Deep Learning Techniques in Small Molecules/Protein Docking

Martijn Sturm
m.j.sturm@students.uu.nl

Supervised by: dr. Manon Réau
m.f.reau@uu.nl

First Examinor: Prof. dr. Alexandre Bonvin
a.m.j.j.bonvin@uu.nl

Second Examinor: Prof. dr. Toine Egberts
a.c.g.egberts@uu.nl

September 2020

Layman's summary (Dutch): Moleculaire docking is een onderzoeksdiscipline in biologie waarbij wordt getracht te voorspellen en te doorgronden hoe moleculen (zoals eiwitten) in het lichaam beïnvloed kunnen worden door andere moleculen die we als medicijnen zouden kunnen gebruiken. Hierbij wordt data van onder andere de ruimtelijke ordening van deze moleculen gebruikt, en aan computermodellen gevoed die vervolgens deze voorspellingen uitvoeren. Door de tijd heen zijn er verschillende systemen ontwikkeld die deze voorspellingen kunnen maken. De eerste docking systemen maakten onder andere gebruik van natuur- en scheikundige wetten om te bepalen of moleculen met elkaar binden, en/of in welke ruimtelijke oriëntatie ze zich tot elkaar verhouden. In de laatste jaren zijn machine learning technieken erg in opmars voor heel veel verschillende toepassingen waarbij voorspellingen moeten worden gemaakt. Deze technieken gebruiken data waarvan de uitkomsten al bekend zijn. Aan de hand van de data wordt een model gebouwd wat de werkelijkheid moet nabootsen. In het geval van docking, wordt data van al bekende medicijn-eiwit interacties gevoed aan de machine learning techniek. Deze techniek leert dan welke eigenschappen van de data belangrijk zijn voor voorspellingen voor moleculaire interactie, en gebruikt dit om een model te bouwen. Het model kan dan ook toegepast worden om de interacties te voorspellen voor moleculen waarvoor we nog niet weten hoe ze in interactie gaan met elkaar. In deze beschouwing wordt wetenschappelijke literatuur behandeld over deep learning technieken die worden toegepast op moleculaire docking. Deep learning technieken (een subset van machine learning) zijn beter in staat om erg basale informatie, zoals moleculaire opbouw, te gebruiken voor het voorspellen van complexe concepten. In dit geval zijn die complexe concepten de binding en ruimtelijke oriëntatie van de moleculen. De deep learning technieken vereisen echter ook vrij veel configuratie. De verschillende deep learning strategieën en configuraties die in de literatuur zijn gebruikt worden in deze beschouwing kwalitatief geëvalueerd. Ook worden andere relevante onderwerpen behandeld, zoals de data en databases die gebruikt worden om de voorspellingen te maken.

Abstract

Molecular docking gathers an ensemble of computational approaches that aim to predict the three-dimensional binding interaction between molecules (*e.g.* small molecules and proteins) using structural information. The output of docking should ideally give information about both the binding mode and the relative binding affinity of the complex compared to other complexes. All methods globally present good sampling performance, *i.e.* they are able to generate complexes that are structurally similar to experimentally solved complexes (*e.g.* X-ray crystallography, NMR, Cryo-EM). However, they present caveats in scoring those complexes, since they are unable to systematically distinguish acceptable generated complexes from unacceptable ones. This weakness limits the reliability of the docking outputs when applied to molecular complexes for which no reference structure is available. Deep learning is one of the approaches that are being extensively explored to overcome this issue. Herein, the docking process and the application of deep learning method to the docking process (data representation, neural network architecture, training procedure, and outcome measures) are reviewed. We focus on geometry and graph-based representations of molecular input data. Finally, various papers that implement deep learning techniques to solve prediction problems related to docking are discussed.

1 Introduction

Biomolecular interactions are a vital part of all processes in biology. Disease is often a result of a dysregulation within interactions between molecules. Hence, many studies are being performed to elucidate the interaction network at cellular, organ, tissue, or entire organism scale. Being able to identify some deleterious impairments in this network and to propose a solution to restore the balance is a key basis of therapeutic research.

For bio-molecular interaction to take place, at least two components are needed that can be either proteins, small molecules, DNA, or RNA. A molecular complex is the combination of entities that are interacting together. In this work we will focus on small molecules (ligands)/ protein (target) interactions.

Many small molecules have been designed by humans to fulfill medicinal purposes. Those molecules are considered drugs once they successfully pass all the pre-clinical and clinical assay steps. Designing small molecules for a specific medicinal aim is a difficult task, in which multiple requirements have to be fulfilled. First, it is necessary to identify a target that is proven to be involved in the disease development and that can be modulated by a small molecule. This notion is referred to as the druggability of a target. Second, for a small molecule to possibly have an effect on a therapeutic target it has to be able to bind to it. Third, through this binding, a desired biological effect has to be induced(Chen, 2015). Finally, the small molecule is modified to improve its ADME-tox properties.

All drug development pipelines thus start with the screening of a large number of small molecules against a druggable target to identify a "binder". The synthesizable small-molecule space is estimated to consist of 10^{60} possible small molecules(Virshup et al., 2013). Simply trying all possibilities through biological experiments is infeasible. Computational techniques could partly solve this problem by means of simulation. A computer model that represents the biochemical reality would allow to predict the interaction between ligands and targets within a given confidence. Therefore, the ligands that are predicted to most likely bind the target, can be chosen to be evaluated by biological assays first. This strategy aims to increase the likelihood that a relevant small molecule will be identified early in the drug discovery process, thus making it more efficient.

Docking is a commonly used computational approach for this purpose. The docking process comprises predicting the binding mode and/or the binding affinity of small molecules on a given target by evaluating different complex conformations (*i.e.* sampling the conformations) and by assigning a score to the binding interaction of each complex (*i.e.* scoring). To date, sampling algorithms have shown high performance in retrieving the biological binding mode of small molecules against proteins, yet it remains challenging to assign the best score to the poses that resemble the most the complex arrangement as resolved by experimental means (*i.e.* X-ray crystallography,

RMN, cryo-EM etc.), and to correlate this score with in vitro or in vivo affinity or activity(Pagadala et al., 2017). Deep learning techniques can contribute to that improvement, since they are well-equiped to learn high-level features from low-level data(Li et al., 2019). In this case, the high-level feature can be the binding mode (that is defined as the distance to the experimentally solved complex in the learning phase) or the binding affinity, while the low-level feature is molecular information of the complex such as molecular composition and spatial arrangement. Multiple options are available to represent this molecular information, such as making use of a graph or a 3D geometric space. Details about these approaches are described in section 3.2.

In this review, we will briefly present the docking process and its actual limitations, then we will detail how deep learning approach can complement the docking process in sampling and scoring, together with the theory behind. Finally, we will describe recent examples from the literature that make use of deep learning for this purpose.

2 Docking

Docking is the prediction of the structure of target-ligand complexes through computational techniques(Rodrigues & Bonvin, 2014). The binding between a ligand and target can be seen as the result of features of both components that are complementary to each other(Tripathi & Bankaitis, 2017). These features need to be represented within the docking system, so that it can effectively and truly use the molecular and physical mechanisms that underly the binding between ligand and target.

The docking process involves three main steps. First, structural data of both the target and ligand needs to be available and collected. Second, the possible conformations (poses) of the complex need to be sampled, which is the orientation of ligand and target in respect to each other(Figure 1). Lastly, a scoring function is applied to assign a score to the generated poses that is supposed to reflect the binding affinity, or the similarity to the experimentally determined binding mode of the complex(Figure 1)(Rodrigues & Bonvin, 2014). The docking approaches fall into 3 categories: the rigid-body docking, that does not account for the flexibility of the docked entities, the flexible docking, that allows rotations of the rotatable bonds of those entities, and the hybrid method that either use rigid-body and flexible docking in a sequential manner, of that divide the system into rigid and flexible areas.

In all docking strategies, data from both the ligand and the target need to be fed to the docking system. The ligand and protein structures need to be represented in a 3D space. The small molecules structure can be collected from multiple databases (*i.e.* ZINC, PubChem, Drugbank etc.). Most of the time, they are provided in 1D or 2D format and their possible 3D conformations must be enumerated with conformer generator tools. Enumerating the possible conformation is of prime importance for both rigid and flexible docking: in the former case, the conformers need to be sampled enough to make sure that the binding conformation is among the docked ones; in the latter case it is important to enumerate the possible conformation of non-rotatable bonds that cannot be sampled during the docking phase to have enough starting points for the flexible docking. On the protein side, the 3D structure can be extracted from database such as the Protein Data Bank (PDB)(Berman et al., 2000) that encompasses structures solved by techniques such as NMR-spectroscopy, X-ray crystallography, and cryo-electron microscopy(Rodrigues & Bonvin, 2014). When no structure is available, protein conformations can be modeled from sequence and eventually homologous proteins folding. The accuracy of both the small molecule and the protein conformation plays a major role in the docking process and impacts its performance(Chen, 2015)

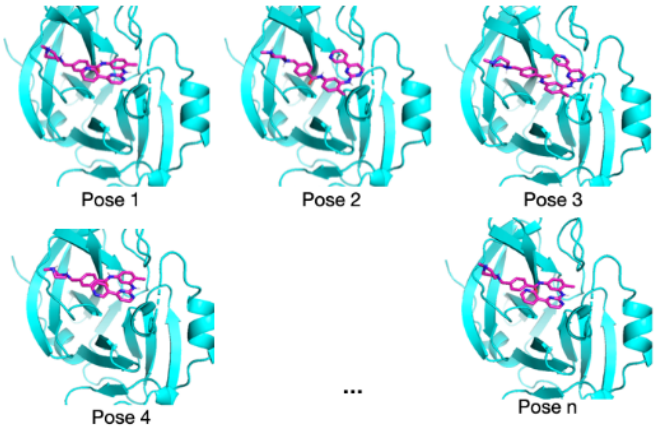
Before a docking system can be used for predictions on complexes with unknown binding properties, its performance should be estimated on benchmarking databases: the sampling phase is evaluated on solved 3D structures, the scoring phase can be either evaluated based on its ability to top rank complexes that are similar to the experimental ones (called near-native complex under a certain distance cutoff), or on its ability to correlate with experimentally measured binding affinity.

The major challenge in docking is to develop reliable and accurate scoring functions, since the sampling part is considered to have a reasonable performance(Pagadala et al., 2017). The next

sections gives an overview about how sampling and scoring are implemented in docking systems.

2.1 Sampling

With sampling, conformations and orientations of the small molecule and target in respect to each other are generated, from which the native poses need to be identified. This can either be done by keeping the molecules rigid or by respecting the freedom that protein have in their rotational bonds. Choosing for the more flexible approach requires more computational resources but mimics the natural docking process better. Contrary, the more rigid approaches are better suited to cover the entire space of conformation possibilities compared to the more flexible approaches(Rodrigues & Bonvin, 2014). Some techniques that are used for sampling include matching algorithms (based on molecular shape and chemical information), fragment-based algorithms (ligands are split up in separate parts that are added to the active site incrementally), and stochastic methods (which generate different poses by changing conformations based on change (*e.g.* Monte Carlo search and



Ideal Docking Performance

pose ID	ranked by docking	RMSD
D	1	
G	2	
E	3	
A	4	
B	5	
C	6	
F	7	

Real Docking Performance

pose ID	ranked by docking	RMSD
B	1	
G	2	
F	3	
A	4	
D	5	
E	6	
C	7	

Figure 1: Docking process. **Top panel:** Here, multiple poses are shown that are generated for a single combination of small molecule and target during the sampling phase of docking. **Bottom panel:** Pose ID links each row to a pose generated during the sampling process of docking (displayed in top panel). For a single target, each pose gets assigned a score on which they are ranked in respect to each other. A scoring function performs well if it ranks the poses with the best root-mean-square deviation (RMSD) of atomic positions compared to the experimentally observed conformation highest. **Left:** Here, the ideal output of scoring functions is shown. The poses with the lowest RMSDs (green) are ranked highest, andt the poses with bigger RMSDs (red) are at the bottom of the ranking. **Right:** This is an example of real-life scoring output. No docking software enables reaching perfect correlation between RMSD compared to the native pose, and ranking calculated by the scoring function, and it is common to evaluate the performance of docking tools by computing the enrichment of good solution in the top ranked compound.

genetic algorithms))(Meng et al., 2011).

When applying rigid docking, one must be critical about the crystal structures. The complex conformation that is used to experimentally obtain a target's holo structure can influence the docking outcomes. For example, ligands that are similar to the ligand that is part of the experimentally observed complex conformation show higher prediction accuracy than dissimilar ligands(Chaput & Mouawad, 2017). Also other experimental conditions (*e.g.* temperature) can influence the structure and hence the prediction accuracy(Chen, 2015).

A benchmark study by Chaput and Mouawad (2017) evaluated a few commercial semi-rigid docking systems, and showed that for most complexes correct conformations were sampled by each docking system. However, when these were applied to virtual screening tasks, the performance did not correlate with the docking performance. When the scoring functions of the different systems were combined, the results were much better in terms of correct rankings compared to the rankings made by each individual docking system(Chaput & Mouawad, 2017). This indicates that there is room for improvement for each scoring function individually, while each system shows the ability to sample the correct conformation for most of the presented targets.

2.2 Scoring Functions

The today-available scoring functions (SFs) can be divided into four categories; physics-based, emperical, knowledge-based, and machine learning-based(Li et al., 2019). Physics-based SFs use force fields, solvation and quantum mechanical properties; Empirical SFs use a linear combination of energetic terms in their formulas; Knowledge-based SFs use information from large quality data sets to assess the favorable and unfavorable characteristics of atom-pair interactions within the complex(Li et al., 2019). These three types of SFs are based on physico-chemical theory and hence need expert knowledge to be constructed and improved. The fourth mentioned type of SFs is the machine learning approach, which is based on a different paradigm. Here, the actual mappings between input features, and output metrics are learned by the machine learning techniques.

A shortcoming of the classical SFs is that any error in the theory that they are based on, will propagate into the docking system(Duy Nguyen et al., 2020). This is not the case for machine learning approaches, which are fundamentally based on data instead of theory. Nevertheless, a sufficient amount of data and relevant features accompanying this data need to be provided. The amount of data that is available might be a limiting factor for the future success of machine learning based approaches(Shen et al., 2020). However, the amount of data is expected to increase, and some machine learning approaches have already outperformed classical SF approaches(Li et al., 2019). Additionally, the data should reflect reality as close as possible. If any systemical error exists in the data as result of biases for example in the measurement procedure, the machine learning technique will propagate these errors and learn wrong mappings between molecular structures and scoring function outputs consequently.

In recent years, a subtype of machine learning techniques, deep learning, has made major progress in classification and regression tasks in computer imaging(Krizhevsky et al., 2012). One benefit of deep learning over classical machine learning techniques is that it requires less feature engineering. The models are able to infer features from raw input data. The extent to which the model is able to do so depends on the architecture and its depth. The successful applications in computer imaging has prompted the scientific community to multiply its application into various domain. Today it is being widely investigated and developed in computer aided drug design to complement the docking scoring phase especially, and the sampling phase to some extent. In the next sections, fundamentals about deep learning and its applicability to the scoring and sampling problems are discussed. Also, recent models from literature are reviewed.

3 Deep Learning approaches for Docking

Deep learning is an approach to solve complex problems in a computational way. It works by learning from experience (*i.e.* supervised learning), which means that known phenomena are pre-

sented to the technique and it then learns to recognize the underlying relations. The 'deep' in deep learning refers to the hierarchical manner in which simple concepts from input data are composited into more complex concepts of interests(Goodfellow et al., 2016). This abstract description can be instantiated for the docking use case. Deep learning can use simple input data such as the spatial structures of components to generate more complex concepts such as binding scores.

The process of solving a problem with deep learning is similar to using traditional machine learning. First, the problem and data need to be fathomed out, and a clear objective needs to be formulated. Then, the architecture of the deep learning network is defined so that the problem of interest can be solved. This is analogous to choosing a traditional machine learning technique (*e.g.* decision tree, logistic regression). The architecture defines the constraints for the model that will be generated. Here, model is defined as the mapping from the representation of real-world data to the output metric. For molecular binding, the model should represent the mechanisms responsible for the molecular interaction. After these steps, hyperparameter optimization is conducted making use of carefully constructed train, test and validation sets. During training, parameters are optimized so that the model can as accurately as possible predict the outcomes of interest.

In the remaining of this section, the main steps in applying deep learning to docking are described in more detail.

3.1 Data

In computer vision tasks, the performance of models increases in a logarithmic manner when the data set size is increased(Sun et al., 2017). One of the major challenges for deep learning applied to docking is the low amount of data included in data sets(Feinberg et al., 2018). This low amount of data implies that the chemical space of small molecules is sparsely covered in the available data sets. This would mean that the relationships learned from using these data sets only cover a small space in the interactome(Torng & Altman, 2019).

Besides the amount, also the quality of the data is important for the reliability and informativity of a model. Data sets such as the DUD(Huang et al., 2006) have been developed to act as a benchmark for docking studies. However, there are still differences in the data introduced between studies due to differences in data preparation(Corbeil et al., 2012). These different preparation steps and former problems in benchmark data sets can lead to errors accumulating or propagating into the data set, resulting in unreliable docking success rates(Corbeil et al., 2012). The way current data sets are often built is by including small molecules that have been found to bind to specific targets experimentally. Therefore, information about the binding properties of the targets with other (non-binding) small molecules is rarely included. This can lead to a bias in models by learning similarity between small molecules to rank their binding propensity to a specific target, rather than really capturing the binding interactions that are underlying this(Torng & Altman, 2019). This can harm the performance of Virtual Screening assays. For models that are used to distinguish native from non-native poses, this is not a problem since non-native poses can be reliably identified if one knows the native pose. Another quality concern is whether the data really represents the reality. For example, some data from the MUV data set are generated by chemical and cell-based assays in the lab. These assays measured bioactivity, though there is no guarantee that this activity is the direct result of the small molecule-target interaction(Torng & Altman, 2019).

Biases are also an issue that researchers have to keep into consideration. Artificial decoys are used in some data sets so that models can also learn from 'non-binding' small molecules. However, these decoys are artificially created, which is the consequence of the tendency in academic literature to favor publication of positive results(Réau et al., 2018). It turns out that these artificial decoys are very similar for different drug targets(Gonczarek et al., 2018). This hence could lead to models being trained on this data to learn the difference between the decoy and the active small molecules. This conflicts with the aim of the model, which is to learn binding relationships between target and small molecule(Gonczarek et al., 2018). The DUD-E data set is constructed by collecting active and non-active small molecules per target. These active small molecules are only associated with

targets they are active for, and hence are never associated to non-activity. Therefore, the model can learn to recognize structures in the active small-molecules, rather than mapping a relationship between the structures of the small molecules to the associated targets they are active for(Torng & Altman, 2019). This shows that also the setup of a data set can lead to bias.

For a learned model to be able to predict binding interactions between a large number of structurally different molecules, it has to be trained on data sets that cover these possibilities to a large extent. When other binding partners for a particular target are known, this can be utilized to learn from a wider variety of molecular interactions. For example, some proteins bind metabolites, this data can be used complementary to the data just about drug compounds and targets(Gainza et al., 2020).

Data quality is of crucial importance to be able to create an accurate model of interactions between small molecule and target. Though, representing this data in a way that a model can learn efficiently is also of high importance.

3.2 Representation of the data

An important property of a neural network is the way the input data is fed into the network. Deep learning networks are able to generate features inside the model, without the need of human feature engineering to some extent. Nevertheless, data can be adjusted or transformed so that the network can more easily learn useful mappings. Useful mappings can be defined as relations between the compound and target that represent the underlying mechanisms of their binding interaction. Properties that are useful for these representations is the structure of the components of the complex, the local properties of interacting biomolecules, and their orientation towards each other.

How should these structures be represented, so that the deep learning technique can efficiently learn about the molecular interactions? We can summarize the suitable approaches as three mathematical techniques. These are algebraic topology, differential geometry, and graph theory(Duy Nguyen et al., 2020).

The structures of the components are often encoded into fingerprints. These encodings can take on any desired dimensionality. Two-dimensional representations offer a lower computational cost, but also result in a bigger loss of information. For three-dimensional representations it is the inverse. Finding a representation harboring enough information, but still computationally efficient to be used in prediction is the main challenge here. In this paper, we reviewed two types of representation techniques to represent 3D structures that can overlap to some extent; geometry-based representations and graph-based representations.

Geometric techniques use distances and coordinates in a 3D Euclidian space as foundation to describe molecular structures. In this way, the interaction interface can be represented, enabling the prediction of binding free energy. When making use of this approach, decisions have to be taken about the resolution of the technique. A trade-off exist between atomistic and molecular-level representations, which can either lead to too much detail, or too few respectively(Duy Nguyen et al., 2020).

Graph techniques make use of nodes and edges. The nodes are often representing atoms, while the edges are the bonds between the atoms. Non-covalent bonds can be easily distinguished from covalent ones using a multi-dimension edge matrices (or adjacency matrices). Including these in the more recent representation approaches, resulted in more powerful representations(Duy Nguyen et al., 2020). In this method, there is no direct 3D representation of the target, though it could be inferred by the deep learning technique from the bonds represented as edges. A benefit is that graphs are rotational invariant and therefore need less data augmentation, which is a stark difference with 3D geometric approaches(Zamora-Resendiz & Crivelli, 2019).

Concluding, both representation approaches described above allow for encoding proximity between different vertices that can correspond either to atoms, chemical groups, or residues together with their type and physicochemical properties. There are even numerous possibilities to include and utilize these properties of the molecular complexes. Examples from literature for geometric and graph-based representations are discussed in more detail in section4.

In order to process the here presented data representations, the network has to be desgined carefully. Many different network structures have been tested and used in literature. Often, a specific type of problem is addressed by a fitting type of network. When a deep learning model's rigid structure is described, it is often referred to as the network's architecture.

3.3 Deep Learning Architectures

The network architecture of the deep learning model is closely bound to the data representation and has a big impact on its performance. Typically, the data representations discussed in the previous section are first fed into a convolutional layer. Then, the flattened outputs from those layers can be used as input for any fully connected dense layer before it goes into the output layer to predict the outcome of interest. An overview of a typical architecture making use of graph-based convolutions is shown in figure 2.

Convolutional Layers

The task of either the graph convolutional layers or the 3D (for geometric representations) convolutional layers is representing the chemical interactions that take place between the small molecule and target. That is the reason why both types use convolutional layers. These can learn features based on locality. Not only atoms in closest proximity to each other define binding characteristics, but also neighboring groups of atoms play a role (Wallach et al., 2015). Therefore, making use of multiple hidden layers in a deep learning model should allow for representation of these non-linear and complex properties closer to reality than techniques that cannot represent deep interactions(Wallach et al., 2015).

Graph Convolutional layers can be broadly divided into two types: spectral and spatial. Spectral graphs use eigenfunctions which process the entire graph at once and are specific for a particular graph (unique combination of edges and vertices). Consequently, this form is less suited for systems that need to represent the different spatial configurations of two molecules. Contrary, spatial graphs utilize the neighborhoods of nodes (Figure 2)(Zhang et al., 2019). The latter results in a more composite representation which is more generalizable to multiple and different graphs.

There are multiple hyperparameters that can be adjusted for a single convolutional layer. The number of filters determines the complexity that a model can take on in this layer. A higher number of filters enables the network to learn more complex features from the same input. However, this also leads to an increased training time since more computations have to be performed. Another hyperparameter is the filter size. Smaller filter sizes lead to more local features being extracted, thus bigger filter sizes lead to more global features. Often, multiple filter sizes are used in a single network, to combine best of both options. Other hyperparameters that can be adjusted are the stride and the type of pooling operation(Goodfellow et al., 2016).

Dense Layers

The dense layers enable the network to learn complex interactions between the multiple vectors that are the output of the convolutional layers. Care has to be taken by determining the depth and the number of neurons in the dense layer part though. Making the network too deep can lead to overfitting.

Output Layers

The form of the output layer depends largely on the prediction task at hand. Binary classification tasks such as the active/non-active predictions just need a single output neuron using binary cross entropy for example. Multiclass classification problems often use multiple neurons with softmax activation functions. For regression problems such as predicting binding affinity, root mean square deviation can be used(Cao & Shen, 2020).

Output and Metrics

Different kind of outputs can have different implications for the model being generated. For example in docking, one can choose to predict the exact binding affinity for multiple ligands per target(Stepniewska-Dziubinska et al., 2018). Or as mentionned before, the model can be used to predict the RMSD of atomic positions of a complex as compared to its unsolved native conformation. The metric that is used to rank prediction models in respect to each other has to be chosen carefully. Feinberg et al. described that the root mean square error does not account for the distribution of the training data, and that this is corrected by the often-used AUC-ROC metric. Though, they still proposed an alternative metric which is called early enrichment. This metric is better suited to score the tails of distributions, which is important in drug discovery since you want to identify the top-performing ligands. Additionally, it can be normalized for the standard deviation of the data set, so that models on different data sets can be compared more reliably(Feinberg et al., 2018).

After the high-level architecture of the network is decided upon, the optimization and learning of the model needs to begin.

3.4 Training procedure

Training includes hyperparameter optimization and weight learning to allow the prediction model to improve its performance over multiple epochs. Different training strategies to achieve this can be constructed. These will influence the maximum performance and the velocity at which the the prediction model can improve.

A lot of attention has to go to splitting the available data into balanced train and test sets. Feinberg et al. make clear that this should mimic the situation in which the prediction model is to be used in reality. Therefore, they introduced a cross validation approach based on agglomerative hierarchical clustering on properties of the proteins. Feinberg et al. tried multiple splits of data to obtain train and test sets, resulting in very different predictive performance for each split. An option to minimize the influence of the possible presence of bias in data sets is to use completely

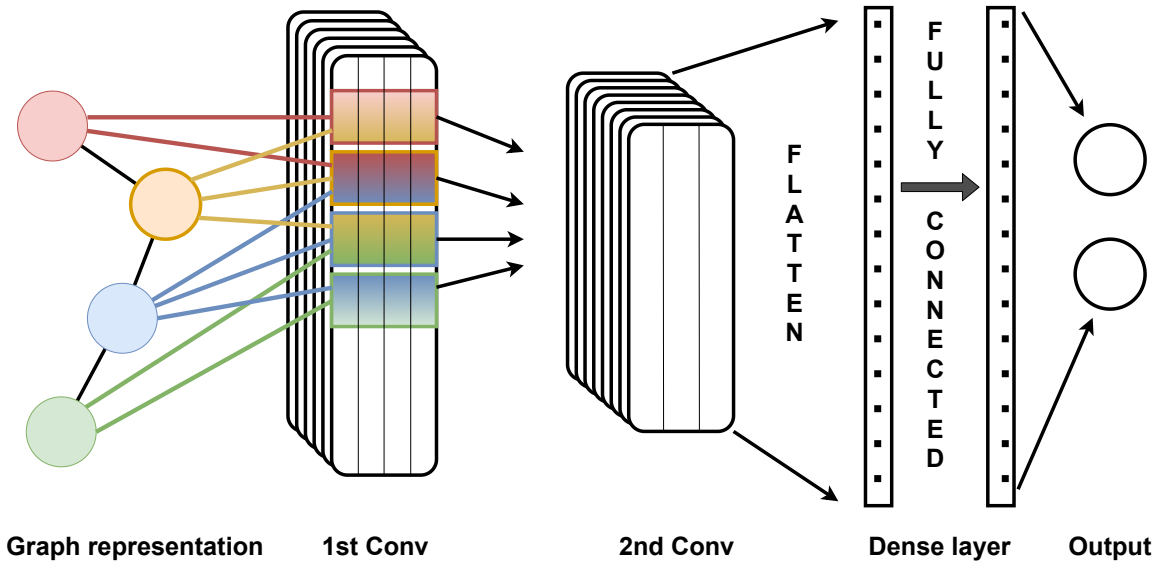


Figure 2: A typical Graph Convolutional Network architecture. In the graph convolutional layers, the network learns about each node's neighboors per stacked layer. In this example, with a stride of 1 neighboor per layer, the 2nd convolutional layer learns about 2 neighboors distanced from the node at hand. As is shown per convolutional layer, multiple filters can be applied (shadows). Between each convolutional layer, typically a max pooling operation is performed (omitted from the figure). Then, the flattened output is fed to a dense layer and subsequently to the output layer.

unrelated data sets for training and testing(Ragoza et al., 2017). However, the data in both sets have to be compatible with each other.

Hyperparameters play a big role in deep learning. Partly because there are so many hyper-parameters to decide upon. Wallach et al. show that a low number of parameters should be considered, since this reduces overfitting and hence improves the learning of more generalizable features. In other words, less complex networks with fewer neurons should be favored over more complex networks(Wallach et al., 2015). Also regularization has to be thought off, as this has shown to decrease the effects of noise for models working with structural and thermodynamic data(Gomes et al., 2017). Regularization keeps the weights of neurons closer to zero compared to when no regularization is applied. Other hyperparameters that need to be considered for optimization are the learning algorithm and coefficient, the weight initialization method, dropout parameter, and batch-related parameters.

Depending on the actual aim of the docking system, a loss function has to be designed so that the deep learning model can update itself. The simplest variant is addressing the prediction task as a binary classification problem, for which binary cross-entropy can be applied as a loss function to predict just two class outcomes (*e.g.* native vs. non-native, or binder vs non-binder). Another approach is to have a continuous outcome to for example predict the exact binding affinity making use of a root-mean-square error(Stepniewska-Dziubinska et al., 2018), or use root-mean square deviation of atomic positions for evaluation of binding poses on a continuous scale.

4 Deep Learning examples

One of the first approaches to deep neural networks using the structures of the components that make up a complex is AtomNet by Wallach et al., which uses data from both the small-molecule and the target. They showed that their deep learning approach could reach better performance on the DUD-E data set(Mysinger et al., 2012) than Smina (variation on Autodock Vina) which is based on an empirical scoring function. They apply convolution on a 3D grid, making use of max pooling to extract local features(Wallach et al., 2015).

Pereira et al. used the output of Autodock Vina and Dock docking systems to generate feature vectors for the atoms of the small molecule based on their structural and chemical proterties. The values of each of these properties are converted into vectors based on learned atom embeddings, and then concatenated into a single feature vector. Via the use of embeddings, they introduced a way to represent properties on a continuous scale. Since the number of atoms that a small molecule consists of can differ, the feature vectors per complex will vary in size. The authors used a global max pooling operation in the convolutional layer, so that for each dimension of the atom feature vectors only the maximum value is used in the output of this layer(Pereira et al., 2016). This means that for every dimension of the atom feature vectors, only 1 atom feature vector value contributes to the resulting feature vector. This approach resulted in better performance on classification of complexes as native or non-native compared to the results that the docking systems could achieve(Pereira et al., 2016).

Gomes et al. used neural networks to predict binding free energy of a complex instead of predicting the nativity on complexes from the PDBBind data set. Three different convolutional networks are learned; one for the complex, one for the target, and one for the ligand. Then, the binding free energy of the complex can be calculated through subtracting the energy of the individual components from the energy of the complex. Hereby, the model needs to incorporate the thermodynamics underlying the binding, though the loss function only considers the predicted and experimentally determined binding free energy and not the energy of each component separately. So theoretically, errors could arise in the 3 individual networks as long as they cancel each other out when the binding free energy is calculated(Gomes et al., 2017).

Ragoza et al. achieved a better performance with their deep learning based model compared to AutoDock Vina on intertarget (rank all poses from all targets) ranking, but this is the other way around for intratarget (rank the poses per target) ranking. For virtual screening on the DUD-E set, the neural network achieved better performance compared to AutoDock Vina in single-pose and

multipose prediction both in the AUC metric as well as in several early enrichment scores(Ragoza et al., 2017). The authors also used a geometric representation of the binding site atoms, with the difference that the atoms of the components are stored in separate channels. They experimented with representing the atoms as a density distribution, but this did not improve the prediction accuracy compared to one-hot encoded atom types. Besides, average pooling as alternative for max pooling prevented the model from learning(Ragoza et al., 2017). Another noticable point is that they assessed the generalizability of the models that were trained through multitask prediction. Using the same model for binding affinity prediction did not lead to high performance, indicating that the model is not very generalizable for this purpose at least. Though, data augmentation in the form of randomly rotating data input structures did lead to less overfitting and higher test performance(Ragoza et al., 2017).

Stepniewska-Dziubinska et al. used the same basis as Ragoza et al., but instead they predicted the exact binding affinity, rather than ranking different ligands for a target. Also, the same atom type encodings were used for the small-molecule and target. A remarkeble point is that they inputted the complex structures as different orientations to the deep learning model, just as you could do for images by rotating them 4 times with 90 degrees for example. They show that although in the first convolutional layers, the activation for rotated input differs, in latter layers this difference is minimal. This suggests that more general features are learned based on the two molecules instead of input orientation, which in turn is expected to lead to better performance on unseen data. The authors managed to outperform classical scoring functions in pose scoring on the CASF-2013 data set and Astex Diverse Set in terms of correlation and RMSE and MAE metrics(Stepniewska-Dziubinska et al., 2018).

Feinberg et al. used a graph representation, instead of a geometric representation (as is used by the papers described above) on the PDBBind 2007 data set(Wang et al., 2004). The innovation about this approach is that it presents an unified framework for biomolecular interactions between the ligand and the target, because both these components are represented by the same atom type representations. This is contrary to other approaches, where the ligand and target are represented by two separately generated abstractions, which are thereafter made to interoperate. The benefit of the approach used by Feinberg et al. is that it seems to be a more natural representation, since both ligand and target are built from the same building blocks in reality as well(*i.e.* atoms). Besides, they include also non-covalent bonds in the adjacency matrix, which is an extension on the use of solely covalent bonds in other papers. They used a gated graph neural network in which gated recurrent units update and propagate each unit's hidden state. Their model achieved better results on the PDBBind 2007 benchmark than RF-Score and X-Score on the Pearson and Spearman correlation metrics(Feinberg et al., 2018).

Gonczarek et al. used a mixed approach of graphs and geometric representation. They split up general atom features and binding features into two vectors. For the target, the binding features were determined by geometric distance, while for the small-molecule, the bonds were one-hot encoded. Then, fingerprints are learned for each component separately through a neural network, before they are used together to calculate the binding potential(Gonczarek et al., 2018). So in this approach, rather than learning a neural network for the two components combined, two separate networks are learned per component. Theoretically, the outputs of these separate networks could also be used for other aims which require structural information about either component. Also a graph and atom convolution implementation were compared, in which the atom convolution variant showed superior performance on the AUC metric. The poses scoring AUC on the DUD-E set outperformed Autodock Smina and the AtomNet model of Wallach et al. (described above) (Gonczarek et al., 2018).

Torng and Altman used separate graphs for the ligand and the binding pockets of protein, which were fed into two different graph convolutional layers to generate fingerprints for each one. Then downstream, the output of these layers would be inputted to an interaction layer, that has the task of learning the binding relations. In this way, an AUC better, but close to the model of Ragoza et al. was reached on the DUD-E data set for predicting whether a small molecule will bind to a target protein(Torng & Altman, 2019) A new invention in deep learning models for docking was the usage

of an autoencoder by Torng and Altman. They used two of such layers to pretrain their eventual model with additional data that would not be suited for training of the actual prediction model. They used an autoencoder layer to pretrain a graph convolutional layer to encode neighborhood graphs of a protein's binding pockets into fixed-size binding pocket fingerprints. The decoder is instructed to reconstruct the neighborhood graphs from the fingerprints(Torng & Altman, 2019). Hereby is ensured that no informative variance is lost by transforming the data into the fingerprints. Then, the encoding layer is copied in the graph convolutional layer of the eventual predictive model. This enables the model to incorporate useful information from a larger variety of binding pockets than just the pockets that are used as input during the prediction task(Torng & Altman, 2019). In this way, the model can be pushed in a direction that favors generalizability by showing it a larger amount of data. Though, this data is not seen anymore during actual training of the entire model, so it does not go further than favorable weight initizialization of the layers it is applied on(Torng & Altman, 2019).

Cao and Shen used a graph representation in which the bonds between atoms were represented as unique codes per atom-pair combination. The binding energy was predicted making use of an adjusted pooling operation based on protein-complex energy in a special energy-based graph convolutional layer. Another special feature of this paper is the addition of a multi-head attention module between the convolutional and dense layers(Cao & Shen, 2020). Attention (which is an component of tranformers) was introduced by Vaswani et al. and allows a network to learn which parts of a vector or sequence are more important (*i.e.* informative) based on similarity for the task at hand(Vaswani et al., 2017). With this model, the authors improved upon scoring performances earlier achieved by the IRAD model on CAPRI sets(Cao & Shen, 2020).

Zamora-Resendiz and Crivelli also used multi-head attention to their graph based approach, in an attempt to make the model insensitive to shifts and truncations of structures in the data. The authors also tried to improve interpretability of the prediction model by looking at the attribution of nodes in the graph (which map to residues), and show that the network is able to recognize secondary structures and other relevant structures for protein function(Zamora-Resendiz & Crivelli, 2019). They compare a 2D CNN method(Zacharaki, 2017) to their graph based approach, and show that in terms of classification accuracy both approaches are on a par, though the 2D one is more often quicker in finishing the job. The authors argue that the graph approach is better suited for interpretation compared to geometric approaches for reasons mentioned above, and that graphs are computationally more efficient compared to the 3D approaches(Zamora-Resendiz & Crivelli, 2019).

Lim et al. used a graph-based neural network in which they try to correct for the model relying on features from each component separately by making use of attention (described above). This is done by subtracting these small-molecule and target-specific features from the features generated for the complex. Thereby, the model can only use features based on the interactions between the two components. They also implemented skip-connections, which is not seen in previous discussed papers(Lim et al., 2019). Skip-connections can improve the performance of networks with a high number of layers since it helps alleviating vanishing gradients(Ryu et al., 2018). Lim et al. claim superior performance of their model on the PDBBind and DUD-E data sets in classification of native and non-native poses compared to the above discussed models of Gonczarek et al.; Ragoza et al.; Torng and Altman; Wallach et al., also their model improves by including attention compared to excluding it(Lim et al., 2019).

Gainza et al. hypothesized that the surfaces of proteins are important for the properties of the molecular interactions. Hence, they take surface patches around the centroid of a protein, and use it as input into their geodesic convolutional layers. This results in fingerprints of those patches, that consequently can be used for interaction prediction tasks based on complementarity or similarity of those fingerprints. Hereby they succeed in generating a 3D representation of the most important areas for molecular interactions, while simplifying less relevant inner-structures of the proteins(Gainza et al., 2020). Gainza et al. show three different implementations of their MaSIF system, which enables researchers to predict ligand binding of proteins and location of binding and binding partners of protein complexes. MaSIF-ligand can identify for a binding pocket which ligand from a set of ligands has the most favorable binding interaction; MaSIF-site predicts

which patches of a protein's surface are most likely to engage in protein interaction; and MaSIF-search uses the surface fingerprints as vectorized descriptors in conjunction with nearest-neighbor techniques to compute the clustering of protein interfaces. Hereby, a wide space of molecules can be scanned for binding partnership much quicker than when traditional docking systems are used that compare one pair at a time and analyze the 3D docking space. Then, additional docking analysis of the partners selected by the scanning algorithm should be performed to rerank the binders with higher accuracy(Gainza et al., 2020). Although the two latter implementations are aimed at protein-protein interaction, these might be useful for small-molecule protein interaction as well if some adjustments are made. The authors show that MaSIF-search virtual screening performance is similar to the docking systems ZDock, PatchDock and ZDock together with ZRank2, though MaSIF-search is much quicker in obtaining those results. Depending on the data and settings, the computing times range from minutes for MaSIF-search to days for the other docking systems(Gainza et al., 2020)

5 Conclusion

In this review, applications of deep learning techniques to molecular docking problems have been described. Molecular docking is a computer technique that can assist the drug development and design process. It comprises of two main phases, sampling and scoring. Although current docking systems obtain a reasonable sampling performance compared to the scoring performance, deep learning techniques can play a role at both phases. Moreover because more data about the structures of complexes and their apo components comes available, the potential of deep learning techniques gets utilized better.

Different data representation approaches have been discussed in more detail. These can be broadly divided into two categories: graph based and 3D geometric based representations. The earlier deep learning implementations for docking mostly used geometric (3D) representations. Logically, a resolution has to be set, and all points in the 3D space needs to be populated with data. Recently, graph approaches were used more often, which seem to resemble the structure of molecules as we chemically represent them: atoms (nodes) connected by bonds (edges). Graph approaches also show some computational advantages since they are more efficient in their data representation. However, the more recent geometric approaches used techniques to simplify the representation, mainly by focussing on contact patches rather than the complete 3D structure of proteins. In this way, the models were able to become less complex, and could hence achieve comparable or better results on limited amounts of data compared to earlier models.

Different aims in docking can be fulfilled by using deep learning techniques. Some models try to score and rank poses, while others try to predict the exact binding affinity of poses. In some papers, the poses generated by other docking systems were used to apply scoring on, but it is also possible to use a deep learning model for both sampling and scoring.

All papers used a form of convolutional layers to make the model learn to recognize local features of importance for the prediction task at hand. Some used rather recent innovations in deep learning architectures such as attention and skip-connections. Multi-task prediction was also tried, but this did not always result in increased performance compared to single-task prediction.

Although, there are many challenges regarding the publicly available data sets, progress has been made by constructing modified data sets and new ways of dealing with the shortcomings have been proposed in several papers. This includes validation of models on data sets completely different than used for training to avoid bias specific to the data set at hand.

Currently, perhaps the greatest challenge of this field is the low availability and quality of data. This could get better rather sooner than later because in recent years more and more data became available in several kinds of domains. The challenge remains to make sure that this data is of sufficient quality. Also, the publication bias (only positive results are published) remains a problem in docking since there is a need for reliable data about decoys(Réau et al., 2018).

When more data becomes available, perhaps the next challenge will be to find a good balance with simplification of the data so that the computational costs do not become too heavy, and

oversimplifying and thereby losing important information.

Acknowledgements

I thank Manon Réau for supervising me during the writing process and providing me with extensive feedback on the progress. I also thank Alexandre Bonvin for acting as examinor and suggesting this topic plus providing feedback. Finally, thanks to Toine Egberts for acting as second examiner, and the swift communications.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, *28*(1), 235–242. https://doi.org/10.1093/nar/28.1.235

Cao, Y., & Shen, Y. (2020). Energy-based graph convolutional networks for scoring protein docking models. *Proteins: Structure, Function, and Bioinformatics*, *88*(8), 1091–1099. https://doi.org/10.1002/prot.25888
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25888

Chaput, L., & Mouawad, L. (2017). Efficient conformational sampling and weak scoring in docking programs? Strategy of the wisdom of crowds. *Journal of Cheminformatics*, *9*. https://doi.org/10.1186/s13321-017-0227-x

Chen, Y.-C. (2015). Beware of docking! *Trends in Pharmacological Sciences*, *36*(2), 78–95. https://doi.org/10.1016/j.tips.2014.12.001

Corbeil, C. R., Williams, C. I., & Labute, P. (2012). Variability in docking success rates due to dataset preparation. *Journal of Computer-Aided Molecular Design*, *26*(6), 775–786. https://doi.org/10.1007/s10822-012-9570-1

Duy Nguyen, D., Cang, Z., & Wei, G.-W. (2020). A review of mathematical representations of biomolecular data. *Physical Chemistry Chemical Physics*, *22*(8), 4343–4367. https://doi.org/10.1039/C9CP06554G

Feinberg, E. N., Sur, D., Wu, Z., Husic, B. E., Mai, H., Li, Y., Sun, S., Yang, J., Ramsundar, B., & Pande, V. S. (2018). PotentialNet for Molecular Property Prediction. *ACS Central Science*, *4*(11), 1520–1530. https://doi.org/10.1021/acscentsci.8b00507

Gainza, P., Sverrisson, F., Monti, F., Rodolà, E., Boscaini, D., Bronstein, M. M., & Correia, B. E. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, *17*(2), 184–192. https://doi.org/10.1038/s41592-019-0666-6

Gomes, J., Ramsundar, B., Feinberg, E. N., & Pande, V. S. (2017). Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. *arXiv:1703.10603 [physics, stat]*.

Gonczarek, A., Tomczak, J. M., Zaręba, S., Kaczmar, J., Dąbrowski, P., & Walczak, M. J. (2018). Interaction prediction in structure-based virtual screening using deep learning. *Computers in Biology and Medicine*, *100*, 253–258. https://doi.org/10.1016/j.compbiomed.2017.09.007

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Huang, N., Shoichet, B. K., & Irwin, J. J. (2006). Benchmarking Sets for Molecular Docking. *Journal of Medicinal Chemistry*, *49*(23), 6789–6801. https://doi.org/10.1021/jm0608356

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc.

Li, J., Fu, A., & Zhang, L. (2019). An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. *Interdisciplinary Sciences: Computational Life Sciences*, *11*(2), 320–328. https://doi.org/10.1007/s12539-019-00327-w

Lim, J., Ryu, S., Park, K., Choe, Y. J., Ham, J., & Kim, W. Y. (2019). Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *Journal of Chemical Information and Modeling*, *59*(9), 3981–3988. https://doi.org/10.1021/acs.jcim.9b00387

Meng, X.-Y., Zhang, H.-X., Mezei, M., & Cui, M. (2011). Molecular Docking: A Powerful Approach for Structure-Based Drug Discovery. *Current Computer - Aided Drug Design*, *7*(2), 146–157. https://doi.org/10.2174/157340911795677602

Mysinger, M. M., Carchia, M., Irwin, J. J., & Shoichet, B. K. (2012). Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, *55*(14), 6582–6594. https://doi.org/10.1021/jm300687e

Pagadala, N. S., Syed, K., & Tuszynski, J. (2017). Software for molecular docking: A review. *Biophysical Reviews*, *9*(2), 91–102. https://doi.org/10.1007/s12551-016-0247-1

Pereira, J. C., Caffarena, E. R., & dos Santos, C. N. (2016). Boosting Docking-Based Virtual Screening with Deep Learning. *Journal of Chemical Information and Modeling*, *56*(12), 2495–2506. https://doi.org/10.1021/acs.jcim.6b00355

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., & Koes, D. R. (2017). Protein–Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, *57*(4), 942–957. https://doi.org/10.1021/acs.jcim.6b00740

Réau, M., Langenfeld, F., Zagury, J.-F., Lagarde, N., & Montes, M. (2018). Decoys Selection in Benchmarking Datasets: Overview and Perspectives. *Frontiers in Pharmacology*, *9*. https://doi.org/10.3389/fphar.2018.00011

Rodrigues, J. P. G. L. M., & Bonvin, A. M. J. J. (2014). Integrative computational modeling of protein interactions. *The FEBS Journal*, *281*(8), 1988–2003. https://doi.org/10.1111/febs.12771
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/febs.12771

Ryu, S., Lim, J., Hong, S. H., & Kim, W. Y. (2018). Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network. *arXiv:1805.10988 [cs, stat]*.

Shen, C., Ding, J., Wang, Z., Cao, D., Ding, X., & Hou, T. (2020). From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *WIREs Computational Molecular Science*, *10*(1), e1429. https://doi.org/10.1002/wcms.1429
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1429

Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., & Siedlecki, P. (2018). Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, *34*(21), 3666–3674. https://doi.org/10.1093/bioinformatics/bty374

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. *Proceedings of the IEEE International Conference on Computer Vision*, 843–852.

Torng, W., & Altman, R. B. (2019). Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *Journal of Chemical Information and Modeling*, *59*(10), 4131–4149. https://doi.org/10.1021/acs.jcim.9b00628

Tripathi, A., & Bankaitis, V. A. (2017). Molecular Docking: From Lock and Key to Combination Lock. *Journal of molecular medicine and clinical applications*, *2*(1).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 5998–6008). Curran Associates, Inc.

Virshup, A. M., Contreras-García, J., Wipf, P., Yang, W., & Beratan, D. N. (2013). Stochastic Voyages into Uncharted Chemical Space Produce a Representative Library of All Possible Drug-Like Compounds. *Journal of the American Chemical Society*, *135*(19), 7296–7303. https://doi.org/10.1021/ja401184g

Wallach, I., Dzamba, M., & Heifets, A. (2015). AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. *arXiv:1510.02855 [cs, q-bio, stat]*.

Wang, R., Fang, X., Lu, Y., & Wang, S. (2004). The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry*, *47*(12), 2977–2980. https://doi.org/10.1021/jm030580l

Zacharaki, E. I. (2017). Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ Computer Science*, *3*, e124. https://doi.org/10.7717/peerj-cs.124

Zamora-Resendiz, R., & Crivelli, S. (2019). Structural Learning of Proteins Using Graph Convolutional Neural Networks. *bioRxiv*, 610444. https://doi.org/10.1101/610444

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: A comprehensive review. *Computational Social Networks*, *6*(1), 11. https://doi.org/10.1186/s40649-019-0069-y