapins, S. Arvidsson, S. Lampa, A. Berg, W. Schaal, J. Alvarsson, O. Spjuth, J. Cheminform. 10 (2018) 17.
[166] ACD/Labs, www.acdlabs.com. (Accessed September 2020).
[167] J.-L. Faulon, D.P. Visco, R.S. Pophale, J. Chem. Inf. Comput. Sci. 43 (2003) 707–720.
[168] Y.W.I. Low, F. Blasco, P. Vachaspati, Eur. J. Pharm Sci. 92 (2016) 110–116.
[169] Y.W. Alelyunas, L. Pelosi-Kilby, P. Turcotte, M.-B. Kary, R.C. Spreen, J. Chromatogr. A 1217 (2010) 1950–1955.
[170] P. Schyman, R. Liu, V. Desai, A. Wallqvist, Front. Pharmacol. 8 (2017) 889.
[171] M. Muehlbacher, G.M. Spitzer, K.R. Liedl, J. Kornhuber, J. Comput. Aided Mol. Des. 25 (2011) 1095–1106.
[172] R. Naef, Molecules 20 (2015) 18279–18351.
[173] M.S. Attene-Ramos, R. Huang, S. Michael, K.L. Witt, A. Richard, R.R. Tice, A. Simeonov, C.P. Austin, M. Xia, Environ. Health Perspect. 123 (2015) 49–56.
[174] A.P. Bento, A. Gaulton, A. Hersey, L.J. Bellis, J. Chambers, M. Davies, F.A. Krüger, Y. Light, L. Mak, S. McGlinchey, et al., Nucl. Acids Res. 42 (2014) D1083–D1090.
[175] Y. Xu, Z. Dai, F. Chen, S. Gao, J. Pei, L. Lai, J. Chem. Inf. Model. 55 (2015) 2085–2093.
[176] S. Wang, Y. Li, J. Wang, L. Chen, L. Zhang, H. Yu, T. Hou, Mol. Pharmaceut. 9 (2012) 996–1010.
[177] D. Li, L. Chen, Y. Li, S. Tian, H. Sun, T. Hou, Mol. Pharmaceut. 11 (2014) 716–726.

[178] F. Broccatelli, E. Carosati, A. Neri, M. Frosini, L. Goracci, T.I. Oprea, G. Cruciani, J. Med. Chem. 54 (2011) 1740–1751.
[179] L. Chen, Y. Li, Q. Zhao, H. Peng, T. Hou, Mol. Pharmaceut. 8 (2011) 889–900.
[180] K. Hansen, S. Mika, T. Schroeter, A. Sutter, A. Ter Laak, T. Steger-Hartmann, N. Heinrich, K.-R. Müller, J. Chem. Inf. Model. 49 (2009) 2077–2081.
[181] R. Liu, G. Tawa, A. Wallqvist, Chem. Res. Toxicol. 25 (2012) 2216–2226.
[182] J. Wenzel, H. Matter, F. Schmidt, J. Chem. Inf. Model. 59 (2019) 1253–1268.
[183] S.K. Kearsley, S. Sallamack, E.M. Fluder, J.D. Andose, R.T. Mosley, R.P. Sheridan, J. Chem. Inf. Comput. Sci. 36 (1996) 118–127.
[184] RDKit: Open-source cheminformatics, http://www.rdkit.org. (Accessed September 2020).
[185] J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl, V. Svetnik, J. Chem. Inf. Model. 55 (2015) 263–274.
[186] S. Crosignani, M. Missotten, C. Cleva, R. Dondi, Y. Ratinaud, Y. Humbert, A.B. Mandal, A. Bombrun, C. Power, A. Chollet, et al., Bioorg. Med. Chem. Lett. 20 (2010) 3614–3617.
[187] I. Bata, Z. Tömösközi, P. Buzder-Lantos, A. Vasas, G. Szeleczky, S. Bátori, V. Barta-Bodor, L. Balázs, G.G. Ferenczy, Bioorg. Med. Chem. Lett. 26 (2016) 5418–5428.
[188] H. Matter, B. Scheiper, H. Steinhagen, Z. Böcskei, V. Fleury, G. McCort, Bioorg. Med. Chem. Lett. 21 (2011) 5487–5492.
[189] B. Scheiper, H. Matter, H. Steinhagen, Z. Böcskei, V. Fleury, G. McCort, Bioorg. Med. Chem. Lett. 21 (2011) 5480–5486.
[190] B. Scheiper, H. Matter, H. Steinhagen, U. Stilz, Z. Böcskei, V. Fleury, G. McCort, Bioorg. Med. Chem. Lett. 20 (2010) 6268–6272.
[191] E.A. Blomme, Y. Will, Chem. Res. Toxicol. 29 (2016) 473–504.
[192] H. Yang, L. Sun, W. Li, G. Liu, Y. Tang, Front. Chem. 6 (2018) 30.
[193] Tox21 Data Challenge, https://tripod.nih.gov/tox21/challenge/. (Accessed September 2020).
[194] A. Chawla, J.J. Repa, R.M. Evans, D.J. Mangelsdorf, Science 294 (2001) 1866–1870.
[195] F. Grün, B. Blumberg, Rev. Endocr. Metab. Disorders 8 (2007) 161–171.
[196] J. Bartkova, Z. Hořejší, K. Koed, A. Krämer, F. Tort, K. Zieger, P. Guldberg, M. Sehested, J.M. Nesland, C. Lukas, et al., Nature 434 (2005) 864–870.
[197] G. Labbe, D. Pessayre, B. Fromenty, Fund. Clin. Pharmacol. 22 (2008) 335–353.
[198] H. Jaeschke, M.R. McGill, A. Ramachandran, Drug Metabol. Rev. 44 (2012) 88–106.
[199] A. Mayr, G. Klambauer, T. Unterthiner, S. Hochreiter, Front. Environ. Sci. 3 (2016) 80.
[200] M. Chen, V. Vijay, Q. Shi, Z. Liu, H. Fang, W. Tong, Drug Discov. Today 16 (2011) 697–703.
[201] C.Y. Liew, Y.C. Lim, C.W. Yap, J. Comput. Aided Mol. Des. 25 (2011) 855.
[202] N. Greene, L. Fisk, R.T. Naven, R.R. Note, M.L. Patel, D.J. Pelletier, Chem. Res. Toxicol. 23 (2010) 1215–1222.
[203] J.J. Xu, P.V. Henstock, M.C. Dunn, A.R. Smith, J.R. Chabot, D. de Graaf, Toxicol. Sci. 105 (2008) 97–105.
[204] H. Hong, Q. Xie, W. Ge, F. Qian, H. Fang, L. Shi, Z. Su, R. Perkins, W. Tong, J. Chem. Inf. Model. 48 (2008) 1337–1344.
[205] C.W. Yap, J. Comput. Chem. 32 (2011) 1466–1474.
[206] A. Zhavoronkov, Y.A. Ivanenkov, A. Aliper, M.S. Veselov, V.A. Aladinskiy, A.V. Aladinskaya, V.A. Terentiev, D.A. Polykovskiy, M.D. Kuznetsov, A. Asadulaev, et al., Nat. Biotechnol. 37 (2019) 1038–1040.
[207] W.P. Walters, M. Murcko, Nature Biotechnol. 38 (2020) 143–145.
[208] A. Zhavoronkov, A. Aspuru-Guzik, Nature Biotechnol. 38 (2020) 146.
[209] J. Jiménez-Luna, F. Grisoni, G. Schneider, Nature Mach. Intell. 2 (2020) 573–584.