Artificial Intelligence in the Life Sciences 3 (2023) 100055

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

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

Review

Artificial intelligence systems for the design of magic shotgun drugs

José Teófilo Moreira-Filho[a,1], Meryck Felipe Brito da Silva[a,1], Joyce Villa Verde Bastos Borba[a], Arlindo Rodrigues Galvão Filho[b], Eugene N Muratov[c,d], Carolina Horta Andrade[a], Rodolpho de Campos Braga[e], Bruno Junior Neves[a,*]

[a] LabMol – Laboratory for Molecular Modeling and Drug Design, Faculdade de Farmácia, Universidade Federal de Goiás, Goiânia, GO, Brazil
[b] Institute of Informatics, Federal University of Goiás, Goiânia, GO, Brazil
[c] Laboratory for Molecular Modeling, UNC Eshelman School of Pharmacy, University of North Carolina Chapel Hill, North Carolina, United States of America
[d] Department of Pharmaceutical Sciences, Federal University of Paraíba, Joao Pessoa, PB, Brazil
[e] InsilicAll Ltda, São Paulo, SP, Brazil

ARTICLE INFO

Keywords:
Multi-target drugs
Deep learning
Predictive modeling
De novo design
Multi-task learning

ABSTRACT

Designing magic shotgun compounds, i.e., compounds hitting multiple targets using artificial intelligence (AI) systems based on machine learning (ML) and deep learning (DL) approaches, has a huge potential to revolutionize drug discovery. Such intelligent systems enable computers to create new chemical structures and predict their multi-target properties at a low cost and in a time-efficient manner. Most examples of AI applied to drug discovery are single-target oriented and there is still a lack of concise information regarding the application of this technology for the discovery of multi-target drugs or drugs with broad-spectrum action. In this review, we focus on current developments in AI systems for the next generation of automated design of multi-target drugs. We discuss how classical ML methods, cutting-edge generative models, and multi-task deep neural networks can help *de novo* design and hit-to-lead optimization of multi-target drugs. Moreover, we present state-of-the-art workflows and highlight some studies demonstrating encouraging experimental results, which pave the way for *de novo* drug design and multi-target drug discovery.

1. Introduction

The principle "one target, one drug" has been the imperative strategy in drug discovery flowcharts for many decades. This paradigm is based on the premise that selective drugs targeting a single biological target (e.g., enzyme or receptor) may avoid side effects arising from binding to other biological targets ("off-targets" effects) [1]. Although numerous single-target drugs have been clinically effective, experience has shown that excessive selectivity may sometimes have fatal consequences [2,3]. In addition, these molecules have limited efficacy against complex diseases (e.g., diabetes, cancer, neurodegenerative diseases, metabolic syndrome, and atherosclerosis) [4,5] in which the pathogenesis is depen-

Abbreviations: 5-HT_{1A}, serotonin subtype 1A receptor; 5-HT_{2A}, serotonin subtype 2A receptor; AI, artificial intelligence; A_1AR, A_1 Adenosine Receptor; A_{2A}AR, A_{2A} Adenosine Receptor; ANN, artificial neural networks; BRAF, serine/threonine kinase protein B-raf; CADD, computer-assisted drug design; Cmax, maximum serum concentration; CNN, convolutional neural networks; D_1, Dopamine receptor D_1; D_2, Dopamine receptor D_2; DT, decision trees; ELU, Exponential Linear Unit; ECFPs, extended-connectivity fingerprints; ECFP4, extended-connectivity fingerprints with diameter 4; FCFPs, functional-class fingerprints; FDA, U.S. Food and Drug Administration; FNN, feed-forward neural networks; GPU, graphical processing units; GRU, gated recurrent unit; hERG, Ether-à-go-go-Related Gene; HTS, high throughput screening; IC_{50}, half maximal inhibitory concentration; LSTM, long-short-term memory; MACCS, Molecular ACCess System; ML, machine learning; MOO, multi-objective optimization; MTL-DNN, multi-task learning deep neural network; PD, promiscuity difference; PCM, proteochemometric modeling; PDGFR, platelet-derived growth factor receptor; pEC_{50}, minus of half maximal effective concentration; PK, pharmacokinetics; pKi, minus of binding affinity; MOSES, Molecular Sets; NSGA-II, non-dominated sorting genetic algorithm; QSAR, quantitative structure-activity relationships; ReLU, Rectified Linear Unit; RF, random forest; RNN, recurrent neural networks; SELFIES, SELF-referencIng Embedded Strings; SMILES, simplified molecular input line entry specification; SVM, support vector machines; $T_{1/2}$, Half-life time; T_{max}, Time to peak drug concentration; TPU, tensor processing units; VEGFR, vascular endothelial growth factor receptor.

* Corresponding author: Dr. Bruno Junior Neves, Federal University of Goias, Brazil.
E-mail address: brunoneves@ufg.br (B.J. Neves).

[1] These authors contributed equally to this study and should be considered as co-first authors.

https://doi.org/10.1016/j.ailsci.2022.100055
Received 19 September 2022; Received in revised form 13 December 2022; Accepted 21 December 2022
Available online 22 December 2022

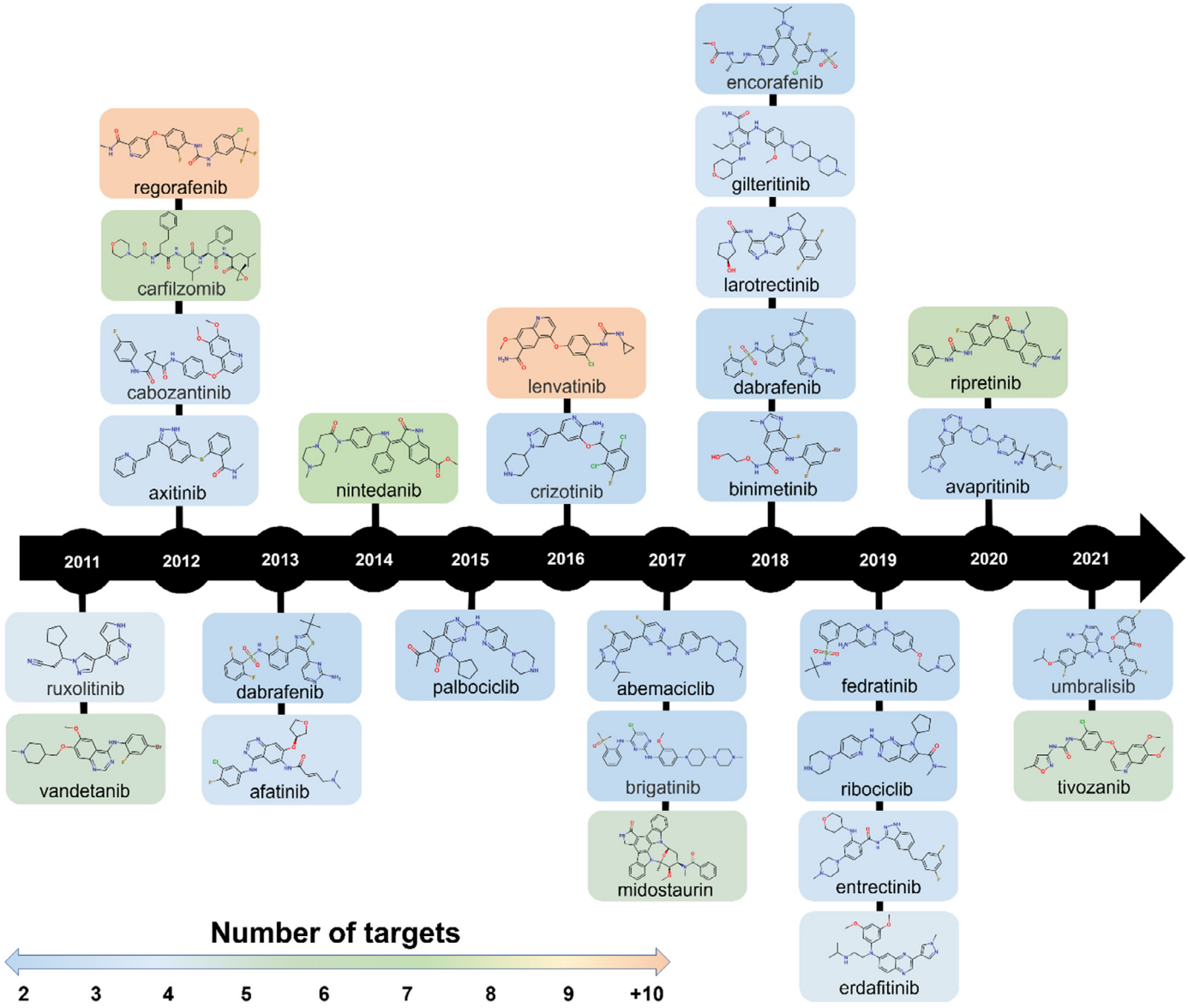

Fig. 1. Examples of multi-target drugs approved by FDA in the last decade.

dent on a set of biological targets operating concomitantly [6]. Consequently, a poor correlation between in vitro selectivity and in vivo efficacy is often found for complex diseases [7,8].

Under such a perspective, multi-target or magic shotgun drugs emerged as efficient therapeutic solutions to overcome drug-resistance issues and increase efficacy against complex diseases [9]. Multi-target drugs contrast the classical drug design paradigm by modulating multiple relevant targets for the disease [10]. Consequently, multi-target drug design has triggered the interest of medicinal chemists, abandoning its status as an emerging paradigm [11,12] to become one of the most effective approaches for drug discovery [7–9].

The most successful multi-target drugs to date have been kinase inhibitors, due to the important role kinases play in critical biological pathways such as cancer development and progression. Over the past decade, several multi-kinase inhibitors emerged from the drug discovery pipelines into clinical use [13]. As shown in Fig. 1, twenty-eight anticancer drugs targeting the human kinome were approved by the U.S. Food and Drug Administration (FDA) between 2011 and 2021. These approved drugs cope with the multifactorial nature of cancer by simultaneously targeting multiple tyrosine kinases. For example, the therapeutic efficacy of regorafenib (Stivarga®; Bayer HealthCare Pharmaceuticals, Inc.) is partly due to its ability to inhibit 10 kinases involved in the regulation of tumor angiogenesis [angiopoietin-1 receptor and vascular endothelial growth factor receptors (VEGFRs) 1, 2, and 3], maintenance of the tumor microenvironment [fibroblast growth factor receptor and platelet-derived growth factor receptor (PDGFR)-β], and oncogenesis [serine/threonine kinase protein B-raf (BRAF), Raf proto-oncogene serine/threonine-protein kinase, stem cell growth factor receptor, and proto-oncogene tyrosine-protein kinase receptor Ret] [14–16].

Despite the obvious advantages, designing new chemical entities with a balanced multi-target profile has been historically difficult. The activity of compounds needs to be simultaneously optimized toward several targets from different protein families with dissimilar binding pockets [17]. Hence, there is an urgent need for innovative *in silico* approaches to make the multi-target drug design more efficient [18,19].

The past decades have brought striking innovations in different fields of drug discovery and related areas such as high throughput screening (HTS), combinatorial chemistry, robotics, genomics, transcriptomics, metabolomics, and chemogenomics [18,20]. These cutting-edge technologies have generated massive amounts of bioassay data, moving the

drug discovery field into the "big data" era [21]. Consequently, massive amounts of data can be found in public domain databases of different categories, including: bioactivity databases (PubChem [22] and ChEMBL [23]); databases for three-dimensional structures (RCSB Protein Data Bank (PDB) [24]); protein-protein interaction databases (STITCH [25] and STRING [25]); pathway/genome databases (BRENDA [26] and BioCyc [27]); chemical toxicology databases (CompTox [28] and EFSA's OpenFoodTox database [29]); and clinical databases (ClinicalTrials.gov [30] and PharmaGKB [31]). The full exploitation of these rich sources of data is of high interest to researchers in the field of cheminformatics and computer-assisted drug design (CADD).

Many *in silico* methods have been developed to approach data-driven multitarget drug design. These methods are usually categorized as either ligand-based (e.g., pharmacophore modeling) or structure-based (e.g., molecular docking), depending on whether ligand and structural information are available [32]. However, the potential for exploring these tools remains untapped since most *in silico* examples of drug discovery are single-target oriented. Thus, there is a need for innovative technologies to handle complex data and transform them into useful information, offering a means towards data-driven decisions and the consequent development of multi-target drugs with broad-spectrum action [33–35].

In recent years, artificial intelligence (AI) approaches based on machine learning (ML) have emerged as promising alternatives that improve decision-making in the context of multi-target drug design [17,36,37]. ML is a subfield of AI that uses different algorithms to enable computers to make efficient predictions from past observations without being continually programmed to this task [38–41]. These methods are able to handle multi-objective optimization (MOO) problems, such as the discovery of multi-target compounds, navigating huge datasets, and allowing better decisions [34]. In this review, we focus on recent developments in AI technologies for the automated design of multi-target drugs. We also highlight literature examples implementing innovative workflows and suggest possible solutions to existing pitfalls.

2. Machine learning methods

More than 50 years ago, classical quantitative structure-activity relationship (QSAR) modeling started using linear models such as linear regression and *k*-nearest neighbors to make predictions of chemical properties [42–44]. Beginning in the early 2000s, QSAR approaches progressed to more sophisticated and non-linear ML methods, especially Decision trees (DTs), random forest (RF), and support vector machines (SVMs) [45,46]. DTs consists of a group of rules that enable establish between molecular features (represented as molecular descriptors) and the activity of the compounds [47,48].

DTs are tree-shaped, presenting a root node at the top, where the data is divided into branches. These branches will continue to divide into gradually smaller subsets until an outcome (represented as a leaf node) is reached [17,49]. DTs are simple to interpret and validate and flexible enough to deal with numerical and categorical inputs and multiclass problems [17,50,51]. The disadvantages of DTs include the loss of performance in noisy or incomplete data, and the sensitivity to high variance data (i.e., small changes can result in different data splits and propagation of errors) [49,52]. An approach to alleviating the bias and variance problem is the use of an ensemble of DTs [53]. In this sense, one of the most widely used and best-performing ensemble algorithms is RF (Fig. 2a) [54,55]. RF comprises an ensemble of individual DTs constructed using different subsets obtained from the training data with replacement [56]. Thereafter, predictions for unseen data are obtained via majority voting (classification) or predictions (regression) of the individual trees.

Another non-linear ML method is the SVM algorithm developed by Vapnik and coworkers [57,58]. SVM is a supervised ML algorithm able to perform regression and classification tasks for compound property/activity predictions [48]. As shown in Fig. 2b, SVM transforms the original molecular descriptors space into a high-dimensional feature

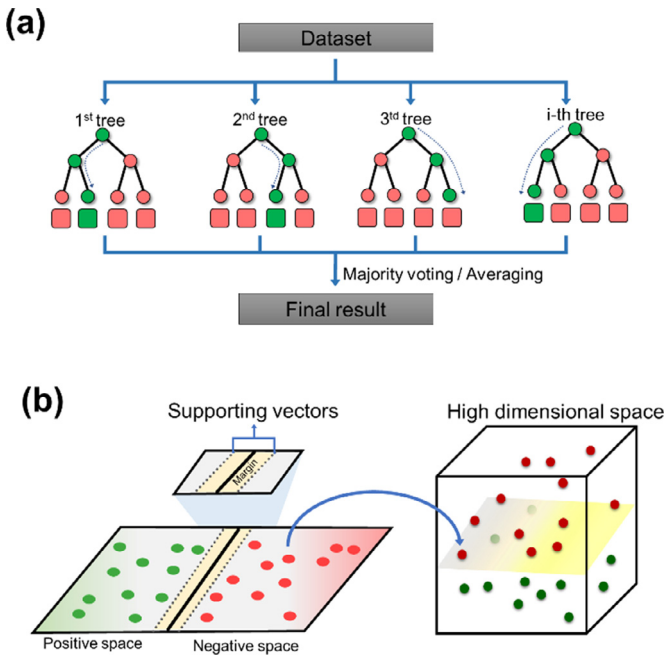

Fig. 2. Classification process based on the **(a)** Random Forest and **(b)** Support Vector Machines.

space, where samples can be linearly separable [49]. This mapping is achieved through the use of a kernel function, (e.g., polynomial, linear, sigmoid, or radial basis) [59]. Then, a hyperplane that best separates the different classes of compounds (e.g., active/inactive) is generated via the definition of the larger margins (support hyperplanes) traced using the nearest data points of each class (support vectors) [17,45,49]. Thus, unseen compounds are also mapped into this high-dimensional feature space and classified according to the side of the boundary where they lie [45]. SVM is one of the best-performing algorithms for the prediction of chemical and biological properties thanks to its ability to deal with complex, non-linear, high-dimensional, and noisy problems [45,48,54,60]. However, depending on the size of the dataset, feature space, and kernel, SVMs can be computationally expensive [49].

Blaschke and coworkers [61] developed SVM, RF, Feedforward Deep Neural Networks (FNNs, see section 5), and Graph Convolution Neural Networks (GCNNs) to identify promiscuous and non-promiscuous compounds based on screening assays and kinase inhibitors data retrieved from several data sources such as PubChem. Then, the compounds were divided into classes using a promiscuity difference (PD) criterion, where compounds active in one target or compounds consistently inactive (screening molecules: PD = 1 and PD = 0, respectively) were considered non-promiscuous while compounds with activity against 10 or more targets (PD ≥ 10) were considered promiscuous. For the SMV, RF, and DNN models, compounds were represented using the extended connectivity fingerprints with a diameter of 4 interactions (ECFP4) and represented as chemical graphs for GCNNs. In general, all models were predictive, with an overall accuracy around 70%, and differences between the ML approaches were minimal. In another study, Heikamp and Bajorath [62] investigated a multi-label prediction model (i.e., models able to address more than two classes) with compounds presenting activity against distinct combinations of targets using SVM models. First, compound datasets with single-, dual-, and triple-target activity together with inactive compounds were retrieved from PubChem database [63] for three cytochrome P450 isoforms and three different dehydrogenases. All Compounds were represented using the ECFP4; however, the generated standard SVM models were not able to distinguish compounds with overlapping yet distinct activity profiles. Then, the authors developed an SVM linear combination approach through the weighting

of different models using positive and negative factors. Consequently, refined models efficiently distinguished between compounds with overlapping activity profiles.

In recent decades, proteochemometric modeling (PCM) has emerged as a suitable technique for discovering multi-target drugs [64,65]. PCM is a computational chemogenomic approach that draws heavily on recent ML developments. The PCM models can be trained on datasets composed of a series of compounds and protein targets or binding pockets (i.e., to allow distinction between allosteric/orthosteric binding and binding modes), where ideally, compounds have been measured on as many targets as possible [64]. The simultaneous modeling of the ligand descriptor (e.g., Hashed fingerprints, circular fingerprints, MACCS keys, atom counts) and target descriptor (e.g., Atom-type based, Z-scales, sequence descriptors) spaces and cross-term descriptors (i.e., an additional class of descriptors that multiplies ligand and target descriptors) allows the researcher to understand better complex drug–target interactions (e.g., multi-target profile) as the influence of target variability on compound activity can be evaluated [64].

3. The rise of deep learning

The term “Deep Learning (DL)” refers to a growing field of ML that uses Artificial Neural Networks (ANNs) composed of multiple hidden layers, also called Deep Neural Networks (DNNs), for learning data representations. The structure of these DNNs mimics the operations of neurons found in the cerebral cortex to learn from feature representations of data with multiple levels of abstraction [66–68]. Much of the extensive popularization of DL is related to cloud computing [69] and the availability of new hardware technologies, such as graphical processing units (GPUs) [70] and tensor processing units (TPUs) [71], that allowed researchers to train networks up to 35 times faster. Importantly, the main advantage of DL is the ability to learn complex features from massive volumes of data. Fig. 3a shows the comparison between DL and ML performance as the training data size increases. For smaller datasets, ML algorithms usually show better performance as compared to DL [72,73]. On the other hand, when the amount of data increases, the performance of ML methods reaches a saturation point and does not improve any further, whereas the performance of DL keeps increasing [72]. For instance, recently, Mayr et al. [74] showed that DNN-based architectures trained using a large-scale drug discovery dataset significantly outperformed all competing classical ML methods (KNN, SVM, Random Forest, and Naive Bayes). However, the superior performance of DL over ML models has not been consistently observed in applications using datasets of different origins and compositions [75,76]. Kato et al. [76] suggest that the performance of DL and ML models may be impacted by the insufficient level of molecular feature representations to predict complex biological mechanisms observed in molecular activity measurements [76]. In addition, alternative data splitting strategies and the resulting test systems (hyperparameters) can influence the relative performance of models, especially for the most challenging compound datasets [75,77].

The fully connected feedforward neural network (FNN) was the first and simplest type of DNN devised. A typical FNN contains many artificial neurons arranged in three types of layers (Fig. 3b) [78]:

- Input layer: accepts the input data and passes it to the first hidden layer;
- Hidden layers: located between the input and output of the algorithm. The hidden layers perform non-linear transformations on the input data that will produce a predicted output that is close enough to the expected output;
- Output layer: is mostly responsible for production output predictions.

The network is composed of an interconnected system of perceptrons that act as basic information-processing units. Fig. 3c displays a perceptron in more detail. The perceptron algorithm consists of four parts: (i) input values, (ii) weights and a bias, (iii) a weighted sum, and (iv) an activation function. The inputs can either come from the input layer or perceptrons in a previous hidden layer [79,80].

The perceptron’s operation consists of a function that multiplies input values by corresponding weights and adds a bias. The bias term is an adjustable term added to a perceptron’s sum of inputs and weights, and it serves as another model parameter [79]. The weights are initialized with random values at the beginning of the input layer. Then, all of these multiplied values are added together to create the weighted sum. The weighted sum is then applied to an activation function that passes through the activation function, which can add a non-linear component or assign linearization to the network. Finally, the output from the activation function moves to the next layer. This forward movement is known as “forward propagation” [79]. Mathematically, the output value (Y_i) is a non-linear weighted sum of input signals of a neuron i. The Y_i can be calculated as shown in Eq. (1):

$$Y_i = g\left(\sum_j W_{ij} * a_j\right) \tag{1}$$

where a_j refers to the input features, W_{ij} is the weight of input neuron j on neuron i, and g is the activation function.

The first perceptron model was created for simple classification, considering a linear classifier capable of predicting binary outputs. However, most computational problems need a classifier able to model non-linear, separable data. Consequently, several non-linear activation functions emerged to identify more complex features in the data. Well-known activation functions (Fig. 3d) in the field of cheminformatics include the Sigmoid, Hyperbolic Tangent (TanH), Rectified Linear Unit (ReLU), Leaky Rectified Linear Unit (Leaky ReLU), and Exponential Linear Unit (ELU). The correct choice of an activation function should consider the type of input feature, as well as the architecture and complexity of the DNN. For instance, the Sigmoid and TanH activation functions cannot be used in DNNs with many layers due to the vanishing gradient problem (i.e., unstable behavior encountered during the training of a DNN). The gradient-based learning methods by backpropagation receives an update weight from each of the neural network and then the error function with respect to the current weight. The problem is that in some cases, the gradient is vanishingly small, effectively preventing the weight from changing value. On the other hand, the weight can receive updated values that culminate in a convergence of very high values, thus causing exploding gradient problem.

At present, ReLU is the most popular activation function, since it typically overcomes the vanishing gradient problem [81]. ReLU outputs the input directly if it is positive; otherwise, it will output zero. ReLU has become the gold standard activation function for many types of DNNs because models that use it are easier to train and often achieves better performance [82].

For any type of problem, the DNN architecture needs to be specified before training. Untrained DNNs are created as "ignorant" systems, and it is only through exposure to the data, that their ignorance slowly decreases. The main way to measure such learning is by monitoring the error produced by the network each time it makes a prediction. Mechanistically, the signal of the input data moves forward through the prespecified network towards the output layer and then backpropagates information about the error measured with a loss function, at which point the neuron connection weights are updated [83]. The propagation and backpropagation (Fig. 3e) are repeated several times, according to the specified number of epochs, a hyperparameter that defines the number of times (i.e., interactions) the error will backpropagate through the network [84]. An epoch is made up of one or more batches, where a part of the dataset is used to train the neural network. Therefore, to train a DNN is to maximize an objective function by optimizing the weights and biases of each neuron [67,83].

Another characteristic of DNNs is that they have flexible architectures. The most popular architectures are FNNs [85], Recurrent Neural Networks (RNNs) [85], and Graph Neural Networks (GNNs) [86,87].

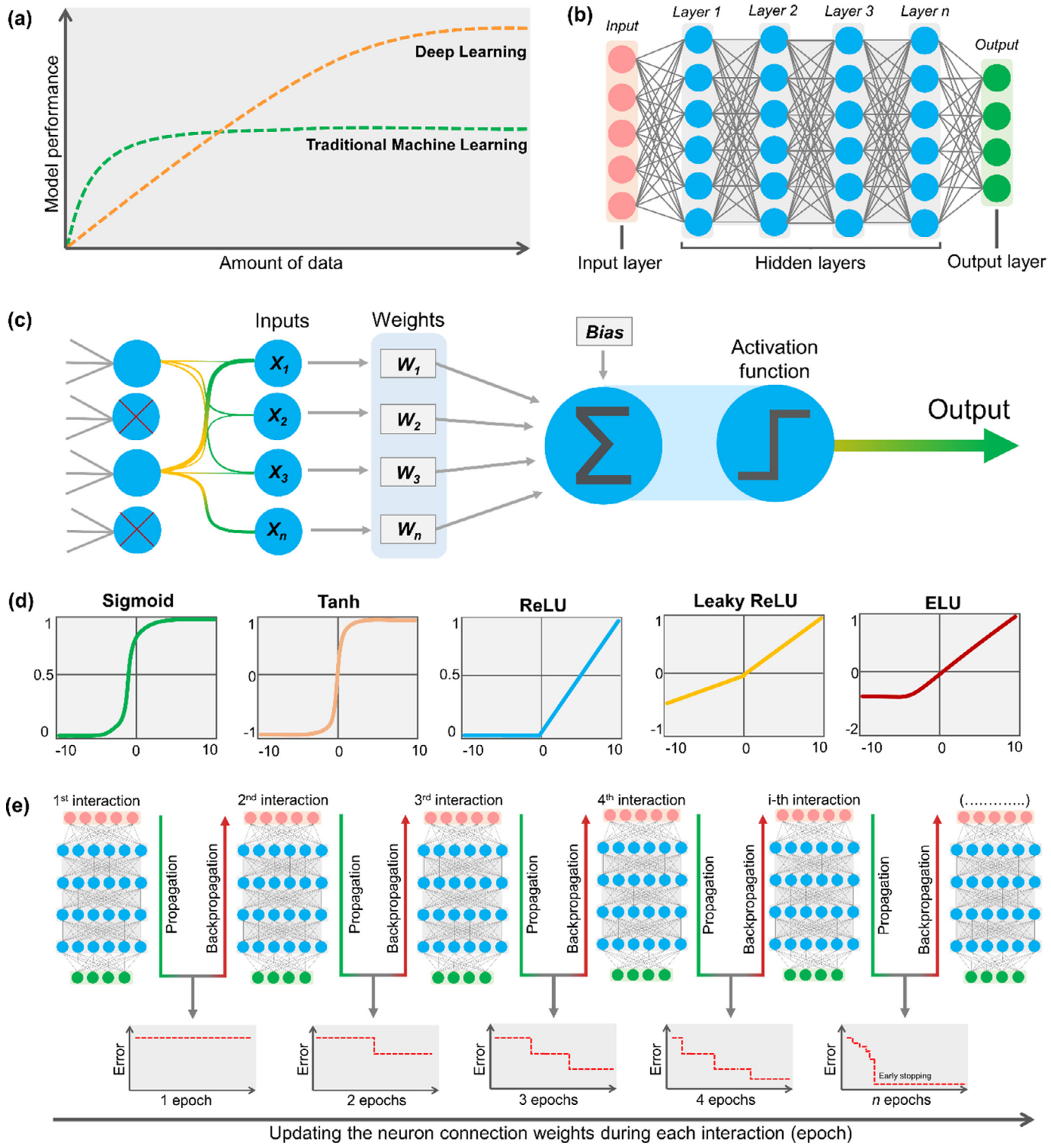

Fig. 3. Learning process based on DNNs. **(a)** Performance of ML vs. DL models with respect to the amount of data; **(b)** Architecture of a typical Feedforward Neural Network (FNN); **(c)** Perceptron model; **(d)** Frequently used activation function; **(e)** Forward propagation and backpropagation of error in typical DNNs.

Each architecture has advantages and disadvantages, which are related to the type of feature being modeled.

The fully connected FNN (Fig. 3b) is the most common architecture of DNNs. In this network, the information moves in a single direction, from the input layer to the output layer, without loops or backward connections [85]. Already, the RNNs were designed to identify patterns in sequential data, such as Simplified Molecular Input Line Entry Specification (SMILES) strings. Although they are still the most used representations in RNNs, SMILES have no mechanisms to ensure that strings are valid with respect to physical principles and syntax. In view of this, SELF-referencing Embedded Strings (SELFIES) have been suggested as a chemically more intuitive alternative, as every combination of symbols in its alphabet maps to a chemically valid graph [88]. The RNN architecture (Fig. 4a) has three types of layers: the input layer X_0, the hidden

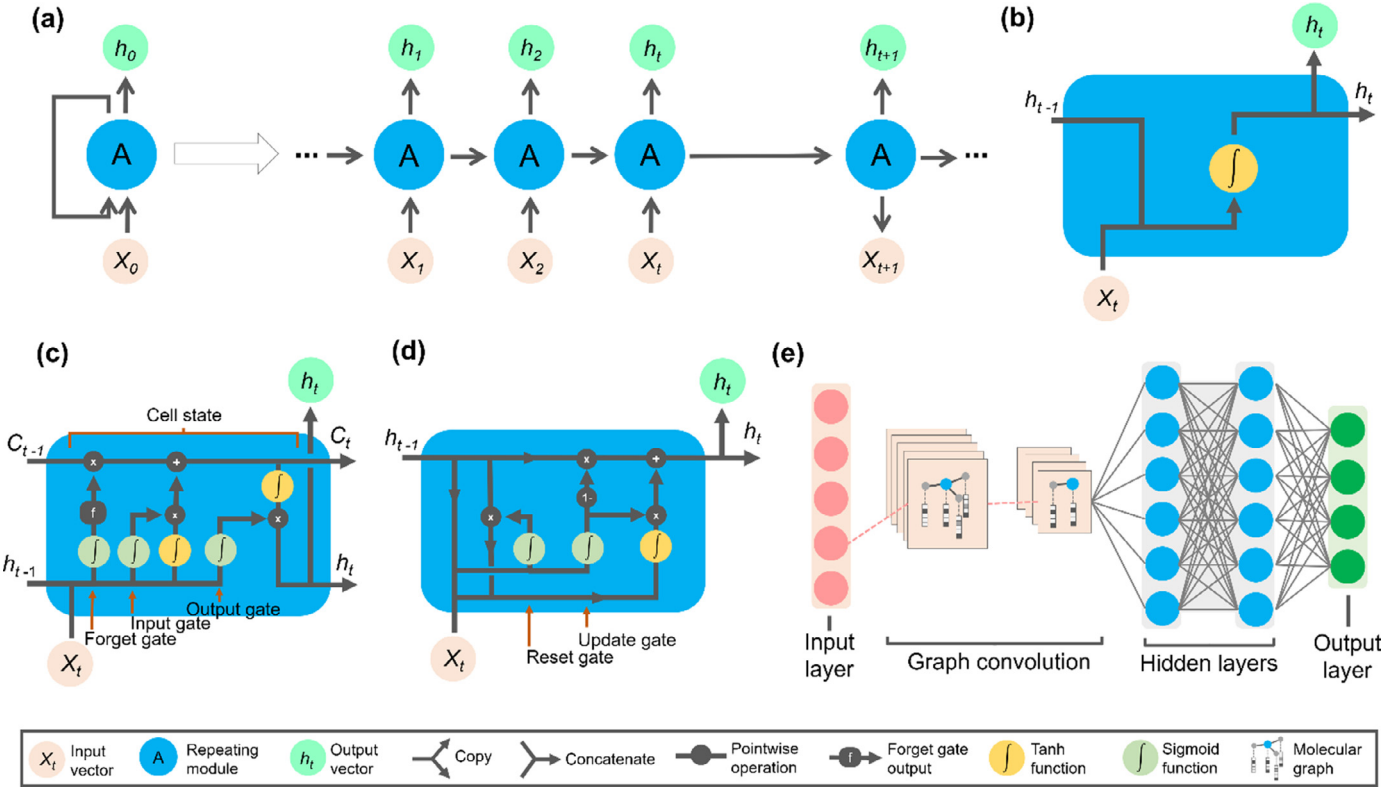

Fig. 4. Deep learning zoo. **(a)** The unfolded Recurrent Neural Network (RNN). Structure diagram of the **(b)** default RNN, **(c)** Long-Short-Term Memory (LSTM), **(d)** Gated Recurrent Unit (GRU), and **(e)** Graph Convolutional Neural Network (GCNN) architectures.

layer A (recurrent state), and the output layer h_0. If the hidden layer loop is unfolded, the default RNNs copy the same structure multiple times, and the "state vector" A (memory) of each copy is taken as input to its successor [85]. Thus, RNNs are a memory-gated architecture: they allow the evaluation of new inputs through persistent information from previous states, meaning that the same input could produce a different output depending on previous inputs in the series [102]. However, a default RNN (Fig. 4b) is unable to capture temporal dependencies that extend to more than a limited number of time steps (long-term dependencies). Therefore, Long-Short-Term Memory (LSTM) [89] and Gated Recurrent Unit (GRU) [90] networks specifically address this limitation.

Compared to the default RNN (Fig. 4b), the cell state of LSTM (Fig. 4c) architecture, introduced by Hochreiter and Schmidhuber [89], offers a solution for vanishing/exploding gradients by preserving and learning long-term dependencies in data. As shown in Fig. 4c, LSTM uses three gates to control a cell state (C_t): (i) forget gate, (ii) input gate, and (iii) output gate. The forget gate decides which information needs attention and which can be ignored. The information from the current input (X_t) and previous hidden state (h_{t-1}) is passed through a Sigmoid function. This function generates values between 0 and 1, which are multiplied by the previous cell state (C_{t-1}) [90,91]. The input gate then decides what new information can be stored in the cell state. This is usually controlled by a Sigmoid function that decides which values will be updated and a TanH operator that creates a vector of new candidate values (between −1 and 1). The output values generated from these activation functions are point-by-point multiplied and then added to the cell state. Finally, an output gate determines the value of the next hidden state (h_t). In this step, the values of current and previous hidden states are passed into Sigmoid and TanH functions. Both these outputs are multiplied point-by-point. Ultimately, based upon the final values, the network decides which information the h_t should carry [90,91].

The GRU [90] (Fig. 4d) was proposed as variant of LSTM to support gating and a hidden state to control the flow of information. When compared to LSTM network, GRU uses only two gates: (i) update gate and (ii) reset gate. The weights from update and reset gate can be calculated as shown in Eq. (2):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z \tag{2}$$

where X_t refers to the input vector, W_z is the weight matrix, the (σ) is the sigmoid function, while the b_z e z_t are connection bias and update gate, respectively. The time step in h_{t-1} signifies that it holds the information of the previous unit and it's multiplied by its weight. The summed values are passed through the sigmoid function (σ) with connection bias at timestep (b_z). The update gate (z_t) is used to control how much of the new state is just a copy of the old state, i.e., the amount of information from h_{t-1} that needs to be passed along to the next state h_t. In contrast, the reset gate controls how much of the h_{t-1} needs to be neglected [92].

The GNNs can be categorized into four groups: (i) graph autoencoders, (ii) spatial-temporal graph neural networks, graph convolutional neural networks (GCNNs), and recurrent graph neural networks (i.e., GAT: graph attention network [93]; MPNN: message passing neural network [94]). The GCNNs (Fig. 4e) are a very powerful type of Convolutional Neural Networks (CNNs) designed to work directly on graphs [86,87]. Mechanistically, convolution is the most fundamental operation in GCNNs. However, the standard convolution operation generally used for processing regular grid-like structures (e.g., images) in CNNs cannot be directly applied to graphs, since the number of node connections may acquire different shapes and sizes (non-Euclidean vector space) [95–97]. To solve this problem, nodes from different graphs are assigned into similar positions if their feature representations are related. Given the selected node sequence, the neighborhood is then assembled for each node by minimizing the expected distance between two graphs in a fixed-size vector space [95,98]. The convolution operation can be run at different levels by considering the contribution of neighboring atoms. The vectors generated from convolutions first go through a non-linear transformation and are then summed to form a

neural fingerprint encoding chemical information. Finally, the neural fingerprints are passed through another fully connected neural network to generate the final output task [68].

Among GCNN representations, geometric DL approaches have been employed for predicting the binding conformations of ligands to protein targets. Concretely, the model learns based on distance likelihood which is tailor-made for each ligand-target pair. In the first step, the neural network extracts a pool of mesh (i.e., collection of nodes, edges, and faces) which defines the shapes of the molecular surfaces of binding sites. In a similar way, ligands are represented as a two-dimensional undirected graph, where atoms and bonds are represented by nodes and edges, respectively. Both the target mesh and the ligand graph are processed by independent residual GCNNs. In the following step, the processed features from the protein targets and ligands are concatenated using a mixture density network to model the interaction of the ligands with the targets [99].

3.1. Multi-task learning

To tackle the challenges of multi-target drug design, numerous efforts are being made in the DL field to develop multi-task models. Multi-task learning (MTL) is an inductive transfer approach in which multiple tasks (e.g., targets) are simultaneously learned by a shared model. Such an approach is particularly interesting since MTL-DNNs can explore representations learned across different tasks and boost the performance of tasks with fewer training examples [100,101]. Previous studies suggest that MTL can offer a strong improvement in predictive performance over single-task methods [102]. According to Rodríguez Pérez and Bajorath [75], MTL models provided incremental advantages for predicting the most challenging biological properties. When MTL models were evaluated on tasks that share chemical information, the obtained results were generally superior to single-task learning (STL) models. In cases where STL model accuracy was limited, the largest relative performance gains of MTL models were observed [75].

There are several approaches for optimizing and calibrating MTL models. Existing works often focus on developing aggregated loss functions adapted to sparse data as a way to learn multiple tasks at once. Different loss weighing mechanisms may be used to aid MTL optimization, such as adding weights to the individual loss functions to prevent a task with more data from dominating the optimization [103].

Most existing MTL models are based on sharing parameters across hidden layers. The parameter-sharing mechanisms in MTL are divided into four categories: (i) hard parameter sharing [104], (ii) soft parameter sharing [104], (iii) hierarchical sharing [105,106], and (iv) sparse sharing [107,108]. In hard parameter sharing, the hidden layers of the neural network are shared while keeping some task-specific output layers. Sharing most of the layers for the related tasks reduces the chances of overfitting, but when tasks conflict, the learning is easily affected by negative transferring, which is harmful to the model performance [104]. The soft parameter sharing mechanism has independent parameters and subnet for each task, but their information can be mutually learned. The approach regularizes the distance between the parameters of the individual subnets to the overall training objective to encourage similar model parameters between the different tasks [104]. The hierarchical sharing considers the progressive relationship between tasks, which means that a task may be a subtask of another one, thus placing them on different network layers [104–106]. On the other hand, the hierarchical sharing mechanism may be a complex problem that relies on prior knowledge of task relations [104,109]. The sparse sharing mechanism automatically extracts subnets for each task based on the idea of the lottery ticket hypothesis. The obtained subnets are overlapped and trained in parallel [107,108].

The MTL modeling is potentially useful for designing multi-kinase inhibitors, given that a considerable number of kinases remains poorly studied and the labeled compounds for such targets are scarce. Conventionally, this limits the opportunities for constructing high-quality prediction models and advancing the understanding of these drug targets. To overcome this issue, a MTL [110,111] was developed to predict the kinome-wide activity profiles of untested compounds. Initially, the network was trained with over 170,000 bioactivity data points (~32,000 chemical structures and of 391 kinases) using ECFP4 fingerprints as inputs [110,111]. Then, the trained MTDNN was implemented to the public in the KinomeX web app (https://kinome.dddc.ac.cn) [110].

This app has an intuitive user interface, in which the user may draw a compound of interest or directly submit the SMILES string of a queried chemical structure to prediction. Given the verified generalizability of KinomeX, Li and coworkers [111] then performed a comprehensive *in silico* analysis of diverse compounds with unknown kinase activity profiles (Fig. 5a). Importantly, the experimental validation of the predicted spectrum shows significant agreement with the experimental data, suggesting that KinomeX can be used to discover multi-target kinase inhibitors [111].

3.2. De novo design

Computational *de novo* design is an efficient approach to create new molecular structures with desired multi-target profiles at a very low cost and in a time-efficient manner. The increase in the use of DL for *de novo* design has motivated the generation of platforms such as Molecular Sets (MOSES) [112], REINVENT v.2.0 [113], and GuacaMol [114] to evaluate and compare generative approaches in a standardized way. The approach aims to obtain compounds with valid structures, similar to known compounds in terms of chemical and biological properties, and structural diversity of scaffolds and fragments.

Inspired by this opportunity, deep generative modeling techniques have been used as emergent *de novo* design approaches in the early stages of drug discovery, yielding the autonomous ability to which automatically design new multi-target drug candidates [115]. The development of generative models typically involves a two-step process. Initially, a generative network is trained to generate syntactically plausible structures using a benchmarking dataset of representative structures. Then, the model is fine-tuned to generate only compounds with the desired properties. Fine-tuning of the generative model is either carried out using transfer or reinforcement learning (Fig. 5b) [116,117]. Whereas early works used transfer learning to bias the generation task by using a focused dataset of compounds with the desired properties [118,119], it is now common to couple the generation task to a reinforcement learning algorithm, which uses a supervised model to provide property-based feedback to the generative model [119,120]. By interactively exposing the generative model to fine-tuning, it learns common features and then updates its structural outputs to increasingly meet desired properties. Recently, Blaschke and Bajorath [119] developed an approach to create multi-target compounds by fine- tuning REINVENT [113], an LSTM cell-based deep generative network [121] originally pre-trained with ~1.4 million bioactive compounds from ChEMBL [122]. Initially, the authors compiled a set of experimentally confirmed no-target (inactive), single-target, and multi-target compounds in the SMILES format. Then, a random selection of 1000 multi-target compounds was used to fine-tuning the REINVENT generative model via transfer learning (Fig. 5b). After fine-tuning, a decision tree ensemble classifier systematically showed a clear tendency of the generative model to create multi-target compounds (26.6% of the newly generated compounds), while decreasing the likelihood of producing single- or no-target compounds. Taken together, these findings indicate that generative models can be adopted for *de novo* multi-target compound design [119].

Recently, Tan and co-workers [123] developed an automated system for the *de novo* design (Fig. 5c) of new multi-target compounds with antipsychotic activity, consisting of a three-step process: (a) generation of a focused library of compounds to simultaneously target the D_1, D_2, 5-HT_{1A}, and 5 HT_{2A} receptors; (b) virtual screening of the focused library via MTL-DNNs; and (c) employment of an interactive reinforcement learning environment to reward high-scoring and penalize low-scoring

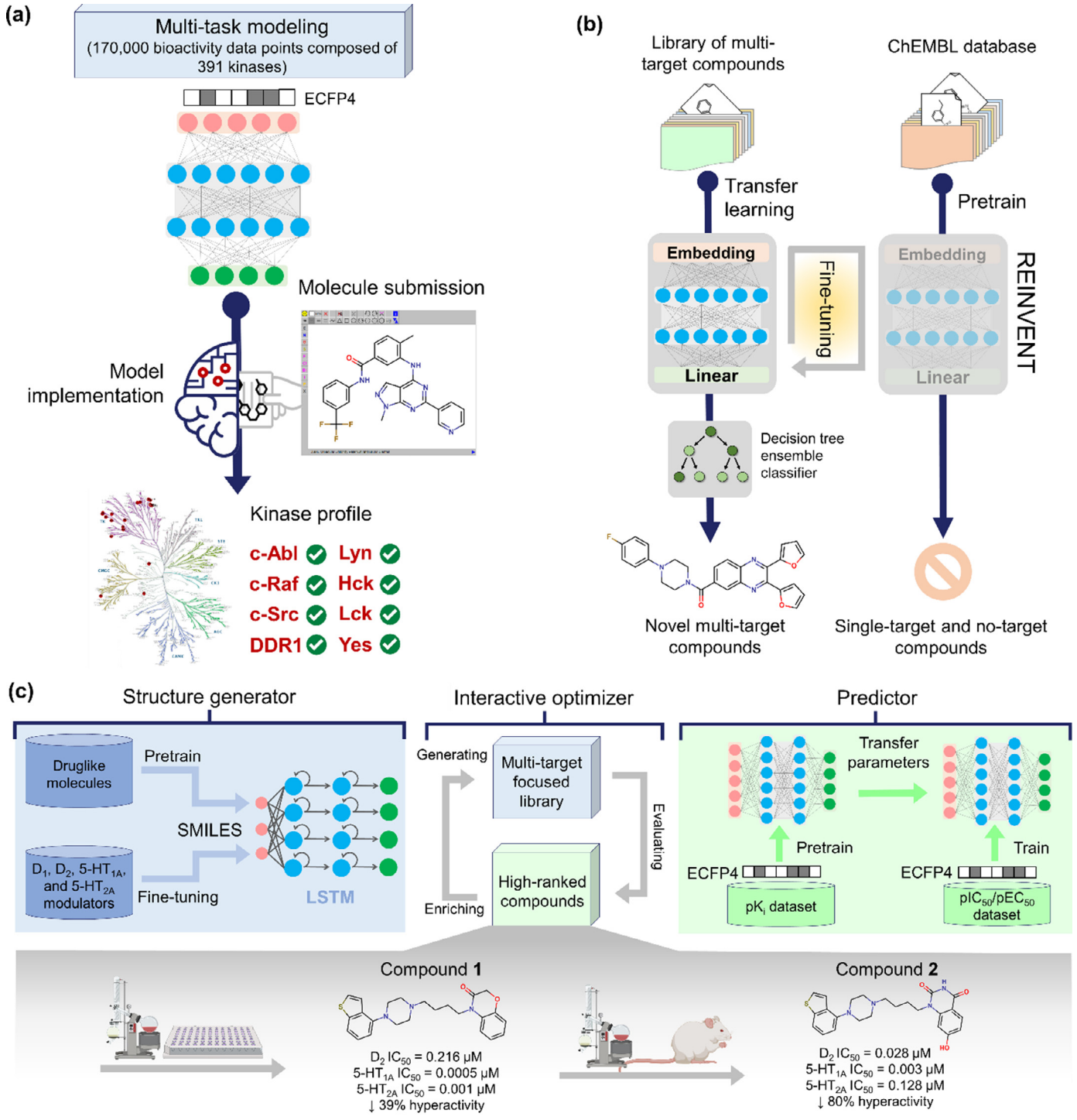

Fig. 5. Examples of automated deep learning (DL) systems explored in multi-target drug discovery. **(a)** The workflow of KinomeX, a web application for the prediction of multi-target kinase inhibitors; **(b)** Fine-tuning REINVENT generative neural network for the *de novo* design of multi-target compounds; and **(c)** Automated *de novo* design and optimization of multi-target lead compounds with antipsychotic activity through the integration of a deep generative network and a multi-task deep neural network (MTL-DNNs).

compounds. This generative model was pre-trained using RNN with two stacked LSTM layers by a large set of drug-like compounds in the SMILES format. Then, the pre-trained model was fine-tuned using a small library of compounds with experimental data against the D_1, D_2, 5-HT_{1A}, and 5 HT_{2A} receptors. In parallel, an MTL-DNNs using ECFP4 (Fig. 5c) fingerprints as inputs was employed to develop a regression model for predicting the activity of generated compounds in the context of the D_1, D_2, 5-HT_{1A}, and 5 HT_{2A} receptors [123]. During this process, a transfer learning strategy was then employed to learn the general features (weights) from a larger dataset of compounds with known pKi values, and apply them to a smaller dataset of compounds with defined pIC_{50} and pEC_{50} values, through fine-tuning. In each epoch, the generated chemical structures were interactively evaluated by the MTL-DNNs model and further screened based on various criteria, including drug-likeness and synthetic accessibility [123]. High-ranking compounds were then sampled to boost the training set for the next iteration. Importantly, this

system has successfully generated new compounds with desired multi-target activities. Among these new compounds, the high-ranking compound 1 demonstrated potent activity against D_2 (IC_{50} = 0.216 μM), 5-HT_{1A} (IC_{50} = 0.0005 μM), and 5-HT_{2A} (IC_{50} = 0.001 μM) receptors in biochemical experiments. The structural optimization of compound 1 based on the MTL-DNNs also led to the discovery of compound 2, which not only exhibited a multi-target profile but also showed a potent antipsychotic effect in mice (80% hyperactivity reduction) with low potential for catalepsy and sedation [123].

3.2.1. Multi-objective optimization problems

The drug discovery literature is riddled with MOO approaches to address multi-target optimization problems. Despite many of which are labeled "multi-objective", the line between multi- and single-objective molecular optimization is quite blurred [124,125]. The MOO refers to finding the optimal solution values of more than one desired target. The motivation for using the MOO is that optimization requires trivial equations, which simplifies the problem [125]. The equation of the MOO problem is defined as follows:

$$\begin{aligned} &min/max\ f_1(x),\ f_2(x), \ldots f_n(x) \\ &\quad \text{subject to}: \ x \in U \end{aligned} \tag{2}$$

where n is the number of objective functions, U is feasible set, x is solution, $f_n(x)$ is the nth objective function, and min/max is combined object operations [125].

A MOO task always begins with some statement of desired properties. First, the chemical and biological properties must be converted to mathematical objectives. If more than one objective exists, they must either be treated with an appropriate multi-objective formulation [124]. In this section, we explore two of these MOO formulations in detail, namely scalarization and Pareto method. An in-depth discussion of MOO methods in drug discovery is provided by Former and Coley [124].

The scalarization method reduces the multi-objective optimization problem into a single scalar vector function as follows:

$$F(x) = \ w_1 f_1(x) + \ w_2 f_2(x) + \ldots + \ w_n f_n(x) \tag{3}$$

where n is the number of objective functions, $F(x)$ is a combined fitness function, x is a solution, $f_n(x)$ is the nth objective function, and w is a constant weight for $F(x)$ [126]. The weight of an objective function will determine the solution of the fitness function and show the performance priority. For example, if a large weight is given to an objective function, it will have priority compared to the ones with a smaller weight [126].

In contrast to scalarization, the Pareto method discovers a set of solutions that reveal the trade-offs between objectives without a choice of any weighting factor of importance for the competing objectives [124,125]. The Pareto optimization can be written as follows:

$$\begin{aligned} f_{1,opt} &= \min f_1(x) \\ f_{2,opt} &= \min f_2(x) \\ &\ \ \vdots \\ f_{n,opt} &= \max f_n(x) \end{aligned} \tag{4}$$

where n is the number of objective functions, $f_n(x)$ is the nth objective function, x is a solution, and min/max is the threshold of the nth objective. Meanwhile, the $f_{1,opt}$ e $f_{2,opt}$ are the first two functions of the set optimal solution space that minimize the objective functions of f_1 and f_2. Mathematically, the Pareto optimization keeps the elements of the solution vectors independent during optimization, and the concept of dominance is there to differentiate the non-dominated and dominated solutions. The goodness of a solution is determined by dominance. A solution is called non-dominated if none of the objective functions can be improved without being detrimental to at least one other objective. The dominated solution is usually achieved when an objective function can be improved without reducing the remaining objective functions [124,125].

In view of the above, Pareto optimization is considered the standard gold MOO approach for *de novo* design [124]. Despite this, there is a lack of *de novo* design studies based on the Pareto front. Apparently, the main example in this field is DrugEx v.2.0, presented by Liu et al. [127]. The work consists of an LSTM cell-based deep generative network and a pool of ML predictors for estimating the activity towards the targets (A_1AR and A_{2A}AR) and one anti-target (hERG) [127]. Both the generative model and the predictors were pre-trained in advance and then interplayed under a reinforcement learning framework. During the interaction loop, the generative model creates a batch of SMILES-based molecules, whereas the scores provided by the predictors were used to construct Pareto ranks of the generated molecules [127]. The final reward of each SMILES is calculated based on the Pareto ranking with the non-dominated sorting genetic algorithm (NSGA-II) [128]. At the end of this process, the framework was able to generate a large percentage of valid SMILES with a predicted selectivity profile towards multiple targets, offering the potential of high efficacy and low toxicity [127].

4. Biological signature profiles

Biological signature matching is based on the comparison of the unique characteristics or "signature" of a drug with that of another compound, disease, or clinical phenotype. The biological signature of a drug could be derived from at least five general types of data: chemical information, OMICs (genomics, proteomics, metabolomics, metagenomics, and transcriptomics), biological pathways, Cells/Organs, and Clinics [129].

Duran-Frigola and co-workers [128] have presented the Chemical Checker, a database of these signatures containing ~800,000 molecules and their similarity at different levels of biology. Chemical Checker divides data into five levels of increasing complexity. The drug is often a compound (chemistry) that interacts with one or several protein receptors (targets), triggering perturbations of biological pathways (networks) and eliciting phenotypic, cell-based assays (cells), which can then be translated into their clinical outcomes (clinics). The authors showed that these signatures can aid drug discovery tasks, including target identification and library characterization.

Finding new molecules with a desired biological activity is a laborious task. At the beginning of the SARS-CoV-2 outbreak [130], in an open collaboration to find "old drugs" against SARS-CoV-2 we employed the PHAARM MatchMaker® InsilicAll's software, which used groundbreaking polypharmacology technology to set up the "molecular and biological signature" patterns, an approach that could bridge chemistry and biology in the long and difficult road of drug discovery. They showed AI tools capable of presenting the complex relationships between the chemical substance with molecular targets, cells and/or organs. This could create a more complete understanding of the "multi-biological profile" caused by the drug analyzed in response to a disease.

Example use cases of PHAARM MatchMaker® include: (a) therapy explorer templates focused on diseases or multiple targets as well as off-target affinity; (b) scaffold hopping to find new off-patent chemicals with similar or superior properties of the respective compounds approved for clinical use; (c) determining the likelihood of a preclinical toxicology event translating into the clinic by biological signature matched compound with the drug safety data across preclinical, clinical, and post-market commercial data; (d) comparison of comparing the biological signature of matched compound with the pharmacokinetic parameters available for marketed drugs (over one million commercial preclinical and clinical PK data, oral bioavailability, volume of distribution, C_{max}, T_{max}, $T_{1/2}$, AUC, etc.), targets, off-targets, phenotypic assays (cells), and clinical outcomes; (e) better anticipation of drug-drug interactions via the metabolizing enzymes and transporters and compare to DDI including positive and negative interaction cases; (f) finding drug repurposing opportunities by searching biological signatures with shared polypharmacology features [130].

Taken together, a pipeline of these AI tools can be generated for multi-target drug discovery. For example, recently, we created a procedure to test the marketed multikinase inhibitor imatinib. First, we used

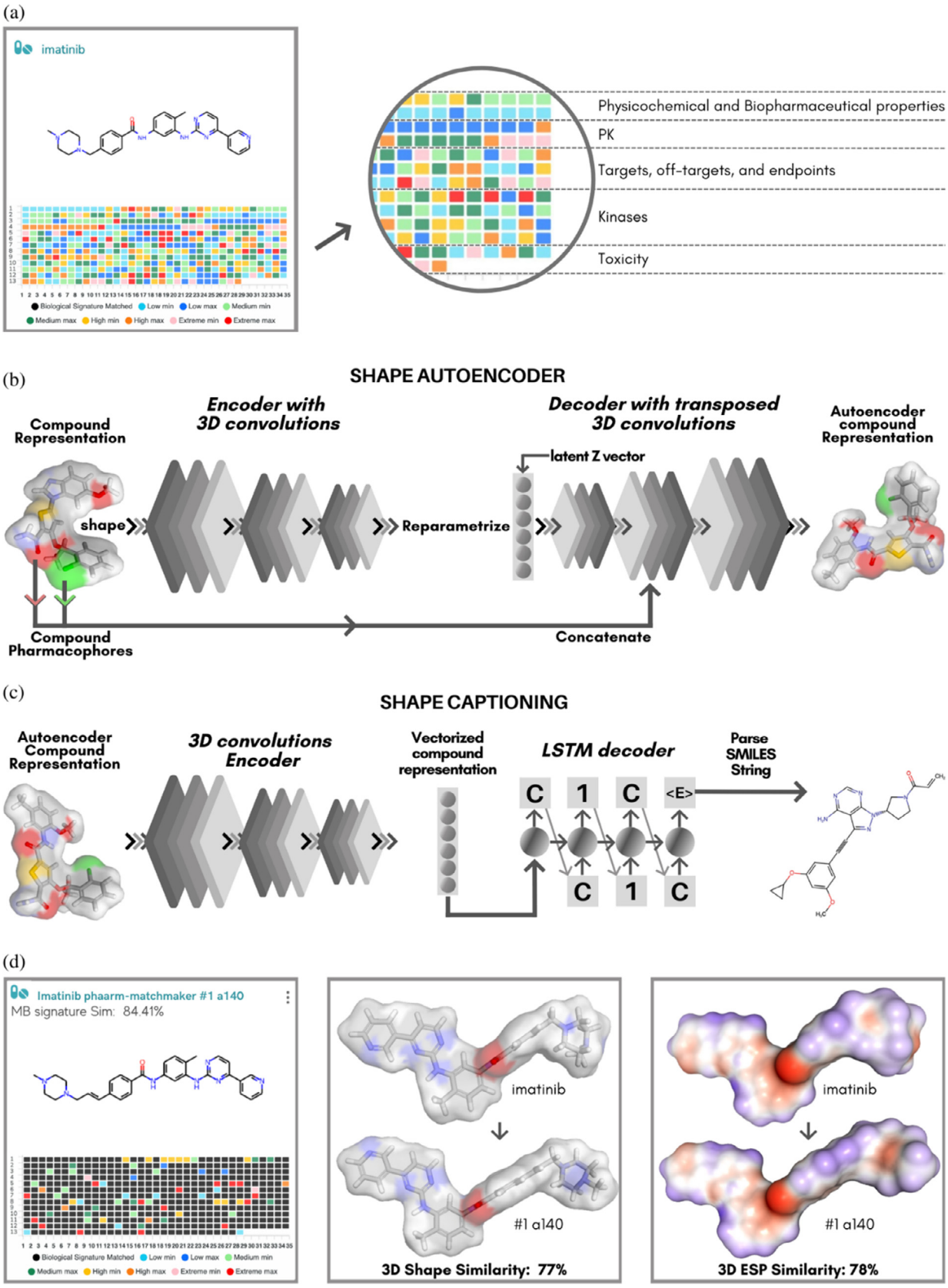

Fig. 6. Deep learning systems explored in multi-target drug discovery. **(a)** Using imatinib, a marketed multikinase inhibitor, as template for the evaluation of the "molecular and biological signature" patterns by PHAARM MatchMaker®; **(b)** scaffold hopping for the target (template) compound using the *de novo* drug design with 3D Shape-Based Generative Modeling to obtain the AI-generated compounds (a-n) (Figure S1) with similar shape and pharmacophore features; **(c)** determination of the 3D shape and electrostatic similarities (ESP-Sim Carbo) for the design of similar compounds; **(d)** generation of the "molecular and biological signature" patterns for new designed compounds with predicted multikinase inhibition. The compounds share a rather large common substructure.

imatinib as a template for PHAARM MatchMaker to evaluate its molecular and biological signature. Then, we used the *de novo* Drug Design with Shape-Based Generative Modeling, in which 3D spatial information is incorporated in two major phases: (i) a shape variational autoencoder employing CNNs encodes the compound representation, and (ii) a mix of CNNs and LSTM networks create SMILES strings. In Fig. 6, one can see a visual representation of the procedure. With the new approach, which uses autoencoders and captioning networks to vary molecules starting from a single volumetric representation, using the imatinib template, compounds were generated by AI (Figure S1a-n) with similar shape and pharmacophore features.

Partial charges are calculated using a neural network that combines B3LYP/6–31G** calculations with multi-task constraint message passing neural networks for atomic/bond properties. This allows the calculation of high-quality RESPs and shape scores for any given structure query giving an accurate prediction on whether it will have desirable sweet spot properties [131–133]. Finally, we generated the "molecular and biological signature" patterns for newly designed compounds with predicted multikinase inhibition (Figure S2). The compounds share a rather large common substructure. Importantly, all compounds activated more than 10 kinases models with strong binding affinity and IC_{50}. We showed compound with high MB signature and 3D Shape and ESP similarities.

The shape and electrostatic properties of molecules are what determine how similar two different substances can be. Learning from past drug discovery experiences, these characteristics should therefore play a major role in determining which comparison method is used. Sometimes, it is hard to reach the desired complex effect with one, one drug (even considering magic shotgun drugs). In this case, combining the drugs with different mechanisms of action and/or hitting a variety of selected targets could serve as an alternative approach. The application of such approaches in the discovery of synergistic drug combinations against SARS-CoV-2 has been reviewed elsewhere.

5. Conclusion

Historically, the discovery of new chemical entities with multi-target profile has been a huge challenge. Methodologically, the activity of these compounds needs to be simultaneously optimized toward several targets with dissimilar binding pockets. However, intuitively finding hidden structural patterns in chemical accessible space of multi-target compounds is quite difficult. To overcome this challenge, ML and DL tools stand out as cutting-edge decision support systems to optimize research pipelines and revolutionize multi-target drug discovery. Such AI systems are able to identify and learn from complex structural patterns and to design new molecular structures with desired multi-target profiles at a very low cost and in a time-efficient manner. However, many of these techniques are currently being evaluated in the context of prospective drug discovery. Thus, only a limited number of studies present experimental validation results. In this paper, we have highlighted some studies showing encouraging experimental results inspired by an automated design of multi-target compounds using deep generative networks and MTL-DNNs. Full development of these tools will require project-oriented interdisciplinary teams that closely integrate data science, synthesis, and biological evaluation to guide the next generation of multi-target drugs.

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.

R.C.B. is CTO of InsilicAll Inc. The other authors declare they have no actual or potential competing financial interests.

Data availability

No data was used for the research described in the article.

Acknowledgements

The authors appreciate the financial support from the Brazilian funding agencies, CNPq, FAPEG, and CAPES. The authors thank the NIH Fellows Editorial Board for the editorial assistance that greatly improved the manuscript.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2022.100055.

References

[1] Bolognesi ML, Cavalli A. Multitarget drug discovery and polypharmacology. ChemMedChem 2016;11:1190–2. doi:10.1002/cmdc.201600161.
[2] Atukorala I, Hunter DJ. Valdecoxib: the rise and fall of a COX-2 inhibitor. Expert Opin. Pharmacother 2013;14:1077–86. doi:10.1517/14656566.2013.783568.
[3] Dogne J-M, Hanson J, Supuran C, Pratico D. Coxibs and cardiovascular side-effects: from light to shadow. Curr. Pharm. Des. 2006;12:971–5. doi:10.2174/138161206776055949.
[4] Agüero F, Al-Lazikani B, Aslett M, Berriman M, Buckner FS, Campbell RK, et al. Genomic-scale prioritization of drug targets: the tdr targets database. Nat Rev Drug Discov 2008;7:900–7.
[5] Hunter DJ. Gene–environment interactions in human diseases. Nat. Rev. Genet. 2005;6:287–98. doi:10.1038/nrg1578.
[6] Bolognesi M L. Polypharmacology in a single drug: multitarget drugs. Curr. Med. Chem. 2013;20:1639–45. doi:10.2174/0929867311320130004.
[7] Ramsay RR, Popovic-Nikolic MR, Nikolic K, Uliassi E, Bolognesi ML. A perspective on multi-target drug discovery and design for complex diseases. Clin. Transl. Med. 2018;7:3. doi:10.1186/s40169-017-0181-2.
[8] Makhoba XH, Viegas C Jr, RA Mosa, FP Viegas, OJ Pooe. Potential impact of the multi-target drug approach in the treatment of some complex diseases. Drug Des Devel Ther. 2020;14:3235–49. doi:10.2147/DDDT.S257494.
[9] Talevi A. Multi-target pharmacology: possibilities and limitations of the "skeleton key approach" from a medicinal chemist perspective. Front Pharmacol 2015;6:205. doi:10.3389/fphar.2015.00205.
[10] Morphy R, Rankovic Z. Designed multiple ligands. an emerging drug discovery paradigm. J. Med. Chem. 2005;48:6523–43. doi:10.1021/jm058225d.
[11] Morphy R, Kay C, Rankovic Z. From magic bullets to designed multiple ligands. Drug Discov Today 2004;9:641–51. doi:10.1016/S1359-6446(04)03163-0.
[12] Roth BL, Sheffler DJ, Kroeze WK. Magic shotguns versus magic bullets: selectively non-selective drugs for mood disorders and schizophrenia. Nat Rev Drug Discov 2004;3:353–9. doi:10.1038/nrd1346.
[13] Li YH, Wang PP, Li XX, Yu CY, Yang H, Zhou J, et al. The human kinome targeted by fda approved multi-target drugs and combination products: a comparative study from the drug-target interaction network perspective. Zou Q, editor. PLoS ONE 2016;11:e0165737. doi:10.1371/journal.pone.0165737.
[14] Schultheis B, Folprecht G, Kuhlmann J, Ehrenberg R, Hacker UT, Köhne CH, et al. Regorafenib in combination with folfox or folfiri as first- or second-line treatment of colorectal cancer: results of a multicenter, phase ib study. Ann. Oncol. 2013;24:1560–7. doi:10.1093/annonc/mdt056.
[15] Crona DJ, Keisler MD, Walko CM. Regorafenib. Ann. Pharmacother 2013;47:1685–96. doi:10.1177/1060028013509792.
[16] Wilhelm SM, Dumas J, Adnane L, Lynch M, Carter CA, Schütz G, et al. Regorafenib (BAY 73-4506): a new oral multikinase inhibitor of angiogenic, stromal and oncogenic receptor tyrosine kinases with potent preclinical antitumor activity. Int. J. Cancer 2011;129:245–55. doi:10.1002/ijc.25864.
[17] Yang X, Wang Y, Byrne R, Schneider G, Yang S. Concepts of artificial intelligence for computer-assisted drug discovery. Chem. Rev. 2019;119:10520–94. doi:10.1021/acs.chemrev.8b00728.
[18] Schneider G. Automating drug discovery. Nat Rev Drug Discov 2017;17:97–113. doi:10.1038/nrd.2017.232.
[19] Abdolmaleki A, Ghasemi J, Ghasemi F. Computer aided drug design for multi-target drug design: sar /QSAR, molecular docking and pharmacophore methods. Curr. Drug Targets 2017;18:556–75. doi:10.2174/1389450117666160101120822.
[20] Andrade CH, Neves BJ, Melo-Filho CC, Rodrigues J, Silva DC, Braga RC, et al. In silico chemogenomics drug repositioning strategies for neglected tropical diseases. Curr. Med. Chem. 2019;26:4355–79. doi:10.2174/0929867325666180309114824.
[21] Zhao L, Ciallella HL, Aleksunes LM, Zhu H. Advancing computer-aided drug discovery (CADD) by big data and data-driven machine learning modeling. Drug Discov Today 2020;25:1624–38. doi:10.1016/j.drudis.2020.07.005.
[22] Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, et al. PubChem in 2021: new data content and improved web interfaces. Nucleic. Acids. Res. 2021;49:D1388–95. doi:10.1093/nar/gkaa971.
[23] Mendez D, Gaulton A, Bento AP, Chambers J, De Veij M, Félix E, et al. ChEMBL: towards direct deposition of bioassay data. Nucleic. Acids. Res. 2019;47:D930–40. doi:10.1093/nar/gky1075.
[24] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. Nucleic. Acids. Res. 2000;28:235–42. doi:10.1093/nar/28.1.235.

[25] Szklarczyk D, Santos A, von Mering C, Jensen LJ, Bork P, Kuhn M. STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data. Nucleic. Acids. Res. 2016;44:D380–4. doi:10.1093/nar/gkv1277.

[26] Chang A, Jeske L, Ulbrich S, Hofmann J, Koblitz J, Schomburg I, et al. BRENDA, the elixir core data resource in 2021: new developments and updates. Nucleic. Acids. Res. 2021;49:D498–508. doi:10.1093/nar/gkaa1025.

[27] Karp PD, Billington R, Caspi R, Fulcher CA, Latendresse M, Kothari A, et al. The biocyc collection of microbial genomes and metabolic pathways. Brief Bioinform 2019;20:1085–93. doi:10.1093/bib/bbx085.

[28] Williams AJ, Grulke CM, Edwards J, McEachran AD, Mansouri K, Baker NC, et al. The comptox chemistry dashboard: a community data resource for environmental chemistry. J Cheminform 2017;9:1–27. doi:10.1186/s13321-017-0247-6.

[29] Lou Dorne J, J Richardson, Kass G, Georgiadis N, Monguidi M, Pasinato L, et al. Editorial: openfoodtox: eFSA's open source toxicological database on chemical hazards in food and feed. EFSA J 2017;15:e15011. doi:10.2903/j.efsa.2017.e15011.

[30] Zarin DA, Tse T, Williams RJ, Califf RM, Ide NC. The clinicaltrials.gov results database — Update and key issues. N. Engl. J. Med. 2011;364:852–60. doi:10.1056/NEJMsa1012065.

[31] Thorn CF, Klein TE, Altman RB. Pharmacogenomics and bioinformatics: pharmGKB. Pharmacogenomics 2010;11:501–5. doi:10.2217/pgs.10.15.

[32] Zhang W, Pei J, Lai L. Computational multitarget drug design. J. Chem. Inf. Model 2017;57:403–12. doi:10.1021/acs.jcim.6b00491.

[33] Bajorath J. Foundations of data-driven medicinal chemistry. Futur Sci OA 2018;4:FSO320. doi:10.4155/fsoa-2018-0057.

[34] Schneider P, Walters WP, Plowright AT, Sieroka N, Listgarten J, Goodnow RA, et al. Rethinking drug design in the artificial intelligence era. Nat Rev Drug Discov 2020;19:353–64. doi:10.1038/s41573-019-0050-3.

[35] Moreira-Filho JT, Silva AC, Dantas RF, Gomes BF, Souza Neto LR, Brandao-Neto J, et al. Schistosomiasis drug discovery in the era of automation and artificial intelligence. Front. Immunol. 2021;12:1–29. doi:10.3389/fimmu.2021.642383.

[36] Li D, Hu J, Zhang L, Li L, Yin Q, Shi J, et al. Deep Learning and Machine Intelligence: New Computational Modeling Techniques for Discovery of the Combination Rules and Pharmacodynamic Characteristics of Traditional Chinese Medicine. Eur J Pharmacol 2022;933:175260. doi:10.1016/j.ejphar.2022.175260.

[37] Cherkasov A, Muratov EN, Fourches D, Varnek A, Baskin II, Cronin M, et al. QSAR modeling: where have you been? where are you going to? J. Med. Chem. 2014;57:4977–5010. doi:10.1021/jm4004285.

[38] Ekins S, Puhl AC, Zorn KM, Lane TR, Russo DP, Klein JJ, et al. Exploiting machine learning for end-to-end drug discovery and development. Nat. Mater. 2019;18:435–41. doi:10.1038/s41563-019-0338-z.

[39] Réda C, Kaufmann E. Delahaye-Duriez A. machine learning applications in drug development. Comput Struct Biotechnol J 2020;18:241–52. doi:10.1016/j.csbj.2019.12.006.

[40] Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, et al. Applications of machine learning in drug discovery and development. Nat Rev Drug Discov 2019;18:463–77. doi:10.1038/s41573-019-0024-5.

[41] Mitchell JBO. Machine learning methods in chemoinformatics. Wiley Interdiscip Rev Comput Mol Sci 2014;4:468–81. doi:10.1002/wcms.1183.

[42] Hansch C, Maloney PP, Fujita T, Muir RM. Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. Nature 1962;194:178–80. doi:10.1038/194178b0.

[43] Patel L, Shukla T, Huang X, Ussery DW, Wang S. Machine learning methods in drug discovery. Molecules 2020;25:5277. doi:10.3390/molecules25225277.

[44] Zhang L, Tan J, Han D, Zhu H. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. Drug Discov Today 2017;22:1680–5. doi:10.1016/j.drudis.2017.08.010.

[45] Maltarollo VG, Kronenberger T, Espinoza GZ, Oliveira PR, Honorio KM. Advances with support vector machines for novel drug discovery. Expert Opin Drug Discov 2019;14:23–33. doi:10.1080/17460441.2019.1549033.

[46] Jiménez-Luna J, Grisoni F, Weskamp N, Schneider G. Artificial intelligence in drug discovery: recent advances and future perspectives. Expert Opin Drug Discov 2021;16:949–59. doi:10.1080/17460441.2021.1909567.

[47] Svetnik V, Liaw A, Tong C, Christopher Culberson J, Sheridan RP, Feuston BP. Random forest: a classification and regression tool for compound classification and qsar modeling. J. Chem. Inf. Comput. Sci. 2003;43:1947–58. doi:10.1021/ci034160g.

[48] Sharma S, Sharma D. Intelligently applying artificial intelligence in chemoinformatics. Curr. Top. Med. Chem. 2018;18:1804–26. doi:10.2174/1568026619666181120150938.

[49] Lavecchia A. Machine-learning approaches in drug discovery: methods and applications. Drug Discov Today 2015;20:318–31. doi:10.1016/j.drudis.2014.10.012.

[50] Pedrycz W, Sosnowski ZA. The design of decision trees in the framework of granular data and their application to software quality models. Fuzzy Sets Syst. 2001;123:271–90. doi:10.1016/S0165-0114(00)00118-4.

[51] Kingsford C, Salzberg SL. What are decision trees? Nat. Biotechnol. 2008;26:1011–12. doi:10.1038/nbt0908-1011.

[52] Podgorelec V, Kokol P, Stiglic B, Rozman I. Decision trees: an overview and their use in medicine. J. Med. Syst. 2002;26:445–63. doi:10.1023/A:1016409317640.

[53] Breiman L. Bagging predictions. Mach Learn 1996;24:123–40.

[54] Carracedo-Reboredo P, Liñares-Blanco J, Rodríguez-Fernández N, Cedrón F, Novoa FJ, Carballal A, et al. A review on machine learning approaches and trends in drug discovery. Comput Struct Biotechnol J 2021;19:4538–58. doi:10.1016/j.csbj.2021.08.011.

[55] Musolf AM, Holzinger ER, Malley JD, Bailey-Wilson JE. What makes a good prediction? feature importance and beginning to open the black box of machine learning in genetics. Hum. Genet. 2021. doi:10.1007/s00439-021-02402-z.

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

[57] Cortes C, Vapnik V. Support-Vector networks. Mach Learn 1995;20:273–97. doi:10.1023/A:1022627411411.

[58] Vapnik VV. The nature of statistical learning theory. 2nd ed. New York: Springer; 2000.

[59] Heikamp K, Bajorath J. Support vector machines for drug discovery. Expert Opin Drug Discov 2014;9:93–104. doi:10.1517/17460441.2014.866943.

[60] Lane TR, Foil DH, Minerali E, Urbina F, Zorn KM, Ekins S. Bioactivity comparison across multiple machine learning algorithms using over 5000 datasets for drug discovery. Mol Pharm 2021;18:403–15. doi:10.1021/acs.molpharmaceut.0c01013.

[61] Blaschke T, Miljković F, Bajorath J. Prediction of different classes of promiscuous and nonpromiscuous compounds using machine learning and nearest neighbor analysis. ACS Omega 2019;4:6883–90. doi:10.1021/acsomega.9b00492.

[62] Heikamp K, Bajorath J. Prediction of compounds with closely related activity profiles using weighted support vector machine linear combinations. J. Chem. Inf. Model. 2013;53:791–801. doi:10.1021/ci400090t.

[63] Wang Y, Xiao J, Suzek TO, Zhang J, Wang J, Zhou Z, et al. PubChem's bioassay database. Nucleic. Acids. Res. 2012;40:D400–12. doi:10.1093/nar/gkr1132.

[64] Cortés-Ciriano I, Ain QU, Subramanian V, Lenselink EB, Méndez-Lucio O, IJzerman AP, et al. Polypharmacology modelling using proteochemometrics (PCM): recent methodological developments, applications to target families, and future prospects. Medchemcomm 2015;6:24–50. doi:10.1039/C4MD00216D.

[65] Schaduangrat N, Anuwongcharoen N, Phanus-umporn C, Sriwanichpoom N, Wikberg JES, Nantasenamat C. Proteochemometric modeling for drug repositioning. In: Silico drug design. Elsevier; 2019. p. 281–302. doi:10.1016/B978-0-12-816125-8.00010-9.

[66] Sejnowski TJ. The unreasonable effectiveness of deep learning in artificial intelligence. Proc. Natl Acad. Sci. 2020;117:30033–8. doi:10.1073/pnas.1907373117.

[67] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. doi:10.1038/nature14539.

[68] Chen H, Engkvist O, Wang Y, Olivecrona M, Blaschke T. The rise of deep learning in drug discovery. Drug Discov Today 2018;23:1241–50. doi:10.1016/j.drudis.2018.01.039.

[69] Spjuth O, Frid J, Hellander A. The machine learning life cycle and the cloud: implications for drug discovery. Expert Opin Drug Discov 2021:1–9. doi:10.1080/17460441.2021.1932812.

[70] Raina R, Madhavan A, Ng AY. Large-scale deep unsupervised. In: Learning using graphics processors. proceedings of the 26th annual international conference on machine learning - ICML '09. New York, New York, USA: ACM Press; 2009. p. 1–8. doi:10.1145/1553374.1553486.

[71] Wang Y.E., Wei G.-Y., Brooks D. Benchmarking T.P.U., GPU, and CPU Platforms for Deep Learning. 2019. Available: http://arxiv.org/abs/1907.10701

[72] Serin G, Sener B, Ozbayoglu AM, Unver HO. Review of tool condition monitoring in machining and opportunities for deep learning. Int. J. Adv. Manuf. Technol. 2020;109:953–74. doi:10.1007/s00170-020-05449-w.

[73] Kolluri S, Lin J, Liu R, Zhang Y, Zhang W. Machine learning and artificial intelligence in pharmaceutical research and development: a review. AAPS J. 2022;24:1–10. doi:10.1208/s12248-021-00644-3.

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

[75] Rodríguez-Pérez R, Bajorath J. Evaluation of multi-target deep neural network models for compound potency prediction under increasingly challenging test conditions. J Comput Aided Mol Des 2021;35:285–95. doi:10.1007/s10822-021-00376-8.

[76] Kato Y, Hamada S, Goto H. Validation study of qsar/dnn models using the competition datasets. Mol Inform 2020;39:1900154. doi:10.1002/minf.201900154.

[77] Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure–activity relationships. J. Chem. Inf. Model. 2015;55:263–74. doi:10.1021/ci500747n.

[78] Chen H, Kogej T, Engkvist O. Cheminformatics in drug discovery, an industrial perspective. Mol Inform 2018;37:1800041. doi:10.1002/minf.201800041.

[79] Sarker IH. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci 2021;2:420. doi:10.1007/s42979-021-00815-1.

[80] Gawehn E, Hiss JA, Schneider G. Deep learning in drug discovery. Mol Inform 2016;35:3–14. doi:10.1002/minf.201501008.

[81] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Procedings 14th Int Conf Artif Intell Stat; 2011. p. 315–23.

[82] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference On International Conference On Machine Learning; 2010. p. 807–14.

[83] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6. doi:10.1038/323533a0.

[84] Afaq S, Rao S. Significance of epochs on training a neural network. Int J Sci Technol Res 2020;19:485–8. Available www.ijstr.org .

[85] Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw. 2015;61:85–117. doi:10.1016/j.neunet.2014.09.003.

[86] Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. Molecular graph convolutions: moving beyond fingerprints. J Comput Aided Mol Des 2016;30:595–608. doi:10.1007/s10822-016-9938-8.

[87] Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, et al. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. J Cheminform 2021;13:12. doi:10.1186/s13321-020-00479-8.

[88] Krenn M., Ai Q., Barthel S., Carson N., Frei A., Frey N.C., et al. SELFIES and the future of molecular string representations. 2022. doi:10.1016/j.patter.2022.100588
[89] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9:1735–80. doi:10.1162/neco.1997.9.8.1735.
[90] Chung J., Gulcehre C., Cho K., Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. Available: http://arxiv.org/abs/1412.3555
[91] Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J. LSTM: a search space odyssey. IEEE Trans Neural Networks Learn Syst 2017;28:2222–32. doi:10.1109/TNNLS.2016.2582924.
[92] Cho K., van Merrienboer B., Bahdanau D., Bengio Y. On the properties of neural machine translation: encoder-decoder approaches. 2014. Available: http://arxiv.org/abs/1409.1259
[93] Veličković P., Cucurull G., Casanova A., Romero A., Liò P., Bengio Y. Graph attention networks. 2017. Available: http://arxiv.org/abs/1710.10903
[94] Gilmer J., Schoenholz S.S., Riley P.F., Vinyals O., Dahl G.E. Neural message passing for quantum chemistry. 2017. Available: http://arxiv.org/abs/1704.01212
[95] Gaudelet T, Day B, Jamasb AR, Soman J, Regep C, Liu G, et al. Utilizing graph machine learning within drug discovery and development. Brief Bioinform 2021;22:1–22. doi:10.1093/bib/bbab159.
[96] Zeng X, Tu X, Liu Y, Fu X, Su Y. Toward better drug discovery with knowledge graph. Curr. Opin. Struct. Biol. 2022;72:114–26. doi:10.1016/j.sbi.2021.09.003.
[97] Zhang Z, Chen L, Zhong F, Wang D, Jiang J, Zhang S, et al. Graph neural network approaches for drug-target interactions. Curr. Opin. Struct. Biol. 2022;73:102327. doi:10.1016/j.sbi.2021.102327.
[98] Sun M, Zhao S, Gilvary C, Elemento O, Zhou J, Wang F. Graph convolutional networks for computational drug development and discovery. Brief Bioinform 2020;21:919–35. doi:10.1093/bib/bbz042.
[99] Méndez-Lucio O, Ahmad M, del Rio-Chanona EA, Wegner JK. A geometric deep learning approach to predict binding conformations of bioactive molecules. Nat Mach Intell 2021;3:1033–9. doi:10.1038/s42256-021-00409-9.
[100] Unterthiner T, Mayr A, Klambauer G, Steijaert M, Wegner J, Ceulemans H, et al. Deep learning as an opportunity in virtual screening. Adv Neural Inf Process Syst 2014;27:1–9.
[101] Xu Y, Ma J, Liaw A, Sheridan RP, Svetnik V. Demystifying multitask deep neural networks for quantitative structure-activity relationships. J. Chem. Inf. Model. 2017;57:2490–504. doi:10.1021/acs.jcim.7b00087.
[102] Ramsundar B, Liu B, Wu Z, Verras A, Tudor M, Sheridan RP, et al. Is multitask deep learning practical for pharma? J. Chem. Inf. Model. 2017;57:2068–76. doi:10.1021/acs.jcim.7b00146.
[103] Perera V, Chung T, Kollar T, Strubell E. Multi-Task learning for parsing the alexa meaning representation language. Proc AAAI Conf Artif Intell 2018;32. doi:10.1609/aaai.v32i1.12019.
[104] Chen Y, Yu J, Zhao Y, Chen J, Du X. Task's choice: pruning-based feature sharing (PBFS) for multi-task learning. Entropy 2022;24:432. doi:10.3390/e24030432.
[105] Sanh V., Wolf T., Ruder S. A hierarchical multi-task approach for learning embeddings from semantic tasks. 2018. Available: http://arxiv.org/abs/1811.06031
[106] Hashimoto K, Xiong C, Tsuruoka Y, Socher R. A joint many-task model: growing a neural network for multiple nlp tasks. In: Proceedings of the 2017 conference on empirical methods in natural language processing. Stroudsburg, PA, USA: Association for Computational Linguistics; 2017. p. 1923–33. doi:10.18653/v1/D17-1206.
[107] Sun T., Shao Y., Li X., Liu P., Yan H., Qiu X., et al. Learning sparse sharing architectures for multiple tasks. 2019. Available: http://arxiv.org/abs/1911.05034
[108] Sun T, Shao Y, Li X, Liu P, Yan H, Qiu X, et al. Learning sparse sharing architectures for multiple tasks. Proc AAAI Conf Artif Intell 2020;34:8936–43. doi:10.1609/aaai.v34i05.6424.
[109] Tang H, Liu J, Zhao M, Gong X. Progressive layered extraction (PLE): a novel multi-task learning (MTL) model for personalized recommendations. In: Fourteenth ACM Conference on Recommender Systems. New York, NY, USA: ACM; 2020. p. 269–78. doi:10.1145/3383313.3412236.
[110] Li Z, Li X, Liu X, Fu Z, Xiong Z, Wu X, et al. KinomeX: a web application for predicting kinome-wide polypharmacology effect of small molecules. Wren J, editor. Bioinformatics 2019;35:5354–6. doi:10.1093/bioinformatics/btz519.
[111] Li X, Li Z, Wu X, Xiong Z, Yang T, Fu Z, et al. Deep learning enhancing kinome-wide polypharmacology profiling: model construction and experiment validation. J. Med. Chem. 2020;63:8723–37. doi:10.1021/acs.jmedchem.9b00855.
[112] Polykovskiy D, Zhebrak A, Sanchez-Lengeling B, Golovanov S, Tatanov O, Belyaev S, et al. Molecular sets (MOSES): a benchmarking platform for molecular generation models. Front Pharmacol 2020;11:565644. doi:10.3389/fphar.2020.565644.
[113] Blaschke T, Arús-Pous J, Chen H, Margreitter C, Tyrchan C, Engkvist O, et al. REINVENT 2.0: an ai tool for de novo drug design. J. Chem. Inf. Model 2020;60:5918–22. doi:10.1021/acs.jcim.0c00915.
[114] Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: benchmarking models for de novo molecular design. J. Chem. Inf. Model 2019;59:1096–108. doi:10.1021/acs.jcim.8b00839.
[115] Meyers J, Fabian B, Brown N. De novo molecular design and generative models. Drug Discov Today 2021. doi:10.1016/j.drudis.2021.05.019.
[116] Martinelli DD. Generative machine learning for de novo drug discovery: a systematic review. Comput. Biol. Med. 2022;145:105403. doi:10.1016/j.compbiomed.2022.105403.
[117] Wang M, Wang Z, Sun H, Wang J, Shen C, Weng G, et al. Deep learning approaches for de novo drug design: an overview. Curr. Opin. Struct. Biol. 2022;72:135–44. doi:10.1016/j.sbi.2021.10.001.
[118] Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent Sci 2018;4:120–31. doi:10.1021/acscentsci.7b00512.
[119] Blaschke T, Bajorath J. Fine-tuning of a generative neural network for designing multi-target compounds. J Comput Aided Mol Des 2021. doi:10.1007/s10822-021-00392-8.
[120] Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. Sci. Adv. 2018;4:eaap7885. doi:10.1126/sciadv.aap7885.
[121] Arús-Pous J, Johansson SV, Prykhodko O, Bjerrum EJ, Tyrchan C, Reymond J-L, et al. Randomized smiles strings improve the quality of molecular generative models. J Cheminform 2019;11:71. doi:10.1186/s13321-019-0393-0.
[122] Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, et al. ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic. Acids. Res. 2012;40:D1100–7. doi:10.1093/nar/gkr777.
[123] Tan X, Jiang X, He Y, Zhong F, Li X, Xiong Z, et al. Automated design and optimization of multitarget schizophrenia drug candidates by deep learning. Eur. J. Med. Chem. 2020;204:112572. doi:10.1016/j.ejmech.2020.112572.
[124] Fromer J.C., Coley C.W. Computer-Aided multi-objective optimization in small molecule discovery. 2022. Available: http://arxiv.org/abs/2210.07209
[125] Gunantara N. A review of multi-objective optimization: methods and its applications. Ai Q, editor. Cogent Eng 2018;5:1502242. doi:10.1080/23311916.2018.1502242.
[126] Murata T, Ishibuchi H, Tanaka H. Multi-objective genetic algorithm and its applications to flowshop scheduling. Comput. Ind. Eng. 1996;30:957–68. doi:10.1016/0360-8352(96)00045-9.
[127] Liu X, Ye K, van Vlijmen HWT, Emmerich MTM, IJzerman AP, van Westen GJP. DrugEx v2: de novo design of drug molecules by pareto-based multi-objective reinforcement learning in polypharmacology. J Cheminform 2021;13:85. doi:10.1186/s13321-021-00561-9.
[128] Deb K., Agrawal S., Pratap A., Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: nsga-ii. 2000. pp. 849–58. doi:10.1007/3-540-45356-3_83
[129] Pushpakom S, Iorio F, Eyers PA, Escott KJ, Hopper S, Wells A, et al. Drug repurposing: progress, challenges and recommendations. Nat Rev Drug Discov 2018;18:41–58. doi:10.1038/nrd.2018.168.
[130] Ekins S, Mottin M, Ramos PRPS, Sousa BKP, Neves BJ, Foil DH, et al. Déjà vu: stimulating open drug discovery for SARS-CoV-2. Drug Discov Today 2020;25:928–41. doi:10.1016/j.drudis.2020.03.019.
[131] Heid E, Fleck M, Chatterjee P, Schröder C, Mackerell AD. Toward prediction of electrostatic parameters for force fields that explicitly treat electronic polarization. J. Chem. Theory Comput. 2019;15:2460–9. doi:10.1021/acs.jctc.8b01289.
[132] Guan Y, Coley CW, Wu H, Ranasinghe D, Heid E, Struble TJ, et al. Regio-selectivity prediction with a machine-learned reaction representation and on-the-fly quantum mechanical descriptors. Chem. Sci. 2021;12:2198–208. doi:10.1039/d0sc04823b.
[133] Rathi PC, Ludlow RF, Verdonk ML. Practical high-quality electrostatic potential surfaces for drug discovery using a graph-convolutional deep neural network. J. Med. Chem. 2020;63:8778–90. doi:10.1021/acs.jmedchem.9b01129.