Machine learning methods used during the D3R protein-ligand docking Grand Challenges

Author: Etjen Ahmic, student number 5617030
Supervisor: Dr. Ed Moret
Examiners: Dr. Ed Moret & Dr. Alexandre Bonvin

Layman's summary

When developing novel drugs to treat disease, the first step involves identifying a protein target associated with disease. Following this step, scientists synthesize and test many small molecule compounds to see which bind to and inhibit the function of the protein best. Computer modeling tools are often used to facilitate this process because they can help predict the best binding compound. One such tool is called molecular docking (or just 'docking'). Docking tries to predict the 3D structure of a compound-protein complex from just its individual structures. Recently, applying artificial intelligence (AI) to improve the performance of existing docking tools has gained enormous interest. To assess the effectiveness of docking tools retrospective benchmarks are usually performed. Prospective benchmarking however is a more robust way to assess a docking tool's performance. The D3R grand challenges were a series of community docking competitions organized to assess the performance of docking tools that have been developed throughout the years. In this review, we outline how researchers applied new AI methods in docking during the D3R grand challenges.

Abstract

The D3R Grand Challenges were a series of prospective blinded protein-ligand docking competitions which attracted community-wide participation. The goal of blind challenges is to benchmark existing docking tools without the inherent bias factor that may be accompanied when conducting retrospective benchmarks. In the D3R GCs participants were asked to predict poses and affinities of small molecules binding to a range of pharmaceutically relevant protein targets. In recent years the explosive use of machine learning afforded state-of-the-art

performances in many domains including applications within the biomolecular sciences. This sparked significant interest to apply novel machine learning methods in docking. In this review we highlight machine learning strategies employed during the D3R docking competitions.

Introduction

Proteins form the cornerstone of life and regulate many of the processes that occur within our cells through interactions with other molecules (1,2). Identifying and characterizing these interactions at the atomic level is imperative for modern drug discovery and development. Traditional small molecule drugs, or 'ligands' for short, operate by binding to and modulating the functions of specific target proteins. To achieve favorable binding between a drug and its target, physicochemical and shape complementarity are necessary (2). During the drug discovery phase these properties are optimized for compounds through iterative cycles of chemical modifications followed by *in vitro/vivo* testing, eventually obtaining compounds that possess optimal interaction and safety profiles.

Computational molecular modeling tools have long been used to assist in this process. Molecular docking is an example of such widely employed tools and is used to predict interactions. Docking tries to predict the bound 3D structure of a biomolecular complex from the free unbound components (1,2). Experimental methods such as X-ray crystallography, nuclear magnetic resonance spectroscopy (NMR) or cryo-electron microscopy (cryo-EM) can be used to solve the *exact* 3D structures of complexes. However, these techniques are very laborious, time-consuming and costly. If only the unbound protein 3D structure is available, docking simulations may help elucidate putative interactions between the protein and ligand and predict how its complex will look like. Compounds with predicted non-favorable interactions can be filtered out early in the drug discovery pipeline, focusing experiments only on narrower selections of compounds, ultimately saving time and expenses (3). Multiple types of docking software have been developed throughout the decades including protein-protein,

protein-nucleic acid, protein-peptide and protein-small molecule (ligand). This review focuses exclusively on the latter: protein-ligand docking.

Artificial intelligence (AI) has become a booming research field because of improvements in computer hardware technologies and modeling techniques (3). A subfield in AI called machine learning (ML) is concerned with learning patterns in large datasets and building models that can make accurate predictions (3). There is significant interest in applying ML to improve docking and other molecular modeling methods, because of the abundance of enormous unexploited biological datasets (3). The huge success of AlphaFold (4) in the recent CASP14 protein structure prediction competition is a testament to the utility of ML applied within the biomolecular domain. Application of ML in docking has shown great potential based on recent benchmarks (5,6).

The drug design data resource D3R grand challenges (GC) (7–10) were a series of blinded protein-ligand docking competitions hosted between 2015-2019, which attracted wide participation from both pharmaceutical industry and academia. The goal of such community-oriented challenges is to benchmark the current state-of-the-art tools in the field. In every GC one or two pharmaceutically relevant protein targets were the main focus. Participants were provided ligands known to interact with the target(s), and were expected to predict the binding modes of the ligands, as well as their affinities. Subsequently, the D3R organizers released the experimentally solved co-crystal structures and affinities of the ligands. Participants were ranked based on how well their submitted ligand binding modes and affinities recapitulated experiment. The merit of prospective blinded competitions is the elimination of bias. Normally, computational tools are evaluated retrospectively on public benchmark datasets composed of solved protein-ligand complexes. However, this approach is prone to introducing bias as the

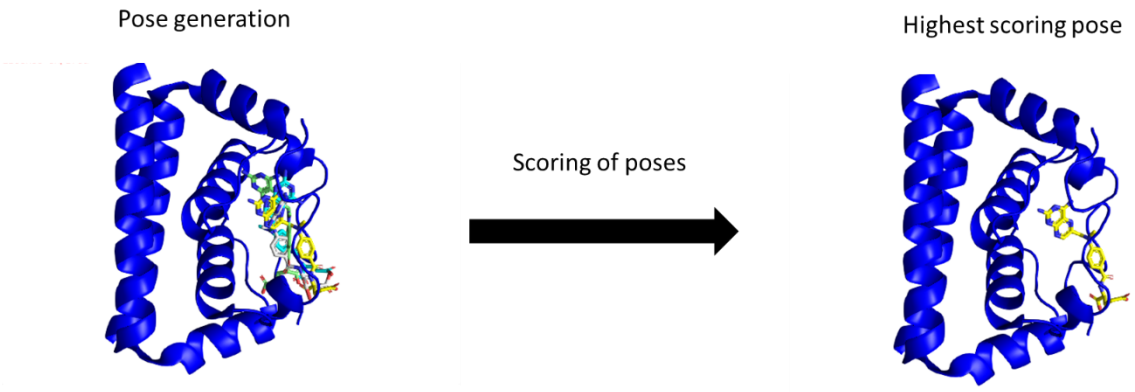

Figure 1 Molecular protein-ligand docking. First, poses of a ligand are sampled with respect to the protein binding site. The generated poses are then ranked by a scoring function which assesses their feasibility.

authors of the developed tool may fine-tune their method to improve performance on the dataset, which is not reflective of how real-life prospective scenarios work like encountered during active drug discovery projects.

In this review, we outline machine learning strategies employed by research groups that participated in the D3R GCs. First, we review the basics of molecular docking and machine learning. Next, we explain the general D3R Grand Challenge format. Finally, we discuss the ML methods used by participants.

Molecular docking

A plethora of excellent reviews on protein-ligand docking (Figure 1) has been published (2,11–14). We will touch on some of the key concepts. The docking problem can be formulated as exploring a roughly shaped energy landscape containing local and global minima that correspond to distinct ligand binding modes in the target protein (Figure 2) (14). The goal is to find the global minimum — the most energetically favorable binding mode. To this end, traditional docking protocols employ two consecutive stages: a conformational sampling stage followed by a scoring stage.

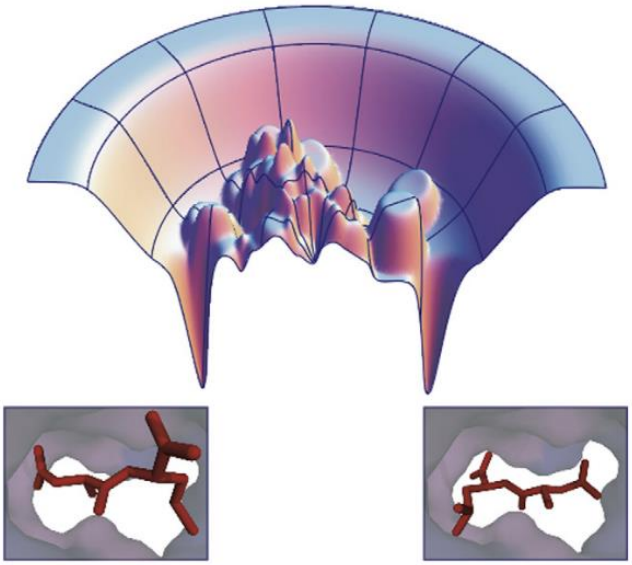

Figure 2 Energy landscape of ligand binding. Multiple local and global minima may exist on the landscape, which correspond to different binding modes of a ligand. Figure ref: Mobley et al. (2009).

Sampling

During sampling the protein and ligand are rotated and translated through space and drawn towards each other, in turn generating multiple different binding modes of the ligand with the protein. The resulting putative binding modes are referred to as "poses". Commonly, both ligand and protein are treated as rigid bodies (11). In reality, molecules in solution are highly flexible because of freely rotatable bonds. However, due to the large degrees of freedom it is computationally infeasible to simulate flexibility for the entire protein. Many docking codes therefore usually only treat the smaller ligand as flexible, keeping the protein rigid. To address the protein flexibility issue, several programs allow residue side chains in close proximity to the ligand to become flexible (11,15–17). Sampling (search) algorithms are divided according to three types: systematic, stochastic and simulation methods (2). Systematic search methods exhaustively sample the landscape of all possible conformations, but as a consequence suffer from combinatorial explosion due to the large degrees of freedom (which may even occur for relatively small molecules). In contrast, stochastic methods apply some type of randomness during sampling. Popular methods include Metropolis Monte Carlo and genetic algorithms, which operate by introducing random conformational, translational and rotational changes to

the ligand. Simulation methods attempt to simulate the dynamics of a system over time. Molecular dynamics (MD) is the most popular atomistic simulation approach and works by solving Newton's equations of motion. Because of the computational intensity of running simulations with full flexibility and buffer molecules, they are typically only used to refine docking poses generated by less time-consuming search methods.

Scoring

The scoring stage subsequently assigns scores to the generated ligand poses on the basis of physicochemical and/or shape complementarities with the protein (2). In a sense they attempt to estimate the binding affinity for a ligand (pose). A good binding mode will be assigned a higher score by the scoring function. To evaluate the goodness of a pose, different types of scoring functions have been devised. They can broadly be divided into four categories: physics-based force field, empirical, knowledge-based and machine learning-based (12). Force field-based scoring functions are the most frequently used type. Force fields describe the potential energy of a system as a summation of bonded and non-bonded interaction energy terms. These terms primarily include van der Waals (vdW) interactions, electrostatic interactions and bonded energy terms involving bond stretching, bond angles and torsions. The non-bonded vdW and electrostatic interaction energies are typically estimated by a Lennard-Jones and Coulombic potential, respectively. Concretely, given a pose in a protein binding site, the sum of all these energy terms is calculated, and the more negative (negative indicates more energetically favorable in molecular binding) the higher the score is for that pose. Empirical scoring functions capitalize on known (3D) structural and activity information from experiments to score poses. They utilize terms or known interactions relevant in molecular recognition (such as hydrogen-bonding, hydrophobic effect, vdW) to reproduce experimental data such as binding affinities. Classically for this purpose simple linear regression models have been used. Binding affinity predictions for newly generated poses can be obtained by deriving the former

mentioned terms from the docked protein-ligand complexes. In essence, they are a simple machine learning model. Knowledge-based scoring functions are also derived from known structural information by performing statistical analysis on commonly occurring atomic contacts in 3D solved complexes from large databases. The presumption here is that particular intermolecular atomic contacts are more likely to occur than others. Machine-learning scoring functions are similar to empirical scoring functions. However, classical empirical scoring functions assume a linear relationship between the terms ("features") and their output. The complexity of biomolecular systems may not lend itself to be modeled by simple linear models. Modern machine learning methods such as random forest, support vector machine, and artificial neural networks have recently been used in an effort to improve existing scoring functions, showing great potential (5,6,12). These techniques are able to incorporate non-linearity, and therefore are able to model more complex relationships. Lastly, consensus scoring functions combine predictions from the aforementioned scoring functions, often improving performance compared to using just a single scoring function for ranking poses (2).

Machine learning in a nutshell

Learning from data

Machine learning (ML) techniques are able to learn patterns in data to build models that make predictions (18). Many of the algorithms and models used in ML are borrowed from the field of statistics (18,19). Broadly speaking, two main types of learning can be distinguished: *supervised* and *unsupervised* learning (18,19). The goal of supervised learning is to estimate a function f that maps some input training data X (also called "*features*/*covariates*/*variables*") to an output Y, in mathematical form: $Y = f(X)$. When Y is continuous it is considered a regression problem (eg, binding affinity prediction). When Y is discrete it is said to be a classification problem (yes/no or probability of yes/no). The function f can take a

predetermined mathematical form, as with linear regression, or a nonpredetermined form. Once estimated, the model is used to predict Y values for new previously unseen values of X. The aim is to build a model that generalizes well enough on a test set, a hold-out dataset which the model was not trained on. Supervised learning is by far the most popular type of learning. In contrast, unsupervised learning is concerned with identifying patterns or groups from data without having available Y labels. Typical applications of unsupervised learning include clustering of similar data points and dimensionality reduction (eg, principal component analysis).

Representing biomolecular structures

When dealing with molecular structures, an efficient representation in the computer is necessary. Machine learning models require the input molecules to be quantized and in vector representation (3,18,19). The better the representation, the easier it is for the algorithm to map the input data to the desired output. Raw atomic 3D coordinates can in principle be fed to learning algorithms by first inserting them in a grid (discussed later in *neural networks*). However, more commonly, machine learning models are trained using molecular descriptors (synonymous: *features/fingerprints*). Molecular descriptors are abstract representations of the physicochemical and structural properties of a compound that are in suitable format for machine learning algorithms to use (3). Many descriptors have been devised throughout the decades, each representing different properties. Not one particular descriptor has been found to substantially outperform another for a given task — usually a combination of different descriptors yields the most predictive models (3). Some examples of relatively simple descriptors include molecular weight, atom type, number of atoms, rotatable bond count and logP/D. Circular fingerprint descriptors describe the topology of a compound and attempt to identify substructures within it by looking at an atom's neighborhood (within a circle of given radius). Furthermore, pharmacophoric fingerprints capture important molecular interaction

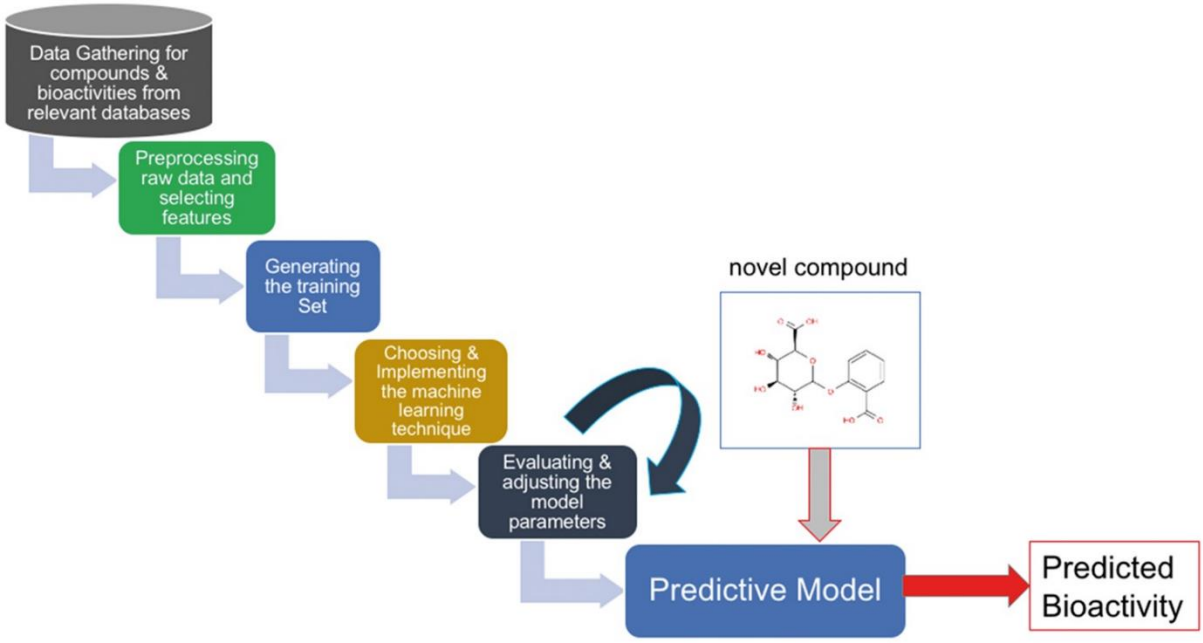

Figure 3 Typical workflow when training machine learning models on biomolecular data. Figure ref: Rifaioglu et al. (2019)

features, such as the presence of hydrogen bond donor/acceptor groups, hydrophobic/philic atoms, aromaticity, and charge within a compound.

The trend in docking recently has been to use molecular descriptors derived from experimentally solved or docked structures to train complex machine learning-based scoring functions that aim to more accurately predict binding affinities and re-score docking poses (3,5,6,12). A typical workflow for training ML models using aforementioned molecular features is shown in Figure 3.

D3R Grand challenges – challenge setup, datasets & evaluation metrics

General challenge setup

The D3R Grand challenges (GC) 1-4 provided an opportunity for the docking community to evaluate their computational methods in a blinded prospective fashion (7–10). In general, the setup for these challenges followed a similar pattern throughout all of the GC competitions. Namely, there was a pose prediction challenge and an affinity prediction (ranking) challenge.

First, candidates were given a set of ligands and the FASTA sequence of a pharmaceutically relevant protein target. For a subset of the ligands participants had to predict the binding poses for the target(s). In addition, participants were asked to rank all compounds according to their predicted affinities. Subsequently, the co-crystal structures were released for the subset of ligands. Participants were then asked to re-rank binding affinities of all compounds, but now with knowledge of the exact binding modes of the subset of ligands for which co-crystal structures were released. The premise being that knowledge of the exact poses should confer an advantage in re-ranking and improve predictions. Additionally, some smaller sub-challenges were devised specifically to test alchemical free energy methods to predict relative binding affinities, for which the organizers provided multiple sets of congeneric ligands. Participants were allowed to use existing structural and bioactivity data to aid in their predictions. Since only the protein sequences of the targets were provided, most participant utilized prior experimentally solved structures of the target co-crystalized with other known ligands (templates). The protein conformation to be selected for docking was often based on the similarity between the co-crystallized ligand and the challenge ligand.

D3R protein targets

The targets of focus in GC1 were the heat shock protein 90 (HSP90) and mitogen activated protein kinase 4 (MAP4K4). HSP90 and MAP4K4 are pharmaceutically interesting targets because of their involvement in various pathological conditions including cancer (7). Pose predictions as well as affinity predictions were sought for ligands of both targets. The GC2 target was the farnesoid nuclear X receptor (FXR). FXR is a bile acid receptor which is highly expressed in the liver, kidney and intestines. It regulates glucose, fatty acid and cholesterol homeostasis (8). FXR agonists were recently identified as promising potential therapeutics for cardiovascular diseases and diabetes (8). Cathepsin S, a cysteine protease, was the main target of GC3. Its implication in various pathological conditions (eg, cancer, diabetes) is well-

established (9). Additionally, datasets for various kinases (JAK2, p38-α, TIE2, VEGFR2 and ABL1) were provided solely for affinity ranking prediction. Finally, in GC4 the targets were beta-secretase-1 (BACE1) and again Cathepsin S (affinity ranking only) (10). BACE1's potential involvement in Alzheimer's disease makes it an attractive target (10).

Evaluation metrics

Several metrics were employed to evaluate the performance of the methods used. The primary metrics were the root-mean-squared-deviation (RMSD) and Kendall's tau (τ) ranking correlation coefficient, for pose prediction and affinity ranking respectively. The RMSD is a measure of fidelity of the docked pose with respect to the pose observed in the experimentally solved co-crystal (native pose). It is computed by superposing a docked model on its corresponding solved co-crystal. Subsequently, the heavy-atom distances between ligand docking and native pose are measured. Docking poses with RMSD < 2 Å were determined to be of acceptable quality. In the affinity ranking task the predicted affinities for the challenge compounds are ranked according to magnitude and compared to the experimentally determined affinity rankings. Kendall's τ ranges from -1 to 1, more positive values indicating better correspondence between predicted and experimental ranking.

Machine learning methods used during D3R

Here we discuss the ML methods used by participants throughout all of the GCs. We outline the types of models used, their inputs, and their outputs. We classify them according to the type of model used. Approaches varied between pure ligand-based methods, structure-based methods or a combination of both. Ligand-based methods do not require a protein-ligand complex, using only ligand properties to make predictions. Structure-based methods on the other hand require a protein-ligand complex, either obtained through docking or template alignment of ligands. Supplementary Table 1 provides a summary of all methods.

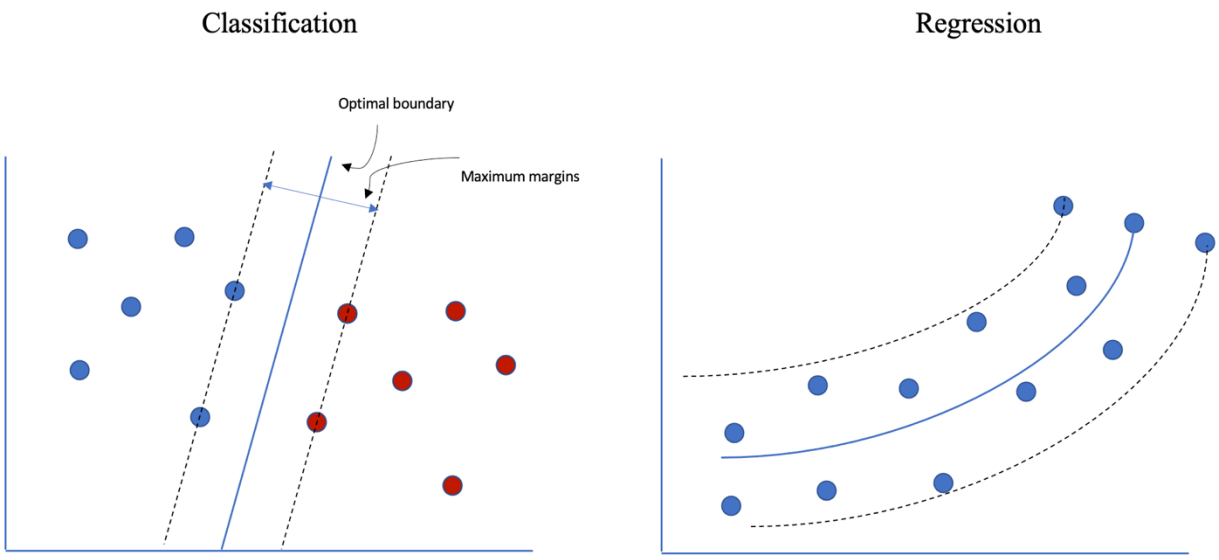

Figure 4 Support vector machine. a) support vector classification separates the training data into two spaces by a linear decision boundary, maximizing the margin between data points of different classes. b) support vector regression fits a regression curve to the data. The kernel trick has been applied in this case to fit a non-linear curve to the data.

Support vector machine

The support vector machine (SVM) is a popular method used in binary classification (19). It is also called a maximal margin classifier, because it maximizes the margin for a linear boundary that separates two classes of data points (Figure 4a). The data points that fall on the margins are called the support vectors. Removing any other data point will not lead to a shift in the linear boundary. When the data cannot be separated well enough by a linear boundary, something called the "kernel trick" may be applied. In essence, this technique projects the data in a higher dimensional space, allowing for more complex non-linear separating boundaries to be learned. A less well-known use case is that SVMs can be extended to regression (SVR). The concept is similar, except the margins now encompass all the data points (Figure 4b). The same kernel trick can be applied in regression to incorporate non-linearity. Because outliers are common in data, soft margins are usually applied to permit misclassifications, or in regression, to allow observations to fall outside the margins.

Two groups used SVM to train ligand-based predictors for ranking binding affinities in GC2 and GC3 (20–22). The underlying idea of ligand-based methods is that similar compounds binding to the same target should also have similar affinities (21). Both groups collected affinity data for compounds known to bind the D3R targets from the CHEMBL and BindingDB databases, and subsequently trained SVR models to predict the affinities of challenge ligands. Baumgartner (20) tested different combinations of molecular descriptors including atom-pair similarity and circular fingerprints, selecting those that gave the best performance on the training set. Bonvin's group [19, 20] exclusively used the atom-pair similarity fingerprint to construct a pairwise similarity matrix from the training compounds. To predict affinities of challenge ligands, similarities between challenge and training ligands were computed and fed to the SVR. While the regressors from both groups performed relatively poorly in predicting affinities of FXR ligands in GC2, the SVR model developed by the Bonvin group was among the top performing methods when ranking affinity of Cathepsin S ligands in GC3 ($\tau = 0.36$, 2nd place), highlighting the power of simple ligand-based quantitative structure-activity relationship (QSAR) approaches.

Random forest and boosting

Random forest (RF) is an 'out-of-the-box' learner that performs well for many tasks (19). RF is based on decision trees (DT), which are intuitively one of the easiest to grasp ML models. DTs resemble flowchart diagrams comprised of a root node, leaf nodes and branches (Figure 5). Each split at a level in the tree represents a partition in the data based on a criterium. Observations that do not meet the criterium (particular value for a feature) are sent to the left,

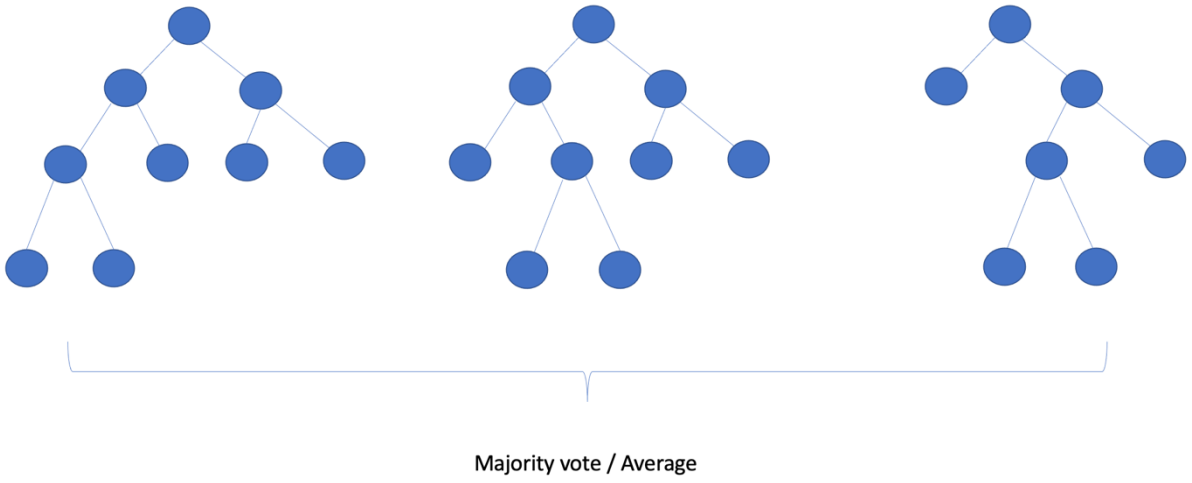

Figure 5 Random forest. Random forest combines predictions from multiple decision trees. To construct multiple trees in RF, a technique called bootstrapping is performed. Bootstrapping involves resampling the training data with replacement. Each tree is learned with a different bootstrap sample, and the results are subsequently aggregated.

and those that do are sent to the right. To make predictions for new data, the tree is traversed from top to bottom until a leaf node is reached. The majority class in the leaf node will be the predicted output in classification, or the average in the case of regression. RF combines multiple trees to make predictions. Combining multiple trees has the effect of reducing the chance to overfit the training data, since single DTs are very prone to overfitting ('high variance'). For classification the majority vote of multiple trees is taken, and for regression the output of multiple trees is averaged.

Similar to the groups that used SVM, the groups that used RF models applied them in the affinity ranking challenges. Both Martins (23) and Prathipati (24) used RF to rank affinities of HSP90 ligands in GC1. Martins used the well-known pre-trained RF-score model (25). RF-score uses protein-ligand atom pairwise interaction counts as features. PocketScore was used to generate poses which were subsequently rescored by RF-score. Prathipati trained a ligand-based RF regressor using only circular fingerprints extracted from HSP90 ligands in the BindingDB database. Challenge ligands were aligned to the most similar ligands from solved co-crystals, minimized by Vina to remove clashes, and re-scored by the RF regressor. Both constants achieved 3rd place in ranking affinity: Prathipati before the HSP90 ligand subset co-

crystal structures were released and Martins after the release of the co-crystal structures. Totrov (26) employed a template similarity-guided docking approach followed by re-scoring with RF for the Cathepsin S ligands in GC3, ranking 1st in the competition (τ = 0.39). Cathepsin S ligands with known affinities were retrieved from the CHEMBL database and docked to representative receptors from the protein data bank. Physics-based terms from the ICM scoring function were then extracted from the docked complexes and used as features to train the RF model.

Yang (27) used extreme gradient boosting (XGBoost) to predict affinities of Cathepsin S ligands in GC4. Boosting is similar to RF and typically DTs are used as the base learners as well. The main difference between RF and boosting is that RF trains independent trees while in boosting trees are trained sequentially. Affinities of Cathepsin S ligands were retrieved from the PDBbind dataset and CHEMBL. The training ligands were subsequently docked using Vina, and interaction terms from Vina's scoring function were extracted for each ligand. Additionally, ligands were cut into fragments, and for each fragment several descriptors were computed (eg, no. heavy atoms, hydrophobic atoms, rotatable bonds). The Vina interactions terms combined with fragment descriptors were then used as features to train the model. Challenge ligands were aligned to the most similar co-crystal ligand and re-scored by the model. The achieved performance was relatively poor compared to the top results achieved by other contestants in the same challenge.

Neural networks (deep learning)

The recent development of advanced neural network (NN) architectures, combined with improved computer speed and hardware technologies (such as GPUs), led to the surge of applying deep NNs in various domains, which has yielded significant state-of-the-art performances in fields like image recognition and natural language processing. As a result,

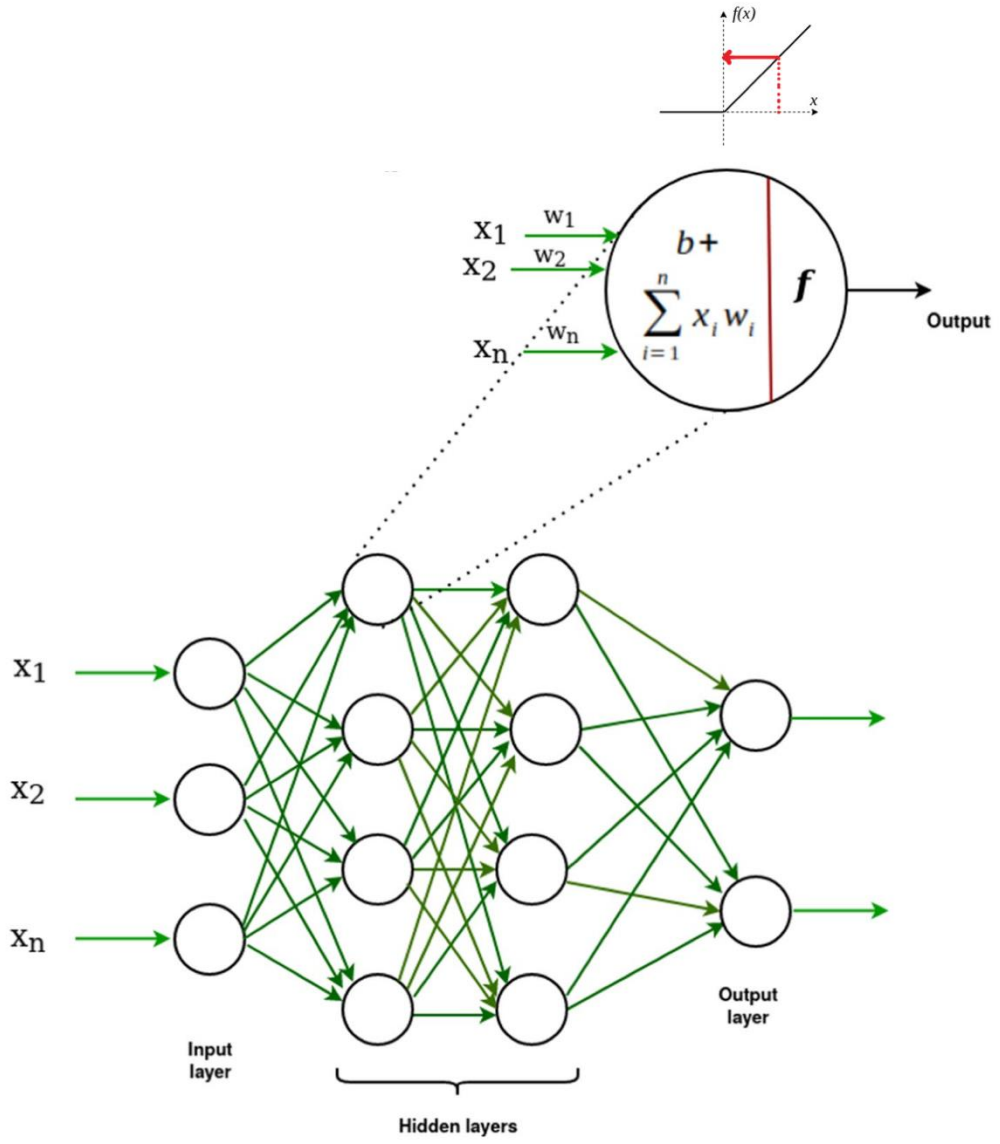

Figure 6 Artificial neural network (NN). NNs are composed of an input layer, multiple hidden layers, and an output layer. A neuron in a particular layer receives input from neurons in the previous layer. In the example, a sum of the weighted inputs is passed to a neuron in the hidden layer. A transformation is applied to this input through an activation function, and the resulting signal is propagated forwards to neurons in the next layer. Commonly used activation functions are ReLU and sigmoid. The weights (w) of the network are learned through a process called backpropagation. Figure ref: Sunny et al. (2022) doi: 10.1007/s10930-021-10031-8

application of these modern architectures in drug design is gaining significant attention, so far having found uses in scoring, binding affinity prediction, pocket detection, virtual screening and protein folding (3,4,13). The inspiration for artificial NN models came from how neurons in the brain function. NNs are composed of three types of fully-connected layers: the input layer, which takes the features as input, one or more hidden layers, and an output layer (Figure

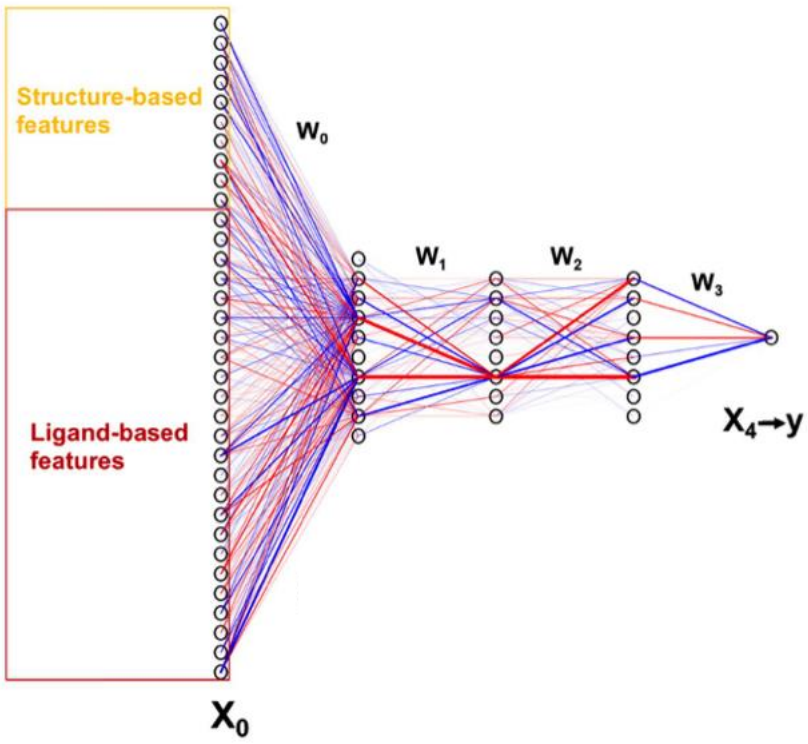

Figure 7 MLP architecture used by the Wang group in GC4 . The neural network takes 10 structure-based features from Vina and 24 ligand-based features to predict binding affinity. The hidden layers are composed of 10x8x8 neurons. ReLU was used as the activation function.

6). Neurons in the hidden layers receive input signals from neurons in the previous layer. A transformation is applied to these input signals through an activation function, and the resulting signal is propagated forwards to neurons in the next layer. The connections in the network have weights which represent the strength of each connection. Learning involves estimating the optimal values for the weights from the training data. The output layer may comprise several neurons for classification, or a single neuron for regression. NN architectures with more than one hidden layer are called multilayer perceptrons (MLP).

Wang (28) trained a simple MLP network using structure and ligand-based features to predict affinity ranking of BACE ligands in GC4. The network architecture is shown in Figure 7. BACE training ligand affinity and structural data were collected from the PDBbind 2017 dataset. Ten structure-based features from Vina's scoring function were then derived for the solved protein-ligand complexes in the training set, in addition to twenty-four ligand-based

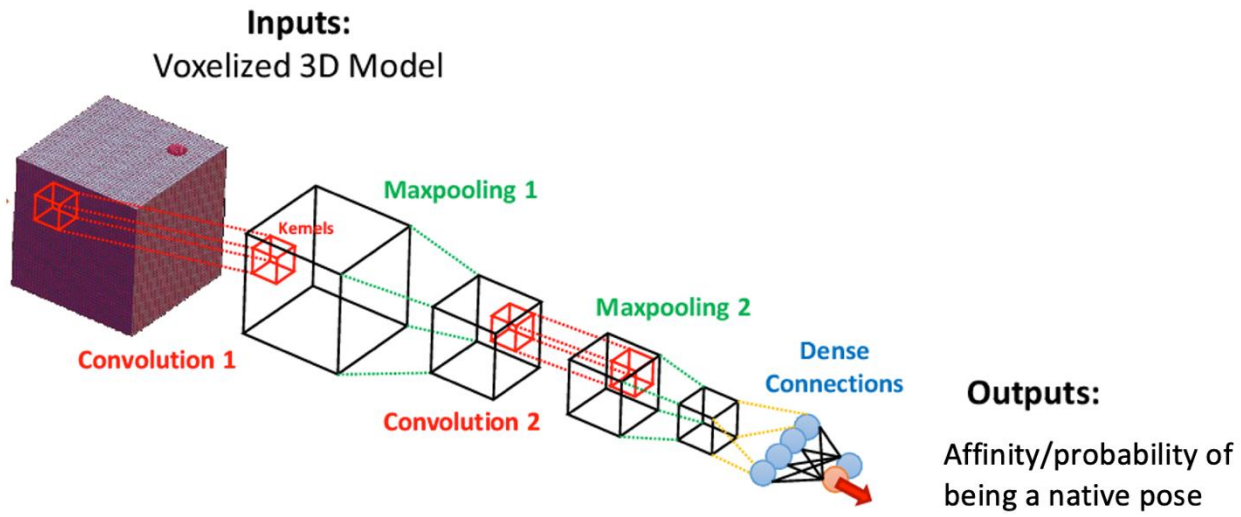

Figure 8 Three-dimensional convolutional neural network (CNN). CNNs consist of sequential convolution and pooling layers, followed by a densely connected NN layer. Convolution applies a filter (kernel) to the input data which moves over the input data, extracting local subfeatures. Pooling layers then reduce the dimensionality of these features. The reduced data is fed to a fully-connected NN layer(s) which outputs predictions. Figure ref: Balu et al. (2016) https://doi.org/10.48550/arXiv.1612.02141

descriptors for each training ligand. Challenge ligands were docked using Vina and binding affinities were predicted by extracting aforementioned features and feeding them to the NN. A post-hoc feature analysis showed that mostly ligand-based features contributed to the predicted affinities, with rotatable bond count and logP having the highest importance. Despite only achieving 5th place in ranking the ligands (τ = 0.30), the performance is impressive for a relatively simple small NN. Similarly, Evangelidis (29) used ligand-based features to train its model deepScaffOpt [unpublished] that too uses MLPs. The model achieved 1st place ranking affinities of Cathepsin S ligands in GC4 (τ = 0.54). The ligand-based features were comprised of circular and 2D pharmacophore fingerprints.

More advanced NN architectures were used by several participants, the most popular of which was the convolutional neural network (CNN, Figure 8). CNNs have become widely used due to their state-of-the-art performance, particularly in image recognition (3). CNNs

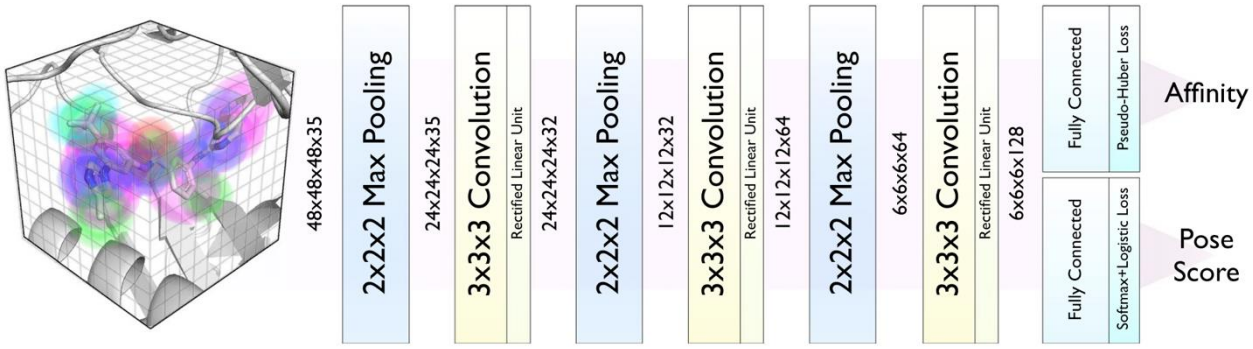

Figure 9. CNN architecture used by the Koes group. The output pose score represents the probability of the input pose being near-native (<2 Å RMSD) .

extract local (spatial) features from the input data by applying series of convolution and pooling operations on the data. In the convolution layers a filter (kernel) slides over the input data and extracts local subfeatures. The pooling layers then subsample and reduce the dimensionality of these features. After numerous sequential convolutions and poolings the data is fed to a fully-connected NN which makes a prediction. The ability to extract local spatial features makes this type of network architecture particularly suitable for 3D biomolecular structural data, because atomic coordinates can be directly used as input. Koes' group (30,31) was the first to apply CNNs in D3R for re-scoring and affinity prediction of docked poses. The input to the model constitutes a voxelized representation of a protein-ligand complex (Figure 9). Atoms are represented as a Gaussian density function; the closer a voxel is to the atom center, the higher its assigned density value. Furthermore, each voxel is 'colored' according to the chemical properties of the ligand atoms (eg, hydrogen bond donor/acceptor, hydrophobic/philic). The network was trained with poses generated by redocking ligands from the PDBbind 2016 refined set using Vina. Poses that had < 2 Å RMSD were labeled as active (native pose) and those above as inactive. Actives were also assigned the experimental affinity values. For the challenge ligands the CNN assigned generated poses after docking with Vina as either native or as non-native (classification) and predicted their affinities (regression). The performance of

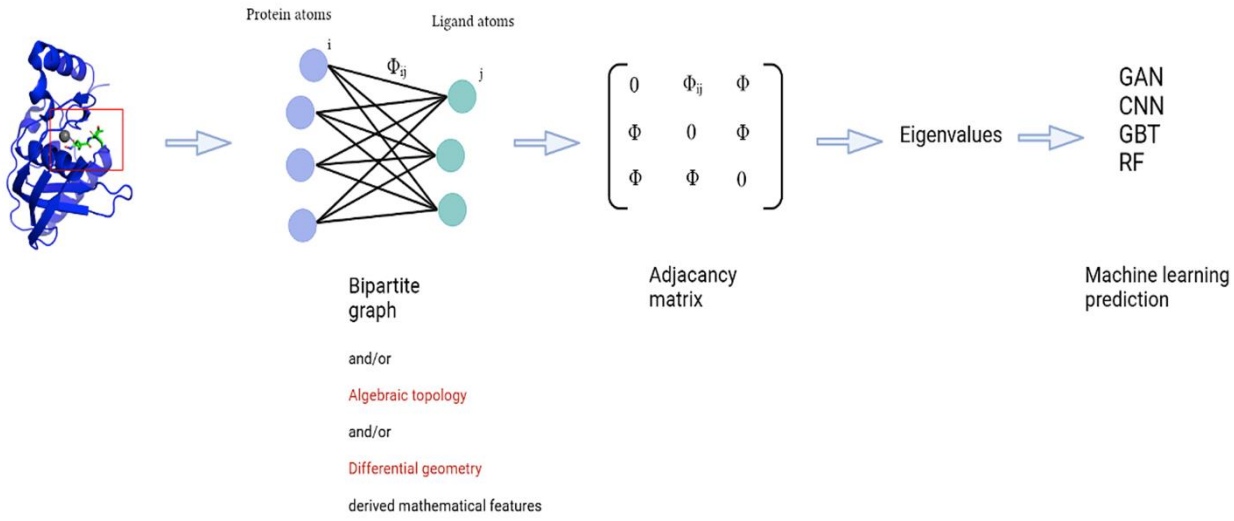

Figure 10 Mathematical machine learning workflow used by Guo-Wei's group. Mathematical features from protein-ligand complexes are extracted and fed to various machine learning models. In the figure, the graph-based representation of molecules is depicted. A bipartite graph is generated, with nodes representing protein and ligand atoms, and edges representing non-covalent interactions. The strength of the edges is determined by a kernel function ϕ, which is a function of distance between two atoms. An adjacency matrix is constructed from the graph from which the eigenvalues (and their statistics) are derived and used as features to train different ML models. Eigenvectors/values are frequently used in dimensionality reduction techniques because they represent a 'compressed' version of the data. Acronyms: GAN, generative adversarial network. CNN, convolutional neural network. GBT, gradient boosted trees. RF, random forest.

the trained CNN model was generally poor compared to submissions by other groups throughout most D3R challenges. However, in GC3 the model achieved top performing affinity ranking prediction for the JAK SC2 kinase sub-challenge, substantially outperforming the 2nd placed method (τ = 0.55 vs 0.36). Yakovenko (32) trained a CNN model to predict poses of FXR ligands in GC2, which placed 2nd overall. A unique approach was taken by combining CNNs with MD. A CNN was trained that takes voxelized docking poses as input, and as output produces poses that would result as if the input complex were run through a hypothetical MD simulation. The rationale behind the approach is grounded in MD's exactness. MD provides a good estimation of entropy (major factor involved in binding) which most docking methods do not take into account. A NN may be able to learn relationships between complexed molecules and their behavior over time, and potentially learn the physical laws and other rules that govern binding. To train the model, a set of twelve resolved complexes were selected initially for which short MD simulations were ran. The resolved complexes as well as the structures obtained from

the last frames of the MD runs were fed to a CNN. The CNN attempts to recreate the MD structures from the input complexes. The network architecture is slightly different compared to Koes' with first a convolution (encoder) layer, followed by 9 densely connected hidden layers in the middle (~500.000 neurons total) and last a deconvolution (decoder) layer which generates the 3D voxel MD output. The challenge ligand poses generated by Vina were used as input to the model, and the most energetically favorable MD poses generated by the CNN were eventually selected. This strategy may find useful applications in virtual screening or as a pose refinement method. Rather than performing long and expensive MD simulations, a NN could quickly predict the MD outcome for a large library of compounds, despite being trained on just a subset of compounds. Lastly, two other groups used CNNs, which include Guo-Wei's group (33,34) and Rial (K_{deep}) (35). Rial used the K_{deep} webserver to predict binding affinities of BACE ligands in GC4 and managed to achieve 1st place (τ = 0.39). Similar to Koes' and Yakovenko's method, a voxelized colored representation is constructed from the input structures. In contrast, Guo-Wei's group represented molecular structures as low-dimensional mathematical features and trained a CNN to predict affinities. These mathematical features are based on graph theory, differential geometry and algebraic topology. Figure 10 shows the general workflow of their approach. The model achieved 3rd place in ranking affinity of Cathepsin S ligands in GC4 (τ = 0.53).

In addition to CNNs, Guo-Wei's group (34) devised a unique approach that uses a generative adversarial network (GAN) to predict binding poses of BACE ligands in GC4. GANs are a new type of generative model pioneered by Goodfellow et al. in 2014 (36). Generative models take a collection of training data and attempt to learn the probability

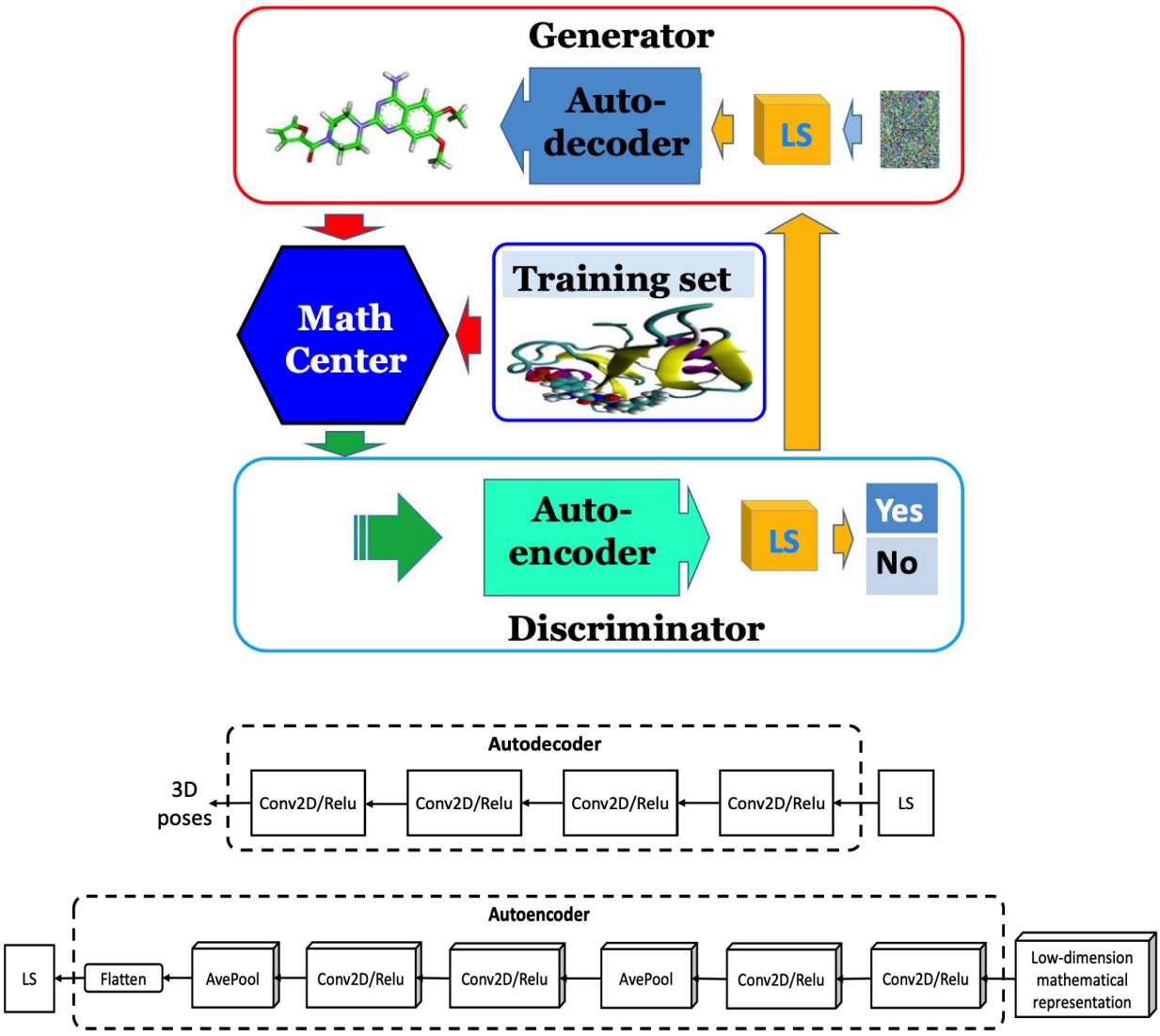

Figure 11 Generative adversarial network approach by Guo-Wei's group. A generator neural network generates fake 3D poses. The discriminator network is provided real poses from training data as well as fake poses generated by the generator. It subsequently tries to discern which of the poses are real and which are fake. At some point the discriminator cannot distinguish real from fake data anymore and the training converges. Poses generated with docking/template alignment can be fed to the discriminator network to determine if they are near-native ('real') or not.

distribution of the data generating process (36). After learning the distribution, new (fake) data points can be generated by the model. GANs bare similarities with autoencoders. Autoencoders try to reconstruct the input data at the output through layers of encoding and decoding neurons. The encoder layer takes input data and reduces its dimensionality to a latent space representation — a compressed version of the data. The decoder layer then tries to recreate the original input data from the latent space. In GANs, two NNs called the 'generator' and 'discriminator' play a game with each other. The generator generates fake data from a latent

space. The discriminator is provided real training data as well as the fake data generated by the generator. It subsequently attempts to discern which are the real and which are the fake data (classification). From the discriminator's feedback the generator iteratively tries to improve its ability to fool the discriminator. At some point a balance (Nash equilibrium) will be reached where the discriminator cannot distinguish real from fake data anymore. The GAN architecture employed by Guo-Wei's group to predict poses is depicted in Figure 11. It consists of a generator (decoder) and discriminator (encoder) network with a math center at its core used to represent the molecules (see also Figure 10). The discriminator encodes the poses from the training set (PDBbind 2018) and fake poses generated by the generator to a latent space and assesses their quality. To predict near-native poses for the challenge ligands, the ligands were first overlayed to the most similar ligand from known BACE co-crystal structures. The candidate poses were then ranked by the discriminator which outputs if a pose is real or fake. This GAN ranking approach was able to secure 1st place in GC4, achieving a median RMSD of 0.55 Å.

AutoML

Automated machine learning (AutoML) is an emerging subfield in ML (37). The goal of AutoML is automate the entire ML pipeline, from hyperparameter optimization (eg, no. trees to use in RF) to model and feature selection. Stroganov (38) used AutoML to predict affinities of Cathepsin S ligands in GC4. A stacked ensemble model was trained that included linear models, random forest, boosted trees and MLP. Cathepsin S ligands with known affinities were collected and blind docked in a representative receptor structure. Energy contributions from the Lead Finder docking program were then extracted from the complexes and used as features. The performance of the model was not among the top (τ = 0.38) and it was outperformed by the previously mentioned deepScaffOpt method.

Concluding remarks

Blind challenges like D3R remain the most robust way to evaluate computational docking tools. We reviewed machine learning methods that participants used during D3R to predict poses and affinities of ligands targeting pharmaceutically relevant targets. Pure ligand-based features that only take into account ligand characteristics as well as structure-based features which use docking, or a combination of both, were employed to train different ML methods. Deep neural network architectures were explored by several groups such as convolutional neural networks and generative models, which managed to rank among the top performers in multiple challenges.

The choice of which learner to use (and its hyperparameter values) is still usually a process of trial and error. No perfect model exists for every dataset. However, some models like random forest and support vector machine perform well all-around out-of-the-box, often not requiring much data. In contrast, training accurate neural networks requires large amounts of data and long training times. A caveat with the models discussed here is that compared to simple linear regression they are effectively a blackbox, in particular AutoML methods which stack multiple models on top of each other. To address the blackbox issue, a novel field of explainable AI has recently started to emerge. For the sake of brevity, we did not discuss linear regression methods here. Some groups applied commonly used ML techniques with linear regression to select important features. Such techniques included coefficient shrinkage (regularization) approaches to reduce dimensionality, resulting in simpler and more explainable predictive models. However, due the inability of linear regression to incorporate enough flexibility it may not be suited to accurately model the complexities of biomolecular systems.

Many challenges and shortcomings still persist in docking that need to be addressed in the future. While most docking tools can generate relatively accurate models given enough

prior information about the target system, they tend to perform poorly when no information is available. This may partly be attributed to the difficulty of modeling induced-fit effects in which a ligand induces conformational changes in the protein upon binding. A common strategy in D3R was therefore to use known receptor conformations retrieved from the protein data bank. Challenge ligands were frequently aligned to the most similar template ligand followed by energy minimization, or the template ligand binding site was used to restrict the docking space. Indeed, this approach showed superior accuracy over blind (ensemble) docking in D3R. In addition, many improvements to existing scoring functions/affinity predictions can still be realized, as evidenced by a simple null model placing 1st in GC2 which ranked the FXR ligand affinities only according to estimated logP. It is known that force fields suffer from inaccuracies and most force fields still do not take into account solvation, entropic or receptor environment induced polarization effects.

After the success of AlphaFold we expect to see more ML applications being developed by the docking community that will substantially improve upon the current state-of-the-art. Integrative modeling (1,39), which incorporates experimental data to drive the docking process, undoubtedly will play a larger role in the future as more high quality omics data become available. Integrating these data with ML algorithms could provide valuable insights and may be the next step in advancing the field.

Supplementing material

See next page for Supplementary Table 1.

Group & Algorithm	Type	Input	Output	Noteworthy performance	Ref
Support vector machine					
Baumgartner	Ligand-based	Multiple fingerprints including atom-pair and circular	Binding affinity	—	(20)
Bonvin	Ligand-based	Atom-pair fingerprint	Binding affinity	2nd place affinity prediction Cathepsin S ligands GC3 (τ = 0.36)	(21,22)
Random forest & Boosting					
Martins	Structure-based	Protein-ligand atom pairwise interaction counts	Binding affinity	3rd place affinity prediction HSP90 ligands GC1 (post co-crystal release)	(23)
Prathipati	Ligand-based	Circular fingerprints	Binding affinity	3rd place affinity prediction HSP90 ligands GC1 (pre co-crystal release)	(24)
Totrov	Structure-based	Physics-based terms from ICM scoring function	Binding affinity	1st place affinity prediction Cathepsin S ligands GC3 (τ = 0.39)	(26)
Yang	Combination	Vina interaction terms + fragment descriptors	Binding affinity	—	(27)

Neural network					
Wang	Combination, MLP	Vina terms + ligand descriptors	Binding affinity	—	(28)
Evangalidis	Ligand-based, MLP	Circular and 2D pharmacophore fingerprints	Binding affinity	1st place affinity prediction Cathepsin S ligands GC4 (τ = 0.54)	(29)
Koes	Structure-based, CNN	3D voxel grid of protein-ligand complex	Pose score or binding affinity	1st place affinity prediction JAK SC2 kinase sub-challenge GC3 (τ = 0.55)	(30,31)
Yakovenko	Structure-based, CNN	3D voxel grid of protein-ligand complex	3D voxel map (MD poses)	2nd place pose prediction FXR ligands GC2 (median RMSD = 1.27 Å)	(32)
Rial	Structure-based, CNN	3D voxel grid of protein-ligand complex	Binding affinity	1st place affinity prediction BACE ligands GC4 (τ = 0.39)	(35)
Guo-Wei	Structure-based, GAN	Mathematical representation of 3D pose	Discriminator outputs if input pose is near-native/'real' or 'fake'	1st place pose prediction BACE ligands GC4 (median RMSD = 0.55 Å)	(33,34)
AutoML					
Stroganov	Structure-based	Energy contributions from Lead Finder scoring function	Binding affinity	—	(38)

Supplementary Table 1. Summary of all machine learning based methods used in the D3R grand challenges. Ligand-based methods only use ligand properties and require no protein-ligand complex for predictions. Structure-based methods on the other hand require a complex, either predicted through docking or ligand template alignment. Acronyms: MLP, multilayer perceptron. CNN, convolutional neural network. GAN, generative adversarial network.

References

1. Koukos PI, Bonvin AMJJ. Integrative modelling of biomolecular complexes. J Mol Biol. 2019;432(9):2861–81.
2. Kitchen DB, Decornez H, Furr JR, Bajorath J. Docking and scoring in virtual screening for drug discovery: Methods and applications. Nat. Rev. Drug Discov. 2004;3(11):935–949.
3. Rifaioglu AS, Atas H, Martin MJ, Cetin-Atalay R, Atalay V, Doğan T. Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases. Brief. Bioinformatics. 2019;20(5):1878–1912.
4. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596(7873):583–589.
5. Wójcikowski M, Ballester PJ, Siedlecki P. Performance of machine-learning scoring functions in structure-based virtual screening. Sci Rep. 2017;7(46710).
6. Shen C, Hu Y, Wang Z, Zhang X, Zhong H, Wang G, et al. Can machine learning consistently improve the scoring power of classical scoring functions? Insights into the role of machine learning in scoring functions. Brief Bioinform. 2021;22(1):497–514.
7. Gathiaka S, Liu S, Chiu M, Yang H, Stuckey JA, Kang YN, et al. D3R grand challenge 2015: Evaluation of protein–ligand pose and affinity predictions. J Comput Aided Mol Des. 2016;30(9):651–68.
8. Gaieb Z, Liu S, Gathiaka S, Chiu M, Yang H, Shao C, et al. D3R Grand Challenge 2: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. J Comput Aided Mol Des. 2018;32(1):1–20.
9. Gaieb Z, Parks CD, Chiu M, Yang H, Shao C, Walters WP, et al. D3R Grand Challenge 3: blind prediction of protein–ligand poses and affinity rankings. J Comput Aided Mol Des. 2019;33(1):1–18.

10. Parks CD, Gaieb Z, Chiu M, Yang H, Shao C, Walters WP, et al. D3R grand challenge 4: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. J Comput Aided Mol Des. 2020;34(2):99–119.
11. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: a review. Biophys. Rev. 2017;9(2):91–102.
12. Li J, Fu A, Le Zhang ·. An Overview of Scoring Functions Used for Protein-Ligand Interactions in Molecular Docking. Interdiscip Sci. 2019;11(2):320–8.
13. Crampon K, Giorkallos A, Deldossi M, Baud S, Steffenel LA. Machine-learning methods for ligand–protein molecular docking. Drug Discov. 2022;27(1):151–64.
14. Mobley DL, Dill KA. Binding of Small-Molecule Ligands to Proteins: "What You See" Is Not Always "What You Get.". Structure. 2009;17(4):1326–1330.
15. Verdonk ML, Cole JC, Hartshorn MJ, Murray CW, Taylor RD. Improved protein-ligand docking using GOLD. Proteins Struct Funct Genet. 2003;52(4):609–23.
16. Trott O, Olson AJ. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2009;31(2):455–461.
17. Dominguez C, Boelens R, Bonvin AMJJ. HADDOCK: A protein-protein docking approach based on biochemical or biophysical information. J Am Chem Soc. 2003;125(7):1731–7.
18. Butler KT, Davies DW, Cartwright H, Isayev O, Walsh A. Machine learning for molecular and materials science. Nature. 2018;559(7715):547–555.
19. Hastie TT. The Elements of Statistical Learning Second Edition. [Book]
20. Baumgartner MP, Evans DA. Lessons learned in induced fit docking and metadynamics in the Drug Design Data Resource Grand Challenge 2. J Comput Aided Mol Des. 2018;32(1):45–58.

21. Kurkcuoglu Z, Koukos PI, Citro N, Trellet ME, Rodrigues JPGLM, Moreira IS, et al. Performance of HADDOCK and a simple contact-based protein–ligand binding affinity predictor in the D3R Grand Challenge 2. J Comput Aided Mol Des. 2018;32(1):175–85.

22. Koukos PI, Xue LC, Bonvin AMJJ. Protein–ligand pose and affinity prediction: Lessons from D3R Grand Challenge 3. J Comput Aided Mol Des. 2019;33(1):83–91.

23. Santos-Martins D. Interaction with specific HSP90 residues as a scoring function: validation in the D3R Grand Challenge 2015. J Comput Aided Mol Des. 2016;30(9):731–42.

24. Prathipati P, Nagao C, Ahmad S, Mizuguchi K. Improved pose and affinity predictions using different protocols tailored on the basis of data availability. J Comput Aided Mol Des. 2016;30(9):817–28.

25. 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. Mol Inform. 2015;34(2–3):115–26.

26. Lam PCH, Abagyan R, Totrov M. Hybrid receptor structure/ligand-based docking and activity prediction in ICM: development and evaluation in D3R Grand Challenge 3. J Comput Aided Mol Des. 2019;33(1):35–46.

27. Yang Y, Lu J, Yang C, Zhang Y. Exploring fragment-based target-specific ranking protocol with machine learning on cathepsin S. J Comput Aided Mol Des. 2019;33(12).

28. Wang B, Ng HL. Deep neural network affinity model for BACE inhibitors in D3R Grand Challenge 4. J Comput Aided Mol Des. 2020;34(2):201–17.

29. Evangelidis T. D3R workshop 2019. https://www.youtube.com/watch?v=Svv1_im7KhY

30. Sunseri J, Ragoza M, Collins J, Koes DR. A D3R prospective evaluation of machine learning for protein-ligand scoring. J Comput Aided Mol Des. 2016;30(9):761–71.

31. Sunseri J, King JE, Francoeur PG, Koes DR. Convolutional neural network scoring and minimization in the D3R 2017 community challenge. J Comput Aided Mol Des. 2019;33(1):19–34.

32. Yakovenko O, Jones SJM. Modern drug design: the implication of using artificial neuronal networks and multiple molecular dynamic simulations. J Comput Aided Mol Des. 2018;32(1):299–311.

33. Nguyen DD, Cang Z, Wu K, Wang M, Cao Y, Wei GW. Mathematical deep learning for pose and binding affinity prediction and ranking in D3R Grand Challenges. J Comput Aided Mol Des. 2019;33(1):71–82.

34. Duy Nguyen D, Gao K, Wang M, Wei G-W. MathDL: mathematical deep learning for D3R Grand Challenge 4. J Comput Aided Mol Des. 2020;34:131–47.

35. Jiménez J, Škalič M, Martínez-Rosell G, De Fabritiis G. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. J Chem Inf Model. 2018;58(2):287–96.

36. Goodfellow IJ, Pouget-Abadie J, Mehdi Mirza BX, Warde-Farley D, Ozairy S, Courville A, et al. Generative Adverserial Nets. Adv Neural Inf Process Syst. 2014;27:2672–80.

37. Hutter F, Kotthoff L, Vanschoren J. Automated Machine Learning Methods, Systems, Challenges. [Book]

38. Stroganov O V., Novikov FN, Medvedev MG, Dmitrienko AO, Gerasimov I, Svitanko I V., et al. The role of human in the loop: lessons from D3R challenge 4. J Comput Aided Mol Des. 2020;34(2):1095–1105.

39. Sunny S, Jayaraj PB. Protein–Protein Docking: Past, Present, and Future. Protein J. 2021;41(1):1–26.