HAL
open science

Machine-learning methods for ligand–protein molecular docking

Kevin Crampon, Alexis Giorkallos, Myrtille Deldossi, Stéphanie Baud, Luiz Angelo Steffenel

To cite this version:

Kevin Crampon, Alexis Giorkallos, Myrtille Deldossi, Stéphanie Baud, Luiz Angelo Steffenel. Machine-learning methods for ligand–protein molecular docking. Drug Discovery Today, 2022, 10.1016/j.drudis.2021.09.007 . hal-03352012v2

HAL Id: hal-03352012

https://hal.univ-reims.fr/hal-03352012v2

Submitted on 14 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Machine learning methods for ligand-protein molecular docking

Kevin Crampon[a,b,c], Alexis Giorkallos[c], Myrtille Deldossi[c], Stephanie Baud[a], Luiz Angelo Steffenel[b,*]

[a]*Université de Reims Champagne Ardenne, CNRS, MEDyC UMR 7369, 51097 Reims, France*
[b]*Université de Reims Champagne Ardenne, LICIIS - LRC CEA DIGIT, Reims, Frane*
[c]*Atos SE, Center of Excellence in Advanced Computing, Echirolles, France*

Abstract

Artificial intelligence (AI) is often presented as a new Industrial Revolution. Many domains use AI, including molecular simulation for drug discovery. In this review, we provide an overview of ligand–protein molecular docking and how machine learning (ML), especially deep learning (DL), a subset of ML, is transforming the field by tackling the associated challenges.

Keywords

Molecular docking; Sampling; Scoring; Machine learning; Deep learning; Data representation

---

Introduction

Drug discovery is a complex process that involves in vitro tests of putative drugs and in vivo validation, among other steps. However, before these steps, researchers need to perform an extensive evaluation of candidate molecules, from which a single drug might become a commercialized product. Testing an extensive database, even in vitro, is time-consuming and costly. Indeed, the drug discovery process takes an average of 12 years from start to commercialization, with an average cost of US$1.8 billion.[1]

Researchers and pharmaceutical industries aim to reduce both the time and cost associated with drug development. Molecular docking can be used as a complex filter to highlight only the most interesting drug candidates. Molecular docking can also be used to detect potential drug side effects or molecule toxicities. Molecular docking uses the 3D structures of two molecules, the ligand and the target, to predict the preferred orientation of the first with respect to the second when bound to each other to form a stable complex.[2] Usually, the ligand is the smallest molecule, although the denomination choice is project dependent. In drug discovery, the ligand is an active principle, and the target is a biological macromolecule (e.g., a protein or DNA). However, the docking covers a wider range of pair possibilities: protein–DNA, protein–RNA, protein–sugar, protein–peptide, and protein–small compounds. We focus here on protein–small compound (commonly called protein–ligand) molecular docking because it covers an important selection of existing docking methods. We present concepts for ligand–protein docking that are also usable for other docking types. Several experimental methods can be used to obtain the 3D structure of a molecule. Although X-ray-based methods are by far the most prominent, nuclear magnetic resonance (NMR)-based and electron microscopy (EM)-based methods also feature. For instance, the Protein Data Bank (PDB), a database of protein 3D structures, includes almost 90% of the structures solved with X-ray crystallography (and almost 99% if only the structures for which the ligand–protein binding affinity is known are taken into account)[3] and almost 8% solved with NMR. Information about methods and statistics are available on the PDB[4] website (www.rcsb.org).

Drug discovery often requires tests against a comprehensive ligand library on one target. This process is called virtual screening (VS)[5] or high-throughput VS (HTVS). It is used to reduce the number of tested ligands with in vitro and in vivo experiments; the ranking of the ligands allows the elimination of candidates displaying very low affinities and, thus, not interesting from a pharmaceutical perspective. Finally, through VS, the most interesting molecules are selected for further in vitro and in vivo testing. VS can be ligand based (i.e., only ligand information is used): depending on the method, its structure, chemical properties, or a combination thereof are used to predict the binding, given that similar ligands will bind similar targets. VS can also be structure based, using the complex molecular structure to determine whether the ligand will bind the target. Molecular docking can be used to perform

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)
Published in Drug Discovery Today (January 2022) https://doi.org/10.1016/j.drudis.2021.09.007

a structure-based VS campaign. Some papers use the molecular docking expression as a structure-based VS synonym.[6, 7, 8] Here, we focus only on structure-based methods.

Detecting the optimally bound ligand among a database is an important use for molecular docking. However, it can also be the basis of research to find new targets (in the case of drug repositioning) or to characterize potential side-effects, such as toxicity,[9] whereby the protein and ligand roles are switched (Fig. 1). The authors call this process variously inverse docking,[10] reverse docking,[11] inverse VS,[9] or target screening.[12] Additionally, inverse docking processes are performed using classical docking methods structured in a specific pipeline.

In recent years, ML methods, such as DL, have been implemented to optimize the docking process. In this review, we discuss ligand–protein docking and the associated ML approaches.

Ligand–protein molecular docking

Based on 3D structures, a molecular docking[13] experiment can predict the conformation of a complex and its binding affinity. Molecular docking is a combination of two processes. The first is sampling, which involves generating a set of conformations from a rigid 3D ligand. The method is evaluated on its capacity to explore the conformational space of the ligand. This space gathers all theoretically possible conformations. The second step is scoring, which evaluates the binding affinity of each protein–ligand complex formed (called a pose). Even if sampling and scoring are introduced separately, they can be significantly correlated because scoring functions (SF) often guide the sampling method.

The main challenges for any molecular docking method (Fig. 2) are dealing with molecular flexibility and faithfully reflecting real binding, both with a reasonable computing time. Here, we summarize problems associated with molecular docking, current challenges, and approaches to address them (without ML).

First challenge: Molecule flexibility

In real conditions, the flexibility of the molecules is reflected through the vibrations of bounds, angles, and dihedrals. Even though it is an essential element in molecular docking, many of the pioneer methods considered molecules as rigid structures and used the principle of lock-and-key[14] to solve docking problems. New approaches based on heuristics and improvements in computing capacity allowed the integration of ligand flexibility by exploring the conformational space of the ligand. These methods are semiflexible because only the smallest molecule is considered flexible, whereas the target is still rigid.[15] Progressively, other methods have been developed to consider both molecules as flexible. Hence, the flexibility of the target can be considered in different ways[16]: the conformational space of the target can be assessed with extensive sampling (e.g., through molecular dynamics), and relevant structures can be selected to perform numerous rigid target docking experiments. Another strategy considers the side-chain flexibility of the residues around the binding site. Given the hypothesis that the presence of the ligand induces these changes, it is known as 'induced fit'.[16]

The ligand conformational space sampling

A molecule can have several degrees of freedom (three to describe its position, three its orientation, and the last to characterize its intrinsic flexibility regarding rotatable bounds or dihedrals), all of which generate the conformational space. Exploring this space is computationally infeasible even for a small compound. Thus, a range of sampling methods exists, each of which optimizes its exploration and to find the best conformations. Sampling methods can be classified into shape matching, systematic, stochastic, and simulation.[17, 18] Table 1 presents examples of docking software and associated sampling techniques. Stochastic methods are currently the most used and involve a broad panel of methods.

Shape-matching methods

Shape matching is a method used by the first docking program, DOCK.[19] Such techniques represent molecules (the ligand and the receptor) with geometrical shapes, such as spheres or polyhedrons, and use the principle of matching or complementary shapes to find new conformations. However, because it does not consider the internal ligand flexibility, one solution is to generate ligand conformations immediately before the search.[16]

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Systematic methods

Systematic methods allow for quantitative exploration of the conformational space of the ligand. Iterative methods (IM) attempt to generate all conformations of a ligand, starting from a given conformer. All degrees of freedom are explored, and a given increment controls the size of the sampling. The generated conformational space can be huge even for a small ligand.

Database methods use databases of conformers, such as Flexibase.[20] These databases contain, for each ligand, a set of conformations and, instead of computing all possible geometries, favor communication with a database holding precomputed conformations. Thus, the computing time is reduced at the expense of important storage space for the databases.

Finally, fragment-based methods (FBs)[21] can be used to search for the best conformation, either through place-and-join strategies or incremental strategies. Place-and-join methods cut the ligand into fragments and place them around the target site. Then, each fragment is moved to minimize its energy and, finally, all fragments are joined to rebuild the ligand. By contrast, incremental methods place the first fragment, minimize its energy, and then add the next fragment, which is also minimized. The process is repeated until the ligand is fully rebuilt. Ligand cutting can bring uncertainty in the final ligand pose. Indeed, energy minimization can differ between an isolated piece of ligand and the whole molecule. The rebuilt poses can sum the imprecision of all ligand pieces.

Stochastic methods

Unlike systematic methods, stochastic methods are used to explore only a small part of the conformational space of the ligand. These methods use pseudo-random functions to generate conformations and SFs to guide them in their exploration of the conformational space. The most used methods are Monte-Carlo (MC),[22] ant colony (AN), genetic algorithm (GA), and particle swarm optimization (PSO).[23] The choice of the hyperparameters influences the stochastic methods and, thus, some relevant areas might be forgotten.

Simulation methods

These methods explore the conformations of the molecule using computed simulations, such as molecular dynamics. Simulation methods use classic physics laws, such as Newton's law, to simulate atomic and molecular motions and generate new conformations. For instance, De Azevebo[24] used the program GROMACS,[25] a molecular dynamic solution. Simulation-based methods notorious drawback is the compute time to explore the conformational space, which is why these methods mainly complement other methods.[17]

Second challenge: The binding scoring

Ranking of the bound conformation of the ligand is managed, for all software, with a scoring function. The SF usually aims to estimate the free energy of binding. Given that computing the exact value of this energy is computing intensive, SFs can be designed to produce a score accurate enough to be used in docking simulation, allowing for many evaluations. In addition, SFs can be used to guide sampling algorithms. Among the different classes of SF, historical and hierarchical families[17, 26, 27] include physics based, empirical, knowledge based, and consensus. First, we review the mathematical foundations of scoring functions: the scoring function space. Examples of software and standalone SFs are presented in Table 1.

The scoring function space

A SF determines the conformation of the ligand that binds best with a given protein. The first protein space definition deals only with sequence.[28] However, the most appropriate definition is 'a space containing all protein folds, where similar structures are close together'.[29] Hence, the ligand can be considered as a chemical space item that gathers all small compounds.[30]

Each complex is a set comprising an item of the protein space and an item of the chemical space. A third space is the SF space, which contains all possible scoring functions. It assumes that at least one SF space item can predict the binding affinity between the structure of a protein space and a compound from the chemical space. Computational

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

methods enable this space to be explored to determine the optimal SF for the considered protein and chemical subsets.

Physics-based scoring functions

The physics-based family was first introduced by Li *et al.*[27] to gather different SF types, the most well known of which is the force field class. This subclass of SFs estimates the free energy with a weighted sum of several energy terms. Which selection depends on the chosen force field. The most common energy terms are Van der Waals, electrostatic interactions, and hydrogen bonds. Numerous force fields are available, including AMBER,[31] GROMOS,[32] OPLS,[33] and CHARMM[34]. Force field-based SFs can be designed using a single or a combination of different force fields. Force field functions are often used for their accuracy related to the use of atomic distances and separate computing of bound and unbound complex energies, such as implemented in AutoDock4.[35] The physics-based family also comprises solvent models and quantum mechanics classes. The former adds solvation/desolvation effects and torsion entropy to classical force-field terms.[36] By contrast, the latter mixes quantum and molecular mechanics to improve SF accuracy in a reasonable computing time.[37] Li *et al.* found that quantum mechanics-based SFs are currently the most promising physics-based subclass.[27]

Empirical scoring functions

Similar to force field-based methods, empirical methods estimate the free energy of binding but without massive computing requirements. This estimation is achieved by evaluating a weighted sum of parameters, such as the number of hydrogen bonds, hydrophobic/hydrophilic contacts, and so on. These parameters are simpler than force-field parameters and, thus, also quicker to compute.

Knowledge-based scoring functions

Knowledge-based SFs rely on the elaboration of a potential of mean force.[38] Based on the statistical analysis of intermolecular interactions within large 3D structural databases of complexes, the score attributed to a new complex considers that intermolecular interactions between certain types of atom or functional group are more probable than others.

Consensus scoring functions

Each choice has compromises, and some SFs perform better on an entire class of complex but poorer on others. Through the combination of different types of SF, consensus SFs aim to optimize their respective advantages. This can be achieved in different ways, such as number-by-number, rank-by-number, vote-by-number, or through a linear combination.[26]

Third challenge: Computing time

Computing time is a key metric for both sampling (huge space to explore) and scoring (invoking occurrences). In both cases, the choice of algorithm and its implementation are crucial. Regarding the sampling, a way to reduce computing time involves docking the ligand on a delimited zone of the protein surface (e.g., a cube with a 20 Å edge centered on a specific point of interest is regularly used). Therefore, knowledge of the localization of the interaction site on the target is crucial and is often related to biological results. The drawback associated with this method lies in the fact that it is not possible to generalize the results to uncharacterized or different systems easily. Indeed, if known ligands are bound on the site of a particular target, there is no guarantee that new ligands bind in the same site. Similarly, a localized search can not be transposed to a new target. Some studies use more demanding docking simulations without any a priori knowledge and explore the surface of whole target to overcome these limitations: this process is called blind docking. The choice of a delimited area significantly impacts the docking accuracy: if the box does not contain the binding site or only a part of it, then docking will be erroneous.

Furthermore, some methods, such as binding site detection, allow the use of delimited searches on a target without a priori information by predicting putative binding sites on the target surface. Commonly, this search is done either by a geometrical search, such as FPocket,[39] or by looking for the most interesting zones regarding the free energy of binding, as in Q-SiteFinder, which uses a −CH3 probe to detect such zones.[40] Usually, no information about the ligand

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

is necessary. Another way to reduce time is to use integrative docking methods, which integrate experimental data to drive the model.[41]

Finally, another way to accelerate computing is to use a high-performance computing (HPC) environment. Even if this avenue is independent of the docking software, it should not be dismissed. A parallel approach was recently developed, called Automatic Molecular Inverse Docking Engine (AMIDE),[42] initially intended for inverse docking and fitted for a classic approach. AMIDE is based on AutoDock4 and its default SF and includes a set of scripts allowing parallel execution on HPC environments.

Data

Data have a key role in the development of molecular docking methods, especially for ML-based methods. Data quantity, and quality, and how the model represents them significantly impact performance and accuracy. Regarding data volume, the PDB provides an extensive database of molecular complexes.

Data quality

When developing ML models for molecular docking, it is important to train and validate the models over established data sets instead of using synthetic or augmented data sets. This guarantees representativeness, exhaustiveness, and variety for the training set, and allows for intermethod comparisons of objective criteria. Common data sets include: (i) PDBbind,[43] which is based on the PDB and updated each year with new complexes; each new version comes with three sets of different sizes: General (21 382), Refined (4852), and Core (285) for the 2019 version; (ii) Directory of Useful Decoys (DUD)[44] and DUD-E[45] (for Enhanced) contain 40 and 102 target molecules and 2950 and 22 886 active ligands, respectively. Each ligand has 36 or 50 decoys, respectively that are physically close but topologically different; (iii) the Maximum Unbiased Validation (MUV)[46] data set contains 17 targets, ligands (30 per target), and decoys (50 per ligand). It is based on the National Institute of Health (NIH) PubChem database; (iv) The Community Structure-Activity Resource (CSAR),[47] which is a docked-complex database; (v) the sc-PDB,[48] which is a database based on the PDB, but compared with the previously mentioned data set, it also contains information about protein binding sites.

Data representation

Data representation is a central piece of the data science response to a specific problem. Data have become more detailed and incorporate increasingly complex pieces of information. The choice of representation type has a significant impact on docking performance. Even though 3D coordinates can be directly used as input, methods often use other representations produced from 3D coordinates, including descriptors, molecule fingerprints, or interaction fingerprints, image-based, or graphs.

A set of descriptors is the easiest way to represent a molecular complex. A descriptor is a hand-engineered feature characterizing a variable degree of the fidelity of a complex or a molecule. Descriptors can also reflect physiochemical properties, such as a list of atoms of certain types, the number of atom pairs between the ligand and the target for a given threshold, or an energy term. Descriptors can also be geometrical if they are derived from the 3D structure of a molecule. Finally, a combination of several of these descriptors is usually used to represent a complex. This descriptors class is often easily understandable and usable, but the descriptors can only represent the complex as a unique object, limiting the model performances.

Fingerprints are a high-level representation of molecules or complexes. The first category relies on the molecular fingerprint, in which 3D data are converted into 1D data, commonly a string of bits, integers, or characters. The chemical formula is not detailed, whereas the structural formula has more details but might be less suitable from a computing standpoint. Fingerprints can represent 2D structures, such as the MACCS molecular fingerprint, which accounts for additional chemical properties,[49] or encode 3D structures, such as FuzCav, which represents the protein binding site 3D structure combined with chemical properties.[50] DL can also be used with Molecular Surface Interaction Fingerprinting (MaSIF) to encode a protein.[51] The second category is based on interaction fingerprinting, the most well known of which are Structural Interaction Fingerprint (SIFt)[52] and Structural Protein-Ligand Interaction Fingerprint (SPLIF).[53] Fingerprints allow improved complex description abstraction. Compared with a set of

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

descriptors that list some chemical/geometrical properties, a fingerprint projects the elements to a latent space more suited for ML. In some ways, they behave like autoencoders, a class of dimensionality reduction algorithms[54] used to reduce the input dimensions.

The emergence of DL and particularly convolutional neural networks (CNNs) has made possible the use of a new kind of data representation in the form of its actual 3D structure. Complexes are first discretized on a 3D grid, in which each cell of the lattice is a voxel (volumetric pixel). Atoms are sparsely distributed in the lattice. Additionally, voxels have channels (e.g., RGB for images) that can complement the set of features with properties such as atom type, charge, and hybridization. Image-based data representations better reflect the complexity, including the 3D structure, compared with a classic fingerprints method. Moreover, even if a lot of information is integrated into this representation, it remains concise. However, the main drawback is that this data representation is sensitive to noise because a slight rotation (nudge) of the molecule in one direction results in a completely different data point. Moreover, the discretization of the coordinates of an atom might involve a loss of accuracy of the conformation of the molecule. These issues can be partially fixed, with data augmentation for the first problem, and a coarser molecular representation (considering residues rather than atoms) for the latter.

Graph data can circumvent previous representation limitations because they change the absolute character of the frame of reference to a relative, more flexible one. Additionally, considering the unstructured nature of their data, graphs are a natural way to represent molecules. A general formalism allows filling nodes, edges, and global properties with all kinds of attribute. Even though molecule representation as a graph is more instinctive, Morrone *et al.*[55] used a graph to represent the interaction between a ligand and a target.

Machine learning for ligand–protein molecular docking

ML can bring new strategies to score a complex, either by optimizing an existing SF (e.g., refining the weights of an empirical function) or by developing a new SF taking the structure of a complex as input.[56] Moreover, ML is sometimes used for VS (classification mode) and binding site detection. Once the data set is chosen and the data representation is decided, the ML model can be developed. The use of ML in molecular docking has evolved rapidly, and the previous decade saw the emergence of numerous methods, all bringing significant improvement.

Here, we provide a comprehensive overview of ML methods used in the context of ligand–protein molecular docking, presenting functions used for scoring, classification (VS mode), and binding site detection. Existing studies[57, 58, 59, 60] provide a comprehensive overall view of the domain, detailed in Table 2, Table 3 for ML and DL, respectively. Although the ML renaissance is more than a decade old, ML methods were introduced to the field of molecular docking relatively recently. Therefore, we classify the methods according to their type.

Linear regression

The most basic use of ML is linear regression, which determines the weights of the linear equation. For instance, Tool to Analyze the Binding Affinity (TABA)[61] represents a ligand-protein interaction as a set of mass-spring contacts and then uses ML methods to parametrize the affinity equation of the complex.

Random Forest methods

Random Forests (RF) were the first attempt to use ML methods for molecular docking. A RF is an ensemble method that builds upon and smooths the results of an ensemble of decision trees. Each tree is built with nodes representing a split on a single and unique criterion. Additionally, training on different randomized subsets reduces variance, thus improving with overfitting issues, a method called 'bagging'.

The first version of RF-Score[62] takes a set of descriptors as input that describe the number of atom pairs from both molecules involved in the docking. Pairs are conserved if the distance between two atoms is less than a certain cut-off (which is a hyperparameter), and atoms belong to one of these types: C, N, O, S, P, F, Cl, Br, or I, for a total of 36 descriptors. RF-Score was updated twice, and the last revision uses energy terms from AutoDock Vina's SF to improve the complex description.[63] All three versions use a set of 500 trees to run their models. In 2017, the same set of models had been trained against the DUD-E data set under the name RF-Score-VS[64] to classify complexes

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

instead of scoring them. More recently, Yasuo *et al.* introduced Similarity of Interaction Energy VEctor-Score (SIEVE-Score).[65] Compared with RF-Scores, SIEVE-Score performs the search on 1000 random trees and uses a residues level representation: for each residue in the targets, three interaction energies with the ligand (Van der Waals, Coulomb, and hydrogen bonds) are computed. A complex is represented by a vector of size $3 * nres$, where *nres* is the number of residues, called the interaction fingerprint of the complex. This method is simple and powerful but still problematic because variable-length input vectors tend to be limiting for many ML models.

Gradient boosting trees method

In Gradient Boosting, submodels are trained sequentially instead of simultaneously and from a residual set of its predecessor. It is a form of knowledge distillation and often shows better results compared with standard bagging.

In 2019, Nguyen *et al.* proposed the Algebraic Graph Learning Score (AGL-Score),[66] which uses a multiscale, multiclass weight-colored subgraph data representation. The entire molecule is a graph in which the attributes of the nodes express the type of a selection of atoms along with spatial positioning, and edges represent noncovalent bonds, such as Van-der-Waals or hydrogen bonds between the connected atoms. Once the graph is built, a series of descriptive statistics is produced from the eigenvalues of the adjacency matrix (or the Laplacian matrix) and is used as the input vector to train a Boosted Tree.

Support vector machine methods

Support vector machines (SVMs) were a popular class of ML algorithms before the development of DL. First introduced for classification problems, SVMs were then adapted for regression (SVRs). The model not only separates classes, but also maximizes the margin between elements closest to its center. Combined with kernel methods, they are a tool capable of solving nonlinear problems.

Li *et al.*[67] introduce two models of this kind. The first is based on a knowledge-based pairwise potential vector (SVR-KB). The other approach takes a set of physicochemical (Van der Waals energy, ratio of ligand-buried solvent-accessible surface area, and hydrophobic effect) descriptors as input (SVR-EP). ID-Score[68] is another SVR for scoring. This method is based on the same representation as SVR-EP only with additional descriptors, such as metal–ligand bonding interactions or desolvation effects. Finally, PLEIC-SVM is a SVM for specific-target VS that relies on an embedding of a fingerprint called the Protein-Ligand Empirical Interaction Component (PLEIC) fingerprint.[69] Three values are computed for the residue of each target: Van der Waals interactions, hydrophobic contacts, and hydrogen bonding. All residue feature vectors are then concatenated to produce the feature vector of the complex used as input by a SVM.

Multilayer perceptron methods

Multilayer perceptrons (MLPs) were the first deep neural network topologies developed, and were inspired by the Perceptron. They comprise stacks of layers formed of a series of units, all connected from layer to layer.

NNscore v1[70] was the first attempt to bring artificial neural networks to molecular docking. It is a simple feed-forward MLP with an input vector of 194 features (including basic pairwise atom binding, energy terms, or the number of rotatable bonds), a single 5-unit hidden layer, and a classification output layer ('good' or 'poor' binder). A year later, its v2[70] made use of energy terms from Vina's SF as primary descriptors and added features from BINding ANAlyser[71] (including v1's descriptors). In addition, the network is rewritten to deal with regression (one output neuron), having a better capacity (hidden layer pushed to ten neurons). In 2020, Gentille *et al.* introduced Deep Docking,[72] in which the labels are produced by performing molecular docking on a subset of the ZINC15[73] ligand database with a specific set of proteins. Given that there is no mention of the network topology, a set of physicochemical descriptors is used instead. Deep Docking takes the Morgan fingerprint[74] to represent the molecular structure of the ligand. Deep Docking trains its network on the previously mentioned subset of ZINC15 and classifies the other ligands between two classes (binder and non-binder).

Convolutional neural network methods

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

CNNs comprise convolutional layers and a tool to catch spatial correlations. Filter weights are learned from sliding across the layer input to build a relevant abstract representation of the original data.

AtomNet[75] is a commercial molecular docking software and one of the first to rely on CNN. It uses a 3D grid, in which each cell represents some basic structural features (e.g., atom types or SPLIF, SIFt fingerprints). The input of the network is a vectorized grid with a 20 Å edge and 1 Å spacing, with four convolutional layers, followed by two hidden layers of 1024 neurons. A logistic regression classifies the input between two classes. For DeepVS, also a CNN, Pereira *et al.*[76] defined the initial atom feature set with a context (atom types, atomic partial charges, amino-acid types, and distance to neighbors) for the atoms of each complex. To compensate for variable input size, the network incorporates a lookup table. The resulting vector is a fixed-size float array that summarizes input data. It is then processed by a single 2D convolutional layer to extract abstract information and two classic layers to produce a classification. Ragoza *et al.* introduced a CNN-based SF[77] that works on similar 3D grid images. The novelty here is that each atom is represented by an uncertainty distribution around the center of the atom instead of a fixed value. The network is a succession of three blocks (convolution and pooling) followed by a fully connected (FC) binary classification layer.

Atomic CNN[78] is built from two types of unique operation: atom type-specialized convolutions of 1 × 1 filters and radial pooling that filters across the atom neighbors. This approach uses atom coordinates and atom types as inputs, the former builds the interatomic distance matrix, and the latter is used to prepare the atom type matrix. The first layer (atomic convolution) combines matrices with each other, and the radial pooling layer is then used to reduce the dimension of the matrix. Finally, an atomistic FC layer flattens the feature volume (signature vector), followed by two FC layers, producing a final regression output. Although previous methods focused exclusively on binding scoring or classification, DeepSite[79] aims to find potential binding sites. The 3D input protein grid is augmented along the channel axis with eight physicochemical descriptors, and the network is a standard CNN (3D convolution followed by MaxPooling), leading to a regression score of potential. Imrie *et al.* developed DenseFS,[80] which combines Ragoza's data representation and a skip-connection network called a Densely Connected Convolutional Network (DenseNet).[81] Stepniewska-Dziubinska *et al.* designed Pafnucy,[82] a classic CNN built to estimate affinity between a ligand and a target from an initial 4D tensor (3D coordinates discretized on a 3D grid and 19 features). The network comprises three convolutional layers followed by three FC layers that produce a binding score. DeepAffinity[83] is another unusual network engineered around recurrent neural networks (RNNs) for scoring, taking a SMILES representation of the ligand, whereas the target embedding is a string called the Structural Property Sequence. Both terms are then independently milled into a sequence-to-sequence (autoencoder) model, their latent vectors processed each by a 1D convolutional layer and then concatenated before a FC layer produces the affinity score.

In DeepBindRG,[84] Zhang *et al.* cleverly flatten the input complex into a projected 2D image and perform residual network (ResNet)[85] computations to produce an affinity score. In OnionNet,[86] Zheng *et al.* suggested a multilayer intermolecular contact, in which a series of shells is built around a central atom. Inside each onion layer, there is a relevant feature set (depending on its encapsulating atoms). This allows the authors to account for nonlocal interactions. Eight atom types (leading to 64 pairs) and 60 shells are stacked for a total of 3840 features. The model is formed of three convolutional layers followed by three FC CNNs. FRSite[87] for faster R-CNN site predictor was developed to predict protein binding sites. It takes a 3D grid with eight commonly used channels to represent the target. The authors used a particular 3D CNN adapted from the Faster R-CNN.[88] This network is split into three subnetworks: the first is a 3D CNN feature extractor, the output of which is fed to the second and third parts of the network. The second part is a 3D region proposal network, which allows extraction of putative binding sites. Finally, the outputs of the first and second parts are given to the third to classify the resulting sites.

Francoeur *et al.*[89] extended work by Ragoza [et al.,][77] taking the same input data representation and general model architecture but performing a comprehensive hyperoptimization to produce more convolutional layers and average pooling instead of max-pooling. The authors of Pafnucy[82] also worked on binding site detection with the same protein representation used in Pafnucy and proposed Kalasanty,[90] in which the protein is discretized on a 3D grid, and 18 descriptors are used for each atom. Taking inspiration from semantic image segmentation, Stepniewska-Dziubinska *et al.* used a U-Net[91] to identify potential binding sites. The Kalasanty data representation was adapted by

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

the DeepSurf[92] authors: instead of discretizing all the molecule atoms as in the original paper, the authors selected only a few points of interest from a solvent-accessible surface mesh. Each point neighborhood is then discretized on a 3D grid with the same features as Kalasanty. Finally, the resulting model is Bottleneck 3D-LDS-Resnet, itself an evolution of ResNet.[85]

Graph neural networks

Graph neural networks (GNNs) are a variety of neural networks that work on graph-formatted data. They have evolved from spectral methods to a more flexible comprehensive modeling tool. Graph convolutional networks (GCNs) are a particular class of GNNs, applying convolution and pooling operations from CNNs to graphs.

The first molecular docking method to use graph data was PotentialNet.[93] Instead of only covalent bonds, it considers additional bonds with one adjacency matrix for each bond type, concatenated along the channel and resulting in a 3D adjacency matrix. Moreover, it uses a distance matrix that indicates the distance between each atom pair. The network is a GCN split into three stages: in the first stage, only covalent bonds are used for the propagation; then, both covalent and noncovalent bonds are used for propagation, and, finally, a 'graph gather' step, which gathers matrix rows by summing, is followed by a FC layer used to produce a binding score. Lim *et al.*[94] introduced a GNN with a gated-augmented attention layer (GAT). For each node, in addition to regular edges, atoms in a close neighborhood (5 Å) are also connected. This method works on three matrices: the first is the node features matrix, the second is the adjacency for only covalent bonds (in the ligand and the protein), and the third is adjacency for intermolecular interactions (in which the second matrix is included). In each step of the network, the node feature matrix is updated by a GAT, and the second matrix is updated by another GAT, which uses the third matrix. Then, the second updated node feature matrix is subtracted from the first. After additional steps, all node feature vectors are summed, and a FC layer uses this vector to classify the complex.

Torgn *et al.* proposed a VS method that uses two graphs to represent the target and the ligand.[95] On the target side, graph nodes are the residues (restricted to the binding site), edges connect every neighbor in a sphere of 7 Å, and the features are extracted from the FEATURE program.[96] The ligand graph is a classic 2D molecular graph. The training is a two-step process: the first encodes the binding site graph (dimensionality reduction). This encoder is kept for the second step, which concatenates its output to a second GCN trained on ligand graphs. The result is fed to the FC layer and a Softmax classifier. Tanebe *et al.*[97] used GNNs to classify good or bad binders. This approach represents the ligand by a graph generated from SMILES string in which nodes are atoms and edges are bonds. The target is a graph in which nodes are residues, and edge types (five in total) depend only on the distance between the C_α of each residue. A GNN then embeds both graphs, and the resulting concatenation is used to classify the complex. In the Tsubaki *et al.* method,[98] the authors used the SMILES representation of the ligand to produce a graph and a GNN to embed this graph into a vector. For the target, the amino acid sequence is embedded by a CNN. Both are concatenated, and an FC followed by Softmax makes a prediction. Recently, Morrone *et al.* proposed a new DL method for the docking problem using GCN.[55] This method uses two graphs as input. The first represents the covalent ligand graph (L). The second graph is a contact graph built by hopping from the protein atoms to the ligand atoms in a 4 Å neighborhood (LP). This modular method can take L, LP, or L + LP as input. In each case, the input is embedded by a GCN and fed to a CNN for prediction.

Comparison of network architectures

The DL family is divided into three classes: MLPs, CNNs, and GNNs. We have discussed three MLPs, but only two describe their architectures, which is not enough to understand the evolution of MLP architecture. Moreover, the two with known architectures are different versions of the same underlying method.

Regarding CNNs, a variety of networks (e.g., ResNet and UNet) bring many architectural possibilities. For CNNs, the main architecture is adapted from 3D grids (3DCNN), such as AtomNet or Pafnucy. The topologies are not identical, but use common CNN layers. However, some methods propose original architectures, such as the Atomic CNN, or use well-known architecture, such as Kalansaty, which is based on UNet, a network historically created for image segmentation. Given the black-box nature of DL-based models, it is difficult to assess the superiority of a CNN topology with regards to its counterparts because this depends on the chosen data representation.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

For GNNs, the limited number of available methods does not allow a particular architecture to be highlighted. Some methods use GNNs as a first step to embed the inputs and then use FC layers or a CNN to produce an output.

Performance measurement

So far, we have focused on modeling only; however, data sets are dedicated not only to training, but also to evaluating and assessing the methods used. Therefore, we present here performance metrics for classification methods (VS), SF, and binding site detection.

VS assessment

In addition to data sets, authors also use a series of metrics to compare with other existing contributions. In VS, the model is evaluated on its capacity to distinguish between binding and nonbinding ligands. Generally, the enrichment factor (EF) or the area under the curve (AUC) of the receiver operating characteristic curve (ROC) are used. EF evaluates whether selected ligands are better binders than randomly selected ones and takes only real positive values: a poor classifier has EF ≤1, whereas a better-than-random one has EF > 1. This metric allows the rate of true binders among the top-ranked ligands (in the top {1%,2%,5%,10%}) to be compared with the rate in a random selection. By contrast, ROC curve is used to visually assess the quality of a classifier as its discrimination threshold varies. The optimal AUC value is 1.0, whereas 0.0 is the worst case (random case being 0.5).

Table 4 details AUC performances from various methods, including those drawn from.[53, 54] This table shows the difficulty in drawing a simple conclusion about VS performances. The data set is the first problem that arises when trying to compare pure performance. Second, most methods are not self-supporting and require the adjunction of other classic sampling software. Thus, even if two methods are assessed on the same data set, their performances are impacted by the chosen sampling method. Moreover, even though the sampling method is theoretically the same, they might differ on parameter initialization, as explained by Shen *et al.*[59] Consequently, we have to use raw performances given in the same paper to compare methods. For example, Lim's method is better than AtomNet and Ragoza's method, according to Lim *et al.*[94]

SF assessment

Comparative Assessment of Scoring Functions (CASF),[99] developed by Su *et al.*, introduced three criteria to assess a SF method: scoring power, ranking power, docking power.

Scoring reflects the ability of a SF 'to produce binding scores in a linear correlation with experimental binding data'. It uses Pearson's correlation coefficient (R_p) (sometimes R_p^2) and the standard deviation in linear regression (*SD*). R_p can be between −1 and +1. The closer to 1, the better the method assessed. For *SD*, the smallest value is optimal. R_p and, to a lesser extent, *SD* are the most-used criteria.

Ranking refers to the ability of a SF 'to correctly rank the known ligands of a certain target protein by their binding affinities when the precise binding poses of those ligands are given'. The assessment uses Spearman's rank correlation coefficient (ρ), Kendall's rank correlation coefficient (τ), and the Predictive Index (*PI*). These criteria have values of [−1,1], whereby +1 indicates a perfect ranking and −1 the reverse.[99]

Docking represents the ability of a SF 'to identify the native ligand binding pose among computer-generated decoys'. Assessment uses RSMD (Eq. (1)) to compare top-ranked ligand by the SF method and native ligand pose. A threshold is used to consider docking as a success. Commonly, the cut-off used is 2.0 Å.

$$RMSD = \sqrt{\frac{\sum_{i=0}^{n}(X_E^i - X_S^i)^2}{n}} \qquad (1)$$

where *E* is the expected coordinates of the atoms, *S* is the simulated coordinates of the atoms, and *n* is the number of atoms in the ligand.

The CASF data set is identical to the PDBbind core set for the corresponding year. Table 5 lists examples of method assessments for scoring power. Compared with Table 4, the entries in Table 5 are more comparable: to assess a SF, authors use a data set of docked complexes. Consequently, the sampling step is unnecessary and, thus, assessments

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

differ only by the used data set. However, a wide data set variety is available, each with several subsets and versions (e.g., PDBbind). If the used data sets are identical, then their respective performance can be compared. For example, OnionNet has a better Rp score than Francoeur's method on the PDBbind 2017 core set.

Binding site detection assessment

There are two main options to assess binding site detection methods. First, we can use a data set of already docked ligand–protein complexes (e.g., PDBbind) and predict binding sites for proteins. Then, for each complex, it is possible to consider the method output as a success if at least one predicted protein site is the real binding site. This approach is interesting if the binding site composition is unknown.

However, all previously mentioned methods used the sc-PDB[48] data set for training and assessment. This data set contains the atomic composition of the sites and, once the predicted sites are defined, atomic compositions can be compared. The authors used two metrics: the distance to the center of the binding site (DCC) metric measures the distance either between the center of the real binding site and the closest atom of the predicted site, or between the center of the real binding site and the center of the predicted site. In both cases, the site detection is a success regarding a threshold fluctuating between 4 Å and 20 Å: the better the success ratio, the better the method.

The second is the discretized volumetric overlap (DVO) metric, which assesses the overlap between the predicted and real binding sites. Authors use the Jaccard Index on the convex hull of the sites. Both volumes are discretized, and the ratio between the overlapping volume and merged volume is computed; the closer to 1 the Jaccard Index, the better the method.

We have not provided a table of the performances of binding site detection methods because data are often provided as charts without the raw values. However, most recent methods are compared by their authors to older methods in their respective papers.

Concluding remarks

Here, we have discussed how ML and particularly DL can help us tackle molecular docking challenges. We have presented three challenges: sampling, scoring, and computing time. However, in terms of the sampling challenge, a ML method that attempts to tackle it has yet to be developed.

The scoring challenge is, without doubt, the most studied problem. Indeed, ML scoring methods are interesting in terms of scoring function space exploration. Many ML methods have been developed, and most outperform classical methods. Thus, ML SF can be regarded as a hybrid of knowledge-based and empirical functions. Indeed, similar to knowledge-based approaches, ML methods extract statistics from a comprehensive database to build the most relevant model. By contrast, ML methods use relatively simple inputs and find links between them. It is even more evident for DL methods that aim to optimize the weights of networks, which is similar to the goal of an empirical function. Although they are not the main focus of this review, a particular class of ML models called physics-informed DL have potential because they incorporate physical constraints in the learning process.

This review has shown that ML methods outperform classic approaches, whether for scoring or classification. Moreover, recently proposed GNN methods have interesting performances but remain underexplored. Therefore, more in-depth research is required of these methods.

The last challenge is computing time. No ML scoring methods are compared with others in terms of computing time required, which makes it difficult to discuss the capacity of ML in terms of time reduction. However, a delimited search can be used to reduce time; some ML methods to predict binding sites are presented herein and outperform classical binding site detection methods, according to their authors. Therefore, we consider GNNs as an interesting approach for improving current ML methods.

One general drawback is that most methods have not been proposed and assessed in a complete docking pipeline; thus, it would be interesting to compare classical methods, such as AutoDock, with the ML workflow. Moreover, the training and inference times of ML methods are rarely mentioned by authors. We believe that this information should be included in future studies because it provides invaluable insights into the complexity of these models.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Acknowledgments

The authors thank Manuel Dauchez for his helpful and fruitful discussions. This work was supported by the Association Nationale de la Recherche et de la Technologie (ANRT) through a CIFRE grant.

Figures

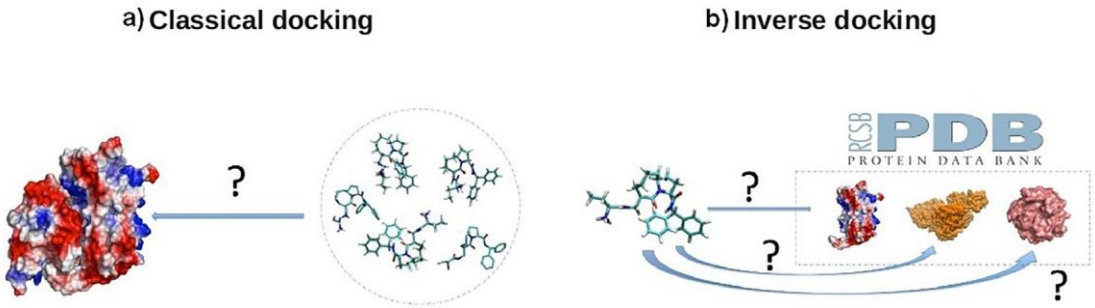

Figure 1. Types of docking. (a) classic docking process, which involves finding the optimal ligand for a given protein. (b) inverse docking process involves finding the optimal target for a given ligand.

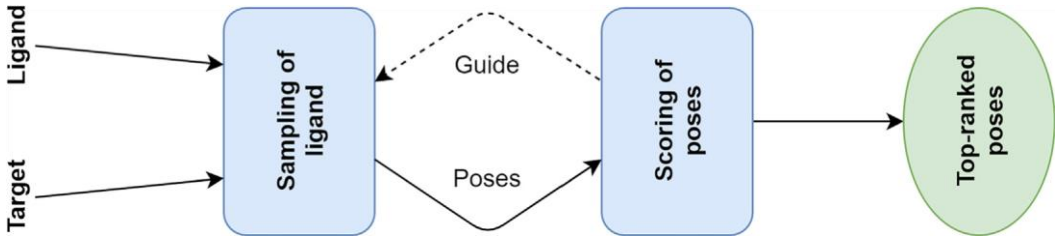

Figure 2: A simplified workflow of the molecular docking process with sampling and scoring subprocesses.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Tables

Table 1: Examples of molecular docking software and their sampling and scoring methods.[a]
[a]Based on: R. Vasseur, PhD thesis, Université de Reims Champagne-Ardenne, 2015.
[b]Scoring functions only.

Software	Year	Sampling	Scoring	Refs
ICM	1994	Stochastic (MC)	Force field (ECEPP)	100
GOLD	1995	Stochastic (GA)	Force field (AMBER)	101
SANDOCK	1998	Shape matching	Empirical	102
MultiScore[b]	2001	–	Consensus	103
LigandFit	2003	Stochastic (MC)	Empirical	104
DrugScore[b]	2005	–	Knowledge based	105
DFire[b]	2005	–	Knowledge based	106
Glide	2006	Stochastic (MC)	Empirical	107
PLANTS	2006	Stochastic (AC)	Empirical	108
SODOCK	2007	Stochastic (PSO)	Force field (AMBER)	109
eHiTS	2007	Systematic (FB)	Empirical	110
KScore[b]	2008	–	Knowledge based	111
AutoDock 4	2009	Stochastic (GA)	Force field (AMBER)	35
AutoDock Vina	2010	Stochastic (GA)	Empirical	112
VoteScore[b]	2011	–	Consensus	113
FRED	2011	Systematic (IM)	Customizable	114
D-Score[b]	2013	–	Force field (Tripos)	115
FlexAID	2015	Stochastic (GA)	Empirical	116
MOLS 2.0	2016	Systematic (IM)	Force field (AMBER)	117
DINC 2.0	2017	Systematic (FB)	Empirical	118

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Table 2: Machine learning methods for ligand–protein docking.[a]
[a]Method developed for a specific type of target.

Name	Year	Input	Usage	Refs
Linear regression				
TABA	2020	Mass-Spring system	Scoring	61
Random Forest				
RF-Score	2010	Set of descriptors	Scoring	62
RF-Score-v2	2014	Set of descriptors	Scoring	119
RF-Score-v3	2015	Set of descriptors	Scoring	63
RF-Score-VS	2017	Set of descriptors	VS	64
SIEVE-Score	2019	Energy vectors	VS	65
Gradient boosting trees				
AGL-Score	2019	Multiscale weighted colored subgraphs	Scoring, VS	66
Support vector machines				
SVR-KB	2011	Pairwise potential vector	Scoring	67
SVR-EP	2011	Set of descriptors	Scoring	
ID-Score (SVR)	2013	Set of descriptors	Scoring	68
PLEIC-SVM	2017	PLEIC Fingerprint	Target-specific VS[a]	69

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Table 3: Deep learning methods for ligand–protein docking.
[a]Based on CNN terminology, channels are used to represent descriptors.
[b]Indicates end-to-end methods.

Name	Year	Input	Usage	Refs
Multilayer perceptrons				
NNscore	2010	Set of descriptors	VS	70
NNscore 2.0	2011	Set of descriptors	Scoring	
Deep docking	2020	2D fingerprints	VS	72
Convolutional neural networks				
AtomNet	2015	3D grid with descriptors	VS	75
DeepVS	2016	Set of descriptors	VS	76
Ragoza2017	2017	3D grid of 34 channels	VS	77
Atomic CNN	2017	3D structure	Scoring	78
DeepSite[a]	2017	3D grid of eight channels	Binding-site detection	79
DenseFS	2018	3D grid of 34 channels	VS	80
Pafnucy	2018	3D grid of 19 channels	Scoring	82
DeepAffinity	2019	Fingerprints	Scoring	83
DeepBindRG	2019	2D matrix	Scoring	84
OnionNet	2019	64 descriptors for each shell	Scoring	86
FRSite[a]	2019	3D grid of eight channels	Binding site detection	87
Francoeur2020	2020	3D grid of 28 channels	Scoring	89
Kalasanty[a]	2020	3D grid of 18 channels	Binding site detection	90
DeepSurf[a]	2020	3D grid of 18 channels	Binding site detection	92
Graph neural networks				
PotentialNet	2018	Atom bond graph	Scoring	93
Lim2019	2019	Atom bond graph	VS	94
Torgn2019[a]	2019	Residue graph of target + Ligand graph	VS	95
Tanebe2019[a]	2019	Residue graph of target + ligand graph	VS	97
Tsubaki2019[a]	2019	Sequence of target + ligand graph	VS	98
Morrone2020	2020	Ligand bond graph + contact graph	VS	55

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Table 4: Assessment of VS methods with DUD, DUD-E, and MUV data sets.
[a]Listed scores are sometimes in supplemental information.
The same method can have different performances on the same data set and with the same docking engine, because of the impact of data preparation and docking engine settings.

Presented method	**Docking engine**	**Assessed method**	**1st Used dataset**	***AUC***	**2nd Used dataset**	***AUC***	**Ref[a]**
PLEIC-SVM	Glide	Gscore **PLEIC-SVM**	DUD	0.82 **0.93**			69
NN-Score methods	Vina	Vina **NNScore-v1** **NNScore-v2**	DUD	0.70 **0.78** **0.76**			70
DeepVS	Dock 6.6	Dock 6.6 **DeepVS**	DUD	0.48 **0.74**			76
	Vina	Vina **DeepVS**	DUD	0.62 **0.81**			
RF-Score-VS	Vina	Vina RF-Score-V3 **RF-Score-VS**	DUD-E	0.74 0.67 **0.84**			64
	Dock 6.6	Dock 6.6 RF-Score-V3 **RF-Score-VS**	DUD-E	0.61 0.66 **0.80**			
AtomNet	Smina	Smina **AtomNet c**	DUD-E	0.696 **0.895**			75
Ragoza's method	Smina	Smina RF-Score NN-Score **Ragoza's method**	DUD-E	0.716 0.622 0.584 **0.868**	MUV	0.549 0.512 0.441 **0.522**	77
DenseFS	Vina	Vina Ragoza's method **DenseFS**	DUD-E	0.703 0.862 **0.917**	MUV	0.546 0.507 **0.534**	80
Lim's method	Smina	Smina AtomNet Ragoza's method **Lim's method**	DUD-E	0.689 0.855 0.868 **0.968**	MUV	0.533 0.518 **0.536**	94
Torgn's method		Vina RF-Score NNScore Ragoza's method **Torgn's method**	DUD-E	0.716 0.622 0.584 0.868 **0.886**	MUV	0.538 0.536 0.454 0.567 **0.621**	95
Morrone's methods	Vina	Vina **Morrone L** **Morrone LP** **Morrone L+LP**	DUD-E	0.70 **0.82** **0.65** **0.81**			55

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

Table 5: Assessment of SF methods.
[a]Listed scores are sometimes in supplemental information. [b]Values = (Rp)2.

Presented method	Dataset	Assessed SF	R_p	*SD*	Ref[a]
RF-Score	PDBbind 2007 Core Set	ChemScore	0.441	2.15	62
		GoldScore	0.295	2.29	
		RF-Score	**0.776**	**1.58**	
RF-Score-v2	PDBbind 2007 Core Set	**RF-Score-v2**	**0.803**	**1.54**	119
RF-Score-v3	PDBbind 2007 Core Set	**RF-Score-v3**	**0.803**	**1.42**	63
SVR-KB and SVR-EP	CSAR-SETI1	**SVR-KB**	**0.59[b]**		67
		SVR-EP	**0.55[b]**		
	CSAR-SETI2	**SVR-KB**	**0.67[b]**		
		SVR-EP	**0.50[b]**		
ID-Score	PDBbind 2007 Core Set	**ID-Score**	**0.753**	**1.63**	68
Atomic CNN	PDBbind 2015 Core Set	**Atomic CNN**	**0.448[b]**		78
	PDBbind 2015 Refined Set	**Atomic CNN**	**0.529[b]**		
Pafnucy	PDBbind 2016 Core Set	**Pafnucy**	**0.78**	**1.37**	82
	CASF-2013	**Pafnucy**	**0.70**	**1.61**	
DeepBindRG	CASF-2013	Vina	0.5725		84
		DeepBindRG	**0.6394**		
AGL-Score	CASF-2007	ID-Score	0.753		66
		Vina	0.554		
		AGL-Score	**0.830**		
	CASF-2013	**AGL-Score**	**0.792**		
OnionNet	PDBbind 2013 Core Set	AutoDock	0.54	1.61	86
		Vina	0.54	1.60	
		ChemScore	0.592	1.82	
		Pafnucy	0.70	1.61	
		RF-Score-V3	0.74	1.51	
		AGL-Score	0.792	1.45	
		OnionNet	**0.78**	**1.45**	
	PDBbind 2016 Core Set	Pafnucy	0.78		
		RF-Score-V3	0.80		
		AGL-Score	0.833		
		OnionNet	**0.816**		
Francoeur's method	PDBbind 2016 Core Set	**Francoeur**	**0.733**		89

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

References

1 S. Sinha, D. Vohora, Drug discovery and development: an overview, in: D. Vohora, G. Singh (Eds.), Pharmaceutical Medicine and Translational Clinical Research, Academic Press, London, 2018, pp. 19–32.

2 T. Lengauer, M. Rarey, Computational methods for biomolecular docking, Current Opinion in Structural Biology 6 (3) (1996) 402–406.

3 M. Veit-Acosta, W.F. de Azevedo Junior, The impact of crystallographic data for the development of machine learning models to predict protein-ligand binding affinity, Current Medicinal Chemistry 28 (2021) 1–17.

4 H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, et al., The Protein Data Bank, Nucleic Acids Research 28 (1) (2000) 235–242. 5 I. Muegge, M. Rarey, Small molecule docking and scoring, Reviews in Computational Chemistry 17 (2001) 1–60.

6 W. Shan, X. Li, Y. Hequan, K. Lin, Convolutional neural network-based virtual screening, Current Medicinal Chemistry 28 (10) (2021) 2033–2047.

7 G. Bitencourt-Ferreira, A. Duarte da Silva, W.F. de Azevedo Junior, Application of machine learning techniques to predict binding affinity for drug targets: a study of cyclin-dependent kinase 2, Current Medicinal Chemistry 28 (2) (2021) 253– 265.

8 S. Musella, G. Verna, A. Fasano, S. Di Micco, New perspectives on machine learning in drug discovery, Current Medicinal Chemistry 27 (2020) 1–24.

9 X. Xu, M. Huang, X. Zou, Docking-based inverse virtual screening: methods, applications, and challenges, Biophysics Reports 4 (1) (2018) 1–16.

10 Y.Z. Chen, D.G. Zhi, Ligand–protein inverse docking and its potential use in the computer search of protein targets of a small molecule, Proteins: Structure, Function, and Bioinformatics 43 (2) (2001) 217–226.

11 J. Fan, A. Fu, L. Zhang, Progress in molecular docking, Quantitative Biology 7 (2) (2019) 83–89.

12 A. Khan, A. Chandra Kaushik, S.S. Ali, N. Ahmad, D.-Q. Wei, Deep-learningbased target screening and similarity search for the predicted inhibitors of the pathways in Parkinson's disease, RSC Adv 9 (18) (2019) 10326–10339. 13 V.B. Sulimov, D.C. Kutov, A.V. Sulimov, Advances in docking, Current Medicinal Chemistry 26 (42) (2019) 7555–7580.

14 E. Fischer, Einfluss der Configuration auf die Wirkung der Enzyme, Berichte der Deutschen Chemischen Gesellschaft 27 (3) (1894) 2985–2993.

15 A.R. Leach, I.D. Kuntz, Conformational analysis of flexible ligands in macromolecular receptor sites, Journal of Computational Chemistry 13 (6) (1992) 730–748.

16 S.-Y. Huang, X. Zou, Advances and challenges in protein-ligand docking, International Journal of Molecular Sciences 11 (8) (2010) 3016–3034.

17 S.F. Sousa, P.A. Fernandes, M.J. Ramos, Protein–ligand docking: current status and future challenges, Proteins: Structure, Function, and Bioinformatics 65 (1) (2006) 15–26.

18 M. Novic, T. Tibaut, M. Anderluh, J. Borisek, T. Tomasic, The comparison of docking search algorithms and scoring functions: an overview and case studies, in: S. Dastmalchi, M. Hamzeh-Mivehroud, B. Sokouti (Eds.), Methods and Algorithms for Molecular Docking-Based Drug Design and Discovery, IGI Global, Pennsylvannia, 2016, pp. 99–127.

19 I.D. Kuntz, J.M. Blaney, S.J. Oatley, R. Langridge, T.E. Ferrin, A geometric approach to macromolecule-ligand interactions, Journal of Molecular Biology 161 (2) (1982) 269–288.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

20 S.K. Kearsley, D.J. Underwood, R.P. Sheridan, M.D. Miller, Flexibases: a way to enhance the use of molecular docking methods, Journal of Computer-Aided Molecular Design 8 (5) (1994) 565–582.

21 R.D. Taylor, P.J. Jewsbury, J.W. Essex, A review of protein-small molecule docking methods, Journal of Computer-Aided Molecular Design 16 (3) (2002) 151–166.

22 N. Metropolis, S. Ulam, The Monte Carlo method, Journal of the American Statistical Association 44 (247) (1949) 335–341.

23 X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, Moscow, 2010.

24 F. de Azevedo, Molecular dynamics simulations of protein targets identified in Mycobacterium tuberculosis, Current Medicinal Chemistry 18 (9) (2011) 1353– 1366.

25 M.J. Abraham, T. Murtola, R. Schulz, et al., GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers, SoftwareX 1–2 (2015) 19–25.

26 A. Oda, K. Tsuchida, T. Takakura, N. Yamaotsu, S. Hirono, Comparison of consensus scoring strategies for evaluating computational models of proteinligand complexes, J Chem Inf Model 46 (1) (2006) 380–391.

27 J. Li, A. Fu, L. Zhang, An overview of scoring functions used for protein–ligand interactions in molecular docking, Interdisciplinary Sciences: Computational Life Sciences 11 (2) (2019) 320–328.

28 J.M. Smith, Natural selection and the concept of a protein space, Nature 225 (5232) (1970) 563–564.

29 Bitencourt-Ferreira G, de Azevedo WF. Exploring the scoring function space. In: de Azevedo Jr. WF, ed. Docking Screens for Drug Discovery. New York: Springer; 2019: 275–281.

30 R.S. Bohacek, C. McMartin, W.C. Guida, The art and practice of structure-based drug design: a molecular modeling perspective, Medicinal Research Reviews 16 (1) (1996) 3–50.

31 P.K. Weiner, P.A. Kollman, AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions, Journal of Computational Chemistry 2 (3) (1981) 287–303.

32 W.F. van Gunsteren, H.J.C. Berendsen, Computer simulation of molecular dynamics: methodology, applications, and perspectives in chemistry, Angewandte Chemie International Edition in English 29 (9) (1990) 992–1023.

33 W.L. Jorgensen, J. Tirado-Rives, The OPLS optimized potentials for liquid simulations potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin, J Am Chem Soc 110 (6) (1988) 1657–1666.

34 B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: A program for macromolecular energy, minimization, and dynamics calculations, Journal of Computational Chemistry 4 (2) (1983) 187– 217.

35 G.M. Morris, R. Huey, W. Lindstrom, M.F. Sanner, R.K. Belew, D.S. Goodsell, et al., AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility, Journal of Computational Chemistry. 30 (16) (2009) 2785– 2791.

36 S. Genheden, U. Ryde, The MM/PBSA and MM/GBSA methods to estimate ligand-binding affinities, Expert Opinion on Drug Discovery 10 (5) (2015) 449– 461.

37 P. Chaskar, V. Zoete, U.F. Röhrig, Toward on-the-fly quantum mechanical/molecular mechanical (QM/MM) docking: development and benchmark of a scoring function, J Chem Inf Model 54 (11) (2014) 3137–3152.

38 I. Muegge, Y.C. Martin, A general and fast scoring function for proteinligand interactions: a simplified potential approach, J Med Chem 42 (5) (1999) 791– 804.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

39 V. Le Guilloux, P. Schmidtke, P. Tuffery, Fpocket: an open source platform for ligand pocket detection, BMC Bioinformatics 10 (1) (2009) 168.

40 A.T.R. Laurie, R.M. Jackson, Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites, Bioinformatics 21 (9) (2005) 1908– 1916.

41 E. Karaca, A.M.J.J. Bonvin, Advances in integrative modeling of biomolecular complexes, Methods 59 (3) (2013) 372–381.

42 R. Vasseur, S. Baud, L.A. Steffenel, X. Vigouroux, L. Martiny, M. Krajecki, et al., Inverse docking method for new proteins targets identification: a parallel approach, Parallel Computing 42 (2015) 48–59.

43 R. Wang, X. Fang, Y. Lu, S. Wang, The PDBbind Database: collection of binding affinities for proteinligand complexes with known three-dimensional structures, J Med Chem 47 (12) (2004) 2977–2980.

44 N. Huang, B.K. Shoichet, J.J. Irwin, Benchmarking sets for molecular docking, J Med Chem 49 (23) (2006) 6789–6801.

45 M.M. Mysinger, M. Carchia, Irwin JohnJ, B.K. Shoichet, Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking, J Med Chem 55 (14) (2012) 6582–6594.

46 S.G. Rohrer, K. Baumann, Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data, J Chem Inf Model 49 (2) (2009) 169–184.

47 R.D. Smith, J.B. Dunbar Jr, P.M. Ung, E.X. Esposito, C.Y. Yang, S. Wang, et al., CSAR benchmark exercise of 2010: combined evaluation across all submitted scoring functions, J Chem Inf Model 51 (9) (2011) 2115–2131.

48 J. Desaphy, G. Bret, D. Rognan, E. Kellenberger, sc-PDB: a 3D-database of ligandable binding sites—10 years on, Nucleic Acids Research 43 (D1) (2015) D399–D404.

49 J.L. Durant, B.A. Leland, D.R. Henry, J.G. Nourse, Reoptimization of MDL keys for use in drug discovery, J Chem Inf Comput Sci 42 (6) (2002) 1273–1280.

50 N. Weill, D. Rognan, Alignment-free ultra-high-throughput comparison of druggable proteinligand binding sites, J Chem Inf Model 50 (1) (2010) 123– 135.

51 P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M.M. Bronstein, et al., Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning, Nature Methods 17 (2) (2020) 184–192.

52 Z. Deng, C. Chuaqui, J. Singh, Structural Interaction Fingerprint (SIFt): a novel method for analysing three-dimensional proteinligand binding interactions, J Med Chem 47 (2) (2004) 337–344.

53 C. Da, D. Kireev, Structural Protein-Ligand Interaction Fingerprints (SPLIF) for structure-based virtual screening: method and benchmark study, J Chem Inf Model 54 (9) (2014) 2555–2561.

54 I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, 2016.

55 J.A. Morrone, J.K. Weber, T. Huynh, H. Luo, W.D. Cornell, Combining docking pose rank and structure with deep learning improves protein–ligand binding mode prediction over a baseline docking approach, J Chem Inf Model 60 (9) (2020) 4170–4179.

56 G.S. Heck, V.O. Pintro, R.R. Pereira, M.B. de àvila, N.M.B. Levin, W.F. de Azevedo, Supervised machine learning methods applied to predict ligandbinding affinity, Current Medicinal Chemistry 24 (23) (2017) 2459–2470.

57 Q.U. Ain, A. Aleksandrova, F.D. Roessler, P.J. Ballester, Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening, WIREs Computational Molecular Science 5 (6) (2015) 405–424.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

58 H. Li, K.-H. Sze, G. Lu, P.J. Ballester, Machine-learning scoring functions for structure–based drug lead optimization, WIREs Computational Molecular Science 10 (5) (2020) e1465.

59 C. Shen, J. Ding, Z. Wang, D. Cao, X. Ding, T. Hou, From machine learning to deep learning: advances in scoring functions for protein–ligand docking, WIREs Computational Molecular Science 10 (1) (2020) e1429.

60 K. Abbasi, P. Razzaghi, A. Poso, S. Ghanbari-Ara, A. Masoudi-Nejad, Deep learning in drug target interaction prediction: current and future perspectives, Current Medicinal Chemistry 28 (11) (2021) 2100–2113.

61 A. Duarte da Silva, G. Bitencourt-Ferreira, W.F. de Azevedo Jr, Taba: a tool to analyse the binding affinity, Journal of Computational Chemistry 41 (1) (2020) 69–73.

62 P.J. Ballester, J.B.O. Mitchell, A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking, Bioinformatics 26 (9) (2010) 1169–1175.

63 H. Li, K.-S. Leung, M.-H. Wong, P.J. Ballester, Improving AutoDock Vina using Random Forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets, Molecular Informatics 34 (2–3) (2015) 115–126.

64 M. Wójcikowski, P.J. Ballester, P. Siedlecki, Performance of machine-learning scoring functions in structure-based virtual screening, Scientific Reports 7 (1) (2017) 46710.

65 N. Yasuo, M. Sekijima, Improved method of structure-based virtual screening via interaction-energy-based learning, J Chem Inf Model 59 (3) (2019) 1050–1061.

66 D.D. Nguyen, G.-W. Wei, AGL-Score: algebraic graph learning score for protein– ligand binding scoring, ranking, docking, and screening, J Chem Inf Model 59 (7) (2019) 3291–3304.

67 L. Li, B. Wang, S.O. Meroueh, Support vector regression scoring of receptor– ligand complexes for rank-ordering and virtual screening of chemical libraries, J Chem Inf Model 51 (9) (2011) 2132–2138.

68 G.-B. Li, L.-L. Yang, W.-J. Wang, L.-L. Li, S.-Y. Yang, ID-Score: a new empirical scoring function based on a comprehensive set of descriptors related to protein– ligand interactions, J Chem Inf Model 53 (3) (2013) 592–600.

69 Y. Yan, W. Wang, Z. Sun, J.Z.H. Zhang, C. Ji, Protein–ligand empirical interaction components for virtual screening, J Chem Inf Model 57 (8) (2017) 1793–1806.

70 J.D. Durrant, A.J. Friedman, K.E. Rogers, J.A. McCammon, Comparing neuralnetwork scoring functions and the state of the art: applications to common library screening, J Chem Inf Model 53 (7) (2013) 1726–1735.

71 J.D. Durrant, J.A. McCammon, BINANA: a novel algorithm for ligand-binding characterization, Journal of Molecular Graphics and Modelling 29 (6) (2011) 888–893.

72 F. Gentile, V. Agrawal, M. Hsing, A.T. Ton, F. Ban, U. Norinder, et al., Deep docking: a deep learning platform for augmentation of structure based drug discovery, ACS Cent Sci 6 (6) (2020) 939–949.

73 T. Sterling, J.J. Irwin, ZINC 15 – ligand discovery for everyone, J Chem Inf Model 55 (11) (2015) 2324–2337.

74 H.L. Morgan, The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service, J Chem Doc 5 (2) (1965) 107–113.

75 Wallach I, Dzamba M, Heifets A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv 2015:

151002855.

76 J.C. Pereira, E.R. Caffarena, C.N. dos Santos, Boosting docking-based virtual screening with deep learning, J Chem Inf Model 56 (12) (2016) 2495–2506.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

77 M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, D.R. Koes, Protein–ligand scoring with convolutional neural networks, J Chem Inf Model 57 (4) (2017) 942–957.

78 Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein–ligand binding affinity. arXiv 2017: 170310603.

www.drugdiscoverytoday.com

79 J. Jiménez, S. Doerr, G. Martínez-Rosell, A.S. Rose, G. De Fabritiis, DeepSite: protein-binding site predictor using 3D-convolutional neural networks, Bioinformatics 33 (19) (2017) 3036–3042.

80 F. Imrie, A.R. Bradley, M. van der Schaar, C.M. Deane, Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data, J Chem Inf Model 58 (11) (2018) 2319–2330.

81 G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017, pp. 4700–4708.

82 M.M. Stepniewska-Dziubinska, P. Zielenkiewicz, P. Siedlecki, Development and evaluation of a deep learning model for protein–ligand binding affinity prediction, Bioinformatics 34 (21) (2018) 3666–3674.

83 M. Karimi, D. Wu, Z. Wang, Y. Shen, DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks, Bioinformatics 35 (18) (2019) 3329–3338.

84 H. Zhang, L. Liao, K.M. Saravanan, P. Yin, Y. Wei, DeepBindRG: a deep learning based method for estimating effective protein–ligand affinity, PeerJ 7 (2019) e7362.

85 K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016, pp. 770–778.

86 L. Zheng, J. Fan, Y. Mu, OnionNet: a multiple-layer intermolecular-contactbased convolutional neural network for protein–ligand binding affinity prediction, ACS Omega 4 (14) (2019) 15956–15965.

87 M. Jiang, Z. Wei, S. Zhang, S. Wang, X. Wang, Z. Li, FRSite: Protein drug binding site prediction based on faster R-CNN, Journal of Molecular Graphics and Modelling 93 (2019) 107454.

88 S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137–1149.

89 P.G. Francoeur, T. Masuda, J. Sunseri, A. Jia, R.B. Iovanisci, I. Snyder, et al., Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design, J Chem Inf Model 60 (9) (2020) 4200–4215.

90 M.M. Stepniewska-Dziubinska, P. Zielenkiewicz, P. Siedlecki, Improving detection of protein–ligand binding sites with 3D segmentation, Scientific Reports 10 (1) (2020) 5035.

91 O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer, Berlin, 2015, pp. 234–241.

92 Mylonas SK, Axenopoulos A, Daras P. DeepSurf: A surface-based deep learning approach for the prediction of ligand binding sites on proteins. arXiv 2020:

200205643.

93 E.N. Feinberg, D. Sur, Z. Wu, et al., PotentialNet for molecular property prediction, ACS Cent Sci 4 (11) (2018) 1520–1530.

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

94 J. Lim, S. Ryu, K. Park, Y.J. Choe, J. Ham, W.Y. Kim, Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation, J Chem Inf Model 59 (9) (2019) 3981–3988.

95 W. Torng, R.B. Altman, Graph convolutional neural networks for predicting drug-target interactions, J Chem Inf Model 59 (10) (2019) 4131–4149.

96 S.C. Bagley, R.B. Altman, Characterizing the microenvironment surrounding protein sites, Protein Science 4 (4) (1995) 622–635.

97 T. Tanebe, T. Ishida, End-to-end learning based compound activity prediction using binding pocket information, in: D.-.S. Huang, K.-.H. Jo, Z.-.K. Huang (Eds.), Intelligent Computing Theories and Application, Springer, Berlin, 2019, pp. 226–234.

98 M. Tsubaki, K. Tomii, J. Sese, Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences, Bioinformatics 35 (2) (2019) 309–318.

99 M. Su, Q. Yang, Y. Du, et al., Comparative assessment of scoring functions: the CASF-2016 update, J Chem Inf Model 59 (2) (2019) 895–913.

100 R. Abagyan, M. Totrov, D. Kuznetsov, ICM—a new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation, Journal of Computational Chemistry 15 (5) (1994) 488–506.

101 G. Jones, P. Willett, R.C. Glen, A.R. Leach, R. Taylor, Development and validation of a genetic algorithm for flexible docking, Journal of Molecular Biology 267 (3) (1997) 727–748.

102 P. Burkhard, P. Taylor, M.D. Walkinshaw, An example of a protein ligand found by database mining: description of the docking method and its verification by a 2.3 Å X-ray structure of a Thrombin-Ligand complex, Journal of Molecular Biology 277 (2) (1998) 449–466.

103 G.E. Terp, B.N. Johansen, I.T. Christensen, F.S. Jørgensen, A new concept for multidimensional selection of ligand conformations (MultiSelect) and multidimensional scoring (MultiScore) of proteinligand binding affinities, J Med Chem 44 (14) (2001) 2333–2343.

104 C.M. Venkatachalam, X. Jiang, T. Oldfield, M. Waldman, LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites, Journal of Molecular Graphics and Modelling 21 (4) (2003) 289–307.

105 H.F.G. Velec, H. Gohlke, G. Klebe, DrugScoreCSD knowledge-based scoring function derived from small molecule crystal data with superior recognition rate of near-native ligand poses and better affinity prediction, J Med Chem 48 (20) (2005) 6296–6303.

106 C. Zhang, S. Liu, Q. Zhu, Y. Zhou, A knowledge-based energy function for proteinligand, proteinprotein, and proteinDNA complexes, J Med Chem 48 (7) (2005) 2325–2335.

107 R.A. Friesner, R.B. Murphy, M.P. Repasky, et al., Extra Precision Glide: docking and scoring incorporating a model of hydrophobic enclosure for proteinligand complexes, J Med Chem 49 (21) (2006) 6177–6196.

108 O. Korb, T. Stützle, T.E. Exner, PLANTS: application of ant colony optimization to structure-based drug design, in: M. Dorigo, L.M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle (Eds.), Ant Colony Optimization and Swarm Intelligence, Springer, Berlin, 2006, pp. 247–258.

109 H.-M. Chen, B.-F. Liu, H.-L. Huang, S.-F. Hwang, S.-Y. Ho, SODOCK: swarm optimization for highly flexible protein–ligand docking, Journal of Computational Chemistry 28 (2) (2007) 612–623.

110 Z. Zsoldos, D. Reid, A. Simon, S.B. Sadjad, A.P. Johnson, eHiTS: A new fast, exhaustive flexible ligand docking system, Journal of Molecular Graphics and Modelling 26 (1) (2007) 198–212.

111 X. Zhao, X. Liu, Y. Wang, Z. Chen, L. Kang, H. Zhang, et al., An improved PMF scoring function for universally predicting the interactions of a ligand with protein, DNA, and RNA, J Chem Inf Model 48 (7) (2008) 1438–1447.
* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)

112 O. Trott, A.J. Olson, AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, Journal of Computational Chemistry 31 (2) (2010) 455–461.

113 D. Plewczynski, M. Łaz_niewski, M.V. Grotthuss, L. Rychlewski, K. Ginalski, VoteDock: Consensus docking method for prediction of protein–ligand interactions, Journal of Computational Chemistry 32 (4) (2011) 568–581.

114 M. McGann, FRED pose prediction and virtual screening accuracy, J Chem Inf Model 51 (3) (2011) 578–596.

115 M. Vaudel, D. Breiter, F. Beck, J. Rahnenführer, L. Martens, R.P. Zahedi, D-score: a search engine independent MD-score, Proteomics 13 (6) (2013) 1036–1041.

116 F. Gaudreault, R.J. Najmanovich, FlexAID: Revisiting docking on non-nativecomplex structures, J Chem Inf Model 55 (7) (2015) 1323–1336.

117 D.S. Paul, N.M.O.L.S. Gautham, 2.0: software package for peptide modeling and protein–ligand docking, Journal of Molecular Modeling 22 (10) (2016) 239.

118 D.A. Antunes, M. Moll, D. Devaurs, K.R. Jackson, G. Lizée, L.E. Kavraki, DINC 2.0: a new protein–peptide docking webserver using an incremental approach, Cancer Res 77 (21) (2017) e55–e57.

119 P.J. Ballester, A. Schreyer, T.L. Blundell, Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity?, J Chem Inf Model 54 (3) (2014) 944–955

* Corresponding author Steffenel, L.A. (angelo.steffenel@univ-reims.fr)