# Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition

Sebastian Raschka[a], Benjamin Kaufman[b]

[a]*University of Wisconsin–Madison, Department of Statistics*
[b]*University of Wisconsin–Madison, Department of Biostatistics and Medical Informatics*

## ABSTRACT

In the last decade, machine learning and artificial intelligence applications have received a significant boost in performance and attention in both academic research and industry. The success behind most of the recent state-of-the-art methods can be attributed to the latest developments in deep learning. When applied to various scientific domains that are concerned with the processing of non-tabular data, for example, image or text, deep learning has been shown to outperform not only conventional machine learning but also highly specialized tools developed by domain experts. This review aims to summarize AI-based research for GPCR bioactive ligand discovery with a particular focus on the most recent achievements and research trends. To make this article accessible to a broad audience of computational scientists, we provide instructive explanations of the underlying methodology, including overviews of the most commonly used deep learning architectures and feature representations of molecular data. We highlight the latest AI-based research that has led to the successful discovery of GPCR bioactive ligands. However, an equal focus of this review is on the discussion of machine learning-based technology that has been applied to ligand discovery in general and has the potential to pave the way for successful GPCR bioactive ligand discovery in the future. This review concludes with a brief outlook highlighting the recent research trends in deep learning, such as active learning and semi-supervised learning, which have great potential for advancing bioactive ligand discovery.

## Abbreviations

**AI** Artificial intelligence

**CNN** Convolutional neural network

**DNN** Deep neural network

**ECFP** Extended connectivity fingerprint

**GNN** Graph neural network

**GPCR** G protein-coupled receptor

**GPU** Graphics processing unit

**GRU** Gated recurrent unit

**MLP** Multilayer perceptron

**NN** Neural network

**QSAR** Quantitative structure-activity relationship

**RNN** Recurrent neural network

*Corresponding author
✉ sraschka@wisc.edu (S. Raschka)
🌐 http://pages.stat.wisc.edu/~sraschka/ (S. Raschka)
ORCID(s): 0000-0001-6989-4493 (S. Raschka)
[1] Medical Science Center, 1300 University Ave, Madison, 53706 WI

arXiv:2001.06545v1 [q-bio.BM] 17 Jan 2020

**SVM** Support vector machine

**VAE** Variational autoencoder

**VS** Virtual screening

## 1. Introduction

G protein-coupled receptors (GPCRs) are one of the most prominent families of integral membrane proteins and are among the most widely studied targets in drug discovery and development. According to [1], 34 percent of all drugs approved by the US Food and Drug Agency are GPCRs. While the exact size of the GPCR family remains to be determined, genome analyses suggest that the GPCR family is represented by approximately 800-1,000 genes in humans [2, 3]. More than 150 GPCRs are characterized as orphan receptors, which means that the receptorsâĂŹ endogenous ligands are still unknown [4]. However, it remains unclear whether endogenous ligands exist for all orphan GPCRs [5].

Since GPCRs are involved in many different cellular and biological processes, make excellent drug targets, and remain orphaned to a relatively large extent, the prediction and consequent identification of GPCR ligands is an active area of research and interest [6]. Recent studies suggest that drugs target only approximately 10% of known GPCRs [7]. The high cost of clinical trials coupled with a relatively low success rate (approximated to be below 6.2% [8, 9]), motivates the development and application of machine learning and artificial intelligence-based methods for ligand discovery, to lower costs as well as to identify candidates that would otherwise be missed using conventional methods.

### 1.1. GPCRs and drug discovery

Many common human diseases involve GPCR signaling, including schizophrenia, glaucoma, depression, and hypertension [2]. Despite the increasing popularity of biologics, which includes gene therapy, recombinant therapeutic proteins, antibody-drug conjugates, and tissues [10], experts see the discovery of small-molecule ligands being crucial for the future of molecular medicine and the treatment of human diseases [11, 12].

GPCR ligands exhibit a wide range of characteristics and are very diverse in their physiochemical properties, shape, and size [7]. Furthermore, GPCR ligands include lipids, peptides, proteins, steroids, and other small organic molecules. Depending on their mechanism of action, GPCR ligands can be described as full or partial agonists, antagonists, or inverse agonists. This diversity among GPCR ligands poses challenges for standardized binding assays used in experimental bioactivity screening. Thus, wet-lab techniques such as high throughput screening have risen in popularity. However, high throughput screening can still be cost-prohibitive and labor-intensive. Another downside of high throughput screening is the limitation of available ligand libraries in terms of size and diversity. With drug discovery being an expensive and labor-intensive process, with estimated costs ranging between 0.5-2.6 billion US dollars, and a timeline ranging between 10-20 years, researchers have begun to favor computational methods for identifying candidate molecules for more elaborate bioactivity assays [13, 14].

When it comes to the identification of protein targets of small-molecule ligands and assessing potential off-target activity, chemical proteomics are considered the gold standard [12, 15–17]. However, similar to the challenges of binding assays for small molecule screening, chemical proteomics experiments are laborious and challenging to streamline [18]. Hence, many researchers are increasingly favoring computational alternatives [12] or augmenting the experimental discovery pipeline with artificial intelligence [19, 20].

### 1.2. Computer-aided ligand discovery

Modern computational databases feature hundreds of millions of molecules and are available free of charge [21–23], which makes computer-aided ligand discovery a particularly compelling alternative to high throughput screening and other experimental approaches during the early stages of ligand discovery. The two major approaches for computer-aided ligand discovery, also known as virtual screening (VS), are ligand-based VS and structure-based VS.

Ligand-based VS approaches focus on the structure and physicochemical properties of ligands in the absence of the receptor structure. Broadly, ligand-based VS can be described as a set of techniques for ligand-based similarity search or property prediction. Ligand-based VS approaches for identifying bioactive molecules are often based on a known bioactive molecule given the hypothesis that similar molecules are likely to bind the target receptor and exhibit a certain level of bioactivity, which has to be determined experimentally.

In contrast to ligand-based VS, structure-based VS assumes knowledge of the receptor structure with molecular docking being one of the most prominent structure-based VS techniques. While molecular docking has led to many successful discoveries, one of its limitations for GPCR ligand discovery is the small number of high-resolution GPCR structures that are currently available. Recent years brought many significant improvements for protein X-ray crystallography and cryo-electron microscopy; however, high-resolution structures still only cover four out of the six different classes of GPCR: classes A, B, C, and F, of which the rhodopsin-like receptors from class A form the most significant portion [24]. Furthermore, another limitation for structure-based approaches for bioactive ligand discovery is the breadth and diversity of GPCR structures that are currently available: only 44 of the 205 GPCR structures correspond to unique GPCRs [1].

All GPCRs consist of seven transmembrane helices, which are relatively conserved across the different classes of GPCRs (A-F). However, GPCRs may differ more noticeably in the intracellular and extracellular loop domains. The extracellular loops play an essential role in many GPCR structures, since they often form the orthosteric ligand binding site or provide access to binding sites that are located within the transmembrane bundle [25]. The diversity of GPCR ligands and GPCR ligand binding sites are a direct consequence of the structural variety of the extracellular loops, which makes employing structure-based approaches extremely challenging in the absence of high-quality structural information. In those cases, ligand-based strategies may represent the only viable alternative [26]. In recent years, both traditional structure-based and ligand-based VS have been augmented or replaced by machine learning techniques. Furthermore, machine learning has also been used for predicting bioactivity from other types of data such as a bioactivity matrix (a targets-by-compounds matrix of functional interactions) [27]. Today, machine learning is widely recognized as an essential method in the chemical biology toolbox for researching ligand binding [12].

## 1.3. Augmenting ligand discovery with artificial intelligence and machine learning

Artificial intelligence (AI) is traditionally considered a subfield of computer science that focuses on tasks that humans are naturally good at, such as image recognition and natural language processing. Furthermore, can be categorized into artificial general intelligence, i.e., human-level intelligence on a breadth of different asks, and "narrow" AI, which focuses on solving a single task well. Examples of narrow AI include tasks such as classifying objects from images, language translation, playing chess, or predicting the bioactivity of small molecules. Currently, the most popular approach for implementing and programming narrow AI is machine learning. Machine learning is a field that focuses on the development of algorithms that allow computers to learn from representative datasets, as opposed to having domain experts developing rules manually.

The three major subcategories of machine learning are supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the goal is to predict a category label (classification) or score (regression analysis) by learning from a large collection of such labeled examples. In other words, supervised learning is concerned with learning a mapping function between the so-called input features or observations (for example, fingerprint representations of small molecules) and a discrete or continuous target variable, for example, an active/inactive label or the binding affinity in the context of a specific receptor. After learning to predict the desired scores or labels from the labeled training dataset, an independent test dataset, consisting of labeled examples that were unseen during training, is used to evaluate the performance of the predictive model. Unsupervised learning is used for tasks that do not involve target labels. A typical example of unsupervised learning is clustering, for instance, grouping small molecules based on a user-specified similarity measure. Reinforcement learning, a third major category of machine learning, focuses on decision making in complex environments, which requires learning an optimal series of actions in response to environmental information as opposed to a target output as in supervised learning. A classic example is the development of a chess program.

In the presence of labeled high-quality data, machine learning provides opportunities for automating the development of customized solutions that are optimized for a specific type of receptor, binding site, or small molecule âĂŞ as opposed to more general scoring functions such as DrugScore [28] or AutoDock Vina [29]. Machine learning has already shown to be successful in various stages throughout typical ligand and drug discovery, including the discovery of novel drug targets and ligands [30, 31], bioactivity prediction [32], predicting new binding sites in GPCRs [33, 34], analyzing the association between the drug targets and diseases [35], studying binding pathways (for example, the binding pathways of opiates to *mu*-opioid receptors [36]), optimizing lead compounds [37], molecular *de novo* design [38], modifying molecular properties [39], and developing biomarkers to assess the efficacy of drugs [40].

## 1.4. Deep learning–representation learning with deep neural networks

In the last decade, deep learning research has seen a substantial increase in attention, since it has allowed researchers to develop state-of-the-art solutions for computer vision, natural language processing, and computational biology [41]. Deep learning is a subfield of machine learning that focuses on artificial neural networks with many layers that can learn multiple levels of abstractions of data that are useful for supervised and unsupervised learning tasks. Modern deep learning architectures consist of hundreds of millions of parameters [42], the so-called model weights, which are learned from a training dataset using the backpropagation algorithm [43]. An overview of the most commonly used deep learning architectures is provided in Section 1.7.

The predictive performance of traditional machine learning methods âĂŞ this includes logistic regression, random forests, support vector machines (SVMs), k-nearest neighbors, and many others – heavily depends on the design of the feature engineering pipeline. For instance, the aforementioned conventional machine learning methods generally do not operate well on high dimensional datasets and are unable to extract knowledge from raw data (such as text or images) [41]. Hence, the training data has to be converted into a tabular format via manual feature engineering âĂŞ for example, by representing a document as a vector of word frequencies [44], or by representing flowers by tabulated measurements of their leaf sizes rather than photographs [45]. Careful feature engineering requires substantial domain expertise, and useful information contained in the data can be lost by feature engineering. Deep learning, on the other hand, relies on general-purpose algorithms that include the automatic extraction of salient information from raw data as part of the modeling architecture and optimization objective. In this respect, deep learning can be characterized as a feature or representation learning method. However, one downside of deep learning is that it is relatively data-hungry, and datasets for supervised learning require large amounts of labeled examples âĂŞ dataset sizes ranging between 50 thousand and 15 million training examples are not unusual [46, 47].

## 1.5. Open source software and datasets

Over the past decade, there has been a remarkable increase in the development of open source software for enabling data science and machine learning [48]. Many general-purpose scientific computing libraries with permissive open source licenses are now widely used in academia as well as in industry, including NumPy and SciPy [49], Matplotlib [50], and Pandas [51]. Furthermore, general-purpose libraries have been developed for the analysis of biological and structural data that lower the barrier of entry to computational biology, for example, BioPython [52] and BioPandas [53].

Under similar open source licensing terms, the Scikit-learn machine learning library [54] has been widely adopted for predictive modeling with traditional machine learning (for example, generalized linear models, SVMs, and tree-based methods such as random forests and gradient boosting). In recent years, computationally efficient and GPU-enabled libraries such as TensorFlow [55] and PyTorch [56] made deep learning accessible to the broad scientific communities similar to how Scikit-learn contributed to popularizing machine learning.

While the use of machine learning and deep learning has seen widespread adoption throughout various research areas, a bottleneck for applications to computational biology was the lack of large, annotated, high-quality datasets. However, recent years brought both improvements of experimental techniques and the âĂIJdata and code sharingâĂİ culture in academia, which led to the increase of publicly available molecular activity and biomedical datasets. For instance, in drug discovery, recent efforts focused on developing an open, annotated datasets that can be utilized for therapeutic target validation [57]. MoleculeNet is a large benchmark dataset consisting of more than 700,000 molecules and their property annotation [58]. The ChEMBL large-scale bioactivity database contains more than 5.4 million bioactivity measurements for 5,200 protein targets and more than 1 million ligands [59]. While the aforementioned databases are general-purpose datasets that are usually used for general algorithm development and benchmarking, these can be used as a starting point for pre-training deep learning models that can then be fine-tuned to smaller datasets containing a specific target using transfer learning, which is discussed in greater detail in Section 6.

## 1.6. Interpretability and repeatability of machine learning

Machine learning models can learn to identify salient patterns in high-dimensional and complex datasets that are not obvious to a human researcher. However, machine learning and particularly deep learning models are often criticized for their black box aspect and lack of interpretability [9]. We agree that machine-learning-based methods, except for the most fundamental decision tree models and rule-based classifiers, are less interpretable than pure âĂIJif/thenâĂİ rules. However, many methods exist that allow scientists to obtain insights into which features a model learns for making particular predictions. For example, in [26], the researchers were able to identify a potent inhibitor of GPCR pheromone

receptor signaling using a machine learning-aided virtual screening approach that provided structure-activity information. Furthermore, by combining machine learning with feature selection algorithms, the researchers were able to identify sulfate groups on the tail end of the candidate ligands that are crucial for bioactivity in SLOR-1 GPCR signaling [60]. While the methods described in [60] were applied to summarize structure-activity relationships of 3D-structural overlays of small molecules that were prioritized for experimental bioassays, the same techniques can be used for different types of data, such as hydrogen bonding patterns [61] or ligand-receptor docking poses [62].

Other commonly used methods for understanding model predictions are general-purpose methods such as LIME [63] or SHAP [64]. Neural network-specific methods include guided backpropagation [65], class activation mapping [66], gradient-weighted class activation mapping [67], and learning important features through propagating activation differences (DeepLIFT) [68].

Another point of criticism against the use of deep neural networks is the lack of repeatability [9]. In this context, repeatability refers to the ability to produce the exact same results upon repeating an experiment under identical conditions; in contrast, reproducibility refers to the ability to obtain similar results in different environments or conditions, for example, if a different research lab conducts similar experiments. According to [9], the issue of repeatability âĂIJarises because [machine learning] outputs are highly dependent on the initial values or weights of the network parameters or even the order in which training examples are presented to the network, as all of them are typically chosen at random.âĂİ We want to highlight that these repeatability issues can easily be circumvented by specifying the seeds of pseudo-random number generators and setting the behavior of machine learning and deep learning software libraries to deterministic. In practice, however, issues with reproducibility and repeatability arise when researchers share insufficient details about the software versions that were used to conduct the experiments. Reproducibility and repeatability require best code sharing practices, which includes not only sharing the exact code but also the experimental settings and software version numbers. Fortunately, sharing data and code has become a recommended practice in the field of machine learning [69], and we hope that computational biology journals will adopt similar best practices such that repeatability and reproducibility issues that plagued the field in previous years can hopefully be avoided in the future.

## 1.7. Commonly used deep learning architectures

This section summarizes the main concepts behind machine learning with a focus on the fundamental deep learning concepts that are relevant for the remainder of this article. For an introduction and overview of general machine learning methods, we recommend [48, 70]. For a more comprehensive introduction to deep learning, we recommend [41, 48, 71].

### 1.7.1. Multilayer perceptrons

Multilayer perceptrons (MLPs) are fully connected artificial neural networks (NNs) that consist of an input layer, an output layer, and at least one hidden layer between the two (Figure 1 A). While there is no precise definition of what constitutes a *deep* neural network (DNN), an NN that has more than one hidden layer is commonly referred to as a DNN. The hidden units in the hidden layer manipulate the input information (observations or features) in a non-linear fashion so that, in the case of supervised classification, the training examples from different classes become linearly separable by the last layer. In DNNs, the early layers can be considered representation learning layers that distort the data in such a way that it can be classified by an output layer that has a similar structure to a generalized linear model. In other words, NN architectures with one or more hidden layers can learn highly complex relationships between the input features and the target label.

An MLP with only one (sufficiently large) hidden layer can already be considered a universal function approximator [72–74]. However, a DNN with many hidden layers can achieve the same expressiveness with fewer parameters than an NN with only one hidden layer. Furthermore, constructing an architecture with multiple layers provides some form of regularization: later layers are constrained by the behavior of earlier layers. Unfortunately, a common problem with deep architectures is that increasing the number of layers exacerbates the vanishing and exploding gradient problems that arise from repeated multiplications via the chain rule in backpropagation, making it hard to parameterize very deep models. A particular focus in the deep learning field has been on developing methods that help with training very deep architectures successfully, as discussed in Section 1.7.2 and Section 1.7.3.

While the early MLP architecture was first proposed in the early 1960s [75–77], efficient ways for training such multilayer neural networks using the backpropagation procedure were formulated by several researchers independently between 1970 and 1986 [43, 78, 79]. Although the backpropagation procedure is still the main learning algorithm in deep learning, the training of DNNs has only become broadly feasible in recent years due to several advances towards
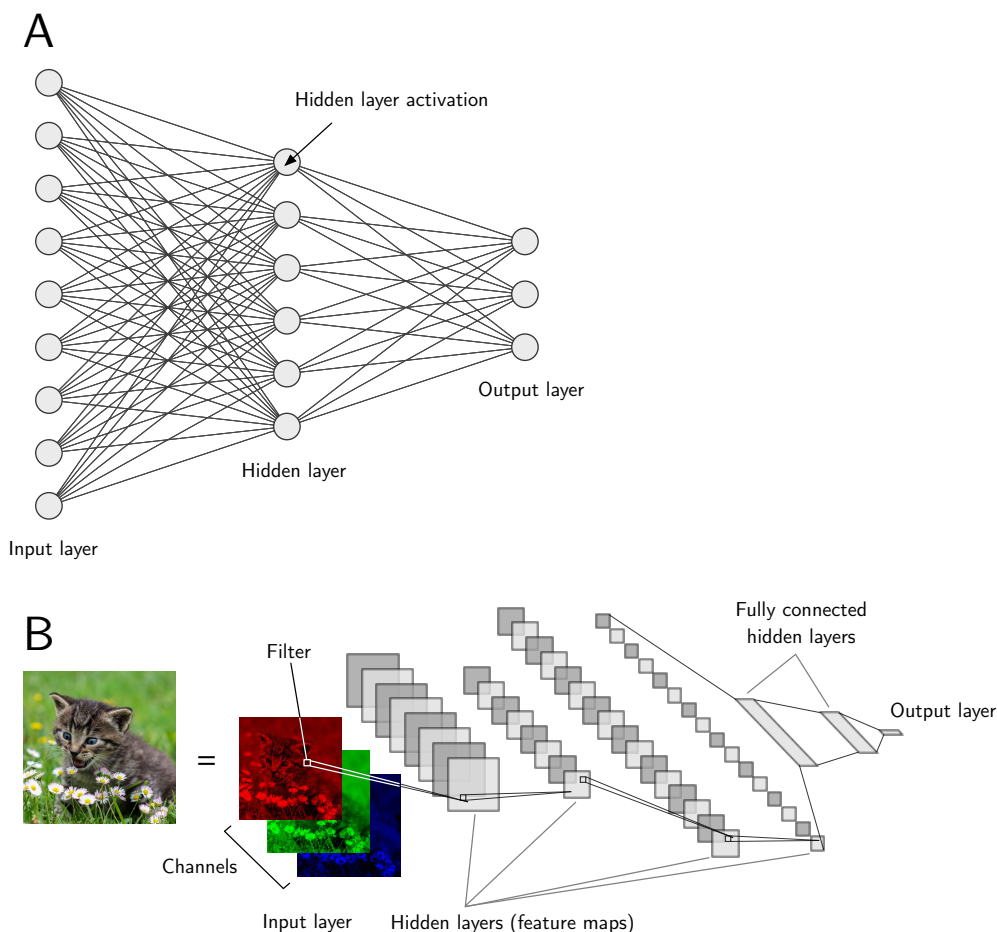
**Figure 1:** (A) A fully connected neural network (multilayer perceptron; MLP) with one hidden layer. The input layer represents the input features from the dataset, and it is connected to the hidden layer via non-linear activation functions. Similarly, non-linear activation functions connect the hidden layer to the output layer. Here, the three units in the output layer may represent three separate class labels to be predicted in a supervised learning context. (B) Depiction of a convolutional neural network (CNN) architecture for image recognition, which consists of multiple feature maps that are connected via convolutional layers. Each unit in a feature map is connected with a local patch of the previous layerâĂŹs feature map through a set of filters or so-called kernel. Traditionally, pooling layers have been inserted between every other feature map but are not necessary [65]. The fully connected hidden layers and output layer resemble an MLP architecture.

making the general training more efficient and effective. These improvements include implementing machine learning models and training DNNs on GPUs [80–83], which are extremely well-suited for performing linear algebra operations such as matrix multiplications efficiently and on a large scale.

In addition to developing software libraries that allow researchers to develop and implement DNNs more easily (see Section 1.5), countless algorithmic advances towards training DNNs more efficiently and robustly have been made in recent years. For example, new activation functions have been proposed, such as the rectified linear unit (ReLU) [84], which has become a popular default choice since it can help with vanishing gradient effects and generally allows for training DNNs faster and with better predictive performance. New stochastic gradient descent-based optimization algorithms such as Adam [85] can accelerate the model training and make it easier to find a proper learning rate setting compared to conventional stochastic gradient descent. Regularization methods such as Dropout [86] help reduce the effect of overfitting by dropping nodes randomly during training and thus make the model less reliant on particular inputs and connectivity patterns. Batch normalization is a method for scaling layer inputs [87], which can speed up learning further by allowing learning with larger batch sizes and converging to local minima on the loss surface in

fewer iterations over the training set. While not all of these improvements exist in all modern architectures, they have been fundamental to allowing experts as well as non-experts to train DNNs successfully on a wide range of different datasets.

In a quantitative structure-activity relationship (QSAR) study, researchers found that in 13 out of 15 cases, very simple DNNs were already able to outperform traditional machine learning methods such as random forest and SVMs for quantitative-activity relationship analysis even when trained on traditional fingerprint representations [88]. This is a remarkable result because, in this case, the model the researchers referred to as DNN was a simple MLP with only one hidden layer. Additional details about the different molecular data input representations will be provided in Section 2.

### 1.7.2. Convolutional neural networks

Convolutional neural networks (CNNs) are feedforward neural networks that utilize the discrete convolution operation as a filtering operation on multi-dimensional arrays (Figure mlp-cnn). Depending on the architectural design and discrete convolution operation, CNNs can handle various data modalities, including 1D (sequences and signals in vector form), 2D (pictures and audio spectrograms), and 3D arrays (such as videos or images with depth dimension like 3D computerized tomography scans).

Today, CNNs are the most widely used deep learning architectures. They are widely recognized for their state-of-the-art performance for image-related tasks such as object classification, detection, and segmentation. The first successful demonstration of CNNs for image classification was the design of the LeNet architecture for handwritten digit and document recognition in the 1990s [89, 90]. While CNNs have shown exceptional performance in various image recognition tasks, including the segmentation of biological images [41, 91], the rise of CNNs to mainstream success can be attributed to 2012 ImageNet competition, where CNNs were able to outperform traditional computer vision methods by doubling the object classification accuracy on a database consisting of millions of images [92].

Similar to MLP architectures, CNNs are feedforward neural networks. However, in contrast to MLPs, CNNs do not use fully connected layers. The main concepts behind CNNs that distinguish this architecture from MLPs are *sparse connectivity* and *parameter sharing*. Here, sparse connectivity refers to the fact that each unit in the hidden layer is only connected to a small number of units in the previous layer. The connection occurs through a so-called kernel or filter, which is a parameter matrix that is commonly interpreted as a feature extractor. Via the filter, information between neighboring pixels is combined to compute the activation of a unit in the next hidden layer. This allows CNNs to compose hierarchies of local features to recognize complex shapes in later layers. For instance, in an image classification task, a multi-layer CNN may learn to detect simple geometric shapes like edges and curves in the earlier layers, and in later layers, it can identify more sophisticated features such as cars or buildings based on the underlying geometric shapes. These feature detectors typically only cover a small portion of the input image or feature map, and in the process of the convolution operation, they can be thought of as a sliding window over the image. Hence, the weights for the different patches of the input image or feature maps are shared within a given layer, which is inspired by the fact that a feature detector that works well in one region may also work well in another part. The advantage of weight sharing is that it reduces the number of parameters that need to be learned, which improves computational efficiency and can reduce overfitting, compared to fully connected networks.

In traditional CNN architectures, feature maps were followed by pooling layers [89, 90]. These pooling layers combine similar features from neighboring regions in the previous feature map into a single feature, which is supposed to help neural networks to learn more complex relationships or geometrical shapes. However, it has been shown that pooling layers are not a requirement in modern CNN architectures [65].

Deep learning is a fast-moving field, and even revolutionary architectures such as AlexNet, which outperformed state-of-the-art computer vision methods at the ImageNet competition in 2012 [92], have long been succeeded by other reference architectures. For example, the VGG architecture has achieved notable performance by shrinking the receptive field and stacking many more layers than AlexNet [93]. The ResNet architecture introduced residual connections, which are shortcut connections between layers that help with vanishing gradient problems and thus improve the training of very deep neural networks [94]. Inception networks improved the extraction aspect of both local and global features by using multiple filter modules in parallel [95]. While the design of CNN architectures still remains an active area of research, recent efforts have been targeted towards improving efficiency, for example, MobileNet [96] and EfficientNet [97]. In a benchmark study, Canziani et al. compare both the predictive and computational performance of these models for further reference [42].
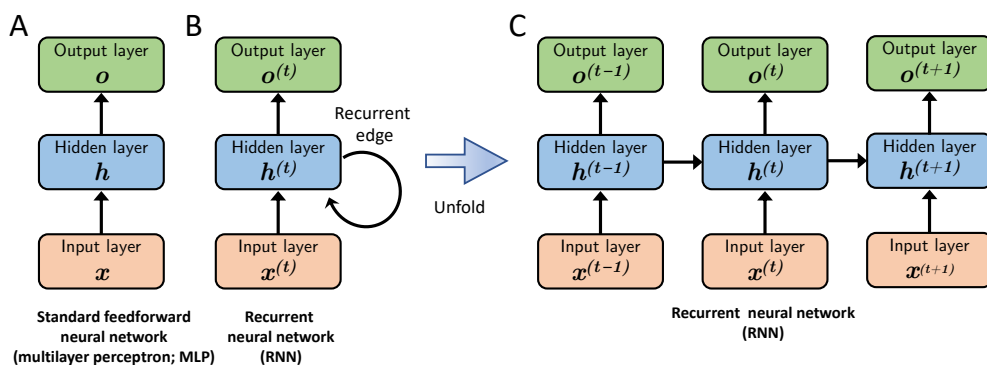
**Figure 2:** Subpanels (A) and (B) depict an MLP with 1 hidden layer and an RNN with 1 hidden layer side by side for comparison. Here, the input layer $x$ containing the input features, the hidden layer $h$, and the output layer $o$ represent vectors with multiple units each. Subpanel (C) depicts the RNN in its unfolded representation. In contrast to an MLP, the hidden layer of an RNN receives its inputs from both the input layer and the hidden layer of the previous time step, $t-1$.

### 1.7.3. Recurrent neural networks

Recurrent neural networks (RNNs) were originally designed for tasks that involve sequence data, for example, textual data and audio signals such as speech. While traditional RNNs consist of fully connected layers similar to MLPs, RNNs have recurrent connections through time. This recurrence property allows RNNs to maintain an internal history of sequence elements that it processed previously (Figure 2).

While RNNs can model arbitrary long sequences, the recurrent edges make this architecture particularly prone to vanishing and exploding gradient problems [98]. One solution to this problem is the so-called long short-term memory mechanism (LSTM) by Hochreiter and Schmidhuber [99], which is a memory cell that replaces the hidden layer of conventional RNNs. A mechanism similar to LSTM, although slightly more computationally efficient, is the Gated Recurrent Unit (GRU) [100].

Since its first proposal in 2017, the Transformer architecture [101] has gained a lot of attention as it has been shown to outperform RNNs on many sequence modeling tasks [102]. The Transformer architecture relies on a so-called attention mechanism, which allows for modeling global dependencies between input and output sequences. In other words, the attention mechanism allows the model to focus on only those parts of the text that are relevant for the prediction task. While the Transformer architecture is still relatively new, it has already been adopted for several applications in computational biology, including retrosynthetic reaction prediction [103], QSAR models [104], and learning molecular fingerprint representations [105].

### 1.7.4. Autoencoders

Autoencoders refer to architectures that were traditionally used for dimensionality reduction and trained in an unsupervised manner [106]. The main idea behind autoencoders is that they learn to preserve the essential part of the data and remove the non-essential ones (Figure 3).

An autoencoder consists of two neural networks: an encoder that transforms a training example into a lower-dimensional representation, and a decoder that reconstructs the original training example from its lower-dimensional representation. For example, if both the encoder and decoder submodules of the autoencoder are fully connected NNs with linear activation functions, the latent feature embedding produced by the encoder is equivalent to features transformed via principal component analysis except for the feature orthogonality constraint.

It shall be noted that regular autoencoders are deterministic models that can be used for data compression or to modify existing data given a user-specified objective [107–110]. A related model is the variational autoencoder, which can be considered a generative model as it is capable of synthesizing entirely new, realistic data records mimicking the data contained in the training dataset [111, 112].

The remainder of this paper is organized as follows. First, we discuss one of the common challenges with applying machine learning to bioactive ligand discovery, namely choosing an appropriate feature representation for the ligand structure, and highlight the recent developments using deep learning. Next, we discuss the latest trends for ligand-based analysis, from molecular property prediction to similarity-based virtual screening. While the ligand-based approaches
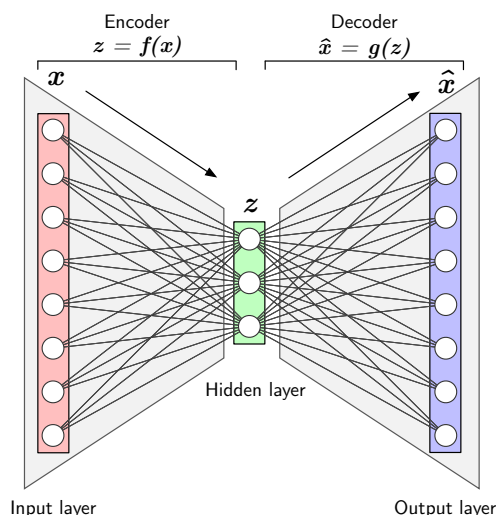
**Figure 3:** A fully connected autoencoder, which consists of an Encoder submodule that compresses an input representation $x$ into a lower-dimensional representation $z$. The Decoder, the second submodule of the autoencoder, converts the lower-dimensional representation as $\hat{x}$, such that reconstruction resembles the original input, $\hat{x} \approx x$.

assume that a high-quality receptor structure is not available, the next section reviews the recent developments for receptor structure-based bioactive ligand discovery. This includes the detection of new GPCR binding sites, protein-ligand docking, and *de novo* small-molecule design. Finally, this review concludes by motivating the use of transfer learning, which allow researchers to make better use of publicly available data in machine learning and AI-based bioactive ligand discovery.

## 2. Molecular feature representations

Machine learning methods excel at prediction tasks across multiple disciplines but require careful data preparation as most methods are designed to operate on tabular datasets. The standard data input format is the so-called *design matrix*, where each row represents a new training example, and the columns correspond to the different feature variables. A common challenge in conventional machine learning is how to prepare datasets as input to machine learning algorithms – in practice, machine learning practitioners have to find a sweet spot between reducing the dimensionality and retaining salient information that the model can learn from. In contrast to conventional machine learning, deep learning excels at learning from *raw* data, such as images and text, directly. However, molecular data, such as conformations of small molecules and receptors, can be challenging to represent in a standard format that most machine learning and deep learning methods have been designed for. Even if the same information can be extracted from two different data representations, an algorithm may be more effective at extracting that information from one over the other. There is no clear best representation of molecules for machine learning methods and indeed certain representations may be better for certain tasks. The following section provides a brief overview of commonly used molecular representations as well as some recent applications of them using AI-based methods.

### 2.1. Property-based feature vectors

A molecular descriptor is the transformation of chemical information into a numeric value [113]. Dragon [114] and Mordred descriptors [115] are examples of sets of molecular descriptors. As an alternative to molecular descriptors, molecular fingerprints encode molecular structure in a vector format, a so-called bit vector consisting of 1âĂŹs and 0âĂŹs. When used as input for machine learning models, both molecular fingerprints and descriptors have historically produced state-of-the-art results on chemical machine learning tasks such as chemical odor prediction and bioactivity [37, 116].

The extended connectivity fingerprint (ECFP) is among the most widely-used 2D fingerprint methods [117], and we use its generation procedure as an example of the general process for generating traditional molecular fingerprints.

A fingerprint is generated by a multistep process in which each atom is associated with a series of integers. In this series the $k$th integer encodes information about the atom it is associated with as well as information about the atoms and bonds within $k$ bonds of that atom – that is, the substructure of the compound that is within $k$ bonds of the atom. Next, the integers associated with each atom are concatenated into an array format, which is then processed via a hashing algorithm to generate a bit vector of a desired length (typically 1024 or 2048 elements). This method captures information about all identified substructures in a compound, resulting in a fixed-length vector regardless of the input compoundâĂŹs size. ECFPs do not explicitly encode the 3D spatial information of a compound; however, specialized fingerprint methods have recently been developed that incorporate 3D-strucutral information [118]. Lastly, there are also fingerprints that can encode protein-ligand interactions [119].

## 2.2. SMILES

Simplified molecular-input line-entry system (SMILES) strings are ASCII string representations of compounds (Figure 4 A), which are generated according to a procedure that guarantees a unique mapping from a SMILES string to a compound structure (though not the inverse) [120]. One benefit of SMILES strings over 2D molecular fingerprints like ECFP is that they encode stereochemistry explicitly. One downside for machine learning is that SMILES do not have a fixed-length; however, certain deep learning architectures designed for processing text documents, like RNNs or 1D CNNs, can handle variable-length inputs.

Recently, Hirohara et al. [121] proposed a novel molecular representation scheme by converting SMILES strings into âĂIJSMILES feature matrices,âĂİ which were used as inputs into a 1D CNN [121]. A SMILES feature matrix was constructed by mapping a SMILES string of length $N$ to a $N \times 42$ matrix, where the $k$th row represents the $k$th character (corresponding to either atom or connectivity information) of the string, and the 42 columns correspond to properties of that character. To address the problem of varying-length input strings, the feature matrix was padded with rows of zeros such that all feature matrices had their number of rows set to the length of the longest SMILE string. The predictive performance of the resulting CNN model was comparable with other deep learning methods on the Tox 21 dataset, which contains 8,000 compounds labeled as active or inactive for 12 proteins [122]. More interestingly though, this novel feature representation method enabled the extraction of a 64-dimensional vector from a convolutional layer referred to as the *SMILES convolution fingerprint*, which can be mapped back to the model input SMILES, providing an interpretable data-driven fingerprint.

Another notable recent development is the SMILES2vec method, which combines components from both recurrent and convolutional neural networks [123]. Here, the model input is a SMILES string converted into a one-hot encoding of characters present in the SMILES in the training dataset. The one-hot encoding then serves as an input to a 1D convolutional layer, which is followed by two GRUs before the fully-connected output layer that returns the target value. This architecture achieved equivalent performance to the state-of-the-art at the time on the ESOL solubility dataset [124].

## 2.3. 3D voxels

Molecular structures can be considered as 3D objects, and modeling them as such would encode their structural properties effectively. The conventional way for encoding 3D objects is by voxelization, which, in this case, takes a 3D space and discretizes it into a 3D grid. Each unit cube of the 3D grid represents a voxel (Figure 4 C), which can be considered as the equivalent of a 2D pixel in a 3D space. Instead of merely labeling each voxel via a binary membership indicator based on what part of the compound occupies the voxel, a feature vector with discrete or continuous-valued attributes can be assigned to each voxel, which can provide further information about the atom type or charge, for example. This adds an additional dimension to the representation. Since this representation is typically associated with a large number of input features, depending on the resolution of the voxel space as well as the size of the vector representation at each voxel position, deep learning approaches used with this type of input representation typically rely on 3D convolutions. This is equivalent to using 2D convolutions, which are commonly used for image analysis (Section 1.7.2), in the 3D voxel space. Unfortunately, these convolutions can still be prohibitively slow, even at relatively low voxel resolutions, which can result in a coarse representation of the compounds properties. Since each cube represents a unit of measurement, and compounds can vary in size, this representation needs to be padded so that it can accommodate for the largest compound in the dataset, which exacerbates the computational efficiency challenges of this method. Likely owed to being computationally very intensive and inefficient, this method is not a common choice for ligand-based virtual screening. However, this representation has shown to be successful for structure-based VS where binding pocket size is consistent regardless of the interacting ligand, and capturing the 3D structure of the
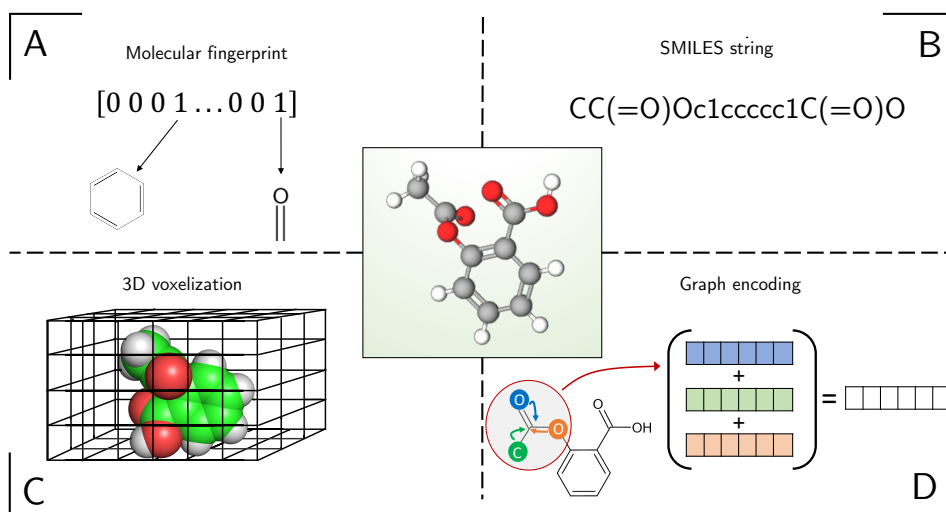
**Figure 4:** Summary of commonly used molecular representation methods based on the example of Aspirin (shown in the center). (A) A molecular fingerprint encodes structural motifs into a sparse bit vector. (B) A SMILES string encoding structural information of the molecule as well as its stereochemistry. (C) A visualization of the 3D voxelization concept. Note that information about which atoms occupy which voxels would be encoded in a 4th dimension which is omitted in this visualization. (D) Illustration of how information is passed to an atom in a simple graph neural network. Note that the graph-structural information will be passed from more distant atoms when the summation is repeated (not shown).

protein-ligand complex is important for the task.

We examine the 3D voxelization process from Ragoza et al. as a concrete example [125]. In this publication the authors create 3D voxel representations of high-scoring docking poses of protein-ligand complexes for use in a 3D CNN that computes binding affinities. This representation voxelizes a 24 Å cube centered on the docked ligand into .5 Å voxels. Each voxel has channels for Smina atom types in the ligand and separately for Smina atom types in the protein [126]. The value contributed to a channel by an atom of the type that is associated with that channel is based on a function of two values: First, the distance between the center of the voxel and the center of that atom, and second, the atomâĂŹs van der Waals radius." This function is a continuous piece-wise combination of a Gaussian and a quadratic based on these two values.

## 2.4. Graph representation

Molecules are commonly visualized as undirected graphs. In this representation, atoms represent the graphâĂŹs nodes, and the bonds are the graphâĂŹs edges. A naive approach that utilizes such graphical information is to consider pictures of molecular graphs as inputs to a DNN. This has been tried in the literature using 2D CNN architectures that have been effective for image classification [127, 128]. However, these models do not outperform MLP and random forest trained on ECFPs. Two issues with this approach are that atomic properties and spatial relationships must be inferred implicitly and that the representation is sparse, with lots of white space that provides little chemically relevant information to the CNNs. Both these issues can be addressed with graph neural networks (GNNs) [129, 130].

NNs operate directly on a molecular graph (Figure 4 D). Similar to images, graphs can encode local structure, but the local structural information is based on the graphs structure rather than Euclidean distance. Like CNNs, GNNs utilize sparse connectivity and parameter sharing, but the connectivity is based on the graph's structure. For a detailed explanation of graph convolutional operators, we refer to [130]. On a high-level, graph convolutions are based on message passing frameworks for GNNs. At each node of the graph, the following steps are performed: a *message* function is applied to each of a nodes' neighbors individually, and the outputs are then added. An *update* function is then applied to the summed message functions and the current node with the output being the updated value for the current node. After graph convolutions are performed, the information in the graph can be aggregated by a *readout* function, which produces the networks' prediction output.

# 3. Ligand-based methods

Ligand-based VS methods are traditionally defined as methods that only rely on physicochemical and structural information about the ligand and sometimes measurements of ligand-receptor interactions. No other information about the target, like the receptor structure, is utilized. The use of ligand-based methods is particularly appealing when working with GPCRs for which only a small set of high-quality structures exist.

In the following subsection, we review the most notable advances in predicting molecular properties, which are applicable to GPCR bioactive molecule VS.

## 3.1. Molecular property prediction

Researchers applied a variety of standard machine learning algorithms to the task of classifying compounds as active or inactive with two GPCRs, cannaboid receptor 1 (CB1) and cannaboid receptor 2 (CB2) [131]. In addition to the activity prediction, for CB1, the researchers also predicted if compounds were allosteric or orthosteric. Here, molecules with a an inhibitory constant ($K_i$) greater than 100 for the given receptor were defined as orthosteric positives and allosteric positives were selected from an existing database of allosteric compounds [132]. The researchers trained a set of machine learning classifiers on each task and compared their performance. This set included SVM, MLPs with 1-5 hidden layers, random forest, naïve Bayes and a logistic regression model. In this study, the researchers considered ECFP fingerprints, MACCS fingerprints, as well as molecular descriptor input feature representations. While multiple models had comparable performances, an MLP with only one hidden layer performed best overall on a variety of performance metrics with the MACCS featurization while logistic regression performed best in combination with the ECFP. There was no consistent top performer with the molecular descriptors. Notably, deeper MLP models were also assessed and did not have strong performance, which is likely due to vanishing gradient problems as discussed in Section 1.3 and the limited dataset size (on the order of 1000s).

The above result suggests different compound representations capture chemical information in manners that are utilized more or less effectively by different machine learning algorithms. If a model took multiple representations as input the ensemble may have better performance. The model CheMixNet followed this approach and used both a SMILES and fingerprint representation of a molecule as input [133]. The authors considered four subnetworks three of which received a one-hot SMILES string as input and one that was trained on an MACCS fingerprint representation. The three SMILES subnetworks respectively consisted of GRU cells, 1D convolutional layers, and a combination of 1D convolutional layers and GRU cells; the fingerprint subnetwork was an MLP. The authors trained models with combinations of these subnetworks, where the outputs of each subnetwork were concatenated and passed through additional fully connected layers before a linear or softmax layer to predict the property. The model was trained and tested multiple times to predict a variety of properties. In all cases certain subnetwork permutations outperformed or matched other state-of-the-art methods at the time. Interestingly, the best subnetwork ensemble for each dataset was not consistent, suggesting that different representations may be better at predicting different chemical properties.

WDL-RF is a novel architecture for predicting molecular properties that was trained and tested on a suite of GPCR ligand datasets [134]. The first part of WDL-RF produces learned molecular fingerprints using a neural network. The NN is a (GNN) that shares weights in the network based on the number of bonds attached to each atom node. Each atoms initial feature vector included a one-hot encoding of its element, degree, number of attached hydrogen atoms, implicit valence and aromaticity indicator. This network was trained on GPCR datasets to predict bioactivity. After it was trained, an internal embedding in the network was extracted for each ligand to use as a data-driven fingerprint. GNN-based fingerprints were introduced in other work [129]. The advantage of such fingerprints is that a network must capture molecular properties in order to predict bioactivity, so the embedding of a ligand in the network will capture its chemical information. The neural fingerprints were then used to train a random forest to predict bioactivity. This model outperformed other random forest models that were trained on different molecular fingerprints on a large majority of the GPCR datasets.

GNNs have also found success in predicting other molecular properties. For instance, they have achieved state-of-the-art performance in odor descriptor prediction [135]. Odor perception involves 300-4000 different types of olfactory receptors, which are rhodopsin like (class A) GPCRs [136]. The odor prediction GNN used *atom-node* vectors based on atom type, charge, etc. Similar to WDL-RF, an embedding within the GNN was found to outperform traditional fingerprints and molecular descriptors when used as input to a random forest model.

One issue with the data-driven fingerprints discussed thus far is that they are generated with labeled training data, which restricts the amount of data that can be used to construct a robust fingerprint. The Seq2Seq model attempted

to address this issue by learning a fingerprint from SMILES in an unsupervised manner [137]. This model was an autoencoder that used an GRU-based RNN to encode SMILES into a fixed-length embedding vector and then recover the original SMILES from the embedding. After the model was trained, the embedding vectors for ligands in a labeled dataset were generated to use as data-driven fingerprints in other machine learning models. Experiments showed that random forest, gradient boosting, and SVMs that were trained on the Seq2Seq fingerprints outperformed the same models trained on ECFP and the original neural fingerprint produced by Duvenaud et al. [129]. One additional benefit of fingerprints generated by Seq2Seq is that they are invertible, so it is easy to recover the original SMILES string if given the fingerprint.

Shortly after the publication of the Seq2Seq, researchers developed a variational autoencoder (VAE) for encoding SMILES into a continuous representation in the latent space [138]. this method also produces invertible fingerprints and a regression model can be trained on these embeddings to predict various molecular properties. The use of a VAE is the game changer here as molecules with similar properties are encouraged to be close together in latent space. As a result slightly perturbing the embedding of a ligand and inverting it will tend to give a different but similar ligand. This property is of interest in *de novo* synthesis which is discussed in Section 5.

## 3.2. Similarity-based virtual screening

Similarity-based screening methods are based on the hypothesis that active compounds share similar properties [139]. However, there is no trivial definition of compound similarity as computational methods work with approximate representations. For this reason, similarity-based methods heavily depend on and are defined by the molecular representations they operate on. Most machine learning-based methods for ligand-based VS are clustering algorithms, and recent advances in this research are focused on developing new data representation methods and similarity measures.

Tanimoto similarity is commonly used when comparing molecular fingerprint vectors as it compares favorably to other similarity metrics on molecular fingerprints, which has been confirmed in analyses based on the sum of ranking differences and ANOVA analysis [140]. Fingerprint-similarity-based VS is not a new methodology [141]. While there are more complex 3D approaches, these methods are still used to query large databases due to their computational efficiency. However, Raschka et al. recently showed that 3D overlay-based VS can be made computationally feasible on large databases if hypotheses based on prior information about the ligand-receptor interactions can be used as filtering criteria [26]. Using this hypothesis-driven approach, combined with machine learning-based molecular feature importance assessments, the researchers discovered a potent GPCR signaling inhibitor by screening more than ten million commercially available molecules [26, 60].

A recent example of a Tanimoto-based fingerprint similarity method is MuSSeL [142]. MuSSeL generates an ensemble of different fingerprints for a database of compounds which were labeled with a receptor as their target class and their binding affinity for the corresponding receptor. Given a query compoundâĂŹs fingerprint ensemble, a molecule is considered as a candidate for a given target receptor class only if it meets a minimum Tanimoto similarity cutoff with at least $N_1$ fingerprints in the set of known actives. If this requirement is met, the compound must also pass the following criterion for $N_2$ fingerprints âĂŞ where $N_2$ is a number that has to be specified by the user, similar to choosing $N_1$. First, the compound must have at least $k$ nearest neighbors above a new user-specified Tanimoto cutoff, and second, the maximum difference between two of the $k$ neighborsâĂŹ binding affinities must be below a user-specified threshold. The maximum difference threshold is included to help avoid activity cliffs [143]. For each fingerprint that meets these conditions, the $k$ neighborsâĂŹ binding affinities are averaged. Finally, for each query compound, the $k$-neighbor averages are averaged for all the fingerprints in the ensemble that passed the second filter to predict the binding affinity of the query compound. The algorithm had good performance on a calibration set and correctly labeled compounds in a number of case studies with GPCR targets such as adenosine $A_{2A}$ receptor and dopamine receptor D4.

Common fingerprints like ECFP and MACCS do not explicitly capture 3D information about the compounds. This is problematic in situations where a moleculeâĂŹs 3D structure is valuable for assessing a property, for instance the steric fit of a predicted pose docked into the receptorâĂŹs binding site, so methods for comparing 3D representations of compounds are of practical use. One approach for comparing 3D representations of molecules could be to find an overlay with maximum shape overlap, but this does not utilize electrostatic information. Thus, similarity scores in this space are generally based on a combination of structural overlap and the overlap of groups with like properties. Methods like ROCS [144] and WEGA [145] represent molecules as overlapping Gaussian spheres with each atom sphere containing information about its location, atom type, and charge. The maximum scoring overlay is then based

on both the structural and chemical overlap. Esim is a recent method that computes overlay similarity in a different way [146]. The model uses a set of âĂIJobserver pointsâĂİ in 3D space that are a fixed distance from one target molecule. At each observer point spatial, angular and electrostatic values are computed that capture properties of the target and query molecule from that observer points perspective. The closer the values for each molecule are when summed over all observer points, the higher the similarity score is. This was found to outperform other 3D similarity methods on DUD-E, a common benchmark set that also includes GPCR targets [147].

Methods that rely on 2D fingerprints or 3D overlays for ligand-based VS utilize different information about the molecular structures – each method has individual strengths and weaknesses [148]. The HybridSim-VS method combines both 2D fingerprint and 3D-structural information [149], and the proposed similarity metric is a combination of the Tanimoto similarity of the 3D shape overlay of a pair of molecules and of its 2D fingerprint. This hybrid metric outperformed MACCS, a 2D fingerprint, and WEGA, a 3D representation, when compared on the tested on the DUD-E benchmark dataset [147].

Section 3.1 summarized different neural network embeddings that can be used as inputs to machine learning models to predict various molecular properties. The same embeddings can be used for clustering-based similarity search. One example of this in practice is a clustering-based SPiDER search [150]. SPiDER uses self-organizing maps, a special type of artificial neural network approach for unsupervised learning, to create low-dimensional embeddings of the molecule space based on the CATS topological pharmacophores models and physiochemical small molecule descriptors [151]. Recently SPiDER was used in tandem with a binding affinity prediction via DEcRyPT to discover celastrol, a cannabinoid receptor-1 and 2 agonist [152, 153]. DEcRyPT is based on random forest models for regression analysis, which have been trained to predict binding affinities from topological pharmacophore features. In this study, the researchers first used SPiDER to select potential cannabinoid receptor agonists via clustering. The selected molecules were then analyzed using DEcRyPT, which was trained on experimental binding affinity values for selected targets, as a predictive model that was applied to the most confident predictions from SPiDER to obtain binding affinity scores (-log10 values). The agonist celastrol was identified by this approach, and the ligand was experimentally validated by biochemical assays including radiological displacement assays, followed by dynamic light scattering measurements to rule out false positive readouts. The authors emphasized that common similarity-based search using conventional ECFP4 Morgan-fingerprints would have disqualified celastrol as a GPCR ligand.

## 4. Receptor structure-based methods

### 4.1. Binding site prediction

One of the most notable achievements in binding site prediction for identifying new targets was Nayal and HonigâĂŹs random forest model, which was trained on 408 features representing structural, geometric, and physicochemical properties [154]. Their method was able to identify 1347 cavities on the surface of 99 diverse proteins, and in 100% of the cases, a drug has been experimentally found to bind at least one of these surface cavities.

A more recent method for discovering binding sites in GPCRs is implemented in the COACH-D server [33, 34], which optionally allows users to provide a ligand structure that is then docked into the predicted binding site of the target protein structure to refine the ligand binding site predictions. The prediction in COACH-D is based on the COACH algorithm, which aggregates binding site predictions from both sequence and structure data based on five other methods and then uses a linear SVM classifier for the final prediction [155].

Challenges for applying deep learning-based models to binding site identification are two-fold. First, the dataset of available GPCR structures is relatively small. However, this problem can potentially be addressed by transfer learning, which is discussed in Section 6. Second, deep learning excels when the models can learn to extract patterns from the raw input data rather than hand-engineered feature descriptors as described in Section 3.1. While deep learning models based on 3D voxelization methods [156, 157] and graph convolutions [129, 130, 158] have been successful in predicting binding affinities and molecular properties from small molecule structures, representing large molecules such as membrane receptors is more computationally prohibitive and remains yet to be explored.

### 4.2. Protein-ligand docking

Structure-based virtual screening for GPCRs is limited by the number of high-resolution GPCR structures that are available. However, advancements in techniques like Cryo-EM have increased the rate at which these structures are solved, such that many general AI-based methods for protein-ligand docking can be applied to modeling GPCR-ligand recognition in the future.

Structure-based VS methods need to be able to score protein-ligand complexes to identify favorable candidates. Scoring functions can be grouped into four classes: Force field, empirical, knowledge-based, and machine learning-based [159]. Popular docking programs typically use scoring from the first three classes [29, 160, 161]. Functions from these three classes are sometimes referred to as classical scoring functions. Classical scoring functions are based on linear combinations of features which restricts they ways they can utilize structural and interaction data [162]. Scoring functions based on machine learning are not restricted in this way. One of the first machine learning scoring functions to outperform classical methods is RF-score, which has been updated multiple times to further improve its performance [37, 163, 164]. All iterations of RF-score use the same algorithm for random forest regression. However, the researchers experimented with different ways for encoding the molecular information in the training dataset to improve the performance of RF-score, which underlines the importance of choosing a good input representation as previously discussed in Section 2. The feature representation of the protein-ligand complex used in the latest version of the RF-score (v3) [164] consists of counts of interacting protein-ligand atom pairs. Two atoms are considered interacting if they are within 10 Å in the structure. For example, a carbon in the protein that is 7 Å away from a nitrogen in the ligand in a binding pose would increment the count. These features are not symmetric; atom $A$ in protein atom $B$ in ligand and vice versa are separate features. Additionally, RF-score v3 uses features produced by AutoDock Vina [29]. When trained and tested to model a VS setting RF-score, outperformed classical scoring functions in various evaluation metrics [165].

Initial attempts at developing NN-based scoring functions in this space struggled with overfitting issues and were outperformed by methods based on conventional machine learning methods, like RF-score [166]. Recently, however, a set of new NN-based scoring functions have shown state-of-the-art performance [156, 157, 167, 168]. All these networks use novel input representations to capture the structural and chemical information of the protein-ligand complex.

A number of NN-based scoring functions voxelize the 3D representation of the protein-ligand complex (Section 2.3) and use functions that capture structural and chemical information to compute the channels of each voxel [156, 157, 168]. All of these methods are based on 3D CNNs, which can be computationally demanding due to their large number of trainable model parameters. Furthermore, the large number of parameters increases the number of training examples required to effectively train the networks without overfitting, which has been identified as an issue for DNNs in this space [125]. However, more recent methods aim to address the overfitting issue by using more modern CNN architectures. For instance, DeepAtom [157] was recently designed based on ShuffleNet v2 [169], and the recent $K_{\text{DEEP}}$ [156] method is based on SqueezeNet [170]. These architectures were designed to achieve good computational performance while keeping the number of trainable parameters relatively small, which helps with minimizing overfitting when deeper architectures are being trained on smaller datasets. Both DeepAtom and $K_{\text{DEEP}}$ achieved state-of-the-art binding affinity predictions on the *PDBbind v. 2016* core dataset [171]. The success of these networks suggests that advances in other subfields of deep learning could be effective in VS as well and could be a fruitful area of future research.

Other NN-based scoring functions achieved state-of-the-art performance by including more information in the input representations. OnionNet uses atom pair counts similar to RF-score, but generates them for multiple cutoffs and makes each pair exclusive to the first cutoff it appears in [167]. These values are then converted into a $n \times m$ matrix where $n$ is the cutoff value and $m$ is the atom pair and this matrix is used as input for a 2D CNN. The use of a 2D CNN reduces the parameter burden of this model so the area of the protein that can be measured increases substantially. While the previously mentioned 3D CNN models, DeepAtom and $K_{\text{DEEP}}$ capture molecular information in a 20 Å cube. As described in [167], OnionNet counted at least all atom pairs within a 61 Å cube. Long-range electrostatic interactions extend beyond 20 Å and are known to be important in protein-ligand binding. Hence, OnionNet may be more effective for VS targets where these long-range interactions are important for activity prediction [172]. In practice, its performance on the *PDBbind v. 2016* core was comparable to the 3D CNNs mentioned above, though there was no performance comparison conducted on individual targets where this information could be useful.

## 5. *De novo* **small-molecule design**

*De novo* molecule design is closely aligned with the goals of VS. While VS attempts to answer questions such as "Does this molecule activate this target?", *de novo* design attempts to answer "Can we generate a molecule that can activate this target?". Answering the second question in the affirmative implies that we understand what makes a molecule active against a given target. *De novo* molecule design can be tied to the idea of inverse QSAR/QSPR, deriving the structures of all molecules that are active with a target or have a certain property. This is a challenging

problem but despite its difficulty, *de novo* synthesis has recently become an active area of research, which is likely due to the recent advances in the development of deep generative models and reinforcement learning.

## 5.1. Generating molecules with variational autoencoders

The most commonly used deep generative model in this space is the variational autoencoder (VAE) [111]. In essence, a VAE learns to reconstruct high-dimensional inputs from a low-dimensional continuous latent space. In contrast to a conventional autoencoder (Section 1.7.4), points close in the latent space are encouraged to have reconstructions with similar properties. In *de novo* design, this property can be utilized to generate novel molecules that have similar properties to selected input molecules by taking that inputâĂŹs latent representation and perturbing it in a controlled manner [138, 173]. One of the challenges with using VAE-based models in this space is that with current architectures, the VAE has no notion of what constitutes a valid molecule [174]. Since these models are trained on large numbers of valid compounds, certain models were still able to achieve reasonable results despite producing some invalid molecules [138, 175].

Jin et al. introduced a novel VAE design that only allowed valid molecules as outputs called a junction tree VAE [174]. This model uses a GNN to encode a molecules' graph and also a junction tree of valid molecular fragments in the graph. The decoder network first decodes a junction tree from the latent space. Then, it decodes a molecule that satisfies that junction tree. Since all molecular fragments of the junction tree are valid, a valid molecule can be generated from the fragments given the nature of a junction tree, which forces outputs to be valid molecules[176]. In addition to always producing valid molecules, the model outperformed other VAEs on a Bayesian optimization task, where the goal was to find a molecule that maximized octanol-water partition coefficients (logP).

As a general example, consider the latent representation of a ligand with a thiol group that induces a positive activity response in a given target receptor. Perturbation of the latent representation of that molecule may result in similar molecules that share the thiol group feature but are not active against the target. As a consequence, the performance for VAEs in this space is usually assessed on simple chemical properties [173–175]. For this reason, while there is still plenty to be investigated about VAEs for *de novo* synthesis, research that has focused on the *de novo* generation of compounds for specific targets has largely utilized reinforcement learning, which will be discussed in the next section.

## 5.2. Reinforcement learning-based molecular design

Reinforcement learning seeks to teach an agent to perform actions that will maximize a cumulative reward over a series of iterations. In this paradigm, the definition of the reward is very flexible, which makes it possible to more directly optimize for specific traits compared to using generative models such as VAEs. As illustrated in Figure 5, at each iteration, the agent is given a state by the environment as well as the reward from the previous iteration. The agent then selects an action based on this information. This selected action is fed into the environment, which then outputs a new state and reward. This cycle continues until the series of iterations, called an episode, is terminated at which point the agentâĂŹs behavior is updated based on the information provided by its trajectory, the ordered list of states, action, and rewards associated with each iteration in the episode. While this high-level overview will suffice to understand how reinforcement learning is utilized for *de novo* design, the reader is referred to [48, 177], which we recommend as resources with more in-depth explanations.

Several examples exist in which reinforcement learning models have been optimized to generate GPCR bioactive molecules. One such example is the REINVENT model [38]. In REINVENT, the agent is an RNN that generates one SMILES string token at a time over the course of multiple iterations. After a batch of 128 SMILES strings are produced, the RNN agent is updated with a modified form of REINFORCE, an algorithm that updates the agent network based on the trajectories of the episodes in the current batch [178]. In one experiment, REINVENT was trained to generate compounds that are active with the GPCR Dopamine receptor $D_2$ (DRD2). In this experiment, the reward function was based on the output of an SVM trained to predict a compoundâĂŹs DRD2 activity. After training, the agent was able to produce compounds that the SVM labeled as active against DRD2 96% of the time. It shall be noted that the SVM was highly accurate in predicting the activity of molecules when evaluated on an independent test set (98% prediction accuracy and a high precision of .96). The reinforcement learning model generated both novel, chemically valid compounds and, impressively, compounds that are known to be active with DRD2 even though these were not included in the training set.

DrugEx is another deep reinforcement model that was directly optimized for generating molecules with GPCR activity [179]. Here, the researchers focused on adenosine $A_{2A}$ receptor, which has been targeted to treat cardiovascular and inflammatory diseases [180]. Similar to REINVENT, the DrugEx model also produces SMILES strings over
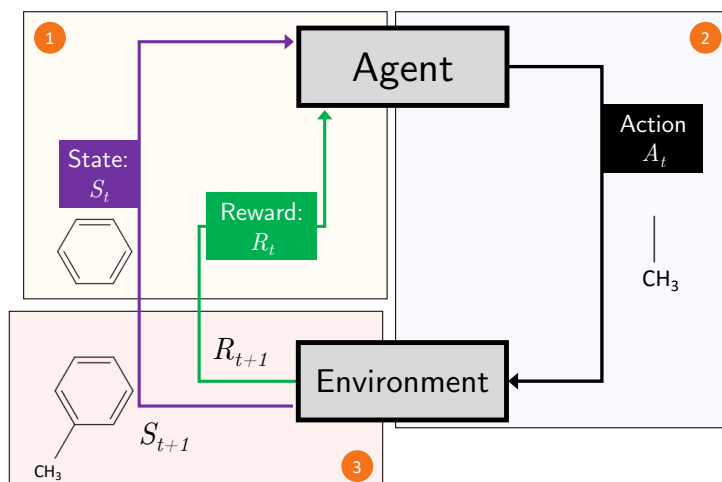
**Figure 5**: Representation of the basic reinforcement learning paradigm with a simple molecular example. (1) Given a benzene ring (state $S_t$ at iteration $t$) and some reward value $R_t$ at iteration $t$, (2) the agent selects an action $A_t$ that adds a methyl group to the benzene ring. (3) The environment considers this information for producing the next state ($S_{t+1}$) and reward ($R_{t+1}$). This cycle repeats until the episode is terminated.

the course of an episode. However, DrugEx adds a stochastic component during training. Before being used in the reinforcement learning training, the RNN agent network was individually trained with a large set of molecule SMILES from ZINC [21]. Two copies of this pre-trained network were then created, with one referred to as the *exploration* network and the other as the *exploitation* network. Only the exploitation network was updated during the reinforcement learning training process; however, with a specified probability at each iteration, the exploration network would be queried for the next token instead. The purpose of this procedure was to explore a wider chemical space during training – afterwards the exploration network was discarded, and only the exploitation network was used to generate new molecules. This method successfully rediscovered some known actives for adenosine $A_{2A}$ receptor. The RNN agent was also able to produce molecules with a large diversity, which was evident by covering all cluster centers with a generated active when using a fingerprint-based clustering.

Other reinforcement learning-based models for *de novo* synthesis described in the literature were not specifically focused on GPCRsâĂŹ bioactive molecule design but could be adopted for such tasks in the future [181, 182]. One such example is Zhou et al.'s Molecule Deep Q-Networks (MOLDQN) approach, which modifies deep Q-networks for molecule generation [181, 183]. This modelâĂŹs agent network takes the current molecular graph's Morgan fingerprint as input and selects an action to modify the molecular graph. The actions include adding an atom, adding a bond, or increasing a bondâĂŹs order. Additionally, the actions that are allowed at a given iteration are restricted if they are invalid so the system will always produce valid molecules. Similar to DrugEx, the model encourages exploration by selecting a random valid action at a given iteration with some probability $\epsilon$. The model achieved state-of-the-art performance when producing molecules that maximized for logP and quantitative estimates of drug likeness separately [184].

While many publications in this area demonstrate the ability to optimize for molecular properties, they unfortunately lack experimental follow-up procedures for further model evaluation. As a notable counter-example, we want to highlight GENTRL, which was used to discover novel inhibitors of discodin domain receptor 1, a tyrosine kinase [14]. The inhibitors were generated computationally and then synthesized and experimentally validated. The experiments *in silico* and *in vitro* were completed in approximately 46 days at a fraction of the cost of a high throughput screening approach.

## 6. Transfer learning

Machine learning, and deep learning in particular, requires large training datasets. It's not atypical for modern deep neural networks to have millions of trainable parameters, which require sufficient data for successful parameterization.

Transfer learning refers to the process of adapting models that have been trained on one task to another, typically similar task [185]. The most common form of transfer learning is to take a DNN that was trained on a large general-purpose dataset and fine-tune it to a smaller dataset of interest. Typically, the more similar the general-purpose dataset is to the target task, the more successful transfer learning can be. Furthermore, the model weights can be frozen and only the last layer replaced and tuned to the target data, or the model parameters of the entire model can be considered starting weights (as opposed to randomly initialized weights) and further fine-tuned.

The advantage of transfer learning is that it is less data- hungry compared to training a model entirely from scratch. This allows researchers to apply DNN models even though the available labeled datasets for the target task are relatively small. One example of this is in a recent structure-based method that that sought to train a 3D CNN to classify molecule activity from high-scoring docking poses [186]. When the network was trained on targets in a DUD-E based training set and then fine-tuned on a smaller protein family-specific dataset, the model performed better on test set targets in that protein family than the same network that was trained only on the family-specific set [147]. Family-specific models typically outperform general models, but data availability may limit their performance [187]. This work shows that transfer learning can help to address this issue. While it has not yet been applied further in bioactivity predictions, transfer learning has been successful in improving the performance of models that predict standard molecular properties like solvency as well as in models that predict quantum mechanical approximations [188, 189].

## 7. Future directions

Currently, one of the most neglected areas of machine learning in bioactive ligand discovery is active learning, which aims to combine artificial and human intelligence. Active learning is a branch of machine learning that focuses on selecting labeled data for supervised learning to improve non-confident predictions and fill the knowledge gaps of the model [190]. Knowledge gaps can be filled through human interaction, by sampling optimal unlabeled training examples for annotations by humans. We think that utilizing domain knowledge from experts is crucial in a highly specialized field such as GPCR bioactive ligand discovery, and we believe that it is highly advantageous to develop machine learning systems that include human feedback loops.

Another avenue that remains largely unexplored for computational ligand discovery is a combination of active learning and transfer learning (Section 6), called active transfer learning. Here, a separate model is trained on an existing validation set to predict whether a given model is able to make correct predictions or not on a given unlabeled dataset, which then can highlight problematic cases for human review [190].

Another recent development and promising research trend in deep learning is semi-supervised learning, which is particularly useful if pre-trained models for transfer learning are not available for the target domain or are infeasible to obtain. Semi-supervised learning is the process of deriving and utilizing label information directly from the data itself rather than having humans annotating it. In a language model, for example, semi-supervised learning can be utilized by training a model to predict the next word in a sequence [191] or, in the context of computer vision, this could be the composition of an image into a jigsaw puzzle that the DNN learns to assemble [192]. The main idea of this approach is to choose a task that requires an understanding of the underlying data in order to be solved. This stage of self-supervised training can be regarded as pre-training, and the model can then be adopted and fine-tuned to solve the target task downstream similar to conventional transfer learning.

## 8. Conclusions

The confluence of the meteoric rise of deep learning methods, the increase in large chemical datasets, and the further ease of access to high powered computing resources have spurred the field of chemical machine learning forward at a rapid pace. The many representations of chemical data allow a variety of methods used in other fields to be put to task on chemical quandaries and have encouraged the development of new machine learning models. As we have discussed, many of these models have been or could be applied to VS on GPCRs, though the types of GPCR data currently available may limit the general applicability of certain models. However, as the amount of bioactivity data available continues to increase, so will the performance of these models. One area in which the space is currently lacking is the experimental validation of the computational method development.

Hopefully, by providing an overview of deep learning methods and detailed explanations of new approaches in the space, we can encourage the more experimentally-minded to utilize methods in this space, or at least to collaborate more confidently. In the other direction, it is important to convey to the methodologically-inclined that while it is great

to develop a new method that performs well on a benchmark, experimental validation is the gold standard, so it is beneficial to seek out experimental collaboration when possible. This will lead to more actionable results and push the development of machine learning VS methods even faster in the future.

# 9. Acknowledgements

# References

[1] Hauser AS, Attwood MM, Rask-Andersen M, Schiöth HB, Gloriam DE. Trends in GPCR drug discovery: new agents, targets and indications. Nature Reviews Drug Discovery. 2017;16(12):829.

[2] Garland SL. Are GPCRs still a source of new targets? Journal of Biomolecular Screening. 2013;18(9):947–966.

[3] Thomsen W, Frazer J, Unett D. Functional assays for screening GPCR targets. Current Opinion in Biotechnology. 2005;16(6):655–665.

[4] Bjarnadóttir TK, Gloriam DE, Hellstrand SH, Kristiansson H, Fredriksson R, Schiöth HB. Comprehensive repertoire and phylogenetic analysis of the G protein-coupled receptors in human and mouse. Genomics. 2006;88(3):263–273.

[5] Davenport AP, Alexander SP, Sharman JL, Pawson AJ, Benson HE, Monaghan AE, et al. International Union of Basic and Clinical Pharmacology. LXXXVIII. G protein-coupled receptor list: recommendations for new pairings with cognate ligands. Pharmacological Reviews. 2013;65(3):967–986.

[6] Raschka S. Automated discovery of GPCR bioactive ligands. Current Opinion in Structural Biology. 2019;55:17–24.

[7] Southan C, Sharman JL, Benson HE, Faccenda E, Pawson AJ, Alexander SP, et al. The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands. Nucleic Acids Research. 2015;44(D1):D1054–D1068.

[8] Wong CH, Siah KW, Lo AW. Estimation of clinical trial success rates and related parameters. Biostatistics. 2019;20(2):273–286.

[9] Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, et al. Applications of machine learning in drug discovery and development. Nature Reviews Drug Discovery. 2019;p. 1.

[10] Food U, Administration D. FDA, editor. What Are Biologics; 2020. Accessed: 2020-01-02. https://www.fda.gov/about-fda/center-biologics-evaluation-and-research-cber/what-are-biologics-questions-and-answers.

[11] Mullard A. 2018 FDA drug approvals. NLM (Medline); 2019.

[12] Rodrigues T, Bernardes GJ. Machine learning for target discovery in drug development. Current Opinion in Chemical Biology. 2020;56:16–22.

[13] Paul SM, Mytelka DS, Dunwiddie CT, Persinger CC, Munos BH, Lindborg SR, et al. How to improve R&D productivity: the pharmaceutical industry's grand challenge. Nature Reviews Drug discovery. 2010;9(3):203.

[14] Zhavoronkov A, Ivanenkov YA, Aliper A, Veselov MS, Aladinskiy VA, Aladinskaya AV, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nature Biotechnology. 2019;37(9):1038–1040.

[15] Parker CG, Galmozzi A, Wang Y, Correia BE, Sasaki K, Joslyn CM, et al. Ligand and target discovery by fragment-based screening in human cells. Cell. 2017;168(3):527–541.

[16] Bar-Peled L, Kemper EK, Suciu RM, Vinogradova EV, Backus KM, Horning BD, et al. Chemical proteomics identifies druggable vulnerabilities in a genetically defined cancer. Cell. 2017;171(3):696–709.

[17] Moellering RE, Cravatt BF. How chemoproteomics can enable drug discovery and development. Chemistry & Biology. 2012;19(1):11–22.

[18] Laraia L, Waldmann H. Natural product inspired compound collections: evolutionary principle, chemical synthesis, phenotypic screening, and target identification. Drug Discovery Today: Technologies. 2017;23:75–82.

[19] Duros V, Grizou J, Xuan W, Hosni Z, Long DL, Miras HN, et al. Human versus robots in the discovery and crystallization of gigantic polyoxometalates. Angewandte Chemie. 2017;129(36):10955–10960.

[20] Häse F, Roch LM, Aspuru-Guzik A. Next-Generation Experimentation with Self-Driving Laboratories. Trends in Chemistry. 2019;1(3):282–291.

[21] Sterling T, Irwin JJ. ZINC 15–ligand discovery for everyone. Journal of Chemical Information and Modeling. 2015;55(11):2324–2337.

[22] Sunseri J, Koes DR. Pharmit: interactive exploration of chemical space. Nucleic Acids Research. 2016;44(W1):W442–W448.

[23] Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, et al. The ChEMBL bioactivity database: an update. Nucleic Acids Research. 2014;42(D1):D1083–D1090.

[24] Basith S, Cui M, Macalino SJ, Park J, Clavio NA, Kang S, et al. Exploring G protein-coupled receptors (GPCRs) ligand space via cheminformatics approaches: impact on rational drug design. Frontiers in Pharmacology. 2018;9:128.

[25] Wheatley M, Wootten D, Conner MT, Simms J, Kendrick R, Logan RT, et al. Lifting the lid on GPCRs: the role of extracellular loops. British Journal of Pharmacology. 2012;165(6):1688–1703.

[26] Raschka S, Scott AM, Liu N, Gunturu S, Huertas M, Li W, et al. Enabling the hypothesis-driven prioritization of ligand candidates in big databases: Screenlamp and its application to GPCR inhibitor discovery for invasive species control. Journal of Computer-aided Molecular Design. 2018;32(3):415–433.

[27] Zhang H, Ericksen SS, Lee Cp, Ananiev GE, Wlodarchak N, Yu P, et al. Predicting kinase inhibitors using bioactivity matrix derived informer sets. PLOS Computational Biology. 2019 aug;15(8):e1006813.

[28] Neudert G, Klebe G. DSX: a knowledge-based scoring function for the assessment of protein–ligand complexes. Journal of Chemical Information and Modeling. 2011;51(10):2731–2745.

[29] Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of Computational Chemistry. 2010;31(2):455–461.

[30] Jeon J, Nim S, Teyra J, Datti A, Wrana JL, Sidhu SS, et al. A systematic approach to identify novel cancer drug targets using machine learning, inhibitor design and high-throughput screening. Genome Medicine. 2014;6(7):57.

[31] Riniker S, Wang Y, Jenkins JL, Landrum GA. Using information from historical high-throughput screens to predict active compounds. Journal of Chemical Information and Modeling. 2014;54(7):1880–1891.

[32] Rifaioglu AS, Atas H, Martin MJ, Cetin-Atalay R, Atalay V, Dogan T. Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. Briefings in Bioinformatics. 2018;10.

[33] Chan HS, Li Y, Dahoun T, Vogel H, Yuan S. New binding sites, new opportunities for GPCR drug discovery. Trends in Biochemical Sciences. 2019;.

[34] Wu Q, Peng Z, Zhang Y, Yang J. COACH-D: improved protein–ligand binding sites prediction with refined ligand-binding poses through molecular docking. Nucleic Acids Research. 2018;46(W1):W438–W442.

[35] Ferrero E, Dunham I, Sanseau P. In silico prediction of novel therapeutic targets using gene–disease association data. Journal of Translational Medicine. 2017;15(1):182.

[36] Farimani AB, Feinberg E, Pande V. Binding pathway of opiates to $\mu$-opioid receptors revealed by machine learning. Biophysical Journal. 2018;114(3):62a–63a.

[37] Ballester PJ, Mitchell JB. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. Bioinformatics. 2010;26(9):1169–1175.

[38] Olivecrona M, Blaschke T, Engkvist O, Chen H. Molecular de-novo design through deep reinforcement learning. Journal of Cheminformatics. 2017;9(1):48.

[39] Kadurin A, Nikolenko S, Khrabrov K, Aliper A, Zhavoronkov A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. Molecular Pharmaceutics. 2017;14(9):3098–3104.

[40] Mamoshina P, Volosnikova M, Ozerov IV, Putin E, Skibina E, Cortese F, et al. Machine learning on human muscle transcriptomic data for biomarker discovery and tissue-specific drug target identification. Frontiers in Genetics. 2018;9:242.

[41] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436.

[42] Canziani A, Paszke A, Culurciello E. An analysis of deep neural network models for practical applications. ArXiv preprint arXiv:160507678. 2016;.

[43] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature. 1986;323(6088):533–536.

[44] Raschka S. Naive Bayes and text classification I–introduction and theory. ArXiv preprint arXiv:14105329. 2014;.

[45] Fisher RA. The use of multiple measurements in taxonomic problems. Annals of Eugenics. 1936;7(2):179–188.

[46] Cao W, Mirjalili V, Raschka S. Rank-consistent ordinal regression for neural networks. ArXiv preprint arXiv:190107884. 2019;.

[47] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 248–255.

[48] Raschka S, Mirjalili V. Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd; 2019.

[49] Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. Computing in Science & Engineering. 2011;13(2):22.

[50] Hunter JD. Matplotlib: A 2D graphics environment. Computing in Science & Engineering. 2007;9(3):90.

[51] McKinney W, et al. Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference. vol. 445. Austin, TX; 2010. p. 51–56.

[52] Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422–1423.

[53] Raschka S. BioPandas: Working with molecular structures in pandas DataFrames. Journal of Open Source Software. 2017;2(14):279.

[54] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825–2830.

[55] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16); 2016. p. 265–283.

[56] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems; 2019. p. 8024–8035.

[57] Koscielny G, An P, Carvalho-Silva D, Cham JA, Fumis L, Gasparyan R, et al. Open Targets: a platform for therapeutic target identification and validation. Nucleic Acids Research. 2016;45(D1):D985–D994.

[58] Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, et al. MoleculeNet: a benchmark for molecular machine learning. Chemical Science. 2018;9(2):513–530.

[59] Gaulton A, Hersey A, Nowotka M, Bento AP, Chambers J, Mendez D, et al. The ChEMBL database in 2017. Nucleic Acids Research. 2016;45(D1):D945–D954.

[60] Raschka S, Scott AM, Huertas M, Li W, Kuhn LA. Automated inference of chemical discriminants of biological activity. In: Computational Drug Discovery and Design. Springer; 2018. p. 307–338.

[61] Raschka S, Wolf AJ, Bemister-Buffington J, Kuhn LA. Protein–ligand interfaces are polarized: discovery of a strong trend for intermolecular hydrogen bonds to favor donors on the protein side with implications for predicting and designing ligand complexes. Journal of Computer-aided Molecular Design. 2018;32(4):511–528.

[62] Raschka S, Bemister-Buffington J, Kuhn LA. Detecting the native ligand orientation by interfacial rigidity: SiteInterlock. Proteins: Structure,

Function, and Bioinformatics. 2016;84(12):1888–1901.

[63] Ribeiro MT, Singh S, Guestrin C. Why should i trust you?: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2016. p. 1135–1144.

[64] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems; 2017. p. 4765–4774.

[65] Springenberg J, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. In: ICLR (workshop track); 2014. p. 1–14.

[66] Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 2921–2929.

[67] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 618–626.

[68] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 3145–3153.

[69] committee N. Reproducibility Checklist;. Accessed: 2020-01-02. https://nips.cc/Conferences/2019.

[70] Burkov A. The hundred-page machine learning book. Andriy Burkov; 2019.

[71] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[72] Csáji BC. Approximation with artificial neural networks. Faculty of Sciences, Etvs Lornd University, Hungary. 2001;24:48.

[73] Cybenko G. Approximations by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems. 1989;2:183–192.

[74] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks. 1989;2(5):359–366.

[75] Schmidhuber J. Deep learning in neural networks: An overview. Neural Networks. 2015;61:85–117.

[76] Ivakhnenko AG, Lapa VG. Cybernetic predicting devices. Purdue University Lafayette and Ind. School of Electrical Engineering; 1966.

[77] Ivakhnenko AG, Lapa VG. Cybernetics and forecasting techniques. North-Holland; 1967.

[78] Linnainmaa S. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ Helsinki. 1970;p. 6–7.

[79] Werbos P. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph D dissertation, Harvard University. 1974;.

[80] Steinkraus D, Buck I, Simard P. Using GPUs for machine learning algorithms. In: 8th International Conference on Document Analysis and Recognition. IEEE; 2005. p. 1115–1120.

[81] Chellapilla K, Puri S, Simard P. High performance convolutional neural networks for document processing. In: In Tenth International Workshop on Frontiers in Handwriting Recognition; 2006. p. 1–7.

[82] Raina R, Madhavan A, Ng AY. Large-scale deep unsupervised learning using graphics processors. In: Proceedings of the 26th International Conference on Machine Learning. ACM; 2009. p. 873–880.

[83] Cireşan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. ArXiv preprint arXiv:12022745. 2012;.

[84] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning; 2010. p. 807–814.

[85] Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv preprint arXiv:14126980. 2014;.

[86] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research. 2014;15(1):1929–1958.

[87] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: International Conference on Machine Learning; 2015. p. 448–456.

[88] Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure–activity relationships. Journal of Chemical Information and Modeling. 2015;55(2):263–274.

[89] LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, et al. Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems; 1990. p. 396–404.

[90] LeCun Y, Bottou L, Bengio Y, Haffner P, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278–2324.

[91] Ning F, Delhomme D, LeCun Y, Piano F, Bottou L, Barbano PE. Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing. 2005;14:1360–1371.

[92] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012. p. 1097–1105.

[93] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:14091556. 2014;.

[94] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 770–778.

[95] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 2818–2826.

[96] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. ArXiv preprint arXiv:170404861. 2017;.

[97] Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. ArXiv preprint arXiv:190511946. 2019;.

[98] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning; 2013. p. 1310–1318.

[99] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997;9(8):1735–1780.

[100] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning; 2015. p. 2342–2350.

[101] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in Neural Information

Processing Systems; 2017. p. 5998–6008.

[102] Merity S. Single headed attention RNN: stop thinking with your head. ArXiv preprint arXiv:191111423. 2019;.

[103] Karpov P, Godin G, Tetko IV. A transformer model for retrosynthesis. In: International Conference on Artificial Neural Networks. Springer; 2019. p. 817–830.

[104] Karpov P, Godin G, Tetko IV. Transformer-CNN: Fast and Reliable tool for QSAR. ArXiv preprint arXiv:191106603. 2019;.

[105] Honda S, Shi S, Ueda HR. SMILES Transformer: Pre-trained molecular fingerprint for low data drug discovery. ArXiv preprint arXiv:191104738. 2019;.

[106] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–507.

[107] Mirjalili V, Raschka S, Namboodiri A, Ross A. Semi-Adversarial Networks: Convolutional autoencoders for imparting privacy to face images. In: Proceedings of 11th IAPR International Conference on Biometrics. Gold Coast, Australia: IEEE; 2018. .

[108] Mirjalili V, Raschka S, Ross A. FlowSAN: privacy-enhancing semi-adversarial networks to confound arbitrary face-based gender classifiers. IEEE Access. 2019;7:99735–99745.

[109] Mirjalili V, Raschka S, Ross A. Gender privacy: An ensemble of semi adversarial networks for confounding arbitrary gender classifiers. In: Proceedings of the 9th International Conference on Biometrics: Theory, Applications and Systems. Los Angeles, USA: IEEE; 2018. .

[110] Mirjalili V, Raschka S, Ross A. PrivacyNet: semi-adversarial networks for multi-attribute face privacy. ArXiv preprint arXiv:200100561. 2020;.

[111] Kingma DP, Welling M. Auto-encoding variational bayes. ArXiv preprint arXiv:13126114. 2013;.

[112] Doersch C. Tutorial on variational autoencoders. ArXiv preprint arXiv:160605908. 2016;.

[113] Todeschini R, Consonni V. Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references. vol. 41. John Wiley & Sons; 2009.

[114] Mauri A, Consonni V, Pavan M, Todeschini R. Dragon software: An easy approach to molecular descriptor calculations. Match. 2006;56(2):237–248.

[115] Moriwaki H, Tian YS, Kawashita N, Takagi T. Mordred: a molecular descriptor calculator. Journal of Cheminformatics. 2018;10(1):4.

[116] Keller A, Gerkin RC, Guan Y, Dhurandhar A, Turu G, Szalai B, et al. Predicting human olfactory perception from chemical features of odor molecules. Science. 2017;355(6327):820–826.

[117] Rogers D, Hahn M. Extended-connectivity fingerprints. Journal of Chemical Information and Modeling. 2010;50(5):742–754.

[118] Axen SD, Huang XP, Caceres EL, Gendelev L, Roth BL, Keiser MJ. A simple representation of three-dimensional molecular structure. Journal of Medicinal Chemistry. 2017;60(17):7393–7409.

[119] Da C, Kireev D. Structural protein–ligand interaction fingerprints (SPLIF) for structure-based virtual screening: method and benchmark study. Journal of Chemical Information and Modeling. 2014;54(9):2555–2561.

[120] Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences. 1988;28(1):31–36.

[121] Hirohara M, Saito Y, Koda Y, Sato K, Sakakibara Y. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. BMC Bioinformatics. 2018;19(19):526.

[122] Huang R, Xia M, Nguyen DT, Zhao T, Sakamuru S, Zhao J, et al. Tox21Challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. Frontiers in Environmental Science. 2016;3:85.

[123] Goh GB, Hodas NO, Siegel C, Vishnu A. SMILES2vec: An interpretable general-purpose deep neural network for predicting chemical properties. ArXiv preprint arXiv:171202034. 2017;.

[124] Delaney JS. ESOL: estimating aqueous solubility directly from molecular structure. Journal of chemical information and computer sciences. 2004;44(3):1000–1005.

[125] Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR. Protein–ligand scoring with convolutional neural networks. Journal of Chemical Information and Modeling. 2017;57(4):942–957.

[126] Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. Journal of Chemical Information and Modeling. 2013;53(8):1893–1904.

[127] Goh GB, Siegel C, Vishnu A, Hodas NO, Baker N. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. ArXiv preprint arXiv:170606689. 2017;.

[128] Meyer JG, Liu S, Miller IJ, Coon JJ, Gitter A. Learning drug function from chemical structure with convolutional neural networks and random forests. BioRxiv. 2019;p. 482877.

[129] Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. In: Advances in neural information processing systems; 2015. p. 2224–2232.

[130] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 1263–1272.

[131] Bian Y, Jing Y, Wang L, Ma S, Jun JJ, Xie XQ. Prediction of orthosteric and allosteric regulations on cannabinoid receptors using supervised machine learning classifiers. Molecular Pharmaceutics. 2019;.

[132] Shen Q, Wang G, Li S, Liu X, Lu S, Chen Z, et al. ASD v3. 0: unraveling allosteric regulation with structural mechanisms and biological networks. Nucleic Acids Research. 2015;44(D1):D527–D535.

[133] Paul A, Jha D, Al-Bahrani R, Liao Wk, Choudhary A, Agrawal A. CheMixNet: Mixed DNN architectures for predicting chemical properties using multiple molecular representations. ArXiv preprint arXiv:181108283. 2018;.

[134] Wu J, Zhang Q, Wu W, Pang T, Hu H, Chan WK, et al. WDL-RF: predicting bioactivities of ligand molecules acting with G protein-coupled receptors by combining weighted deep learning and random forest. Bioinformatics. 2018;34(13):2271–2282.

[135] Sanchez-Lengeling B, Wei JN, Lee BK, Gerkin RC, Aspuru-Guzik A, Wiltschko AB. Machine learning for scent: learning generalizable perceptual representations of small molecules. ArXiv preprint arXiv:191010685. 2019;.

[136] Su CY, Menuz K, Carlson JR. Olfactory perception: receptors, cells, and circuits. Cell. 2009;139(1):45–59.

[137] Xu Z, Wang S, Zhu F, Huang J. Seq2Seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. ACM; 2017. p. 285–294.

[138] Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, et al. Automatic chemical design using a data-driven continuous representation of molecules. ACS central science. 2018;4(2):268–276.

[139] Johnson MA, Maggiora GM. Concepts and applications of molecular similarity. Wiley; 1990.

[140] Bajusz D, Racz A, Heberger K. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics. 2015;7(20).

[141] Willett P. Similarity-based virtual screening using 2D fingerprints. Drug Discovery Today. 2006;11(23-24):1046–1053.

[142] Alberga D, Trisciuzzi D, Montaruli M, Leonetti F, Mangiatordi GF, Nicolotti O. A new approach for drug target and bioactivity prediction: The multifingerprint similarity search algorithm (MuSSeL). Journal of Chemical Information and Modeling. 2018;59(1):586–596.

[143] Stumpfe D, Bajorath J. Exploring activity cliffs in medicinal chemistry: miniperspective. Journal of Medicinal Chemistry. 2012;55(7):2932–2942.

[144] Hawkins PC, Skillman AG, Nicholls A. Comparison of shape-matching and docking as virtual screening tools. Journal of Medicinal Chemistry. 2007;50(1):74–82.

[145] Yan X, Li J, Liu Z, Zheng M, Ge H, Xu J. Enhancing molecular shape comparison by weighted Gaussian functions. Journal of Chemical Information and Modeling. 2013;53(8):1967–1978.

[146] Cleves AE, Johnson SR, Jain AN. Electrostatic-field and surface-shape similarity for virtual screening and pose prediction. Journal of Computer-aided Molecular Design. 2019;33(10):865–886.

[147] Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. Journal of Medicinal Chemistry. 2012;55(14):6582–6594.

[148] Hu G, Kuang G, Xiao W, Li W, Liu G, Tang Y. Performance evaluation of 2D fingerprint and 3D shape similarity methods in virtual screening. Journal of Chemical Information and Modeling. 2012;52(5):1103–1113.

[149] Shang J, Dai X, Li Y, Pistolozzi M, Wang L. HybridSim-VS: a web server for large-scale ligand-based virtual screening using hybrid similarity recognition techniques. Bioinformatics. 2017;33(21):3480–3481.

[150] Reker D, Rodrigues T, Schneider P, Schneider G. Identifying the macromolecular targets of de novo-designed chemical entities through self-organizing map consensus. Proceedings of the National Academy of Sciences. 2014;111(11):4067–4072.

[151] Reutlinger M, Koch CP, Reker D, Todoroff N, Schneider P, Rodrigues T, et al. Chemically advanced template search (CATS) for scaffold-hopping and prospective target prediction for âĂŸorphanâĂŹmolecules. Molecular informatics. 2013;32(2):133–138.

[152] Rodrigues T, Werner M, Roth J, da Cruz EH, Marques MC, Akkapeddi P, et al. Machine intelligence decrypts $\beta$-lapachone as an allosteric 5-lipoxygenase inhibitor. Chemical Science. 2018;9(34):6899–6903.

[153] Rodrigues T, de Almeida BP, Barbosa-Morais NL, Bernardes GJ. Dissecting celastrol with machine learning to unveil dark pharmacology. Chemical Communications. 2019;55(45):6369–6372.

[154] Nayal M, Honig B. On the nature of cavities on protein surfaces: application to the identification of drug-binding sites. Proteins: Structure, Function, and Bioinformatics. 2006;63(4):892–906.

[155] Yang J, Roy A, Zhang Y. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. Bioinformatics. 2013;29(20):2588–2595.

[156] Jiménez J, Skalic M, Martinez-Rosell G, De Fabritiis G. K DEEP: protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. Journal of Chemical Information and Modeling. 2018;58(2):287–296.

[157] Li Y, Rezaei MA, Li C, Li X, Wu D. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. ArXiv preprint arXiv:191200318. 2019;.

[158] Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A. Quantum-chemical insights from deep tensor neural networks. Nature communications. 2017;8:13890.

[159] Li J, Fu A, Zhang L. An overview of scoring functions used for protein–ligand interactions in molecular docking. Interdisciplinary Sciences: Computational Life Sciences. 2019;p. 1–9.

[160] Verdonk ML, Cole JC, Hartshorn MJ, Murray CW, Taylor RD. Improved protein–ligand docking using GOLD. Proteins: Structure, Function, and Bioinformatics. 2003;52(4):609–623.

[161] Vilar S, Cozza G, Moro S. Medicinal chemistry and the molecular operating environment (MOE): application of QSAR and molecular docking to drug discovery. Current Topics in Medicinal Chemistry. 2008;8(18):1555–1572.

[162] Li H, Peng J, Sidorov P, Leung Y, Leung KS, Wong MH, et al. Classical scoring functions for docking are unable to exploit large volumes of structural and interaction data. Bioinformatics. 2019;.

[163] Ballester PJ, Schreyer A, Blundell TL. Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity? Journal of Chemical Information and Modeling. 2014;54(3):944–955.

[164] Li H, Leung KS, Wong MH, Ballester PJ. Improving AutoDock Vina using random forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. Molecular Informatics. 2015;34(2-3):115–126.

[165] Wójcikowski M, Ballester PJ, Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. Scientific Reports. 2017;7:46710.

[166] Sunseri J, Ragoza M, Collins J, Koes DR. A D3R prospective evaluation of machine learning for protein-ligand scoring. Journal of Computer-aided Molecular Design. 2016;30(9):761–771.

[167] Zheng L, Fan J, Mu Y. OnionNet: a multiple-layer inter-molecular contact based convolutional neural network for protein-ligand binding affinity prediction. ACS Omega. 2019 Oct;4(14):15956–15965.

[168] Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. Bioinformatics. 2018;34(21):3666–3674.

[169] Ma N, Zhang X, Zheng HT, Sun J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European

Conference on Computer Vision (ECCV); 2018. p. 116–131.

[170] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. ArXiv preprint arXiv:160207360. 2016;.

[171] Wang R, Fang X, Lu Y, Yang CY, Wang S. The PDBbind database: methodologies and updates. Journal of Medicinal Chemistry. 2005;48(12):4111–4119.

[172] Dagliyan O, Proctor EA, D'Auria KM, Ding F, Dokholyan NV. Structural and dynamic determinants of protein-peptide recognition. Structure. 2011;19(12):1837–1845.

[173] Dai H, Tian Y, Dai B, Skiena S, Song L. Syntax-directed variational autoencoder for structured data. ArXiv preprint arXiv:180208786. 2018;.

[174] Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. ArXiv preprint arXiv:180204364. 2018;.

[175] Kusner MJ, Paige B, Hernández-Lobato JM. Grammar variational autoencoder. In: Proceedings of the 34th International Conference on Machine Learning. JMLR. org; 2017. p. 1945–1954.

[176] Wiegerinck W. Variational approximations between mean field theory and the junction tree algorithm. In: Proceedings of the 16 Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.; 2000. p. 626–633.

[177] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT press; 2018.

[178] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning. 1992;8(3-4):229–256.

[179] Liu X, Ye K, van Vlijmen HW, IJzerman AP, van Westen GJ. An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine A 2A receptor. Journal of Cheminformatics. 2019;11(1):35.

[180] Chen JF, Eltzschig HK, Fredholm BB. Adenosine receptors as drug targetsâĂŤwhat are the challenges? Nature Reviews Drug discovery. 2013;12(4):265.

[181] Zhou Z, Kearnes S, Li L, Zare RN, Riley P. Optimization of molecules via deep reinforcement learning. Scientific Reports. 2019;9(1):1–10.

[182] You J, Liu B, Ying Z, Pande V, Leskovec J. Graph convolutional policy network for goal-directed molecular graph generation. In: Advances in Neural Information Processing Systems; 2018. p. 6410–6421.

[183] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529.

[184] Leeson P. Drug discovery: Chemical beauty contest. Nature. 2012;481(7382):455.

[185] Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. 2009;22(10):1345–1359.

[186] Imrie F, Bradley AR, van der Schaar M, Deane CM. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. Journal of Chemical Information and Modeling. 2018;58(11):2319–2330.

[187] Ross GA, Morris GM, Biggin PC. One size does not fit all: the limits of structure-based models in drug discovery. Journal of chemical theory and computation. 2013;9(9):4266–4274.

[188] Goh GB, Siegel C, Vishnu A, Hodas N. Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM; 2018. p. 302–310.

[189] Smith JS, Nebgen BT, Zubatyuk R, Lubbers N, Devereux C, Barros K, et al. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. Nature Communications. 2019;10(1):1–8.

[190] Munro R. Human-in-the-loop machine learning. Manning; 2020.

[191] Howard J, Ruder S. Universal language model fine-tuning for text classification. ArXiv preprint arXiv:180106146. 2018;.

[192] Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision. Springer; 2016. p. 69–84.