

# Machine Learning for Kinase Drug Discovery

**Dissertation**

zur Erlangung des Grades eines  
Doktors der Naturwissenschaften  
am Fachbereich Mathematik und Informatik  
der Freien Universität Berlin

Talia B. Kimber

Berlin, 2022

**Erstgutachter:**

Prof. Dr. Frank Noé  
Fachbereich Mathematik und Informatik  
Freie Universität Berlin  
Arnimallee 12  
14195 Berlin

**Zweitgutachterin:**

Prof. Dr. Andrea Volkamer  
Institut für Physiologie  
Charité - Universitätsmedizin Berlin  
Charitéplatz 1  
10117 Berlin

**Tag der Disputation:**

13. Februar 2023

*Dedicated to Judith Fillinger.*



# Abstract

Cancer is one of the major public health issues, causing several million losses every year. Although anti-cancer drugs have been developed and are globally administered, mild to severe side effects are known to occur during treatment. Computer-aided drug discovery has become a cornerstone for unveiling treatments of existing as well as emerging diseases. Computational methods aim to not only speed up the drug design process, but to also reduce time-consuming, costly experiments, as well as *in vivo* animal testing. In this context, over the last decade especially, deep learning began to play a prominent role in the prediction of molecular activity, property and toxicity.

However, there are still major challenges when applying deep learning models in drug discovery. Those challenges include data scarcity for physicochemical tasks, the difficulty of interpreting the prediction made by deep neural networks, and the necessity of open-source and robust workflows to ensure reproducibility and reusability.

In this thesis, after reviewing the state-of-the-art in deep learning applied to virtual screening, we address the previously mentioned challenges as follows: Regarding data scarcity in the context of deep learning applied to small molecules, we developed data augmentation techniques based on the SMILES encoding. This linear string notation enumerates the atoms present in a compound by following a path along the molecule graph. Multiplicity of SMILES for a single compound can be reached by traversing the graph using different paths. We applied the developed augmentation techniques to three different deep learning models, including convolutional and recurrent neural networks, and to four property and activity data sets. The results show that augmentation improves the model accuracy independently of the deep learning model, as well as of the data set size. Moreover, we computed the uncertainty of a model by using augmentation at inference time. In this regard, we have shown that the more confident the model is in its prediction, the smaller is the error, implying that a given prediction can be trusted and is close to the target value. The software and associated documentation allows making predictions for novel compounds and have been made freely available.

Trusting predictions blindly from algorithms may have serious conse-

quences in areas of healthcare. In this context, better understanding how a neural network classifies a compound based on its input features is highly beneficial by helping to de-risk and optimize compounds. In this research project, we decomposed the inner layers of a deep neural network to identify the toxic substructures, the toxicophores, of a compound that led to the toxicity classification. Using molecular fingerprints —vectors that indicate the presence or absence of a particular atomic environment —we were able to map a toxicity score to each of these substructures. Moreover, we developed a method to visualize in 2D the toxicophores within a compound, the so-called cytotoxicity maps, which could be of great use to medicinal chemists in identifying ways to modify molecules to eliminate toxicity. Not only does the deep learning model reach state-of-the-art results, but the identified toxicophores confirm known toxic substructures, as well as expand new potential candidates.

In order to speed up the drug discovery process, the accessibility to robust and modular workflows is extremely advantageous. In this context, the fully open-source TeachOpenCADD project was developed. Significant tasks in both cheminformatics and bioinformatics are implemented in a pedagogical fashion, allowing the material to be used for teaching as well as the starting point for novel research. In this framework, a special pipeline is dedicated to kinases, a family of proteins which are known to be involved in diseases such as cancer. The aim is to gain insights into off-targets, i.e. proteins that are unintentionally affected by a compound, and that can cause adverse effects in treatments. Four measures of kinase similarity are implemented, taking into account sequence, and structural information, as well as protein-ligand interaction, and ligand profiling data. The workflow provides clustering of a set of kinases, which can be further analyzed to understand off-target effects of inhibitors. Results show that analyzing kinases using several perspectives is crucial for the insight into off-target prediction, and gaining a global perspective of the kinome.

These novel methods can be exploited in the discovery of new drugs, and more specifically diseases involved in the dysregulation of kinases, such as cancer.

# Acknowledgments

In preamble to this thesis, I would like to thank and express my gratitude to the many people who were involved in this journey.

Firstly, I would like to thank Andrea Volkamer and John Chodera for finding the financial resources for me to embark on the PhD adventure. Without their support and dedication, none of it would have been possible. Moreover, I would like to thank Frank Noé for agreeing to be my supervisor from the Freie Universität, and following my progress from afar. A special thank you to Prof. Reinert and Dr. Lluís Raich for agreeing to be part of the PhD committee.

I would like to express my gratitude to the people in the Volkamer lab and Chodera lab, more specifically, Andrea Morger, Jaime Rodríguez-Guerra, David Schaller, Corey Taylor and Yonghui Chen, and Ivy Zang, Josh Fass, and Yuanqing Wang.

I would like to thank Greg Landrum for his investment and availability related to the RDKit, and the FMP (Leibniz-Forschungsinstitut für Molekulare Pharmakologie) for providing us with the cytotoxicity data set. A special thank you to Antonija Burcul for her mentorship during the last few months.

The PhD endeavour would not have been possible without the support of my family and friends: Beatrice, Anying and Christian, Teri and Jan, Kelly, Daphné, Lidia, Justine, Rosella, Barbara. Thank you so much to Sineth and Pascal for allowing me to spend some time in their heavenly chalet! I am forever grateful for the support of Katrin Kurze. I highly appreciate the support from my brother Kevin and his family, Allison, and Lia. A very special thank you to my father, John Kimber, who was wholeheartedly involved in the L<sup>A</sup>T<sub>E</sub>X typesetting of this manuscript.

The whole adventure was only possible thanks to my incredible safety net of women. Firstly, Dominique Sydow, who started as an esteemed colleague, trustworthy, and reliable, who became one of my best friends and eventually work soul-mate, supporting me so closely through this bumpy journey. Secondly, my dearest mother, Iris Fillinger, the ultimate caretaker, and listener, constantly available, and giving the best advice. Thirdly, the wonder woman that is my sister, Leah Kimber, always supportive, always finding the right words to make me see the positive side, motivating and guiding me.

And finally, the most important person throughout this process, the per-

son who picked me up at the end of each day, who dried my tears, saw me during my darkest hours but yet has stood by my side, and supported me, my best friend, my colleague, my climbing partner, my life partner, Maxime Gagnebin.



# Publications

The results of this work were published in

- Talia B. Kimber, Yonghui Chen, and Andrea Volkamer. Deep learning in virtual screening: recent applications and developments. *International Journal of Molecular Sciences*, 22(9):4435, 2021. ISSN 1422-0067. URL <https://doi.org/10.3390/ijms22094435>
- Talia B. Kimber, Maxime Gagnebin, and Andrea Volkamer. Maxsmi: Maximizing molecular property prediction performance with confidence estimation using SMILES augmentation and deep learning. *Artificial Intelligence in the Life Sciences*, 1:100014, 2021. ISSN 2667-3185. URL <https://doi.org/10.1016/j.aillsi.2021.100014>
- Henry E. Webel, Talia B. Kimber, Silke Radetzki, Martin Neuenschwander, Marc Nazaré, and Andrea Volkamer. Revealing cytotoxic substructures in molecules using deep learning. *Journal of Computer-Aided Molecular Design*, 34(7):731–746, 2020. URL <https://doi.org/10.1007/s10822-020-00310-4>
- Dominique Sydow, Jaime Rodríguez-Guerra, Talia B. Kimber, David Schaller, Corey J. Taylor, Yonghui Chen, Mareike Leja, Sakshi Misra, Michele Wichmann, Armin Ariamajd, and Andrea Volkamer. TeachOpenCADD 2022: open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research. *Nucleic Acids Research*, 50(W1):W753–W760, 05 2022. ISSN 0305-1048. URL <https://doi.org/10.1093/nar/gkac267>
- Talia B. Kimber, Dominique Sydow, and Andrea Volkamer. Kinase Similarity Assessment Pipeline for Off-Target Prediction [Article v1.0]. *Living Journal of Computational Molecular Science*, 3(1):1599, Jun 2022. URL <https://doi.org/10.33011/livecoms.3.1.1599>



# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Publications</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Deep learning today . . . . .	1
1.1.1 Deep learning in the landscape of machine learning . .	2
1.1.2 The success of deep learning . . . . .	2
1.2 Machine learning in drug design . . . . .	4
1.2.1 Drug design . . . . .	6
1.2.2 Computer-aided drug design . . . . .	6
1.2.3 Machine learning-based challenges in drug design . . .	8
1.3 Kinases as drug targets . . . . .	10
1.4 The goal of this thesis . . . . .	13
<b>2 Deep Learning in Virtual Screening: Recent Applications and Developments</b>	<b>17</b>
2.1 Introduction . . . . .	18
2.1.1 Virtual screening . . . . .	18
2.1.2 Machine learning and deep learning . . . . .	20
2.1.3 Data availability and big data . . . . .	21
2.1.4 Deep learning in virtual screening . . . . .	22
2.2 Methods & Data . . . . .	22
2.2.1 Encodings in virtual screening . . . . .	22
2.2.2 Deep learning models in virtual screening . . . . .	30
2.2.3 Data sets and benchmarks in virtual screening . . . . .	34
2.3 Recent developments . . . . .	39
2.3.1 Complex-based models . . . . .	39
2.3.2 Pair-based models . . . . .	45
2.4 Conclusion and discussion . . . . .	49

<b>3</b>	<b>Improving molecular property prediction using data augmentation</b>	<b>55</b>
3.1	Introduction . . . . .	56
3.2	Methods . . . . .	58
3.2.1	Augmentation strategies . . . . .	59
3.2.2	SMILES augmentation as ensemble learning for compound prediction and confidence measure . . . . .	60
3.2.3	Deep learning models . . . . .	61
3.3	Data and experimental setup . . . . .	62
3.3.1	Provenance . . . . .	63
3.3.2	Data preprocessing and input featurization . . . . .	63
3.3.3	Important steps in SMILES augmentation . . . . .	64
3.3.4	Experimental setup and model evaluation . . . . .	65
3.3.5	Code and documentation . . . . .	66
3.4	Results and discussion . . . . .	66
3.4.1	SMILES augmentation improves model performance . . . . .	66
3.4.2	Ensemble learning for compound prediction and confidence measure . . . . .	72
3.4.3	Comparison to other studies . . . . .	75
3.4.4	Test case: EGFR affinity data following the guideline . . . . .	77
3.4.5	Maxsmi models available for user predictions . . . . .	78
3.5	Conclusion . . . . .	79
<b>4</b>	<b>Interpreting model prediction for cytotoxicity</b>	<b>81</b>
4.1	Introduction . . . . .	82
4.2	Data & Methods . . . . .	84
4.2.1	Data . . . . .	84
4.2.2	Machine Learning Model Generation . . . . .	86
4.2.3	Deep Taylor Decomposition . . . . .	87
4.2.4	Identification of Toxicophores and Visualization as Cytotoxicity Maps . . . . .	88
4.2.5	Used Software and Libraries . . . . .	91
4.3	Results and Discussion . . . . .	91
4.3.1	Model Evaluation and Comparison . . . . .	92
4.3.2	Potential Toxicophores . . . . .	95
4.4	Conclusion . . . . .	102
<b>5</b>	<b>Kinase-centric drug design: the importance of pipelines in drug campaigns</b>	<b>105</b>
5.1	TeachOpenCADD 2022: Open-Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research . . . . .	108
5.1.1	Introduction . . . . .	108
5.1.2	New talktorials . . . . .	110

5.1.3	Best practices . . . . .	116
5.1.4	TeachOpenCADD usage . . . . .	116
5.1.5	Conclusion . . . . .	117
5.1.6	Code and data availability . . . . .	118
5.2	Kinase similarity assessment pipeline for off-target prediction	119
5.2.1	Introduction . . . . .	119
5.3	Prerequisites . . . . .	120
5.4	Method . . . . .	123
5.5	Pipeline . . . . .	129
5.6	Conclusion . . . . .	132
<b>6</b>	<b>Conclusion</b>	<b>135</b>
<b>A</b>	<b>Evaluation strategies and metrics</b>	<b>139</b>
<b>B</b>	<b>Figures</b>	<b>141</b>
<b>C</b>	<b>Tables</b>	<b>149</b>
	<b>Acronyms</b>	<b>151</b>
	<b>Bibliography</b>	<b>154</b>
	<b>Zusammenfassung</b>	<b>205</b>



# Chapter 1

## Introduction

### 1.1 Deep learning today

Machine learning, and more specifically deep learning, has penetrated every aspect of our society, from art to science and technology. Colossal businesses, such as Amazon [6] or Netflix [7], use deep learning algorithms extensively and thrive on it. For example, they are able to suggest to clients new movies or TV shows in the case of Netflix, or new purchases in the case of Amazon, that fit nearly perfectly the client's taste. But how about science, and more specifically healthcare and drug design?

Artificial Intelligence (AI) builds algorithms that enable them to solve human tasks [8], and it is believed that AI is revolutionizing science [9]. To some extent, it is. A notable example is AlphaFold, the AI system developed by Jumper et al. [10] to tackle the protein structure prediction problem. Proteins, macromolecules made of chains of amino acids, are involved in several aspects of living organisms, such as catalysis and cell signaling. The structure—the three dimensional atomic coordinates—plays a crucial role in the function of a protein. The structure prediction problem, which emerged in the 1960s, was classified as one of the greatest challenges to be solved in the computational sciences [11] and aims at determining the structure of a protein based on its amino acid sequence. In 2021, Jumper et al. [10] won the 14th Critical Assessment of protein Structure Prediction (CASP14) [12] by training extensively a deep neural network and therefore accurately solving the protein structure prediction problem.

In this chapter, we introduce the concepts of machine learning and deep learning, discuss the main reasons why deep learning has become such a great success, and explore how it is being applied to drug design. More specifically, we see how *in silico* methods—methods relying on computer simulations—can be used to improve drug design. We also discuss the challenges and shortcomings that are inherent to deep learning models. Finally, we present the goals and objectives of this thesis.

### 1.1.1 Deep learning in the landscape of machine learning

Machine learning (ML) aims at computationally learning a task from data by optimizing a performance measure. In this sense, ML is an approach to AI [13]. Within ML, there exist three main categories [14]: (1) **Unsupervised learning**, in which the goal is to find patterns of the underlying structure and gain interpretability of the data. (2) **Reinforcement learning**, in which an agent evolves in an environment and uses the data learned from experience. (3) **Supervised learning**, in which an algorithm is trained on inputs to predict some labeled output. The latter is the focus of this section.

Traditional supervised ML methods follow the idea that given some data, a predictive model is constructed by optimizing the difference between a given labeled output and the output predicted by the model. Some of these methods date back to the last century. For example, neural networks were first developed in the 60s by Rosenblatt [15]. Later, in the 80s, Breiman et al. [16] published the book *Classification and regression trees*. In the 90s, Cortes and Vapnik [17] introduced support vector machines (SVMs) and more recently, in the early 2000s, Breiman [18] proposed random forests (RFs).

Deep learning (DL) [13] is a subset of ML in which the input features are combined using hidden layers that constitute a network. Each hidden layer is made up of a linear and a non-linear part, the non-linear part called the activation function. The information then propagates through the network. Figure 1.1 displays a high level abstraction of a neural network with three hidden layers. The resulting predictive model is highly flexible and is able to extract complex patterns thanks to the non-linearities. More details on the types of neural networks are discussed in Section 2.2.2.

### 1.1.2 The success of deep learning

Over the last decade, words such as "machine learning", "deep learning", and "AI", have been used in many situations, from scientific articles to conferences, newspapers, blog posts, podcasts, and social media in general. The attraction to deep learning, which led to its popularity, may be explained by several factors:

1. **Computing power:** Over the last few years, computing technologies have evolved rapidly. Since the first GPU (Graphics Processing Unit) in 1999, Nvidia [19] and other companies have created more powerful GPUs. Commercially available and similarly priced models underwent a twenty-fold increase in processing power, such as Nvidia's GTX 480 in 2010 vs. RTX 3070 in 2020. In 2015, Google developed TPUs (Tensor Processing Units), a technology adapted for their deep learning framework TensorFlow [20]. This hardware evolution accelerated the training of deep neural networks extensively. For example, in the case of the well-known neural network AlexNet [21], training takes 1.5



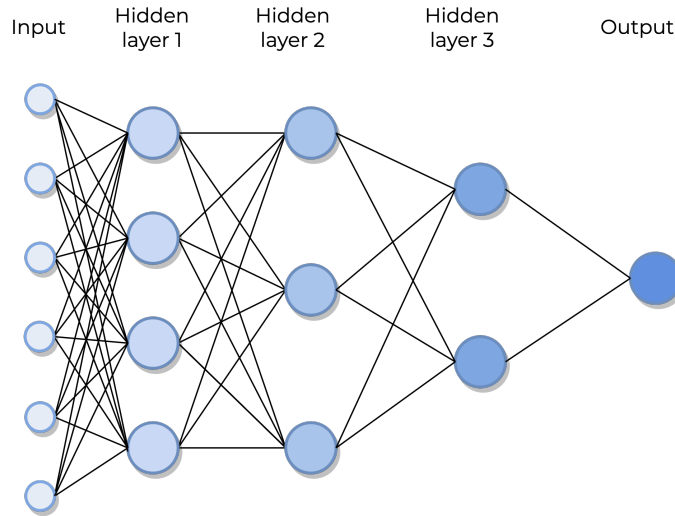


Figure 1.1: **Abstraction of a deep neural network.** The inputs are linearly combined, activation functions applied iteratively to produce an output. In this figure, the input vector contains six units, there are three hidden layers of four, three, and two units respectively, to produce a single value output.

seconds per mini-batch with a CPU (Central Processing Unit) with 8 threads, while only 0.042 seconds with a single GPU [22, Table 7]. Although the hardware necessary for large scale experiments, such as the one from AlphaFold, is extremely expensive, multiple options now exist to access GPUs or TPUs for free, for instance Google Colab [23]. This makes developing and testing deep learning models available to the whole world.

2. **Data:** Over the last decades, data has become abundant. Firstly because of digitalization; what may have been stored on paper in the last century has now been digitized, making data available for processing and analysis. Secondly, data collection has been automatized, increasing the amount of data gathered every day. Lastly, data storage is also increasing; current hardware, such as external hard drives, can store dozens of TB (terabyte) [24]. Cloud storage has also become popular, allowing easy remote access to files and data [25].
3. **Research:** The basic machine learning framework is very generic and allows input of very varied forms. This motivated numerous fields to research how to integrate machine learning methods into their workflow. In both academia and in industry, labs around the world have developed data and models that fit their needs. Moreover, the algorithms

themselves have improved. As described by Chollet [26], first the activation functions, the weight initialization schemes, and the optimization schemes and then batch normalization and residual connections improved back-propagation. More details on deep neural networks are given in Chapter 2.

4. **Software development:** Multiple algorithms necessary for model training and testing have been developed since the 60s [15–17] and are now fully integrated in easy-to-use libraries such as Scikit-learn [27], Keras [28], PyTorch [29], and TensorFlow [20]. In-built functionalities allow the building and training of deep neural networks in just a few lines of code, such as depicted in Keras’ documentation [28]. Not only are these libraries well-documented and functional, but also open-source and free. Additionally, GPU-centric languages such as CUDA [30] are well-documented for frameworks like TensorFlow and PyTorch. This simplifies greatly the training of models on GPUs. Although Google’s platform TensorFlow was the leading framework for deep learning in 2015, PyTorch has recently experienced a tremendous boost and has been used over TensorFlow in research since 2019, as shown in Figure 1.2 taken from He [31]. The shift probably comes from the simplicity of PyTorch, which fits the Python ecosystem, resembling Numpy [32], as well as a stable, well-designed API (application programming interface).
5. **Growing community:** The communities built around data science and AI in general have blossomed around the world. Just to name a few, Kaggle [33] is one of the leading platforms for all levels of machine learning challenges. Hugging Face [34] is an AI community which offers a wide variety of data sets and models. And the WiMLDS (Women in Machine Learning and Data Science) organization promotes worldwide women and gender minorities in machine learning, setting up hackathons, networking events and workshops.

Considering all the above-mentioned points, there is no surprise that deep learning, and more generally machine learning, have gained so much popularity over the last few years.

## 1.2 Machine learning in drug design

Machine learning has been integrated into several areas of healthcare, among which drug design campaigns are no exception.

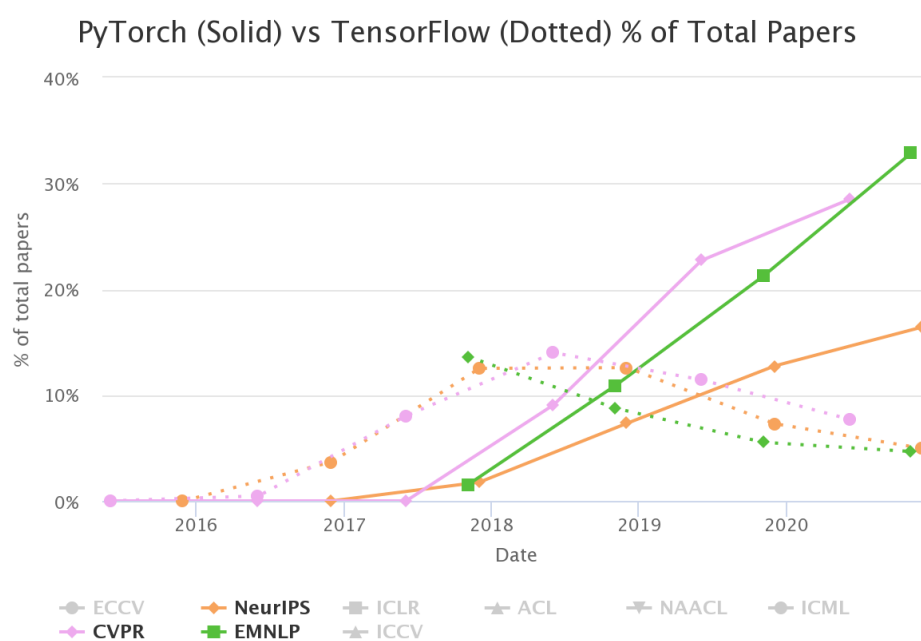


Figure 1.2: **PyTorch getting more popular than TensorFlow.** The two well-known deep learning frameworks have diverging trends. While TensorFlow was widely popular in 2018, PyTorch has risen since. The figure is taken from He [31].

### 1.2.1 Drug design

Rational drug design, as opposed to serendipitous discovery based on trial and error, is the process of designing a drug for a disease, and it commonly consists of two main stages: drug discovery and drug development [35], which themselves can be divided into different steps, described below:

Firstly, a biological target, referring to a protein under investigation, which is often dysregulated in a disease, is identified. This step is usually referred to as target identification and validation [36]. Secondly, a set of molecules that could alter the behavior of the target are selected. These molecules, considered active against the target, may come from an existing database of synthesized substances or a number of other "hit finding" strategies. Among these molecules, a subset of the most promising ones is selected for further investigation, the lead molecules. These are altered to fit even better the target at hand, commonly known as optimized lead compounds. A bioactive compound refers to a molecule with a biological effect, and can be experimentally identified using, for example, target-based assays [37]. Finally, the selected molecules are tested for their behavior in an organism disease model and for the following properties relevant to human pharmacology: absorption, distribution, metabolism, excretion, and toxicity, also known as ADMET [38]. This cycle completes the drug discovery stage. The end goal is to produce drug candidates that bind well to the target, that are efficient, non-toxic, and with the least potential for side effects (selective).

The second part of drug campaigns deals with the development of the drug, involving further animal preclinical testing, regulatory filings, clinical trials, market authorization, and finally reaching manufacture and market access.

The whole process is time-consuming as well as costly: on average, it takes 15 years for a drug to reach the market, and close to \$5 billion is spent by pharmaceutical companies on the design and development of drugs when the cost of failures is factored in [39]. Moreover, toxicity checks require *in vivo* animal testing, for which *in silico* alternatives should be sought, allowing to reduce the number of animal tests.

### 1.2.2 Computer-aided drug design

In this context, computer-aided drug design can be extremely advantageous. *In silico* methods assist the drug design process by modeling certain stages, reducing time, costs, and the required amount of animal testing [40]. In the following, we describe how computational methods can be applied to drug design.

The structure of a protein corresponds to the 3D atomic coordinates and determines its biological function. The Protein Data Bank (PDB) database [41, 42] stores protein structures determined using technologies such as X-

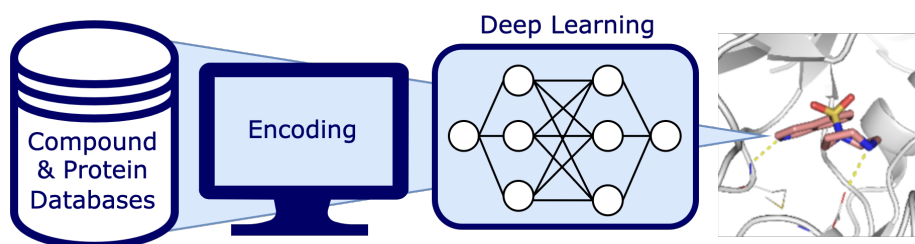


Figure 1.3: **Virtual screening workflow.** Starting from large databases of molecules, compounds and proteins are numerically encoded, permitting the application of deep learning models for activity prediction. Virtual screening allows the rapid selection of promising molecules. The figure is adapted from Kimber et al. [1].

ray crystallography, nuclear magnetic resonance (NMR) [43], or cryo-electron microscopy (cryo-EM) [44]. When selecting a protein structure for further investigation, the structure should be 1. as high resolution as possible in the case of an X-ray structure, 2. it should be as complete as possible, i.e. no missing residues, 3. and, presuming the target of interest is not a mutant, the structure should also be a wildtype protein, i.e. a protein with no mutated residues. However, if no structure exists for a given target, homology modeling [45] is a computational method that produces a structure based on the amino acid sequence of the target and another protein template for which a structure does exist. This provides a first example of the necessity of computational models in drug design.

In the second stage of drug discovery, in the case of the selection of promising molecules, high-throughput screening [46] is a lab experiment that tests properties of compounds, such as activity against a given target. Although it can be efficient (millions of compounds can be screened using robotics), these large scale experiments are expensive and not all targets may be suitable for high-throughput screening. Alternatively, virtual screening can be applied. This method aims at computationally screening large databases of molecules, and using machine learning models, predicts the ones that are potentially active against a given target. Such a method allows to rapidly select, for further investigation and testing, compounds at minimal cost. Figure 1.3, adapted from [1], displays a simplified virtual screening workflow. Virtual screening is commonly split into two categories: 1. ligand-based methods that focus on the ligand, i.e. a small molecule that binds to a macromolecule forming a complex. These methods, also commonly referred to as quantitative structure-activity relationship (QSAR) modeling [47, 48], have become a cornerstone in computer-aided drug discovery. 2. structure-based methods, for which a target structure is needed. In struc-

ture-based drug design, it is believed that structure, the three dimensional geometry, holds more information than sequence, the linear enumeration of the amino acids. Moreover, within structure-based methods, molecular docking [49] is a modeling technique that predicts the position and orientation of a small molecule, when it is bound to a protein, and ranks the compounds that are likely to bind the target. As discussed in greater detail in Section 2.1.1, this split can be further refined and take into account not only the ligand-based and the structure-based methods, but also pair-based methods, which lie at the intersection, treating the ligand and the target as two independent entities.

Finally, *in silico* methods can also be applied to toxicity predictions [50]. As shown in Chapter 4, machine learning models have been trained to identify the toxicity of certain substructures of molecules. Being able to assess the toxic behavior of compounds early in the drug discovery process could help to reduce additional *in vivo* experiments.

Although research has shown the efficacy of these methods, there are still challenges that remain, which are discussed in the next section.

### 1.2.3 Machine learning-based challenges in drug design

Computer-aided drug design is often used in drug campaigns to guide the design process and help to prioritize and optimize the most promising molecules. However, no model is perfect, and in this section we discuss challenges that are faced when applying machine learning models to drug design.

1. **Data scarcity:** While we are living in the era of "big data", and public data sets containing several million data points are freely available (see Section 1.1.2), the contrast with drug design is evident. While the Open Images V4 data set [51] used for image classification contains over 9 million images, the size of the data sets that contain a molecule, label pair are much smaller. For example, the ESOL data [52, 53], which consists of measured water solubility and is important in drug design for the distribution of a compound in an organism, contains solely 1,128 data points. One reason that could explain the data deficit such as ESOL is that obtaining values from experimental data requires sophisticated machinery in labs, expertise, and manpower, which in turn is expensive and time-consuming. Approaches to overcome this challenge have been developed through, for example, data augmentation techniques. More details on this topic are given in Chapter 3. However, not every data set exploited in drug design is so limited and there are databases that store large amounts of bio-measurements data. For example, the latest version to date of ChEMBL [54] (version 30) contains 14,855 targets, 2,157,379 distinct compounds, and 19,286,751 activities [55], and is commonly used for activity prediction [56]. Al-

though containing large amounts of data, the coverage per target varies greatly, ranging from single to hundreds of data points.

- Data heterogeneity:** If data is available, then it could be very heterogeneous. Such is the case with the ChEMBL database [54]. The activity measurements are reported using different metrics, different units, different assay parameters, which requires thorough data pre-processing. The provenance of the data can play an important role in the non-uniformity of the labels. Indeed, experimentally measured values can differ greatly, since they are intrinsically sensitive to assay conditions [57]. Results may very well vary from one experiment to another, not only when conducted in the same lab (inter-day), but also if the same experiment was conducted in separate labs (intra-lab).
- Molecular encoding:** As discussed in greater details in Section 2.2.1 (Chapter 2), molecules, regardless of their size, are complex objects obeying chemical, biological, and physical rules. One great challenge is the encoding of such objects in a computer readable format. While small molecules might be somewhat easier to encode, for example SMILES—simplified molecular-input line-entry system[58]—is a popular encoding, it becomes challenging when dealing with macromolecules, or proteins. Not only do they contain ten of thousands of atoms, requiring more storage, but they are also dynamic by essence.
- Model interpretability:** Deep learning models are highly flexible and provide predictions for a given task. However, they are often considered as a black box, and the prediction that is given for an input might seem like magic. This phenomenon increases with the depth of the neural network: the more layers, the more complex the system, the more the mechanism in the inner layers might be opaque. Trusting the outcome of a black box-like model can have serious consequences in areas such as healthcare [59], such as the (mis)diagnosis of a disease that has a direct impact on human lives [60]. In order to overcome such issues, methods have been developed to better understand the prediction of a model. Chapter 4 is dedicated to this topic and introduces an approach based on the decomposition of the inner layers of a deep neural network to explain the toxicity classification.
- Closed/proprietary source code & data:** As mentioned in Section 1.1.2, machine learning has grown remarkably popular, and consequently, many data sets and models have been made purposely available, but free and open-source science is not the norm in drug design: from proprietary data which hinders reproducibility and benchmarking, to licensed and priced software, therefore not accessible to everyone, and finally closed-source models, and model implementa-

tion, which again greatly affect reproducibility and transparency. For the reusability of workflows and the reproducibility of results, widely applying the FAIR —Findability, Accessibility, Interoperability, and Reusability —principles [61] is a first step towards the promotion of open-source science. Pipelines following these principles are described in detail in Chapter 5.

### 1.3 Kinases as drug targets

Cancer remains one of the main causes of death in the world. As shown in Figure 1.4 taken from Roser and Ritchie [62], over 10 million people globally died in 2019 because of cancer, behind cardiovascular diseases that caused over 18 million losses. More optimistically, cancer treatments do exist. For example, erlotinib is administered for the treatment of lung and pancreatic cancer [63]. Moreover, between 2015 and 2020, 29% of FDA-approved (Food and Drug Administration) drugs were anticancer [64]. However, mild to severe side effects are experienced with most treatments and drug resistance is yet a substantial challenge [65, 66]. In order to prevent such undesired consequences of cancer treatment, developing drugs that are selective towards the target of interest is of utmost importance. Research has shown that kinases are protein targets that are frequently dysregulated in cancer [67]. Kinase inhibitors —small molecules that block catalytic effect —are therefore a therapeutic route for combating diseases such as cancer [68].

Kinases are a family of proteins and the human kinome, the superfamily of all kinases expressed in the human body, consists of approximately 540 kinases [69] (see Figure 1.5). These kinases contain one or more *kinase domains*, which are the catalytic domains responsible for transferring a phosphate from ATP —adenosine triphosphate, the source of inorganic phosphate in cells —to serine, threonine, or tyrosine residues of substrate proteins. This phosphorylation activity, in turn, allows downstream cell signaling that regulates processes such as cell division, cell migration, or cell death. The significant consequences of these downstream signaling events means that the kinase domain activity is often tightly regulated by other domains, binding partners, or upstream signaling proteins that must phosphorylate and activate the kinase domain. If kinases instead become dysregulated, such as becoming hyper activated to transmit phosphorylation signals inappropriately, such behavior could lead to diseases such as cancer. Figure 1.5a shows the 3D crystal structure of the erlotinib inhibitor in complex with the EGFR (Epidermal Growth Factor Receptor) kinase (PDB identifier 1M17 [70, 71]). Erlotinib acts as an EGFR inhibitor and is administered for the treatment of lung cancer. Figure 1.5c displays, in yellow, the amino acid binding site sequence of EGFR, and, in blue, the SMILES encoding of erlotinib.



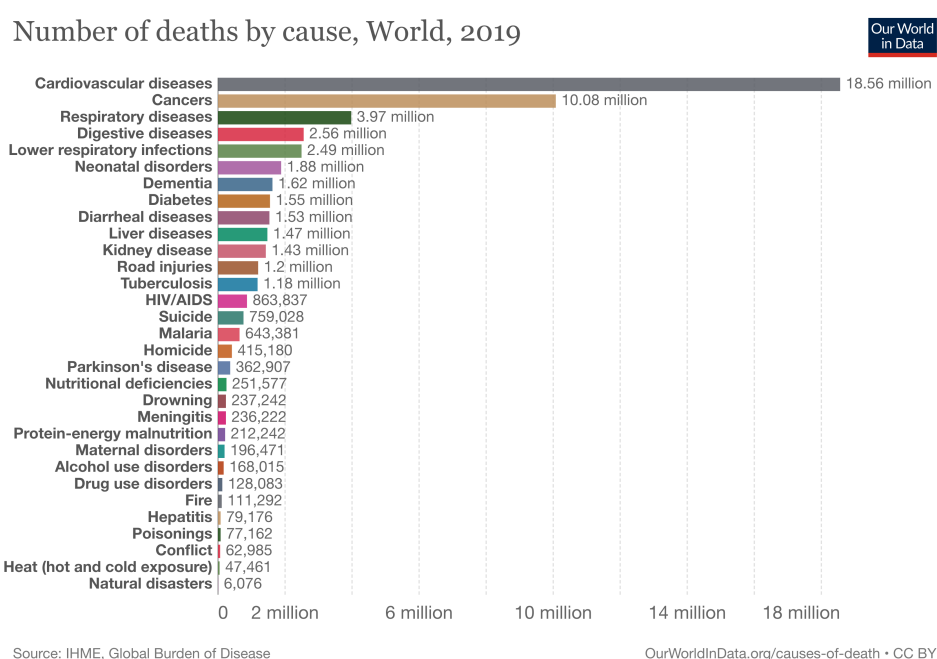
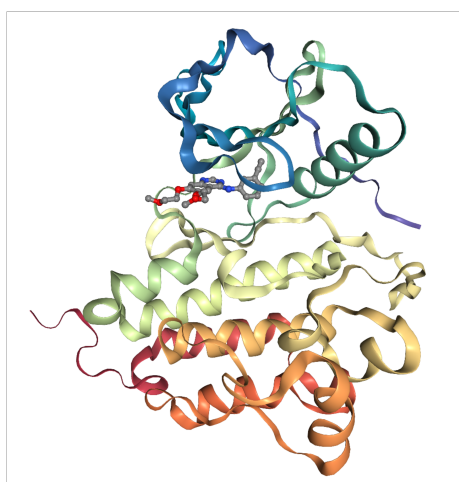
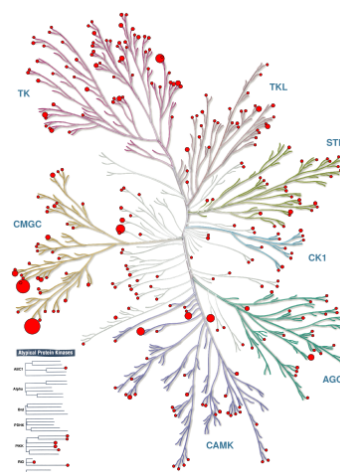


Figure 1.4: **Cancer killed 10.08 million people globally in 2019.** As shown in the figure taken from [62], cancer was the number two cause of death worldwide in 2019. Kinase drug design aims at developing kinase inhibitors as therapeutic route for combating cancer.



(a) **3D depiction of the EGFR kinase bound to erlotinib inhibitor.** The erlotinib drug, in gray, acts as an EGFR inhibitor and is commonly administered for the treatment of lung and pancreatic cancer. The structure corresponds to the 1M17 PDB identifier [70, 71].



(b) **Imbalanced kinase coverage depicted on the kinome tree.** The red circle is proportional to the number of PDB structure per kinase, showing great discrepancy between the data available among kinases. The figure is taken from [5].

Target	Binding site amino acid sequence
EGFR	KVLGSGAFGTVYKVAIKELEILDEAYVMASVDPHVCRLLGIQLITQLMPFGCLLDYVREYLEDRRRLVHRDLAARNVLTDFGLA
Drug	Molecular encoding: SMILES
erlotinib	<chem>C#Cc1cccc(Nc2ncnc3cc(OCCOC)c(OCCOC)cc23)c1</chem>

(c) The table shows the amino acid sequence of the binding site of EGFR (in yellow) and the SMILES of erlotinib (in blue).

**Figure 1.5: Protein kinases as drug targets.** When dysregulated, kinases cause severe diseases such as cancer. There are over 500 kinases in the human body, constituting the human kinome.

## Challenges for kinase inhibitor discovery

The inhibition of kinases by small molecules has proven to be a successful therapeutic route for combating cancer. However, designing drugs that are selective and that minimize the potential for side effects severe enough to cause discontinuation of therapy is highly non-trivial. Two of the main challenges to kinase inhibitor drug discovery are discussed below.

1. **Imbalanced exploration of the human kinome:** As mentioned previously, over 500 kinases are identified in humans, out of which approximately 40% have not been structurally resolved [72], meaning that the 3D atomic coordinate of the kinase has not been determined. In contrast, there are some kinases for which large amounts of data exist. For example, the Cyclin-dependent kinase 2, or CDK2, which plays a role in the proliferation of cancer cells [73], has over 400 available PDB structures, see Figure 1.5b. The imbalance of knowledge and data availability in the human kinome makes it difficult to use structure-based computational methods to, for example, predict the selectivity of inhibitors towards kinases, a task for which 3D structures of all kinases is required.
2. **Conservation of the ATP binding site:** The active site, or ATP binding site, of a kinase plays a very important role, since it is where ATP-competitive inhibitors (which constitute the vast majority of approved kinase inhibitor drugs) bind. In the case of kinases, this catalytic cleft is highly conserved across the kinome, meaning that there has been only very little change in the amino acids over evolutionary time. From a drug design point of view, this makes the discovery of selective kinase inhibitors [74] and, consequently, the prevention of side effects very challenging. Indeed, if the binding site of kinases is where the ligand binds and these sites are very similar across different kinases, then it would be likely that a ligand will bind not only to its intended target (on-target), but also to other similar targets (off-targets), probably inducing adverse effects.

In order to find selective kinase inhibitors to be able to reduce, or even better, prevent, side effects in cancer treatments, having good knowledge of kinases, their mechanism, their role, their similarities, would be highly beneficial.

## 1.4 The goal of this thesis

Diseases such as cancer are a leading cause of death globally, and the treatments that currently exist are not without consequences. Side effects occur

and can range from mild to severe. In this context, computer-aided drug discovery has become an integral part of Research and Development in pharmaceutical companies, and participates in finding new, efficient, selective, non-toxic drugs with minimal side effects.

While computational methods, and more specifically machine and deep learning applications, have matured in the field of drug design, major challenges still remain: training deep learning models on scarce data sets describing physicochemical properties; quantifying the uncertainty of a property prediction for a novel compound; understanding the decision-making process that leads deep neural networks from an input to a prediction; and improving availability and usability of free, open-source, and modular workflows.

In the upcoming chapters of this thesis, we present research on all these different challenges and discuss solutions to the underlying issues.

More specifically, Chapter 2 reviews the state-of-the-art in virtual screening, a method commonly used in computer-aided drug design to rapidly select promising molecules from a large database that are likely to bind to a given target. We explore how novel encodings for small molecules, proteins, or a complex, are improving the accuracy of the models. We also explore which deep learning models are reaching outstanding results. Firstly, we distinguish the approaches used in virtual screening: ligand-/pair-/ and complex-based and explain their particularities. Secondly, we cover encodings for three molecular entities: 1. For ligands, popular encodings such as fingerprints, SMILES, and graphs are described. 2. For proteins, sequence and structural encodings are examined, and 3. Encodings such as 3D grid, graph, interaction fingerprints are included for the protein-ligand complex. Thirdly, neural networks are described in detail, covering specificities such as convolutional and recurrent layers, as well as graph neural networks. Furthermore, common benchmark and bioactivity data sets in virtual screening are discussed. Finally, the state-of-the-art results from the last decade are summarized in a table showing the remarkable progress of virtual screening over the years.

Chapter 3 deals with data scarcity in the context of deep learning models applied to small molecules. As discussed in Section 1.2.3, data scarcity remains one of the major challenges when applying deep learning models to drug design-related tasks. Indeed, the size of relevant data sets are multiple orders of magnitude smaller than the ones used in other areas where deep learning is applied, such as speech recognition [75] and image classification [51]. In light of data augmentation techniques that have successfully been applied to image recognition, we develop data augmentation techniques to improve molecular property prediction, a task considerably useful in lead optimization. We also deliver guidelines on how to best use augmentation in QSAR modeling. We train and test our models on properties important in drug design: 1. Lipophilicity, which plays an important role in absorption [76] (see ADMET), 2. Water solubility, which participates in the distribution

of a compound in an organism [52], 3. Hydration free energy [77], and 4. The activity towards the EGFR kinase, which, when overly expressed, is found in cancer prognosis [78]. Further, we show that using augmentation at inference time allows to formalize the confidence of a model in its prediction. All developed software is made available and can be used to make predictions on all the aforementioned properties for novel compounds.

In Chapter 4, we aim to reveal the decision-making process of a neural network in the context of cytotoxicity prediction. The goal being to identify potential toxic substructures, or toxicophores, in a compound. As previously mentioned in Section 1.2.3, neural networks are often considered as a black box and better understanding the prediction of a model from the input features is highly beneficial, especially in the context of toxicity. Cytotoxicity, leading to cell death, plays an important role in drug design. The assessment of the toxicity of a compound at an early stage of the drug design process would allow to not only reduce *in vivo* animal testing, and save costs, but could also help to optimize compounds. We expand on the Deep Taylor Decomposition [79], a technique introduced in the context of image classification, to improve our understanding of toxic substructures. We develop a visualization of the identified toxicophores providing a simple image to be analyzed by medicinal chemists for validation.

Chapter 5 focuses on the importance of pipelines and on the role of kinases in drug design. As mentioned previously, the drug design process can be time-consuming and requires several iterations. Being able to automate each step in a modular and robust way would speed up the design of new compounds, saving time and use of costly experiments. As discussed in Section 1.3, kinases are known drug targets, and developing strategies to compare kinases could help to understand off-targets. We therefore introduce two main pipelines as part of the TeachOpenCADD project [4, 80]. The first one is an automated structure-based virtual screening pipeline that involves the binding site detection, docking calculations and protein ligand interaction visualization. The second one focuses on the role of kinases as drug targets, and we create a pipeline that allows the comparison of kinases, providing insight into off-targets. We implement four measures of comparison which take into account knowledge on sequence, structure, protein-ligand interactions, and bioactivity data, spanning a wide range of kinase information.

Finally, in Chapter 6, the conclusion, we summarize our findings, and discuss potential extensions.



## Chapter 2

# Deep Learning in Virtual Screening: Recent Applications and Developments

The contents of this chapter were published as Kimber, T.B.\* , Chen, Y.\* , & Volkamer, A. (2021). Deep Learning in Virtual Screening: Recent Applications and Developments. *International Journal of Molecular Sciences*, 22(9), 4435 [1], under the Creative Commons Attribution (CC BY) license, <https://creativecommons.org/licenses/by/4.0/>. The content from this publication is presented here with the permission of MDPI publishing.

The contributions of the authors are as follows: All authors contributed to examining the literature and describing the methods. TBK led the deep learning section, molecular encodings, as well as the pair-based state-of-the-art methods. The text and figures were written and produced by all authors.

### Chapter summary

Drug discovery is a cost and time-intensive process that is often assisted by computational methods, such as virtual screening, to speed up and guide the design of new compounds. For many years, machine learning methods have been successfully applied in the context of computer-aided drug discovery. Recently, thanks to the rise of novel technologies as well as the increasing amount of available chemical and bioactivity data, deep learning has gained a tremendous impact in rational active compound discovery. Herein, recent applications and developments of machine learning, with a focus on deep learning, in virtual screening for active compound design are reviewed. This

---

\*These authors have shared first authorship.

includes introducing different compound and protein encodings, deep learning techniques as well as frequently used bioactivity and benchmark data sets for model training and testing. Finally, the present state-of-the-art, including the current challenges and emerging problems, are examined and discussed.

## 2.1 Introduction

### 2.1.1 Virtual screening

Drug discovery remains a key challenge in the field of bio-medicine. Traditionally, the discovery of drugs begins with the identification of targets for a disease of interest. It is followed by high-throughput screening (HTS) experiments to determine hits within the synthesized compound library, i.e. compounds showing promising bioactivity. Then, the hit compounds are optimized to lead compounds to increase potency and other desired properties, such as solubility, or vanishing toxic and off-target effects. After these pre-clinical studies, potential drug candidates have to pass a series of clinical trials to become approved drugs. On average, more than 2 billion US dollars and about 10-15 years are spent for developing a single drug [81]. While HTS experiments are very powerful, they remain time and cost-intensive, since they require several thousands of synthesized compounds, a large number of protein supplies, and mature methods for bioactivity testing in the laboratory [82].

To rationalize and speed up drug development, computational methods have been widely incorporated in the design process in the past three decades. One prominent method is virtual screening (VS), which is used to prioritize compounds from (ultra) large compound libraries which have a high potential to bind to a target of interest [83]. VS methods can efficiently scan millions of (commercially) available compounds, such as ZINC [84] or MolPORT [85], at low cost and prioritize those to be tested, synthesized in-house, or purchased from external suppliers. Besides, VS can be carried out in virtual compound libraries, which expands the chemical space, such as Enamine REAL [86] with over 17 billion make-on-demand molecules and a database containing close to two billion drug-like compounds. Although VS methods are not always able to find the most active compound, they can narrow the search space down to few hundreds of compounds with desired properties to be further investigated [87].

Nowadays, VS has become an integral part of drug discovery. It is usually implemented in the form of a hierarchical workflow, combining different methods (sequentially or in parallel) as filters to prioritize potentially active compounds [87, 88]. VS methods are often divided into two major categories: (1) structure-based methods, which focus on the complementarity of the target binding pocket and the ligand; as well as (2) ligand-based



methods, which rely on the similarity of novel compounds to known active molecules.

Structure-based methods (1) require 3D structural information of both ligand and protein as a complex or at least of the protein with some knowledge about the binding site. The most commonly used technique is molecular docking, which predicts one or several binding pose(s) of a query ligand in the receptor structure and estimates their binding affinity [89]. While protein-ligand docking shows great ability in enriching likely active compounds over inactive ones, there are still complications in placing or scoring the individual poses, some of which can be unmasked by visual inspection [90–93]. During the molecular docking process, thousands of possible ligand poses are generated based on the target structure and ranked by a scoring function (SF) [94]. There are three classical types of scoring functions: physics-, empirical-, and knowledge-based [95, 96]. Physics-based methods rely on molecular mechanics force fields. In short, non-bonded interaction terms such as Van der Waals interactions, electrostatics, and hydrogen bonds are summed. Similarly, empirical SFs sum weighted energy terms. Items describing for example rotatable bonds or solvent-accessible-surface area are also added and all terms are parameterized against experimental binding affinities. In contrast, knowledge-based methods rely on statistical analyses of observed atom pair potentials from protein-ligand complexes. More recently, new groups of scoring functions were introduced, namely machine/deep learning-based SFs. One group of models is based on classical SFs which try to learn the relationship between the interaction terms to predict binding affinity (see the review by Shen et al. [96]). Others models encode the complex via protein-ligand interaction fingerprints, grid- or graph-based methods [97]. Such models will be referred to as complex-based methods throughout this review and discussed in greater details, see Figure 2.1. Note that pharmacophore-based VS has also incorporated machine learning, and is suitable to screen very large databases, see for example Pharmit [98]. However, these methods are not the focus of this review and recent developments in the pharmacophore field are described by Schaller et al. [99].

Ligand-based methods (2), including QSAR (quantitative structure - activity relationship) modeling, molecular similarity search and ligand-based pharmacophores, are relatively mature technologies [47]. Unlike structure-based methods, ligand-based methods only require ligand information. Note that they are not the focus of this review and the reader is kindly referred to the respective literature, e.g. [47, 100]. Nevertheless, the latter category can also be enriched by *simple* protein—mostly sequence-based—information and is often referred to as proteochemometric (PCM) modeling, which will be further addressed in this review. PCM combines both ligand and target information within a single model in order to predict an output variable of interest, such as the activity of a molecule in a particular biological assay [101, 102]. Thus, PCM methods do not only rely on ligand similarities,

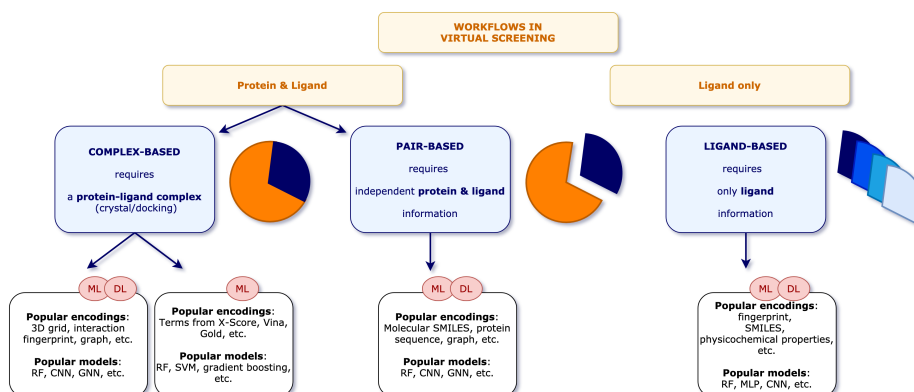


Figure 2.1: **Workflows in virtual screening.** The first split separates the schemes that contain (1) protein and ligand information and (2) ligand information only, which are typically used in models for QSAR predictions. For details on solely ligand-based methods, see e.g. MoleculeNet [104]. For (1), a second split makes the differences between complex-based and pair-based models. Complex-based models describe the protein and ligand in a complex, whereas pair-based models (also PCM in the broader sense) treat the protein and ligand as two independent entities. The latter typically use protein sequence and molecular SMILES information as input, while the complex-based models use, for example, a 3D grid of the protein-ligand binding site or interaction fingerprints.

but incorporate information from the target they bind to, and have been found to outperform truly ligand-based methods [103]. Note that in some PCM applications an additional cross-term is introduced that can describe the interaction between the two objects [101]. To distinguish the herein described methods, which handle the two objects individually, we refer to them as pair-based methods, see Figure 2.1.

### 2.1.2 Machine learning and deep learning

Machine learning (ML) aims at learning a task from data by optimizing a performance measure. There exist three main approaches [14]: (1) Unsupervised learning in which the goal is to find patterns of the underlying structure and gain interpretability of the data. (2) Reinforcement learning in which an agent evolves in an environment and uses the data learned from experience. (3) Supervised learning in which an algorithm is trained on inputs to predict some labeled output. The latter technique will be the focus of this review.

Traditional supervised ML methods follow the idea that given some data, a predictive model is constructed by optimizing the difference between a given labeled output and the output predicted by the model. Some of these

methods date back to the last century. For example, neural networks were first developed in the 60s by Rosenblatt [15]. Later, in the 80s, Breiman et al. [16] published the book *Classification and regression trees*. In the 90s, Cortes and Vapnik [17] introduced support vector machines (SVMs) and more recently, in the early 2000s, Breiman [18] proposed random forests (RFs).

Nevertheless, over the last few years, ML methods have gained a lot of popularity. This may be explained by three major aspects: (1) Data availability: thanks to automation and digitalization, as well as memory capacities, the amount of stored data has never been greater. (2) Computing power, such as graphics processing units (GPUs) and parallelization, has significantly allowed expensive model training. Cloud computing, for instance, Google Colaboratory [105], allows any user to train resource intensive machine learning models using powerful tensor processing units (TPUs). (3) The theoretical research on the learning algorithms has enabled the development of sophisticated models and training schemes.

Deep learning (DL) [13] is a subset of ML in which the input features are combined using hidden layers that constitute a network. Each hidden layer is made up of a linear and a non-linear part, the non-linear part called the activation function. The information then flows through the network. The resulting predictive model is highly flexible and is able to extract complex patterns thanks to the non-linearities. Since describing (and understanding) the interactions between molecular structures in a biological context is highly complex, it is not surprising that applying deep learning to such objects could yield excellent performance.

### 2.1.3 Data availability and big data

As mentioned above, automation and storage have had a major impact on the amount of data existing nowadays. Recently, Google has published an image data set of over 9 million data points called "Open Images Dataset V4" [51] as well as "YouTube-8M", a video data set of 8 million URLs. These large open-source data sets have enabled researchers to build highly efficient models in fields such as image classification. Benchmark data sets are also widely used in the machine learning community to train, test, and compare new models and architectures. One of the popular benchmark data sets in image classification is the MNIST database of handwritten digits [106] which has a training set of 60,000 examples and a test set of 10,000 examples. Kaggle [33] is a community that hosts competitions in very diverse fields, including e.g. drug activity prediction, where the data are made public. These competitions allow to prospectively evaluate all kinds of different schemes and rank them using hold out data sets.

In the biomedical field, the size of the data sets is starting to reach similar scales. The amount of publicly available bioactivity data keeps increasing every year. To date, the well-known ChEMBL database [107] has 17, 276, 334

registered activity entries [108] and has become a valuable resource in many types of life science research. Furthermore, a considerable amount of structural data has also been published over the last decades. The freely available Protein Data Bank (PDB) [41, 42] logged 14,047 new entries in 2020. In March 2021, the total number of available entries has surpassed 175,000 [109] and will most probably keep increasing. The structural data come from experimental methods, such as X-ray crystallography, nuclear magnetic resonance spectroscopy and electron microscopy, technologies that have improved in precision and throughput over the last years [110, 111]. Publicly available screening libraries also have big data potential. For example, the ZINC database [84] contains over 230 million of commercially available compounds. More details on specific data sets will be given in the Methods & Data section.

#### 2.1.4 Deep learning in virtual screening

Given the increasing amount of available structural and bioactivity data as well as the recent progress in machine—especially deep—learning, it is no wonder that virtual screening strategies could benefit from this synergy.

While ML methods have been applied in the field for over two decades already [112–114], DL has begun to rise in the drug discovery area, especially in VS [115]. Given the new developments, various reviews about ML and DL in VS have recently been published [96, 116–121]. For example, Shen et al. [96] and Li et al. [119] review differences between more traditional ML—and DL—based scoring functions (SFs). Rifaioğlu et al. [121] present an overview of recent applications of DL and ML on *in silico* drug discovery. In contrast, this review focuses on the one hand on advances regarding DL-based VS in recent years, and on the other hand covers two main groups of models, both including information from the protein and the ligand: (1) Complex-based models, which are trained on information/encodings from complexes or docking poses of protein and ligand for predicting the binding affinity of a given molecule; and (2) pair-based models or PCM, which are primary ligand-based but include simple information from the protein they bind to.

## 2.2 Methods & Data

In this section, the main encodings of ligand, protein and complex, the different deep learning models as well as the most used (benchmark) data sets are introduced.

### 2.2.1 Encodings in virtual screening

The interactions between protein and ligands are complex, and encoding the most informative bits in a computer-readable format is one of the main

challenges in both cheminformatics and bioinformatics. In the following sections, the encodings for ligands in virtual screening are described, followed by protein and complex encodings. The details are reported for those used in the studies discussed in the Recent developments section. A more exhaustive list of ligand encodings is carefully outlined in the review by Lo et al. [122]. For protein descriptors, the work by Xu et al. [123] describes common sequence- as well as structure-based descriptors, embedding representations and possible mutations.

### Ligand encodings

The starting point of several ligand encodings is the molecular graph, where nodes and edges represent the molecular atoms and bonds, respectively (see Figure 2.2).

**Graph** The molecular graph can be encoded using two matrices: the first one, called the feature matrix  $X$ , gives a per atom description, where the type of information stored in each node is decided *a priori*. Common per atom examples are atomic type and degree [126]. The dimension of  $X$  is  $N \times D$ , where  $N$  is the number of nodes, i.e. atoms, in the graph and  $D$  the number of pre-defined features. The second matrix, called connectivity matrix, describes the structure of the molecule. Its purpose is to illustrate how the nodes are connected in the graph, i.e. via bonds. Two frequent formats store this information: (1) the adjacency matrix  $A$  of dimension  $N \times N$ , where  $A_{ij} = 1$  if node  $i$  is connected to node  $j$  and 0 otherwise. (2) The coordinate (COO) format of dimension  $2 \times E$ , where  $E$  represents the number of edges in the graph. Thus, the first and second rows represent the index of the source and target nodes, respectively. Using both the feature matrix and the connectivity matrix, the molecular graph encoding can further be used to apply machine learning algorithms.

**SMILES** An efficient way of storing information from the molecular graph using string characters is the simplified molecular input line entry system (SMILES) developed by Weininger [58]. The main idea behind SMILES is the linearization of the molecular graph by enumerating the nodes and edges following a certain path. Due to the randomness in the choice of the starting atom and the path followed along the 2D graph, there exist several valid SMILES for one molecule [127]. However, it may be desirable to have one unique SMILES for a given compound, called the canonical SMILES, and most software have their own canonization algorithm. In order to apply mathematical operations in the context of machine learning, SMILES still need to be transformed into numerical values, where both label and one-hot encoding are often used [128, 129]. Please find more information on these encodings below.

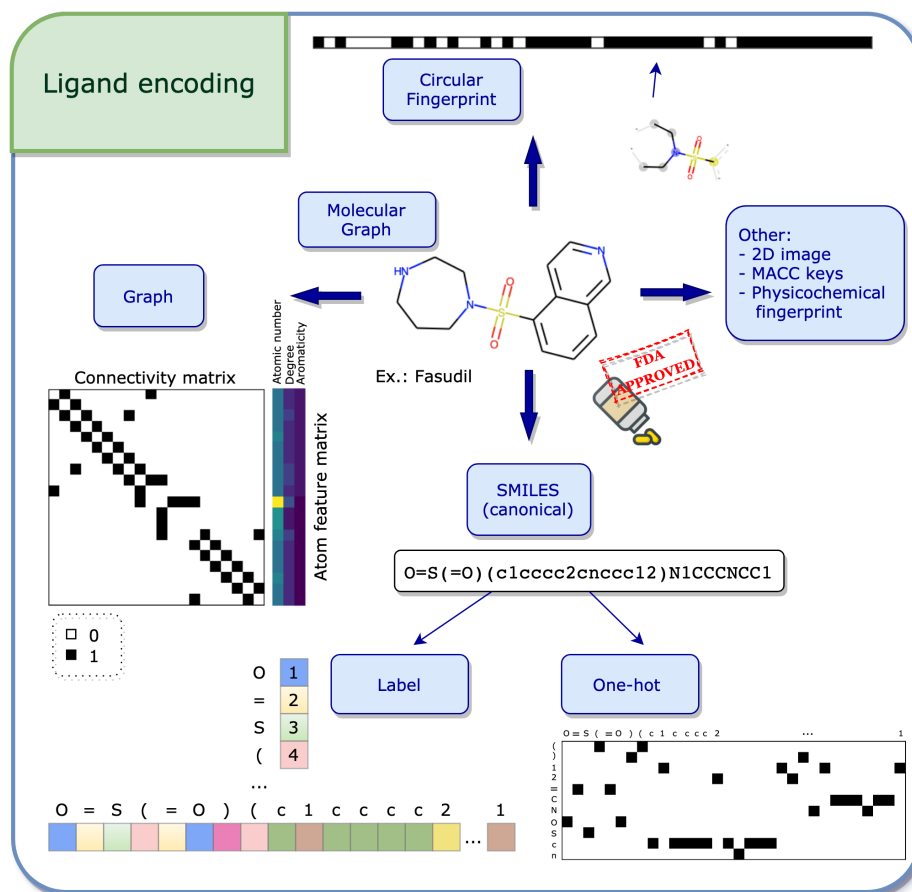


Figure 2.2: **Ligand encoding.** Having a computer-readable format is one of the starting points for machine—and deep—learning. The example molecule is the FDA-approved drug fasudil [124] taken from the PKIDB database [125]. Recent studies focused on virtual screening (detailed in the Recent developments section, see Table 2.4) commonly use SMILES, circular fingerprints or graphs to encode the ligand.

**Label and one-hot encoding** In this section, the concepts of label and one-hot encoding are explained in the context of SMILES, but the idea can be translated to broader applications, such as natural language processing (NLP) [13]. Mathematical operations such as matrix computations cannot be applied directly to string characters and therefore these strings have to be transformed into numerical entities. The first step towards this transformation is the specification of a dictionary of considered characters, which can be done in two ways: either by inserting all the characters existing in the data set, or by exhaustively enumerating a list of all necessary characters. Once the dictionary is defined, label or one-hot encoding can be used.

In label encoding, the characters in the dictionary are enumerated. This enumeration using integer numbers is also sometimes referred to as integer encoding. Using the integer labels, a SMILES can be transformed into an integer vector by associating each character with its integer label (see Figure B.1(a)). The advantage of such a representation is its compact form, leading to a simple integer vector. A disadvantage however is the natural hierarchy in the numbering, giving higher values to some characters in the dictionary.

The one-hot encoding resolves this issue by assigning a binary vector to each character in the dictionary. A SMILES can then be constructed by concatenating the binary vectors as they appear in the SMILES (see Figure B.1(b)). The main disadvantage of using the one-hot transformation is that the resulting matrix may be large and sparse. In both label and one-hot encoding, having all elements in a data set with the same dimension is often required in a machine learning setting. A way to account for the different dimensions is to use padding, which adds zeros to the vector (in the label encoding case) or to the matrix (in the one-hot encoding case) up to a maximum length, usually determined by the longest element in the data set.

**Circular fingerprint** Circular fingerprints are, once folded, fixed-length binary vectors that determine the presence (encoded by 1) of a substructure or the absence of it (encoded by 0). The recursive algorithm behind extended-connectivity fingerprints (ECFP) [130] starts with an atom initializer for each node and updates the atom identifiers using a hash function by accumulating information from neighboring nodes. The substructures which are identified using the local atom environments correspond to the bits in the fingerprints. A free version of the algorithm is available in the open-source cheminformatics software RDKit [131] (under the name of Morgan fingerprints), which will not produce the same results as the original implementation in Pipeline Pilot [132] due to the difference in the hash functions, but will yield similar results.

**Other encodings** Ligands are evidently not restricted to these encodings [122]. For example, different types of fingerprints may be used, such as a physicochemical-based vector, describing the global properties of the molecule, as in the study by Kundu et al. [133]. Also, the 166-bit long MACCS keys [134] are a common way to encode molecular compounds as a fingerprint. Recently, learned fingerprints have also shown to be effective in QSAR predictions [129, 135]. Another way of employing the molecular structure as input to machine learning is the 2D image itself. Rifaioğlu et al. [136] use the 200-by-200 pixel 2D image generated directly from the SMILES using the canonical orientation/depiction as implemented in RDKit [131].

### Protein encodings

Proteins are macromolecules that are involved in many biochemical reactions. They are composed of distinct amino acid sequences, which result in folding in specific 3D protein structures [137].

**Protein identifier** A simple way to discriminate models from ligand information only is to include the identifier (ID) of the protein. Such a descriptor adds no information whatsoever about the physicochemical properties of the protein, the amino acid composition, nor the 3D conformation. It is merely a way for a machine learning model to be able to differentiate several proteins. For example, the one-hot encoding of the protein ID can be used, as in the study by Sorgenfrei et al. [138].

**Protein sequence** The (full) sequence of a protein, often referred to as the primary structure, is the enumeration of the amino acids as they appear from the beginning (N-terminus) to end (C-terminus) of the protein, in which each of the 20 standard amino acids can be encoded as a single letter. The length of a protein can vary greatly, some of them containing thousands of amino acid residues. Although the sequence is a compact way of storing information about the primary structure, it does not give any information about the 3D structure of the protein. In opposition to the full sequence, it is possible to only consider the sequence from the binding site, reducing greatly the number of residues.

**Z-scales** The z-scale descriptors published in the late 80s by Hellberg et al. [139] are constructed by considering, for each of the 20 amino acids, 29 physicochemical properties such as the molecular weight, the logP and the logD (see [139, Table 1]). A principal component analysis (PCA) on the  $20 \times 29$  matrix is performed and the three principal components  $z_1$ ,  $z_2$  and  $z_3$  for each of the amino acids are retained. The authors suggest interpreting  $z_1$ ,  $z_2$  and  $z_3$  as hydrophilicity, bulk and electronic properties, respectively.



**Domains and motifs** A domain is a structural unit within a protein that is conserved and the overall role of a protein is often governed by its domain function. PROSITE [140] and Pfam [141] are two popular databases that store a large variety of protein domains. Therefore, a possible way of encoding proteins is through a binary vector which indicates the presence or absence of a particular domain.

**Structural property sequence** In the study by Karimi et al. [142], the proteins are encoded using structural property sequences, which describe the structural elements of the protein but do not require the 3D structures. The secondary structure is predicted from the sequence using SSpro, developed by Magnan and Baldi [143]. Neighboring residues are thereby grouped together to form *secondary structure elements*. Then, four letters are assigned to each of these elements. The first letter represents the secondary structure: alpha helix (A), beta sheet (B) or coil (C). The second letter determines the solvent exposure: N as "not exposed" or E as "exposed". The third letter describes the physicochemical properties, i.e. non-polar (G), polar (T), acidic (D) or basic (K). The last letter represents the length: small (S), medium (M) or large (L).

### Complex encodings

Describing the protein-ligand complex involves descriptions that capture the interactions between the two binding partners. Herein, we group them into interaction fingerprints, 3D grids, graphs and other.

**Interaction fingerprint** Interaction fingerprints (IFPs) describe—as the name implies—the interactions between a protein and a ligand based on a defined set of rules [100, 144]. Typically, the IFP is represented as a bit string, which encodes the presence (1) or absence (0) of interactions between the ligand and the surrounding protein residues. In most implementations, each binding site residue is described by the same number of features, which usually include interaction types such as hydrophobic, hydrogen bond donor and acceptor.

*IFPs encoding interaction types:* The structural interaction fingerprint (SIFt) [145] describes the interactions between the ligand and  $n$  binding site residues as an  $n \times 7$  long bit string. Here, the seven interaction types include whether the residue (i), and more precisely their main (ii) or side (iii) chain atoms, are in contact with the ligand; whether a polar (iv) or apolar (v) interaction is involved; and whether the residue provides hydrogen bond acceptors (vi) or donors (vii). Similarly, the protein-ligand interaction fingerprint (PyPLIF) [146] and the IChem's IFP [147] encode each residue also by seven though slightly different types, while PADIF [148] uses the Gold [149] scoring function contributions as interaction types.

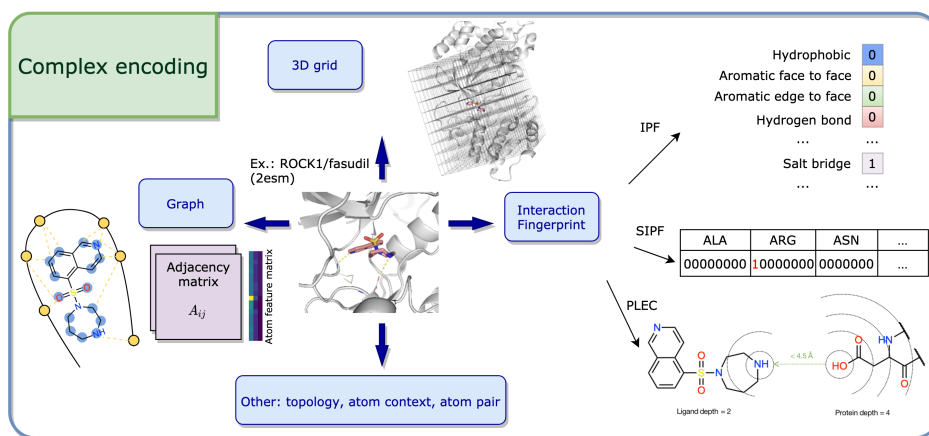


Figure 2.3: **Complex encoding.** Visual representation of encodings for protein-ligand complexes used in structure-based virtual screening, exemplified with the drug fasudil co-crystallized with the ROCK1 kinase (PDB ID: 2esm). 3D grids, graphs and interaction fingerprints are among popular encodings for complexes, as discussed in the Recent developments section (see Table 2.3).

The aforementioned IFPs vary in size and are sensitive to the order of the residues, which limits their application for ML purposes. Thus, SILRID [150], a binding site independent and fixed-length IFP was introduced. SILRID generates a 168 long integer vector per binding site, obtained by summing the bits corresponding to a specific amino acid or cofactor (20 + 1), while each amino acid is described by eight interaction types.

*IFPs including distance bits:* To describe the interactions more explicitly, distances with respect to interaction pairs or triples were introduced. APIF [151], an atom-pair based IFP, encodes three interaction types for the protein and ligand atoms: hydrophobic contact, hydrogen bond donor and acceptor. Combinations of these three types lead to six pairings, including for example a protein acceptor-hydrophobic pair complemented with a ligand donor-hydrophobic atom-pair. Moreover, for each pairwise interaction in the active site, the respective receptor and ligand atom distances are measured and binned into seven ranges. In this way, the total APIF is composed of 6 types  $\times$  7 protein distances  $\times$  7 ligand distances = 294 bits. Pharm-IF [152], while using slightly different interaction type definitions, calculates distances between the pharmacophore features of their ligand atoms. Finally, triplets between interaction pseudoatoms are introduced in TIFP [153]. The fingerprint registers the count of unique interaction pseudoatom triplets encoded by seven properties (e.g. hydrophobic, aromatic or hydrogen-bond) and the related distances between them, discretized into six intervals. Redundant

and geometrically invalid triplets are removed, and the fingerprint is pruned to 210 integers representing the most frequently occurring triplets in the appointed data set.

*IFPs including circular fingerprint idea:* To become more independent of pre-defined interaction types, circular fingerprint inspired IFPs were introduced by encoding all possible interaction types (e.g.  $\pi - \pi$ ,  $\text{CH} - \pi$ ) implicitly via the atom environment. The structural protein-ligand interaction fingerprint (SPLIF) [154] is constructed using the extended connectivity fingerprint (ECFP, see the Ligand encodings section for more information). For each contacting protein-ligand atom pair (i.e. distance less than 4.5 Å), the respective protein and ligand atoms are each expanded to circular fragments using ECFP2 and hashed together into the fingerprint. Similarly, ECFP is integrated in the protein-ligand extended connectivity (PLEC) fingerprint [155], where  $n$  different bond diameters (called "depth") for atoms from protein and ligand are used.

**3D grid** Another type of encoding are 3D grids, in which the protein is embedded into a three-dimensional Cartesian grid centered on the binding site. Similar to pixel representation in images, each grid point holds one (or several) values that describe the physicochemical properties of the complex at this specific position in 3D space. Such grids can, for example, be unfolded to a 1D floating point array [156] or transformed into a 4D tensor [157] as input for a DL model. Depending on the implementation, the cubic grids vary in size between 16 Å and 32 Å, as well as grid spacing (resolution) usually being either 0.5 Å or 1 Å [156–159]. Per grid point attributes can be (1) simple annotations of atom types or IFPs, such as in AtomNet [156] and DeepAtom [160], (2) physicochemical or pharmacophoric features, e.g. Pafnucy [157] and BindScope [161], or (3) energies based using one or several probe atoms as in AutoGrid/smina [158, 162].

**Graph** Although the description of a small molecule as a graph seems natural, the idea can be adapted to a molecular complex. As in the ligand case (see the Ligand encodings section), two main components have to be considered in the graph description of such protein-ligand structures: the nodes, with an associated feature vector, and the relationship between them, usually encoded in matrix form. When considering a complex, the atoms from both the protein and the ligand can simply be viewed as the nodes of the graph and the atomic properties can vary depending on the task at hand. Some might consider, among other characteristics, the one-hot encoded atom type/degree and a binary value to describe aromaticity, as in [163]. As simple as the node description is for complexes, the intricacy arises when describing the interactions between the atoms, which should account for covalent and non-covalent bonds. The focus here will be on two different ways of

describing such structures. The first one, developed by Lim et al. [163], considers two adjacency matrices  $A^1$  and  $A^2$ .  $A^1$  is constructed in such a way that it only takes into account covalent bonds, more precisely  $A_{ij}^1 = 1$  if  $i, j$  are covalently connected, and 0 otherwise.  $A^2$ , on the other hand, not only captures bonded intramolecular and non-bonded intermolecular interactions, but also their strength through distances. Mathematically, this can be translated as follows: if atom  $i$  belongs to the ligand, atom  $j$  to the protein, and they live in a neighborhood of 5 Å, then

$$A_{ij}^2 = e^{-\frac{(d_{ij} - \mu)^2}{\sigma}},$$

where  $d_{ij}$  is the distance between atoms  $i$  and  $j$ , and  $\mu$  and  $\sigma$  are learned parameters. The smaller the distance between the atoms to  $\mu$  is, the stronger the bond is. If atoms  $i$  and  $j$  both belong to either the ligand or the protein, then  $A_{ij}^2 = A_{ij}^1$ .

The other graph form of protein-ligand developed by Feinberg et al. [164] consists of an enlarged adjacency matrix  $A \in \mathbb{R}^{N \times N \times N_{et}}$ , where  $N$  is the number of atoms and  $N_{et}$  the number of edge types.  $A_{ijk} = 1$  if atom  $j$  is in the neighborhood of atom  $i$  and if  $k$  is the bond type between them. If not, that same entry is 0. This scheme numerically encodes the spatial graph as well as the bonds through edge type.

**Other encodings** Moreover, there are also other encoding methods to describe a complex, which will only be shortly introduced here. Topology-based methods, as reported by Cang and Wei [165], describe biomolecular data in a simplified manner. The topology thereby deals with the connectivity of individual parts and characterizes independent entities, rings and higher dimensional faces. In this way, element-specific topological fingerprints can retain the 3D biological information and the complex can be represented by an image-like topological representation (resembling barcodes).

Also, simply the protein-ligand atom pairs together with their distances can be used as input. In the work by Zhu et al. [166], all atom pair energy contributions are summed, where the contributions themselves are learned through a neural network considering the properties of the two atoms and their distances. Similarly, Pereira et al. [167] introduced the atom context method to represent the environment of the interacting atoms, i.e. atom and amino acid embeddings.

### 2.2.2 Deep learning models in virtual screening

As mentioned in the introduction, machine learning can be split into supervised, unsupervised and reinforcement learning. In this section, we focus on supervised learning which is a framework that is used when the data is constituted of some input and an associated label and the aim is to predict the

outcome corresponding to a given input. Subsequently, typical evaluation strategies of machine learning models will shortly be introduced.

### Supervised deep learning models

In the supervised framework, two subclasses are usually considered: the first one, called classification, deals with discrete outputs. In the binary case, this reduces to outputs that can take either 0 or 1 values. In the context of virtual screening, a simple example would be the activity determination of a compound against a protein: active (1) or inactive (0). The second subclass is regression, where the target value takes a real value instead. An analogous example in VS would be to predict the half maximal inhibitory concentration  $IC_{50}^*$  of a compound.

Common machine learning algorithms include tree-based methods such as random forests (RFs), tree boosting, and support vector machines (SVMs). However, over the last decades, deep learning has gained a lot of momentum and the rest of this section will be dedicated to the idea behind the deep learning models described in the Recent developments section (see Figure 2.4). For more rigorous definitions and mathematical notations, the reader is kindly referred to the book by Goodfellow et al. [13].

**Neural networks** Neural networks (NNs) [13], also sometimes called artificial neural networks (ANNs), are models that take as input a set of features on which mathematical computations are performed that depend on a set of parameters. The sequential computations between the input and the output are called hidden layers and the final one, the last layer, should account for the targeted prediction: classification or regression. The information flows through the network and is monitored by non-linearities called activation functions that determine if or how much of the information can be passed on to the next layer. The parameters in the network are optimized using back-propagation [13, Chapter 6].

A simple example of a neural network connects the input to the output with one single hidden layer and is sometimes called a "vanilla network" or a "single layer perceptron" [168], in opposition to a multilayer perceptron (MLP) that has more than one hidden layer. In a single layer perceptron, the hidden layer is composed of a set of nodes where each input element is connected to every hidden node and every node in the hidden layer is connected to the output. When all nodes from one layer are connected to the next, the layer is called fully-connected, or dense. If the network contains only such layers, then it is usually referred to as a fully-connected neural network, a dense neural network, or a multilayer perceptron. Note that throughout this review, the term MLP is used, while in the original

---

\*The half maximal inhibitory concentration, noted  $IC_{50}$ , describes the amount of a substance that is needed to inhibit a target protein/assay by 50%.

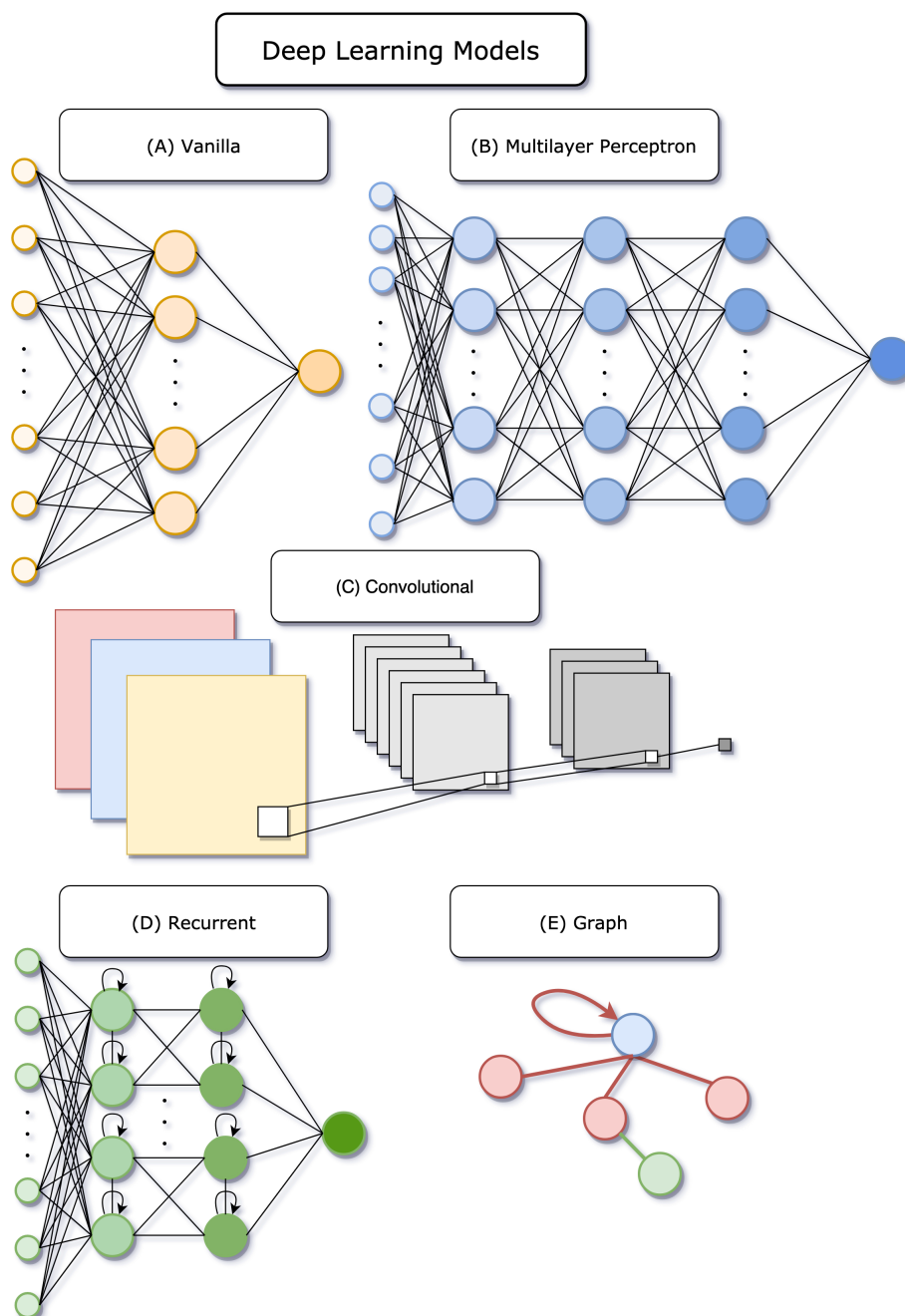


Figure 2.4: **Deep learning models.** Schematic illustration of the neural networks described in the Recent developments section. (A) Vanilla neural network. (B) Multilayer perceptron with three hidden layers (MLP). (C) Convolutional neural network (CNN). (D) Recurrent neural network (RNN). (E) Graph neural network (GNN). CNNs and GNNs particularly have become very popular in recent virtual screening studies (see Table 2.3 and 2.4).

publications other terms might be preferred. Such models can be easily applied to a set of vectors with a corresponding target value, as exemplified in Chapter 4 for chemical compound fingerprints and associated cytotoxicity values.

**Convolutional neural networks** Convolutional neural networks (CNNs) [13, Chapter 9] are a special kind of network where computations in the hidden layer make use of convolutions. They are most commonly applied in image classification, where their forte is extracting features in a picture, such as edge detection [21, 169, 170]. 1D, 2D, and 3D convolutions can be used depending on the input data. In the one-dimensional case, the data might be akin to time series. 2D convolutions are used on planar grid-like structures, such as images. 3D convolution can be applied to three-dimensional tensors, such as 3D images. Although convolutional neural networks exhibit excellent performance, they have the disadvantage of having a fast-growing number of parameters as the network becomes deeper, especially in the 3D case, making the training slow. In the area of binding affinity, successful predictions have been made using as input the 3D representation of the protein-ligand binding site [159].

**Recurrent neural networks** Recurrent neural networks, or RNNs [13, Chapter 10], are dissimilar from MLPs and CNNs in their ability to reuse internal information, that can be thought of as loops in the network (see Figure 2.4 (D)). RNNs are well suited for sequential data, such as sentences. The input at a certain time makes use, employing a series of computations, of the input at the previous time, leading to its name "recurrent". In molecular prediction, SMILES encoding of molecules can be interpreted as sequential data and RNNs were successfully applied in the QSAR context [127].

**Graph neural networks** Graph neural networks (GNNs) [171] are models that require graph-structured data. Loosely put, the input should represent a set of vertices (or nodes) and some structure that determines the relationship between them. The graph neural network will act on the nodes while taking into account the neighbors of each node, i.e. the set of vertices that are connected to that particular node. Each node is updated to a latent feature vector that contains information about its neighborhood, hence resembling a convolution operation and leading to the denomination of graph convolution neural network (GCNN). This latent node representation, often called an embedding, can be of any chosen length. These embeddings can further be used for node prediction, if some properties of vertices are of interest, or they can be aggregated to obtain predictions at the graph level, i.e. some information on the graph as a whole.

A subtlety that can be added to GNNs is a gated recurrent unit (GRU),

which is a way to moderate the flow of information coming from other nodes and from previous computation steps [172]. These particular units are often applied at an intermediary stage, once the embeddings from previous steps are set. GRUs consists of two gates: the update gate is responsible for updating the weights and biases and the reset gate for controlling the amount of information that can be forgotten. Graph networks using GRUs are called gated graph neural networks (GGNNs) [172].

Graph attention neural networks (GANNs) [173] are graph neural networks with an added attention mechanism. In a graph setting, this can be viewed as ranking the nodes in a neighborhood of a given vertex and giving more or less importance to each of them. Certain atoms, and therefore interactions, may have more significance for a given task. This can be represented by including distances between atoms in the adjacency matrix, as in [163]. A feature node is then obtained using a linear combination of its neighbors taking the attention coefficient into account.

More details on graph neural networks can be found in the review by Zhou et al. [174]. In the context of molecular prediction, dozens of examples use GNNs, as summarized in the review by Wieder et al. [175].

### Model evaluation strategies and metrics

To assess the performance of any machine learning method, the data is commonly split into training and test sets. The model is trained on the training set and evaluated by comparing the predicted labels to the given labels on the hold out (test) set. Here, the metrics used in the Recent developments section are simply listed, for a detailed description please refer to the Evaluation strategies and metrics.

For regression tasks, the metrics reported are the mean squared error (MSE) and the root mean squared error (RMSE). For classification tasks, the area under the ROC—receiver operating characteristic—curve (AUC), the accuracy or the enrichment factor (EF) are used. For both regression and classification, the Pearson correlation coefficient  $R$ , the Spearman's correlation coefficient  $\rho$  or the coefficient of determination  $R^2$  may be reported.

Cross-validation (CV) is very often used to estimate the prediction error and usually performed using five or ten folds, and the results are reported as mean performance ( $\pm$  standard deviation). Additionally, CV can be used for hyper-parameter tuning. Please refer to the work by Hastie et al. [168] for a full description of this method.

### 2.2.3 Data sets and benchmarks in virtual screening

The quality and quantity of data sets in the biomedical field have increased largely over the last years, boosting the usage of ML and DL models in drug discovery. The main source of freely available 3D structural information of



proteins as well as protein-ligand complexes is the well-known Protein Data Bank (PDB) [41], holding, as of March 2021, 175,282 biological macromolecular structures [109], a number which includes proteins that have been solved numerous times. Furthermore, labeled bioactivity data, i.e. the measured activity of a specific compound against a target of interest, are necessary for training, validating, and testing DL models. The two most well-known examples of bioactivity databases are PubChem [176] and ChEMBL [107]. Note that while for the pair-based methods, the information in the latter databases is sufficient, for complex-based methods the bioactivity and structural information has to be linked. Below, the most widely used labeled bioactivity data sets and their composition will be introduced (see Table 2.1).

### Structure-based data sets

**PDBbind** The PDBbind [177] database collects experimentally measured binding affinity data from scientific literature for a large number of biomolecular complexes deposited in the PDB database. In the current release, PDBbind v.2019 provides binding data of a total of 21,382 biomolecular complexes as the general set, including 17,679 protein-ligand complexes. Furthermore, a refined and a core set with higher quality data are extracted from the general set. In the refined set, the 4,852 protein-ligand complexes meet certain quality criteria (e.g. resolution, R-factor, protein-ligand covalent bonds, ternary complexes or steric clashes, and type of affinity value). The core set, resulting after further filtering, provides 285 high-quality protein-ligand complexes for validating docking or scoring methods.

**BindingDB** BindingDB [178] is a publicly accessible database, which collects experimental protein-ligand binding data from scientific literature, patents, and other. The data extracted by BindingDB includes not only the affinity, but also the respective experimental conditions (i.e. assay description). BindingDB contains 2,229,892 data points, i.e. measured binding affinity for 8,499 protein targets and 967,208 compounds, including 2,823 protein-ligand crystal structures with mapped affinity measurements (requiring 100% sequence identity), as of March 1, 2021 [179].

**BindingMOAD** BindingMOAD (Mother of All Databases) [180, 181] is another database focused on providing combined high-quality structural and affinity data, similar to PDBbind. BindingMOAD (release 2019) contains 38,702 well-resolved protein-ligand crystal structures, with ligand annotation and protein classifications, of which 15,964 are linked to experimental binding affinity data with biologically-relevant ligands.

### Bioactivity data sets

Table 2.1: **Bioactivity data sets.** The table lists common labeled data sets used in virtual screening studies. Freely available data is increasing each year and is an essential element for affinity prediction using machine and deep learning models. The table summarizes the name, the size and the content covered as well as links to the respective website.

Name	Size and content <sup>a</sup>	Availability
PDBbind v.2019	structures and activities: general: 21,382 refined: 4,852 core: 285	<a href="http://www.pdbbind.org.cn/">http://www.pdbbind.org.cn/</a>
BindingDB	2,823 structures and activities ; 2,229,892 activities	<a href="https://www.bindingdb.org">https://www.bindingdb.org</a>
Binding- MOAD 2019	38,702 structures 15,964 structures and activities	<a href="https://bindingmoad.org/">https://bindingmoad.org/</a>
PubChem BioAssay 2020	> 280 million activities	<a href="https://pubchem.ncbi.nlm.nih.gov/">https://pubchem.ncbi.nlm.nih.gov/</a>
ChEMBL v.28	17,276,334 activities	<a href="https://www.ebi.ac.uk/chembl/">https://www.ebi.ac.uk/chembl/</a>

<sup>a</sup>Structures refers to protein-ligand X-ray structures. Activities refer to measured compound-target bioactivity data points as reported in the respective data source.

**PubChem BioAssay** PubChem [176] is the world’s largest freely available database of chemical information, e.g. chemical structures, physicochemical properties, biological activities, patents, health, safety and toxicity data, collected from more than 770 data sources (as of March 2021 [182]). PubChem BioAssay contains bioactivity data, more precisely biological assay descriptions and test results, for compounds and RNAi reagents assembled from high-throughput screens and medical chemistry studies. As of March 2021,

PubChem deposited more than 109 million unique chemical structures as well as over 280 million bioactivity data points collected from more than 1.2 million biological assays experiments.

**ChEMBL** ChEMBL [107, 183] is a widely used open-access bioactivity database with information about compounds and their bioassay results extracted from full-text articles, approved drugs and clinical development reports. The last release, ChEMBL v.28, contains 14,347 targets and over 17 million activities, which are collected from more than 80,000 publications and patents [108], alongside deposited data and data exchanged with other databases such as BindingDB and PubChem BioAssay.

**Target-family specific data sets (such as kinases)** Since some protein families sparked a special interest in pharmaceutical sciences due to their central therapeutic role, target-family specific data sets have been composed. Kinases, for example, play a major role in many diseases and have been extensively studied, also computationally [184], for drug design. Data sets comprise profiling studies, such as the one reported by Davis et al. [185], which provides information about a full matrix of 72 inhibitors tested against a set of 442 kinases in competition binding assays (measured dissociation constant  $K_d$ ). To be able to combine data from different sources, reported as different bioactivity measurements (e.g.  $IC_{50}$ ,  $K_d$  and inhibition constant  $K_i$ ), Tang et al. [186] derived a kinase inhibitor bioactivity (KIBA) score, an adjusted Cheng-Prusoff model, which allows to integrate data from the above mentioned measurement types and to assemble a freely available drug-target bioactivity matrix of 52,498 chemical compounds and 467 kinase targets, including 246,088 KIBA scores.

### Benchmarking data sets

The above introduces data sets commonly used for training ML models in the context of VS. Nevertheless, defined benchmarking data sets are needed for a standardized comparison among different methods and studies [187]. Here, frequently used benchmarking data sets for structure- and ligand-based VS are introduced (see Table 2.2).

**CASF** The comparative assessment of scoring functions (CASF) benchmark [188] is developed to monitor the performance of structure-based scoring functions. In the latest version, CASF-2016, the PDBbind v.2016 core set was incorporated with 285 high-quality protein-ligand complexes assigned to 57 clusters. Scoring functions can be evaluated by four metrics: (1) The scoring power, indicating the binding affinity prediction capacity using the Pearson correlation coefficient  $R$  [189]. (2) The ranking power, showing affinity-ranking capacity using the Spearman correlation coefficient  $\rho$  [190, 191].

Table 2.2: **Benchmark data sets.** Evaluating novel models on labeled benchmark data is crucial for any machine learning task, including deep learning-based virtual screening. The table depicts some commonly used databases with their respective size, the origin of the data, provided information (affinity or activity) as well as their availability through websites.

Name	Size	Data source	Label	Availability
CASF-2016	57 targets 285 complexes	PDBbind	affinity	<sup>a</sup>
DUD-E	102 targets 22,886 actives 50 decoys per active	PubChem, ZINC	active/ decoy	<sup>b</sup>
MUV	17 targets ~ 90,000 compounds	PubChem, ZINC	active/ decoy	MUV@TU Braunschweig <sup>c</sup>

<sup>a</sup><http://www.pdbbind.org.cn/casf.php>

<sup>b</sup><http://dude.docking.org/>

<sup>c</sup><https://www.tu-braunschweig.de/pharmchem/forschung/baumann/muv>

(3) The docking power, using the root mean square deviation (RMSD) [192] to analyze how well the method has placed the ligand (pose prediction).  
 (4) The screening power measures the enrichment factor (EF) [193], showing the ability of the function to prioritize active over inactive compounds. Note that the CASF team has evaluated scoring functions from well-known docking programs, such as AutoDock vina [194], Gold [149], and Glide [195], and published the results on their website [188].

**DUD(-E)** The directory of useful decoys (DUD) [196] is a virtual screening benchmarking set providing 2,950 ligands for 40 different targets, and 36 decoy molecules per ligand drawn from ZINC [84]. Decoys, i.e. negative samples, are chosen to have similar physicochemical properties, but dissimilar 2D topology to the respective active molecules. DUD-E [197] is an enhanced and rebuilt version of DUD, with 22,886 active compounds and affinity values against 102 diverse targets. On average, 50 decoys for each active compound are selected. DUD-E is usually used in classification tasks to benchmark molecular docking programs with regard to their ability to rank active compounds over inactive ones (decoys).

**MUV** The maximum unbiased validation (MUV) data set [198] is based on the PubChem BioAssay database mostly for ligand-based studies, using

refined nearest neighbor analysis to select actives and inactives, to avoid analogue bias and artificial enrichment. It contains 17 different target data sets, each containing 30 actives and 15,000 inactives. Note that in contrast to DUD(-E) decoys, the inactives have experimental validated activities.

**Benchmarking set collections** Note that several collections of data sets for the purpose of benchmarking molecular ML models, with focus on model architectures and/or encodings, have recently been made freely available. These include, but are not limited to, (1) MoleculeNet [104] to benchmark molecular machine learning, currently providing a total of 700,000 compounds tested on diverse properties from not only quantum mechanics, but physical chemistry, biophysics (including MUV and PDBBind) and physiology; (2) Therapeutics Data Commons (TDC) data sets [199], including 22 machine learning tasks and 66 associated data sets covering various therapeutic domains; or (3) the work by Riniker and Landrum [200], covering compounds for 118 targets from MUV, DUD and ChEMBL with focus on benchmarking fingerprints in ligand-based virtual screening.

## 2.3 Recent developments

In this section, recent developments in virtual screening (VS) are described and specifically how deep learning (DL) helps to improve drug-target binding, i.e. activity/potency prediction. Our review focuses on methods using protein and ligand information (see Figure 2.1), either in form of a protein-ligand complex (complex-based) or considering protein and ligand as two independent entities (pair-based/PCM). It is imperative to state that the aim of this section is not to directly compare different studies or models, but to describe them and put them into context. The list of abbreviations can be found at the end of this review.

### 2.3.1 Complex-based models

In this section, recent methods that require complex structure information, usually explicitly or implicitly described by the interactions between the protein and the ligand, will be discussed (see Table 2.3). The various methods are grouped by the type of encodings used for the complex structure: IFPs, 3D grids, graphs and other (see Methods & Data section).

**Interaction fingerprint-based studies** Interaction fingerprints, which are often used for binding site comparison [100, 208], have also been successfully applied to VS. Due to the difference in length of some IFP implementations, binding site independent IFPs are more commonly used for machine learning applications.

Table 2.3: **Complex-based models.** Summary of recent work using a protein-ligand complex for active molecule or binding affinity prediction. The year of publication, the name of the authors or the model, the complex encoding and the machine/deep learning model(s) are shown in the respective columns. Classification (class.) implies predicting e.g. hit or non-hit, whereas regression (reg.) evaluates an affinity measure, e.g.  $pIC_{50}$  values. CNNs, coupled with 3D grids, have become frequent in state-of-the-art studies.

Year	Name	Complex encoding	ML/DL model	Framework
2010	Sato et al. [152]	IFP <sup>a</sup>	SVM, RF, MLP	class.
2016	Wang et al. [201]	IFP	Adaboost-SVM	class.
2019	Li et al. [202]	IFP	MLP	class.
2018	gnina [158]	3D grid	CNN	class.
2018	KDEEP [159]	3D grid	CNN	reg.
2018	Pafnucy [157]	3D grid	CNN	reg.
2018	DenseFS [203]	3D grid	CNN	class.
2019	DeepAtom [160]	3D grid	CNN	reg.
2019	Sato et al. [204]	3D grid	CNN	class.
2019	Erdas-Cicek et al. [162]	3D grid	CNN	reg.
2019	BindScope [161]	3D grid	CNN	class.
2018	PotentialNet [164]	graph	GGNN	reg.
2019	Lim et al. [163]	graph	GANN	class.
2017	TopologyNet [165]	topol. <sup>b</sup>	CNN	reg.
2019	Math-DL [205]	topol.	GAN, CNN	reg.
2018	Cang et al. [206]	topol.	CNN	reg.
2016	DeepVS [167]	atom contexts	CNN	class.
2019	OnionNet [207]	atom pairs	CNN	reg.
2020	Zhu et al. [166]	atom pairs	MLP	reg.

<sup>a</sup>interaction fingerprints

<sup>b</sup>algebraic topology

In 2009, Sato et al. [152] combined machine learning (among others SVM, RF, and MLP) and the pharmacophore-based interaction fingerprint (Pharm-IF) are incorporated for screening five selected protein data sets *in silico*. For training a model per protein, the respective co-crystallized ligands served as active samples (between 9 and 197) and the Glide docking poses of 2,000 randomly selected decoys from PubChem as negative samples. For the test set, 100 active samples (after clustering) were drawn from the StARlite database (data now contained in ChEMBL) together with 2,000 negatives samples (as above) and all compounds were docked using Glide. The combination of SVM and Pharm-IF performed best with a high mean  $EF_{10\%}$  of 5.7 (over the five protein sets) compared to Glide scores (4.2) and a residue-based IF (PLIF) model (4.3), as well as a high mean AUC value of 0.82 compared to Glide (AUC 0.72) and PLIF model (AUC 0.78). Interestingly, in this study, the Pharm-IF SVM model outperformed the respective MLP model (average  $EF_{10\%}$  4.42, AUC 0.74). In 2016, Wang et al. [201] used ensemble learning to improve the SVM performance (using Adaboost-SVM) with the Pharm-IF encoding for two proteins from the same data set and gained even higher  $EF_{10\%}$  values.

In 2019, Li et al. [202] introduced an application of a target-specific scoring model to identify potential inhibitors for 12 targets from the (S)-adenosyl-L-methionine-dependent methyltransferase (SAM MTase) family. In total, 1,740 molecules were collected from experimental data and from the DUD-E website (446 actives and 1,294 decoys), and docked using Glide. An MLP was chosen and the complexes encoded by the TIFP. The data set was randomly split into training and test sets with a 10 : 1 ratio. In a binary classification experiment, the MLP showed e.g. a AUC of 0.86 and a  $EF_{5\%}$  of 3.46 on the test set, and thus, outperformed the traditional docking tools Glide (0.75 and 2.97), and AutoDock vina (0.61 and 0.99).

**3D grid-based studies** Many methods using a 3D grid representation of a protein-ligand complex—comparable to pixels in 3D images—for affinity prediction, have evolved over the last years [156, 157, 160, 161], especially due to the increased popularity of deep CNNs.

One of the first published models, AtomNet [156] uses a CNN, composed of an input layer, i.e. the vectorized 3D grids, several 3D convolutional and fully-connected layers, as well as an output layer, which assigns the probability of the two classes: active and inactive. Among other data sets, the DUD-E benchmark, consisting of 102 targets, over 20,000 actives and 50 property matched decoys per active compound, was used for evaluation. 72 targets were randomly assigned as training set, the remaining 30 targets as test set (DUDE-30). For each target, a holo structure from the scPDB [209] is used to place the grid around the binding site and multiple poses per molecule are sampled. Finally, the grid is fixed to a side length of 20 Å and

a 1 Å grid spacing, in which each grid point holds some structural feature such as atom-type or IFP. On the DUDE-30 test set, AtomNet achieves a mean AUC of 0.855 over the 30 targets, thus outperforming the classical docking tool smina [210] (mean AUC of 0.7). Furthermore, AUC values greater than 0.9 were reported for 46% of the targets in the DUDE-30 test set.

Similarly, BindScope [161] voxelizes the binding pocket by a 16 Å grid of 1 Å resolution, molecules are placed using smina, and each voxel is assigned a distance-dependent input based on eight pharmacophoric feature types [159]. The 3D-CNN model architecture was adapted from DenseNet [211] and yields a mean AUC of 0.885 on the DUD-E benchmark in a five-fold cross-validation (folds were assigned based on protein sequence similarity-based clusters). Comparable AUC values on the DUD-E set were reported by Ragoza et al. [212] (mean AUC of 0.867), a similar grid-based CNN method, which outperformed AutoDock vina on 90% of the targets.

DeepAtom [160] uses a 32 Å box with 1 Å resolution and assigns a total of 24 features to each voxel (11 Arpeggio atom types [213] and an exclusion volume for ligand and protein respectively) in individual channels to encode the protein-ligand complex. The PDBbind v.2016 served as baseline benchmark data, split into 290 complexes for testing and 3,767 non-overlapping complexes between the refined and core sets for training and validation. In particular, each original example gets randomly translated and rotated for data argumentation, which aims to improve the learning capacity. The performance of the built 3D-CNN model, trained on the PDBbind refined set, in predicting the affinity for the core set in a regression setting was reported with a low mean RMSE of 1.318 ( $R$  of 0.807) over five runs. In this case, DeepAtom outperformed RF-Score [214], a classical ML method (mean RMSE of 1.403), as well as Pafnucy [157] (mean RMSE of 1.553), a similar 3D-CNN method, trained and applied to the same data using their open-source code. Note that in the original publication, Pafnucy [157] achieved prediction results with an RMSE of 1.42 on the PDBbind core set v.2016. In a further study, the training set for DeepAtom was extended by combining BindingMOAD and PDBbind subsets, resulting in 10,383 complexes. While the mean RMSE of DeepAtom slightly decreased to 1.232, the  $R$  value increased to 0.831 for the PDBbind core set.

The presented examples show the effectiveness of 3D grid-based encodings and CNN models for affinity prediction, which seem to be well suited to implicitly capture the variety of information important for ligand-binding. However, disadvantages are the high memory demand of 3D grids and CNNs, as well as the implicit grid boundary definition to capture the protein-ligand interactions.



**Graph-based studies** Graph neural networks have proven to be some of the most effective deep learning models, easily reaching state-of-the-art performance. In this context, two recent applications of such models in virtual screening are described.

Lim et al. [163] construct a graph representation of the protein predicted binding pose complex, obtained using smina [210] and train a graph neural network to successfully predict activity. The node feature vector concatenates atomic information from the ligand and from the protein. The features considered for both are the one-hot encoding of the following atomic properties: type (10 symbols), degree (6 possibilities for 0 to 5 neighbors), number of hydrogens (5 entries for 0 to 4 possible attached Hs), implicit valence of electrons (6 entries) and a binary entry for aromaticity. This leads to 28 entries for the ligand, another 28 for the protein, generating a feature vector of size 56. The 3D information is encoded in the two matrices  $A^1$  and  $A^2$  described in Section 2.2.1, for covalent and non-covalent interactions, respectively. The model applies four layers of GAT (gate augmented graph attention) to both  $A^1$  and  $A^2$ , before aggregating the node information using summation. A 128-unit fully-connected layer is then applied to this vector, which leads to binary activity prediction. DUD-E is used for training and testing the VS performance, where the training set contains 72 proteins with 15,864 actives and 973,260 inactives and the test set another 25 proteins with 5,841 actives and 364,149 inactives. The AUC value on the test data set reaches 0.968, which is high compared to the value of 0.689 obtained with the smina docking tool. The model also obtains better scores than other deep learning (DL) models such as the CNN-based models AtomNet [156] and the one developed by Ragoza et al. [212]. The same trend holds for the reported PDBBind data set study. However, when testing their model and docking results on external data sets such as ChEMBL and MUV, the performance drops, hinting to the fact that the DL model might not be able to generalize to the whole chemical space.

The graph convolution family PotentialNet developed by Feinberg et al. [164] predicts protein-ligand binding at state-of-the-art scales. The atomic features are atom type, formal charge, hybridization, aromaticity, and the total numbers of bonds, hydrogens (total and implicit), and radical electrons. The structure between the atoms is described using  $A \in \mathbb{R}^{N \times N \times N_{et}}$ , the extended representation of an adjacency matrix, as described in Section 2.2.1. The PotentialNet model uses a Gated Graph Neural Network (GGNN), which means that unlike GNNs, the update function is a GRU, leading to the new node vector, depending on its previous state and the message from its neighbors, in a learned manner. PotentialNet also considers different stages, where stage 1 makes use of only the bonded part of the adjacency matrix, leading to node updates for connectivity information, stage 2 considers spatial information, and stage 3 sums all node vectors from ligands before applying a fully-connected layer for binding affinity prediction.

The model is trained on complexes of the PDBBind v.2007 using a subset of size 1,095 of the initial refined set for training, and then tested on the core set of 195 data points. The model reaches a test  $R^2$  value of 0.668 and a test  $R$  value of 0.822, outperforming RF-Score ( $R$  of 0.783) and X-Score ( $R$  of 0.643) [164, Table 1]. However, similar results were reported by the CNN-based model TopologyNet [165], introduced below.

**Other studies** In MathDL [205] and TopologyNet [165], the complexes—and thus the interactions between protein and ligand—are encoded using methods from algebraic topology. In MathDL, advanced mathematical techniques (including geometry, topology and/or graph theory) are used to encode the physicochemical interactions into lower-dimensional rotational and translational invariant representations. Several CNNs and GANs (Generative Adversarial Networks) are trained on the PDBbind v.2018 data set and applied on the data of the D3R Grand Challenge 4 (GC4), a community-wide blind challenge for compound pose and binding affinity prediction [215]. The models are among the top performing methods in pose prediction on the beta secretase 1 (BACE) data set with an RMSD <sup>†</sup> of 0.55 Å and a high  $\rho$  of 0.73 in affinity ranking of 460 Cathepsin S (CatS) compounds (additionally good performance was reported on the free energy set of 39 CatS compounds). TopologyNet [165], a family of multi-channel topological CNNs, represent the protein-ligand complex geometry by a 1D topological invariant (using element-specific persistent homology) for affinity prediction and protein mutation. In the affinity study, a TopologyNet model (TNet-BP) is trained on the PDBbind v.2007 refined set (excluding the core set) and achieves an  $R$  of 0.826 and an RMSE of 1.37 in  $pK_d/pK_i$  units <sup>‡</sup>. Thus, TNet-BP seems to outperform other well-known tools such as AutoDock vina and GlideScore-XP on this data set (note that the results are adopted from the original study by Li et al. [216]).

DeepBindRG [217] and DeepVS [167] focus on the interacting atom environments in the complex using atom pair and atom context encodings, respectively. DeepBindRG, a CNN model trained on PDBbind v.2018 (excluding targets that appear in the respective test set), achieves good performance on independent data sets such as the CASF-2013 and DUD-E subsets, with an RMSE varying between 1.6 and 1.8 for a given protein and an  $R$  between 0.5 and 0.6. With these values, DeepBindGP performs slightly better than AutoDock vina, while being in a similar range as Pafnucy [157]. DeepVS, another CNN model, trained and tested on the DUD data set using leave-one-out cross-validation outperforms, with an AUC of 0.81, AutoDock vina 1.2 which has an AUC value of 0.62.

---

<sup>†</sup>The RMSD [192] measures the average distance of atomic positions, e.g. between a co-crystallized ligand and the docked poses.

<sup>‡</sup>The  $pK_d$  and  $pK_i$  values describe the negative decimal logarithm of  $K_d$  and  $K_i$  values, respectively.

### 2.3.2 Pair-based models

In this section, pair-based/PCM models from the literature are presented (see Table 2.4). As discussed above, pair-based methods do not require the crystal structure of a protein, nor the docked pose of a ligand. Indeed, the ligand is modeled independently from the protein and vice versa. This framework resembles proteochemometric (PCM) models [101, 102], whereas the cross-term, including some interactions between the ligand and the protein, which can be used in PCM, is not present in the herein reported pair-based setting. The discussed studies are grouped by the type of ligand encoding they use: SMILES, fingerprint and graph.

**Ligand as SMILES** In 2018, Öztürk et al. [128] proposed the DeepDTA (Deep Drug-Target Binding Affinity Prediction) regression model which takes the SMILES and a fixed length truncation of the full protein sequence as features for the ligand and protein, respectively. In the study, two kinase-focused data sets are used: the Davis data [185] and the KIBA data [186] with roughly 30,000 and 250,000 data points, respectively. The first reports  $K_d$  values, which represents the dissociation constant, while the second reports Kinase Inhibitor BioActivity (KIBA) scores, which combines information from  $IC_{50}$ ,  $K_i$  and  $K_d$  measurements. As input for the CNN, both the SMILES and the protein sequence are label encoded independently. The authors apply convolutions to the embeddings of each object, before concatenating them and predicting the  $pK_d$  value or KIBA score, depending on the data set used. The data are randomly split into six equal parts where one of them is used as a test set to evaluate the model and the five remaining compose the folds for cross-validation and parameter tuning. On the Davis and KIBA test sets, the model exhibits an MSE of 0.261 and 0.194, respectively (see [128, Table 3-4]), which outperforms baselines such as KronRLS [223], a variation of least squares regression and SimBoost [224], a tree-based gradient boosting method. The success of the deep learning model could be explained by the use of convolution layers which are able to extract information from the protein-ligand pair.

The same authors extended DeepDTA to WideDTA [218]. This time, instead of only considering the SMILES label encoding for the ligand, substructure information is also included where a list of the 100,000 most frequent maximum common substructures defined by Woźniak et al. [225] are used. For the protein description, approximately 500 motifs and domains are extracted from the PROSITE database [226] and label encoded. The deep learning architecture is similar to DeepDTA, but WideDTA does achieve slightly better results, e.g. an MSE of 0.179 on the KIBA data [218, Table 5].

In another study, Karimi et al. [142] use SMILES as ligand and structural property sequences as protein descriptors to predict protein-ligand affinities.

Table 2.4: **Pair-based models.** The listed models consider information from the protein and the ligand, but the encodings are built independently of each other. The year of publication, the name of the authors or the model, the ligand and the protein encodings and the machine/deep learning model(s) are shown in the respective columns. Classification (class.) implies hit or non-hit, whereas regression (reg.) evaluates an affinity measure, e.g.  $pIC_{50}$  values. Graphs and associated GCNNs have become prominent in recent years.

Year	Name	Ligand encoding	Protein encoding	ML/DL model	Framework
2018	DeepDTA [128]	SMILES	full seq. <sup>a</sup>	CNN	reg.
2019	WideDTA [218]	SMILES & MCS <sup>b</sup>	full seq. & domains/motifs	CNN	reg.
2019	DeepAffinity [142]	SMILES	struct. property seq.	RNN+ CNN	reg.
2016	DL-CPI [219]	substructure FP <sup>c</sup>	domains	MLP	class.
2018	Kundu et al. [133]	div. feat. FP	div. feat. <sup>d</sup> FP	RF & SVM & MLP	reg.
2018	Sorgenfrei et al. [138]	Morgan FP	z-scales	RF	class.
2019	DeepConv-DTI [220]	Morgan FP	full seq.	CNN	class.
2019	Tornø and Altman [126]	graph	graph	GCNN	class.
2020	DGraphDTA [221]	graph	graph	GCNN	reg.
2018	PADME [222]	graph (or Morgan FP)	seq. comp. <sup>e</sup>	GCNN (or MLP)	reg.

<sup>a</sup>sequence

<sup>b</sup>maximum common substructure

<sup>c</sup>fingerprint

<sup>d</sup>diverse feature count, physicochemical and structural properties

<sup>e</sup>composition

The first learning task is an auto-encoder which aims at giving a latent representation of a compound-target pair. Once the neural network is trained in an unsupervised setting, the resulting fingerprint is then fed to recurrent plus convolution layers with an attention mechanism to predict  $pIC_{50}$  values. The BindingDB data set [227], containing close to 500,000 labeled protein-compound pairs after curation, is used for their study. After removing four protein classes as generalization sets, the remaining  $\sim 370,000$  pairs are split into train (70%) and test (30%) sets. On the test set, while RF yields an RMSE of 0.91 and a Pearson’s  $R$  of 0.78, the DeepAffinity model reaches an  $R$  of 0.86 and a lower RMSE of 0.73 [142, Table 2], thus outperforming conventional methods such as RF. A GCNN is also tested on the graph encoding of the compounds, but this alternative did not show improvements with respect to the SMILES notation.

**Ligand as fingerprint** The DL-CPI model suggested by Tian et al. [219] stands for Deep Learning for Compound-Protein Interactions and applies four fully-connected hidden layers to a 6,404 long input binary vector which is the concatenation of compound and protein features; 881 entries for substructure identification in the ligand and another 5,523 for Pfam [141] identified protein domains. Using five-fold cross-validation, the AUC varies between 0.893 and 0.919 depending on the ratio of negative samples in the data set for DL-CPI and between 0.687 and 0.724 for an RF model [219, Table 2]. The high accuracy performance is explained by the abstraction coming from the hidden layers of the network.

The study by Kundu et al. [133] compares various types of models trained and tested on a subset of 2,864 instances of the PDDBind v.2015 data set. The 127 long input feature vector combines features of the protein, such as the percentage of amino acids, the accessible surface area of the protein, the number of chains, etc., as well as physicochemical (e.g. molecular weight, topological surface area, etc.) and structural properties (e.g. ring count) of the ligand. RF is shown to outperform models such as MLP and SVM in the task of predicting inhibition constant ( $K_i$ ) and dissociation constant ( $K_d$ ) values. One possible reason for these results might originate from the size of the data set: RF models can be very successful when the available data is small.

The study undertaken by Sorgenfrei et al. [138] focuses on the RF algorithm. They encode the ligand with Morgan fingerprints and note that using z-scales descriptors from the binding site of the protein highly improves the performance of the model compared to the baseline which only considers the one-hot encoded ID of the target. The data set used contains over 1,300,000 compound-kinase activities and comes from combined sources such as ChEMBL and the KIBA data provided by Tang et al. [186]. The activity threshold for  $pIC_{50}$  values ( $pIC_{50} = -\log_{10}(IC_{50})$ ) is set at 6.3. On

a test set in which both target and compound are left out during training, the AUC reaches a value of 0.75 [138, Table 1], justifying the usefulness of pair-based/PCM methods in hit identification.

Morgan fingerprints are also used in the study by Lee et al. [220] to represent the ligand, while the full raw protein sequence is used as protein input. The deep learning model, DeepConv-DTI, consists of convolutions applied to the embeddings of the full protein sequence (padded to reach the length of 2,500), which are then combined with the ligand descriptors in fully-connected layers to predict whether the drug is a hit or not. The model is built on combined data from various open sources, such as DrugBank [228], KEGG [229] and IUPHAR [230]. After curation and generating negative samples, the training set contains close to 100,000 data points. The model is externally tested on PubChem where both protein and compound had not been seen during training. DeepConv-DTI reaches an accuracy close to 0.8 [220, Figure 3D] and seems to outperform the DeepDTA model by Öztürk et al. [128], see [220, Figure 4].

**Ligand as graph** In the study by Torng and Altman [126], GCNNs are used on both target and ligand. The residues in the binding pocket of the protein correspond to the nodes and the 480 long feature vector, computed with the program developed by Bagley and Altman [231], represents their physicochemical properties. The small molecule is also treated as a graph and properties such as the one-hot encoded atomic element, degree, attached hydrogen(s), valence(s), and aromaticity, are included in the 62 long feature vector. Graph convolutional layers are applied to both graphs independently and the resulting vectors from both entities are concatenated. A fully-connected layer is applied to the concatenated vector to learn the interaction between the molecule and the target, leading to an interaction vector which is then used to predict binding or non-binding. The model is trained on the DUD-E data set and externally tested on MUV, and reaches an AUC value of 0.621, which is better than results from 3D CNNs, AutoDock vina and RF-Score [126, Table 2].

The research undertaken by the authors Jiang et al. [221] uses a similar workflow, where GNNs are applied to both the ligand graph and the protein graph based on the contact map. More precisely, the atomic properties of the ligand nodes are the one-hot encoding of the element (44 entries), the degree (11 entries), the total (implicit and explicit) number of attached hydrogens (11 entries), the number of implicit attached hydrogens only (11 entries), and aromaticity (binary value), leading to a vector of length 78. The atomic properties of the protein include the one-hot encoding of the residue (21 entries), the position-specific scoring matrix for amino acid substitutions (21 entries), and the binary values of the 12 following properties: being aliphatic, aromatic, polar, acidic, or basic, the weight, three differ-

ent dissociation constants, the pH value and hydrophobicity at two different pH values. The contact map, which can be computed using the PconsC4 tool [232] directly from the protein sequence, can be used as a proxy for the adjacency matrix. Given the graph representation for the ligand as well as the protein, three graph convolutional, pooling layer, and fully-connected layers are subsequently applied to both ligand and protein independently, then concatenated, and finally the binding affinity is predicted after another two fully-connected layers. The deep learning model is called DGraphDTA, which stands for "Double Graph Drug-Target Affinity predictor". The MSE on the Davis and KIBA data sets are as low as 0.202 and 0.126, respectively and DGraphDTA seems to give better results than both DeepDTA and WideDTA, see [221, Table 7, 8]. This hints to the fact that graph representations are well suited for drug-target interaction prediction.

The PADME model, an acronym for "Protein And Drug Molecule interaction prEdiction" [222], suggests two variants for ligand encoding in the regression context of drug-target interaction prediction. The ECFP fingerprint (implemented as the Morgan fingerprint in the code) as well as the graph encoding are used along side a 8,421 long protein feature vector describing the sequence composition (8,420 entries for the amino acid, dipeptide, and tripeptide composition computed using the propy tool [233]) and one entry for phosphorylation. The deep learning model is adapted depending on the encoding of the ligand. In the case of circular fingerprint, the protein and ligand vectors are concatenated to form a "combined input vector", on which fully-connected layers are then applied. In the graph setting, a graph layer is applied to the ligand, resulting in a vector, which is then again concatenated to the protein vector as in the previous case. The regression models (either graph or circular fingerprint) consistently outperform baseline models such as KronRLS and SimBoost. The simulations are run on several kinase data sets such as Davis [185] and KIBA [186]. Using cross-validation schemes that involve testing the model on the fold for which no protein was trained on, the RMSE on the KIBA data with the PADME graph setting is 0.6225 and on the Davis data with the circular fingerprint setting is 0.5639 [222, Table 2]. This study provides further evidence that deep learning models could indeed improve drug-target prediction compared to standard machine learning algorithms.

## 2.4 Conclusion and discussion

Over the last decade(s), a wave of deep learning methods and applications to boost virtual screening, i.e. affinity—but also other properties, such as ADMETox—prediction, has emerged. This development is coupled to the availability of more and more compounds, structures and mapped bioactivity data, together with novel encoding techniques and deep learning technolo-

gies. These include not only model architectures, that seem to fit well the nature of biological objects such as ligands and proteins, but also open-source software and computer hardware evolution.

Around thirty papers related to deep learning-based virtual screening are described in details in this review (see Table 2.3 and Table 2.4). Most of these studies were published between 2018 and 2020, giving an overview of the current state-of-the-art and the advancements of deep learning in the field. The encodings for protein and ligand (Section 2.2.1), the machine learning models (Section 2.2.2), the data sets (Section 2.2.3) as well as the model performances (Section 2.3) are reported and put in context. These studies show overall very promising results on typical benchmarks and often outperform the respective classical approach chosen for comparison, such as docking or more standard machine learning models. This is also exemplified on the Merck Molecular Activity Kaggle competition data, where deep neural networks have shown to routinely perform better than random forest models [234]. Similarly, in other blind challenges for pose and affinity prediction such as the D3R grand challenges, deep learning-based methods increasingly make it to the top ranges [215, Table 1]. One possible reason for such outstanding achievements may be explained by the way biological entities are encoded: for example, rather than using human-engineered descriptors, features are learned by the models. Also, novel encodings, such as voxels (where physicochemical atomic properties are pinned to locations in 3D space) and graphs (that describe the connectivity, bonded and non-bonded, between the atoms), seem to capture well the variety of information important for ligand-binding. For example, DeepAtom [160], a 3D grid-based method where each grid cell is assigned a different physicochemical property seems well suited to model the complexity of protein-ligand binding using 3D information. Encoding chemical and biological objects in graph form also seems to be very fitting, as shown in the study by Lim et al. [163] and the DGraphDTA model by Jiang et al. [221].

Nevertheless, several challenges still remain open and new ones have also emerged, including (1) precision of chemical encoding, (2) generalization of chemical space, (3) lack of (big and high-quality) data, (4) comparability of models, and (5) interpretability, which will be discussed in the following.

(1) *Precision of chemical encoding*: The better performance of structure-based methods using ML-based vs. classical SFs is often attributed to the avoidance of a pre-determined functional form of the protein-ligand complexes, meaning that the precision of the chemical description does not necessarily lead to more accurate binding affinity prediction [235]. Contributing factors might be associated with a) modeling assumptions, where more precise descriptions may introduce errors. b) The dependence of encoding and regression technique: more precise description might produce longer and sparser features which could be problematic in cases such as RF models. c) Restrictions to data in the bound state, neglecting contribution from



both partners in solvation and induced fit phenomena. Or missing consideration of conformational heterogeneity, where multiple conformations might co-exist with different probabilities.

(2) *Generalization of chemical space*: As mentioned in the work by Lim et al. [163], although some deep learning models perform outstandingly well, there seems to still exist some issues exploring the whole chemical space, a challenge also occurring in classical machine learning methods. Some less successful results have been detected when evaluating some models on external data sets, showing that since the data used for training is not representative of the immense chemical space, the model, instead of learning and exploring it, is rather memorizing patterns from it [236].

(3) *Lack of (big and high-quality) data*: Deep learning is very data greedy and usually, the bigger the training set is, the better the results. Goodfellow et al. [13] suggest that a model trained on a data set of size of the order of 10 million may surpass human performance. However, as previously discussed, biochemical data are still considerably smaller than, for example, image or video data sets. Therefore, depending on the data at hand, choosing more standard machine learning approaches, or more shallow neural networks, that require less parameter training, may perform just as well. Examples are shown in the studies by Kundu et al. [133], which employs random forest for activity prediction, or by Göller et al. [237], which summarizes DL and ML models for ADMETox predictions. Another alternative is to find a way to acquire more data, through, for example, data augmentation. In image classification, this can be done using image rotating, cropping, recoloring, etc., which can be adapted to virtual screening tasks. The Pharm-IF method [152] performs better with more crystal structures or by employing additional docking poses. DeepAtom [160] translates and rotates the protein-ligand complex to gain more training data. In QSAR predictions, using SMILES augmentation has also become popular as means to enlarge the training set [127, 129]. Note that not only the quantity of data, but also its quality is often unsatisfactory, such as low resolution of crystal structures or relying on docked poses, as well as activity data taken from various experiments (and conditions) providing different measurements, e.g.  $K_d$ ,  $K_i$ ,  $IC_{50}$  or  $EC_{50}$  (which is the measured half maximal effective concentration of a drug).

(4) *Comparability of models, benchmark data, open-source*: Reviewing a multitude of studies and wanting to compare and rank them is understandable, but also unreasonable for several reasons, starting with the data and the splits. Models that have been trained on different data or even different tasks should hardly be compared; a regression task or a classification task, even when using similar performance metrics, are not analogous. Assuming that the models do use the same data, if the splits are different, then the evaluation can no longer be directly compared. Assuming now that the splits are identical, then if the metrics used are different, again no fair compari-

son can be made, as pointed out by Feinberg et al. [164]. This means that there are a chain of elements that have to be considered before comparing and ranking methods blindly. To this end, two major elements become crucial: (1) open-source data and (2) open-source code. Having benchmark data sets freely available—together with a code basis—such as MoleculeNet [104], TDC [199] or work by Riniker and Landrum [200], and updated regularly, such as ChEMBL, is highly beneficial for academic research and method publication. Moreover, having access to the source code of newly developed methods and being able to reproduce results is also becoming more and more essential in the field especially as the number of models developed is becoming larger (as embraced by the FAIR principles [61]).

Moreover, while several data sets are available to benchmark the performance of different approaches in VS, Sieg et al. [187] recently elaborated on the need of bias control for ML-based virtual screening studies. Several types of biases exist. For example, domain bias, which may be due to insufficient generalization as discussed above, but still acceptable, if the models are applied in a narrow chemical space. Nevertheless, non-causal bias is dangerous, when there is correlation but no causation. While mainly focusing on structure-based ML models for VS on DUD, DUD-E and MUV, Sieg et al. [187] found that small molecule features dominated the predictions across dissimilar proteins even when structure-based methods/descriptors are used. Thus, special care needs to be taken when methods and descriptors are evaluated on benchmark sets, if the compilation protocol of the benchmark is suited for the context of the methodology. In another study, Chen et al. [238] also claim hidden analogue and decoy bias in the DUD-E database that may lead to superior performance of CNN models during VS. Thus, there is urgent need for bias control in benchmarking data sets, especially for structure-enabled ML-based VS.

(5) *Interpretability*: With the rise of deep learning, the complexity of the architectures and the depth of the models comes the issue of interpretability. Such models are often considered as black boxes and understanding the mechanism in the hidden layers is a challenge. However, research undertaken in this direction aims at deciphering what the algorithm has learned, see Chapter 4 and [239]. This may also be important in detecting bias in the data [187].

For further considerations on the type, quality and quantity of the data as well as the challenges of DL models built thereof to impact different areas of drug discovery, the reader is kindly referred to two recent reviews by Bender and Cortés-Ciriano [240, 241].

In this work, the recent progress in DL-based VS methods has been reviewed, exemplifying the boost in development and application over the last few years. While some challenges due to, for example data coverage and unbiased evaluation sets, molecular encoding and modeling the respective biological protein-ligand binding event still remain, the reported results show

the unprecedented advances in the field.

### **Data availability**

The Python code to generate most components of the figures in the review is available on GitHub at [https://github.com/volkamerlab/DL\\_in\\_VS\\_review](https://github.com/volkamerlab/DL_in_VS_review), using packages such as RDKit [131], NGLview [242], the Open Drug Discovery Toolkit (ODDT) [243] and PyMOL [244].



## Chapter 3

# Improving molecular property prediction using data augmentation

The contents of this chapter were published as Kimber, T. B., Gagnebin, M., & Volkamer, A. (2021). Maxsmi: Maximizing Molecular Property Prediction Performance with Confidence Estimation Using SMILES Augmentation and Deep Learning. *Artificial Intelligence in the Life Sciences, 1*, 100014 [2], under a Creative Commons license (CC-BY-NC-ND), <https://creativecommons.org/licenses/by-nc-nd/4.0/>. The content from this publication is presented here with the permission of Elsevier publishing.

### Contributions:

TBK conceived the project, laid out the theory with MG, implemented the algorithms, performed the computational experiments. TBK and MG analyzed and visualized the results. TBK wrote the paper. AV supervised the work.

### Chapter summary

Accurate molecular property or activity prediction is one of the main goals in computer-aided drug design. Quantitative structure-activity relationship (QSAR) modeling and machine learning, more recently deep learning, have become an integral part of this process. Such algorithms require lots of data for training which, in the case of physico-chemical and bioactivity data sets, remains scarce. To address the lack of data, augmentation techniques are increasingly applied in deep learning. Here, we exploit that one compound can be represented by various SMILES strings as means of data augmentation and we explore several augmentation techniques. Convolutional

tional and recurrent neural networks are trained on four data sets, including experimental solubility, lipophilicity, and bioactivity measurements. Moreover, the uncertainty of the models is assessed by applying augmentation on the test set. Our results show that data augmentation improves the accuracy independently of the deep learning model and of the size of the data. The best strategies lead to the Maxsmi models, the models that **maximize** the performance in **SMILES** augmentation. Our findings show that the standard deviation of the per SMILES prediction correlates with the accuracy of the associated compound prediction. In addition, our systematic testing of different augmentation strategies provides an extensive guideline to SMILES augmentation. A prediction tool using the Maxsmi models for novel compounds on the aforementioned physico-chemical and bioactivity tasks is made available at <https://github.com/volkamerlab/maxsmi>.

### 3.1 Introduction

Drug design is a time-consuming and costly process [245, 246] with high attrition rates [247]. It can be supported with *in silico* methods by guiding the design process, optimizing compounds, and discarding those with undesired properties at an early stage of development. In this context, computer-aided drug design (CADD) has become central in the drug discovery pipeline and is widely adopted in research and development in both academia and pharmaceutical companies.

Over the last few decades, there has been a keen interest in machine learning (ML) and more specifically deep learning (DL), which have been applied to a variety of areas, covering computer vision [248], speech recognition [249], as well as the life sciences. Only to name a few, AlphaFold 2 from DeepMind which predicts protein folding [10], PotentialNet which focuses on protein-ligand binding affinity [250], *de novo* molecular design suitable for compound optimization [251], and cytotoxicity prediction as in Chapter 4. Such excitement in DL may be explained by the main three following factors [13].

1. The gain of computational power through graphics processing units (GPUs) and tensor processing units (TPUs). Platforms such as Google Colaboratory [105] allow any user to exploit high performance computing resources without any cost and such free and easy access is unprecedented.
2. The ever growing amount of available data. More data are created and stored in databases every day in various fields. Many processes are automatized making data more accessible and usable either internally, as in pharmaceutical companies, or publicly. For example in academic research, in competitions, such as Kaggle [33], or in challenges, such as

D3R – the Drug Design Data Resource challenge [215] or the Tox21 challenge [252].

3. The advances in algorithms, making models perform better than ever before. Deep learning algorithms may surpass human performance if trained on a data set containing over 10 million data points, as suggested by Goodfellow et al. [13].

With the rise of ML/DL research, many applications have been extended to the field of molecular property and affinity prediction, even though insufficient data remains a challenge in the field.

Encoding molecular compounds in both human- and computer-readable formats is a necessary step in CADD. A convenient encoding is SMILES, or simplified molecular-input line-entry system [58]. As the name suggests, SMILES is a linear notation of a molecule based on atom and bond enumeration, as well as branch, ring closure, and disconnection specification. Several advantages arise from this compact representation.

1. The printable characters make SMILES easily readable by computers and decipherable by humans.
2. Being a single line, SMILES resemble words and are therefore cheap to store.
3. Such a notation is very popular and many open-source databases store compounds in SMILES.

However, there is a trade-off between readability and specification: having a compact encoding means losing detailed information about the molecule such as 2D or 3D features. Moreover, subtle chemistry rules such as aromaticity do not have a standard way of being handled [253].

The implementation of a SMILES given a compound can be described as follows: from any starting atom in the molecule, enumerate the atoms and bonds following a path in the molecular graph. Two aspects of this construction lead to the non-uniqueness of SMILES: 1. the atom to start the enumeration from, and 2. the path to follow along the graph. Therefore, one molecule can have many different valid SMILES, simply by starting the enumeration from a different atom or by choosing a different path. Nevertheless, in some settings, having a bijection between a molecule and its SMILES notation may be sought. For example, when determining the overlapping molecules from two data sets. In this context, most cheminformatics tools have their own algorithm implemented allowing them to always retrieve the same SMILES given a molecular graph, such a SMILES is called canonical [254].

As mentioned previously, deep learning being data greedy and both physico-chemical and bioactivity databases being meager, elaborate techniques have to be integrated to unleash the full potential of deep neural

networks. In this context, data augmentation in general [255, 256], and more specifically SMILES augmentation [127, 129, 257, 258], is a powerful assistance in molecular prediction. From a machine learning perspective, data augmentation allows the model to see the same object through different angles and has been successfully applied in image classification [248, 259], where images undergo transformations such as flipping, coloring, cropping, rotating, and translating. From a computational perspective, SMILES augmentation is advantageous because generating random SMILES is fast and memory efficient, and even though training a model may be more computationally expensive, it remains cheap to evaluate.

The first occurrence of SMILES augmentation in QSAR modeling was developed by Bjerrum [127], where affinity against dihydrofolate reductase (DHFR) is predicted on a small data set of 756 compounds. The model consists of long short-term memory (LSTM) layers and a fully connected layer for the normalized  $\log IC_{50}$  value. Each molecule in the data set is augmented on average 130 times. The model with SMILES augmentation reaches a test correlation coefficient of 0.68, a 0.12 increase with respect to the canonical model. From then on, several studies have built on the same idea, applying SMILES augmentation in QSAR modeling [257]. Moreover, convolutional neural networks have successfully been applied in the context of SMILES augmentation, outperforming models using traditional molecular descriptors [129, 258]. Such augmentation techniques have also emerged in related fields, such as retrosynthesis [260, 261] and generative modeling [262, 263]. While all these studies show the benefit of augmenting the data, none of them focus, to the best of our knowledge, on a systematic analysis on how to augment the data set best, and most decide *a priori* on an augmentation number. This study aims at filling this gap by offering a systematic augmentation approach, both in the augmentation strategies and by how much the data should be augmented. Moreover, a command-line interface is available for users interested in the prediction of physico-chemical properties for novel molecules and assessing the uncertainty of the prediction. To this end, all code is made freely available at <https://github.com/volkamerlab/maxsmi>.

## 3.2 Methods

In this section, we first describe different augmentation strategies which can be used for data augmentation when dealing with SMILES. Second, we illustrate how SMILES augmentation can be viewed as an ensemble learning technique when it comes to prediction. We then examine the deep learning models that are trained in this study.



### 3.2.1 Augmentation strategies

As discussed in the Introduction, one compound can have several valid SMILES, since both the starting atom and the path along the molecular graph used to generate the SMILES can differ. Here, the way a user can explore such random SMILES is detailed and five strategies to augment a single SMILES to multiple SMILES are described: no augmentation, augmentation with duplication, augmentation without duplication, augmentation with reduced duplication, and augmentation with estimated maximum. For the following sections, we assume that we are given a data set  $D$ , containing  $N$  pairs of {compound, label}. Label refers to the measured property, such as lipophilicity or solubility. The implementation of these strategies is based on the open-source cheminformatics software RDKit [131].

#### No augmentation

The level zero to augmentation is having no augmentation or, in other terms, augmentation of zero. This means that given a data set  $D$  with  $N$  compounds, the "no augmentation" version of  $D$  also contains  $N$  SMILES. More specifically, in this setting, the SMILES associated with each compound is the canonical SMILES.

#### Augmentation with duplication

Generating random SMILES implies picking at random an initial atom and following a random path along the molecular graph. Augmenting the data set  $D$  by  $m$  means that for each of the  $N$  compounds in  $D$ ,  $m$  instances of random SMILES are drawn and the associated labels are matched for each compound. In this case, augmenting  $D$  by  $m$  would result in the augmented data set containing  $N \times m$  data points. In this scenario, all molecules in  $D$  are multiplied by the same factor  $m$ . Consequently, smaller molecules, with fewer SMILES variations, will contain more duplicates whereas larger molecules are more likely to cover a diverse set of random SMILES. A disadvantage of such an augmentation strategy is that SMILES corresponding to small molecules will be over-represented in the data set and could create a bias in model training.

#### Augmentation without duplication

Removing duplicated entries is common in data wrangling [264]. In the context of SMILES augmentation, this translates to discarding duplicates after having generated a number of random SMILES. For data set  $D$ , the final number of data points after augmentation varies according to the augmentation number, i.e. the number of times a sample is drawn from the valid SMILES space, and the size of the molecules in the data set. A disadvantage

of such an augmentation strategy is that small molecules, which presumably possess fewer unique SMILES representatives, will be under-represented in the data set and could create a bias in model training.

### Augmentation with reduced duplication

In order to find a compromise between keeping or removing all duplicates, the notion of augmentation with reduced duplication is introduced. In this setting, only a fraction of the number of duplicates is kept. Mathematically speaking, if the data set  $D$  is augmented by  $m$ , then a function  $f(m)$  which grows slower than linear is used to control the number of replicas kept for each SMILES. Sensible functions would be the squared root function  $f(m) = \sqrt{m}$  or the natural logarithm  $f(m) = \ln(m)$ , the former being used for the experiments in this study.

A corner case of the former three augmentation strategies by augmentation number  $m$  is when  $m = 1$ . In this instance, a random SMILES will be generated and the number of data points would still be  $N$ , as in the "no augmentation" case, with the difference that "no augmentation" contains canonical SMILES only.

### Augmentation with estimated maximum

The final strategy described is augmentation with estimated maximum, which aims to cover a wide range of the valid SMILES space for a given compound, or in other words, to generate a number of unique SMILES that depends on the compound. In our study, the implementation of this augmentation strategy randomly samples SMILES corresponding to a compound, and the sampling process is stopped once the same SMILES string has been generated a pre-defined number of times. The experiments of this study set 10 generations of the same SMILES as a stopping criterion. It is noteworthy that the number of SMILES this method generates is highly dependent on the size of the compound, unlike the previous methods which always generate a number of SMILES bounded by  $m$ . For example, our implementation of this augmentation strategy generated 50,659 unique SMILES variations for the compound given by the canonical SMILES CC(=O)C1(C)CCC2C3C=C(C)C4=CC(=O)CCC4(C)C3CCC21C, whereas only three were generated for the canonical SMILES C=CC=C, namely C(=C)C=C, C(C=C)=C, and C=CC=C.

### 3.2.2 SMILES augmentation as ensemble learning for compound prediction and confidence measure

The application of data augmentation strategies during training has proven to be successful, as shown in previous works [248, 259]. In QSAR modeling

particularly, SMILES augmentation is not only beneficial on the training set [129], but there are also advantages of augmenting the test set, or more generally, an unlabeled data set, as explained in this section.

Let us assume that a model  $M$  with a set of parameters  $\Theta$  was trained for a certain number of epochs. Let us consider an unlabeled data set containing  $N$  compounds for which we want to make predictions. Each compound  $C$  can be augmented using random SMILES:  $S_1(C), S_2(C), \dots, S_k(C)$ , where  $k$  depends on the strategy. The model  $M_\Theta$  produces a prediction for each of those SMILES, i.e. for  $i \in \{1, \dots, k\}$

$$\hat{y}_i(C) = M_\Theta(S_i(C)), \quad (3.1)$$

leading to a *per SMILES* prediction rather than a *per compound* prediction. Using an aggregation function  $A: \mathbb{R}^k \rightarrow \mathbb{R}$ , such as the mean, a prediction for compound  $C$  can be computed as

$$\hat{y}(C) = A(\hat{y}_1(C), \dots, \hat{y}_k(C)). \quad (3.2)$$

Such aggregation can be viewed as a consensus among the SMILES prediction and interpreted as ensemble learning for a given compound.

Additionally, if the standard deviations of the predictions are computed, they can be interpreted as a confidence in the molecular property or activity prediction. If the standard deviation is large, then there is a high variation in the per SMILES prediction, and the model is uncertain in its per compound prediction. An illustration of such a molecular prediction is shown in Figure 3.1. Following the rationale by Tagasovska and Lopez-Paz [265], the aleatoric and epistemic uncertainties are often intertwined; the uncertainty computed in our work rather falls into the aleatoric category, a type of uncertainty linked to the model predictions and the randomness in the input data [265–267].

### 3.2.3 Deep learning models

Neural networks are powerful algorithms that allow accurate predictions on various tasks. In the case of QSAR/ML/DL modeling and more specifically the use of SMILES representation, two types of models can be applied, convolutional [13, Chapter 9] and recurrent [13, Chapter 10] neural networks.

In this study, comparing deep learning models and how they perform with respect to data augmentation is one of the key focuses. To this end, three types of models are architected and trained, namely 1D and 2D convolutional neural networks (CONV1D, CONV2D) as well as a recurrent neural network (RNN). The architecture of the recurrent network consists of an LSTM layer, followed by two fully connected layers of 128 and 64 units, respectively. It was inspired by Bjerrum [127], in which an LSTM layer is followed by a single 64 unit fully connected layer. Using a similar approach,

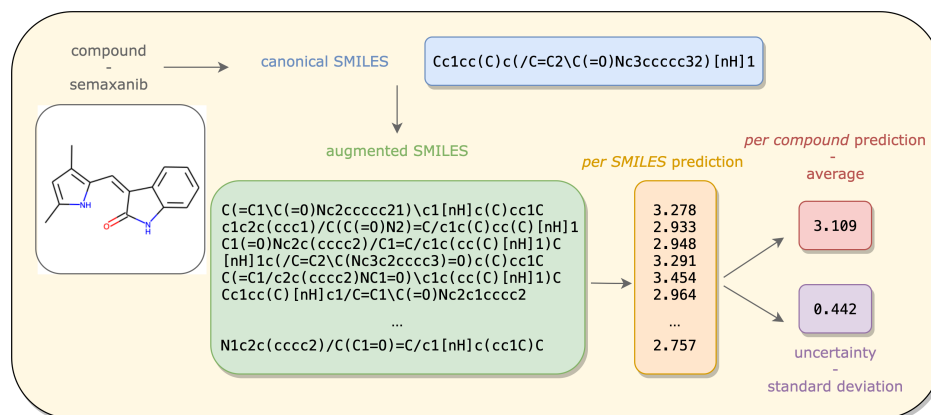


Figure 3.1: **Compound prediction and confidence measure thanks to SMILES augmentation.** Given a compound represented by its canonical SMILES, a set of random SMILES are generated. The trained machine learning model produces a prediction for each of the SMILES variations. Aggregating these values leads to a per compound prediction and computing the standard deviation is interpreted as an uncertainty in the prediction.

a single 1D convolutional layer of kernel size 10 and stride 1 is applied in the CONV1D model. Two fully connected layers follow the convolution. The CONV2D adheres to the same pattern but instead of using a 1D convolution, a 2D convolution operation is performed using one single channel. Finally, all three architected models stay consistent in the depth of the network and remain shallow.

In this study, all deep learning models are trained for 250 epochs, using mini-batches of size 16, where the mean squared error is the considered loss. Optimization is done with stochastic gradient descent and a learning rate of 0.001. Note that a fixed number of epochs is used in this study, but for sake of completeness, three sample models with early stopping were also run. The results with and without early stopping did not change significantly (data not shown). Moreover, some models were trained by adapting the number of epochs with respect to the augmentation number, but this only proved to overfit the training set and yielded the same results on the test set as training with 250 epochs (data not shown).

### 3.3 Data and experimental setup

This section introduces the data sets used in this study, namely their provenance as well as the required preprocessing. Furthermore, a step by step instruction for efficient SMILES augmentation is described. Finally, the evaluation design and experimental setup are covered.

### 3.3.1 Provenance

The data in this research come from two sources: MoleculeNet [104], and the ChEMBL database [54], chosen for two main reasons. 1. They are freely available and easily downloadable or retrievable. 2. They are often used as benchmarks for study comparison [129, 268, 269]. For tasks in MoleculeNet, we focus on physico-chemical prediction tasks and retrieve the data from the three following sets of varying sizes, all available as part of DeepChem [53] at <https://deepchem.readthedocs.io/en/latest>.

1. Measured water solubility is referred to as the ESOL data set [52]. The raw data contains 1,128 data points. This data set is further processed to only include molecules with at most 25 heavy atoms for experimental setup and is referred to as ESOL\_small.
2. The FreeSolv [77] data set consists of 642 pairs of SMILES and experimental hydration free energy of small molecules in water (kcal/mol).
3. The lipophilicity data set originates from ChEMBL [54] and contains 4,200 pairs of SMILES and experimental values of octanol/water distribution coefficient.

Bioactivity data can be found in large quantities in ChEMBL. To date, over 18 million activities are stored in the database, covering more than 14,000 targets and two million compounds [270]. Among targets, kinases are a well studied protein family due to their involvement, among others, in cancer and inflammatory diseases [271]. Kinodata, from the Openkinome organization [272], provides an already curated data set of human kinase bioactivities, retrieved from one of the latest versions to date of ChEMBL (version 28) and is freely available at <https://github.com/openkinome/kinodata>. Moreover, for this study, Kinodata is further filtered for the epidermal growth factor receptor (EGFR) kinase [273], since it is known to be an important drug target. Its UniProt identifier is given by P00533 [274]. Affinity towards the EGFR kinase is quantified using  $pIC_{50}$  values, the negative base 10 logarithm of  $IC_{50}$  [275]. Information about the data set provenance and size are detailed in Table 3.1.

### 3.3.2 Data preprocessing and input featurization

In order to train a deep learning neural network on data containing molecular compounds, the data set undergoes preprocessing and compounds encoding.

Once the data sets are retrieved from their original source, invalid SMILES, detected by RDKit [131], not available (NA) values and disconnected compounds, marked by a dot in a SMILES, are removed. Molecules are transformed to the canonical SMILES representation, using RDKit functionalities.

Table 3.1: **Data sets for this study.** Size of the data sets before and after preprocessing, as well as the size of the training and test sets before applying an augmentation strategy, and the provenance of the data.

Data set	Size before pre- processing	Size after pre- processing	Train set 80%, before augmentation	Test set 20%, before augmentation	Provenance
ESOL	1,128	1,128	902	226	MoleculeNet <sup>a</sup>
ESOL_small	1,128	1,068	854	214	MoleculeNet
FreeSolv	642	642	513	129	MoleculeNet
Lipophilicity	4,200	4,199	3,359	840	MoleculeNet
Affinity (EGFR)	6,026	5,849	4,679	1,170	Kinodata <sup>b</sup>

<sup>a</sup><https://deepchem.readthedocs.io/en/latest>

<sup>b</sup><https://github.com/openkinome/kinodata>

For model training, the SMILES are one-hot encoded, based on a dictionary of unique symbols constructed from the SMILES in the data. Atoms represented by two letters, such as Br for bromine or Cl for chlorine, as well as @@ for chirality specification, are treated as if single symbols. Finally, all inputs are padded up to the length of the longest SMILES. The reader is kindly referred to Chapter 2 for further details on one-hot encoding and padding.

### 3.3.3 Important steps in SMILES augmentation

When processing SMILES for augmentation, some technical aspects are essential. This section assumes a training and test split, but the rationale is the same in the presence of a validation set.

Firstly, it is important that the data are first split and then augmented, rather than augmented and split. In the latter case, one compound could have SMILES appearing in both training and testing leading to most probably excellent performance, but yet statistically incorrect.

Secondly, storing values such as the length of the longest SMILES or the dictionary of characters should be done not before but after augmenting the data. Indeed, augmentation may lead to the extension of the dictionary as well as the lengthening of SMILES. For example, the canonical SMILES CCCC consists of the letter C solely and contains four characters. However, one of its possible random variations is C(C)CC, which not only introduces new characters, such as the opening "(" and closing ")" of branches but is composed of six characters. Therefore, critical values such as length and dictionary should be retained after augmentation.

Finally, these same values should be computed on the union of the training and the test set for the smooth training and evaluation of the model. In-

deed, if the dictionary of characters is only built on the basis of the SMILES in the training data, there might be additional atoms, or characters in the test set that the model will not recognize and will not be able to one-hot encode. Moreover, if the length of the longest SMILES is taken from the training set and not the union of the training and test sets, augmentation on the test set could produce a longer SMILES than the longest one in the training set, leading to dimensionality errors.

For all the above reasons, it is important for machine learning engineers to abide by the steps as described in this section for statistically correct results, as well as programmatic error-free model training and evaluation.

### 3.3.4 Experimental setup and model evaluation

In order to draw a conclusion on the efficiency of data augmentation, three data sets of varying sizes are considered, namely ESOL, FreeSolv, and lipophilicity (see the Provenance section). For each of these sets, the data are split once into 80% training and 20% test set, with a fixed random seed for testing to be consistent with the augmentation schemes. Given all possible combinations between the five augmentation strategies and different augmentation numbers, the three deep learning models, and the various data sets, including cross-validation would have added considerable computational costs and has therefore not been implemented in this study.

For model evaluation, the root mean squared error (RMSE) [276] on the test set is reported, so that the lower the RMSE value, the better the model. However additional information such as the measure of goodness of fit, also known as the R<sup>2</sup> value [277], on both training and test sets, as well as the time required for model training and evaluation are also stored.

Five augmentations strategies are studied: No augmentation, which considers the canonical SMILES representation. The augmentations with, without, and with reduced duplication, for numerous augmentation numbers: a finer grid from 1 to 20 with a step size of 1, and a coarser grid from 20 to 100 with a step size of 10. Finally, the estimated maximum strategy where a SMILES representation has to be generated 10 times for the process to stop. For this last strategy, the ESOL\_small data set (see Table 3.1) is used to keep the augmentation to a reasonable time-scale. For the same reason, the same augmentation strategy is not run on the lipophilicity data set.

The augmentation strategies are applied to both the training set and the test set, so that for example, if the FreeSolv training data set is augmented 20 times without duplication, then so would the FreeSolv test set.

Ensemble learning is applied on each test set and the mean is used as aggregation. However, a user could easily adapt it to another function, such as the median. The standard deviation is stored for each compound in the test set.

Moreover, a Random Forest (RF) model [278] is used as a baseline, with

all default parameters from Scikit-learn [27]. The inputs to the model are the Morgan fingerprints of radius 2 and length 1,024. Augmentation strategies as discussed above are not applicable in the context of fingerprints.

Simulations are run on a GeForce GTX 1080 Ti, provided by the central HPC cluster of the Freie Universität Berlin [279].

### 3.3.5 Code and documentation

All code is written in Python 3 [280] following PEP8 style guide [281] and is freely available at <https://github.com/volkamerlab/maxsmi>. Results of this study can be found at the same link. Examples and documentation, generated via Read the Docs [282], can be found at <https://maxsmi.readthedocs.io/en/latest/>.

Package management is done with Anaconda [283]. RDKit [131] is used for cheminformatics, PyTorch [29] for deep learning, and other popular packages such as Scikit-learn [27], Numpy [32], and Pandas [284] for general purposes. Continuous integration is deployed with Github actions [285] ensuring runs on Linux, Mac, and Windows operating systems. Unit tests are done with Pytest [286], and code coverage is measured via Codecov [287].

## 3.4 Results and discussion

This section gives a thorough analysis of the results that are obtained using the experimental setup described in the previous section and provides the reader with guidelines on data augmentation applicable to new data and exemplified with affinity measurements towards the EGFR kinase. An example of the user prediction for compounds through a simple command-line interface is described.

### 3.4.1 SMILES augmentation improves model performance

As mentioned previously, deep learning models are data greedy and the findings of our study reinforce this statement by a systematic analysis of performance differences when augmenting the input data. Feeding a neural network with different SMILES representations of the same compound leads to better performing models, as shown in Figures 3.2, B.2, and B.3. Improvements are also visible with respect to the baseline model. These observations are made on all three physico-chemical data sets, namely ESOL, FreeSolv, and lipophilicity, independently of the data set size that ranges between approximately 600 and 4,000 compounds (see Table 3.1). For example, the ESOL performance with no augmentation has an RMSE value of 0.839 for the CONV1D model, whereas the performance of the same model with reduced augmentation and  $m = 70$  achieves an RMSE as low as 0.569, see Figure 3.2. As the number of augmentation increases, the RMSE values



become smaller, indicated by lighter shades of purple in Figures 3.2, B.2, and B.3. Note that at first, as the augmentation number increases in the single digits, there is a clear increase in the performance of the models. For example, for the lipophilicity data set augmented with duplication and the CONV2D model, the single random SMILES model has an RMSE value of 1.309 and reaches values below 1 as of an augmentation number of 4 (see Figure B.3). On the ESOL data set, the RNN model without duplication starts at an RMSE of 1.016 and reaches values below 0.8 after only an augmentation of 5 (see Figure 3.2). However, the performance steadily reaches a plateau. For example, the RMSE of the CONV1D model trained on FreeSolv is slightly above 1 as of 20 number of augmentation and fluctuates around this value thereafter, as shown in Figure 3.3. Similar observations can be made for ESOL and lipophilicity. Using the same model, the RMSE on ESOL reaches a plateau around 0.60 at 40 augmentation steps (see Figure B.5) and lipophilicity around 0.60 at 60 (see Figure B.6). This result suggests the following:

1. There does not seem to be one optimal value that particularly stands out.
2. A trade-off between performance and computation time must be found. As expected, the computation time increases as the number of data points increases, as shown in Figure B.7.

### Deep learning model performance by architecture

Not only does augmenting the data set overall help the learning for all three considered tasks, but so is the case for all three deep learning architectures. This leads to the observation that augmentation improves performance independently of the deep learning model, suggesting that for any future QSAR study for molecular property prediction using SMILES and deep learning, SMILES augmentation should be the method of choice. However, in this particular study and these particular deep learning architectures, results point to the fact that the CONV1D model tends to outperform the RNN model, which itself seems to outperform CONV2D. As shown in Figure 3.4, on the ESOL data using augmentation with reduced duplication, as of an augmentation number of 40, the RMSE value of the CONV2D model fluctuates around 0.7, the RNN model around 0.65 and the CONV1D around 0.6, promoting the latter model to best performing model. This exhibits the power of convolutions and their ability to extract relevant features in compounds based on one-hot encoded SMILES input. This also implies that although applying 2D convolutions to SMILES is programmatically feasible, 1D convolutions are better suited than 2D convolutions, the latter having shown great success in image classification. Indeed, when considering the one-hot

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	0.964	1.009	1.016	0.975	0.978	1.020	0.964	0.986	1.022
2	0.785	0.787	0.964	0.786	0.768	0.935	0.785	0.769	0.943
3	0.785	0.726	0.896	0.784	0.962	0.899	0.785	0.728	0.805
4	0.732	0.761	0.881	0.718	0.761	0.847	0.739	0.774	0.817
5	0.716	0.748	0.791	0.716	0.723	0.789	0.712	0.730	0.793
6	0.666	0.743	0.788	0.679	0.695	0.771	0.673	0.714	0.760
7	0.660	0.676	0.773	0.667	0.670	0.775	0.658	0.683	0.764
8	0.712	0.692	0.743	0.666	0.700	0.744	0.672	0.721	0.733
9	0.642	0.761	0.727	0.642	0.655	0.718	0.641	0.671	0.715
10	0.646	0.689	0.729	0.663	0.706	0.696	0.647	0.692	0.752
11	0.638	0.668	0.696	0.620	0.720	0.760	0.639	0.653	0.696
12	0.606	0.666	0.715	0.615	0.652	0.673	0.613	0.678	0.670
13	0.621	0.692	0.698	0.613	0.664	0.682	0.615	0.669	0.720
14	0.633	0.656	0.702	0.631	0.645	0.670	0.628	0.631	0.687
15	0.622	0.640	0.676	0.596	0.652	0.672	0.626	0.650	0.681
16	0.624	0.680	0.726	0.615	0.662	0.638	0.624	0.660	0.670
17	0.608	0.675	0.689	0.615	0.661	0.667	0.626	0.657	0.658
18	0.615	0.671	0.705	0.606	0.633	0.660	0.592	0.697	0.745
19	0.605	0.664	0.666	0.604	0.658	0.642	0.599	0.749	0.682
20	0.615	0.654	0.656	0.604	0.649	0.630	0.605	0.666	0.632
30	0.612	0.664	0.626	0.592	0.668	0.637	0.619	0.653	0.609
40	0.583	0.684	0.626	0.595	0.656	0.620	0.589	0.674	0.605
50	0.593	0.688	0.641	0.583	0.662	0.601	0.599	0.661	0.617
60	0.589	0.666	0.616	0.584	0.659	0.608	0.584	0.692	0.638
70	0.588	0.660	0.607	0.577	0.675	0.589	0.569	0.659	0.592
80	0.582	0.647	0.642	0.573	0.675	0.632	0.587	0.651	0.622
90	0.589	0.676	0.600	0.598	0.664	0.633	0.579	0.654	0.632
100	0.580	0.650	0.591	0.596	0.658	0.646	0.582	0.684	0.623
	no augmentation			estimated maximum			baseline		
	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	RF		
	0.839	0.895	0.930	0.576	0.657	0.683	1.102		

Figure 3.2: **Test RMSE using data augmentation on the ESOL data set.** The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the ESOL data set is augmenting the data set 70 times using a reduced number of duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

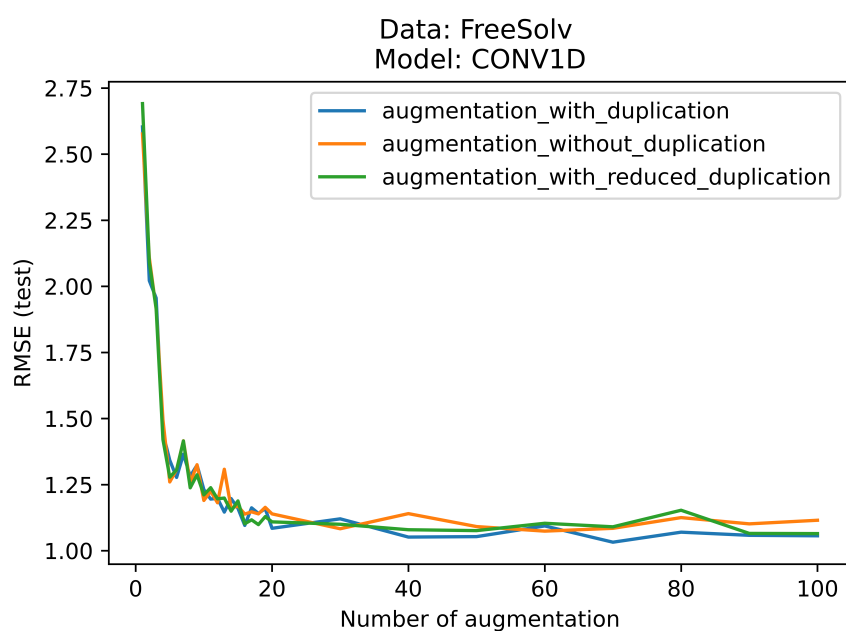


Figure 3.3: **Performance reaches a plateau independently of the augmentation strategy.** The performance of the CONV1D model trained and evaluated on the FreeSolv data set reaches a test RMSE value slightly above 1 as of 20 augmentation steps and fluctuates around this value thereafter, for all augmentation strategies: with, without, and with reduced duplication. For the ESOL and lipophilicity data, see Figures B.5 and B.6.

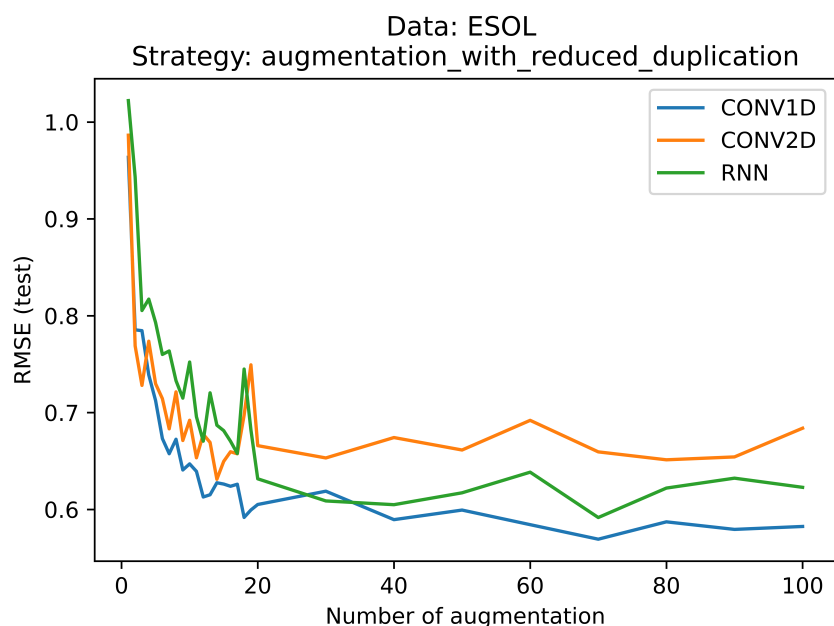


Figure 3.4: **The 1D convolutional (CONV1D) model outperforms the recurrent (RNN) and 2D convolution (CONV2D) models.** The figure shows the evolution of the root mean squared error (RMSE) on the test set with respect to the number of augmentation using reduced duplication on the ESOL data. CONV1D outperforms RNN, which outperforms CONV2D.

encoded matrix, SMILES are more similar to words, in which the position of the atoms is important, rather than to images.

### **There is no *best* augmentation strategy applying to all data sets**

From an augmentation strategy point of view, conclusions are not straightforward. The three augmentation strategies, namely with, without, and with reduced duplication, all perform similarly well, without one standing out. For example, the test RMSE on the FreeSolv data set trained using the CONV1D model reaches values just above 1 for all three strategies, as shown in Figure 3.3.

Moreover, generating a large portion of the SMILES space using the strategy with estimated maximum surprisingly does not lead to the best results. On the ESOL data set, this strategy reaches a test RMSE of 0.683 using RNN, whereas the same model but using strategies with, without, and with reduced duplication already outperforms the estimated maximum as of an augmentation number of 19 and onward, as shown in Figure 3.5. Although less obvious than in the ESOL case, a similar conclusion can be made on the FreeSolv data set and for example the CONV1D model, as shown in

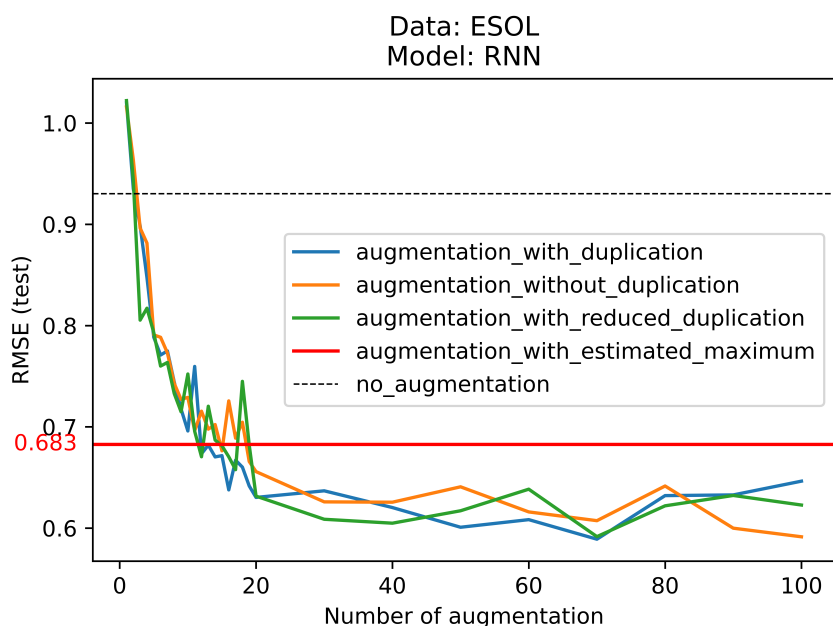


Figure 3.5: **Generating a large portion of the SMILES space does not necessarily lead to the best performance.** Even though the RNN model is presented with SMILES variations that cover a large portion of the SMILES space using the augmentations strategy with estimated maximum, on the ESOL data set, this strategy does not achieve the best results.

Figure B.4. This suggests that there might be a point of saturation, where the neural network stops learning, even though being fed more data.

### Maxsmi models: best performing model per data set

From the results of the experiments, as mentioned previously, there does not seem to be one augmentation strategy that fully stands out, neither does a particular model. However, from a purely numerical standpoint, there is an optimal performance value and this value is highlighted in yellow in Figures 3.2, B.2, and B.3. For the ESOL data set, the tuple of (model, augmentation number, augmentation strategy) that yields best performance is the CONV1D model, an augmentation number of 70 and keeping a reduced number of duplicates. For the FreeSolv data set, the same model but generating 70 random SMILES keeping all duplicates is the best setting. Finally, for lipophilicity, generating 80 random SMILES and removing duplicates leads to the best performance. Given these three best models, we select them for further analysis, henceforth calling them Maxsmi models and summarized in Table 3.2.

Table 3.2: **The best augmentation strategies define the Maxsmi models.** After training three data sets (ESOL, FreeSolv, and lipophilicity) on various deep learning models (CONV1D, CONV2D, and RNN), using different augmentation numbers and strategies, the setting that yields the best performance, or lowest root mean squared error (RMSE) on the test set is selected and named the Maxsmi model.

Data	Model	Augmentation number	Augmentation strategy	Test RMSE
ESOL	CONV1D	70	With reduced duplication	0.569
FreeSolv	CONV1D	70	With duplication	1.032
Lipophilicity	CONV1D	80	Without duplication	0.593

### Performance comparison between canonical and random SMILES

One interesting observation from this study is the performance comparison between training a model with canonical SMILES versus training a model using one random SMILES representation, in other terms, augmentation of 1. The canonical model systemically outperforms the model that uses a random SMILES. More specifically, for the ESOL data set, the canonical model reaches an RMSE value of 0.839 using CONV1D, whereas the random version 0.964 with the same model. In the FreeSolv and lipophilicity cases, the canonical model yields an RMSE value of 1.963 and 0.994, versus 2.577 and 1.268 for random SMILES. A possible explanation for such an outcome is the simplicity in the canonical SMILES representation. The algorithm in RDKit produces the more readable SMILES representation, one that avoids branches, as well as nested branches. Table 3.3 shows some of these differences. For example, a random version might add brackets, where the canonical version has none (see the first row in Table 3.3), it might add sets of brackets, where the canonical version keeps them to a minimum (see the second row in Table 3.3) and the random version even allows nested brackets where the canonical version avoids them (see the last row in Table 3.3).

To conclude with this observation, if SMILES augmentation cannot be applied for future studies for any reason, practitioners are highly recommended to consider the canonical SMILES representation rather than a random one.

#### 3.4.2 Ensemble learning for compound prediction and confidence measure

Using the Maxsmi models established above, we look into more details at the information gained from ensemble learning for molecular prediction, and more specifically at the average and standard deviation computed from the per SMILES prediction. Feeding different SMILES representations to the

Table 3.3: **Models based on the canonical SMILES outperform the ones based on a single random SMILES.** The test prediction for a model trained and evaluated on the RDKit canonical SMILES systematically performs better than the same model trained and evaluated on a single random SMILES. ESOL is the prediction task leading to the values in the table.

Canonical SMILES	Random SMILES	True value	Canonical SMILES prediction (&error)	Random SMILES prediction (&error)
CCCCC	C(C)CCCC	-3.84	-2.87 (0.97)	-2.77 (1.07)
CCCC(=O)CC	C(=O)(CCC)CC	-0.83	-1.37 (0.54)	-1.65 (0.82)
CCCC(=O)OCC	C(OC(CCC)=O)C	-1.36	-1.14 (0.22)	-0.55 (0.81)

model and aggregating the prediction for each SMILES variation to obtain a single prediction per compound is valuable not only from a practical point of view where molecular prediction is more informative than a SMILES prediction, but it also allows the model to merge information coming from different perspectives of the same compound. Moreover, the standard deviation associated with the SMILES predictions allows to quantify the uncertainty of the prediction of the model toward a given compound. The higher the standard deviation for a molecule, the less concurrent are the predictions by the model, and thus, less confident.

### Difference between canonical vs. averaged prediction

Considering the Maxsmi models trained with their respective augmentations, we analyze the difference in prediction on the test set when using the canonical or the averaged prediction. More specifically, we compare the prediction error of the Maxsmi models when evaluated on the test set twice: once using the canonical SMILES for compound prediction and a second time averaging the per SMILES prediction using the same augmentation number and strategy which was used for training. For both evaluations, the error between the prediction and the true value is computed. Figure 3.6 shows the histogram of these errors on the ESOL data. As shown in the figure, more compounds have an error close to zero using the ensemble learning evaluation rather than the canonical, which incentives the use of ensemble learning for future studies. However, this gain is marginal and the canonical prediction performs similarly well compared to the averaged prediction. In light of the overall gain in the accuracy of the models, this indicates that augmentation during training is the more crucial step. As discussed in the following paragraph, an advantage of using augmentation on the test set is to estimate the confidence of the model in its prediction.

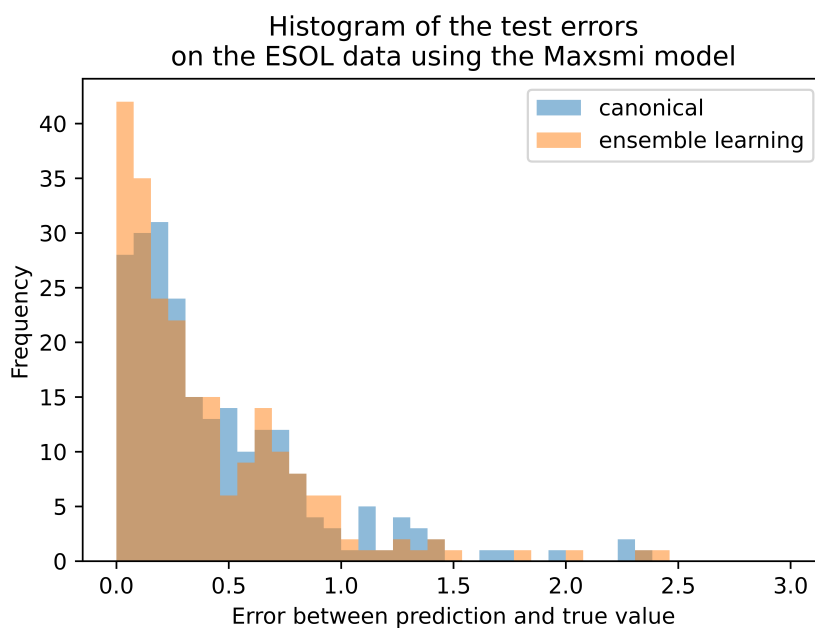


Figure 3.6: **Lower errors when evaluating the Maxsmi model using ensemble learning.** There are fewer errors in the evaluation of the trained Maxsmi models when using ensemble learning (i.e. the averaged per SMILES prediction) vs. the canonical prediction.



### More confident model implies smaller prediction error

As mentioned in the SMILES augmentation as ensemble learning for compound prediction and confidence measure section, computing the standard deviation of the per SMILES prediction provides a confidence measure in the compound prediction. In this section, we analyze the relationship between high confidence and small prediction error on the test set for the Maxsmi models. A way of visually evaluating uncertainty is to plot the confidence curve [266], which displays how the error varies with the sequential removal of compounds from lowest to highest confidence. Figure 3.7 shows the confidence curve of the Maxsmi model used on the FreeSolv data. As shown in the figure, as molecules with low confidence are sequentially removed, the mean prediction error decreases. In other words, the error vanishes as only compounds with the highest certainty predictions are kept, demonstrating a relationship between high confidence and small prediction error. Figure B.8 shows the confidence curves of the Maxsmi models for the ESOL and lipophilicity data. The general trend of the curve is decreasing in the ESOL case. Once the 10% of compounds with the highest confidence are kept, the error is below 0.25. However, in the lipophilicity case, although the general trend is also decreasing, even when keeping the 10% of compounds with the highest confidence, the error is still above 0.3.

### 3.4.3 Comparison to other studies

Given the results of the Maxsmi models, their performance is compared to other studies, namely MoleculeNet [104], CNF [129], and MolPMoFiT [257], that are trained and evaluated on the same data sets as Maxsmi, see Table 3.4.

The first considered study is MoleculeNet, where several molecular encodings and models are trained and evaluated, but where no augmentation is used. In MoleculeNet, the data is randomly split into training, validation, and test set, using an 80 : 10 : 10 ratio and run three times on different seeds. The best performing model on the test set for both ESOL and FreeSolv is a message passing neural network with an RMSE and standard deviation of  $0.58 \pm 0.03$  and  $1.15 \pm 0.12$ , respectively [104, Table S5]. On the lipophilicity data set, a slightly different graph model performs best with  $0.655 \pm 0.036$ . On all three tasks, the Maxsmi results perform better than MoleculeNet (see Table 3.4), suggesting that SMILES augmentation with shallow neural networks could perform at least as well as, if not better, than graph neural networks (GNNs).

The second study we consider is the Convolutional Neural Fingerprint (CNF) model [129, 258], in which SMILES augmentation is applied, generating unique representations for each compound, i.e. augmentation without duplication. The CNF model is evaluated using five-fold cross-validation (CV),

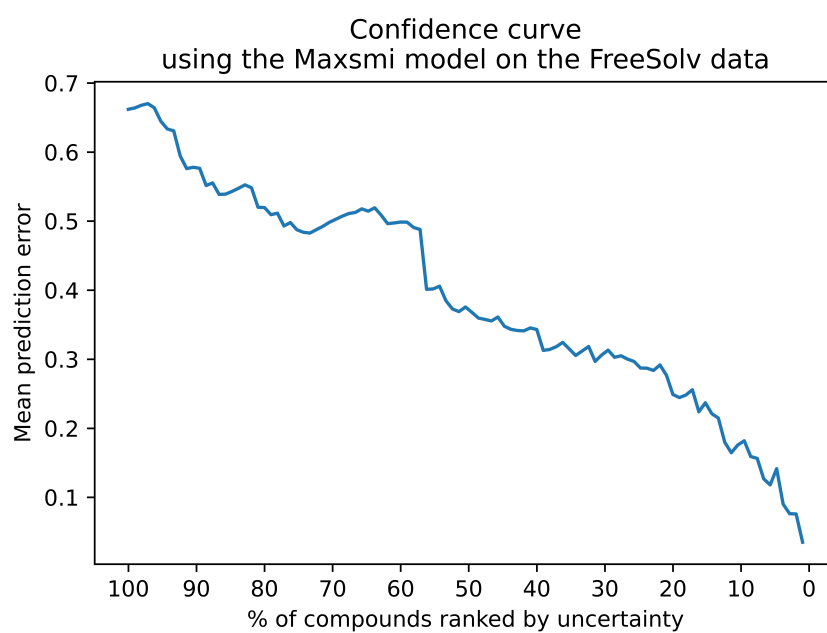


Figure 3.7: **More confident Maxsmi model on FreeSolv implies smaller prediction error.** The general trend of the confidence curve is decreasing, showing that as compounds with high uncertainty are removed, the error becomes smaller.

Table 3.4: **The Maxsmi models reach state-of-the-art results.** Comparison of four studies on the same data sets (ESOL, FreeSolv, and lipophilicity). The Maxsmi model outperforms most of the other models with a lower RMSE on a randomly split test set.

Abbreviations: RMSE = root mean squared error, std = standard deviation, CNN = Convolutional Neural Network, GNN = Graph Neural Network, RNN = Recurrent Neural Network, NA = not available, CV = cross-validation.

Study	Test RMSE ( $\pm$ std if available)			Split (Random)		Model
	ESOL	FreeSolv	Lipophilicity	Fold	Ratio % train:valid:test	
Maxsmi	<b>0.569</b>	<b>1.032</b>	0.593	Single	80 : 0 : 20	CNN
MoleculeNet[104]	0.58 $\pm$ 0.03	1.15 $\pm$ 0.12	0.655 $\pm$ 0.036	3	80 : 10 : 10	GNN
CNF[129]	0.62	1.11	0.67	5-fold CV	NA	CNN
MolPMoFiT[257]	NA	1.197 $\pm$ 0.127	<b>0.565 <math>\pm</math> 0.037</b>	10	80 : 10 : 10	RNN

however standard deviations are not reported. Test RMSE values are 0.62, 1.11 and 0.66 on the ESOL, FreeSolv, and lipophilicity data set, respectively, see [129, Table S1]. Similar to MoleculeNet, the Maxsmi model slightly outperforms these results on all three tasks. This suggests that augmenting the data set by greater factors, e.g. closer to 70 as in Maxsmi, yields better results than 10 times augmentation as in CNF.

Finally, the Molecular Prediction Model Fine-Tuning (MolPMoFiT) [257] study builds an RNN model based on LSTM layers using SMILES augmentation with duplication. The lipophilicity data is augmented 25 times whereas the FreeSolv data 50 times. MolPMoFiT is trained and evaluated using 10 splits of ratio 80 : 10 : 10 for the training, validation, and test sets. The model reaches an RMSE value (and standard deviation) of 1.197  $\pm$  0.127 on the FreeSolv data and 0.565  $\pm$  0.037 on the lipophilicity data, see [257, Figure 3, 4]. While Maxsmi leads on the FreeSolv prediction problem, MolPMoFiT slightly outperforms Maxsmi on the lipophilicity data (Table 3.4).

Lastly, study comparison should be treated with utmost attention, since results can not be compared blindly. For instance, if the data preprocessing is done differently in each study, or the splits are not identical, or the parameters of the experiments are not set to be the same, then the results are not fairly comparable.

#### 3.4.4 Test case: EGFR affinity data following the guideline

Given the results of the Maxsmi models on the physico-chemical data sets, we now discuss guidelines to apply SMILES augmentation on a new data set. The EGFR affinity data, discussed in the Provenance section and henceforth simply referred to as affinity data, is used as a test case, but the idea can be applied to different data sets and broader use cases.

Table 3.5: **The Maxsmi models strike again!** The Maxsmi model developed for the affinity against the EGFR kinase and the Random Forest (RF) baseline model outperform the canonical model.

Name	Model	Augmentation number	Augmentation strategy	Test RMSE	Test R2
Maxsmi	CONV1D	70	Augmentation with reduced duplication	0.777	0.712
Canonical	CONV1D	0	No augmentation	1.031	0.494
Baseline	RF	0	No augmentation	0.758	0.726

Since the affinity data set contains 5,849 data points after preprocessing (see Table 3.1), the lipophilicity and affinity data are of a similar order of magnitude, although the latter is somewhat larger. Therefore, a compromise between the size of the data set and the tuple that gives the best results for the FreeSolv, ESOL, and lipophilicity data (see Table 3.2) is found: for the affinity data, the CONV1D model is chosen (similarly to lipophilicity, ESOL, and FreeSolv), the number of augmentation is set to 70, as for ESOL and FreeSolv, and the augmentation strategy is set to augmentation with reduced duplication for a less computational intensive training than augmentation with duplication. As comparison, the Maxsmi, the canonical, and the baseline models on affinity are trained and evaluated. The same experimental setup for splitting and evaluation as mentioned in the Data and experimental setup section is applied. On the test set, the canonical model reaches an RMSE value and coefficient of correlation R2 value of 1.031 and 0.494, respectively. In comparison, the Maxsmi model shows great improvement with test RMSE, R2 values of 0.777 and 0.712, respectively (see Table 3.5). Surprisingly, the RF baseline model performs similarly to the Maxsmi model, with an RMSE of 0.758 and an R2 of 0.726.

### 3.4.5 Maxsmi models available for user predictions

Given the good performance of the Maxsmi models for all three physico-chemical tasks and for EGFR affinity, we retrained them on all points in the data set as a final product. The aim is to offer a single command-line interface for prediction. A user can provide a SMILES as input, choose a given task and they will receive an output file in the form of a CSV table with relevant information, such as 1. the user input SMILES itself, 2. whether the compound was in the training set, 3. the canonical SMILES, and its associated variations 4. the per SMILES predictions, 5. the per compound prediction, 6. and the standard deviation. A PNG file of the 2D molecular graph associated with the input SMILES is also generated. For example, for the semaxanib drug, taken from the PKIDB database [125], and given by the SMILES O=C2C(\1cccc1N2)=C/c3c(cc([nH]3)C)C, lipophilicity is

predicted using the command-line

```
$ python maxsmi/prediction_unlabeled_data.py  
--task="lipophilicity"  
--smiles_prediction="O=C2C(\1cccc1N2)=C/c3c(cc([nH]3)C)C"
```

The command above was used to generate the values in Figure 3.1.

## 3.5 Conclusion

In this study, SMILES augmentation applied to deep learning molecular property and activity prediction is investigated. Five augmentation strategies that can be applied as SMILES augmentation are explored, together with three neural network architectures, and the performance thoroughly assessed on three molecular data sets: ESOL, FreeSolv, and lipophilicity.

Our findings show that augmentation improves the performance of deep learning models not only independently of the model, but also with respect to the size of the data set. This suggests that the choice of augmentation strategy can be viewed as hyper-parameter tuning.

The tuple consisting of (model, augmentation number, augmentation strategy) that maximizes the performance on the test set leads to the definition of the Maxsmi models. Our findings also show that the model using canonical SMILES outperforms the one using single random SMILES, thanks to the simplicity of the canonical notation.

Additionally, the Maxsmi models outperform, or perform at least as well as state-of-the-art models such as MoleculeNet, CNF, and MolPMoFiT, on the three physico-chemical data sets. This suggests that applying simple SMILES augmentation techniques can reach similar or even better performance as sophisticated models such as graph-based neural networks, as in the case of MoleculeNet. Moreover, we use our findings to guide the application of SMILES augmentation on a new data set and provide a test case with data on affinity against the EGFR kinase. Finally, we provide an easy to use framework for out-of-sample prediction on four tasks: ESOL, FreeSolv, lipophilicity, and affinity against EGFR, which should be helpful to assess properties of novel compounds. The open-source code allows to perform similar studies on different data sets with minor programmatic adjustments.

As an outlook, we observe that strategies that keep all, or a fraction of duplicates, may help the model to learn inherent symmetry in a compound. Indeed the same random SMILES representation will certainly be generated multiple times for a symmetric molecule even though the initial atom and the path along the graph are different. In this sense, SMILES duplication is not an artificial construction, and keeping replicas could retain important information about the underlying symmetry of a compound.



## Chapter 4

# Interpreting model prediction for cytotoxicity

The contents of this chapter were published as Webel, H. E.\* , Kimber, T. B.\* , Radetzki, S., Neuenschwander, M., Nazaré, M., & Volkamer, A. (2020). Revealing Cytotoxic Substructures in Molecules Using Deep Learning. *Journal of computer-aided molecular design*, 34 (7), 731-746 [3], under the Creative Commons Attribution (CC BY) license, <https://creativecommons.org/licenses/by/4.0/>. The content from this publication is presented here with the permission of Springer publishing.

Contributions:

AV conceived the project. TBK laid out part of the theory, and methodology, analyzed the results of the experiments. The manuscript was written by TBK, HEW, MN, AV.

### Chapter summary

In drug development, late stage toxicity issues of a compound are the main cause of failure in clinical trials. *In silico* methods are therefore of high importance to guide the early design process to reduce time, costs and animal testing. Technical advances and the ever growing amount of available toxicity data enabled machine learning, especially neural networks, to impact the field of predictive toxicology.

In this chapter, cytotoxicity prediction, one of the earliest handles in drug discovery, is investigated using a deep learning approach trained on a highly consistent in-house data set of over 34,000 compounds with a share of less than 5% of cytotoxic molecules. The model reached a balanced accuracy of

---

\*These authors have shared first authorship.

over 70%, similar to previously reported studies using Random Forest. Albeit yielding good results, neural networks are often described as a black box lacking deeper mechanistic understanding of the underlying model. To overcome this absence of interpretability, a Deep Taylor Decomposition method is investigated to identify substructures that may be responsible for the cytotoxic effects, the so-called toxicophores. Furthermore, this study introduces cytotoxicity maps which provide a visual structural interpretation of the relevance of these substructures.

Using this approach could be helpful in drug development to predict the potential toxicity of a compound as well as to generate new insights into the toxic mechanism. Moreover, it could also help to de-risk and optimize compounds.

## 4.1 Introduction

Over the past two decades, an increasing number of new chemicals have been synthesized every year [288] and fast prior analysis of their potentially toxic effects on humans and animals has become crucial [289]. In drug development, late stage safety and toxicity issues are still the main causes of failure in clinical trials [290, 291]. Moreover many animals (ca. 2.8 Mio, BMEL [292]) are deployed for testing in research and development. Therefore, *in silico* methods are highly valuable during early drug development to reduce costs, human discomfort and animal testing [293] and might contribute to the early identification of harmful substances according to the REACH regulation [294]. *Machine learning (ML)* algorithms, more specifically deep learning methods, have proven to perform well in different fields, such as speech recognition [295] or image classification [296], and are now also broadly used in drug design [297–301]. A recent review of deep learning in chemistry can be found in [302]. ML-based endpoint prediction in computational chemistry follows the principle that compounds with similar substructures or features may cause similar effects. Given a labeled data set with known outcome, the ML algorithm learns to identify the often highly non-linear combination of physico-chemical and structural features in the compound, commonly encoded by circular fingerprints (e.g. Morgan/ECFP), that may be responsible for their (toxic) effect [303–306]. Such models can be built for target-specific endpoints (binding assays) as well as for more complex biological endpoints (cell-based assays), such as cytotoxicity. While more data might be available for the former group, the models might be less relevant for *in vivo* situations [307].

Cellular *cytotoxicity* is a high-level property of molecules as it can be caused by different mechanisms. It refers to cell-death by cell membrane damage and necrotic lysis or cell processes such as apoptosis, autophagy or regulated necrosis [308]. Cytotoxicity is experimentally assessed by counting



survival rates after treating a cell line with a given substance [309]. In pharmaceutical drug discovery, cytotoxicity is one of the earliest handles for assessing toxicity of a drug. Discarding compounds with undesired features early in the development stage is of high practical value, following the "fail early - fail cheap" derisking principle.

Some *computational cytotoxicity* models have already been published, most of them applying random forest algorithms [308, 310, 311], others using Bayesian methods with physico-chemical properties and/or circular fingerprints as descriptors [312]. Additionally, a naive Bayes approach in combination with activity spectra has been introduced for cytotoxicity prediction [313]. Furthermore, previous studies have shown the success of *Feedforward Neural Networks (FNN)* [314, Chapter 6] especially in predicting different toxic endpoints [56, 315]. The ability of such networks to model and learn non-linear, complex relationships have gained more and more attention in the context of chemistry [316]. While showing promising results, two major challenges remain for such methods in drug design.

The first challenge is the availability of sufficient and reliable data [317]. Many models are trained on scattered publicly available - and thus, heterogeneous data - due to assay diversity, as well as highly variable conditions and setups used throughout different laboratories. Therefore, thorough data curation is crucial [318]. Second, ML algorithms and especially Deep Neural Networks (DNN) may act as a black box and one is often unable to understand the intricacies in the hidden layers. The deeper the network the more complicated the interpretation becomes. Over the last years, several techniques to interpret such models have been introduced in the broader context of drug discovery [319–322], including but not limited to atom-level coloration [320], integrated gradients [321], attention-vector based relevant latent features exploration [322], masking and gradient techniques applied to 3D convolutional neural networks [323] and partial derivative-based methods [324].

To overcome these hurdles, a DNN model is trained in this study using a highly consistent data set from the Leibniz Associations Research Institute for Molecular Pharmacology (FMP: Leibniz-Forschungsinstitut für Molekulare Pharmakologie), with approximately 34,000 compounds (remaining standardized compounds after data preprocessing) measured for their cytotoxic potential. The effect on cell viability, including sublethal effects on cell proliferation, was measured using a high-content screening assay. This assay enables to visualize and quantify phenotypic changes due to compound treatment. Furthermore, a new technique is used here to unleash the black box effect by identifying relevant features for toxicity prediction. One recent approach, known as the layer-wise relevance propagation (LRP), decomposes the output scores layer by layer back to the original inputs of the network, yielding information on which features are important for the prediction. One special case of the LRP method, called *Deep Taylor Decomposition* (DTD)

developed by Montavon et al. [79], uses the Taylor decomposition to redistribute the output score. This study is the first, to the best of our knowledge, that uses the DTD in the molecular context. In order to obtain a visual representation of the atom environments potentially relevant for cytotoxicity determined by the DTD method, a technique developed by Riniker and Landrum [325], called similarity maps, is employed to depict the 2D plots of the molecules where the relevances of the potentially cytotoxic substructures are highlighted. The application of similarity maps in the context of cytotoxicity prediction will further be referred to as cytotoxicity maps. With this approach, potential cytotoxic compounds could be identified and prioritized for experimental testing and verification.

## 4.2 Data & Methods

This section describes the data set and the preprocessing steps, as well as the machine learning models that are used for this study. Furthermore, the Deep Taylor Decomposition to identify potential toxicophores and the visualization using cytotoxicity maps are introduced.

### 4.2.1 Data

**Data Collection and Cytotoxicity Definition** The compound library available at the FMP comprises a collection of 74,000 chemically distinct substances that were assembled at the FMP [326]. Among them, more than 34,000 compounds were purchased from commercial vendors. These commercial compounds were selected after an analysis of the World Drug Index (database of 70,000 approved drugs and natural products annotated for bioactivity) for privileged substructures frequently occurring in different drugs. According to the approximately 561 identified main chemotypes, which represent a major part of the currently known chemical space of drug-like molecules, compounds presenting these privileged motifs in different combinations and variations were selected. Prior incorporation into the library, a filtering against known reactive groups (similar to filtering against pan-assay interference compounds [327]) was performed as described in Lisurek et al. [326].

The initial data set from the FMP available for this study contained 34,848 compounds that were tested for their cytotoxic effects on two cell lines, HepG2 and HEK293, as well as another 1,408 compounds that were tested only on the HepG2 cell line. Cells were seeded onto 384-well plates, compounds added to a concentration of 10  $\mu\text{M}$ , and cells incubated for additional 72 hours. Resulting cell numbers were then determined by staining of the nuclei using Hoechst 33,342 technique\* [328] and counting the nuclei

---

\*Hoechst 33,342 is a cell-permeable minor groove-binding DNA stain, which starts to

with fluorescence microscopy. In order to increase reliability, three technical replicates (replicating the steps of cell seeding, compound addition and cell counting) were generated. The high concentration justifies two assumptions: first, the permeability of molecules does not need to be taken into account as the high concentration likely leads to cell membrane penetration and relevant intracellular concentrations. Second, the high concentration should also reliably reveal existing toxicity of the compounds.

Cytotoxicity of a molecule is defined using the relative growth inhibition measurement comparing two samples of a cell line, untreated and treated, respectively. A molecule is labeled cytotoxic if it inhibits growth by at least 50% compared to the untreated samples and the cell count should be three standard deviations lower than the median of the cell lines on a specific plate. This effect had to be observed in at least two of the three technical replicates.

In case a compound is toxic at the same concentration range as applied for the measurements ( $10\mu\text{M}$ ), small differences in sensitivity between the different cell lines may lead to a compound being determined toxic in one cell line but not in the other. Thus for this study, a compound is considered cytotoxic if it is measured cytotoxic on at least one of the two cell lines (HEK293 or HepG2).

**Compound Data Preprocessing** All molecules are processed with RDKit [131], of which 157 are discarded due to sanitization issues. After sanitization, the remaining molecules are preprocessed by applying certain structure standardization rules, e.g. removing salts, normalizing charges and handling tautomers, using the tool developed in the scope of IMI eTox [329]. Subsequently, duplicates produced by the standardization process are removed. This results in 34,366 compounds that are considered in this study. Only 4.65% of the molecules in the preprocessed data set are labeled cytotoxic, leading to highly imbalanced data (see Figure 4.1).

**Compound Encoding** All molecules in the preprocessed data set are transformed into Morgan fingerprints using RDKit [131]. Atom environments are only considered at an exact radius of two bonds and the length of the fingerprint is set to 2,048. Environments are only included if they appear at least five times in the data set, yielding 14,245 unique hash keys. This selection omits 40,507 substructures as they were present less than five times in the data set. This feature selection is equivalent to the first step of Gütlein and Kramer [330, Tab. 6]. Note that due to the hashing of the features to a 2,048 bit fingerprint, different atom environments may be mapped to the same bit, known as bit collision.

---

fluoresce bright-blue upon DNA binding. Stained nuclei are then easily distinguishable from background using fluorescence excitation in the UV range.

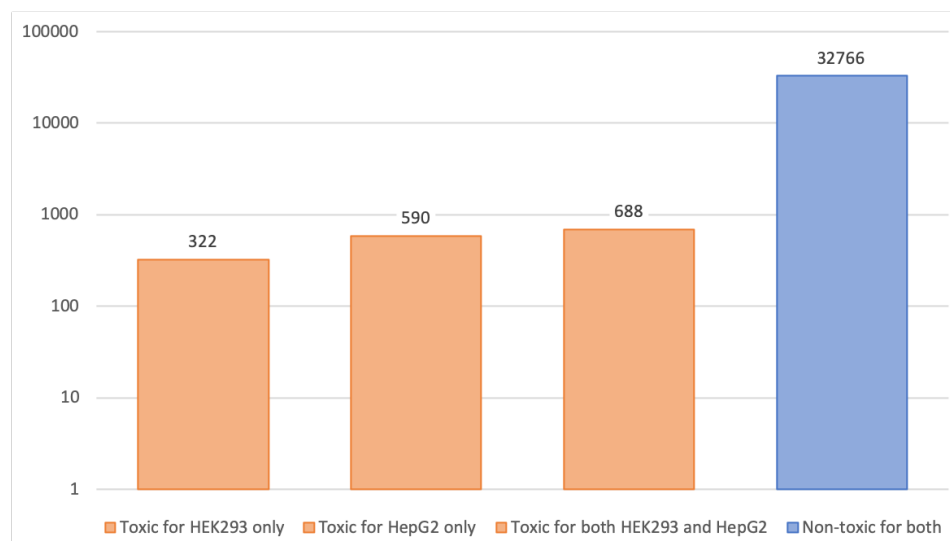


Figure 4.1: The logarithmic scale plot shows the number of toxic and non-toxic molecules for the two cell lines HEK293 and HepG2. There are approximately 20 times more molecules that are labeled non-toxic than toxic, making the data set highly imbalanced.

#### 4.2.2 Machine Learning Model Generation

**FNN Model Setup** In this study, a feedforward fully-connected neural network (FNN) is used to predict cytotoxicity of compounds, a model similar to Mayr et al. [319] in the Tox21 challenge. The inputs are given by the 2,048 long fingerprints and the outputs are binary variables indicating if a molecule is cytotoxic or not. The architecture of the model considers three dense hidden layers with respectively 512, 192 and 128 units. The activation function used in the hidden layers of the network is the ReLU function, defined as  $\text{ReLU}(x) = \max\{x, 0\}$  [314, p.170]. For the final classification, a sigmoid function, defined as  $\sigma(x) = \frac{1}{1+e^{-x}}$ , is applied to obtain prediction values that range between 0 and 1. These values correspond to the probability of belonging to either the cytotoxic or the non-cytotoxic class. To avoid overfitting, the output layer is regularized using dropout [331], where 40% of hidden units in the last hidden layer are set to zero at random during each mini-batch gradient updating step. Additionally, toxic molecules are weighted five times more in the loss function than non-toxic ones in order to statistically increase their prevalence. The Adam method [332] is chosen as the network optimizer with an initial learning rate of 0.0001. The model has been established by running a random hyperparameter search (data not shown).

**RF Baseline Model Setup** To compare the results of the deep learning model, a baseline is computed using a Random Forest (RF) model. This tree-based method has shown to perform particularly well in cheminformatics [333]. The default settings in Scikit-learn [27] are used; more specifically 50 trees are fitted, each of them selecting randomly 45 out of the 2,048 bits of the fingerprint as features. The same strategy as for FNN is used to account for the imbalanced data.

**Model Validation** As a model setup, a 10-fold nested cross-validation with validation and test set is used. The preprocessed data is randomly split into 10 parts. First, one of these parts is randomly selected as test set (10% of the data set), another as validation set (10% of the data) and the remaining as training set (80% of the data). Finally, all possible combination of these three sets are considered leading to 90 model evaluations (see Table 4.1). For each combination, also called run, the FNN and the RF models as previously described are trained on the training set, using the validation set for hyperparameter tuning, and evaluated on the test set. Note that for the FNN production run and the toxicophore evaluation, a separate model with a random split into the same proportions has been setup. For model evaluation, the balanced accuracy (AccB) [334], the true positive rate (TPR) and the true negative rate (TNR) [335, Table 1] are used as comparison metrics. The formulas for these three metrics are shown in Equations 4.1, 4.2 and 4.3, where TP represents the true positive counts, TN the true negative counts, FP the false positive counts and FN the false negative counts. Note that AUC values are not included since this metric may be misleading when evaluating model performance on imbalanced data sets, as suggested by Saito and Rehmsmeier [335].

$$\text{AccB} = \frac{1}{2} \left( \frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right), \quad (4.1)$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (4.2)$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (4.3)$$

### 4.2.3 Deep Taylor Decomposition

When training a model, besides model performance, the relevance of certain features that lead to the predictions may be of high interest. For this purpose, Bach et al. [336] proposed a method to decompose layer-wise a given model score and redistribute the decomposed scores to the inputs. For a specific input  $x$ , node  $i$  and layer  $l = 0, \dots, L$ , we note  $R_i^l(x)$  the associated relevance score. The layer-wise relevance propagation has the desired property to

Table 4.1: Number of toxic and non-toxic compounds in each of the split sets: training, validation and test.

	training 80%	validation 10%	test 10%	total 100%
non-toxic compounds	26,212	3,277	3,277	32,766
toxic compounds	1,280	160	160	1,600
total compounds	27,492	3,437	3,437	34,366

redistribute the overall relevance between two layers, meaning that the sum over the relevances assigned to the inputs equals the probability of the model score. The initial relevance,  $R^L(x)$ , is given by the model score.

The relevance is back-propagated to previous layers following only positive weights. This is known as the  $z^+$  rule. Let  $w_{ij} = w_{ij}^{l,l+1}$  be the weight that connects non-zero hidden node  $x_i$  in layer  $l$  with hidden node  $x_j$  in layer  $l+1$ . Only positive weights are considered, namely  $w_{ij}^+ = \max(0, w_{ij})$ . Then the  $z^+$  rule is defined as follows

$$R_i^l = \sum_j \frac{x_i^l w_{ij}^+}{\sum_k x_k^l w_{kj}^+} R_j^{l+1} = \sum_j \frac{z_{ij}^+}{\sum_k z_{kj}^+} R_j^{l+1}. \quad (4.4)$$

The name  $z^+$  rule is derived from the definition  $z_{ij}^+ = x_i^l w_{ij}^+$ . Redistributing positive scores to the input using this rule allows to assign a positive relevance to each bit, which in this study encodes an atom environment (see Figure 4.2).

Note that this method is not applied directly to the sigmoid model score, but to its logarithm of odds,  $\log\left(\frac{\sigma(x)}{1-\sigma(x)}\right)$ , the so-called logit. Model scores with positive logits, i.e. probabilities greater than 0.5, are further referred to as *decomposable*. Moreover, the method is restricting biases in ReLU activations to be negative in order to ensure the applicability of the Taylor decomposition. For further details, please refer to the paper by Montavon et al. [79].

#### 4.2.4 Identification of Toxicophores and Visualization as Cytotoxicity Maps

To reveal the features having a high impact on the cytotoxicity classification of a molecule, the Deep Taylor Decomposition (DTD) method, as described in the previous section, is applied. Furthermore, for better interpretability, the features are mapped back to the molecular structure and are visualized using similarity maps, introducing the concept of cytotoxicity maps.

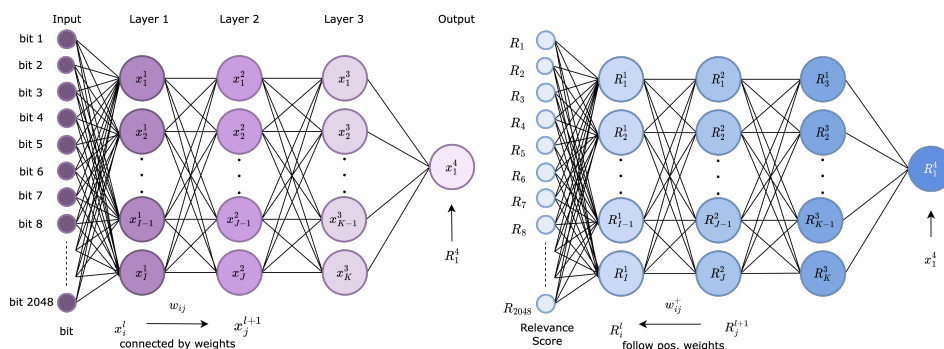


Figure 4.2: The Deep Taylor Decomposition method applied to a three hidden layer feedforward neural network. The inputs to the network are 2,048 fingerprint bits. The left diagram represents the network with ReLU activation function and the right diagram the assigned relevances using the  $z^+$  rule.  $x_i^l, R_i^l$  represent the  $i^{\text{th}}$  node, relevance at layer  $l$ , respectively.

**Detection of Potential Toxicophores** Toxicophores, in this study, are substructures in a molecule that highly contribute to the toxicity prediction. In order to identify the toxicophores in the data set, the bit-wise relevance scores, encoded by the fingerprint bits, are investigated and averaged over the complete set of molecules with decomposable scores. Such molecules will further be referred to as decomposable molecules.

For each decomposable molecule  $m \in \{1, \dots, M\}$  and for each fingerprint bit  $j \in \{1, \dots, N\}$ , a relevance score  $R_{m,j}$  is retrieved using the DTD method, see Figure 4.2. The relevance scores for each bit are aggregated by taking the mean over all atom environments setting a bit in decomposable molecules, denoted as  $N_j$ . Therefore, each atom environment  $j$  will be assigned a score  $R_j$  which was averaged on the selected data defined as the global mean relevance score

$$R_j = \frac{1}{N_j} \sum_m R_{m,j}. \quad (4.5)$$

With this approach, the  $k \in \mathbb{N}$  most likely cytotoxic substructures, or toxicophores, can be identified by selecting the  $k$  highest global mean relevance scores  $R_{(1)}, \dots, R_{(k)}$ , noting  $R_{(i)} \geq R_{(j)}, \forall i \geq j$  the ordered relevance scores. The associated workflow is illustrated in Figure 4.3. For each decomposable molecule, the subset of the identified  $k$ -most relevant toxicophores is indicated on the structure by highlighting in red all atoms that are part of the identified relevant substructure using pre-implemented plotting functions in RDKit. If a molecule contains more than one of the most likely substructures, where these cases can include disconnected, nested or overlapping substructures, the union of these substructures is displayed (i.e. each atom that is part of at least one of these environments is highlighted once).

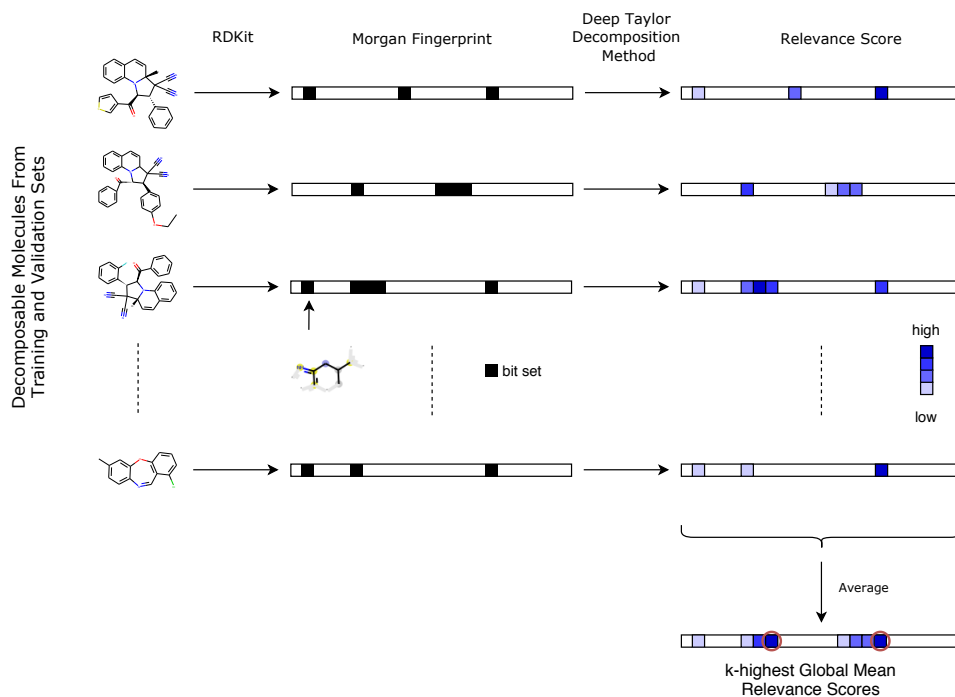


Figure 4.3: **Workflow for identifying potential toxicophores.** The first arrow describes the transformation from the molecules in the training and validation sets into 2,048 long binary vector describing the Morgan fingerprints of radius 2, using RDKit. Each bit represents one (or more) atom environment(s). The black box indicates if the corresponding atom environment is present in the molecule. The second arrow shows that relevance scores can be obtained for each compound using the Deep Taylor Decomposition method described in Section 4.2.3 and illustrated in Figure 4.2. Once all relevance scores are computed for each molecule, they are averaged using Equation 4.5. The bits corresponding to the  $k$  highest global mean relevance scores are stored and used for further analysis as potential toxicophores.



**Cytotoxicity Maps** To visualize the contribution of all atom environments contained in a molecule to the cytotoxicity prediction, similarity maps developed by Riniker and Landrum [325] are used. This technique allows to identify and visualize atom contribution from a prediction computed by a ML algorithm. In the original study, this is done as follows: Given a fingerprint of a molecule, a pre-trained ML model and a prediction value for the fingerprint, a set of weights for each atom in the molecule have to be calculated. These weights, which will define the atom contribution of the prediction, are computed in the following way: Recursively each atom is removed from the molecule and a new fingerprint is generated. The prediction of the new fingerprint is evaluated with the pre-trained ML model. Finally, the weight associated to that atom is the difference between the prediction of the fingerprint generated with and without the presence of that same atom. For visualization, bivariate Gaussian distributions centered at the atom position using these weights are generated and the plots show the superimposition of the atom positions and the contour lines of the distributions.

In this study, the weights are computed slightly differently. Indeed the weights considered are the relevance scores which are directly generated from the DTD method. Note that in contrast to the original work, the weights here can only be positive. However, as discussed in Section 4.2.3, these scores are associated to each bit in a decomposable molecule and not to each atom. Therefore, the global mean relevance score is attributed to each atom in the atom environment. Consequently each atom in the decomposable molecule is mapped to a weight and the similarity map and plots can be generated in this context. Some of the substructures might overlap and have atoms in common. In this case, the weight of an atom part of several substructures will be given the maximum value of the global relevance scores associated to the atom environments. In the cytotoxicity maps, substructures with high relevance scores will stand out and could hint to toxicophores.

#### 4.2.5 Used Software and Libraries

RDKit [131] is used for molecular encoding, fingerprint generation and plotting of molecules. Scikit-learn [27] is employed for the Random Forest model. The deep learning model is implemented using Keras with Tensorflow backend [28]. For the score decomposition, DTD implementations as provided by iNNvestigate [337] are used. The similarity maps visualization is used as in the original paper [325].

### 4.3 Results and Discussion

In the following, the results of the deep learning model as well as the baseline model are discussed and then compared to other studies on *in silico* cytotoxicity predictions. Additionally the toxicophores identified using the DTD

method and the cytotoxicity maps are presented.

### 4.3.1 Model Evaluation and Comparison

In this study, an FNN model for cytotoxicity prediction is established based on the final set of 34,366 preprocessed compounds provided by the FMP, which were tested for their cytotoxic effect on two cell lines. Out of these compounds, 32,353 are commercial compounds selected using the strategy described by Lisurek et al. [326], another 2,013 are commercial compounds with known biological activity ("LOPAC®1280" library from Sigma-Aldrich [338]) and FDA-approved drugs ("FDA Approved Drug Library L1300" from Selleckchem [339]). The data can be considered as highly consistent and curated, since it has been produced in the same laboratory using the same cell line and experimental setup with several reference compounds as control for each assay campaign. Note that the data set is highly imbalanced with a share of only 4.65% of toxic molecules.

**FNN vs. RF Cross-Validation Results** First, the results of the nested cross-validation (CV) of the FNN model are compared to the baseline RF model. Overall both the FNN and the RF models perform similarly well regarding balanced accuracy on the given data set. On the training set, RF seems to highly overfit the data (see Train row in Table 4.2), meaning that the model would tend to memorize patterns instead of learning them. On the test set, the FNN and RF models yield similar results with a mean balanced accuracy of approximately 68%, with a slightly higher mean and narrower standard deviation for the FNN setup (see Table 4.2). This is a fair increase in performance when comparing these results to the 50% AccB of a naive classifier, which would always predict all compounds to the majority class (non-toxic in this study). Furthermore, the FNN tends to produce more balanced TPR and TNR results compared to RF: a mean of 61.57% TPR and 76.22% TNR for the FNN opposed to 51.48% TPR and 85.02% TNR for RF. This observation is especially important when the task requires identifying potentially cytotoxic molecules in a highly imbalanced data set. Note that AccB, TPR and TNR are based on an automatically set cutoff yielding the maximum balanced accuracy on the respective validation split (0.17 for FNN and 0.06 for RF). The cutoff adaption is necessary because of the highly imbalanced nature of the underlying data set. This strategy is preferred over under-sampling in order to use as many data points as possible (see [340]).

**Comparison to Other Studies** Next, the CV results of the FNN and RF models trained on the FMP data are discussed in the context of three other recently presented models for cytotoxicity prediction [308, 310, 311], mainly using random forest models on freely available data (see Table 4.3). Note

Table 4.2: 10-fold nested cross-validation results (mean and standard deviation) for the FNN and RF baseline models. Reported performance measures in % are balanced accuracy (AccB), true positive rate (TPR) and true negative rate (TNR). The best results on the test set are displayed in bold.

		FNN			Random Forest		
%		AccB	TPR	TNR	AccB	TPR	TNR
Train	mean	84.28	90.66	77.90	97.85	100.00	95.69
	std	2.22	4.03	6.64	1.26	0.00	2.52
Val	mean	70.13	63.94	76.32	68.72	52.35	85.09
	std	1.30	6.92	6.82	1.71	6.96	5.70
Test	mean	<b>68.89</b>	<b>61.57</b>	76.22	68.25	51.48	<b>85.02</b>
	std	1.46	7.39	6.62	1.96	1.82	5.94

that results are only partly comparable between different studies since both data sets and methods may vary. Even in the case of same data, different splits can make comparison of methods difficult, as mentioned by Wu et al. [316].

Mervin et al. [308] trained a random forest model on publicly available NCBI BioAssay data, standardized using an in-house script. Molecules are considered cytotoxic if they have a  $pIC_{50}$  above 5.0 in the tested assay. Undersampling from millions of non-toxic molecules, the final public training data set contains a total of 14,880 molecules of which 3,720 are labeled cytotoxic. With 25%, the share of toxic molecules is higher than in this study, but a similar weighting approach is used to balance the training data statistically. The external test data set consists of 988 molecules with an even higher share of 45% cytotoxic molecules [308, Table 8] and the model exhibits a balanced accuracy of 76.69%. Svensson et al. [311] trained a random forest model on extracted and standardized [329] molecules from PubChem, which were tested on a variety of cell lines and the cytotoxicity definition varied from one data set to the other. Their external data set consisted of 3,295 molecules of which only 48 were labeled cytotoxic. Having a share of less than 1.5% is below the share of this study. Furthermore, they use conformal prediction models based on RF classifiers. The conformal prediction balanced accuracy of their model is 69.15%. However conformal prediction metrics do not necessarily translate to performance measured by metrics on pure model predictions. Banerjee et al. [310] report the highest balanced accuracy of 83.60% on their test data split. They extracted data from ChEMBL [54] and used cytotoxicity based on  $IC_{50}$  values at a concentration cutoff of  $10\mu\text{M}$ . The random forest classifier is trained on 5,487 samples and evaluated on a test set of 610 samples, each containing one third of cytotoxic molecules [310, Table S1]. In the presented study, approximately seven times

Table 4.3: Comparison of FNN and RF performance of this study with other existing models for cytotoxicity prediction (reported are mean CV results, noting that CV setup differ between methods). Balanced accuracy (AccB.), true positive rate (TPR) and true negative rate (TNR) are presented in %. The last column describes the size of the test data, as well as the number and share of cytotoxic compounds. The best results are displayed in bold.

Models	AccB.	TPR	TNR	Test Set		
				total	toxic count	percent
FNN (this work)	68.89	61.57	76.22	3,437	160	4.6
RF (this work)	68.25	51.48	85.02			
RF, Mervin [308, Tab. 8, public]	76.69	56.90	<b>96.50</b>	988	445	45.0
CP/RF, Svensson [311, Tab. 5]	(69.15)	(73.80)	(64.50)	3,295	48	1.5
RF, Banerjee [310, Tab. 2]	<b>83.60</b>	<b>93.00</b>	74.00	610	205	33.6

less toxic molecules were in the data set.

To conclude, Table 4.3 seems to suggest that models with more balanced data sets lead to better performance, as is illustrated with a 83.60% balanced accuracy from Banerjee et al. [310] and 76.69% from Mervin et al. [308]. However, as stated above, first, comparisons between the models should be made with care. Second, while having more balanced data sets may facilitate the modeling task, the question remains which resembles better the real live scenario. The results of the models trained on highly imbalanced data sets lie in the same range as shown with the FMP data and FNN as well as RF with a balanced accuracy of approximately 69% from this study and the RF-based CP model from Svensson et al. [311]. While Mervin et al. [308] obtain a TNR of 96.50%, the TPR is only 56.90%. In the FNN model used in this study, the TPR and TNR are more balanced, with a TNR of 76.22% and a TPR as high as 61.57%. This result may be more valuable in this context, since the main goal is to identify cytotoxic molecules. From an application point of view, correctly predicting cytotoxicity for novel molecules that would indeed later show toxic behavior (in *in vitro* or *in vivo* studies) may be more crucial, because these compounds could be excluded from further development.

**FNN Production Run Results** After successful CV evaluation of the FNN model and comparison to a baseline RF as well as other published studies, a FNN was built for production run, showing a balanced accuracy of 70.73% on the test set. This model is used for the DTD in order to identify and highlight toxicophores in molecular structures.

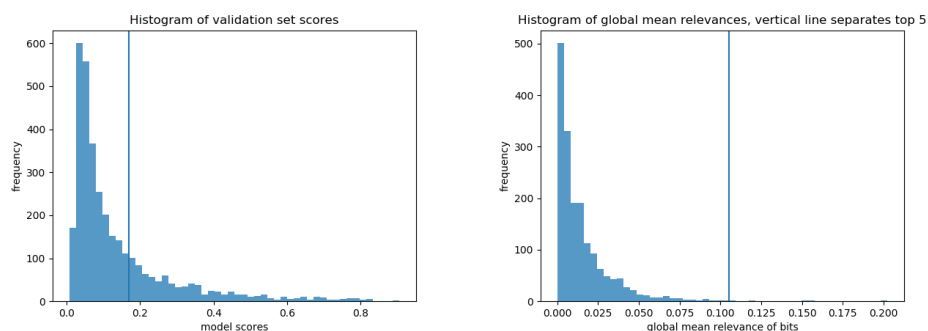
Table 4.4: Model metrics in % at 0.17 cutoff yielding maximum balanced accuracy on the validation set (in bold) as well as another cutoff at 0.20 yielding higher TNR rates on the validation set (in bold).

	cutoff=0.17			cutoff=0.20		
	AccB	TPR	TNR	AccB	TPR	TNR
Train	85.76	92.50	79.02	86.43	89.53	83.32
Val	<b>69.46</b>	62.50	76.41	67.19	53.75	<b>80.62</b>
Test	70.73	63.12	78.33	69.53	56.88	82.18

The cutoff value which yields the maximum balanced accuracy (69.46%) on the validation data is 0.17 (see Figure 4.4a for the distribution of model scores corresponding to that specific cutoff). The TPR and TNR associated to that cutoff on the validation set are 62.50% and 76.41% respectively. Note that since the TPR and the TNR are directly related to a chosen cutoff, varying this cutoff value would immediately result in the change of these rates. Aiming towards a higher TPR or a higher TNR may depend on the research question at hand and the cutoff should be chosen accordingly. A cutoff of 0.20 would for example yield on the validation set a lower TPR of 53.75% but a higher TNR of 80.62% (see Table 4.4), and the same trend can be observed on the test set. Since the aim of this study is to reveal potential cytotoxic compounds which could then undergo further (experimental) testing, reaching a higher TPR is of more importance.

### 4.3.2 Potential Toxicophores

The current study aims to provide a visual structural interpretation of the model outcomes with the aim of identifying novel toxicophores. From the 30,929 molecules that are present in the training and validation set, a total of 1,210 molecules are decomposable ( $\sim 4\%$ ), which is in line with the share of cytotoxic molecules in the complete data set. As discussed in Section 4.2.4, relevance scores are obtained for each of the 2,048 atom environments from these decomposable molecules. The workflow in Figure 4.3 describes the process of going from decomposable molecules to global mean relevance scores per bit. Atom environments referring to high scoring bits generally contribute greatly to the predicted toxic value of the compound and thus represent potential toxicophores.



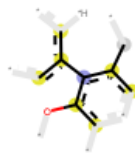
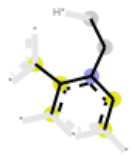

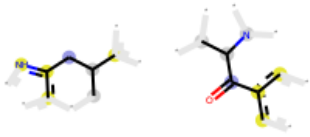
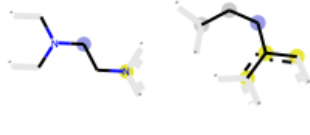
(a) Histogram of validation set scores. (b) Histogram of global mean relevances.

Figure 4.4: (a) Distribution of predicted scores for molecules from the validation set, which was used to calibrate the cutoff 0.17 (indicated by the vertical line) of the model to classify compounds as cytotoxic. (b) Distribution of global mean relevances of set bits in decomposable compounds in the training and validation set, which were used to determine the five most important bits (indicated by the vertical line).

**Identification of Potential Toxicophores Based on Most Important Bits** Note that for the analysis of the most important bits, global mean relevance scores were calculated per bit. These scores range from 0.0 to 0.2, and the distribution shows a drastic drop in values indicating that only few bits have a high impact (see Figure 4.4b). In the following, the  $k = 5$  bits with the highest scores are selected for further analysis. Note that with increasing values of  $k$ , more often several of these bits appear together in one molecule and overlap. Thus, the portion of the molecule that is covered by these bits, which likely contribute to cytotoxicity, becomes larger and closer to a full scaffold. In this case study, selecting the five highest relevance scores seems appropriate to reveal meaningful substructures. Table 4.5 displays these bits in decreasing order with respect to the global mean relevances as well as the predictions (TP and TN counts) given by the FNN model. On the training and validation set, the molecules that contain at least one of these bits are correctly predicted cytotoxic by the model 85% of the time. If the counts from bit 85 are removed, this number increases to over 90%. Similar findings can be assessed on the test set: the model yields 69% and 75% correctly predicted values, including and excluding bit 85 respectively. This observation indicates two facts: First, the results of the DTD method are meaningful and useful in assessing the cytotoxicity of compounds; novel molecules containing these bits should be treated with special attention in future laboratory experiments. Second, bit 85 seems to be an outlier which will be discussed later in greater details.

In the test set, 17 molecules contain at least one of these top five atom

Table 4.5: Bits with the five highest global mean relevance scores (rel. score) are shown in decreasing order, as well as the predictions (TP and TN counts) given by the FNN model on both the training and validation sets (Train+Val) and on the test set for molecules that contain these bits. The last column shows the 2D image of atom environments associated to the Morgan fingerprint bit in the test set (two images to exemplify bit collisions), where the blue, yellow and gray circles represent central, aromatic and aliphatic ring atoms, respectively.

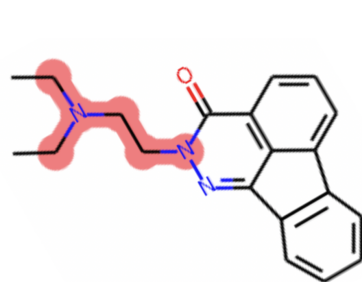
Bit	Mean rel. score	TP - FP Train+Val	TP - FP Test	Atom environment(s) associated to bit
904	0.20	50 - 2	4 - 1	
812	0.16	54 - 7	4 - 2	
1,316	0.15	57 - 6	5 - 2	
85	0.12	39 - 24	5 - 4	
713	0.11	52 - 5	5 - 1	

environments (see Figure B.9, bits highlighted in red). For example, test molecule 1, an indenophtalazinone derivative, was correctly labeled cytotoxic by the FNN model and contains bit 713 (see Figure 4.5a). To verify this prediction, the eMolTox tool developed by Ji et al. [341], an *in silico* drug safety analysis system, was queried. The authors constructed Mondrian conformal prediction models for 174 toxicology-related *in vitro* and *in vivo* experimental data sets. eMolTox predicts the compound with high confidence as potentially being genotoxic, interacting with the CNS, and/or with the liver. Most interesting are two similar compounds that exist in the underlying database which were tested active in the context of genotoxicity (i.e. the drug flurazepam, ChEMBL968 in the ChEMBL database [54]) and liver damage (amonafile, Phase III, ChEMBL428676). While the unrelated scaffold systems of these active molecules, such as the benzodiazepine scaffold from flurazepam differ from the compound in this study, they also contain the tertiary substituted ethylenediamine corresponding to bit 713 in molecule 1. Moreover, eMolTox offers the detection and highlighting of toxic substructures in each query molecule, based on a list of structural alerts collected from literature (see Table S2 in Ji et al. [341]). For the query molecule, several structural alerts are identified. Among them, the tertiary amine is highlighted being potentially involved in covalent DNA binding. The toxicophore identified here seems to contain but extend the known structural alert to a larger moiety that is potentially involved in cytotoxicity. Figure 4.5b illustrates the cytotoxicity map for the considered molecule. The atom environment associated to bit 713 stands out compared to the other substructures in the molecule and therefore may be designated as a toxicophore. Furthermore, the right part of the fused ring system also shows some intensity (relevance) and actually describes a part of the molecule that was also highlighted by eMolTox’s structural alerts and annotated as potentially kidney toxic or hepatotoxic.

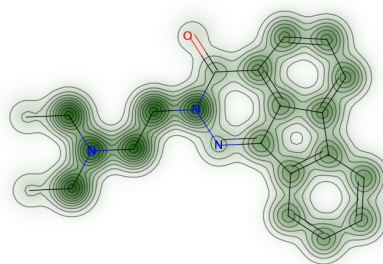
Additionally, in five molecules of the test set (2A-2E in Figure 4.6, see also Figure B.9) four of the five most relevant bits (namely bits 713, 812, 904, 1,316) appear together and form a potential toxicophore which covers a larger 6,7-dihydrobenzo[a]heptalen-9(5H)-one core structure including methoxy and amino substituents. This combined substructure is present in five compounds from the test set of which four are indeed experimentally labeled cytotoxic (molecules 2A to 2E in Figure 4.6, left) and the FNN predicts them as toxic with a high mean probability of 0.89 (see Table C.1). This assumption is supported by the cytotoxicity map exemplified for test molecule 2B (see Figure 4.5c).

Using the eMolTox tool, a toxicity prediction for the visually determined maximum common substructure of these five compounds was performed (see Figure 4.6). The most similar active compound in the eMolTox data set to the queried common core is the known drug demecolcine (ChEMBL312862), a colchicine derivative, which is used in chemotherapy and shows cytotoxic

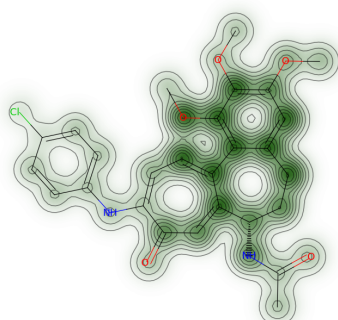




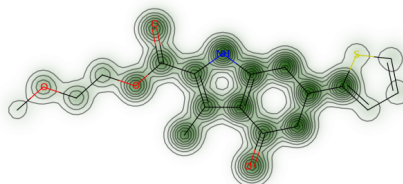
(a) Molecule 1 with bit 713 highlighted.



(b) Cytotoxicity map: molecule 1.



(c) Cytotoxicity map: molecule 2B.



(d) Cytotoxicity map: molecule 3A.

Figure 4.5: The figure shows three compounds from the test set, namely molecule 1, molecule 2B and molecule 3A, that were correctly labeled cytotoxic by the FNN model. Figure 4.5a highlights bit 713 in red in molecule 1. Figures 4.5b, 4.5c & 4.5d illustrate the cytotoxicity maps for these molecules. The atomic weights are computed using the approach discussed in Section 4.2.4. The higher the value of the respective global mean relevance, the darker the green coloring.

activity. In accordance with being predicted cytotoxic in this study, the queried common substructure is predicted by eMolTox to further cause DNA damage, genotoxicity, as well as interacting with the liver and endocrine system (see Figure 4.6, right). Furthermore, eMolTox identified the following toxic alerts: covalent binding to proteins or DNA (because of potential electrophilic reactivity), as well as skin sensitization and/or hepatotoxicity (the latter two caused by catechol or catecholdimethyl ethers or p-alkoxy aromatic ethers). The identified 4-bit substructure in this study extends the alerts and suggests a larger substructural entity, namely the 6,7-dihydrobenzo[a]heptalen-9(5H)-one core structure bearing methoxy and amino substituents, being involved in cytotoxicity (see Figure 4.6).

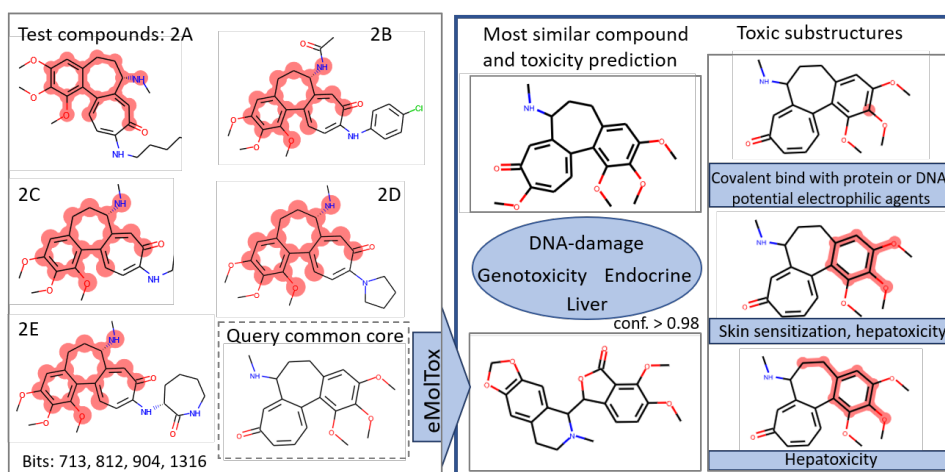


Figure 4.6: Schematic description of analysis: On the left, molecules 2A-2E from the test set are shown together with the relevant bits highlighted in red. The common core of these five molecules is used as query for the eMolTox server and the results of eMolTox are summarized on the right, with predicted toxic endpoints in blue.

As described above, bit 85 was identified as one of the five bits with the highest global mean relevance for cytotoxicity and thus, a potential toxicophore. Surprisingly, in the training and validation set, only 39 out of the 63 decomposable molecules containing this bit were experimentally tested as cytotoxic (61.9%). In contrast, high precision ( $TP/(TP+FP)$ ) ranging between 88.5% and 96.2% were achieved for the decomposable molecules containing one of the other four bits (see Table 4.5). Also, 4 out of 9 decomposable molecules in the test set containing bit 85 are falsely predicted as toxic. Therefore, bit 85 was further analysed uncovering two interesting aspects: First, five different atom environments are mapped to bit 85, of which the two most common ones (72% and 10%, named bit 85\_t1 and 85\_t2 in the following) are depicted in Table 4.5 and are present in molecules 3A to

3G and in molecule 4 of the test set, respectively (Figure B.9). This behavior is known as bit collision when working with folded molecular fingerprints, as mentioned in Section 4.2.1. Folding is a compromise between accuracy and performance since unfolded fingerprints can become enormously long. In this study, the unfolded fingerprints could already be reduced to a size of 14,245 bits by introducing a filtering step, but are afterwards folded to 2,048 bits, as described in Section 4.2.1. Considering the 63 decomposable molecules containing an atom environment that is mapped to bit 85, 52 cases represent type 85\_t1, the remaining 11 type 85\_t2 (see Table 4.5). All molecules from the latter group were indeed experimentally tested toxic (similar to molecule 4). In contrast, almost half of the 52 molecules of the former group (similar to molecules 3A to 3G) were experimentally tested non-toxic (FPs). This indicates that the model could be improved by reducing such bit overlap. Note that these collisions seem to be less problematic in the case of bit 713. Most of the decomposable molecules in the training set which contain bit 713, with different associated atom environments (as shown in Table 4.5), do indeed belong to the toxic class. Second, the low precision for compounds containing bit 85 points to the fact that this class of molecules might be challenging for the algorithm. While having a common 1,5,6,7-Tetrahydro-4H-indol-4-one core, the toxicity of the compounds seems to depend on the peripheral substitution and the functionalization. This points to the concept of activity cliffs, which are a challenge for many predictive modeling approaches [342]. While the FNN generates many FPs for the decomposable molecules of this compound class, the algorithm nevertheless predicts the TPs (3A in Figure 4.5d, 3C, 3D and 3G) with higher mean probability than the FPs (3B, 3E, 3F and 3H), 0.77 vs. 0.64, respectively (see Table C.1).

Note that molecule 5 (which contains bit 1,316) and molecule 6 (which contains bit 812) are wrongly predicted as cytotoxic by the FNN. The most relevant bits they contain refer to bit collision and are different from the major bit types shown in Table 4.5. Furthermore, the predicted scores are slightly lower than for the TPs mentioned above, i.e. 0.59 for molecule 5 and 0.69 for molecule 6 (see Table C.1).

These observations highlight the value of the DTD method during model setup and evaluation. Using the features learned by the algorithm and mapping the scores back to the structure, shortcoming of the model can be pinpointed and actions could be taken such as enlarging the fingerprint length to minimize bit collision, or to investigate in more detail specific difficult compound classes in the data set.

**Cytotoxicity Maps and Comparison to Other Methods** Besides the identification of such novel toxicophores, the DTD relevance scores of all atom environments in a molecule can be depicted to produce a cytotoxicity map of the molecule (adapted from the similarity maps [325] as also used by

Preuer et al. [321, Fig. 4]). Thus, the decomposition of a single molecule is presented entirely which allows easy interpretation of the results, as shown in Figures 4.5b, 4.5c & 4.5d. In this study, the DTD approach is used to select relevant bits to be able to interpret what the model learned. Furthermore, this provides a data-driven approach to identify novel toxicophores.

Other approaches exist that try to unleash the black box in ML, for example, Mayr et al. [319] compare the neurons in the network to predefined toxicophores. Sheridan [320] use a leave-one-feature-out approach on many different modeling settings in order to identify feature importance. Relevances are assigned based on the difference between model scores with a particular feature being present and absent. Recently, Manica et al. [322] published an attention-based neural network architecture to predict  $IC_{50}$  values for known drugs using RNA and SMILES data. The attention vector is calculated from the latent representations and is used to identify the most relevant latent features [343] in the SMILES encoding. Closest to the study presented here is the work by Preuer et al. [321]. In spite of technical details such as model architecture, data set, input featurization, both studies try to understand the toxic mechanism using deep learning. However, not only are the endpoints that are considered different, but the problem is tackled from different angles. The study by Preuer et al. [321] investigates, among other, the role of units in hidden layers as pharmacophore detectors and the issue of bit collision is not addressed. Moreover the method used to investigate the interpretability of neural networks, the so-called Integrated Gradients Method, is different from the Deep Taylor Decomposition as presented in this study. The Integrated Gradients Method, as the name suggests, integrates all the gradients that lie on the path between an input  $x$  and a predefined baseline  $x'$  to obtain a score for each dimension of the input. The integration is numerically approximated by a sum, where the number of steps is predetermined. Obtaining an accurate approximation of this integral requires many time steps (1,000 in the study by Preuer et al. [321]). When comparing the DTD method to Integrated Gradients, DTD is computationally more efficient as only one backpropagation is needed to assign relevances in comparison to 1,000 time steps for a single decomposition in [321]. Both Integrated Gradients and leave-one-feature-out are model agnostic and straightforward to apply, but in contrast the DTD is very intuitive and consistent.

## 4.4 Conclusion

In this study, a deep learning approach to predict the cytotoxicity of compounds is presented using a highly consistent data set of over 34,000 compounds provided by the FMP. Note that the data was composed as screening data set, thus not focusing on cytotoxicity, which led to a low share of cyto-

toxic molecules. Most importantly, a procedure is introduced to make deep learning models more interpretable. In this way, the Deep Taylor Decomposition is used to identify toxicophores in a molecule from a fully-connected feedforward neural network by mapping relevance scores back to atom environments.

The results of the experiments show that the model is competitive with the current literature given data sets with similar share of toxic and non-toxic molecules. The best balanced accuracy on the test set which the FNN model reached is as high as 70.73% which is significantly better than random classification at 50% and the FNN model yielded more balanced results than the baseline RF model. Moreover, using the DTD method, atom environments could be identified which are likely to be involved in cytotoxic behavior of the compounds. As example, the five atom environments with the highest global mean relevance scores were identified and discussed in this study. Molecules in the test set containing these bits were mostly correctly predicted cytotoxic by the FNN model. These findings are coherent with the current literature and especially some of the identified substructures extend the known list of structural alerts. Furthermore, cytotoxicity maps are generated that highlight the contribution of each individual bit, which allow chemists to identify, from these plots, their own relevant toxicophores in newly synthesized compounds.

One aspect that should be considered carefully when applying the approach developed in this study to new molecules is to verify that the compounds are in the scope of the model. For more details on the concept of defining the applicability domain, please refer to Hanser et al. [344]. Generalization to the entire chemical space may be difficult when training any ML model on a static data set. Furthermore, regarding the input features of the model, a noticeable limitation of fingerprints is bit collision which may be ambiguous when trying to identify substructures likely to produce toxic compounds. Using longer fingerprint vectors may help prevent bit collision. An alternative would be to choose a different molecular encoding, such as the SMILES representation as in [345], or a learned representation as developed by Winter et al. [346].

Concluding, the study presents a novel way of interpreting the outcome of the FNN model to help understand what the model learned in the context of molecular toxicity. While most toxicophores are selected by humans, the relevance scores together with the cytotoxicity maps are a technique that identifies these substructures in a data-driven fashion. Spotting such substructures at an early stage of drug design can be highly beneficial for pharmaceutical research to reduce costly and time-intensive laboratory experiments.



## Chapter 5

# Kinase-centric drug design: the importance of pipelines in drug campaigns

The contents of this chapter were published as:

- Sydow, D., Rodríguez-Guerra, J., Kimber, T. B., Schaller, D., Taylor, C. J., Chen, Y., Leja, M. Misra, S., Wichmann, M., Ariamajd, A. & Volkamer, A. (2022). TeachOpenCADD 2022: Open Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research. *Nucleic Acids Research*, 50(W1), W753-W760 [4], under the Creative Commons Attribution (CC BY) license, <https://creativecommons.org/licenses/by/4.0/>.

Contributions:

TBK conceived the theory for some of the notebooks, and was involved in their formal analysis, methodology, validation and visualization. TBK developed parts of the software, and was involved in the maintenance. The text was written by all authors.

- Kimber, T. B.\*, Sydow, D.\*, & Volkamer, A. (2022). Kinase similarity assessment pipeline for off-target prediction [Article v1.0], *Living Journal of Computational Molecular Science*, 3(1), 1599-1599 [5], under the Creative Commons Attribution (CC BY) license, <https://creativecommons.org/licenses/by/4.0/>.

Contributions:

TBK, DS, AV conceived the project. TBK and DS did the formal analysis, developed the methodology, as well as the software, visualized and analyzed the results. TBK led the writing of the text.

---

\*These authors have shared first authorship.

The content from these publications are presented here with the permission Oxford University Press on behalf of Nucleic Acids Research, and the University of Colorado Boulder.



## Chapter summary

Computational pipelines have become a crucial part of modern drug discovery campaigns. Setting up and maintaining such pipelines, however, can be challenging and time-consuming — especially for novice scientists in this domain. TeachOpenCADD is a platform that aims to teach domain-specific skills and to provide pipeline templates as starting points for research projects. We offer Python-based solutions for common tasks in cheminformatics and structural bioinformatics in the form of Jupyter notebooks, based on open-source resources only. Including the 12 newly released additions, TeachOpenCADD now contains 22 notebooks that cover both theoretical background as well as hands-on programming. To promote reproducible and reusable research, we apply software best practices to our notebooks such as testing with automated continuous integration and adhering to idiomatic Python style. The new TeachOpenCADD website is available at <https://projects.volkamerlab.org/teachopencadd> and all code is deposited on GitHub.

Kinases are established drug targets to combat cancer and inflammatory diseases. Despite decades of kinase research, challenges still remain, such as the under-exploration of a large fraction of the kinome and the promiscuous binding of many kinase inhibitors. Due to the highly conserved orthosteric ATP binding site in kinases, ligands may bind not only to their designated kinase (on-target) but also to other kinases (off-targets). Such promiscuous binding can cause mild to severe side effects, and the prediction of these off-targets is highly non-trivial. Therefore, we propose a pipeline that allows the study of kinase similarities from four different angles in an automated and modular fashion. The first method considers the binding site sequence. The second method uses structural information via KiSSim, a newly developed fingerprint that considers both physico-chemical and spatial properties of the binding site. The third method involves kinase-ligand interaction fingerprints as provided by KLIFS, and the last method utilizes the measured activity of ligands on kinases based on ChEMBL data. Finally, results for a given set of kinases are collected and analyzed to gain insight into potential off-targets from the different aforementioned perspectives. Since the pipeline is set up as a series of Jupyter notebooks covering both theoretical and practical aspects, the target audience ranges from beginners to advanced users working in the field of natural and computer sciences. The pipeline is part of the TeachOpenCADD project and extends it with this special kinase edition. All code is free, open-source, and made available at <https://projects.volkamerlab.org/teachopencadd>.

## 5.1 TeachOpenCADD 2022: Open-Source and FAIR Python Pipelines to Assist in Structural Bioinformatics and Cheminformatics Research

### 5.1.1 Introduction

Computational methods play an integral role in the design-make-test-analyze (DMTA) cycle that drives real-world drug design projects [347]. To address questions raised during this cycle, a single method does not suffice to deliver an answer; instead, a pipeline combining different methods can produce complementary and useful insights. Setting up such complex pipelines, however, can be difficult and time-consuming for many reasons: the scientist may not have had the training necessary to tackle these tasks [348], tools and their usage are constantly evolving (or becoming deprecated), and feeding the output from one tool into another is often not straightforward. On top of these considerations, sustainable pipelines need to be findable, accessible, interoperable, and reusable (FAIR principles [61]) — not only today but in many years from now — to drive reproducible research.

In 2019, we launched the teaching platform TeachOpenCADD [80] on GitHub to help face these challenges. TeachOpenCADD teaches by example how to build Python pipelines with open-source resources used in the fields of cheminformatics and structural bioinformatics to answer central questions in computer-aided drug design (CADD). With these ready-to-use pipelines, we target students and teachers who need training material to CADD-related topics, as well as researchers who need a template or an inspiration to tackle their research questions. The theoretical and practical aspects of each topic are covered in an interactive Jupyter notebook [349]. This setup makes it easy for users from different fields to understand the computational concepts and to get started with hands-on Python programming. We call these Jupyter notebooks *talktorials* (talk + tutorial) because their format is suited for presentations as well. The initial stack of talktorials T001–T010 covers common CADD tasks involving webserver queries, cheminformatics, and structural bioinformatics [80]. We show how to fetch chemical and structural data from the ChEMBL [350] and PDB [41, 42] databases and how to encode, filter, cluster, and screen such data sets to find novel drug candidates and off-targets [80]. The talktorials are inspired by several online resources, recommended for further reading [351, 352] and (Practical Cheminformatics, RDKit blog, Is life worth living?). Over the last two years, the TeachOpenCADD GitHub repository underwent many additions and changes: we now have more than doubled our content and extended the application of software best practices rigorously. The full collection of talktorials is easily accessible on the new TeachOpenCADD website. We comply with software best practices regarding the code style as well as maintenance and facilitate installation with a dedicated conda package.

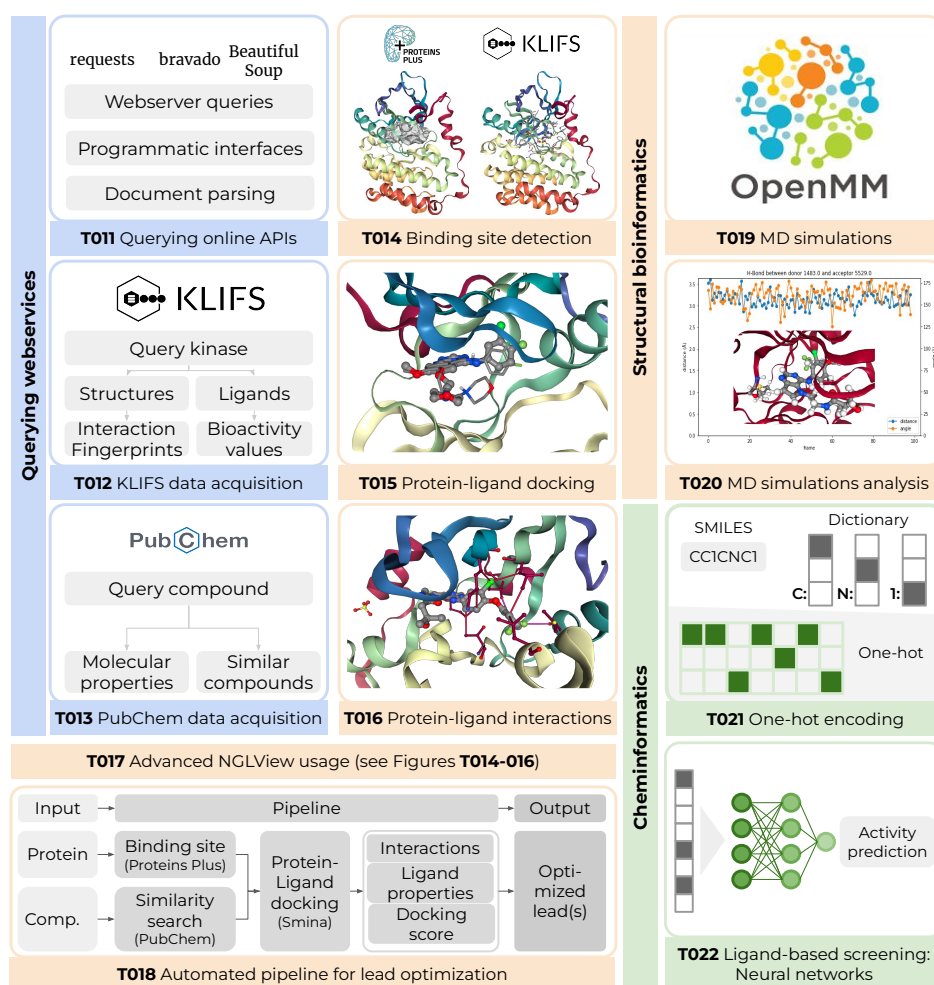


Figure 5.1: Overview of 12 new talktorials. (i) Querying webservices (blue): T011 gives a broad introduction on programmatic access to webservices from Python, T012 and T013 demonstrate how to query the KLIFS [353] and PubChem [354] databases for kinase and compound data, respectively. (ii) Structural bioinformatics (orange): T014 detects the binding site in an EGFR kinase structure and compares the prediction to the binding site defined by KLIFS [353]. T015 performs a re-docking for an EGFR-ligand site complex with Smina [355]. T016 detects protein-ligand interactions in an EGFR-ligand complex structure with PLIP [356]. T017 introduces basic and advanced usages of the molecular visualization tool NGLView [357], used throughout most of TeachOpenCADD’s talktorials. T018 outlines a fully automated lead optimization pipeline: Based on an input structure, the pocket is detected and a set of compounds similar to a selected ligand are fetched from PubChem [354]. These compounds are docked into the selected binding site. The most promising compounds with respect to docking scores and interaction profiles are proposed as optimized compounds. T019 demonstrates how to set up and run a molecular dynamics (MD) simulation on Google Colab with OpenMM [358]. T020 analyzes the resulting MD trajectory with a focus on the root-mean-square deviation (RMSD) between trajectory frames and the dynamics of protein-ligand interactions using MDAnalysis [359, 360]. (iii) Cheminformatics (green): T021 exhibits the steps to numerically encode a small molecule from its SMILES representation. T022 lays the groundwork for deep learning and focuses on a simple feed-forward neural network for activity prediction using molecular fingerprints.

### 5.1.2 New talktorials

The new stack of talktorials showcases data acquisition from additional CADD-relevant databases, adds many examples for structure-based tasks, and extends the cheminformatics side with straightforward DL applications. Our example use case is the EGFR kinase [273] but the talktorials are easily adaptable to other targets as long as sufficient data is available. Besides the domain-specific resources described below, we rely in all talktorials on established Python packages for data science and visualization such as Numpy [32], Pandas [361], Scikit-learn [27], Matplotlib [362], and Seaborn [363].

#### Webservices queries

Over the last decades, the scientific community has produced an incredible amount of data and analysis software, and adapted modern technologies to make these resources easily available via online webservices [364]. However, it might not always be obvious to the beginner how to use a web application programming interface (API) to access such data and how to integrate them into larger pipelines. TeachOpenCADD dedicates several talktorials to the usage of different webservers relevant for the life sciences.

In the first TeachOpenCADD release from 2019, we already showed how to query the ChEMBL [350] and PDB [41, 42] databases. From the ChEMBL webservice, compounds and bioactivities are fetched for the EGFR kinase using the ChEMBL webresource client [183] (T001). This data set is used in many downstream talktorials for common cheminformatics tasks.

From the PDB webservice, we demonstrate how to fetch a set of EGFR kinase structures based on criteria such as "ligand-bound structures from X-ray experiments with a resolution below 3.0 Å" using the biotite [365] and PyPDB [366] (T008) packages.

In the 2021 release, we now have added three more notebooks covering the usage of additional online API webservices (Figure 5.1 T011-T013).

**T011: Querying online API webservices.** We added a broad introduction on how to programmatically use online webservices from Python with a focus on REST services and web scraping. The usage of several libraries is demonstrated; e.g. we use requests to retrieve content from UniProt [367], bravado to generate a Python client for OpenAPI-compatible services — exemplified for the KLIFS database [353] —, and Beautiful Soup to scrape (parse) HTML content from the web.

**T012: Data acquisition from KLIFS.** KLIFS [353] is a kinase database gathering information about experimental kinase structures and interacting inhibitors. The talktorial shows how to quickly fetch data from KLIFS given a query kinase or ligand. For example, we spot frequent key ligand-interactions in EGFR based on KLIFS interaction fingerprints and we assess kinome-wide bioactivity values for the inhibitor gefitinib. These queries are

demonstrated by using the KLIFS OpenAPI directly with bravado, or by using the KLIFS-dedicated wrapper OpenCADD-KLIFS [368] implemented in the Python package OpenCADD.

**T013: Data acquisition from PubChem.** PubChem [354] is a database holding chemical information for over 100 million compounds. We demonstrate how to fetch data from PubChem’s PUG-REST API [369], given the name or SMILES [58] of a query ligand. For example, we show how to fetch molecular properties for a ligand of interest by name (aspirin) and how to query PubChem for the most similar compounds given a query SMILES (gefitinib).

**Data acquisition case study.** A summary of the information that can be acquired automatically for a target of interest using these web services is exemplified in Figure 5.2. Using the Uniprot ID of EGFR kinase as query input only, (i) 227 available EGFR structures from the PDB can be retained and further filtered (T008); (ii) 446 available complex structures and their interaction fingerprints can be retained from KLIFS (T012); or (iii) a total of 8,463  $IC_{50}$  values of molecules measured against EGFR can be obtained from ChEMBL (T001). Finally, (iv) a PubChem query with the molecule name "gefitinib" showcases how to gather ligand properties or to perform a similarity search (T013).

### Pocket detection, ligand-protein docking and interactions

During a drug discovery campaign, frequent questions are: What should I test next? Can you suggest a diverse set of small molecules likely to bind to this protein? How should I modify the lead compound to increase the binding affinity? Answering these questions involves multiple scientific observations, and thus, multiple computational steps as addressed in talktorials T014–T017. Finally, an automated pipeline is compiled (T018) to process a protein structure and a lead compound, and propose several similar ligands with optimized estimated affinities and interactions based on the docked protein-ligand structures.

**T014: Binding site detection.** First, we need to know where ligands may bind to a protein of interest. Sometimes the binding site is known from experimental protein-ligand structures. If only experimental apo structures are available, putative binding sites can be predicted with computational methods. We demonstrate how to use the REST API of the ProteinsPlus webserver [370] to detect the main pocket of an EGFR structure using the DoGSiteScorer [371] pocket detection algorithm. To validate our results, the predicted pocket is compared with the KLIFS-defined kinase pocket, which encompasses 85 residues in contact with ligands in over 2000 kinase-ligand structures [372].

**T015: Protein-ligand docking.** Next, we introduce molecular docking to predict the binding mode of a ligand to its protein target by explaining

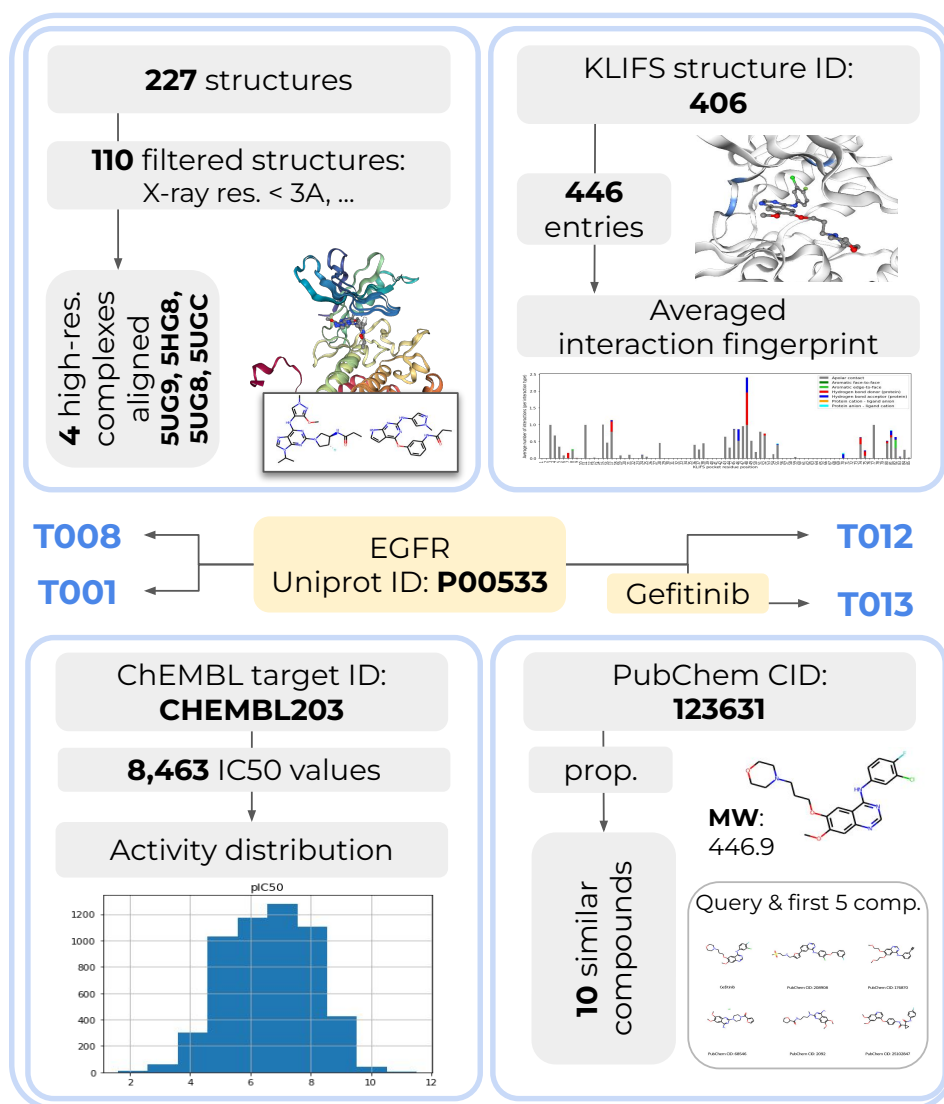


Figure 5.2: Data and information that can be automatically gathered for the EGFR kinase using the different web query talktorials as of September 2021, created based on ChEMBL v.27 [350] (T001), PDB [42] (T008), PubChem [354] (T013), and KLIFS [353] (T012). Input: yellow box, output: grey boxes, plots, and molecule visualizations (using NGLView [357] and RDKit).

several sampling algorithms and scoring functions, as well as commenting on limitations and interpretation of docking results. The theoretical background is then applied in a re-docking experiment aiming to reproduce the binding mode observed in a published X-ray structure of EGFR. Protein and ligand are prepared using Pybel [373], the ligand is docked into the protein using Smina [355], and finally, the docking poses are visually inspected using NGLView [357]. We refer to JupyterDock for further reading on different docking protocols run from Jupyter notebooks.

**T016: Protein-ligand interactions.** Understanding which forces and interactions drive molecular recognition is important for drug design [374]. In this talktorial, we give an introduction to relevant protein-ligand interactions and their programmatic detection using the protein-ligand interaction profiler PLIP [356]. To this end, all interactions in an EGFR-ligand complex fetched from the PDB are detected and visualized in 3D using NGLView.

**T017: Advanced NGLView usage.** Since the molecular visualization package NGLView is invoked in many talktorials, we give a dedicated overview of its usage and show some advanced cases on how to customize residue coloring, and how to create interactive interfaces with IPyWidgets. In addition, access to the JavaScript layer NGL [375, 376] is showcased to perform operations that are not exposed to the Python wrapper NGLView.

**T018: Automated pipeline for lead optimization.** All previous talktorials are composed of stand-alone tasks that can be completed independently. Proposing ligand modifications that will improve interaction patterns with target proteins in a complete end-to-end process, however, necessitates orchestration of code and concepts implemented in the previously discussed talktorials T014–T017. A docking pipeline is constructed in T018 that is comprised of both a step-by-step demonstration and a fully automated procedure. Given a query protein and a lead compound, similar ligands fetched from PubChem are suggested, which show optimized affinity estimates and interaction profiles based on generated docking poses.

**Lead optimization case study** As a case study, an EGFR crystal structure (PDB: 3W32) and its co-crystallized ligand were used as inputs for the pipeline. A similarity search led to the generation of a small library of compounds from PubChem for docking and further analysis to find compounds ideally more affine than the co-crystallized ligand. Using the pipeline, an approved breast cancer drug, gefitinib, was found in the top-50 of docked poses (Figure 5.3). Gefitinib ( $IC_{50} = 0.17\text{nM}$  [377]) is at least an order of magnitude more affine for EGFR than the measured affinity of the input ligand ( $IC_{50} = 75\text{nM}$  [378]). Its predicted geometry was  $< 2 \text{ \AA}$  RMSD from a crystal structure of wild-type EGFR (PDB: 2ITY). This retrospective example demonstrates the utility of a fully automated pipeline and potential application as prospective tool.

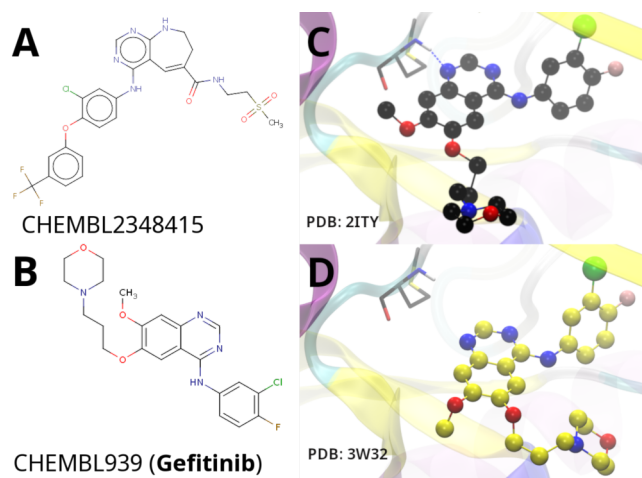


Figure 5.3: Case study for talktorial T018 depicting (A) 2D structure of the input ligand for the pipeline that was used with an EGFR crystal structure (PDB: 3W32,  $IC_{50} = 75\text{nM}$ ); (B) 2D structure of gefitinib ( $IC_{50} = 0.17\text{nM}$ ), an EGFR ligand found during similarity searches; (C) crystal structure of gefitinib co-crystallized with EGFR (PDB: 2ITY, black CPK representation); (D) docked pose (yellow CPK representation). Some segments of the protein structure have been removed for clarity. With an RMSD between the docked pose and crystallized ligand of  $< 2 \text{ \AA}$  and discovery of a higher-affinity ligand.



## Molecular dynamics

Experimentally resolved structures offer immense insights for drug design but can only provide a static snapshot of the full conformational space that represents the flexible nature of biological systems. Molecular dynamics (MD) simulations approximate such flexibility *in silico* with a trajectory of atom positions over a series of time steps (frames). These trajectories thereby reveal a more detailed — albeit still incomplete — picture of drug-target recognition and binding by providing access to protein-ligand interaction patterns over time [379–381]. These insights can for example help in lead discovery to examine the stability and validity of a predicted ligand docking pose, and in lead optimization phases to estimate the effect of a chemical modification on binding affinity.

**T019: MD simulations.** We explain the key concepts behind MD simulations and provide the code to run a short MD simulation of EGFR in complex with a ligand on a local machine or on Google Colab with `condacolab`, which allows for GPU-accelerated simulations. The protein and ligand are thereby separately prepared with `pdffixer` and `RDKit`, and subsequently combined using `MDTraj` [382] and `openff-toolkit`. The simulation is run with `OpenMM` [358], a high-performance toolkit for molecular simulation. The talktorial produces a 100 ps trajectory if run on Google Colab. On a local machine, only 20 fs are generated by default to keep computational efforts reasonable. We refer to the work by Arantes et al. [383] for further reading on different MD protocols run with `OpenMM` using Jupyter notebooks on Google Colab.

**T020: Analyzing MD simulations.** We analyze and visualize the results of the trajectory using the Python packages `MAnalysis` [359, 360] and `NGLView`. First, the protein is structurally aligned across all trajectory frames, followed by calculating the root-mean-square deviation (RMSD) for different system components; i.e. protein, backbone, and ligand. Then, we take a closer look at a selected interaction between ligand and protein atoms, showcasing the contribution of distance and angle to the hydrogen bond strengths.

## Deep learning

Machine learning and more specifically deep learning have gained in popularity over the last few decades thanks to powerful computational resources (GPUs), novel algorithms, and the growing amount of available data [13]. Applications to CADD are diverse, ranging from molecular property prediction [104] to *de novo* molecular design [384]. Here, the focus is the featurization of molecular entities (T021) and ligand-based screening (T022).

**T021: One-hot encoding.** In CADD, machine learning algorithms require as input a numerical representation of small molecules. Besides

molecular fingerprints (see T004), a popular featurization is the SMILES notation [58]. However, these representations are composed of strings and therefore cannot simply be input to an algorithm. One-hot encoding provides a solution for SMILES usage, explained in T021.

**T022: Ligand-based screening: neural networks.** We introduce the basics of neural networks and build a simple two-layer neural network. A model is trained on a subset of ChEMBL data to predict the  $pIC_{50}$  values of compounds against EGFR using MACCS keys as input. This talktorial is meant as groundwork for the understanding of neural networks. More complex architectures such as convolutional and recurrent neural networks will be explored in future notebooks. Such models may use the one-hot encoding of SMILES as input [129].

### 5.1.3 Best practices

We provide reliable and reproducible TeachOpenCADD pipelines, periodically checked via automated testing mechanisms, and a streamlined and easy-to-understand code style across all talktorials.

**Testing.** Reproducibility is ensured by testing if the notebooks can run without errors and whether the output of specific operations can be reproduced. For this purpose, we use the tools `pytest` and `nbval`.

**Continuous integration.** We are testing the talktorials regularly for Linux, OSX, and Windows and different Python versions on GitHub Actions. This ensures identical behavior across different operating systems and Python versions and also spots issues like conflicting dependency updates or changing outputs.

**Repository structure.** The repository structure is based on the CMS cookiecutter template, which provides a Python-focused project scaffold with pre-configured settings for packaging, continuous integration, Sphinx-based documentation, and much more. We have adapted the template to our notebook-focused needs.

**Code style.** We aim to adhere to the PEP8 style guide for Python code, which defines how to write idiomatic Python (Pythonic) code. Such rules are important so that new developers — or in our case talktorial users — can quickly read and understand the code. Furthermore, we use `black-nb` to format the Python notebooks compliant with PEP8.

### 5.1.4 TeachOpenCADD usage

There are many ways to use the talktorials. If users simply want to go through the material, they can use the read-only website version. If users would rather like to execute and modify the Jupyter notebooks, this can be done online thanks to the Binder integrations or locally using the new `conda` package.

**New website.** Firing up Jupyter notebooks can entail unexpected complications if one wants to simply read through a talktorial. To make the access easy and fast, we launched a new TeachOpenCADD website. The website statically renders the talktorials for immediate online reading using sphinx-nb and provides detailed documentation for local usage, contributions, and external resources.

**New Binder support.** The Binder project offers a place to share computing environments via a single link. The environment setup of TeachOpenCADD can take a couple of minutes but does not require any kind of action on the user's end. This access option is recommended if the user plans on executing the material but does not need to save the changes.

**New conda package.** To make the local installation of TeachOpenCADD as easy as possible, we offer a conda package that ships all Jupyter notebooks with all necessary dependencies. The installation instructions are lined out in the TeachOpenCADD documentation. This access option is recommended if the user plans on adapting the material for individual use cases.

### 5.1.5 Conclusion

The increasing amount of data and the focus on data-driven methods call for reproducible and reliable pipelines for computer-aided drug design (CADD). Knowing how to access and use these resources programmatically, however, requires domain-specific training and inspiration. The TeachOpenCADD platform showcases webserver-based data acquisition and common tasks in the fields of cheminformatics and structural bioinformatics. The theoretical and programmatic aspects of each topic are outlined side-by-side in Jupyter notebooks (talktorials) using open-source resources only. To foster FAIR research, we apply software best practices such as testing, continuous integration, and idiomatic coding throughout the whole project. The talktorials are accessible via our website, Binder, and conda package to accommodate different use cases such as reading, executing, and modifying, respectively. We believe that TeachOpenCADD is not only a rich resource for CADD pipelines and teaching material on computational concepts and programming but as well a good example of how to set up websites, automated testing, and packaging for notebook-centric repositories. TeachOpenCADD is a living resource; problems can be voiced via GitHub issues and contributions can be made in the form of pull requests on GitHub. TeachOpenCADD is meant to grow; everyone is welcome to add new topics. Whenever you explore a new topic for your work, we invite you to fill our talktorial template with what one learns along the way and to submit it to TeachOpenCADD.

### 5.1.6 Code and data availability

- TeachOpenCADD website: <https://projects.volkamerlab.org/teachopencadd/>
- TeachOpenCADD GitHub repository: <https://github.com/volkamerlab/teachopencadd>

## 5.2 Kinase similarity assessment pipeline for off-target prediction

### 5.2.1 Introduction

Kinases are involved in most cellular processes by phosphorylating—and thereby activating—themselves or other proteins. This family is among the most frequently mutated proteins in tumors and kinases have been successfully studied as drug targets for many decades [385]. Thanks to the long-standing research, a plethora of kinase data is freely available, i.e., as part of databases such as UniProt [386], PDB [41] or ChEMBL [54], and has been made easily accessible via kinase resources such as the KLIFS—Kinase-Ligand Interaction Fingerprints and Structures—database [353]. As of February 2022, 5,911 X-ray structures of human kinases have been resolved (see the KLIFS database [72]) and 70 FDA-approved small molecule protein kinase inhibitors are on the market [387]. Most of the approved drugs bind in the ATP binding pocket and intermediate surroundings (orthosteric binding site).

Although structural data provides rich information, kinases have been widely classified based on sequence. Manning et al. [388] clustered the human protein kinases based on their sequence similarity into eight major groups (AGC, CAMK, CK1, CMGC, STE, TK, TKL, and "Other") as well as atypical kinases. The resulting Manning kinome tree depicts kinase clustering (see Figure 5.4).

Despite decades of kinase research, challenges still remain [69]. For example:

1. A large fraction of the kinome is un-/underexplored. Figure 5.4a shows the number of PDB structures per kinase, unveiling a vast imbalance between structurally resolved kinases and unexplored ones. For example, CDK2 has been resolved in 426 PDB structures, while only 313 kinases [72] out of approximately 540 in the kinome [69] have been structurally resolved.
2. Many kinase inhibitors are promiscuous binders, causing off-target effects or enabling polypharmacology [385, 389]. For example, the Epidermal Growth Factor Receptor (EGFR) inhibitor erlotinib shows affinities to other kinases in the highly sequentially-similar TK kinase group, but also strongly affects off-targets in more remote kinase groups (see Figure 5.4b).

Therefore, assessing kinase similarity from different angles may be a crucial step in understanding and predicting off-targets to help to design more selective drugs and to avoid side effects.

## Scope

In this study, similarities between a set of kinases are investigated based on methods offering different perspectives on this challenging topic with a focus on orthosteric binding sites (here referred to as binding sites), as summarized in Table 5.1. The first method considers the binding site sequence as deposited in the KLIFS database. The second method uses KiSSim [390], a recently developed fingerprint that considers physico-chemical as well as spatial properties of the binding site. The third method involves protein-ligand interaction fingerprints as provided in the KLIFS database, and the last method utilizes the measured activity of ligands against kinases based on ChEMBL data [54]. The different methods are preceded by a general introduction to kinases and the challenges faced in kinase-centric drug design, and succeeded by a comparison between the different kinase similarity methods.

Note that this study focuses on the similarities between ATP binding sites. Therefore, kinase polypharmacology and off-targets can only be assessed within the scope of orthosteric binding sites, even though the promiscuity of some ligands may be explained by binding to allosteric binding sites. Potential allosteric binding sites are summarized in the Kinase Atlas [391].

This study has been assembled into a modular pipeline that enables the research of kinase similarities in an automated fashion, allowing users to simply use it out of the box, or adapt it to their needs.

This workflow is integrated in the context of TeachOpenCADD [4, 80], a teaching platform for computer-aided drug design (CADD) using open-source packages and data. Specific tasks in cheminformatics and structural bioinformatic are described and solved using Python-based Jupyter notebooks [394] as interactive platform. All code has been deposited on GitHub, see <https://github.com/volkamerlab/teachopencadd>. The project website can be found at this link, <https://projects.volkamerlab.org/teachopencadd>.

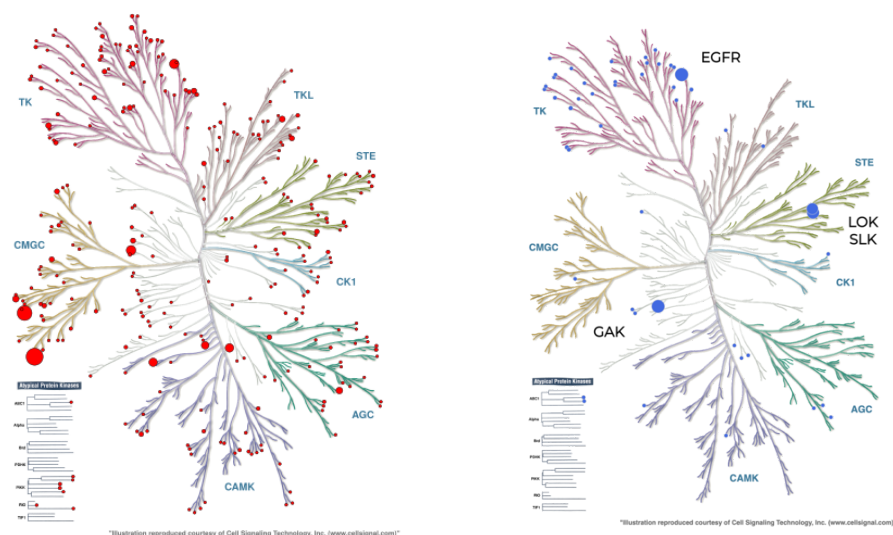
## 5.3 Prerequisites

### Target audience

The notebooks were developed to support researchers interested in kinase-centric computational drug design, with a focus on understanding and predicting kinase off-targets. As this collection is part of the TeachOpenCADD training material [4, 80], we also recommend the notebooks to teachers as pedagogical interactive material in structural bioinformatics and cheminformatics.

Table 5.1: TeachOpenCADD kinase edition overview: Notebook topics, description, and index with a hyperlink to the associated notebook.

<b>Topic</b>	<b>Description</b>	<b>Hyperlink</b>
What is a kinase?	Introduction to kinases and challenges in drug discovery.	T023
Pocket sequence	Pairwise similarities/identities between 85 residue long KLIFS pocket sequences.	T024
Pocket structure	Pairwise similarities between 1,032-bit long KiSSim fingerprints, which encode spatial and physico-chemical pocket properties.	T025
Pocket-ligand interactions	Pairwise similarities between 595-bit long KLIFS kinase-ligand interaction fingerprints (IFP).	T026
Ligand profile	Pairwise similarity based on the ratio of compounds tested active against kinase pairs.	T027
Kinase similarity	Comparison between predicted off-targets based on calculated kinase similarities using aforementioned methods.	T028



(a) Number of PDB structures per kinase. The figure shows the imbalance between highly explored kinases, for example, the groups TK and CMGC. The CDK2 kinase in the CMGC group has the most structures, with 426. The red circle is proportional to the number of PDB structures for each kinase, such that the greater is the circle, the higher is the number of structures.

(b) Developing selective kinase inhibitors is non-trivial since kinases are highly conserved in the ATP binding site. EGFR inhibitor erlotinib binds not only to its intended target EGFR, but also to kinases in remote groups, such as SLK/LOK in the STE group and GAK in the "Other" group. The blue circle is proportional to the  $K_d$  value in nM taken from the Karaman et al. [392] dataset.

Figure 5.4: Visual representation using the Manning tree of existing challenges in kinase research: un-/underexplored kinase groups (left) and the promiscuous behavior of kinases (right). The figure is taken from [https://projects.volkamerlab.org/teachopencadd/talktorials/T023\\_what\\_is\\_a\\_kinase.html](https://projects.volkamerlab.org/teachopencadd/talktorials/T023_what_is_a_kinase.html) and is generated using KinMap [393].



## Background knowledge

The notebooks are constructed in a way that no in depth prior knowledge besides an affinity for the natural or computer sciences is required. Each notebook eases into the topic of kinase drug development and kinase similarity with a lot of theoretical background and comments on all content as well as programming-related steps in great detail. Nevertheless, users will benefit from a basic understanding of the Python programming language and the usage of Jupyter notebooks. If such basic introduction is needed, please refer to training material as listed on the TeachOpenCADD website [395].

## Software requirements

The notebooks are written in Python and rely on open-source packages such as pandas [284], numpy [396], scipy [397], matplotlib [398], seaborn [363], scikit-learn [27], rdkit [399], biotite [365], opencadd [368], kissim [400], and requests [401].

The user only needs to install the *teachopencadd* conda-forge package [402] (see installation [403]), which will install all relevant packages and save a copy of all TeachOpenCADD notebooks—including the kinase edition discussed in this paper—on the user's local machine. A read-only mode of the notebooks is accessible via the TeachOpenCADD website at <https://projects.volkamerlab.org/teachopencadd/>. Online execution can be done via Binder [404], using the following link <https://mybinder.org/v2/gh/volkamerlab/TeachOpenCADD/master>.

## 5.4 Method

In this section, the four methods that are introduced to measure kinase similarity are described, namely the pocket sequence, the KiSSim fingerprint, the interaction fingerprint, and the ligand profile. Note that the theoretical and practical aspects of each method are also covered in great detail in the individual notebooks of this kinase collection (Table 5.1). As discussed in the "Scope" section of this manuscript, we focus on kinase similarity based on orthosteric binding sites.

### Pocket sequence

The full amino acid sequence is often used to assess similarities between kinases (see the phylogenetic tree developed by Manning et al. [388]). Since binding sites are often more conserved than the whole protein, van Linden et al. [405] defined as part of KLIFS a 85-long pocket sequence that is aligned across the kinome. Using a sequence that focuses on the binding site seems appropriate in the case of kinases, since this is where the ligand is likely to

bind. Moreover, working with a fixed length sequence is practical from a computational point of view.

In this study, two methods are used to compute relationships based on sequence, namely the sequence identity and the sequence similarity, which are described below.

**Sequence identity** The pairwise sequence identity, or simply sequence identity, is a similarity based on character-wise discrepancy, in other terms, the number of residues that match in two aligned sequences [406]. More formally, given two kinase sequences  $S$  and  $S'$  of same lengths  $L$ , the sequence identity can be defined as

$$\text{sequence identity}(S, S') = \frac{1}{L} \sum_{n=1}^L I(S[n], S'[n]), \quad (5.1a)$$

where  $I$  is the identity matrix of the amino acids, and  $S[n]$  the amino acid at position  $n$  of the kinase sequence  $S$ . Note that not all kinases have residues present at each of the 85 alignment positions. Such gaps are represented by "-" and count as mismatch to any amino acid.

**Sequence similarity** Unlike sequence identity which treats all residues uniformly, pairwise sequence similarity, or sequence similarity, takes into account the change of the amino acids over evolutionary time, thus, reflecting relationships between amino acids. It is based on a substitution matrix  $M$ , where each entry gives a score between two amino acids. In this study, the BLOSUM substitution matrix [407], as implemented in biotite [408], is used. Formally, the following is defined:

$$\text{sequence similarity}(S, S') = \frac{1}{L} \sum_{n=1}^L M'(S[n], S'[n]), \quad (5.1b)$$

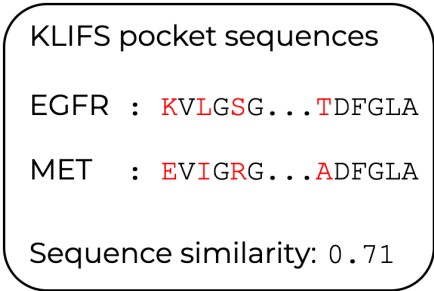
where  $M'$  is the translated and rescaled version of the substitution matrix  $M$ .

For both the sequence identity and similarity, the closer the value is to 1, the more similar are the kinases.

Figure 5.5 shows the sequence similarity between the KLIFS pocket sequence of EGFR and MET kinases. Sequence similarity is used by default in the pipeline for further analysis.

### The KiSSim fingerprint

In order to assess the pairwise similarity of kinases from a structural point of view, the newly developed KiSSim (**K**inase **S**tructure **S**imilarity) fingerprint [390, 400] is used. This fingerprint describes the physico-chemical



KLIFS pocket sequences

EGFR : KVLGSG...TDFGLA

MET : EVIGRG...ADFGLA

Sequence similarity: 0.71

Figure 5.5: Sequence similarity between EGFR and MET. The 85-residue pocket sequence is retrieved from KLIFS. The pairwise sequence similarity takes into account the change of the amino acids over evolutionary time.

and spatial properties of structurally resolved kinases, while focusing on the KLIFS pocket residues. Each structure is mapped to a fingerprint composed of 1,032 bits, the first 680 ( $= 85 \times 8$ ) bits describing physico-chemical features and the remaining 352 ( $= 85 \times 4 + 12$ ) bits spatial information (see Figure 5.6).

**From several structures to one kinase** A kinase can be represented by one or even a hundred resolved crystal structures in the PDB (see Figure 5.4a). In this study, we aim at comparing different kinases and not individual structures. Since KiSSim generates a fingerprint for each structure, the following mapping from structures to kinase is applied:

Given two kinases  $K$  and  $K'$ , all available structures in KLIFS for these kinases are fetched using `opencadd` [368], namely  $s_1, \dots, s_m$  for kinase  $K$ , and  $s'_1, \dots, s'_n$  for kinase  $K'$ , noting that the number of structures might be different for each kinase. Each structure  $s_i, s'_i$  is then mapped to its corresponding KiSSim fingerprint  $fp_i, fp'_i$ , see Figure 5.7. The fingerprints  $fp, fp'$  corresponding to kinases  $K, K'$  respectively, are the ones for which the Euclidean distance is minimized (Figure 5.7). Note that these *minimal distance* fingerprints vary for each kinase depending on the compared  $K, K'$  pair.

Finally, two kinases  $K, K'$  are compared based on their respective *minimal distance* between KiSSim fingerprint  $fp, fp'$  using the Euclidean norm:

$$\text{KiSSim dissimilarity}(fp, fp') = \|fp - fp'\|_2. \quad (5.2)$$

In this case, the closer the value to 0, the more similar the kinases.

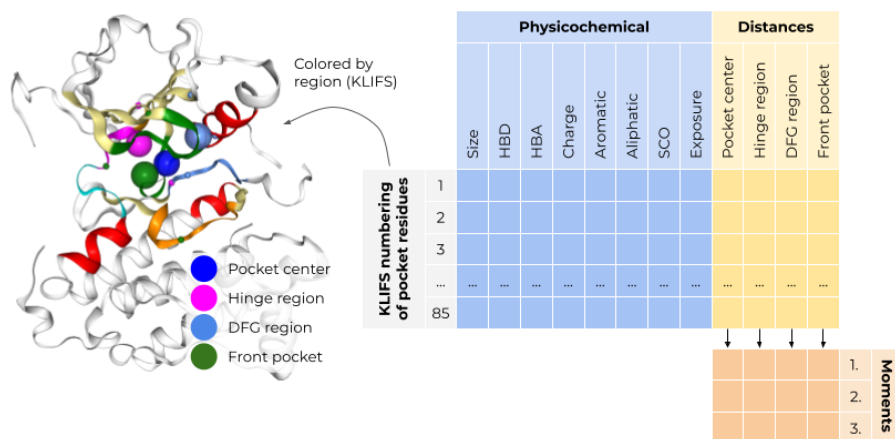


Figure 5.6: The 1,032-long KiSSim fingerprint encodes physico-chemical and spatial properties of the kinase’s pocket, adding a structural perspective on kinases. The figure is adapted from [400].

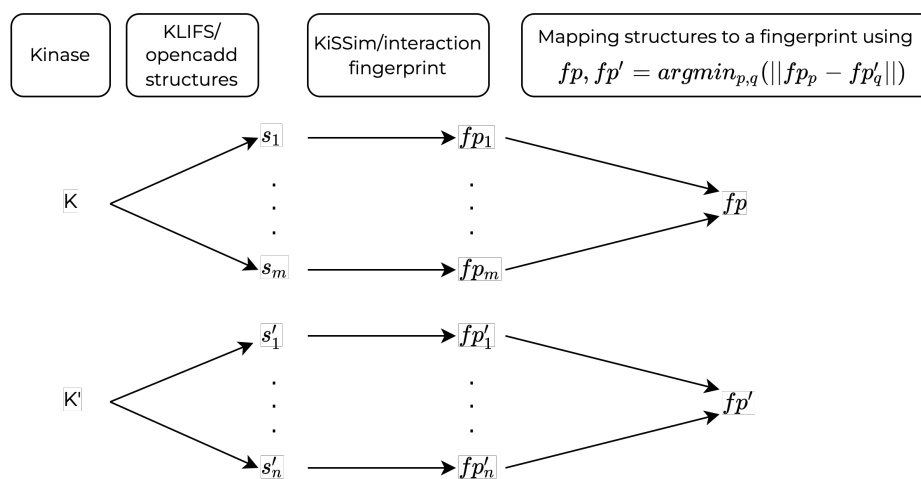


Figure 5.7: Associating one structural fingerprint per kinase. All available structures are retrieved for two given kinases and all fingerprints are computed. The fingerprints selected to be associated with the kinase in the present kinase pair are the ones for which the computed distance is minimized.

1								2								3								85							
HYD	F-F	F-E	DON	ACC	ION+	ION-		HYD	F-F	F-E	DON	ACC	ION+	ION-		HYD	F-F	F-E	DON	ACC	ION+	ION-		HYD	F-F	F-E	DON	ACC	ION+	ION-	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	

Figure 5.8: The KLIFS interaction fingerprint encodes seven interaction types for each of the 85 residues in the binding site. Interaction types include: hydrophobic contacts (HYD), face to face aromatic interactions (F-F), face to edge aromatic interactions (F-E), protein H-bond donors (DON), protein H-bond acceptors (ACC), protein cationic interactions (ION+), and protein anionic interactions (ION-). The figure is taken from [409].

### The interaction fingerprint

Interaction fingerprints (IFPs) encode the binding mode of a ligand in a binding site, i.e., the protein-ligand interactions that are present in a structurally resolved complex. If a ligand can form similar interaction patterns in proteins other than its designated protein (off- vs. on-target), it is possible that this ligand will cause unintended side effects. Knowledge about binding mode similarities can therefore help to avoid such off-target effects.

The KLIFS interaction fingerprint describes seven possible interactions for each of the 85 residues in the binding pocket. Interactions include 1. hydrophobic contacts, 2. aromatic interactions, face to face, 3. aromatic interactions, edge to face, 4. H-bond donors, 5. H-bond acceptors, 6. cationic interactions, and 7. anionic interactions. The 595-bit long vector describes the presence or absence of such interactions for all 85 residues (see Figure 5.8).

Similarly to the KiSSim comparison, given two kinases  $K$  and  $K'$ , all available structures in KLIFS for these kinases are fetched using `opencadd` [368]. Each structure is mapped to its corresponding IFP. The interaction fingerprints  $fp$ ,  $fp'$  corresponding to kinases  $K$ ,  $K'$  respectively are the ones for which the Jaccard distance [410] is minimized (Figure 5.7). Note that the Euclidean distance is used in case of the KiSSim fingerprint, which contains continuous and discrete values, while the Jaccard distance is employed in case of the binary IFPs.

Finally, two kinases  $K$ ,  $K'$  are compared using their respective *minimal distance* between interaction fingerprint  $fp$ ,  $fp'$  and calculating the Jaccard distance:

$$\text{IFP dissimilarity } (fp, fp') = d_J(fp, fp'), \quad (5.3)$$

where  $d_J$  is the Jaccard distance.

In this case, the closer the value to 0, the more similar the kinases.

### Ligand profile

In the context of drug design, the following assumption is often made: if a compound was tested active on two different kinases, it is suspected that these two kinases may have some degree of similarity [411]. This is the rationale behind the ligand profile similarity. Given bioactivity data for a set of compounds measured against a set of targets—in this case kinases—and two kinases  $K$ ,  $K'$ , ligand profile similarity is defined as

$$\text{lig. profile similarity}(K, K') = \frac{\# \text{ actives on both } K \text{ and } K'}{\# \text{ tested on both } K \text{ and } K'}. \quad (5.4)$$

The closer the value is to 1, the more similar are the kinases. If no compounds were commonly tested on two kinases, then the similarity is set to 0. Computing the similarity between a kinase and itself may be interpreted as kinase promiscuity, where the similarity described above would therefore represent the fraction of active compounds over all tested compounds for this kinase.

**Bioactivity data** The bioactivity data used for this method comes from Kinodata [412], from the Openkinome organization [413]. It is a pre-processed kinase subset of the ChEMBL data [54], version 29. Further processing includes keeping only  $IC_{50}$  values given in nM, and converting them to  $pIC_{50}$  values. If there are several measurements for a kinase-compound pair, then the most active value, i.e., the entry with the highest  $pIC_{50}$  value, is kept. Finally, the  $pIC_{50}$  values are binarized using a 6.3 cutoff to discriminate between an active or inactive compound as described in [414].

In the pipeline, one can additionally compute the non-reduced ratio of number of active compounds against the total number of compounds to gain insight into the actual number of measurements for each kinase pair.

### Kinase comparison and clustering

To assess kinase similarities based on the calculated (dis)similarity matrices, two visualization methods are used, namely heatmaps and dendrograms.

**Heatmaps** The heatmaps are generated using matplotlib [398] to depict the similarity between a set of kinases. The maximum value is 1, indicating exact similarity, as is the case for diagonal entries. The value 0 indicates total dissimilarity. Plotting such figures allows to see and extract patterns thanks to the gradient of colors, see top row in Figure 5.9.

**Dendrograms** Clustering algorithms are used to identify groups such that the similarities within clusters are higher than compared to other clusters [415]. In this study, hierarchical clustering is used, and, unlike heatmaps,

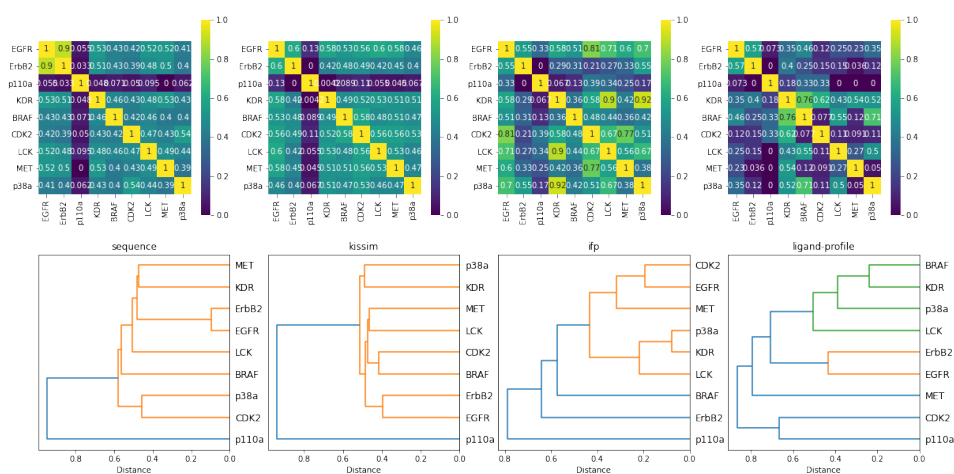


Figure 5.9: Visualization of kinase similarity from four different angles: sequence, KiSSim, interaction fingerprint (ifp) as well as ligand-profile. The top, bottom row shows four heatmaps, dendrograms respectively for a set of nine study kinases.

it is based on distance (or dissimilarity). Hierarchical clustering can be graphically displayed using a dendrogram (see bottom row in Figure 5.9), where the height of each node is proportional to the dissimilarity between its two daughter clusters. The clustering and plotting is done using scikit-learn [27] and matplotlib [398], respectively.

For fair comparison, the distance matrices for all four methods are normalized so that each entry lives between 0 and 1. Similarity matrices—as used for the heatmaps—are then computed using  $1 - \text{distance}$  matrix. Contrary to the dendrograms, that use the distance matrix.

## 5.5 Pipeline

Measuring kinase similarity is a non-trivial task; distinct measures can provide different insights, which can be complementary, confirmatory, or contradictory, and therefore expand our knowledge on the target(s) at hand. However, implementing multiple methods can be time-consuming and comparing results across many output types can be laborious. Turning such processes into a functional pipeline helps to avoid the scattering of scripts and to speed up iterations of the design-make-test-analyze cycle [416] of drug design campaigns. Moreover, following the findable, accessible, interoperable, and reusable (FAIR) principles [61] makes such pipelines long-lasting and available to the community.

In the pipeline presented herein, we implemented the different methods once and streamlined each method’s results into a standardized output with

a pre-defined set of visualization tools for easy comparison. Moreover, the pipeline is flexible enough so that adding new methods or new visualization tools is effortless, making the whole process easy to understand, maintain, and expand.

### **Means of the pipeline**

The proposed pipeline is a collection of six Jupyter notebooks [394] that allows the study of kinase similarity from four different angles in an automated and modular fashion (Figure 5.10).

### **Structure of the notebooks**

The structure of all notebooks is as follows: the first section covers the theory written in Markdown and summarizes the necessary concepts to understand the task. Relevant references are also mentioned. The second part of a notebook deals with the actual implementation of the task in a pedagogical manner, including motivation for practical steps and detailed comments on coding decisions. Finally, a discussion and a quiz section wrap up the notebook. This structure is very well suited from a teaching perspective, since it contains both theory and hands on programming. The notebook can easily be used as a medium for a presentation, and it allows for self-study as well as usage in own research projects.

### **About the code**

The programming section is done in Python exclusively and the code follows the latest software best practices. It is written pythonically and contains lots of code comments. Thanks to the continuous integration (CI), all outputs and results are fully reproducible and the maintenance of the pipeline is facilitated.

### **Content of the pipeline**

As mentioned previously, the proposed pipeline contains six notebooks, described below:

The first notebook sets the stage with a kinase introduction and references/tools on where to find kinase-related information. It is also in this first notebook that a set of kinases of interest is defined. In this study, nine kinases are selected, the same nine as in the paper by Schmidt et al. [417], where the authors discussed the challenges and advantages of tackling kinase similarity from multiple perspectives. Table 5.2 summarizes the information used for these kinases. The pipeline can be executed out of the box with the defined set of kinases, but it can equally be run with a different user defined set of kinases. The only condition is that the uploaded CSV file with the



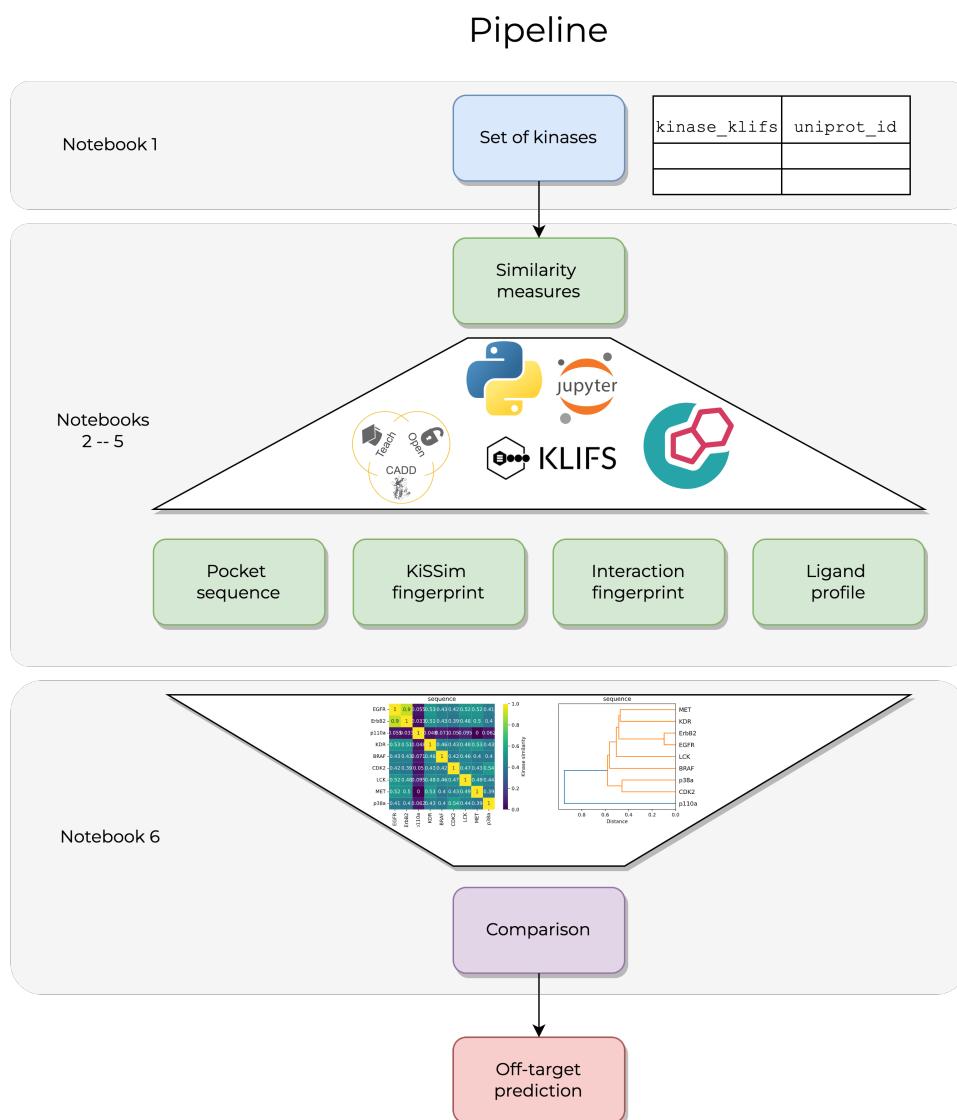


Figure 5.10: The proposed pipeline consists of six Jupyter notebooks [394]. Given a set of kinases in a CSV format, four similarity measures are implemented, and kinases are compared using heatmaps and dendrograms. The project is part of TeachOpenCADD [4, 80] and uses open-source tools and databases such as KLIFS [353] and ChEMBL [54].

kinases of interest contains two mandatory columns, namely `kinase_klifs`, which is the KLIFS name of the kinase, and `uniprot_id`, the Uniprot identifier (ID) [386] of the kinase (Figure 5.10).

The four following notebooks describe one similarity method at a time as discussed in Section 5.4: the pocket sequence, the KiSSim fingerprint, the interaction fingerprint, and the ligand profile.

The final notebook collects the information from the previous ones and compares the different perspectives with easy-to-understand visualization such as heatmaps and dendrograms (see Section 5.4). Additionally, an equally weighted average to combine distance and similarity matrices from all four perspectives can be computed, yielding a single heatmap, and a single dendrogram. The user can easily extend this to a knowledge-informed weighting scheme based on their own research focus.

### Features of the pipeline

The developed pipeline contains many useful features. Firstly, it is part of the TeachOpenCADD project [4, 80] and extends it with this special kinase edition. Being part of TeachOpenCADD has the following advantages:

1. TeachOpenCADD is open-source and freely available at <https://github.com/volkamerlab/teachopencadd>, under the Attribution 4.0 International (CC BY 4.0) license.
2. A dedicated conda package [418] facilitates installation.
3. Online execution is possible via the Binder project [404].
4. The teaching approach makes the notebooks easy to follow.

Moreover, the pipeline is easily adaptable to new sets of kinases as well as new similarity methods, defined by a user.

## 5.6 Conclusion

In this study, a full pipeline for the assessment of kinase similarity is presented, using four methods of comparison. The pipeline is composed of six Jupyter notebooks:

1. An introduction to kinases and their central role in drug discovery, as well as the collection of the kinase set for the downstream notebooks.
2. The similarity from a pocket sequence point of view.
3. The similarity based on the KiSSim fingerprint, which encodes physico-chemical and spatial properties of the kinase pocket.

Table 5.2: Set of defined kinases. The table lists the kinases used in the pipeline, the same nine as in the study by Schmidt et al. [417]. It is noteworthy that the pipeline is applicable to an arbitrary set of kinases, the only condition being that the input CSV file should contain two columns, **kinase\_klifs** and **uniprot\_id**, displayed in bold.

kinase	<b>kinase_klifs</b>	<b>uniprot_id</b>	group	full kinase name
EGFR	EGFR	P00533	TK	Epidermal growth factor receptor
ErbB2	ErbB2	P04626	TK	Erythroblastic leukemia viral oncogene homolog 2
PI3K	p110a	P42336	Atypical	Phosphatidylinositol-3-kinase
VEGFR2	KDR	P35968	TK	Vascular endothelial growth factor receptor 2
BRAF	BRAF	P15056	TKL	Rapidly accelerated fibrosarcoma isoform B
CDK2	CDK2	P24941	CMGC	Cyclic-dependent kinase 2
LCK	LCK	P06239	TK	Lymphocyte-specific protein tyrosine kinase
MET	MET	P08581	TK	Mesenchymal-epithelial transition factor
p38a	p38a	Q16539	CMGC	p38 mitogen activated protein kinase alpha

4. The similarity based on KLIFS interaction fingerprints between the kinase pocket residues and a co-crystallized ligand.
5. The similarity based on ligand profiling data collected from ChEMBL, measuring a compound's activity on a kinase.
6. An analysis notebook which collects the proximity matrices calculated for the four methods, visualizes the similarities with heatmaps and the clusters with dendrograms, and finally discusses the results.

We encourage users to develop their own similarity methods and to contribute to the existing pipeline.

This paper could be of interest to

1. researchers who want to gain insights into off-target prediction and kinase similarity, and integrate their new comparison methods to a working workflow,
2. beginners in software development who need inspiration to set up a fully functional pipeline,
3. teachers who want a starting point for lecture material,
4. students with a background in bioinformatics, cheminformatics, and the life sciences in general,
5. anyone who is curious.

## Chapter 6

# Conclusion

The global COVID-19 pandemic hit the world by surprise in March 2020. Nobody knew for certain how long it would last: a year? Two years? A decade? Or even more? To date, although lockdown and restrictions have been lifted in most, if not all, European countries, coronavirus cases are still emerging, and death casualties are exceeding 6 million worldwide [419]. If there is one take away message from this global crisis, it is that new diseases will keep appearing and finding efficient treatments rapidly is of utmost importance. In other words, drug design is not ready to fade away. Tremendous collaborative scientific effort has been devoted to finding efficient oral drugs for coronavirus, as in the case of the Covid Moonshot project [420].

In this thesis, we have shown, among other, how implementing modular open-source pipelines could potentially speed up the discovery of new treatments, and how deep learning models show great improvements in the prediction of important physicochemical properties of drugs.

More specifically, in Chapter 1, we discussed the challenges that arise in the development of new drugs, and particularly, cancer treatments. We discussed the role of kinases as drug targets and explained how designing selective drugs is challenging due to the highly conserved binding site in the human kinome.

In Chapter 2, we exhibited the state-of-the-art in virtual screening and described methods to find molecules in the huge chemical space, which is approximately  $10^{60}$  [421], that are active and selective toward a given target. Virtual screening workflows were described, including ligand-based, that take into account ligand information only, complex-based, that requires a protein-ligand complex, and pair-based, that depends on independent protein and ligand information. In this context, several molecular encodings were described. For ligands, molecular graphs, SMILES, and circular fingerprints were covered. In the case of proteins, identifier, sequence, and structural fingerprints were described, and finally a variety of interaction fingerprints, three-dimensional grids, and complex graph were outlined. Moreover, deep

learning models were introduced, encompassing multilayer perceptrons, convolutional, recurrent, and graph neural networks. Popular data sets used in virtual screening were summarized, including PDBbind, DUD-E, ChEMBL, and kinase-specific data sets. Finally, research using prominent deep learning models reaching state-of-the-art results were analyzed, showing remarkable progress in the accuracy of affinity prediction.

Chapter 3 dealt with one of the major challenges in drug design, the scarcity of data, data which is crucial when applying a deep learning model. We developed augmentation techniques revolving around the idea that multiple valid SMILES exist for one molecular compound. Models based on convolutional and recurrent layers were built and trained on physicochemical and bioactivity tasks. More specifically, the affinity towards the EGFR kinase was predicted based on a preprocessed version of ChEMBL data. Performance metrics greatly improved when applying these augmentation strategies. For example, on the FreeSolv data set, the root mean squared error on the test set dropped from 1.96 using one SMILES per compound to 1.03 generating 70 SMILES per compound. The results of this research outperformed the results from studies using the same data sets, but without augmentation techniques.

In Chapter 4, we described a way to interpret the outcome of a deep learning model. The model built, FNN —feedforward fully-connected neural network—, is composed of three fully-connected hidden layers of 512, 192, and 128 units, respectively. The input to the model is the 2,048-bit long Morgan fingerprint, a binary vector indicating the presence or absence of molecular substructures. The output determines whether a molecule should be classified as cytotoxic. The model was trained on an imbalanced, yet consistent data set from the FMP, the Leibniz-Forschungsinstitut für Molekulare Pharmakologie, containing over 34,000 molecule, label pairs. The model reached similar performance metrics as ones trained on similarly composed data sets. Moreover, using the Deep Taylor Decomposition, we were able to map the output of the model to its input, assigning to each atom environment a cytotoxicity, or relevance, score. The substructures with the highest score were identified as toxicophores, and a visualization technique to show these toxicophores within a 2D molecular compound was developed.

Chapter 5 stressed the importance of automated pipelines in drug design and showcased the implementation of a workflow in the context of kinases, known for their role as drug targets. The pipeline is part of TeachOpenCADD, a platform dedicated to open-source tasks in computer-aided drug design. Various similarity measures were explored and implemented, including the ATP binding site sequence from KLIFS, the KiSSim fingerprint which consists of physicochemical and spacial information, the kinase-ligand interaction fingerprint defined in KLIFS, as well as ligand profiling data, queried from ChEMBL. The pipeline generates cluster trees such that kinases that are considered similar given a particular measure are grouped together. The

results showed that there is some concordance across some perspectives. For example, the EGFR and ErbB2 kinases, of the same family, were grouped together in three out of four approaches. And the atypical p110a kinase is a singleton when using sequence, structure and interaction. Such a pipeline confirms the importance of exploring kinases using different measures in order to gain insight into off-targets.

Although astounding progress has been made in computer-aided drug design, challenges still remain. One of the most crucial ones is data: from scarcity, to heterogeneity, labelling, and provenance, as reported by Bender and Cortes-Ciriano [241]. In light of Chapter 3 and in the aim of providing solutions to data scarcity, although SMILES notation for small molecules is largely popular in cheminformatics and can be easily computed using software such as RDKit [131], no 3D information about the compound is retained. Would using the atomic coordinates provide beneficial information to improve the accuracy of the models? Knowing that different conformations of a molecule could be exploited as data augmentation, would training a deep neural network be a sustainable solution given the tremendous computational cost it would require? Future work could investigate the existence of a trade-off between the simplicity (and therefore low computational cost of SMILES) vs. the rich, three dimensional atomic information (and the expensive compute cost) of a molecule.

Regarding Chapter 5, the main focus was on kinases, a family of proteins, known for their involvement in various diseases such as cancer. Outlook and extension could cover the possibility of translating the methods developed in the context of kinases to other protein families. For example, G-protein-coupled receptors (GPCRs) that are involved in various diseases and for which only meager data exist. More broadly, how could the kinase-centric methods and associated pipelines be applied to proteins in general to steep up the drug discovery process?

Given the necessity of designing new treatments for both existing and emerging diseases, this thesis presents improvements for multiple challenges related to machine learning in computed-aided drug design. It also provides useful methods and pipelines, and discusses the possibilities for future research.





## Appendix A

# Evaluation strategies and metrics

In the following section, metrics commonly used to evaluate the performance of a given model are described. Depending on the learning task, these metrics may vary. First, metrics used in binary prediction leading to a classification framework are discussed. Then, the typical metrics used in the regression framework are described and finally the metrics that can be applied in both cases.

**Classification** Often times, the area under the ROC curve (AUC) [422] is reported, when the learning task requires determining if there is a hit or non-hit given a ligand and a protein. The receiver operating characteristic (ROC) curve plots the true positive rate ( $y$ -axis) versus the false positive rate ( $x$ -axis) when the threshold of the classifier varies. Obtaining a 100% true positive rate and 0% false positive rate would be the ideal setting and would yield an AUC value of 1. As a measure to compare models, the closer the AUC value is to 1, the better the classifier. A detailed explanation of ROC curves and AUC values can be found in [423].

The accuracy (Acc) [424] is defined by

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}},$$

where TP, TN, FN, FP are the true positives, true negatives, false negatives and false positives, respectively.

The enrichment factor (EF) [193] is a measure for evaluating screening efficiency. At a pre-defined sampling percentage  $\chi$ ,  $\text{EF}_{\chi\%}$  shows the proportion of true active compounds in the sampling set in relation to the proportion of true active compounds in the whole data set and is defined by

$$\text{EF}_{\chi\%} = \frac{\frac{n_s}{N_s}}{\frac{n}{N}},$$

where  $N$  is the number of compounds in the entire data set,  $n$  the number of compounds in the sampling set,  $N_s$  the number of true active compounds in the entire data set, and  $n_s$  the number of true active compounds in the sampling set.

**Regression** In the following,  $y_i$  is assumed to be the  $i^{\text{th}}$  true value of  $y$ ,  $\hat{y}_i$  the  $i^{\text{th}}$  predicted value of  $\hat{y}$  by the algorithm, and  $n$  the number of data points considered.

The mean squared error (MSE) measures the difference between  $y$  and  $\hat{y}$  using the Euclidean norm:

$$\text{MSE} = \text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

The root mean squared error (RMSE), as its name suggests, is simply the squared root of the MSE,

$$\text{RMSE} = \sqrt{\text{MSE}}.$$

Since the MSE and the RMSE are metrics that represent the difference between the true and predicted values, the smaller these errors are, the better the model.

**Classification & regression**  $R^2$  is a measure of goodness of fit which can be applied in both the classification and the regression framework [425, 426] and is defined by

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

The closer is the  $R^2$  to 1, the better the fit.  $R^2 = 0$  when the model predicts all values to the mean  $\bar{y}$ .

The Pearson's correlation coefficient  $R$  [189, 221, 427] defined below is also often used.

$$R = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2 \sum_i (\hat{y}_i - \bar{\hat{y}})^2}}.$$

The Spearman's rank correlation coefficient  $\rho$  [190, 191] can be calculated with

$$\rho = 1 - \frac{6 \times \sum_i (\text{rank}(y_i) - \text{rank}(\hat{y}_i))^2}{n \times (n^2 - 1)},$$

where  $n$  is the number of observations. In contrast to Pearson's correlation coefficient  $R$ , the ranks between the observed ( $y_i$ ) and predicted ( $\hat{y}_i$ ) values are compared.



## Appendix B

### Figures

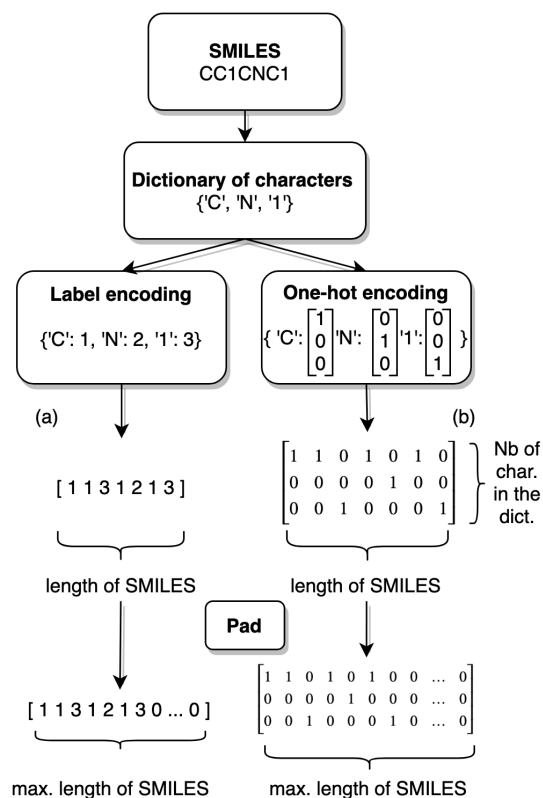


Figure B.1: **Encoding and padding.** Starting from a SMILES string, the characters can be stored in a dictionary (or the dictionary can be constructed prior using a set of known characters). Label encoding consists of enumerating the characters in the dictionary. One-hot encoding consists of assigning a binary vector to each character. (a) Constructing the label encoding for a given SMILES by assigning the integer associated to the character as they appear in the SMILES. (b) Constructing the one-hot encoding by concatenating the binary vectors of the characters as they appear in the SMILES. (Pad) Inputs of same dimension are often required when using machine learning. A common solution is to use padding, which consists of adding zeros to either the label vector or to the one-hot matrix up to the maximum length of the SMILES in the data set.

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	2.577	2.868	2.514	2.603	2.881	2.396	2.691	3.017	2.431
2	2.108	2.222	1.989	2.021	2.103	2.122	2.091	1.979	2.002
3	1.914	2.315	1.900	1.956	1.967	1.810	1.917	2.371	2.029
4	1.494	1.646	1.545	1.444	1.692	1.479	1.420	1.594	1.568
5	1.260	1.524	1.381	1.342	1.517	1.332	1.278	1.478	1.469
6	1.306	1.580	1.398	1.277	1.723	1.352	1.306	1.656	1.358
7	1.416	1.745	1.612	1.365	1.494	1.446	1.415	1.559	1.398
8	1.258	1.446	1.273	1.282	1.472	1.288	1.238	1.375	1.259
9	1.325	1.578	1.238	1.324	1.598	1.240	1.287	1.498	1.253
10	1.190	1.485	1.303	1.231	1.517	1.290	1.211	1.544	1.229
11	1.225	1.434	1.331	1.195	1.519	1.224	1.238	1.410	1.283
12	1.181	1.393	1.422	1.198	1.438	1.273	1.196	1.449	1.199
13	1.308	1.522	1.392	1.146	1.584	1.327	1.199	1.557	1.266
14	1.162	1.483	1.240	1.197	1.501	1.279	1.149	1.501	1.477
15	1.169	1.516	1.237	1.160	1.545	1.202	1.188	1.441	1.223
16	1.138	1.306	1.256	1.096	1.488	1.157	1.102	1.397	1.250
17	1.147	1.354	1.321	1.162	1.470	1.231	1.117	1.501	1.295
18	1.139	1.504	1.260	1.141	1.494	1.569	1.099	1.572	1.213
19	1.164	1.475	1.244	1.159	1.561	1.531	1.129	1.558	1.272
20	1.139	1.523	1.330	1.085	1.449	1.274	1.109	1.502	1.199
30	1.083	1.374	1.299	1.120	1.553	1.349	1.100	1.492	1.431
40	1.140	1.525	1.311	1.051	1.700	1.159	1.079	1.605	1.187
50	1.091	1.579	1.291	1.053	1.452	1.250	1.076	1.437	1.174
60	1.074	1.554	1.264	1.094	1.427	1.226	1.104	1.522	1.284
70	1.085	1.679	1.239	1.032	1.476	1.109	1.090	1.453	1.254
80	1.125	1.639	1.270	1.070	1.449	1.341	1.153	1.523	1.176
90	1.101	1.506	1.280	1.058	1.556	1.243	1.065	1.520	1.323
100	1.115	1.631	1.276	1.056	1.598	1.281	1.065	1.635	1.184
	no augmentation			estimated maximum			baseline		
	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	RF		
	1.963	2.090	2.373	1.124	1.585	1.289	2.563		

Figure B.2: **Test RMSE using data augmentation on the FreeSolv data set.** The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the FreeSolv data set is augmenting the data set 70 times keeping all duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

Strategy	without duplication			with duplication			with reduced duplication		
Model	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN	CONV1D	CONV2D	RNN
0									
1	1.268	1.286	1.144	1.208	1.309	1.145	1.211	1.260	1.144
2	1.042	1.115	1.073	1.048	1.140	1.078	1.033	1.126	1.083
3	0.933	1.004	1.077	0.949	1.010	1.015	0.921	1.016	1.026
4	0.874	0.956	0.918	0.886	0.967	0.942	0.859	0.921	0.967
5	0.861	0.944	0.931	0.849	0.939	0.908	0.857	0.961	0.920
6	0.815	0.903	0.862	0.804	0.911	0.875	0.815	0.890	0.881
7	0.783	0.886	0.828	0.807	0.871	0.848	0.797	0.892	0.848
8	0.782	0.880	0.785	0.787	0.871	0.797	0.779	0.864	0.838
9	0.769	0.846	0.793	0.783	0.854	0.785	0.791	0.852	0.795
10	0.755	0.852	0.781	0.751	0.844	0.789	0.752	0.839	0.797
11	0.754	0.832	0.745	0.737	0.838	0.744	0.738	0.837	0.745
12	0.730	0.837	0.737	0.724	0.816	0.746	0.721	0.843	0.753
13	0.717	0.811	0.723	0.722	0.829	0.719	0.725	0.816	0.730
14	0.715	0.818	0.721	0.713	0.819	0.772	0.716	0.808	0.715
15	0.712	0.808	0.724	0.707	0.804	0.715	0.716	0.810	0.701
16	0.704	0.804	0.694	0.702	0.789	0.720	0.711	0.798	0.701
17	0.706	0.808	0.707	0.699	0.791	0.686	0.709	0.797	0.698
18	0.686	0.807	0.679	0.691	0.801	0.690	0.678	0.791	0.699
19	0.681	0.809	0.703	0.672	0.785	0.700	0.686	0.793	0.688
20	0.671	0.787	0.673	0.657	0.777	0.672	0.671	0.779	0.673
30	0.640	0.752	0.680	0.645	0.747	0.684	0.655	0.754	0.679
40	0.638	0.723	0.660	0.637	0.742	0.662	0.634	0.746	0.651
50	0.614	0.744	0.691	0.623	0.727	0.692	0.617	0.724	0.680
60	0.604	0.735	0.671	0.606	0.725	0.654	0.609	0.738	0.723
70	0.597	0.717	0.670	0.602	0.716	0.673	0.608	0.737	0.692
80	0.593	0.723	0.687	0.603	0.720	0.695	0.606	0.718	0.687
90	0.606	0.735	0.694	0.609	0.730	0.649	0.600	0.718	0.693
100	0.596	0.715	0.706	0.600	0.715	0.688	0.596	0.723	0.720
	no augmentation			baseline					
	CONV1D	CONV2D	RNN	RF					
	0.994	1.060	1.023	0.860					

Figure B.3: **Test RMSE using data augmentation on the lipophilicity data set.** The table shows the root mean squared error (RMSE) on the test set for three deep learning models and five SMILES augmentation strategies, using various augmentation numbers, as well as a baseline consisting of a Random Forest (RF) model with Morgan fingerprint as input. The lighter the purple color, the better the model. The overall best setting is highlighted in yellow, which for the lipophilicity data set is augmenting the data set 80 times removing duplicates and training a 1D convolutional neural network (CONV1D). For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

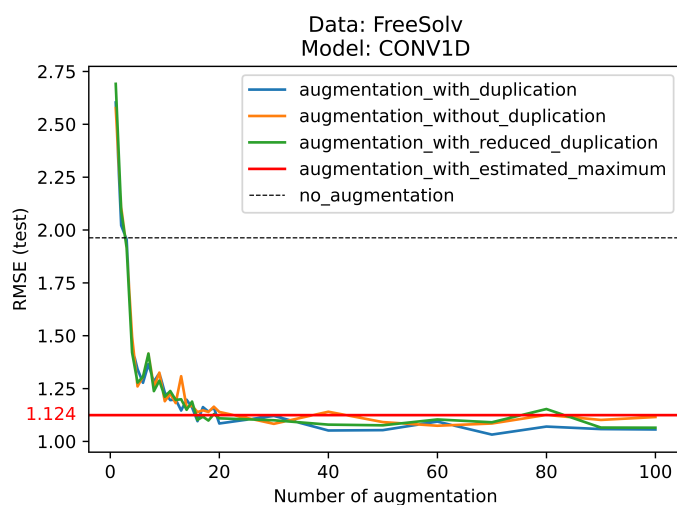


Figure B.4: **Generating a large portion of the SMILES space does not lead to the best performance.** Even though the CONV1D model is presented with SMILES variations that cover a large portion of the SMILES space using the augmentations strategy with estimated maximum, on the FreeSolv data set, this strategy does not achieve the best results.

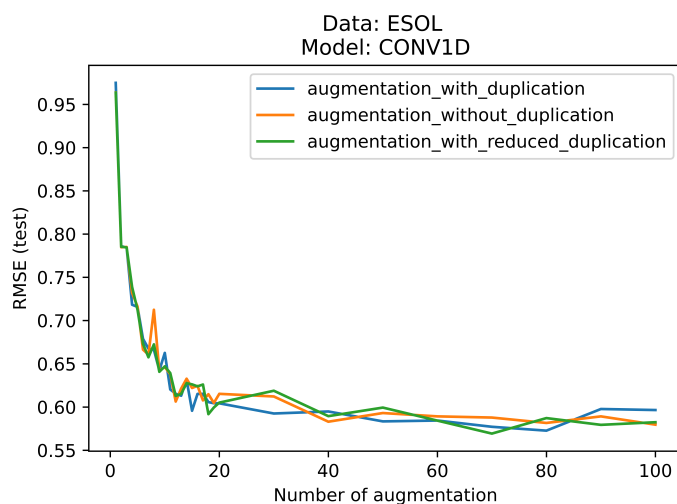


Figure B.5: **Performance reaches a plateau independently of the augmentation strategy.** The performance of the CONV1D model trained and evaluated on the ESOL data set reaches a test RMSE value slightly below 0.6 as of 40 augmentation steps and fluctuates below this value thereafter, for all augmentation strategies: with, without, and with reduced duplication.

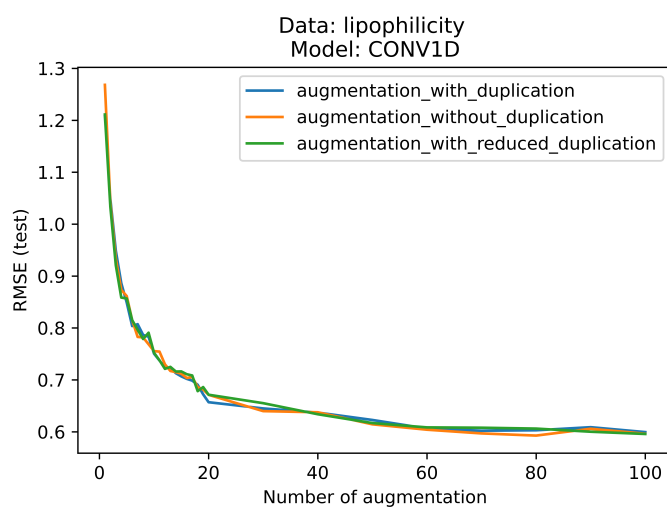


Figure B.6: **Performance reaches a plateau independently of the augmentation strategy.** The performance of the CONV1D model trained and evaluated on the lipophilicity data set reaches a test RMSE value slightly below 0.6 as of 60 augmentation steps and fluctuates below this value thereafter, for all augmentation strategies: with, without, and with reduced duplication.



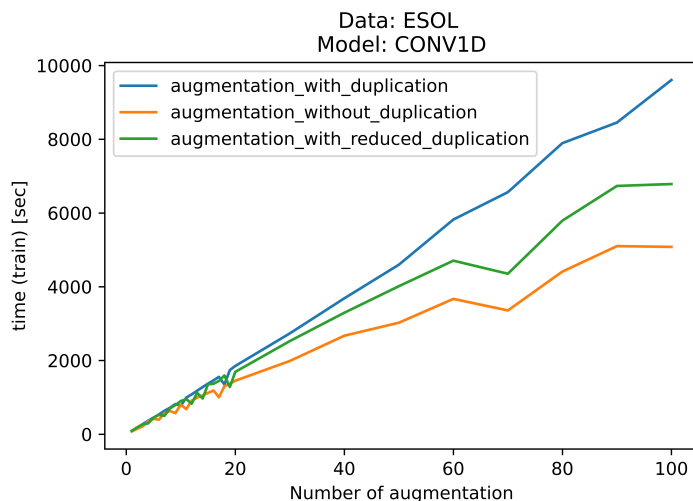


Figure B.7: **Trade-off between performance and computation time.** As expected, the training time of the CONV1D model on the ESOL data increases with the augmentation number for all augmentation strategies: with, without, and with reduced duplication. Augmenting the training set by 100 and keeping duplicate leads to 90,200 data points (see Table 3.1). Training the model on a GPU takes approximately three hours and reaches a test RMSE of 0.580 (see Figure 3.2). However, augmenting the data by just 19 leads to a test RMSE of 0.605 in less than 30 minutes.

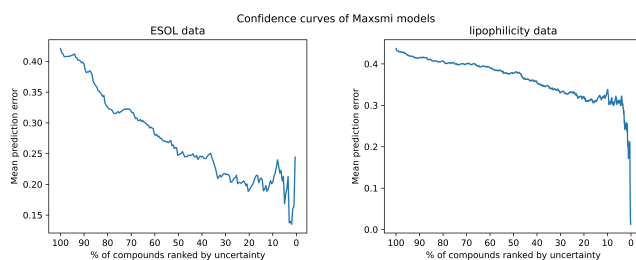


Figure B.8: **Confidence curves of the Maxsmi models on the ESOL and lipophilicity data.** The general trend of the curve in the left plot (the ESOL data) is decreasing, showing a relationship between high confidence and small mean prediction error. Although also generally decreasing, the mean prediction error in the right plot (the lipophilicity data) is still above 0.3 when only keeping the 10% of compounds with the highest confidence.

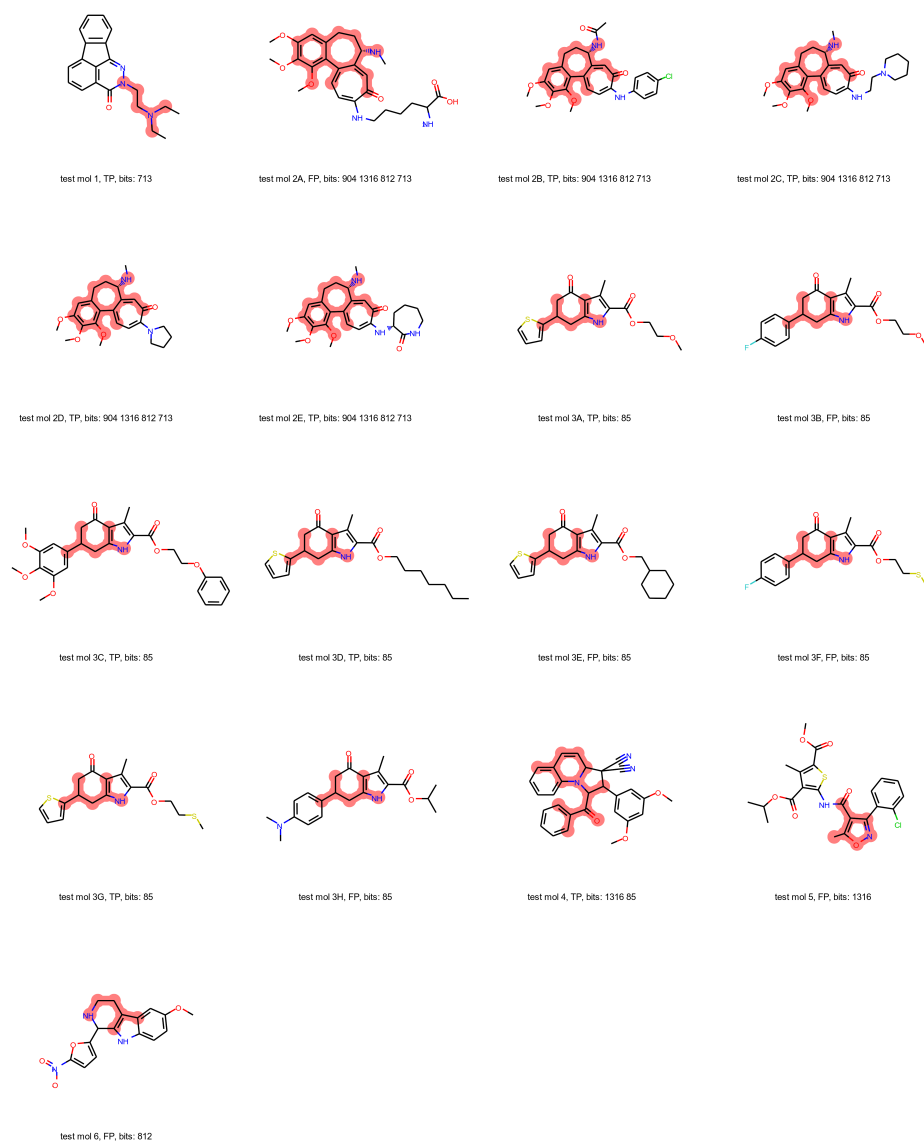


Figure B.9: The Deep Taylor Decomposition was used on a feed-forward neural network to assign a relevance score to each atom environment, which can be used to identify potential toxicophores. The five highest scores were selected and are associated to bits 85, 713, 812, 904 and 1,316. The figure shows molecules in the test set which contain at least one of the five atom environments, highlighted in red. The label for each molecule specifies its name, whether it was correctly predicted cytotoxic (TP) by the model or not (FP: False Positive) and lastly the bit(s) it contains.

## Appendix C

### Tables

Table C.1: The table shows the IDs of the decomposable molecules in the test set sorted by decreasing order with respect to the FNN model prediction probability, the true value experimentally determined (1: toxic, 0: non-toxic) and the value predicted by the model (TP: true positive, FP: false positive).

Molecule ID	FNN Score	True Value	Predicted Value
2E	0.91	1	TP
2C	0.90	1	TP
2B	0.89	1	TP
2D	0.89	1	TP
2A	0.86	0	FP
4	0.83	1	TP
3C	0.79	1	TP
3D	0.78	1	TP
3A	0.78	1	TP
1	0.75	1	TP
3G	0.72	1	TP
6	0.69	0	FP
3B	0.68	0	FP
3H	0.67	0	FP
3E	0.61	0	FP
3F	0.61	0	FP
5	0.58	0	FP

# Acronyms

**ADMET** Absorption, distribution, metabolism, excretion, and toxicity

**AI** Artificial intelligence

**ANN** Artificial neural network

**API** Application programming interface

**ATP** Adenosine triphosphate

**AUC** Area under the roc curve

**CADD** Computer-aided drug design

**CASF** Comparative assessment of scoring functions

**CI** Continuous integration

**CNF** Convolutional neural fingerprint

**CNN** Convolutional neural network

**CONV1D** 1D convolutional neural network

**CONV2D** 2D convolutional neural network

**CPU** Central processing unit

**CV** Cross-validation

**DL** Deep learning

**DUD** Directory of useful decoys

**ECFP** Extended-connectivity fingerprint

**EF** Enrichment factor

**EGFR** Epidermal growth factor receptor

**FAIR** Findable, accessible, interoperable, and reusable

**FDA** Food and drug administration

**GAN** Generative adversarial network

**GANN** Graph attention neural network

**GCNN** Graph convolution neural network

**GGNN** Gated graph neural network

**GNN** Graph neural network

**GPU** Graphics processing unit

**GRU** Gated recurrent unit

**HTS** High-throughput screening

**ID** Identifier

**IFP** Interaction fingerprint

**KIBA** Kinase inhibitor bioactivity

**KiSSim** Kinase structure similarity

**KLIFS** Kinase-ligand interaction fingerprints and structures

**LSTM** Long short-term memory

**MCS** Maximum common substructure

**ML** Machine learning

**MLP** Multilayer perceptron

**MolPMoFiT** Molecular prediction model fine-tuning

**MSE** Mean squared error

**MUV** Maximum unbiased validation

**NA** Not available

**NN** Neural network

**PCM** Proteochemometric

**PDB** Protein data bank

**QSAR** Quantitative structure-activity relationship

**RF** Random forest

**RMSD** Root mean square deviation

**RMSE** Root mean squared error

**RNN** Recurrent neural network

**ROC** Receiver operating characteristic

**SF** Scoring function

**SMILES** Simplified molecular input line entry

**SVM** Support vector machine

**TDC** Therapeutics data commons

**TPU** Tensor processing unit

**VS** Virtual screening

---

## Bibliography

- [1] Talia B. Kimber, Yonghui Chen, and Andrea Volkamer. Deep learning in virtual screening: recent applications and developments. *International Journal of Molecular Sciences*, 22(9):4435, 2021. ISSN 1422-0067. URL <https://doi.org/10.3390/ijms22094435>.
- [2] Talia B. Kimber, Maxime Gagnebin, and Andrea Volkamer. Maxsmi: Maximizing molecular property prediction performance with confidence estimation using SMILES augmentation and deep learning. *Artificial Intelligence in the Life Sciences*, 1:100014, 2021. ISSN 2667-3185. URL <https://doi.org/10.1016/j.ailsci.2021.100014>.
- [3] Henry E. Weibel, Talia B. Kimber, Silke Radetzki, Martin Neuenchwander, Marc Nazaré, and Andrea Volkamer. Revealing cytotoxic substructures in molecules using deep learning. *Journal of Computer-Aided Molecular Design*, 34(7):731–746, 2020. URL <https://doi.org/10.1007/s10822-020-00310-4>.
- [4] Dominique Sydow, Jaime Rodríguez-Guerra, Talia B. Kimber, David Schaller, Corey J. Taylor, Yonghui Chen, Mareike Leja, Sakshi Misra, Michele Wichmann, Armin Ariamajd, and Andrea Volkamer. TeachOpenCADD 2022: open source and FAIR Python pipelines to assist in structural bioinformatics and cheminformatics research. *Nucleic Acids Research*, 50(W1):W753–W760, 05 2022. ISSN 0305-1048. URL <https://doi.org/10.1093/nar/gkac267>.
- [5] Talia B. Kimber, Dominique Sydow, and Andrea Volkamer. Kinase Similarity Assessment Pipeline for Off-Target Prediction [Article v1.0]. *Living Journal of Computational Molecular Science*, 3(1):1599, Jun 2022. URL <https://doi.org/10.33011/livecoms.3.1.1599>.
- [6] Amazon. <https://www.amazon.com/>, 2022. [Online; accessed 01-May-2022].
- [7] Netflix, Inc. <http://netflix.com/>, 2022. [Online; accessed 01-May-2022].
- [8] Margaret A Boden. *Artificial intelligence*. Elsevier, 1996.
- [9] Yuval Noah Harari. Reboot for the AI revolution. *Nature*, 550(7676):324–327, oct 2017. doi: 10.1038/550324a. URL <https://doi.org/10.1038/550324a>.
- [10] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates,



- Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstern, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2.
- [11] Ken A. Dill, S. Banu Ozkan, M. Scott Shell, and Thomas R. Weikl. The protein folding problem. *Annual Review of Biophysics*, 37(1):289–316, 2008. doi: 10.1146/annurev.biophys.37.092707.153558. PMID: 18573083.
- [12] Lisa N. Kinch, R. Dustin Schaeffer, Andriy Kryshtafovych, and Nick V. Grishin. Target classification in the 14th round of the critical assessment of protein structure prediction (casp14). *Proteins: Structure, Function, and Bioinformatics*, 89(12):1618–1632, 2021. doi: <https://doi.org/10.1002/prot.26202>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26202>.
- [13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [14] Taiwo Oladipupo. *Types of Machine Learning Algorithms*. IntechOpen, 2010. doi: 10.5772/9385.
- [15] Frank Rosenblatt. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961. URL <https://apps.dtic.mil/dtic/tr/fulltext/u2/256582.pdf>.
- [16] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.
- [17] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.
- [18] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.
- [19] Nvidia. <https://www.nvidia.com/>, 2022. [Online; accessed 01-May-2022].
- [20] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean,

Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.

- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, may 2017. doi: 10.1145/3065386. URL <https://doi.org/10.1145/3065386>.
- [22] Shaohuai Shi, Qiang Wang, Pengfei Xu, and Xiaowen Chu. Benchmarking state-of-the-art deep learning software tools, 2016. URL <https://arxiv.org/abs/1608.07249>.
- [23] Google Research. Google Colab. <https://colab.research.google.com/>, 2021. URL <https://colab.research.google.com/>. [Online; accessed 2021-03-17].
- [24] LaCie. <https://www.lacie.com/>, 2022. [Online; accessed 01-May-2022].
- [25] Amazon Web Services (AWS). <https://aws.amazon.com/products/storage/>, 2022. [Online; accessed 01-May-2022].
- [26] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL <http://jmlr.org/papers/v12/pedregosa11a.html>.
- [28] François Chollet et al. Keras. <https://keras.io>, 2015.
- [29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.

- Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [30] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020. URL <https://developer.nvidia.com/cuda-toolkit>.
- [31] Horace He. The state of machine learning frameworks in 2019. *The Gradient*, 2019.
- [32] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.
- [33] Kaggle. <https://www.kaggle.com/>, 2021. [Online; accessed 27-August-2021].
- [34] Hugging Face. <https://github.com/huggingface>, 2022. URL <https://huggingface.co/>. [Online; accessed 01-May-2022].
- [35] Robin Duelen, Marlies Corvelyn, Ilaria Tortorella, Leonardo Leonardi, Yoke Chin Chai, and Maurilio Sampaolesi. *Medicinal Biotechnology for Disease Modeling, Clinical Therapy, and Drug Discovery and Development*, pages 89–128. Springer International Publishing, Cham, 2019. ISBN 978-3-030-22141-6. doi: 10.1007/978-3-030-22141-6\_5. URL [https://doi.org/10.1007/978-3-030-22141-6\\_5](https://doi.org/10.1007/978-3-030-22141-6_5).
- [36] JP Hughes, S Rees, SB Kalindjian, and KL Philpott. Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239–1249, 2011. doi: <https://doi.org/10.1111/j.1476-5381.2010.01127.x>. URL <https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/j.1476-5381.2010.01127.x>.
- [37] Conrad Stork, Neann Mathai, and Johannes Kirchmair. Computational prediction of frequent hitters in target-based and cell-based assays. *Artificial Intelligence in the Life Sciences*, 1:100007, 2021. ISSN 2667-3185. doi: <https://doi.org/10.1016/j.aillsi.2021.100007>. URL

<https://www.sciencedirect.com/science/article/pii/S2667318521000076>.

- [38] Longfei Guan, Hongbin Yang, Yingchun Cai, Lixia Sun, Peiwen Di, Weihua Li, Guixia Liu, and Yun Tang. Admet-score - a comprehensive scoring function for evaluation of chemical drug-likeness. *Med. Chem. Commun.*, 10:148–157, 2019. doi: 10.1039/C8MD00472B. URL <http://dx.doi.org/10.1039/C8MD00472B>.
- [39] Michael Schlander, Karla Hernandez-Villafuerte, Chih-Yuan Cheng, Jorge Mestre-Ferrandiz, and Michael Baumann. How much does it cost to research and develop a new drug? a systematic review and assessment. *PharmacoEconomics*, 39(11):1243–1269, 2021. doi: 10.1007/s40273-021-01065-y. URL <https://doi.org/10.1007/s40273-021-01065-y>.
- [40] Nathan Brown. *In Silico Medicinal Chemistry*. Theoretical and Computational Chemistry Series. The Royal Society of Chemistry, 2016. ISBN 978-1-78262-163-8. doi: 10.1039/9781782622604. URL <http://dx.doi.org/10.1039/9781782622604>.
- [41] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL <https://doi.org/10.1093/nar/28.1.235>.
- [42] Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V Crichlow, Cole H Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M Duarte, Shuchismita Dutta, Zukang Feng, Sai Ganesan, David S Goodsell, Sutapa Ghosh, Rachel Kramer Green, Vladimir Guranović, Dmytro Guzenko, Brian P Hudson, Catherine L Lawson, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Persikova, Chris Randle, Alexander Rose, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Yi-Ping Tao, Maria Voigt, John D Westbrook, Jasmine Y Young, Christine Zardecki, and Marina Zhuravleva. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*, 49(D1):D437–D451, 2020. doi: 10.1093/nar/gkaa1038.
- [43] Kurt Wüthrich. The way to nmr structures of proteins. *Nature Structural Biology*, 8(11):923–925, nov 2001. doi: 10.1038/nsb1101-923. URL <https://doi.org/10.1038/nsb1101-923>.

- 
- [44] Xiao chen Bai, Greg McMullan, and Sjors H.W Scheres. How cryo-EM is revolutionizing structural biology. *Trends in Biochemical Sciences*, 40(1):49–57, 2015. ISSN 0968-0004. doi: <https://doi.org/10.1016/j.tibs.2014.10.005>. URL <https://www.sciencedirect.com/science/article/pii/S096800041400187X>.
- [45] Marc A. Martí-Renom, Ashley C. Stuart, András Fiser, Roberto Sánchez, Francisco Melo, and Andrej Šali. Comparative protein structure modeling of genes and genomes. *Annual Review of Biophysics and Biomolecular Structure*, 29(1):291–325, 2000. doi: 10.1146/annurev.biophys.29.1.291. URL <https://doi.org/10.1146/annurev.biophys.29.1.291>.
- [46] Jürgen Bajorath. Integration of virtual and high-throughput screening. *Nature Reviews Drug Discovery*, 1(11):882–894, 2002. doi: 10.1038/nrd941. URL <https://doi.org/10.1038/nrd941>.
- [47] Alexander Tropsha. Best practices for QSAR model development, validation, and exploitation. *Molecular Informatics*, 29(6-7):476–488, 2010. doi: 10.1002/minf.201000061.
- [48] Matthew A Sellwood, Mohamed Ahmed, Marwin HS Segler, and Nathan Brown. Artificial intelligence in drug discovery. *Future Medicinal Chemistry*, 10(17):2025–2028, 2018. doi: 10.4155/fmc-2018-0212. URL <https://doi.org/10.4155/fmc-2018-0212>.
- [49] Xuan-Yu Meng, Hong-Xing Zhang, Mihaly Mezei, and Meng Cui. Molecular docking: A powerful approach for structure-based drug discovery. *Current Computer Aided-Drug Design*, 7(2):146–157, 2011. doi: 10.2174/157340911795677602. URL <https://doi.org/10.2174/157340911795677602>.
- [50] Jennifer Hemmerich and Gerhard F. Ecker. In silico toxicology: From structure-activity relationships towards deep learning and adverse outcome pathways. *WIREs Computational Molecular Science*, 10(4):e1475, 2020. doi: <https://doi.org/10.1002/wcms.1475>. URL <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1475>.
- [51] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malcolli, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. doi: 10.1007/s11263-020-01316-z.
- [52] John S. Delaney. Esol: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 44(3):1000–1005, 2004. doi: 10.1021/ci034243x.

- 
- [53] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. ISBN 9781492039839.
- [54] Anna Gaulton, Anne Hersey, Michał Nowotka, A. Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J. Bellis, Elena Cibrián-Uhalte, Mark Davies, Nathan Dedman, Anneli Karlsson, María Paula Magariños, John P. Overington, George Papadatos, Ines Smit, and Andrew R. Leach. The ChEMBL database in 2017. *Nucleic Acids Research*, 45(D1):D945–D954, nov 2016. doi: 10.1093/nar/gkw1074. URL <https://doi.org/10.1093/nar/gkw1074>.
- [55] ChEMBL. <https://www.ebi.ac.uk/chembl/>, 2022. [Online; accessed 01-May-2022].
- [56] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K. Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science*, 2018. doi: <https://doi.org/10.1039/c8sc00148k>.
- [57] Tuomo Kallioikoski, Christian Kramer, Anna Vulpetti, and Peter Gedeck. Comparability of mixed ic50 data - a statistical analysis. *PLOS ONE*, 8(4):1–12, 04 2013. doi: 10.1371/journal.pone.0061007. URL <https://doi.org/10.1371/journal.pone.0061007>.
- [58] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.
- [59] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x. URL <https://doi.org/10.1038/s42256-019-0048-x>.
- [60] Cynthia Rudin and Berk Ustun. Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces*, 48(5):449–466, 2018. doi: 10.1287/inte.2018.0957. URL <https://doi.org/10.1287/inte.2018.0957>.
- [61] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard

- Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3(1), mar 2016. doi: 10.1038/sdata.2016.18. URL <https://doi.org/10.1038/sdata.2016.18>.
- [62] Max Roser and Hannah Ritchie. Cancer. *Our World in Data*, 2015. <https://ourworldindata.org/cancer>.
- [63] M.A. Bareschino, C. Schettino, T. Troiani, E. Martinelli, F. Morgillo, and F. Ciardiello. Erlotinib in cancer treatment. *Annals of Oncology*, 18:vi35–vi41, 2007. ISSN 0923-7534. doi: <https://doi.org/10.1093/annonc/mdm222>. New Trends in Clinical Oncology - 9th National GOIM Congress 17-19 June 2007, Potenza, Italy.
- [64] Priyadeep Bhutani, Gaurav Joshi, Nivethitha Raja, Namrata Bachhav, Prabhakar K. Rajanna, Hemant Bhutani, Atish T. Paul, and Raj Kumar. U.s. fda approved drugs from 2015-june 2020: A perspective. *Journal of Medicinal Chemistry*, 64(5):2339–2381, 2021. doi: 10.1021/acs.jmedchem.0c01786. PMID: 33617716.
- [65] Amanda B. Methvin and Roberta E. Gausas. Newly recognized ocular side effects of erlotinib. *Ophthalmic Plastic and Reconstructive Surgery*, 23(1):63–65, 2007. doi: 10.1097/iop.0b013e31802d97f0.
- [66] Neil Vasani, José Baselga, and David M. Hyman. A view on drug resistance in cancer. *Nature*, 575(7782):299–309, 2019. doi: 10.1038/s41586-019-1730-1. URL <https://doi.org/10.1038/s41586-019-1730-1>.
- [67] Khushwant S. Bhullar, Naiara Orrego Lagarón, Eileen M. McGowan, Indu Parmar, Amitabh Jha, Basil P. Hubbard, and H. P. Vasantha Rupasinghe. Kinase-targeted cancer therapies: progress, challenges and future directions. *Molecular Cancer*, 17(1), feb 2018. doi: 10.1186/s12943-018-0804-2. URL <https://doi.org/10.1186/s12943-018-0804-2>.
- [68] Philip Cohen. Protein kinases — the major drug targets of the twenty-first century? *Nature Reviews Drug Discovery*, 1(4):309–315, apr 2002. doi: 10.1038/nrd773. URL <https://doi.org/10.1038/nrd773>.

- [69] Albert J. Kooistra and Andrea Volkamer. Kinase-centric computational drug development. In *Annual Reports in Medicinal Chemistry*, pages 197–236. Elsevier, 2017. doi: 10.1016/bs.armc.2017.08.001. URL <https://doi.org/10.1016/bs.armc.2017.08.001>.
- [70] J. Stamos, M.X. Sliwkowski, and C. Eigenbrot. Epidermal growth factor receptor tyrosine kinase domain with 4-anilinoquinazoline inhibitor erlotinib, sep 2002. URL <https://doi.org/10.2210/pdb1m17/pdb>.
- [71] Jennifer Stamos, Mark X. Sliwkowski, and Charles Eigenbrot. Structure of the epidermal growth factor receptor kinase domain alone and in complex with a 4-anilinoquinazoline inhibitor. *Journal of Biological Chemistry*, 277(48):46265–46272, nov 2002. doi: 10.1074/jbc.m207135200. URL <https://doi.org/10.1074/jbc.m207135200>.
- [72] KLIFS. <https://klifs.net/>, 2022. [Online; accessed 01-February-2022].
- [73] Solomon Tadesse, Abel T. Anshabo, Neil Portman, Elgene Lim, Wayne Tilley, C. Elizabeth Caldon, and Shudong Wang. Targeting cdk2 in cancer: challenges and opportunities for therapy. *Drug Discovery Today*, 25(2):406–413, 2020. ISSN 1359-6446. doi: <https://doi.org/10.1016/j.drudis.2019.12.001>. URL <https://www.sciencedirect.com/science/article/pii/S135964461930460X>.
- [74] David H. Drewry, Carrow I. Wells, David M. Andrews, Richard Angell, Hassan Al-Ali, Alison D. Axtman, Stephen J. Capuzzi, Jonathan M. Elkins, Peter Ettmayer, Mathias Frederiksen, Opher Gileadi, Nathanael Gray, Alice Hooper, Stefan Knapp, Stefan Laufer, Ulrich Luecking, Michael Michaelides, Susanne Müller, Eugene Muratov, R. Aldrin Denny, Kumar S. Saikatendu, Daniel K. Treiber, William J. Zuercher, and Timothy M. Willson. Progress towards a public chemogenomic set for protein kinases and a call for contributions. *PLOS ONE*, 12(8):1–20, 08 2017. doi: 10.1371/journal.pone.0181585. URL <https://doi.org/10.1371/journal.pone.0181585>.
- [75] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. Librilight: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673, 2020. doi: 10.1109/ICASSP40776.2020.9052942.
- [76] Zhongqiu Liu, Stephen Wang, and Ming Hu. Chapter 11 - oral absorption basics: Pathways, physico-chemical and biological factors affecting



- absorption. In Yihong Qiu, Yisheng Chen, Geoff G.Z. Zhang, Lirong Liu, and William R. Porter, editors, *Developing Solid Oral Dosage Forms*, pages 263–288. Academic Press, San Diego, 2009. ISBN 978-0-444-53242-8. doi: <https://doi.org/10.1016/B978-0-444-53242-8.00011-4>.
- [77] David L. Mobley and J. Peter Guthrie. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design*, 28(7):711–720, 2014. doi: [10.1007/s10822-014-9747-x](https://doi.org/10.1007/s10822-014-9747-x).
- [78] Nicola Normanno, Antonella De Luca, Caterina Bianco, Luigi Strizzi, Mario Mancino, Monica R. Maiello, Adele Carotenuto, Gianfranco De Feo, Francesco Caponigro, and David S. Salomon. Epidermal growth factor receptor (egfr) signaling in cancer. *Gene*, 366(1):2–16, 2006. ISSN 0378-1119. doi: <https://doi.org/10.1016/j.gene.2005.10.018>.
- [79] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, 65:211–222, may 2017. doi: <https://doi.org/10.1016/j.patcog.2016.11.008>.
- [80] Dominique Sydow, Andrea Morger, Maximilian Driller, and Andrea Volkamer. TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data. *Journal of Cheminformatics*, 11(1), apr 2019. doi: [10.1186/s13321-019-0351-x](https://doi.org/10.1186/s13321-019-0351-x). URL <https://doi.org/10.1186/s13321-019-0351-x>.
- [81] Nurken Berdigaliyev and Mohamad Aljofan. An overview of drug discovery and development. *Future Medicinal Chemistry*, 12(10):939–947, 2020. doi: [10.4155/fmc-2019-0307](https://doi.org/10.4155/fmc-2019-0307).
- [82] M Butkiewicz, Y Wang, SH Bryant, EW Lowe Jr, DC Weaver, and J Meiler. High-throughput screening assay datasets from the pubchem database. *Chemical Informatics (Wilmington, Del.)*, 3(1), 2017. PMID: 29795804.
- [83] W.Patrick Walters, Matthew T Stahl, and Mark A Murcko. Virtual screening—an overview. *Drug Discovery Today*, 3(4):160–178, 1998. doi: [10.1016/S1359-6446\(97\)01163-X](https://doi.org/10.1016/S1359-6446(97)01163-X).
- [84] Teague Sterling and John J. Irwin. Zinc 15–ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. doi: [10.1021/acs.jcim.5b00559](https://doi.org/10.1021/acs.jcim.5b00559).
- [85] MolPORT. <https://www.molport.com>, 2021. [Online; accessed 02-March-2021].

- 
- [86] Enamine REAL. <https://enamine.net/library-synthesis/real-compounds>, 2021. [Online; accessed 02-March-2021].
- [87] Thomas Scior, Andreas Bender, Gary Tresadern, José L. Medina-Franco, Karina Martínez-Mayorga, Thierry Langer, Karina Cuanaló-Contreras, and Dimitris K. Agrafiotis. Recognizing pitfalls in virtual screening: A critical review. *Journal of Chemical Information and Modeling*, 52(4):867–881, 2012. doi: 10.1021/ci200528d.
- [88] Ashutosh Kumar and Kam Y.J. Zhang. Hierarchical virtual screening approaches in small molecule drug discovery. *Methods*, 71:26–37, 2015. doi: 10.1016/j.ymeth.2014.07.007.
- [89] Natasja Brooijmans and Irwin D. Kuntz. Molecular recognition and docking algorithms. *Annual Review of Biophysics and Biomolecular Structure*, 32(1):335–373, 2003. doi: 10.1146/annurev.biophys.32.110601.142532.
- [90] Vladimir B. Sulimov, Danil C. Kutov, and Alexey V. Sulimov. Advances in docking. *Current Medicinal Chemistry*, 26(42):7555–7580, jan 2020. doi: 10.2174/0929867325666180904115000. URL <https://doi.org/10.2174/0929867325666180904115000>.
- [91] André Fischer, Martin Smieško, Manuel Sellner, and Markus A. Lill. Decision making in structure-based drug discovery: Visual inspection of docking results. *Journal of Medicinal Chemistry*, 64(5):2489–2500, 2021. doi: 10.1021/acs.jmedchem.0c02227. URL <https://doi.org/10.1021/acs.jmedchem.0c02227>. PMID: 33617246.
- [92] Gerhard Klebe. Virtual ligand screening: strategies, perspectives and limitations. *Drug Discovery Today*, 11(13):580–594, 2006. ISSN 1359-6446. doi: 10.1016/j.drudis.2006.05.012. URL <https://www.sciencedirect.com/science/article/pii/S1359644606001784>.
- [93] Adrian Kolodzik, Nadine Schneider, and Matthias Rarey. *Structure-Based Virtual Screening*, chapter 6.8, pages 313–331. John Wiley & Sons, Ltd, 2018. ISBN 9783527806539. doi: 10.1002/9783527806539.ch6h. URL <https://onlineibrary.wiley.com/doi/abs/10.1002/9783527806539.ch6h>.
- [94] Nataraj S. Pagadala, Khajamohiddin Syed, and Jack Tuszynski. Software for molecular docking: a review. *Biophysical Reviews*, 9(2):91–102, 2017. doi: 10.1007/s12551-016-0247-1.
- [95] Jin Li, Ailing Fu, and Le Zhang. An overview of scoring functions used for protein–ligand interactions in molecular docking. *Interdisciplinary Sciences: Computational Life Sciences*, 11(2):320–328, 2019. doi: 10.1007/s12539-019-00327-w.

- 
- [96] Chao Shen, Junjie Ding, Zhe Wang, Dongsheng Cao, Xiaoqin Ding, and Tingjun Hou. From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *WIREs Computational Molecular Science*, 10(1), 2019. doi: 10.1002/wcms.1429.
- [97] Qurrat Ul Ain, Antoniya Aleksandrova, Florian D. Roessler, and Pedro J. Ballester. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(6): 405–424, 2015. doi: 10.1002/wcms.1225. URL <https://doi.org/10.1002/wcms.1225>.
- [98] Jocelyn Sunseri and David Ryan Koes. Pharmit: interactive exploration of chemical space. *Nucleic Acids Research*, 44(W1):W442–W448, 04 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw287. URL <https://doi.org/10.1093/nar/gkw287>.
- [99] David Schaller, Dora Šribar, Theresa Noonan, Lihua Deng, Trung Ngoc Nguyen, Szymon Pach, David Machalz, Marcel Bermudez, and Gerhard Wolber. Next generation 3d pharmacophore modeling. *WIREs Computational Molecular Science*, 10(4):e1468, 2020. doi: 10.1002/wcms.1468. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1468>.
- [100] Dominique Sydow, Lindsey Burggraaff, Angelika Szengel, Herman W. T. van Vlijmen, Adriaan P. IJzerman, Gerard J. P. van Westen, and Andrea Volkamer. Advances and challenges in computational target prediction. *Journal of Chemical Information and Modeling*, 59(5): 1728–1742, 2019. doi: 10.1021/acs.jcim.8b00832.
- [101] Maris Lapinsh, Peteris Prusis, Alexandrs Gutcaits, Torbjörn Lundstedt, and Jarl E.S. Wikberg. Development of proteo-chemometrics: a novel technology for the analysis of drug-receptor interactions. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1525(1):180–190, 2001. ISSN 0304-4165. doi: 10.1016/S0304-4165(00)00187-2.
- [102] Gerard J. P. van Westen, Jörg K. Wegner, Adriaan P. IJzerman, Herman W. T. van Vlijmen, and A. Bender. Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets. *Med. Chem. Commun.*, 2:16–30, 2011. doi: 10.1039/C0MD00165A.
- [103] Hanna Geppert, Jens Humrich, Dagmar Stumpfe, Thomas Gärtner, and Jürgen Bajorath. Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors. *Journal of Chemical Information and Modeling*, 49(4):767–779, 2009. doi: 10.1021/ci900004a. PMID: 19309114.

- 
- [104] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9:513–530, 2018. doi: 10.1039/C7SC02664A.
- [105] Ekaba Bisong. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA, 2019. ISBN 978-1-4842-4470-8. doi: 10.1007/978-1-4842-4470-8\_7.
- [106] Yann LeCun and Corinna Cortes. MNIST handwritten digit database, 2010. URL <http://yann.lecun.com/exdb/mnist/>. <http://yann.lecun.com/exdb/mnist/>.
- [107] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, 11 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1075.
- [108] ChEMBL. <https://www.ebi.ac.uk/chembl/>, 2021. [Online; accessed 02-March-2021].
- [109] RCSB PDB. <http://www.rcsb.org/stats/growth/growth-release-d-structures>, 2021. [Online; accessed 02-March-2021].
- [110] Helen M. Berman, Brinda Vallat, and Catherine L. Lawson. The data universe of structural biology. *IUCrJ*, 7(4):630–638, Jul 2020. doi: 10.1107/S205225252000562X.
- [111] John R. Helliwell. New developments in crystallography: exploring its technology, methods and scope in the molecular biosciences. *Bioscience Reports*, 37(4), 07 2017. ISSN 0144-8463. doi: 10.1042/BSR20170204.
- [112] Ajay, W. Patrick Walters, and Mark A. Murcko. Can we learn to distinguish between “drug-like” and “nondrug-like” molecules? *Journal of Medicinal Chemistry*, 41(18):3314–3324, 1998. doi: 10.1021/jm970666c.
- [113] Frank R. Burden and David A. Winkler. Robust QSAR models using bayesian regularized neural networks. *Journal of Medicinal Chemistry*, 42(16):3183–3187, 1999. doi: 10.1021/jm980697n.
- [114] Frank R. Burden, Martyn G. Ford, David C. Whitley, and David A. Winkler. Use of automatic relevance determination in QSAR studies

- 
- using bayesian neural networks. *Journal of Chemical Information and Computer Sciences*, 40(6):1423–1430, 2000. doi: 10.1021/ci000450a.
- [115] Igor I. Baskin, David Winkler, and Igor V. Tetko. A renaissance of neural networks in drug discovery. *Expert Opinion on Drug Discovery*, 11(8):785–795, 2016. doi: 10.1080/17460441.2016.1201262.
- [116] Kristy A Carpenter, David S Cohen, Juliet T Jarrell, and Xudong Huang. Deep learning and virtual drug screening. *Future Medicinal Chemistry*, 10(21):2557–2567, 2018. doi: 10.4155/fmc-2018-0314.
- [117] Sally R. Ellingson, Brian Davis, and Jonathan Allen. Machine learning and ligand binding predictions: A review of data, methods, and obstacles. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1864(6):129545, 2020. doi: 10.1016/j.bbagen.2020.129545.
- [118] Sofia D’Souza, K.V. Prema, and Seetharaman Balaji. Machine learning models for drug–target interactions: current knowledge and future directions. *Drug Discovery Today*, 25(4):748–756, 2020. doi: 10.1016/j.drudis.2020.03.003.
- [119] Hongjian Li, Kam-Heung Sze, Gang Lu, and Pedro J. Ballester. Machine-learning scoring functions for structure-based drug lead optimization. *WIREs Computational Molecular Science*, 10(5), feb 2020. doi: 10.1002/wcms.1465. URL <https://doi.org/10.1002/wcms.1465>.
- [120] Hongjian Li, Kam-Heung Sze, Gang Lu, and Pedro J. Ballester. Machine-learning scoring functions for structure-based virtual screening. *WIREs Computational Molecular Science*, 11(1), apr 2020. doi: 10.1002/wcms.1478. URL <https://doi.org/10.1002/wcms.1478>.
- [121] Ahmet Sureyya Rifaioglu, Heval Atas, Maria Jesus Martin, Rengul Cetin-Atalay, Volkan Atalay, and Tunca Doğan. Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Briefings in Bioinformatics*, 20(5):1878–1912, 2018. doi: 10.1093/bib/bby061.
- [122] Yu-Chen Lo, Stefano E. Rensi, Wen Torng, and Russ B. Altman. Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today*, 23(8):1538–1546, 2018. ISSN 1359-6446. doi: 10.1016/j.drudis.2018.05.010.
- [123] Yuting Xu, Deeptak Verma, Robert P. Sheridan, Andy Liaw, Junshui Ma, Nicholas M. Marshall, John McIntosh, Edward C. Sherer, Vladimir Svetnik, and Jennifer M. Johnston. Deep dive into machine

- learning models for protein engineering. *Journal of Chemical Information and Modeling*, 60(6):2773–2790, 2020. doi: 10.1021/acs.jcim.0c00073.
- [124] Jennifer E. Bond, George Kokosis, Licheng Ren, M. Angelica Selim, Andre Bergeron, and Howard Levinson. Wound contraction is attenuated by fasudil inhibition of rho-associated kinase. *Plastic and Reconstructive Surgery*, 128(5):438e–450e, 2011. doi: 10.1097/prs.0b013e31822b7352.
- [125] Fabrice Carles, Stéphane Bourg, Christophe Meyer, and Pascal Bonnet. Pkidb: A curated, annotated and updated database of protein kinase inhibitors in clinical trials. *Molecules*, 23(4), 2018. ISSN 1420-3049. doi: 10.3390/molecules23040908.
- [126] Wen Torng and Russ B. Altman. Graph convolutional neural networks for predicting drug-target interactions. *Journal of Chemical Information and Modeling*, 59(10):4131–4149, 2019. doi: 10.1021/acs.jcim.9b00628.
- [127] Esben Jannik Bjerrum. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *CoRR*, abs/1703.07076, 2017. URL <http://arxiv.org/abs/1703.07076>.
- [128] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 09 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty593.
- [129] Talia B. Kimber, Sebastian Engelke, Igor V. Tetko, Eric Bruno, and Guillaume Godin. Synergy effect between convolutional neural networks and the multiplicity of SMILES for improvement of molecular prediction. *CoRR*, abs/1812.04439, 2018. URL <http://arxiv.org/abs/1812.04439>.
- [130] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.
- [131] RDKit. RDKit: Open-Source Cheminformatics, 2022. URL <http://www.rdkit.org>. [Online; accessed 2022-02-02].
- [132] Moises Hassan, Robert D. Brown, Shikha Varma-O’Brien, and David Rogers. Cheminformatics analysis and learning in a data pipelining environment. *Molecular Diversity*, 10(3):283–299, 2006. doi: 10.1007/s11030-006-9041-5.

- 
- [133] Indra Kundu, Goutam Paul, and Raj Banerjee. A machine learning approach towards the prediction of protein-ligand binding affinity based on fundamental molecular properties. *RSC Advances*, 8(22):12127–12137, 2018. doi: 10.1039/c8ra00003d.
- [134] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, 2002. doi: 10.1021/ci010132r.
- [135] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.*, 10:1692–1701, 2019. doi: 10.1039/C8SC04175J.
- [136] Ahmet Sureyya Rifaioglu, Esra Nalbat, Volkan Atalay, Maria Jesus Martin, Rengul Cetin-Atalay, and Tunca Doğan. Deepscreen: high performance drug-target interaction prediction with convolutional neural networks using 2-d structural compound representations. *Chem. Sci.*, 11:2531–2557, 2020. doi: 10.1039/C9SC03414E.
- [137] Robert K. Murray, David A. Bender, Kathleen M. Botham, Peter J. Kennelly, Victor W. Rodwell, and P. Anthony Weil. *Harper’s illustrated biochemistry, Twenty-Eighth Edition*. McGraw-Hill Medical McGraw-Hill distributor, New York, USA, 2009. ISBN 978-0-07-170197-6.
- [138] Frieda A Sorgenfrei, Simone Fulle, and Benjamin Merget. Kinome-wide profiling prediction of small molecules. *ChemMedChem*, 13(6):495–499, 2018. doi: 10.1002/cmdc.201700180.
- [139] Sven Hellberg, Michael Sjoestroem, Bert Skagerber, and Svante Wold. Peptide quantitative structure-activity relationships, multivariate approach. *Journal of Medicinal Chemistry*, 30(7):1126–1135, 1987. doi: 10.1021/jm00390a003.
- [140] Christian J. A. Sigrist, Edouard de Castro, Lorenzo Cerutti, Béatrice A. Cuche, Nicolas Hulo, Ala Bridge, Lydie Bougueleret, and Ioannis Xenarios. New and continuing developments at PROSITE. *Nucleic Acids Research*, 41(D1):D344–D347, 2012. doi: 10.1093/nar/gks1067.
- [141] Robert D. Finn, Alex Bateman, Jody Clements, Penelope Coghill, Ruth Y. Eberhardt, Sean R. Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, Erik L. L. Sonnhammer, John Tate, and Marco Punta. Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1223.

- 
- [142] Mostafa Karimi, Di Wu, Zhangyang Wang, and Yang Shen. DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, 35(18):3329–3338, 02 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz111.
- [143] Christophe N. Magnan and Pierre Baldi. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, 30(18):2592–2597, 05 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu352.
- [144] Renato Ferreira de Freitas and Matthieu Schapira. A systematic analysis of atomic protein–ligand interactions in the PDB. *MedChemComm*, 8(10):1970–1981, 2017. doi: 10.1039/c7md00381a.
- [145] Zhan Deng, Claudio Chuaqui, and Juswinder Singh. Structural interaction fingerprint (SIFt): a novel method for analyzing three-dimensional protein-ligand binding interactions. *Journal of Medicinal Chemistry*, 47(2):337–344, 2004. doi: 10.1021/jm030331x.
- [146] Muhammad Radifar, , Nunung Yuniarti, and Enade Perdana Istyastono. PyPLIF: Python-based protein-ligand interaction fingerprinting. *Bioinformation*, 9(6):325–328, 2013. doi: 10.6026/97320630009325.
- [147] Franck DaSilva, Jeremy Desaphy, and Didier Rognan. IChem: A versatile toolkit for detecting, comparing, and predicting protein-ligand interactions. *ChemMedChem*, 13(6):507–510, 2017. doi: 10.1002/cm dc.201700505.
- [148] Julia B. Jasper, Lina Humbeck, Tobias Brinkjost, and Oliver Koch. A novel interaction fingerprint derived from per atom score contributions: exhaustive evaluation of interaction fingerprint performance in docking based virtual screening. *Journal of Cheminformatics*, 10(1), 2018. doi: 10.1186/s13321-018-0264-0.
- [149] Marcel L. Verdonk, Jason C. Cole, Michael J. Hartshorn, Christopher W. Murray, and Richard D. Taylor. Improved protein-ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, aug 2003. doi: 10.1002/prot.10465. URL <https://doi.org/10.1002/prot.10465>.
- [150] Vladimir Chupakhin, Gilles Marcou, Helena Gaspar, and Alexandre Varnek. Simple ligand–receptor interaction descriptor (SILIRID) for alignment-free binding site comparison. *Computational and Structural*



- 
- Biotechnology Journal*, 10(16):33–37, 2014. doi: 10.1016/j.csbj.2014.05.004.
- [151] Violeta I. Pérez-Nueno, Obdulia Rabal, José I. Borrell, and Jordi Teixidó. APIF: A new interaction fingerprint based on atom pairs and its application to virtual screening. *Journal of Chemical Information and Modeling*, 49(5):1245–1260, 2009. doi: 10.1021/ci900043r.
- [152] Tomohiro Sato, Teruki Honma, and Shigeyuki Yokoyama. Combining machine learning and pharmacophore-based interaction fingerprint for in silico screening. *Journal of Chemical Information and Modeling*, 50(1):170–185, 2009. doi: 10.1021/ci900382e.
- [153] Jérémy Desaphy, Eric Raimbaud, Pierre Ducrot, and Didier Rognan. Encoding protein–ligand interaction patterns in fingerprints and graphs. *Journal of Chemical Information and Modeling*, 53(3):623–637, 2013. doi: 10.1021/ci300566n.
- [154] C. Da and D. Kireev. Structural protein–ligand interaction fingerprints (SPLIF) for structure-based virtual screening: Method and benchmark study. *Journal of Chemical Information and Modeling*, 54(9):2555–2561, 2014. doi: 10.1021/ci500319f.
- [155] Maciej Wójcikowski, Michał Kukiełka, Marta M Stepniewska-Dziubinska, and Paweł Siedlecki. Development of a protein–ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. *Bioinformatics*, 35(8):1334–1341, 2018. doi: 10.1093/bioinformatics/bty757.
- [156] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855, 2015. URL <https://arxiv.org/abs/1510.02855>.
- [157] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Paweł Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018. doi: 10.1093/bioinformatics/bty374.
- [158] Jocelyn Sunseri, Jonathan E. King, Paul G. Francoeur, and David Ryan Koes. Convolutional neural network scoring and minimization in the d3r 2017 community challenge. *Journal of Computer-Aided Molecular Design*, 33(1):19–34, 2018. doi: 10.1007/s10822-018-0133-y.
- [159] José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. KDEEP: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of Chemical Information and Modeling*, 58(2):287–296, 2018. doi: 10.1021/acs.jcim.7b00650.

- 
- [160] Yanjun Li, Mohammad A. Rezaei, Chenglong Li, and Xiaolin Li. DeepAtom: A framework for protein-ligand binding affinity prediction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019. doi: 10.1109/bibm47256.2019.8982964.
- [161] Miha Skalic, Gerard Martínez-Rosell, José Jiménez, and Gianni De Fabritiis. PlayMolecule BindScope: large scale CNN-based virtual screening on the web. *Bioinformatics*, 35(7):1237–1238, 2018. doi: 10.1093/bioinformatics/bty758.
- [162] Ozlem Erdas-Cicek, Ali Osman Atac, A. Selen Gurkan-Alp, Erdem Buyukbingol, and Ferda Nur Alpaslan. Three-dimensional analysis of binding sites for predicting binding affinities in drug design. *Journal of Chemical Information and Modeling*, 59(11):4654–4662, 2019. doi: 10.1021/acs.jcim.9b00206.
- [163] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. 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, 2019. doi: 10.1021/acs.jcim.9b00387.
- [164] Evan N. Feinberg, Debnil Sur, Zhenqin Wu, Brooke E. Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S. Pande. PotentialNet for molecular property prediction. *ACS Central Science*, 4(11):1520–1530, 2018. doi: 10.1021/acscentsci.8b00507.
- [165] Zixuan Cang and Guo-Wei Wei. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLOS Computational Biology*, 13(7):e1005690, 2017. doi: 10.1371/journal.pcbi.1005690.
- [166] Fangqiang Zhu, Xiaohua Zhang, Jonathan E. Allen, Derek Jones, and Felice C. Lightstone. Binding affinity prediction by pairwise function based on neural network. *Journal of Chemical Information and Modeling*, 60(6):2766–2772, 2020. doi: 10.1021/acs.jcim.0c00026.
- [167] Janaina Cruz Pereira, Ernesto Raúl Caffarena, and Cicero Nogueira dos Santos. Boosting docking-based virtual screening with deep learning. *Journal of Chemical Information and Modeling*, 56(12):2495–2506, 2016. doi: 10.1021/acs.jcim.6b00355.
- [168] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

- 
- [169] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. URL <http://arxiv.org/abs/1409.1556>.
- [170] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. doi: 10.1109/cvpr.2015.7298594.
- [171] Zhiyuan Liu and Jie Zhou. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(2):1–127, 2020. doi: 10.2200/s00980ed1v01y202001aim045.
- [172] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *CoRR*, abs/1511.05493, 2017. URL <http://arxiv.org/abs/1511.05493>.
- [173] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2018. URL <http://arxiv.org/abs/1710.10903>.
- [174] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *CoRR*, abs/1812.08434, 2018. URL <http://arxiv.org/abs/1812.08434>.
- [175] Oliver Wieder, Stefan Kohlbacher, Méline Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 2020. ISSN 1740-6749. doi: 10.1016/j.ddtec.2020.11.009.
- [176] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. Pubchem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa971.
- [177] Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts of Chemical Research*, 50(2):302–309, 2017. doi: 10.1021/acs.accounts.6b00491.

- 
- [178] Michael K. Gilson, Tiqing Liu, Michael Baitaluk, George Nicola, Linda Hwang, and Jenny Chong. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research*, 44(D1):D1045–D1053, 2015. doi: 10.1093/nar/gkv1072.
- [179] BindingDB. <https://www.bindingdb.org/bind/index.jsp>, 2021. [Online; accessed 02-March-2021].
- [180] Aqeel Ahmed, Richard D. Smith, Jordan J. Clark, James B. Dunbar, and Heather A. Carlson. Recent improvements to binding MOAD: a resource for protein–ligand binding affinities and structures. *Nucleic Acids Research*, 43(D1):D465–D469, 2014. doi: 10.1093/nar/gku1088.
- [181] Richard D. Smith, Jordan J. Clark, Aqeel Ahmed, Zachary J. Orban, James B. Dunbar, and Heather A. Carlson. Updates to binding MOAD (mother of all databases): Polypharmacology tools and their utility in drug repurposing. *Journal of Molecular Biology*, 431(13):2423–2433, 2019. doi: 10.1016/j.jmb.2019.05.024.
- [182] PubChem. <https://pubchem.ncbi.nlm.nih.gov/>, 2021. [Online; accessed 02-March-2021].
- [183] Mark Davies, Michał Nowotka, George Papadatos, Nathan Dedman, Anna Gaulton, Francis Atkinson, Louisa Bellis, and John P. Overington. ChEMBL web services: streamlining access to drug discovery data and utilities. *Nucleic Acids Research*, 43(W1):W612–W620, apr 2015. doi: 10.1093/nar/gkv352. URL <https://doi.org/10.1093/nar/gkv352>.
- [184] Albert J. Kooistra and Andrea Volkamer. Kinase-centric computational drug development. In *Annual Reports in Medicinal Chemistry*, pages 197–236. Elsevier, 2017. doi: 10.1016/bs.armc.2017.08.001.
- [185] Mindy I Davis, Jeremy P Hunt, Sanna Herrgard, Piotr Ciceri, Lisa M Wodicka, Gabriel Pallares, Michael Hocker, Danie K Treiber, and Patrick P Zarrinkar. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29(11):1046–1051, 2011. doi: 10.1038/nbt.1990.
- [186] Jing Tang, Agnieszka Szwejda, Sushil Shakyawar, Tao Xu, Petteri Hintsanen, Krister Wennerberg, and Tero Aittokallio. Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014. doi: 10.1021/ci400709d.

- 
- [187] Jochen Sieg, Florian Flachsenberg, and Matthias Rarey. In need of bias control: evaluating chemical data for machine learning in structure-based virtual screening. *Journal of chemical information and modeling*, 59(3):947–961, 2019. doi: 10.1021/acs.jcim.8b00712.
- [188] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2): 895–913, 2018. doi: 10.1021/acs.jcim.8b00545.
- [189] Joseph Lee Rodgers and W. Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988. doi: 10.1080/00031305.1988.10475524.
- [190] C. Spearman. The proof and measurement of association between two things. *Am. J. Psychol.*, 15:72–101, 1904. doi: 10.2307/1422689. URL <https://doi.org/10.2307/1422689>.
- [191] Gerald J. Glasser and Robert F. Winter. Critical values of the coefficient of rank correlation for testing the hypothesis of independence. *Biometrika*, 48(3/4):444, dec 1961. doi: 10.2307/2332767. URL <https://doi.org/10.2307/2332767>.
- [192] Robert D. Wells, Judith S. Bond, Judith Klinman, and Bettie Sue Siler Masters, editors. *RMSD, Root-Mean-Square Deviation*, pages 1078–1078. Springer New York, New York, NY, 2018. ISBN 978-1-4614-1531-2. doi: 10.1007/978-1-4614-1531-2\_100140. URL [https://doi.org/10.1007/978-1-4614-1531-2\\_100140](https://doi.org/10.1007/978-1-4614-1531-2_100140).
- [193] Jean-François Truchon and Christopher I. Bayly. Evaluating virtual screening methods: good and bad metrics for the “early recognition” problem. *Journal of Chemical Information and Modeling*, 47(2):488–508, feb 2007. doi: 10.1021/ci600426e. URL <https://doi.org/10.1021/ci600426e>.
- [194] Oleg Trott and Arthur J. Olson. AutoDock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31:455–461, 2010. doi: 10.1002/jcc.21334. URL <https://doi.org/10.1002/jcc.21334>.
- [195] Thomas A. Halgren, Robert B. Murphy, Richard A. Friesner, Hege S. Beard, Leah L. Frye, W. Thomas Pollard, and Jay L. Banks. Glide: a new approach for rapid, accurate docking and scoring. 2. enrichment factors in database screening. *Journal of Medicinal Chemistry*, 47(7): 1750–1759, mar 2004. doi: 10.1021/jm030644s. URL <https://doi.org/10.1021/jm030644s>.

- 
- [196] Niu Huang, Brian K. Shoichet, and John J. Irwin. Benchmarking sets for molecular docking. *Journal of Medicinal Chemistry*, 49(23):6789–6801, 2006. doi: 10.1021/jm0608356.
- [197] Michael M. Mysinger, Michael Carchia, John. J. Irwin, and Brian K. Shoichet. Directory of useful decoys, enhanced (DUD-e): Better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, 2012. doi: 10.1021/jm300687e.
- [198] Sebastian G. Rohrer and Knut Baumann. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *Journal of Chemical Information and Modeling*, 49(2):169–184, 2009. doi: 10.1021/ci8002649.
- [199] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets for therapeutics. <https://tdcommons.ai>, 2020.
- [200] Sereina Riniker and Gregory A. Landrum. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of Cheminformatics*, 5(26):1758–2946, 2013. doi: 10.1186/1758-2946-5-26.
- [201] Mengyu Wang, Peng Li, and Peili Qiao. The virtual screening of the drug protein with a few crystal structures based on the adaboost-SVM. *Computational and Mathematical Methods in Medicine*, 2016:1–9, 2016. doi: 10.1155/2016/4809831.
- [202] Fei Li, Xiaozhe Wan, Jing Xing, Xiaoqin Tan, Xutong Li, Yulan Wang, Jihui Zhao, Xiaolong Wu, Xiaohong Liu, Zhaojun Li, Xiaomin Luo, Wencong Lu, and Mingyue Zheng. Deep neural network classifier for virtual screening inhibitors of (s)-adenosyl-l-methionine (SAM)-dependent methyltransferase family. *Frontiers in Chemistry*, 7, 2019. doi: 10.3389/fchem.2019.00324.
- [203] Fergus Imrie, Anthony R. Bradley, Mihaela van der Schaar, and Charlotte M. Deane. Protein family-specific models using deep neural networks and transfer learning improve virtual screening and highlight the need for more data. *Journal of Chemical Information and Modeling*, 58(11):2319–2330, 2018. doi: 10.1021/acs.jcim.8b00350.
- [204] Atsuko Sato, Naoki Tanimura, Teruki Honma, and Akihiko Konagaya. Significance of data selection in deep learning for reliable binding mode prediction of ligands in the active site of CYP3a4. *Chemical and Pharmaceutical Bulletin*, 67(11):1183–1190, 2019. doi: 10.1248/cpb.c19-00443.

- 
- [205] Duc Duy Nguyen, Kaifu Gao, Menglun Wang, and Guo-Wei Wei. MathDL: mathematical deep learning for d3r grand challenge 4. *Journal of Computer-Aided Molecular Design*, 34(2):131–147, 2019. doi: 10.1007/s10822-019-00237-5.
- [206] Zixuan Cang, Lin Mu, and Guo-Wei Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Computational Biology*, 14(1):e1005929, 2018. doi: 10.1371/journal.pcbi.1005929.
- [207] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. OnionNet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS Omega*, 4(14):15956–15965, 2019. doi: 10.1021/acsomega.9b01997.
- [208] Stefan Mordalski, Tomasz Kosciolok, Kurt Kristiansen, Ingebrigt Sylte, and Andrzej J. Bojarski. Protein binding site analysis by means of structural interaction fingerprint patterns. *Bioorganic & Medicinal Chemistry Letters*, 21(22):6816–6819, nov 2011. doi: 10.1016/j.bmcl.2011.09.027.
- [209] Jérémy Desaphy, Guillaume Bret, Didier Rognan, and Esther Kellenberger. sc-PDB: a 3d-database of ligandable binding sites—10 years on. *Nucleic Acids Research*, 43(D1):D399–D404, oct 2014. doi: 10.1093/nar/gku928. URL <https://doi.org/10.1093/nar/gku928>.
- [210] David Ryan Koes, Matthew P. Baumgartner, and Carlos J. Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of Chemical Information and Modeling*, 53(8):1893–1904, 2013. doi: 10.1021/ci300604z. PMID: 23379370.
- [211] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. doi: 10.1109/CVPR.2017.243.
- [212] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein-ligand scoring with convolutional neural networks. *Journal of Chemical Information and Modeling*, 57(4):942–957, 2017. doi: 10.1021/acs.jcim.6b00740.
- [213] Harry C Jubb, Alicia P Higuero, Bernardo Ochoa-Montaño, Will R Pitt, David B Ascher, and Tom L Blundell. Arpeggio: A web server for calculating and visualising interatomic interactions in protein structures. *Journal of Molecular Biology*, 429(3):365–371, feb 2017. doi: 10.1016/j.jmb.2016.12.004.

- 
- [214] Pedro J. Ballester and John B. O. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010. doi: 10.1093/bioinformatics/btq112.
- [215] Conor D. Parks, Zied Gaieb, Michael Chiu, Huanwang Yang, Chenghua Shao, W. Patrick Walters, Johanna M. Jansen, Georgia McGaughey, Richard A. Lewis, Scott D. Bembenek, Michael K. Ameriks, Tara Mirzadegan, Stephen K. Burley, Rommie E. Amaro, and Michael K. Gilson. D3r grand challenge 4: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. *Journal of Computer-Aided Molecular Design*, 34(2):99–119, 2020. doi: 10.1007/s10822-020-00289-y.
- [216] Hongjian Li, Kwong-Sak Leung, Man-Hon Wong, and Pedro 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):115–126, 2015. doi: 10.1002/minf.201400132.
- [217] Haiping Zhang, Linbu Liao, Konda Mani Saravanan, Peng Yin, and Yanjie Wei. DeepBindRG: a deep learning based method for estimating effective protein–ligand affinity. *PeerJ*, 7:e7362, july 2019. doi: 10.7717/peerj.7362.
- [218] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. WideDTA: prediction of drug-target binding affinity. *CoRR*, abs/1902.04166, 2019. URL <https://arxiv.org/abs/1902.04166>.
- [219] Kai Tian, Mingyu Shao, Yang Wang, Jihong Guan, and Shuigeng Zhou. Boosting compound-protein interaction prediction by deep learning. *Methods*, 110:64 – 72, 2016. ISSN 1046-2023. doi: 10.1016/j.ymeth.2016.06.024.
- [220] Ingoo Lee, Jongsoo Keum, and Hojun Nam. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLOS Computational Biology*, 15(6):e1007129, 2019. doi: 10.1371/journal.pcbi.1007129.
- [221] Mingjian Jiang, Zhen Li, Shugang Zhang, Shuang Wang, Xiaofeng Wang, Qing Yuan, and Zhiqiang Wei. Drug-target affinity prediction using graph neural network and contact maps. *RSC Adv.*, 10:20701–20712, 2020. doi: 10.1039/D0RA02297G.
- [222] Qingyuan Feng, Evgenia V. Dueva, Artem Cherkasov, and Martin Ester. PADME: A deep learning-based framework for drug-target



- 
- interaction prediction. *CoRR*, abs/1807.09741, 2018. URL <http://arxiv.org/abs/1807.09741>.
- [223] Twan van Laarhoven, Sander B. Nabuurs, and Elena Marchiori. Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics*, 27(21):3036–3043, 09 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr500.
- [224] Tong He, Marten Heidemeyer, Fuqiang Ban, Artem Cherkasov, and Martin Ester. SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*, 9(1), Apr 2017. doi: 10.1186/s13321-017-0209-z.
- [225] Michał Woźniak, Agnieszka Wołos, Urszula Modrzyk an Rafał L. Górski, Jan Winkowski, Michał Bajczyk, Sar Szymkuć, Bartosz A. Grzybowski, and Maciej Eder. Linguistic measures of chemical diversity and th “keywords” of molecular collections. *Scientific Reports*, 8(1), 2018. doi: 10.1038/s41598-018-25440-6.
- [226] Christian J. A. Sigrist, Lorenzo Cerutti, Edouard de Castro, Petra S. Langendijk-Genevaux, Virginie Bulliard, Amos Bairoch, and Nicolas Hulo. PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, 38(suppl\_1):D161–D166, 10 2009. ISSN 0305-1048. doi: 10.1093/nar/gkp885.
- [227] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N. Jorissen, and Michael K. Gilson. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, 35(suppl\_1):D198–D201, 12 2006. ISSN 0305-1048. doi: 10.1093/nar/gkl999.
- [228] Vivian Law, Craig Knox, Yannick Djoumbou, Tim Jewison, An Chi Guo, Yifeng Liu, Adam Maciejewski, David Arndt, Michael Wilson, Vanessa Neveu, Alexandra Tang, Geraldine Gabriel, Carol Ly, Sakina Adamjee, Zerihun T. Dame, Beomsoo Han, You Zhou, and David S. Wishart. DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, 42(D1):D1091–D1097, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1068.
- [229] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45(D1):D353–D361, 11 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw1092.
- [230] Christopher Southan, Joanna L. Sharman, Helen E. Benson, Elena Faccenda, Adam J. Pawson, Stephen P. H. Alexander, O. Peter Buneman,

- Anthony P. Davenport, John C. McGrath, John A. Peters, Michael Spedding, William A. Catterall, Dorian Fabbro, Jamie A. Davies, and NC-IUPHAR. The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands. *Nucleic Acids Research*, 44(D1):D1054–D1068, 10 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1037.
- [231] Steven C. Bagley and Russ B. Altman. Characterizing the microenvironment surrounding protein sites. *Protein Science*, 4(4):622–635, 1995. doi: 10.1002/pro.5560040404.
- [232] Mirco Michel, David Menéndez Hurtado, and Arne Elofsson. PconsC4: fast, accurate and hassle-free contact predictions. *Bioinformatics*, 35(15):2677–2679, 12 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty1036.
- [233] Dong-Sheng Cao, Qing-Song Xu, and Yi-Zeng Liang. propy: a tool to generate various modes of Chou’s PseAAC. *Bioinformatics*, 29(7):960–962, 02 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt072.
- [234] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015. doi: 10.1021/ci500747n.
- [235] Pedro J. Ballester, Adrian Schreyer, and Tom L. Blundell. Does a more precise chemical description of protein–ligand complexes lead to more accurate prediction of binding affinity? *Journal of Chemical Information and Modeling*, 54(3):944–955, feb 2014. doi: 10.1021/ci500091r.
- [236] Izhar Wallach and Abraham Heifets. Most ligand-based classification benchmarks reward memorization rather than generalization. *Journal of Chemical Information and Modeling*, 58(5):916–932, 2018. doi: 10.1021/acs.jcim.7b00403. PMID: 29698607.
- [237] Andreas H. Göller, Lara Kuhnke, Floriane Montanari, Anne Bonin, Sebastian Schneckener, Antonius ter Laak, Jörg Wichard, Mario Lobell, and Alexander Hillisch. Bayer’s in silico admet platform: a journey of machine learning over the past two decades. *Drug Discovery Today*, 25(9):1702–1709, 2020. ISSN 1359-6446. doi: <https://doi.org/10.1016/j.drudis.2020.07.001>. URL <https://www.sciencedirect.com/science/article/pii/S1359644620302609>.
- [238] Lieyang Chen, Anthony Cruz, Steven Ramsey, Callum J Dickson, Jose S Duca, Viktor Hornak, David R Koes, and Tom Kurtzman. Hidden bias in the dud-e dataset leads to misleading performance of

- deep learning in structure-based virtual screening. *PloS one*, 14(8): e0220113, 2019. doi: 10.1371/journal.pone.0220113.
- [239] José Jiménez-Luna, Miha Skalic, Nils Weskamp, and Gisbert Schneider. Coloring molecules with explainable artificial intelligence for pre-clinical relevance assessment. *Journal of Chemical Information and Modeling*, 0(0):null, 2021. doi: 10.1021/acs.jcim.0c01344. PMID: 33629843.
- [240] Andreas Bender and Isidro Cortés-Ciriano. Artificial intelligence in drug discovery: what is realistic, what are illusions? part 1: Ways to make an impact, and why we are not there yet. *Drug Discovery Today*, 2020. ISSN 1359-6446. doi: <https://doi.org/10.1016/j.drudis.2020.12.009>. URL <https://www.sciencedirect.com/science/article/pii/S1359644620305274>.
- [241] Andreas Bender and Isidro Cortes-Ciriano. Artificial intelligence in drug discovery: what is realistic, what are illusions? part 2: a discussion of chemical and biological data. *Drug Discovery Today*, 2021. ISSN 1359-6446. doi: <https://doi.org/10.1016/j.drudis.2020.11.037>. URL <https://www.sciencedirect.com/science/article/pii/S1359644621000428>.
- [242] Hai Nguyen, David A Case, and Alexander S Rose. NGLview-interactive molecular graphics for Jupyter notebooks. *Bioinformatics*, 34(7):1241–1242, 12 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx789. URL <https://doi.org/10.1093/bioinformatics/btx789>.
- [243] Maciej Wójcikowski, Piotr Zielenkiewicz, and Pawel Siedlecki. Open drug discovery toolkit (ODDT): a new open-source player in the drug discovery field. *Journal of Cheminformatics*, 7(1), jun 2015. doi: 10.1186/s13321-015-0078-2. URL <https://doi.org/10.1186/s13321-015-0078-2>.
- [244] Schrödinger, LLC. The PyMOL molecular graphics system, version 1.8, November 2015.
- [245] Steven M. Paul, Daniel S. Mytelka, Christopher T. Dunwiddie, Charles C. Persinger, Bernard H. Munos, Stacy R. Lindborg, and Aaron L. Schacht. How to improve r&d productivity: the pharmaceutical industry's grand challenge. *Nature Reviews Drug Discovery*, 9(3):203–214, 2010. doi: 10.1038/nrd3078.
- [246] Jack W. Scannell, Alex Blanckley, Helen Boldon, and Brian Warrington. Diagnosing the decline in pharmaceutical r&d efficiency. *Nature Reviews Drug Discovery*, 11(3):191–200, 2012. doi: 10.1038/nrd3681.

- 
- [247] Michael J. Waring, John Arrowsmith, Andrew R. Leach, Paul D. Leeson, Sam Mandrell, Robert M. Owen, Garry Pairaudeau, William D. Pennie, Stephen D. Pickett, Jibo Wang, Owen Wallace, and Alex Weir. An analysis of the attrition of drug candidates from four major pharmaceutical companies. *Nature Reviews Drug Discovery*, 14(7):475–486, 2015. doi: 10.1038/nrd4609.
- [248] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017. ISSN 0001-0782. doi: 10.1145/3065386.
- [249] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013. doi: 10.1109/ICASSP.2013.6638947.
- [250] Evan N. Feinberg, Debnil Sur, Zhenqin Wu, Brooke E. Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S. Pande. Potentialnet for molecular property prediction. *ACS Central Science*, 4(11):1520–1530, 2018. doi: 10.1021/acscentsci.8b00507.
- [251] Boris Sattarov, Igor I. Baskin, Dragos Horvath, Gilles Marcou, Esben Jannik Bjerrum, and Alexandre Varnek. De novo molecular design by combining deep autoencoder recurrent neural networks with generative topographic mapping. *Journal of Chemical Information and Modeling*, 59(3):1182–1196, 2019. doi: 10.1021/acs.jcim.8b00751.
- [252] Ruili Huang and Menghang Xia. Editorial: Tox21 challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental toxicants and drugs. *Frontiers in Environmental Science*, 5:3, 2017. ISSN 2296-665X. doi: 10.3389/fenvs.2017.00003.
- [253] Noel M O’Boyle. Towards a universal SMILES representation - a standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics*, 4(1), 2012. doi: 10.1186/1758-2946-4-22.
- [254] David Weininger, Arthur Weininger, and Joseph L. Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101, 1989. doi: 10.1021/ci00062a008.
- [255] Jennifer Hemmerich, Ece Asilar, and Gerhard F. Ecker. COVER: conformational oversampling as data augmentation for molecules. *Journal of Cheminformatics*, 12(1), 2020. doi: 10.1186/s13321-020-00420-z.

- 
- [256] Yanjun Li, Mohammad A. Rezaei, Chenglong Li, and Xiaolin Li. Deepatom: A framework for protein-ligand binding affinity prediction. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 303–310, 2019. doi: 10.1109/BIBM47256.2019.8982964.
- [257] Xinhao Li and Denis Fourches. Inductive transfer learning for molecular activity prediction: Next-gen QSAR models with MolPMoFiT. *Journal of Cheminformatics*, 12(1), 2020. doi: 10.1186/s13321-020-00430-x.
- [258] Igor V. Tetko, Pavel Karpov, Eric Bruno, Talia B. Kimber, and Guillaume Godin. Augmentation is what you need! In Igor V. Tetko, Věra Kůrková, Pavel Karpov, and Fabian Theis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*, pages 831–835, Cham, 2019. Springer International Publishing. ISBN 978-3-030-30493-5. doi: 10.1007/978-3-030-30493-5\_79.
- [259] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 2019. doi: 10.1186/s40537-019-0197-0.
- [260] Igor V. Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1), 2020. doi: 10.1038/s41467-020-19266-y.
- [261] Dean Sumner, Jiazhen He, Amol Thakkar, Ola Engkvist, and Esben Jannik Bjerrum. Levenshtein augmentation improves performance of smiles based deep-learning synthesis prediction. *ChemRxiv*, 2020. doi: 10.26434/chemrxiv.12562121.v2.
- [262] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics*, 11(1), 2019. doi: 10.1186/s13321-019-0393-0.
- [263] Ruud van Deursen, Peter Ertl, Igor V. Tetko, and Guillaume Godin. GEN: highly efficient SMILES explorer using autodidactic generative examination networks. *Journal of Cheminformatics*, 12(1), 2020. doi: 10.1186/s13321-020-00425-8.
- [264] Jacqueline Kazil and Katharine Jarmul. *Data Wrangling with Python: Tips and Tools to Make Your Life Easier*. O’Reilly Media, Inc., 1st edition, 2016. ISBN 1491948817.

- 
- [265] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. *arXiv preprint arXiv:1811.00908*, 2019. URL <https://arxiv.org/abs/1811.00908>.
- [266] Gabriele Scalia, Colin A. Grambow, Barbara Pernici, Yi-Pei Li, and William H. Green. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of Chemical Information and Modeling*, 60(6):2697–2717, 2020. doi: 10.1021/acs.jcim.9b00975.
- [267] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks, 2018. URL <https://openreview.net/pdf?id=rJZz-knjz>.
- [268] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K. Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chem. Sci.*, 9:5441–5451, 2018. doi: 10.1039/C8SC00148K.
- [269] Yao Zhang and Alpha A. Lee. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chem. Sci.*, 10:8154–8163, 2019. doi: 10.1039/C9SC00616H.
- [270] ChEMBL. <https://www.ebi.ac.uk/chembl/>, 2021. [Online; accessed 27-August-2021].
- [271] Albert J. Kooistra and Andrea Volkamer. Kinase-centric computational drug development. In *Annual Reports in Medicinal Chemistry*, pages 197–236. Elsevier, 2017. doi: 10.1016/bs.armc.2017.08.001.
- [272] OpenKinome. <http://openkinome.org/>, 2021. [Online; accessed 27-August-2021].
- [273] Roy S Herbst. Review of epidermal growth factor receptor biology. *International Journal of Radiation Oncology\*Biophysics*, 59(2, Supplement):S21–S26, 2004. ISSN 0360-3016. doi: 10.1016/j.ijrobp.2003.11.041.
- [274] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100.
- [275] Stefan Offermanns and Walter Rosenthal, editors. *IC50 Values*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-38918-7. doi: 10.1007/978-3-540-38918-7\_5943.

- 
- [276] Claude Sammut and Geoffrey I. Webb, editors. *Mean Squared Error*. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi: 10.1007/978-0-387-30164-8\_528.
- [277] Tarald O. Kvålseth. Cautionary note about  $r^2$ . *The American Statistician*, 39(4):279–285, 1985. doi: 10.1080/00031305.1985.10479448.
- [278] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 0885-6125, 1573-0565. doi: 10.1023/A:1010933404324.
- [279] Loris Bennett, Bernd Melchers, and Boris Proppe. Curta: A general-purpose high-performance computer at zedat, freie universität berlin, 2020. URL <http://dx.doi.org/10.17169/refubium-26754>.
- [280] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.
- [281] Guido van Rossum, Barry Warsaw, and Nick Coghlan. Style guide for Python code, 2001. URL <https://www.python.org/dev/peps/pep-0008/>.
- [282] Read the Docs. <https://readthedocs.io/en/stable/>, 2021. [Online; accessed 30-July-2021].
- [283] Anaconda software distribution, 2020. URL <https://anaconda.com/>.
- [284] The pandas development team. pandas-dev/pandas: Pandas, feb 2020.
- [285] GitHub. GitHub Actions. <https://docs.github.com/en/actions>, 2021. URL <https://docs.github.com/en/actions>. [Online; accessed 2021-10-06].
- [286] pytest. pytest. <https://docs.pytest.org/>, 2021. URL <https://docs.pytest.org/>. [Online; accessed 2021-10-06].
- [287] Codecov. <https://docs.codecov.com/docs>, 2021. [Online; accessed 30-July-2021].
- [288] CAS. CAS REGISTRY, 2018. URL <https://www.cas.org/support/documentation/chemical-substances>.
- [289] Thomas Hartung. Making big sense from big data in toxicology by read-across. *ALTEX - Alternatives to animal experimentation*, 33(2): 83–93, May 2016. doi: 10.14573/altex.1603091. URL <https://www.altex.org/index.php/altex/article/view/160>.
- [290] Michael J Waring, John Arrowsmith, Andrew R Leach, Paul D Leeson, Sam Mandrell, Robert M Owen, Garry Pairaudeau, William D Pennie, Stephen D Pickett, Jibo Wang, et al. An analysis of the attrition of

- drug candidates from four major pharmaceutical companies. *Nature reviews Drug discovery*, 14(7):475, 2015. doi: <https://doi.org/10.1038/nrd4609>.
- [291] James M McKim. Building a tiered approach to in vitro predictive toxicity screening: a focus on assays with in vivo relevance. *Combinatorial chemistry & high throughput screening*, 13(2):188–206, feb 2010. doi: <https://doi.org/10.2174/138620710790596736>.
- [292] BMEL - Übersicht: BMEL informiert über Tierschutz - Verwendung von Versuchstieren im Jahr 2016, 2018. URL [https://www.bmel.de/DE/Tier/Tierschutz/\\_texte/Versuchstierzahlen2016.html#doc10323474bodyText6](https://www.bmel.de/DE/Tier/Tierschutz/_texte/Versuchstierzahlen2016.html#doc10323474bodyText6).
- [293] Pau Carrió, Ferran Sanz, and Manuel Pastor. Toward a unifying strategy for the structure-based prediction of toxicological endpoints. *Archives of Toxicology*, 90(10):2445–2460, oct 2016. doi: <https://doi.org/10.1007/s00204-015-1618-2>.
- [294] Regulation (EC) No 1907/2006 of the European Parliament and of the Council of 18 December 2006 concerning the Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH), 2020. URL [https://ec.europa.eu/environment/chemicals/reach/reach\\_en.htm](https://ec.europa.eu/environment/chemicals/reach/reach_en.htm).
- [295] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013. URL <http://arxiv.org/abs/1303.5778>.
- [296] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL <http://papers.nips.cc/paper/4824-image-net-classification-with-deep-convolutional-neural-networks.pdf>.
- [297] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018. doi: 10.1126/sciadv.aap7885. URL <https://advances.sciencemag.org/content/4/7/eaap7885>.
- [298] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science*, 4(1):120–131, 2018. doi: 10.1021/acscentsci.7b00512. URL <https://doi.org/10.1021/acscentsci.7b00512>. PMID: 29392184.



- [299] Evgeny Putin, Arip Asadulaev, Yan Ivanenkov, Vladimir Aladin-skiy, Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik, and Alex Zha- voronkov. Reinforced Adversarial Neural Computer for de Novo Molec- ular Design. *Journal of Chemical Information and Modeling*, 58(6): 1194–1204, 2018. doi: 10.1021/acs.jcim.7b00690. URL <https://doi.org/10.1021/acs.jcim.7b00690>. PMID: 29762023.
- [300] Thomas Blaschke, Marcus Olivecrona, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Application of Generative Autoencoder in De Novo Molecular Design. *Molecular Informatics*, 37(1-2):1700123, 2018. doi: 10.1002/minf.201700123. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201700123>.
- [301] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572. URL <https://doi.org/10.1021/acscentsci.7b00572>. PMID: 29532027.
- [302] Adam C. Mater and Michelle L. Coote. Deep learning in chemistry. *Journal of Chemical Information and Modeling*, 59(6):2545–2559, 2019. doi: 10.1021/acs.jcim.9b00266. URL <https://doi.org/10.1021/acs.jcim.9b00266>.
- [303] Ye Hu, Dagmar Stumpfe, and Jürgen Bajorath. Advancing the activity cliff concept. *F1000Research*, 2, sep 2013. ISSN 2046-1402. doi: 10.12688/f1000research.2-199.v1. URL <http://f1000research.com/articles/2-199/v1>.
- [304] Kaitlyn M. Gayvert, Neel S. Madhukar, and Olivier Elemento. A Data-Driven Approach to Predicting Successes and Failures of Clinical Trials. *Cell Chemical Biology*, 23(10):1294–1301, oct 2016. ISSN 24519456. doi: 10.1016/j.chembiol.2016.07.023. URL <http://www.ncbi.nlm.nih.gov/pubmed/27642066>.
- [305] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, feb 2015. ISSN 1549-9596. doi: 10.1021/ci500747n. URL <http://pubs.acs.org/doi/10.1021/ci500747n>.
- [306] Serena Nembri, Francesca Grisoni, Viviana Consonni, and Roberto Todeschini. In Silico Prediction of Cytochrome P450-Drug Interaction: QSARs for CYP3A4 and CYP2C9. *International Journal of Molecular*

- Sciences*, 17(6):914, jun 2016. doi: 10.3390/ijms17060914. URL <http://www.mdpi.com/1422-0067/17/6/914>.
- [307] Andreas Bender. 'AI' in Toxicology (In Silico Toxicology) - The Pieces Don't Yet Fit Together, 2019. URL <http://www.drugdiscovery.net/tag/insilicotox/>.
- [308] Lewis H. Mervin, Qing Cao, Ian P. Barrett, Mike A. Firth, David Murray, Lisa McWilliams, Malcolm Haddrick, Mark Wigglesworth, Ola Engkvist, and Andreas Bender. Understanding Cytotoxicity and Cytostaticity in a High-Throughput Screening Collection. *ACS Chemical Biology*, 11(11):3007–3023, nov 2016. ISSN 1554-8929. doi: 10.1021/acscchembio.6b00538. URL <http://pubs.acs.org/doi/10.1021/acscchembio.6b00538>.
- [309] Terry L Riss, Richard A Moravec, and Andrew L Niles. Cytotoxicity testing: measuring viable cells, dead cells, and detecting mechanism of cell death. In *Mammalian Cell Viability*, pages 103–114. Springer, 2011. doi: [https://link.springer.com/protocol/10.1007/978-1-61779-108-6\\_12](https://link.springer.com/protocol/10.1007/978-1-61779-108-6_12).
- [310] Priyanka Banerjee, Andreas O Eckert, Anna K Schrey, and Robert Preissner. ProTox-II: a webserver for the prediction of toxicity of chemicals. *Nucleic Acids Research*, apr 2018. ISSN 0305-1048. doi: 10.1093/nar/gky318. URL <https://academic.oup.com/nar/advance-article/doi/10.1093/nar/gky318/4990033>.
- [311] Fredrik Svensson, Ulf Norinder, and Andreas Bender. Modelling compound cytotoxicity using conformal prediction and PubChem HTS data. *Toxicology Research*, 6(1):73–80, 2017. ISSN 2045-452X. doi: 10.1039/C6TX00252H. URL <http://xlink.rsc.org/?DOI=C6TX00252H>.
- [312] Sarah R. Langdon, Joanna Mulgrew, Gaia V. Paolini, and Willem P. Van Hoorn. Predicting cytotoxicity from heterogeneous data sources with Bayesian learning. *Journal of Cheminformatics*, 2(1):11, dec 2010. ISSN 17582946. doi: 10.1186/1758-2946-2-11. URL <https://jcheminf.biomedcentral.com/articles/10.1186/1758-2946-2-11>.
- [313] Alexey A. Lagunin, Varvara I. Dubovskaja, Anastasia V. Rudik, Pavel V. Pogodin, Dmitry S. Druzhilovskiy, Tatyana A. Glorizova, Dmitry A. Filimonov, Narahari G. Sastry, and Vladimir V. Poroikov. CLC-Pred: A freely available web-service for in silico prediction of human cell line cytotoxicity for drug-like compounds. *PLOS ONE*, 13(1):1–13, 01 2018. doi: 10.1371/journal.pone.0191838. URL <https://doi.org/10.1371/journal.pone.0191838>.

- 
- [314] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [315] Thomas Unterthiner, Andreas Mayr, Günter Klambauer, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, and Sepp Hochreiter. Deep learning as an opportunity in virtual screening. In *Proceedings of the deep learning workshop at NIPS*, volume 27, pages 1–9, 2014. URL <https://pdfs.semanticscholar.org/95f7/b2c0fe75f08e3ce0d2ac4315166f4239db5c.pdf>.
- [316] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.*, 9:513–530, 2018. doi: 10.1039/C7SC02664A. URL <http://dx.doi.org/10.1039/C7SC02664A>.
- [317] Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low Data Drug Discovery with One-Shot Learning. *ACS Central Science*, 3(4):283–293, apr 2017. ISSN 2374-7943. doi: 10.1021/acscentsci.6b00367. URL <http://pubs.acs.org/doi/10.1021/acscentsci.6b00367>.
- [318] Denis Fourches, Eugene Muratov, and Alexander Tropsha. Trust, but verify: On the importance of chemical structure curation in cheminformatics and qsar modeling research. *Journal of Chemical Information and Modeling*, 50(7):1189–1204, 2010. doi: 10.1021/ci100176x. URL <https://doi.org/10.1021/ci100176x>. PMID: 20572635.
- [319] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. DeepTox: Toxicity Prediction using Deep Learning. *Frontiers in Environmental Science*, 3:80, feb 2016. doi: <https://doi.org/10.3389/fenvs.2015.00080>.
- [320] Robert P. Sheridan. Interpretation of QSAR models by coloring atoms according to changes in predicted activity: How robust is it? *Journal of Chemical Information and Modeling*, 59(4):1324–1337, 2019. doi: 10.1021/acs.jcim.8b00825. URL <https://doi.org/10.1021/acs.jcim.8b00825>. PMID: 30779563.
- [321] Kristina Preuer, Günter Klambauer, Friedrich Rippmann, Sepp Hochreiter, and Thomas Unterthiner. *Interpretable Deep Learning in Drug Discovery*, pages 331–345. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6\_18. URL [https://doi.org/10.1007/978-3-030-28954-6\\_18](https://doi.org/10.1007/978-3-030-28954-6_18).

- [322] Matteo Manica, Ali Oskooei, Jannis Born, Vigneshwari Subramanian, Julio Sáez-Rodríguez, and María Rodríguez Martínez. Toward explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *Molecular Pharmaceutics*, 16(12):null, 2019. doi: 10.1021/acs.molpharmaceut.9b00520. URL <https://doi.org/10.1021/acs.molpharmaceut.9b00520>. PMID: 31618586.
- [323] Joshua Hochuli, Alec Helbling, Tamar Skaist, Matthew Ragoza, and David Ryan Koes. Visualizing convolutional neural network protein-ligand scoring. *Journal of Molecular Graphics and Modelling*, 84:96–108, 2018. ISSN 1093-3263. doi: <https://doi.org/10.1016/j.jmgm.2018.06.005>. URL <http://www.sciencedirect.com/science/article/pii/S1093326318301670>.
- [324] Petar Žuvela, Jonathan David, and Ming Wah Wong. Interpretation of ANN-based QSAR models for prediction of antioxidant activity of flavonoids. *Journal of Computational Chemistry*, 39(16):953–963, 2018. doi: 10.1002/jcc.25168. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.25168>.
- [325] Sereina Riniker and Gregory A Landrum. Similarity maps - a visualization strategy for molecular fingerprints and machine-learning methods. *Journal of Cheminformatics*, 5(1):43, sep 2013. ISSN 1758-2946. doi: 10.1186/1758-2946-5-43. URL <http://jcheminf.springeropen.com/articles/10.1186/1758-2946-5-43>.
- [326] Michael Lisurek, Bernd Rupp, Jörg Wichard, Martin Neuenschwander, Jens Peter von Kries, Ronald Frank, Jörg Rademann, and Ronald Kühne. Design of chemical libraries with potentially bioactive molecules applying a maximum common substructure concept. *Molecular Diversity*, 14(2):401–408, May 2010. ISSN 1573-501X. doi: 10.1007/s11030-009-9187-z. URL <https://doi.org/10.1007/s11030-009-9187-z>.
- [327] Jonathan B. Baell and Georgina A. Holloway. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *Journal of Medicinal Chemistry*, 53(7):2719–2740, 2010. doi: 10.1021/jm901137j. URL <https://doi.org/10.1021/jm901137j>. PMID: 20131845.
- [328] Michelle T.Z. Spence Iain Johnson. *The Molecular Probes Handbook: A Guide to Fluorescent Probes and Labeling Technologies*. Live Technologies Corporation, eleventh edition, 2010. ISBN 978-0-9829279-1-5. URL <http://probes.invitrogen.com/handbook/>.

- 
- [329] Francis Atkinson. standardiser 0.1.9, 8 2017. URL <https://pypi.org/project/standardiser/>.
- [330] Martin Gütlein and Stefan Kramer. Filtered circular fingerprints improve either prediction or runtime performance while retaining interpretability. *Journal of Cheminformatics*, 8(1):60, dec 2016. ISSN 1758-2946. doi: 10.1186/s13321-016-0173-z. URL <http://jcheminf.springeropen.com/articles/10.1186/s13321-016-0173-z>.
- [331] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- [332] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [333] Vladimir Svetnik, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003. doi: 10.1021/ci034160g. URL <https://doi.org/10.1021/ci034160g>. PMID: 14632445.
- [334] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124, Aug 2010. doi: 10.1109/ICPR.2010.764.
- [335] Takaya Saito and Marc Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3):1–21, 03 2015. doi: 10.1371/journal.pone.0118432. URL <https://doi.org/10.1371/journal.pone.0118432>.
- [336] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):e0130140, jul 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0130140. URL <http://dx.plos.org/10.1371/journal.pone.0130140>.
- [337] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. iNNvestigate

- neural networks. *Journal of Machine Learning Research*, 20(93):1–8, 2019.
- [338] 2020. URL <https://www.sigmaaldrich.com/life-science/cell-biology/bioactive-small-molecules/lopac1280-navigator.html>.
- [339] 2020. URL <https://www.selleckchem.com/screening/fda-approved-drug-library.html>.
- [340] Greg Landrum. Working with unbalanced data, part I. <http://rdkit.blogspot.com/2018/11/working-with-unbalanced-data-part-i.html>, 2018. [Online; accessed 28-November-2018].
- [341] Changge Ji, Fredrik Svensson, Azedine Zoufir, and Andreas Bender. eMolTox: prediction of molecular toxicity with confidence. *Bioinformatics*, 34(14):2508–2509, 03 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty135. URL <https://doi.org/10.1093/bioinformatics/bty135>.
- [342] Maykel Cruz-Monteagudo, José L. Medina-Franco, Yunierkis Pérez-Castillo, Orazio Nicolotti, M. Natália D.S. Cordeiro, and Fernanda Borges. Activity cliffs in drug discovery: Dr jekyll or mr hyde? *Drug Discovery Today*, 19(8):1069–1080, 2014. ISSN 1359-6446. doi: <https://doi.org/10.1016/j.drudis.2014.02.003>. URL <https://www.sciencedirect.com/science/article/pii/S1359644614000361>.
- [343] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2015. URL <https://arxiv.org/abs/1409.0473>.
- [344] T. Hanser, C. Barber, J. F. Marchaland, and S. Werner. Applicability domain: towards a more formal definition. *SAR and QSAR in Environmental Research*, 27(11):865–881, nov 2016. ISSN 1029046X. doi: 10.1080/1062936X.2016.1250229. URL <https://www.tandfonline.com/doi/full/10.1080/1062936X.2016.1250229>.
- [345] Talia B. Kimber, Sebastian Engelke, Igor V. Tetko, Eric Bruno, and Guillaume Godin. Synergy Effect between Convolutional Neural Networks and the Multiplicity of SMILES for Improvement of Molecular Prediction. *arXiv preprint arXiv:1812.04439*, 2018. URL <http://arxiv.org/abs/1812.04439>.
- [346] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by

- translating equivalent chemical representations. *Chem. Sci.*, 10:1692–1701, 2019. doi: 10.1039/C8SC04175J. URL <http://dx.doi.org/10.1039/C8SC04175J>.
- [347] Petra Schneider, W. Patrick Walters, Alleyn T. Plowright, Norman Sieroka, Jennifer Listgarten, Robert A. Goodnow, Jasmin Fisher, Johanna M. Jansen, José S. Duca, Thomas S. Rush, Matthias Zentgraf, John Edward Hill, Elizabeth Krutoholow, Matthias Kohler, Jeff Blaney, Kimito Funatsu, Chris Luebke, and Gisbert Schneider. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discovery*, 19(5):353–364, 2020. doi: 10.1038/s41573-019-0050-3.
- [348] Ashley Ringer McDonald. *Teaching Programming across the Chemistry Curriculum*, chapter Teaching Programming across the Chemistry Curriculum: A Revolution or a Revival?, pages 1–11. American Chemical Society, 2021.
- [349] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter Notebooks - A Publishing Format For Reproducible Computational Workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Netherlands, 2016. IOS Press. [doi:10.3233/978-1-61499-649-1-87].
- [350] David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R Leach. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, 2018. doi: 10.1093/nar/gky1075.
- [351] S Riniker, GA Landrum, F Montanari, SD Villalba, J Maier, JM Jansen, WP Walters, and AA Shelat. Virtual-screening workflow tutorials and prospective results from the teach-discover-treat competition 2014 against malaria [version 2; peer review: 3 approved]. *F1000Research*, 6(1136), 2018. doi: 10.12688/f1000research.11905.2.
- [352] Egon L. Willighagen, John W. Mayfield, Jonathan Alvarsson, Arvid Berg, Lars Carlsson, Nina Jeliaskova, Stefan Kuhn, Tomáš Pluskal, Miquel Rojas-Chertó, Ola Spjuth, Gilleain Torrance, Chris T. Evelo, Rajarshi Guha, and Christoph Steinbeck. The chemistry development

- kit (cdk) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of Cheminformatics*, 9(1):33, 2017. doi: 10.1186/s13321-017-0220-4. [doi:10.1186/s13321-017-0220-4].
- [353] Georgi K Kanev, Chris de Graaf, Bart A Westerman, Iwan J P de Esch, and Albert J Kooistra. KLIFS: an overhaul after the first 5 years of supporting kinase research. *Nucleic Acids Research*, 49(D1):D562–D569, oct 2020. doi: 10.1093/nar/gkaa895. URL <https://doi.org/10.1093/nar/gkaa895>.
- [354] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 2020. doi: 10.1093/nar/gkaa971.
- [355] David Ryan Koes, Matthew P. Baumgartner, and Carlos J. Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of Chemical Information and Modeling*, 53(8):1893–1904, 2013. doi: 10.1021/ci300604z.
- [356] Sebastian Salentin, Sven Schreiber, V. Joachim Haupt, Melissa F. Adasme, and Michael Schroeder. PLIP: fully automated protein-ligand interaction profiler. *Nucleic Acids Research*, 43(W1):W443–W447, 2015. doi: 10.1093/nar/gkv315.
- [357] Hai Nguyen, David A Case, and Alexander S Rose. NGLView - Interactive Molecular Graphics For Jupyter Notebooks. *Bioinformatics*, 34(7):1241–1242, 2017. doi: 10.1093/bioinformatics/btx789.
- [358] Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, Kyle A. Beauchamp, Lee-Ping Wang, Andrew C. Simmonett, Matthew P. Harrigan, Chaya D. Stern, Rafal P. Wiewiora, Bernard R. Brooks, and Vijay S. Pande. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):1–17, 2017. doi: 10.1371/journal.pcbi.1005659.
- [359] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32(10):2319–2327, 2011. doi: 10.1002/JCC.21787.
- [360] Richard J. Gowers, Max Linke, Jonathan Barnoud, Tyler J. E. Reddy, Manuel N. Melo, Sean L. Seyler, Jan Domański, David L. Dotson, Sébastien Buchoux, Ian M. Kenney, and Oliver Beckstein. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics



- 
- Simulations. In Sebastian Benthall and Scott Rostrup, editors, *Proceedings of the 15th Python in Science Conference*, pages 98 – 105, 2016. doi: 10.25080/Majora-629e541a-00e.
- [361] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a. [doi:10.25080/Majora-92bf1922-00a].
- [362] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55. [doi:10.1109/MCSE.2007.55].
- [363] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021.
- [364] Sam M Ireland and Andrew C R Martin. GraphQL for the Delivery of Bioinformatics Web APIs and Application to ZincBind. *Bioinformatics Advances*, 2021. doi: 10.1093/bioadv/vbab023. [doi:10.1093/bioadv/vbab023].
- [365] Patrick Kunzmann and Kay Hamacher. Biotite: a unifying open source computational biology framework in python. *BMC Bioinformatics*, 19(1):346, 2018. doi: 10.1186/s12859-018-2367-z.
- [366] William Gilpin. PyPDB: A Python API For The Protein Data Bank. *Bioinformatics*, 32:159–60, 9 2015. doi: 10.1093/bioinformatics/btv543.
- [367] The UniProt Consortium. UniProt: the universal protein knowledge-base in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2020. doi: 10.1093/nar/gkaa1100.
- [368] Dominique Sydow, Jaime Rodríguez-Guerra, and Andrea Volkamer. Opencadd-klifs: A python package to fetch kinase data from the klifs database. *Journal of Open Source Software*, 7(70):3951, 2022. doi: 10.21105/joss.03951. URL <https://doi.org/10.21105/joss.03951>. [doi:10.21105/joss.03951].
- [369] Sunghwan Kim, Paul A. Thiessen, Tiejun Cheng, Bo Yu, and Evan E. Bolton. An update on pug-rest: Restful interface for programmatic access to pubchem. *Nucleic Acids Research*, 46(W1), 2018. doi: 10.1093/nar/gky294.
- [370] Rainer Fährrolfes, Stefan Bietz, Florian Flachsenberg, Agnes Meyder, Eva Nittinger, Thomas Otto, Andrea Volkamer, and Matthias Rarey. ProteinsPlus: a web portal for structure analysis of macromolecules.

- 
- Nucleic Acids Research*, 45(W1):W337–W343, 2017. doi: 10.1093/nar/gkx333.
- [371] Andrea Volkamer, Daniel Kuhn, Thomas Grombacher, Friedrich Rippmann, and Matthias Rarey. Combining global and local measures for structure-based druggability predictions. *Journal of Chemical Information and Modeling*, 52(2):360–372, 2012. doi: 10.1021/ci200454v.
- [372] Oscar P. J. van Linden, Albert J. Kooistra, Rob Leurs, Iwan J. P. de Esch, and Chris de Graaf. Klifs: A knowledge-based structural database to navigate kinase-ligand interaction space. *Journal of Medicinal Chemistry*, 57(2):249–277, 2014. doi: 10.1021/jm400378w.
- [373] Noel M. O’Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(1):33, 2011. doi: 10.1186/1758-2946-3-33.
- [374] Gerhard Klebe. *Drug Design: Methodology, Concepts, and Mode-of-Action*, chapter Protein–Ligand Interactions as the Basis for Drug Action, pages 61–88. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-17907-5\_4. [doi:10.1007/978-3-642-17907-5\_4].
- [375] Alexander S. Rose and Peter W. Hildebrand. NGL Viewer: a web application for molecular visualization. *Nucleic Acids Research*, 43(W1):W576–W579, 2015. doi: 10.1093/nar/gkv402.
- [376] Alexander S Rose, Anthony R Bradley, Yana Valasatava, Jose M Duarte, Andreas Prlić, and Peter W Rose. NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics*, 34(21):3755–3758, 2018. doi: 10.1093/bioinformatics/bty419.
- [377] Youichi Kawakita, Masaki Seto, Tomohiro Ohashi, Toshiya Tamura, Tadashi Yusa, Hiroshi Miki, Hidehisa Iwata, Hidenori Kamiguchi, Toshimasa Tanaka, Satoshi Sogabe, Yoshikazu Ohta, and Tomoyasu Ishikawa. Design and synthesis of novel pyrimido[4,5-b]azepine derivatives as her2/egfr dual inhibitors. *Bioorganic and Medicinal Chemistry*, 21(8):2250–2261, 2013. ISSN 0968-0896. doi: <https://doi.org/10.1016/j.bmc.2013.02.014>. URL <https://www.sciencedirect.com/science/article/pii/S0968089613001387>.
- [378] Jing Yang, Zhengchao Tu, Xin Xu, Jinfeng Luo, Xing Yan, Chongzhao Ran, Xinliang Mao, Ke Ding, and Chunhua Qiao. Novel conjugates of endoperoxide and 4-anilinoquinazoline as potential anticancer agents. *Bioorganic and Medicinal Chemistry Letters*, 27(6):1341–1345, 2017. ISSN 0960-894X. doi: <https://doi.org/10.1016/j.bmcl.2017.02.023>.

---

URL <https://www.sciencedirect.com/science/article/pii/S0960894X17301555>.

- [379] Jérémie Mortier, Christin Rakers, Marcel Bermudez, Manuela S. Murgueitio, Sereina Riniker, and Gerhard Wolber. The impact of molecular dynamics on drug design: applications for the characterization of ligand-macromolecule complexes. *Drug Discovery Today*, 20(6):686–702, 2015. doi: <https://doi.org/10.1016/j.drudis.2015.01.003>.
- [380] Marco De Vivo, Matteo Masetti, Giovanni Bottegoni, and Andrea Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of Medicinal Chemistry*, 59(9):4035–4061, 2016. doi: [10.1021/acs.jmedchem.5b01684](https://doi.org/10.1021/acs.jmedchem.5b01684).
- [381] Veronica Salmaso and Stefano Moro. Bridging molecular docking to molecular dynamics in exploring ligand-protein recognition process: An overview. *Frontiers in Pharmacology*, 9:923, 2018. doi: [10.3389/fphar.2018.00923](https://doi.org/10.3389/fphar.2018.00923).
- [382] Robert T. McGibbon, Kyle A. Beauchamp, Matthew P. Harrigan, Christoph Klein, Jason M. Swails, Carlos X. Hernández, Christian R. Schwantes, Lee-Ping Wang, Thomas J. Lane, and Vijay S. Pande. Mdtraj: A modern open library for the analysis of molecular dynamics trajectories. *Biophysical Journal*, 109(8):1528–1532, 2015. doi: [10.1016/j.bpj.2015.08.015](https://doi.org/10.1016/j.bpj.2015.08.015).
- [383] Pablo R. Arantes, Marcelo D. Polêto, Conrado Pedebos, and Rodrigo Ligabue-Braun. Making it rain: Cloud-based molecular simulations for everyone. *Journal of Chemical Information and Modeling*, 61(10):4852–4856, 2021. doi: [10.1021/acs.jcim.1c00998](https://doi.org/10.1021/acs.jcim.1c00998).
- [384] Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. doi: [10.1021/acs.jcim.8b00839](https://doi.org/10.1021/acs.jcim.8b00839).
- [385] Philip Cohen, Darren Cross, and Pasi A. Jänne. Kinase drug discovery 20 years after imatinib: progress and future directions. *Nature Reviews Drug Discovery*, 20(7):551–569, may 2021. doi: [10.1038/s41573-021-00195-4](https://doi.org/10.1038/s41573-021-00195-4). URL <https://doi.org/10.1038/s41573-021-00195-4>.
- [386] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 11 2020. ISSN 0305-1048. doi: [10.1093/nar/gkaa1100](https://doi.org/10.1093/nar/gkaa1100). URL <https://doi.org/10.1093/nar/gkaa1100>.

- 
- [387] North Carolina USA Blue Ridge Institute for Medical Research in Horse Shoe. FDA-approved small molecule protein kinase inhibitors. <http://www.brimr.org/PKI/PKIs.htm>, 2022. [Online; accessed 01-February-2022].
- [388] G. Manning, D. B. Whyte, R. Martinez, T. Hunter, and S. Sudarsanam. The protein kinase complement of the human genome. *Science*, 298(5600):1912–1934, 2002. doi: 10.1126/science.1075762. URL <https://www.science.org/doi/abs/10.1126/science.1075762>.
- [389] Richard Morphy. Selectively Nonselective Kinase Inhibition: Striking the Right Balance. *J. Med. Chem.*, 53(4):1413–1437, 2009. doi: 10.1021/JM901132V.
- [390] Dominique Sydow, Eva Aßmann, Albert J. Kooistra, Friedrich Rippmann, and Andrea Volkamer. Kissim: Predicting off-targets from structural similarities in the kinome. *Journal of Chemical Information and Modeling*, 62(10):2600–2616, 2022. doi: 10.1021/acs.jcim.2c00050.
- [391] Christine Yueh, Justin Rettenmaier, Bing Xia, David R. Hall, Andrey Alekseenko, Kathryn A. Porter, Krister Barkovich, Gyorgy Keseru, Adrian Whitty, James A. Wells, Sandor Vajda, and Dima Kozakov. Kinase atlas: Druggability analysis of potential allosteric sites in kinases. *J. Med. Chem.*, 62(14):6512–6524, 2019. doi: 10.1021/acs.jmedchem.9b00089.
- [392] Mazen W Karaman, Sanna Herrgard, Daniel K Treiber, Paul Gallant, Corey E Atteridge, Brian T Campbell, Katrina W Chan, Pietro Cicceri, Mindy I Davis, Philip T Edeen, Raffaella Faraoni, Mark Floyd, Jeremy P Hunt, Daniel J Lockhart, Zdravko V Milanov, Michael J Morrison, Gabriel Pallares, Hitesh K Patel, Stephanie Pritchard, Lisa M Wodicka, and Patrick P Zarrinkar. A quantitative analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 26(1):127–132, jan 2008. doi: 10.1038/nbt1358. URL <https://doi.org/10.1038/nbt1358>.
- [393] Sameh Eid, Samo Turk, Andrea Volkamer, Friedrich Rippmann, and Simone Fulle. KinMap: a web-based tool for interactive navigation through human kinome data. *BMC Bioinformatics*, 18(1), jan 2017. doi: 10.1186/s12859-016-1433-7. URL <https://doi.org/10.1186/s12859-016-1433-7>.
- [394] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter

- notebooks - a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016. doi: 10.3233/978-1-61499-649-1-87.
- [395] List of Python introduction resources. <https://github.com/volkamerlab/teachopencadd#python-programming-introduction>, 2022. [Online; accessed 21-February-2022].
- [396] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2.
- [397] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.
- [398] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/mcse.2007.55. URL <https://doi.org/10.1109/mcse.2007.55>.
- [399] RDKit. RDKit: Open-Source Cheminformatics, 2022. URL <http://www.rdkit.org>. [Online; accessed 2022-02-02].
- [400] Volkamerlab. KiSSim open-source Python package. <https://github.com/volkamerlab/kissim>, 2022. [Online; accessed 01-February-2022].
- [401] requests. requests. <https://docs.python-requests.org/>, 2022. URL <https://docs.python-requests.org/>. [Online; accessed 2022-02-02].

- 
- [402] TeachOpenCADD conda-forge package. <https://anaconda.org/conda-forge/teachopencadd>, 2022. [Online; accessed 2022-02-02].
- [403] TeachOpenCADD. TeachOpenCADD installation instructions. <https://projects.volkamerlab.org/teachopencadd/installing.html>, 2021. URL <https://projects.volkamerlab.org/teachopencadd/installing.html>. [Online; accessed 2021-10-06].
- [404] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Navarte, Andrew Osherooff, M Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan Kelley, and Carol Willing. Binder 2.0 - Reproducible, Interactive, Sharable Environments For Science At Scale . In Fatih Akici, David Lippa, Dillon Niederhut, and M Pacer, editors, *Proceedings of the 17th Python in Science Conference*, pages 113 – 120, 2018. doi: 10.25080/Majora-4af1f417-011. [doi:10.25080/Majora-4af1f417-011].
- [405] Oscar P. J. van Linden, Albert J. Kooistra, Rob Leurs, Iwan J. P. de Esch, and Chris de Graaf. KLIFS: A knowledge-based structural database to navigate kinase–ligand interaction space. *Journal of Medicinal Chemistry*, 57(2):249–277, sep 2013. doi: 10.1021/jm400378w. URL <https://doi.org/10.1021/jm400378w>.
- [406] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, 02 1999. ISSN 1741-0126. doi: 10.1093/protein/12.2.85. URL <https://doi.org/10.1093/protein/12.2.85>.
- [407] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, nov 1992. doi: 10.1073/pnas.89.22.10915. URL <https://doi.org/10.1073/pnas.89.22.10915>.
- [408] Patrick Kunzmann and Kay Hamacher. Biotite: a unifying open source computational biology framework in python. *BMC Bioinformatics*, 19(1), oct 2018. doi: 10.1186/s12859-018-2367-z. URL <https://doi.org/10.1186/s12859-018-2367-z>.
- [409] TeachOpenCADD. TeachOpenCADD website. <https://projects.volkamerlab.org/teachopencadd/>, 2022. [Online; accessed 2022-02-02].
- [410] Sven Kosub. A note on the triangle inequality for the jaccard distance. *Pattern Recognition Letters*, 120:36–38, 2019. ISSN 0167-8655. doi: <https://doi.org/10.1016/j.patrec.2018.12.007>. URL <https://www.sciencedirect.com/science/article/pii/S0167865518309188>.

- 
- [411] Sarah Barelier, Teague Sterling, Matthew J. O'Meara, and Brian K. Shoichet. The recognition of identical ligands by unrelated proteins. *ACS Chemical Biology*, 10(12):2772–2784, 2015. doi: 10.1021/acscchembio.5b00683. URL <https://doi.org/10.1021/acscchembio.5b00683>.
- [412] Kinodata. <https://github.com/openkinome/kinodata>, 2022. [Online; accessed 01-February-2022].
- [413] OpenKinome. <http://openkinome.org/>, 2022. [Online; accessed 01-February-2022].
- [414] Benjamin Merget, Samo Turk, Sameh Eid, Friedrich Rippmann, and Simone Fulle. Profiling prediction of kinase inhibitors: Toward the virtual assay. *Journal of Medicinal Chemistry*, 60(1):474–485, 2017. doi: 10.1021/acs.jmedchem.6b01611. URL <https://doi.org/10.1021/acs.jmedchem.6b01611>.
- [415] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009. doi: 10.1007/978-0-387-84858-7. URL <https://doi.org/10.1007/978-0-387-84858-7>.
- [416] Petra Schneider, W. Patrick Walters, Alleyn T. Plowright, Norman Sieroka, Jennifer Listgarten, Robert A. Goodnow, Jasmin Fisher, Johanna M. Jansen, José S. Duca, Thomas S. Rush, Matthias Zentgraf, John Edward Hill, Elizabeth Krutoholow, Matthias Kohler, Jeff Blaney, Kimito Funatsu, Chris Luebke, and Gisbert Schneider. Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery*, 19(5):353–364, dec 2019. doi: 10.1038/s41573-019-0050-3. URL <https://doi.org/10.1038/s41573-019-0050-3>.
- [417] Denis Schmidt, Magdalena M. Scharf, Dominique Sydow, Eva Aßmann, Maria Martí-Solano, Marina Keul, Andrea Volkamer, and Peter Kolb. Analyzing kinase similarity in small molecule and protein structural space to explore the limits of multi-target screening. *Molecules*, 26(3):629, jan 2021. doi: 10.3390/molecules26030629. URL <https://doi.org/10.3390/molecules26030629>.
- [418] conda-forge community. The conda-forge Project: Community-based Software Distribution Built on the conda Package Format and Ecosystem, July 2015.
- [419] World Health Organization. <https://covid19.who.int/>, 2022. [Online; accessed 01-May-2022].
- [420] COVID Moonshot. <https://postera.ai/moonshot>, 2022. [Online; accessed 01-May-2022].

- [421] Jean-Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3):722–730, 2015. doi: 10.1021/ar500432k. URL <https://doi.org/10.1021/ar500432k>. PMID: 25687211.
- [422] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 0031-3203. doi: 10.1016/S0031-3203(96)00142-2.
- [423] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010.
- [424] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), jan 2020. doi: 10.1186/s12864-019-6413-7. URL <https://doi.org/10.1186/s12864-019-6413-7>.
- [425] Tarald O. Kvålseth. Cautionary note about  $r^2$ . *The American Statistician*, 39(4):279–285, 1985. doi: 10.1080/00031305.1985.10479448.
- [426] Arlene Ash and Michael Shwartz. R2: a useful measure of model performance when predicting a dichotomous outcome. *Statistics in Medicine*, 18(4):375–384, 1999. doi: 10.1002/(SICI)1097-0258(19990228)18:4<375::AID-SIM20>3.0.CO;2-J.
- [427] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-00296-0\_5.



# Declaration of authorship

Name: Kimber

First name: Talia B.

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.

Date: September 2022

Signature:



## Zusammenfassung

Krebserkrankungen sind eines der größten Probleme des Gesundheitswesens und verursachen jedes Jahr mehrere Millionen Todesfälle. Die computergestützte Arzneimittelforschung ist zu einem Eckpfeiler für die Entwicklung von Therapien für bestehende und neu auftretende Krankheiten geworden. Sie zielt nicht nur darauf ab, den Prozess der Arzneimittelenwicklung zu beschleunigen, sondern auch kostspielige Experimente und In-vivo-Tierversuche zu reduzieren. Innerhalb der letzten zehn Jahre hat Deep Learning eine wichtige Rolle bei der Vorhersage von molekularer Aktivität, Eigenschaften und Toxizität eingenommen.

Wir haben Techniken zur Daten-Augmentation entwickelt, die auf der SMILES-Kodierung von Molekülen basieren, und sie auf drei Deep-Learning-Modelle sowie auf vier Eigenschafts- und Aktivitätsdatensätze angewendet. Die Ergebnisse zeigen, dass die Datenerweiterung die Modellgenauigkeit unabhängig vom Deep-Learning-Modell und der Größe des Datensatzes verbessert. Die Berechnung der Unsicherheit des Modells mit Hilfe der Augmentation zum Zeitpunkt der Inferenz hat gezeigt, dass der Fehler umso kleiner ist, je sicherer das Modell ist. Das bedeutet, dass einer gegebenen Vorhersage vertraut werden kann und sie nahe am Zielwert liegt.

Um besser zu verstehen, wie ein neuronales Netzwerk eine Substanz auf der Grundlage ihrer Input-Merkmale klassifiziert, haben wir die inneren Schichten eines tiefen neuronalen Netzwerks zerlegt, um die toxischen Substrukturen einer Substanz zu identifizieren, und eine Methode entwickelt, um sie in 2D zu visualisieren. Das Deep-Learning-Modell erreicht nicht nur Ergebnisse auf dem neuesten Stand der Technik, sondern die identifizierten Toxikophore bestätigen bekannte toxische Substrukturen und liefern neue potenzielle Kandidaten.

Um den Prozess der Arzneimittelforschung zu beschleunigen, ist der Zugang zu robusten und modularen Arbeitsabläufen äußerst vorteilhaft. In diesem Zusammenhang wurde das vollständig quelloffene TeachOpenCADD-Projekt mit einer speziellen Pipeline für Kinasen entwickelt —eine Familie von Proteinen, von denen bekannt ist, dass sie an Krankheiten wie Krebs beteiligt sind. Es wurden vier Maßstäbe für die Ähnlichkeit von Kinasen implementiert, die Sequenz- und Strukturinformationen sowie Protein-Ligand-Interaktions- und Ligandenprofilierungsdaten berücksichtigen und die Analyse von Off-Target-Effekten von Inhibitoren ermöglichen. Die Ergebnisse zeigen, dass die Analyse von Kinasen aus verschiedenen Blickwinkeln entscheidend für den Einblick in die Off-Target-Vorhersage ist.

Diese neuartigen Methoden können bei der Entdeckung neuer Arzneimittel und insbesondere bei Krankheiten genutzt werden, die mit einer Dysregulation von Kinasen einhergehen, wie z. B. Krebserkrankungen.