Deep Learning Approaches for Pre-Clinical Drug Discovery

Fergus Imrie

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary Term 2021

Acknowledgements

When I started my DPhil in 2017, I knew it would be one of the most challenging but rewarding experiences of my life. To say I could never have expected what followed is most certainly an understatement. I am incredibly grateful to everyone who has helped me along the way and has made the last three and a half years so special.

First, I would like to thank my supervisors, without whom this thesis would not have been possible. Charlotte, none of this would have happened without you, and for that I will be forever grateful. Thanks for taking a chance on me and making me the scientist that I am today. Anthony, thank you for all of your encouragement, for challenging everything I say, and for never hesitating to call me out. Thanks for broadening my academic horizons and trying your best to make my chemistry knowledge at least passable. Mihaela, thanks for always encouraging me to be the change I wanted to see in the world and to think BIG.

I would like to thank everyone in OPIG for being so welcoming; you are the research group I never knew I needed. You have been a constant sounding board for all of my ideas and an almost infinite source of knowledge. You make the Department of Statistics the only place I would ever want to have undertaken my DPhil. In particular, to everyone in 2.17; I will miss the nerf wars, literary readings, keyboard bashing, and G&T Fridays.

From my second year, I have been fortunate to have an amazing industrial partner in Exscientia, providing me a taste of drug discovery in action.

A special mention must also go to Professor Christina Goldschmidt, without whose support and advice I might not have embarked on this journey.

Finally, to my parents. I cannot imagine how my world would look without having had you in it. You have supported my every move and have made me who am I today.

Abstract

Deep learning methods have experienced a revolution, driven by their successful application in fields such as computer vision and natural language processing. In this thesis, we describe several novel methodologies leveraging deep learning for applications to pre-clinical drug discovery.

First, we propose a generative approach to the design of molecular linkers which incorporates basic 3D information. In large-scale tests, we find that our method substantially outperforms a database-based approach, the previous *de facto* approach for this problem. Through a series of case studies, we demonstrate the application of our approach to scaffold hopping, fragment linking and PROTAC design. We then extend this framework to incorporate physically-meaningful 3D structural information, providing a richer prior for the generative process, and also apply our method to molecular elaboration tasks, such as R-group design.

We then turn our attention to predictive modelling, in particular structure-based virtual screening. We find that the advances in convolutional neural networks (CNNs) for general computer vision tasks are applicable to structure-based virtual screening. In addition, we propose two techniques to incorporate domain-specific knowledge into this framework. First, we show that limitations in docking necessitate the use of multi-pose scoring and demonstrate the benefits of an average scoring policy. Second, we propose a transfer learning approach to construct protein family-specific models, utilising knowledge of the differences between protein families.

Finally, we investigate how a generative approach can be used to improve the training and benchmarks sets employed in structure-based virtual screening. We propose a deep learning method that generates decoys to a user's preferred specification in order to control decoy bias or construct sets with a defined bias. We show that our approach significantly reduces the bias contained in such sets. We validate that our generated molecules are more challenging for docking-based approaches to separate from bioactive compounds than previous decoys. In addition, we show that CNN-based structure-based virtual screening methods can be trained on such compounds.

Contents

List of Figures

List of Tables

List of Abbreviations

1-D, 2-D, 3D . One-, two-, or three- dimensional.

Å . . . . . Ångström.

ADME . . . . Absorption, distribution, metabolism, and excretion.

ANDR . . . . Androgen receptor.

ATP . . . . . Adenosine triphosphate.

AUC . . . . . Area under curve.

AVE bias . . . Asymmetric validation embedding bias.

BAF . . . . . Barrier-to-autointegration factor.

BindingDB . . The Binding Database.

CADD . . . . . Computer-aided drug design.

cAMP . . . . . Cyclic adenosine monophosphate.

CASF . . . . . Comparative Assessment of Scoring Functions.

ChEMBL . . . Chemical database curated by the European Molecular Biology Laboratory.

COVID-19 . . Coronavirus disease 2019, the disease caused by severe acute respiratory syndrome coronavirus 2

CNN . . . . . Convolutional neural network.

CYP450 . . . . Cytochrome P450.

Da . . . . . Dalton.

DEKOIS . . . Demanding Evaluation Kits for Objective *In Silico* Screening.

DenseNet . . . Densely connected convolutional neural network.

DOE . . . . . Deviation from optimal embedding.

DUD . . . . . Database of Useful Decoys.

DUD-E . . . . Database of Useful Decoys: Enhanced.

EC50 . . . . . Half maximal effective concentration.

ECFP . . . . . Extended connectivity fingerprint.

EF . . . . . Enrichment factor.

FA7 . . . . . Coagulation factor VII.

FBDD . . . . . Fragment-based drug discovery or fragment-based drug design.

FCFP . . . . . Functional class fingerprints.

FPR . . . . . False positive rate.

GAN . . . . . Generative adversarial network.

GGNN . . . . Gated-graph neural network.

GI50 . . . . . Half maximal cell growth inhibition.

Glu . . . . . Glutamic acid.

GNN . . . . . Graph neural network.

GPCR . . . . . G protein-coupled receptors.

hERG . . . . . Human ether-à-go-go-related gene

HTS . . . . . High-throughput screening.

IC50 . . . . . Half maximal inhibitory concentration.

ILSVRC . . . ImageNet Large Scale Visual Recognition Challenge.

IMPDH . . . . Inosine 5-monophosphate dehydrogenase.

JNK3 . . . . . c-Jun N-terminal kinase 3.

Kd . . . . . Dissociation constant.

Ki . . . . . Inhibitory constant.

KL . . . . . Kullback-Leibler.

LADS . . . . . Latent actives in the decoy set.

LBDD . . . . . Ligand-based drug discovery or ligand-based drug design.

LBVS . . . . . Ligand-based virtual screening.

M . . . . . Molar.

MLL . . . . . Mixed lineage leukemia fusion protein.

MMPA . . . . Matched molecular pair analysis.

MPNN . . . . Message-passing neural network.

MUBD . . . . Maximal Unbiased Benchmarking Data Sets.

MUV . . . . . Maximum Unbiased Validation data sets.

NRAM . . . . Neuraminidase.

p38 . . . . . p38 mitogen-activated protein kinase.

PAINS . . . . Pan-assay interference compounds.

PDB . . . . . Protein Data Bank.

PRC . . . . . Precision recall curve.

PROTAC . . . Proteolysis targeting chimera.

QSAR . . . . . Quantitative structure activity relationship.

ReLU . . . . . Rectified linear unit.

ResNet . . . . Residual neural network.

RMSD . . . . . Root-mean-square deviation.

RNN . . . . . Recurrent neural network.

ROC . . . . . Receiver operating curve.

SA . . . . . Synthetic accessibility.

SAHH . . . . . Adenosylhomocysteinase.

SAR . . . . . Structure-activity relationship.

SBDD . . . . . Structure-based drug discovery or structure-based drug design.

SBVS . . . . . Structure-based virtual screening.

$\mathbf{SC_{RDKit}}$ . . . . A shape and colour similarity score based on RDKit functions.

SGD . . . . . Stochastic gradient descent.

SMARCA2 . . SWI/SNF Related, Matrix Associated, Actin Dependent Regulator Of Chromatin, Subfamily A, Member 2.

SMARCA4 . . SWI/SNF Related, Matrix Associated, Actin Dependent Regulator Of Chromatin, Subfamily A, Member 4.

SMILES . . . . Simplified molecular-input line-entry system.

SWI/SNF . . Switch/Sucrose Non-Fermentable.

TPR . . . . . True positive rate.

VAE . . . . . Variational autoencoder.

VHL . . . . . Gene that encodes von Hippel-Lindau tumor suppressor protein.

ZINC . . . . . ZINC Is Not Commercial.

Every new beginning comes from some other beginning's end.

— Lucius Annaeus Seneca

1 Introduction

Contents

This chapter contains material from the following book chapter:

Fergus Imrie, Anthony R. Bradley, and Charlotte M. Deane (2021). Virtual Screening with Convolutional Neural Networks in Brown, N. (ed.) *Artificial Intelligence in Drug Discovery*, The Royal Society of Chemistry, pp. 151-183.

1.1 Motivation

Drugs are key therapeutic treatments for a wide variety of medical conditions and are an essential building block of a functioning health system (World Health

Organization, 2010). However, there are many medical needs, both existing and emergent, that are currently unmet by existing medicines (Kaplan et al., 2013). The ability to rapidly and effectively address unmet medical needs as and when they occur has been further highlighted by the recent worldwide emergency caused by the current coronavirus pandemic (COVID-19, Rosa et al., 2020).

Developing new therapeutics is an extremely challenging, multi-stage process, involving many disciplines and typically requiring many years to complete. On average each new therapeutic is estimated to cost $1.5-3 billion, depending on how this is calculated, (Avorn, 2015; DiMasi et al., 2016) and takes over ten years (Paul et al., 2010). On average, the FDA has approved 31 novel drugs per year from 2008-16 (U.S. Food and Drug Administration, 2018a). These figures are not improving and, as such, current practices have been called unsustainable (Moors et al., 2014; Ernst & Young, 2017).

Much of the cost of drug discovery arises from the high chance of failure, with the investment of sufficient time and financial resource far from a guarantee of success. A recent study found that only 13.8% of all drug development programs eventually lead to approval while the overall success rate for drugs that treat rare diseases, also known as 'orphan drugs', was as low as 6.2% (Wong et al., 2018). Failures occur for a variety of reasons, which we will discuss in more detail in Section 1.3.1. The high cost and low productivity in drug development is a long-standing problem, for which a solution is critically important (Myers and Baker, 2001).

Computer-aided drug design (CADD) is seen as having the potential to accelerate this process and reduce the expense of developing new therapeutics (Ou-Yang et al., 2012). However, despite broad adoption of computational methodologies across the entire drug discovery workflow, costs have continued to increase (DiMasi et al., 2003; Avorn, 2015; DiMasi et al., 2016) with sustained low productivity (Khanna, 2012). New techniques and approaches are still sorely needed to revolutionise drug discovery.

Recently, there has been renewed interest in the use of artificial intelligence across a broad range of fields, driven by the rise of deep learning. While many of the core principles of deep learning were introduced decades ago (e.g. Rosenblatt,

1958; Fukushima, 1980; Rumelhart et al., 1986), it was not until 2012 that the power and effectiveness of such techniques was demonstrated, in what is now known as the "ImageNet moment". In the annual ImageNet Large Scale Visual Recognition Challenge, Krizhevsky et al. (2012) performed 41% better than the next best competitor by adopting a deep neural network. It is widely acknowledged that this breakthrough was made possible by the combination of unprecedented availability of labelled data and computational power. This has led to learning-based systems matching, and indeed often surpassing, humans at image recognition (He et al., 2015), single-player games (Mnih et al., 2015), and two-player games including Go (Silver et al., 2016; Silver et al., 2017), Chess (Silver et al., 2018), and StarCraft II (Vinyals et al., 2019).

These advances quickly caught the attention of the field of cheminformatics, with several early promising results reported. In 2013, deep neural networks were the best performing models in the Merck molecular activity challenge (Ma et al., 2015) while a similar result was obtained in the Tox21 toxicity data challenge in 2015 (Mayr et al., 2016).

Learning-based algorithms have a long history in drug discovery. Early quantitative structure activity relationship (QSAR) models were first described in the early 1960s (Hansch et al., 1962) and have become commonplace (Salt et al., 1992). However, traditional machine learning and classical statistical approaches typically require the explicit featurisation of the target input, such as molecules or protein-ligand complexes, in the form of a one-dimensional vector (Klambauer et al., 2019). This requirement has resulted in the development of hundreds of descriptors for molecular property prediction alone (e.g. Deng et al., 2004; Zhang et al., 2006; Durrant and McCammon, 2011). However, an advantage of deep learning methods that is seen as key to their success is the ability to eliminate the need for abstraction and allow a much broader variety of data types be learnt from directly (Klambauer et al., 2019).

Finally, the QSAR models discussed above are typically bespoke models, constructed within the context of a specific drug discovery project based on a small

amount of data. Thus, while useful, they do not have general applicability, and often do not extend beyond a specific chemical series. Success in other domains (e.g. ImageNet, Deng et al., 2009) has shown that a key requirement for general-purpose models is sufficient data (Halevy et al., 2009; Sun et al., 2017). Over the past decade, there has been a rapid increase in the amount of publicly available molecular activity and biochemical data (e.g. Kim et al., 2015; Papadatos et al., 2015), as well as structural data (Berman et al., 2000; Burley et al., 2019), largely due to increased focus and the emergence of new experimental techniques (e.g. high-throughput screening, Inglese et al., 2007).

An example of the benefits of the availability of such data that would not have been possible otherwise are the recent successes in the field of protein-structure prediction, culminating in the performance of AlphaFold (Senior et al., 2020) and AlphaFold 2 (Jumper et al., 2020) in CASP 13 and 14, respectively (Kryshtafovych et al., 2019). While there are many challenges in applying deep learning to drug discovery that are discussed further in Section 1.5, this is an incredibly encouraging development further highlighting the promise that deep learning holds.

This thesis focuses on developing deep learning methodologies for the drug discovery process. In this chapter, we begin by discussing several key developments in machine learning and provide an introduction to two general deep learning methods which can be utilised in cheminformatics. We then briefly summarise the drug discovery process and discuss how computational methods are used in drug discovery. We focus our discussion on how such methods can be used to screen large virtual libraries of compounds for initial hit molecules. We then describe how computational, and in particular machine learning-based, methods can be used to design new compounds and highlight several of the common challenges of molecule design. We finish by outlining the structure and key contributions of this thesis.

1.2 Machine Learning

Machine learning attempts to learn patterns directly from data without explicit functional pre-specification for use in prediction, decision making, or other out-

comes of interest (Mitchell, 1997; Murphy, 2012). These methodologies are often classified into several paradigms: supervised learning, unsupervised learning, and reinforcement learning. However, these paradigms are not mutually exclusive, and there are many connections between them.

In supervised learning we are interested in fitting a function $f : X \to Y$ using a data set, D, of n labelled observations

$$D = \{(x_i, y_i), i = 1 \dots n\}, \tag{1.1}$$

where $x_i \in X$ and $y_i \in Y$. Typical applications include regression and classification tasks.

In unsupervised learning, we do not have access to labels and thus our data set, D, consists of only observations of the source domain X, reducing to

$$D = \{x_i, i = 1 \dots n\}. \tag{1.2}$$

In this paradigm, the goal is to find some notion of internal structure or common featurisation. Clustering (Lloyd, 1982), anomaly detection (Hodge and Austin, 2004), and dimensionality reduction techniques (Maaten et al., 2007) are common unsupervised methodologies.

Finally, reinforcement learning aims to learn an optimal policy for an agent in a environment, given some notion of reward. While many formulations exist, a basic and natural one is

$$D = \{(s_i, a_i, r_i), i = 1 \dots n\}, \tag{1.3}$$

where $s_i \in S$ is the state of the system of environment, $a_i \in A$ is the action taken by the agent, and $r_i \in R$ is the reward given for taking action a_i in state s_i.

For the work presented in this thesis we mostly are concerned with the supervised and unsupervised paradigms, with a particular emphasis on deep learning-based models and their applications to drug discovery. In the remainder of this section, we will introduce two broad categories of machine learning algorithms that are applicable within any of the paradigms outlined above. The first, convolutional

neural networks (CNNs), led to the "ImageNet moment" discussed previously. We will utilise CNNs in Chapter 3 for generative modeling and Chapter 4 for predictive modelling. The second, graph neural networks, extend modern deep learning methods beyond the regular structure of Euclidean domains. We will demonstrate their application to generative modelling in Chapters 2, 3, and 5.

1.2.1 Convolutional Neural Networks

CNNs: A Primer

CNNs represent a class of deep learning methods that are designed to capture local and spatial connectivity. CNNs have been extensively applied in computer vision for analysing images; they have also been applied to other areas, such as neural machine translation (Gehring et al., 2017) and latent semantic modelling (Shen et al., 2014).

The precise architecture of CNNs can vary greatly, even for models designed for use on the same data set; however, CNNs share many commonalities, and are constructed from smaller building blocks, known as layers. All CNNs contain an input and an output layer, as well as multiple hidden layers (Figure 1.1). The hidden layers of CNNs typically consist of convolutional layers, pooling layers, and fully connected layers, together with activation functions and normalisation layers. It is the different combinations of these layers and how they are connected that leads to the differences between networks.

Input layer. The input layer is where data is fed into the model. CNNs take input with fixed spatial dimensions (e.g. the height and width of an image) and a fixed depth dimension (e.g. 1 for a greyscale image or 3 for a standard RGB image).

Convolutional layers. A convolutional layer applies a series of convolution operations to its input to produce an output. Convolutional layers consist of convolutional filters, or kernels, which are convolved with the input to produce feature maps. Mathematically, a convolution is an operation on two functions returning a function that represents how the shape of one is modified by the other. In a CNN, the convolution operation is represented by matrix operations,

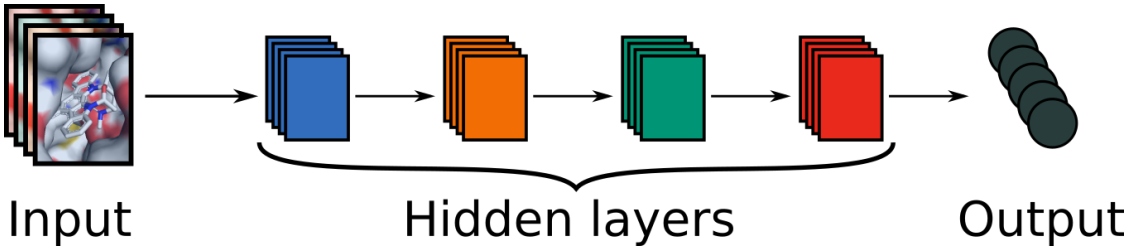


Figure 1.1: Overview of the basic format of a neural network architecture. The input is processed by a number of hidden layers to produce the output.

specifically the dot products of portions of the input to a convolutional layer and each of the layer's filters (Figure 1.2a).

Convolutional filters are small, learnt, matrices, with dimensions equal to the size of the receptive field of the layer. The dimension of a convolutional filter determines how much of the input the filter acts upon. For example, a 3×3 filter acts on a square of pixels with side length 3. Each filter is passed over the input in an independent manner, producing an activation map or feature map. These activation maps are then concatenated to form the output of the convolutional layer. An example of the convolution operation can be seen in Figure 1.2a.

Pooling layers. Pooling layers are used to reduce the spatial size of the representation passing through the network. This is done both to reduce the number of parameters and computation in the network, and also to control overfitting. In addition, this reduction of resolution helps to make the representation invariant to small translations of the input (Goodfellow et al., 2016). Pooling layers are commonly inserted between successive convolutional layers, or after a block of layers. The pooling layer operates independently on every depth slice of the input and resizes it spatially, typically using either the maximum or average operation of the pooling layer's receptive field. The most common form of pooling layer uses the maximum operation, and has filters of size 2×2 with a stride of 2 (the size of the step the filter moves each time); this results in a downsampling of every depth slice in the input by 2 along both width and height, discarding 75% of the activations by taking only the maximum in each 2×2 square of the input (Figure 1.2b). A standard pooling operation does not affect the depth dimension, and

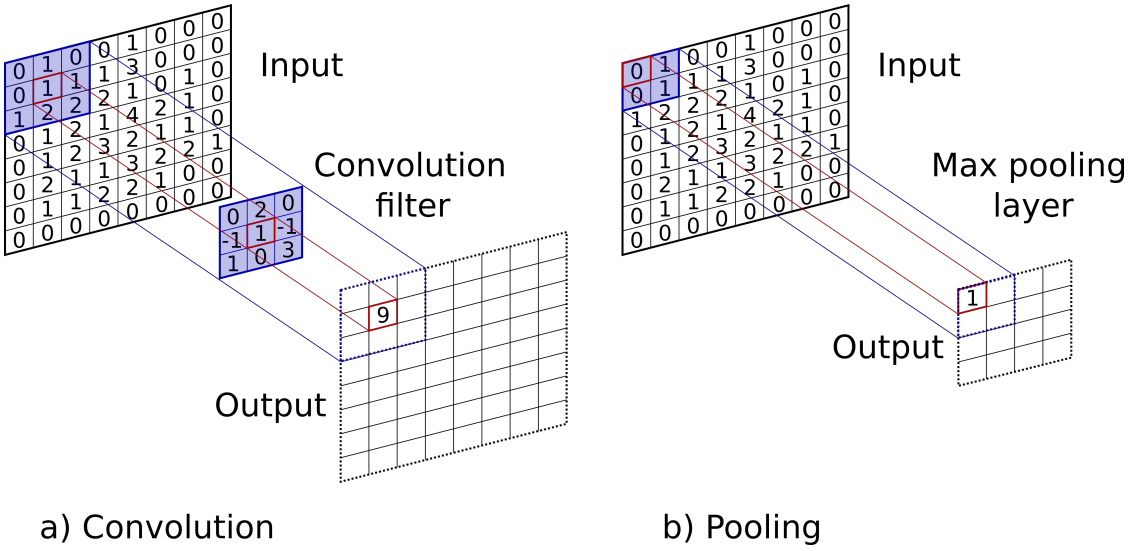


Figure 1.2: Illustration of the two most common layers used in CNNs. a) A 3×3 convolution is passed across the input to the layer. The output pixel is a weighted sum of the source pixel and the elements around it. b) A 2×2 max pooling operation with stride 2 results in a downsampling of the input from 8×8 to 4×4, taking the maximum element in each 2×2 grid.

acts on individual feature maps independently. It is possible to use any function to perform pooling, with common examples maximum, average and sometimes L2-norm pooling. Average pooling was often used historically but has recently fallen out of favor compared to the max pooling operation, which has been shown to work better in practice (Yu et al., 2014). Several alternative pooling methods have been proposed that claim improvement over max and average pooling (Yu et al., 2014; Lee et al., 2016), however they do not appear to have been generally adopted.

Other Layer Types. In addition to convolutional and pooling layers, there are several other types of commonly used layers. Here, we will briefly discuss activation and normalisation layers. More detail can be found in Goodfellow et al. (2016) or Gu et al. (2018). Activation layers are used to introduce non-linearities into the model. Without activation layers, neural networks would simply be linear functions of the input features. The most common form is the rectified linear unit, or ReLU (Nair and Hinton, 2010), which for every number in the input, takes the maximum of that number and zero as its output. Other commonly used activation functions are sigmoid, hyperbolic tangent, and leaky ReLU (Maas et al., 2013). Choosing

an activation function is normally determined by a hyperparameter search, or is selected to produce output within a specific range.

The initial input to machine learning models is often normalised in some way to control for the range of possible inputs. Normalisation layers can also be incorporated within a model to standarise the input to the subsequent layer in a particular way. Normalisation layers are not always included in CNNs; however, one commonly used example is batch normalisation (Ioffe and Szegedy, 2015). Batch normalisation aims to stabilise the distribution (over a mini-batch) of inputs to a given network layer during training, in order to make the optimisation landscape smoother (Santurkar et al., 2018). Batch normalisation layers are typically placed between convolutional layers, either before or after the non-linearity.

Training. Like most neural networks, CNNs are trained through backpropagation, which updates the parameters of the convolutional filters based on errors during the training process. The number of parameters in CNNs can be extremely large (>1,000,000) and parameters are not independent. Thus, finding optimal parameters is challenging. Stochastic gradient descent (Rumelhart et al., 1986), or other optimisers such as Adam (Kingma and Ba, 2015), are used to update the weights in the neural network based on small batches of data, known as minibatches. These optimisation techniques are stochastic in nature, thus the parameters of the trained model will depend on the random seed used during the training process. Ruder (2016) provides a recent overview of optimisation techniques for training neural networks.

ImageNet

While CNNs were first introduced in the 1980s as a translation equivariant method for pattern recognition (Fukushima, 1980) and found some early use cases in image recognition (for example LeNet for reading numbers on cheques, Lecun et al. 1998; LeCun et al. 1999), widespread use has only occurred more recently. Before 2012, traditional image processing and analysis techniques were used in computer vision, such as pyramid matching (Lazebnik et al., 2006) and saliency maps (Itti et al.,

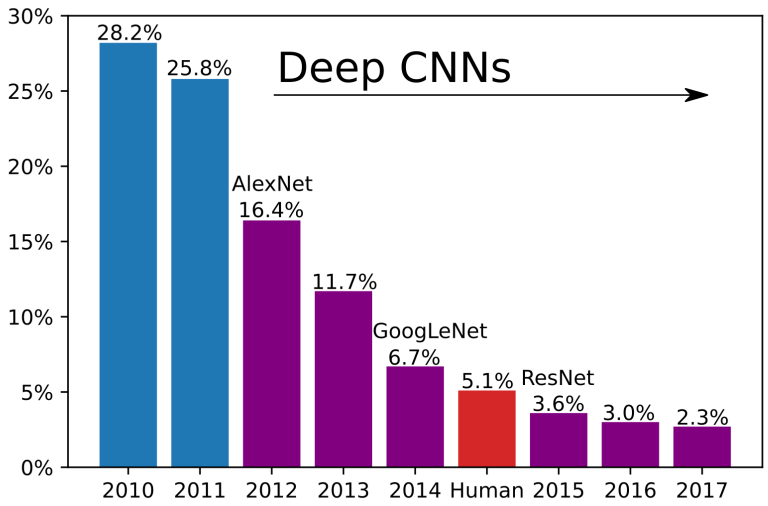


Figure 1.3: Top 5 error rate of the winning entries in the annual ImageNet competition since 2010. Deep CNNs have been responsible for the winning entry each year since 2012, with expert human performance surpassed in 2015.

1998). In contrast to CNNs, these often required substantial pre-processing of images, and did not learn the underlying features from scratch, but instead learnt how to combine and interpret pre-computed features. However, following their successful application as part of the annual ImageNet competition in 2012, CNNs have become the *de facto* method for almost all computer vision tasks.

The annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC, or ImageNet, Russakovsky et al., 2015) was an annual competition in object category classification and detection of hundreds of object categories and millions of images, that was introduced in 2010 and ran until 2017. Since its introduction, the competition has been the source of many of the substantial advances not just in computer vision, but in machine learning more broadly. Crucial to the development of more advanced algorithms was the release of the then largest annotated corpus of images that could be used for training, consisting of over 1.4 million images across 1,000 different classes. This compares with around 20,000 images from 20 classes in the previous largest data set.

The competition has led to substantial improvement in several tasks including image classification. For example, all entries in the first two years of the challenge

had top five error rates of over 25% at the image classification task, while the winner of the 2017 edition, SENets (Hu et al., 2017), achieved a 2.3% error rate (Figure 1.3). A key shift occurred in 2012, with the introduction of CNNs. This caused an drop in classification error rate from 26% to 16%. Since then, all winning entries have been based on convolutional architectures, each with their own algorithmic advances, some of which will be described below.

Architectures

The majority of winning entries since 2012 have incorporated substantial algorithmic advances, as opposed to either hyperparameter or model architecture optimisation, or increases in the quantity or indeed quality of training data. We will briefly highlight several of the architectural advancements that have allowed more powerful networks to be trained.

AlexNet (Krizhevsky et al., 2012). The first work that popularised CNNs in computer vision was AlexNet. AlexNet was submitted to ILSVRC in 2012 and significantly outperformed the second placed method, achieving a top 5 error of 16% compared to the runner-up which had 26% error. The network had a basic convolutional architecture, but was deeper (more layers) and wider (more filters per layer) than previously used networks. The model also stacked convolutional layers on top of each other whereas previously it was common to have pooling layer immediately following each convolutional layer.

GoogLeNet (Szegedy et al., 2014). The ILSVRC 2014 winning entry utilised a novel way of combining convolutional layers, developing the Inception Module (Figure 1.4) that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M). By combining convolutions with different kernel sizes, their model is able to extract features of different sizes at each point. Additionally, the authors used average pooling instead of fully connected layers at the end of their model, eliminating a large number of parameters that did not seem to impact performance. There have been several subsequent versions of GoogLeNet, most recently Inception-v4 (Szegedy et al., 2017).

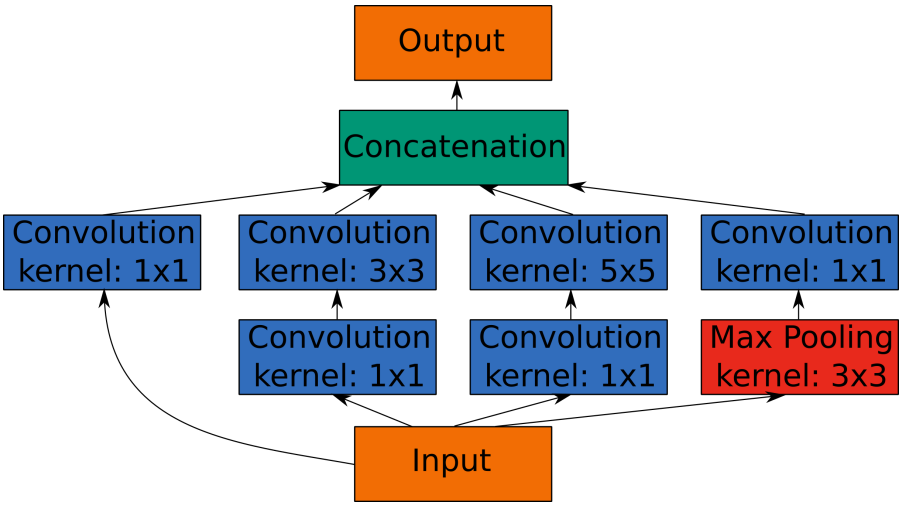


Figure 1.4: Architecture of the Inception Module, the main building block of the GoogLeNet (Szegedy et al., 2014) model, the winning entry of the 2014 ImageNet competition.

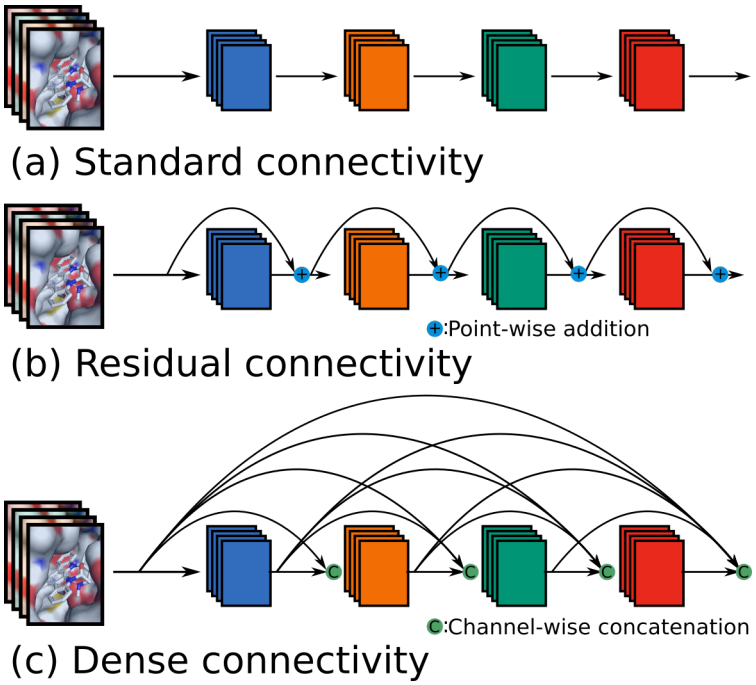


Figure 1.5: Illustration of connectivity in a standard convolutional neural network (a) compared to a ResNet (b) and DenseNet (c).

ResNet (He et al., 2016) and DenseNet (Huang et al., 2016). Residual networks (ResNets) developed by He et al. (2016) were the winner of ILSVRC 2015, while the DenseNet architecture was introduced by Huang et al. (2016) in 2016 and achieved state-of-the-art performance on several benchmarks. Both ResNet and DenseNet featured novel skip connections between layers and made substantial use of batch normalisation. The key architectural difference in ResNet and DenseNet compared with other convolutional networks is the fashion in which layers are connected (Figure 1.5). Normally, each layer will receive as input solely the output of the previous layer (standard connectivity). In a ResNet, a layer receives as input the sum of the outputs from the previous two layers (residual connectivity), while in a DenseNet, within each dense block, a layer receives the output of all prior layers within that block (dense connectivity). The ability for the output of a layer to skip the next allows feature maps of different depths and hence complexities to form, and for new feature maps to be learnt from a combination of existing maps of differing complexities. Furthermore, the residual and dense connections improve gradient flow during backpropagation (Rumelhart et al., 1986), allowing deeper models to be trained effectively while exhibiting substantially improved parameter efficiency (Huang et al., 2016).

CNNs can be applied to tasks within drug discovery by representing 3D structures as voxel grids. In this thesis, we explore the use of CNNs both for generating molecules (Chapter 3) and for structure-based virtual screening (Chapter 4).

1.2.2 Graph Neural Networks

A number of important learning tasks necessitate methods that can learn from graph-based data. Graphs can be used to represent of a broad range of systems spanning many disciplines including social science (e.g. social networks, Hamilton et al., 2017; Kipf and Welling, 2017), natural science (e.g. physical systems, Sanchez-Gonzalez et al., 2018; Battaglia et al., 2016; or protein-protein interaction networks, Fout et al., 2017) among other research areas. Of particular importance

in the context of this thesis is the natural application of graphs to chemistry and molecules (e.g. Kearnes et al., 2016).

As discussed in Sections 1.1 and 1.2.1, deep learning models have demonstrated great success for inputs such as speech, images, or video. The commonality between these inputs is that they all have an underlying Euclidean structure, i.e. they can be represented on an grid of dimension n (Figure 1.6a). This allows deep neural networks to leverage statistical properties of the data such as stationarity (i.e. shift invariance) and compositionality (e.g. larger objects are often composed from smaller ones, Bronstein et al., 2017). However, for non-Euclidean geometric data, of which graphs are one example (Figure 1.6b), this structure does not typically exist and thus new methods are needed to learn from such information (Bronstein et al., 2017). In this section we will describe graph neural networks, a category of learning algorithms designed for graphical applications.

Preliminaries

First, we must formalise the notion of a graph. Mathematically, a graph G is defined by a set of vertices or nodes, V, and edges, $E \subseteq V \times V$,

$$G = (V, E)\,. \tag{1.4}$$

For $V = v_1, \ldots, v_n$ a set of n vertices, let $e_{ij} = (v_i, v_j) \in E$ denote an edge pointing from v_i to v_j.

Two useful concepts to define are the neighbourhood of a node and the adjacency matrix of the graph. The neighbourhood of a node v is defined as $N(v) = \{u \in V | (v, u) \in E\}$. The adjacency matrix A is a $n \times n$ matrix with $A_{ij} \neq 0$ if $e_{ij} \in E$ and $A_{ij} = 0$ if $e_{ij} \notin E$. The adjacency matrix is particularly important since it fully describes the connectivity of a graph.

For simplicity, and due to their application to molecular tasks, we will typically consider only unweighted, undirected graphs. The graph being undirected implies that $(v_i, v_j) \in E \iff (v_j, v_i) \in E$. Unweighted means that edges are either present, or they are not, i.e. in the adjacency matrix, $A_{ij} = 1$ if $e_{ij} \in E$, $A_{ij} = 0$ if $e_{ij} \notin E$.

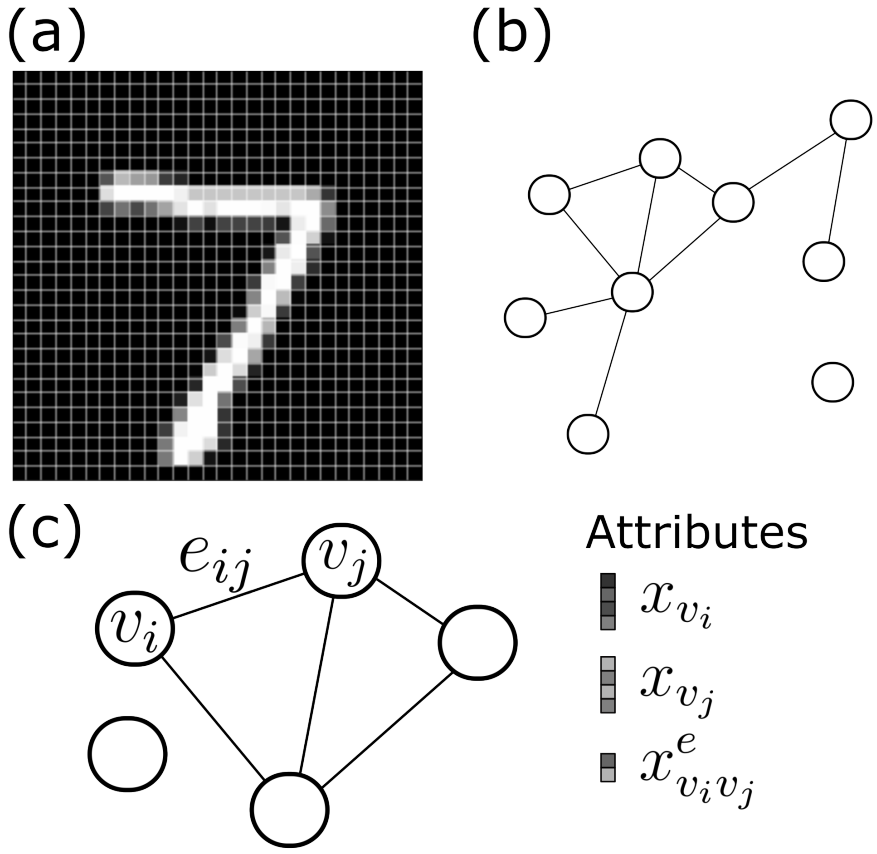


Figure 1.6: Illustrations of (a) image in Euclidean space, (b) graph in non-Euclidean space. (c) our definition of a graph.

This formulation of graphs can be extended by the introduction of *attributes*. An attributed graph has node attributes, X, and edge attributes, X^e. $X \in \mathbb{R}^{n \times d}$ is a node feature matrix with $x_v \in \mathbb{R}^d$ representing the feature vector of a node v. $X^e \in \mathbb{R}^{m \times c}$, is an edge feature matrix with $x^e_{vu} \in \mathbb{R}^c$ representing the feature vector of an edge (v, u). An illustration of an attributed graph is shown in Figure 1.6c.

These attributes can be used to used to label vertices and edges, for example to represent an atom or bond type, as well as used by learning frameworks to contain representations of vertices and edges.

Message Passing Neural Networks

Graph neural networks (GNNs) were first proposed in Scarselli et al. (2009) with the general goal of learning an embedding $h_v \in \mathbb{R}^s$ capturing the local structure of the graph around each node. The node embeddings can be used to produce a node-

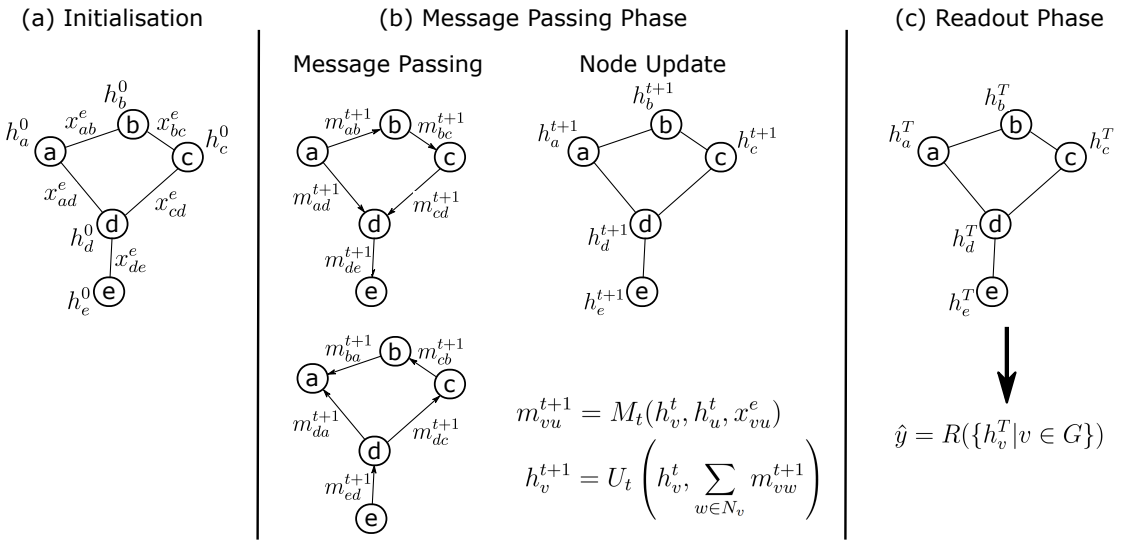


Figure 1.7: Illustration of the message passing algorithm. (a) Graph is initialised. (b) Message passing runs for T iterations. First messages are passed along the edges of the graph and then the nodes are updated according to those messages and the previous state of the node. (c) Finally, a readout function can be applied to the final states of the nodes.

level output, such as the node label, or aggregated with other node embeddings to produce a graph-level output.

Experimental results showed that GNNs were a powerful architecture for modeling structural data (Scarselli et al., 2009). However, the original GNN had many limitations, such as strong assumptions and limited representational capacity (Zhou et al., 2018). Numerous methods were proposed to tackle these limitations and allow for the use of GNNs on other types of graph (e.g. Schlichtkrull et al., 2018). For a more detailed discussion on GNNs, see the review articles by Bronstein et al. (2017), Zhou et al. (2018), Wu et al. (2019) and Zhang et al. (2020).

We will focus our discussion of GNNs on the message-passing neural networks (MPNNs) introduced by Gilmer et al. (2017), due to their use in the methods introduced in Chapters 2, 3, and 5 of this thesis. MPNNs unified a number of graph neural network and graph convolutional network approaches by analogy to message-passing in graphical models.

For simplicity, we describe MPNNs which operate on undirected graphs, G, with node features, x_v, and edge features, x_{vw}^e. The MPNN model contains two phases, a message passing phase and a readout phase (Figure 1.7). The

message passing phase, or propagation step, runs for T time steps and is defined in terms of a message function, M_t, and a node update function, U_t. At time $t \in \{1, \ldots, T\}$, using messages, m_v^t, and hidden states, h_v^t, (letting $h_v^0 = x_v$), the update functions are as follows:

$$\begin{aligned} m_v^{t+1} &= \sum_{w \in N_v} M_t(h_v^t, h_w^t, x_{vw}^e) \\ h_v^{t+1} &= U_t(h_v^t, m_v^{t+1}) \,. \end{aligned} \tag{1.5}$$

The readout phase then computes a feature vector, $\hat{y}$, for the whole graph by applying a readout function, R, to the hidden states at terminal time T according to

$$\hat{y} = R(\{h_v^T | v \in G\}) \,. \tag{1.6}$$

In this framework, the message functions, M_t, vertex update functions, U_t, and readout function, R, are all learned differentiable functions. We note that as R takes as input the set of node states, it must be invariant to permutations of the node ordering in order for MPNNs to be invariant to graph isomorphisms.

By altering the forms of M_t, U_t, and R, the MPNN framework is able to act as a generalised graph neural network, and indeed Gilmer et al. (2017) demonstrated how a number of existing models were captured by their approach. One such example is gated-graph neural networks (GGNNs, Li et al., 2016), which take the following settings:

$$\begin{aligned} M_t(h_v^t, h_w^t, x_{vw}^e) &= E_{e_{vw}} h_w^t \\ U_t(h_v^t, m_v^{t+1}) &= GRU(h_v^t, m_v^{t+1}) \\ R(\{h_v^T | v \in G\}) &= \sum_{v \in V} \sigma\left(i(h_v^T, h_v^0)\right) \odot \left(j(h_v^T)\right) \end{aligned} \tag{1.7}$$

where $E_{e_{vw}}$ is an edge-type specific neural network for the edge type of e_{vw}, GRU is the gated recurrent unit of Cho et al. (2014), i and j are neural networks. and $\sigma(x) = 1/(1 + e^{-x})$ is the logistic sigmoid function.

Generative Graph Neural Networks

Generating new graphs using graph neural networks is an important task but has a number of additional challenges compared to generating Euclidean data (Bronstein

et al., 2017). Due to the size of molecular graphs, much of the work on generative graph models have demonstrated their applications to molecule generation. Here we provide a brief overview of several approaches to graph generation. Specific applications for molecule design will be discussed further in Section 1.4.2.

NetGAN (Bojchevski et al., 2018) was one of the first works to propose a graph generative model using neural networks. They transformed the problem of graph generation to the problem of generating biased random walks. To do this, they took the random walks on a specific graph as input and trained a model to generate new random walks using a generative adversarial network (GAN, Goodfellow et al., 2014)-based architecture.

You et al. (2018b) took a different approach to graph generation and sought to generate the adjacency matrices of graphs. Their model, GraphRNN, did this in a sequential manner, generating the adjacency vector of each node in a step-by-step manner. There have been attempts to generate the adjacency matrix in a single step (e.g. De Cao and Kipf, 2018) with mixed success. Li et al. (2018) took a more granular approach than You et al. (2018b), proposing a model which sequentially generates individual nodes and edges. While this was a promising approach, conditioning each decision in the graph generation process on the entire generation sequence led to stability and scalability problems (Li et al., 2018). To address this issue, Liu et al. (2018) introduced CGVAE, a similar sequential graph generation model to You et al. (2018b) but with each step conditioned only on the current partial graph. In addition, they adopted a variational autoencoder setup (Kingma and Ba, 2015) and primarily applied their method to small molecule generation.

Similar sequential ideas have been combined with reinforcement learning to generate graphs under non-differential objectives and constraints (You et al., 2018a). These approaches model graph generation as a Markov decision process and the generative model is regarded as a agent operating in the graph generation environment.

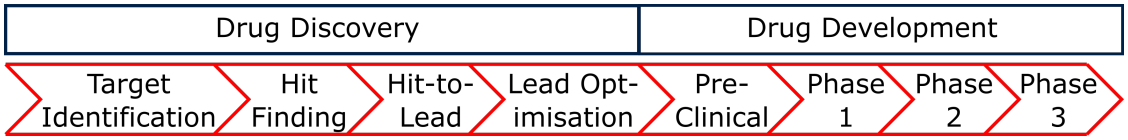


Figure 1.8: Overview of the various stages of therapeutic development, split into drug discovery and drug development.

1.3 Drug Discovery

Drugs are chemical or biological entities that affect the physiological state. They achieve this goal by modulating the functionality of one or more target biomolecules. The majority of drugs take the form of small organic molecules, or "ligands", although there is growing interest in biologics, such as antibodies (Nelson et al., 2010), nanobodies (Williams, 2013), and polypeptides (Craik et al., 2013). In this thesis, we will only consider small molecule drugs.

1.3.1 Drug Discovery Pipeline

The development of a new therapeutic agent can be divided functionally into two stages: discovery and development (Panchagnula and Thomas, 2000, Figure 1.8). Discovery is the process of proposing a candidate therapeutic and can be broadly grouped into target identification and validation, hit identification, and hit-to-lead optimisation. Development focuses on evaluating the safety and efficacy of new drug molecules in their target organism (typically humans), and consists of pre-clinical and clinical evaluation. The goal of drug development is to establish evidence for the effectiveness and safety profile of a drug, together with dosage and formulation intended for marketing.

Pre-Clinical Development

Target Identification and Validation

Target identification is the process of establishing a biological component for the drug to target, typically a protein, or proteins (Hughes et al., 2011). The target should be a critical driver of a disease, and thus by modulating the target the therapeutic will disrupt the pathway for that disease. This approach is known

as target-driven drug discovery (Brown, 2007); the other paradigm is phenotypic drug discovery. Instead of establishing the biological foundation for a disease, in phenotypic drug discovery the biological targets of the disease are only established after a compound that causes a desirable phenotypic change in a cell or animal model is identified (Kurosawa et al., 2008). In what follows we discuss the process of target-driven drug discovery.

Once the proteins implicated in a disease pathway have been identified, attempts are made to validate the targets and identify which protein(s) should be the focus of the drug discovery process. Target validation seeks to determine whether the biologically-relevant protein is essential in the pathway, i.e. whether modulating the target will positively affect the phenotype, and if the target is amenable to chemical modulation, often known as druggability (Hughes et al., 2011).

Hit Finding

After a drug discovery project has identified and validated a protein target, or targets, molecules which modulate the protein must be found. There are several well-established methods of hit finding for small molecules (Hughes et al., 2011). The simplest way is to search public databases and/or patent literature to find compounds known to bind the target of interest, or a highly similar one. If the target is well established, compounds with activity against the target will often have been reported. In some cases, small molecules drugs can be designed to mimic a known peptide binder (Hummel et al., 2006). However, in many settings, there is no previous data and thus other methods are required.

High-throughput screening (HTS) allows for hundreds of thousands to tens of millions of molecules to be tested in an assay to establish bioactivity against a target (Fox et al., 2006). However, HTS has several drawbacks: the assays used typically have relatively high false positive rates[1], experimentally screening up to hundreds of thousands of molecules is expensive, this type of "brute force" approach has very low hit rates as any given molecule has a very small probability of binding, and hits are often hard to optimise (Lahana, 1999; Ramesha, 2000).

[1]Even a very low false positive rate (e.g. 0.1%) would lead to many thousands of false positives given the number of molecules typically screened ($>$ 1 million).

Fragment-based drug discovery (FBDD) has become an increasingly important tool for finding hit compounds for difficult targets (Chen and Hubbard, 2009). In contrast to HTS, FBDD utilises smaller than drug-like compounds to identify low potency, high quality leads. Due to the use of much smaller molecules, typically molecular weight < 300 Da, fragments may only bind weakly to the target, but typically form high-quality interactions and display high ligand efficiency (Hopkins et al., 2004). In addition, the observed hit rates for fragment screens are 10–1,000 times higher than conventional high-throughput screens (Hajduk and Greer, 2007). While typically less potent than hits from HTS, fragment hits can be grown or combined to produce leads with high affinity (Murray and Rees, 2009).

Hit-to-Lead and Lead Optimisation

The goal of hit-to-lead and lead optimisation is primarily to modify the hit molecules to increase the potency and selectivity of the molecule against the target (Hughes et al., 2011). There are a number of other desirable features of a small molecule that are considered at this point in the process, several of which are typically specific to the project. These can include ensuring the molecule maintains oral bioavailability, or is able to cross the blood-brain barrier if the protein is found in the brain. This stage is normally undertaken by a diverse project team, including medicinal chemists, through a sequence of iterative modifications to the initial hit, forming a "design, make, test, analyse" cycle (Andersson et al., 2009). It should be noted that the synthesis of novel chemical matter is challenging in many cases, and often the envisaged ability to synthesise a chemical structure heavily informs the design process (Roughley and Jordan, 2011).

There are several main paradigms for molecular optimisation: ligand-based, structure-based, and fragment-based. Ligand-based drug discovery (LBDD) methodologies optimise molecular features using data from other chemical structures and fundamental chemical principles, but without explicitly understanding the method of action of such molecules. Structure-based drug discovery (SBDD), on the other hand, utilises 3D structural information of protein-ligand binding to rationally introduce functionality to either improve existing protein-ligand interactions or find

novel ones. The structural information required for this approach can either be experimentally gathered, typically via X-ray crystallography (Zheng et al., 2014), computationally generated (e.g. via homology modelling and docking, Hillisch et al., 2004), or a combination of both.

Generating high-resolution protein-ligand crystal structures is a time-consuming process and therefore initially SBDD was not often used to guide molecular optimisation, but was employed retrospectively to rationalise the observed effects of small molecule alterations. However, developments in high-throughput crystallography have allowed SBDD to be utilised more proactively in drug design (Blundell et al., 2002). These advances have also facilitated the adoption of fragment-based approaches (Hajduk and Greer, 2007), which typically rely on structural data. There are three main strategies to design lead-like molecules from fragment screening using structural data: growing, linking, and merging (Murray and Rees, 2009; Scott et al., 2012), as shown in Figure 1.9.

The first reported use of fragment-based techniques was by Shuker et al. (1996). Since then, there have been many other reported successful applications (Wyatt et al., 2008; Erlanson et al., 2016; Cox et al., 2016). However, while effective, the techniques utilised for advancing fragment hits remain somewhat nascent, with fragment library design one of the few areas to receive substantial attention (Keserű et al., 2016; Cox et al., 2016). Once an initial fragment screen has taken place, deciding which hits to follow-up, and in what way, is a key challenge in FBDD, with this process currently remaining somewhat of an art with limited principled, systematic methods for fragment elaboration.

Finally, these methods are by no means mutually exclusive and the choice of method often depends on the specifics of the project. Indeed ligand-based, structure-based and fragment-based approaches are often all used as part of the same drug discovery program.

Pre-Clinical Testing

Pre-clinical testing seeks to establish whether the compound is safe for human administration and determine an initial dose for human trials. An important step

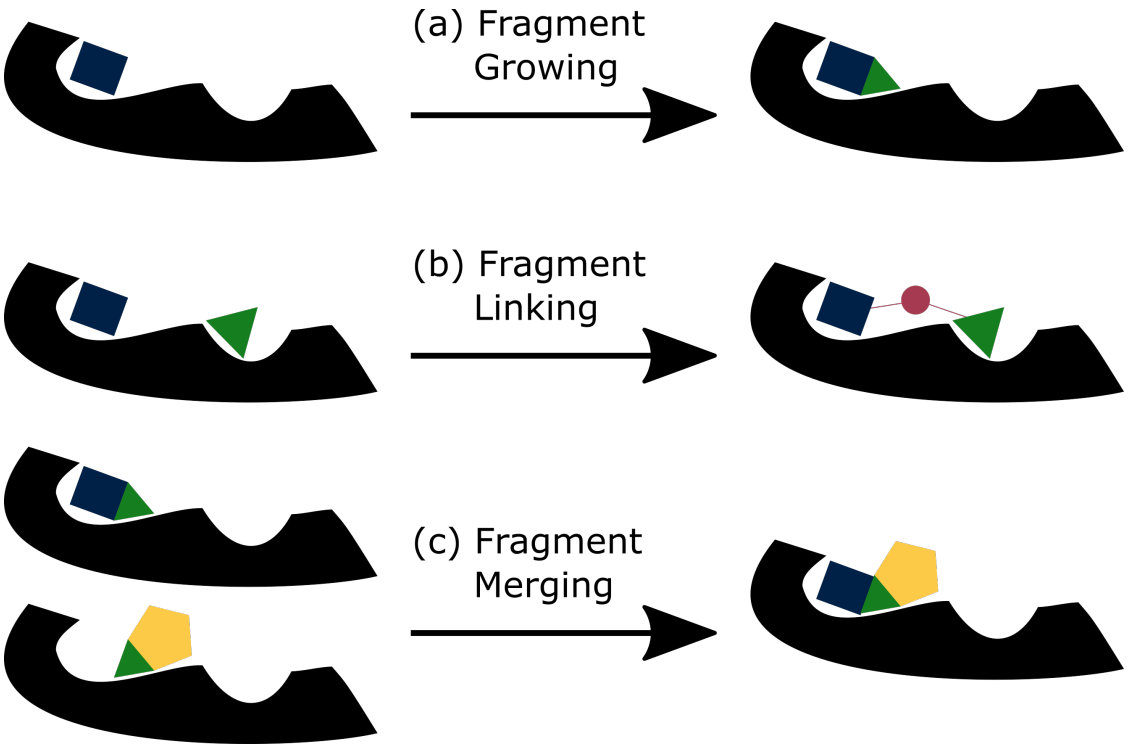


Figure 1.9: FBDD uses small molecules that bind the target at specific subpockets, typically forming high quality interactions. Hits at nearby locations are then often combined into a larger molecule via (a) growing, (b) linking, or (c) merging one, or multiple, fragments.

in this process involves characterising the pharmacokinetics, and specifically ADME (absorption, distribution, metabolism, and excretion) properties, that govern how the molecule passes through the body (Kramer et al., 2007). In addition, there are a number of routine experiments (e.g. hERG and CYP450 inhibition tests, Hughes et al., 2011) to determine whether the compound makes negative interactions with unsuitable biological systems for modulation. Using animal models, such as rats or mice, for initial toxicity testing is the gold standard, but by no means guarantees safety in humans (Kramer et al., 2007). As such, it remains routine to find lead molecules which behave appropriately in standard toxicity profiles but cause failures (DiMasi et al., 2016).

Clinical Trials

Once the lead molecule has passed pre-clinical testing, the drug discovery project moves into clinical trials. Phase I trials are solely to determine baseline safety in

humans and establish dosage for later stages of clinical trials (U.S. Food and Drug Administration, 2018b). To do this, healthy human volunteers are dosed with the proposed treatment and monitored for any side effects. Phase I trials are not used to gauge whether the therapeutic treats the disease in question. Trials typically consist of a small number of subjects, normally ranging from 20-80 (U.S. Food and Drug Administration, 2018b). Note that in some cases, such as chemotherapy, phase I clinical trials can be skipped. The success rate of phase I trials is around 60% (DiMasi et al., 2016) for a new therapeutic, demonstrating that the current process of estimating toxicity still leaves much to be desired.

Phase II trials aim to determine efficacy and are significantly larger than phase I, with typically a few hundred patients treated (U.S. Food and Drug Administration, 2018b). Unlike in phase I, in phase II the patients selected suffer from the disease attempting to be treated. Patients are either given the therapeutic being tested, or one of a placebo or the standard of care treatment. Phase II trials have the highest failure rates of all the stages in the clinical pipeline, with roughly 65% of compounds failing this phase (DiMasi et al., 2016). Almost 50% of failures at phase II are due to lack of efficacy (Harrison, 2016). This is a result of either an incorrect hypotheses about targets, or the molecule not effectively modulating the target *in vivo*.

Phase III trials are typically enlarged versions of phase II trials. While phase II aims to establish biological activity in a clinical setting, phase III trials are to assess the effectiveness of the proposed treatment (U.S. Food and Drug Administration, 2018b). This typically requires up to a few thousand patients to be enrolled in the trials (U.S. Food and Drug Administration, 2018b). The success rate for phase III trials is higher than that for phase II, at around 60% (DiMasi et al., 2016). The relatively low rate even at this stage in the process demonstrates the challenges for a successful drug discovery project and highlights in particular the challenges present in designing clinical trials. Many therapeutic candidates display moderate to low effectiveness in phase II trials, which becomes statistically insignificant in the larger phase III setting (Harrison, 2016).

After phase III has been successfully completed, marketing authorisation can be sought for the drug. This approval allows the drug to be administered to patients outside of clinical trials. Once drugs have been approved, phase IV studies are conducted to continue to monitor the side effects and safety of the drug, the long term risks and benefits, and the efficacy of the drug.

1.3.2 The Cost of Drug Discovery

The process of drug discovery is extremely expensive, both in terms of time and financial resources (Moors et al., 2014). Estimates of the exact cost vary depending on the analysis method, but most estimates fall into the range of $1-3 billion of expenditure per approved therapeutic (Paul et al., 2010; Avorn, 2015; DiMasi et al., 2016). Splitting this into preclinical and clinical expenses, according to one recent estimate, preclinical expenditures total $430 million, and clinical expenditures $960 million on average (DiMasi et al., 2016). In addition to the financial cost, the entire process takes over 10 years from start to finish on average, with clinical evaluation typically taking slightly longer than preclinical development (Paul et al., 2010).

This is unsustainable and solutions need to be found to improve the efficiency of drug discovery (Moors et al., 2014; Ernst & Young, 2017). Clinical trials, or the development phase, constitute more than half of the overall costs, both financial and time, of drug discovery (DiMasi et al., 2016; Paul et al., 2010). However, a substantial portion of the overall expense is still incurred during the discovery phase. In addition, if better candidate molecules are taken into human trials, it could substantially reduce clinical expenses by improving the success rates or reducing the number of patients required for statistically significant trials. Hence finding ways of improving efficiency in the discovery stage would greatly impact the overall process.

1.4 Computer-Aided Drug Design

There is significant interest in the use of computational methods to address the challenges facing drug discovery. Computer-aided drug design (CADD) offers the potential to expedite the design of new drugs primarily by replacing many expensive,

time-consuming *in vitro* and *in vivo* experiments with efficient, cheap (in terms of financial cost) *in silico* ones that can scale better and allow larger regions of chemical space to be explored (Shekhar, 2008; Ou-Yang et al., 2012). A further attraction of many CADD methods, for example physical modelling-based approaches, is the additional information regarding the possible drivers of protein-ligand binding compared to HTS, for example the binding mode (Macalino et al., 2015).

A comparative screen for inhibitors of tyrosine phosphatase-1B demonstrated the promise of CADD over experimentally-driven approaches (Doman et al., 2002). A virtual screen using docking suggested 365 compounds, 127 of which achieved IC50 values of less than 100 μM, a hit rate of nearly 35%. A traditional HTS against the same target yielded just 81 inhibitory compounds out of the 400,000 compounds tested (hit rate 0.021%). While this is a particular successful example, CADD methods are used throughout the drug discovery process and have made significant contributions to the development of many compounds that are either in clinical use or late-stage trials (Stahl et al., 2006).

CADD comprises a broad range of theoretical and computational approaches that are part of drug discovery. In the context of drug design, computational methods seek either to predict properties of compounds or to generate new molecular structure. Often both are combined in order to design compounds with specific properties. In this section, we will discuss virtual screening and computational methods for generating new molecular structures.

1.4.1 Virtual Screening

Experimental screening of compounds for binding is financially expensive and time consuming; in addition, hit rates are often extremely low (Lahana, 1999; Ramesha, 2000). Virtual screening can be used to improve the efficiency and effectiveness of experimental screens (Leelananda and Lindert, 2016). In particular, it can be used to refine the compound library for experimental screening, and inform optimisation of compounds (Klebe, 2006).

Virtual screening is very quick and cheap to perform, allowing orders of magnitude more compounds to be assessed than is possible through experimental screens (Lyu et al., 2019), and has been shown to lead to enrichment of actives in follow-up experimental screens. As such, it has been heavily used in many successful drug discovery projects (Siedlecki et al., 2006; Kiss et al., 2008; Odolczyk et al., 2013; Gau et al., 2017; Bollini et al., 2018).

Virtual screening has several limitations over experimental approaches. Virtual screening requires prerequisite knowledge about the factors responsible for the desired biological response (e.g. binding to a protein target). In addition, the accuracy of virtual screening approaches is not sufficient to draw definitive conclusions about compounds: false negatives can lead to promising compounds being missed, while false positives still represent the majority of highly scored compounds in most cases, even when substantial enrichment has been achieved (Schneider and Böhm, 2002). Given these limitations, it is still the case that experimental verification is needed to confirm a hit.

There are two fundamentally different methodologies for virtual screening: (i) ligand-based virtual screening and (ii) structure-based virtual screening.

Ligand-Based Virtual Screening. Ligand-based virtual screening (LBVS) uses information about only the small molecule compounds, without any explicit information about the protein target. Using a set of ligands that have known binding properties (i.e. active or inactive), features of these ligands can be used to predict whether a new compound will bind. There are numerous methodologies that can be broadly grouped into pharmacophoric models (Leach et al., 2010), chemical similarity search (Willett et al., 1998), and machine learning models (Ramsundar et al., 2015; Wu et al., 2018). LBVS can be applied without a structure for the protein target, and without knowledge of the binding mode. A major limitation of LBVS is that models are built for a specific biological response, and are not generalisable to new targets. To build LBVS models, interaction data for the target in question is required, so LBVS cannot be used in advance of experimental work.

Structure-Based Virtual Screening. In contrast, structure-based virtual screening (SBVS) uses information about the protein target in question, typically a 3D structure, to help determine if a compound will bind to the protein (Leelananda and Lindert, 2016). Unlike LBVS, SBVS can be used in advance of any experimental binding data for the target of interest.

SBVS is a multi-step process, with the following representing a typical pipeline. First a suitably prepared (Madhavi Sastry et al., 2013) 3D structure for the protein target is needed. Preferentially this is determined experimentally, for example using X-ray crystallography, or alternatively a model of the protein can be constructed from its amino acid sequence. Once a 3D structure is available, binding pockets, locations on the target where small molecules can bind, need to be established (Leelananda and Lindert, 2016). These can also either be determined experimentally, or by computational methods (Zheng et al., 2013). Prospective ligands are then docked into the binding pockets of the protein target. Docking consists of two components: first possible binding modes, or poses, of the ligand within the active site of the protein are proposed; next, poses are ranked in order to determine the most likely representation of a possible true binding mode (pose prediction). Ranking of poses has traditionally been achieved by using an estimate of the binding affinity (e.g. AutoDock Vina, Trott and Olson, 2010); however this need not be the case and machine learning methods have also been used to train scoring functions specifically for this task (Ragoza et al., 2017; Hochuli et al., 2018; Sunseri et al., 2019).

SBVS utilises one, or several, potential poses for a protein-ligand complex to assess the likelihood that the compound will bind to the protein based on the poses generated (Klebe, 2006). In this chapter, we will focus on the design and development of scoring functions for this final step, as opposed to the docking protocol more generally.

Traditional Approaches to Virtual Screening

Traditional approaches for scoring protein-ligand complexes have typically used experimental data to parametrise a physically inspired function (Böhm, 1994;

Eldridge et al., 1997; Friesner et al., 2004; Trott and Olson, 2010; Koes et al., 2013; Gohlke et al., 2000; Huang et al., 2010; Zhou and Skolnick, 2011). These functions are then used to guide the search for the binding mode and to assess the strength of interactions between the protein and ligand, often providing an estimate of the binding affinity.

Scoring functions used in docking protocols can generally be categorised as one of four types depending on the information and method used: force-field, knowledge-based, empirical, or hybrid (Wang et al., 2003).

Force-field scoring functions estimate the change in free energy upon binding by summing the terms of a molecular mechanics force field, such as van der Waals and electrostatic interactions (Guedes et al., 2018). Other contributions to the interaction energy are also often included, for example solvent effects on exposed hydrophilic and hydrophobic atoms, along with the intramolecular energies, or strain energies, of the two binding partners. This approach has both the advantage and disadvantage of adopting physical models which do not depend on experimental data (Huang et al., 2006). One such advantage is the interpretability of such models. However, they are inherently limited by our understanding of the physical system being modelled, often do not incorporate all physical processes that govern molecular recognition (Ain et al., 2015), and often rely heavily on the parametrisation of the system (Madhavi Sastry et al., 2013).

A key application of molecular mechanics force fields that remains highly relevant is in biomolecular simulation-based approaches, such as molecular dynamics (Huggins et al., 2019). Physics-based simulation methods have demonstrated accurate performance in the prediction of both absolute (e.g. Aldeghi et al., 2016; Aldeghi et al., 2017) and relative (e.g. Wang et al., 2015a) binding free energy calculations. However, the high computational cost and the sensitivity of the performance to the specific system means that such methods are far from a solution to the challenge of finding potent binders (Schindler et al., 2020). In particular, computational demands prevents these techniques being applicable to virtual screening pipelines due to the limited throughput restricting evaluation to a

handful of molecules. As such, they are typically used in late stage development, for example lead optimisation.

Knowledge-based scoring functions, or statistical potentials, are usually derived from the distribution of pairwise distances between atoms (or another feature) observed in a corpus of protein structures, such as the Protein Data Bank (PDB, Berman et al., 2000; Burley et al., 2019), the principal database for 3D structural data of biological macromolecules and their complexes (Guedes et al., 2018). These are then converted into an energy function that describes the preferred geometries of the pairs of atoms (Miyazawa and Jernigan, 1985; Sippl, 1990).

Empirical scoring functions aim solely to reproduce experimental affinity data rather than capture the underlying physics of the interactions (Pason and Sotriffer, 2016; Du et al., 2016). They consist of a weighted sum of largely uncorrelated physically-meaningful terms, for example the number of hydrogen bonds present or the molecular weight of the ligand. The weighting of these features is either determined using experimental binding data for known protein-ligand complexes or assigned manually.

Scoring functions which combine elements of these approaches are known as hybrid scoring functions. These approaches all make assumptions regarding the functional form of the relationship between the input features and the binding affinity (or other quantity being predicted). While this results in these techniques being interpretable, the use of rigid functional forms inherently limits their ability to capture complex relationships, and is constrained to our current understanding of inter- and intra-molecular interactions.

Machine Learning Scoring Functions

Machine learning techniques have begun to be used over the last decade to develop more accurate scoring functions for pose prediction (Hochuli et al., 2018), virtual screening (Ragoza et al., 2017; Imrie et al., 2018), and affinity prediction (Ballester and Mitchell, 2010; Jiménez et al., 2018). Machine learning approaches differ fundamentally in that they do not assume the functional form of the relationship

between descriptive features and binding affinity, or other target quantity such as active or inactive. Instead, machine learning methods are able to infer relationships between features and the target quantity directly from the data.

Many models, particularly early attempts, reuse the features of traditional approaches (Durrant and McCammon, 2011; Wójcikowski et al., 2017; Li et al., 2014a), but exploit the greater flexibility in model structure to produce better representations of the same input data (Li et al., 2015). For example, Li et al. (2014a) demonstrated the ability of a random forest model to continue to learn from new data and additional features long after the performance of a multiple linear regression model plateaus.

Other machine learning approaches have either designed specific features (Durrant and McCammon, 2011; Ballester and Mitchell, 2010) or used more general features (Wu et al., 2018), such as molecular fingerprints, as inputs. Two notable examples of such approaches are RF-Score (Ballester and Mitchell, 2010) and NNScore (Durrant and McCammon, 2011). We will briefly outline the method of both approaches to help highlight the difference in approach that deep learning, and in particular CNNs, offer.

RF-Score. In 2010, Ballester and Mitchell (2010) published RF-Score, a structure-based method for predicting protein-ligand binding affinity using random forest regression. They represented protein-ligand complexes using the number of protein-ligand atom pairs with a separation of less than 12 Å. Four protein atom types (C, N, O, S) and nine permitted ligand atom types (C, N, O, F, P, S, Cl, Br, I) combine to give 36 distinct atom pairs, whose counts for a given protein-ligand complex serve as the input features to the model.

The authors trained a random forest to predict binding affinity using a set of 1,105 complexes from PDBbind (Cheng et al., 2009). In a benchmark of RF-Score against the state-of-the-art traditional scoring functions, including those found in GOLD (Jones et al., 1997) and Glide (Friesner et al., 2004; Friesner et al., 2006), RF-Score outperformed these functions on a test set of 195 complexes. In a later publication, Wójcikowski et al. (2017) trained an updated version of RF-Score for

virtual screening, performing roughly in line with AutoDock Vina (Trott and Olson, 2010) in a cross-target split of DUD-E (Mysinger et al., 2012).

NNScore. Around the same time, Durrant and McCammon published NNScore (Durrant and McCammon, 2010; Durrant and McCammon, 2011), a structure-based approach which used an ensemble of neural networks, each with a single hidden layer, to predict binding affinity. Similar to RF-Score, the representation of the protein-ligand complex is primarily based on atom pairs, in this case AutoDock atom types (Trott and Olson, 2010), while the authors distinguished between atom pairs within 2 Å and pairs within 4 Å. In addition, they also included an estimate of the total electrostatic energy for each atom pair within 4 Å, basic counts of 13 different ligand atom types, and the number of rotatable bonds in the ligand as addition features, resulting in a total of 194 distinct input features. In virtual screening tests on influenza N1 neuraminidase, the authors reported slightly enriched performance over AutoDock Vina (Durrant and McCammon, 2010), with similar levels of performance reported in a follow-up publication (Durrant and McCammon, 2011).

Deep Learning Approaches

For both traditional approaches and standard machine learning models, the particular features are chosen prior to fitting parameters. This biases the model to the choice of features and leads to an unnecessary loss of information through the elimination or approximation of the raw structural data. Considering the above examples, RF-score loses the counts of any atoms pairs not included as features, or counts further than 12 Å apart, while making the approximate assumption that all atom pairs within 12 Å are equivalent and the number of occurrences is all that matters. The features of NNScore are more descriptive, but make similar approximations.

Deep learning approaches are able to minimise the initial featurisation process by taking as input a format that more closely mimics the underlying protein-ligand complex. The model is then able to learn directly from the data the features that explain a given property, such as whether a compound binds, rather than the features being defined in advance.

There have been several recent deep learning approaches described that have learnt features for virtual screening or binding affinity prediction in an end-to-end manner and have been shown to outperform approaches that take as input precomputed features (Wallach et al., 2015; Ragoza et al., 2017; Gomes et al., 2017; Jiménez et al., 2018). In particular, Ragoza et al. (2017) developed a CNN-based approach for SBVS that minimised initial featurisation of input data, using only a 3D voxel grid of atomic coordinates with basic atom typing.

Datasets for Structure-Based Virtual Screening

There are many possible databases upon which to train and test virtual screening methods, and the most appropriate choice will depend on the goals of the specific project. A common goal is to develop a universal scoring function. Universal in this sense means not directly optimised for a particular group or family of proteins, with the goal of being able to make accurate predictions for arbitrary proteins. The majority of data sets are designed for this purpose and contain a diverse range of proteins. Robust assessment of scoring functions is essential for the continued development of virtual screening, and thus suitable benchmarks are critically important. The following represent the most commonly used benchmarking sets. We discuss their construction, strengths, and limitations.

DUD-E. A widely used virtual screening benchmark is the Database of Useful Decoys: Enhanced (DUD-E, Mysinger et al., 2012) data set. DUD-E consists of 102 targets across eight protein categories, more than 20,000 active molecules, and over one million decoy molecules. Active molecules were extracted from ChEMBL09 (Bento et al., 2011) if their activity/affinity (Ki, Kd, IC50, EC50) was $\leq 1\mu$M. Ideally, inactive molecules should also be selected based on experimental evidence (Lagarde et al., 2015a). However, this is typically not possible. As a result, compounds that are presumed inactives are used as decoys due to the very low chance for an arbitrary molecule to bind to a given target. Due to the scarcity of true inactive molecules, the majority of virtual screening benchmarks include putative inactives, or decoys, instead. Mysinger et al. (2012) took inactives from ChEMBL

where available (with affinity $\geq 30\mu$M), but the majority of decoy compounds were selected from ZINC (Irwin and Shoichet, 2005).

Decoys were selected to have similar physicochemical properties to active compounds while being topological dissimilar. This is done to unbias properties that, while important, should not distinguish binders from non-binders (e.g. molecular weight); the topological filter was applied to avoid including active compounds in the decoy sets. In total, 50 decoys were chosen per active compound. In an effort to reduce analogue bias, active ligands were clustered by their Bemis-Murcko (Bemis and Murcko, 1996) atomic frameworks, with only the most active ligand included, subject to a minimum number of actives per target. Despite these efforts, active compounds in some of the targets remain highly similar, and substantial physicochemical property biases remain (Chaput et al., 2016; Sieg et al., 2019). This issue of property bias within decoys is not unique to DUD-E.

DEKOIS. In 2011, Vogel et al. (2011) proposed a new generator of decoy compounds sets called Demanding Evaluation Kits for Objective *In Silico* Screening (DEKOIS). Their workflow was further refined in 2013, giving rise to DEKOIS 2.0 (Bauer et al., 2013), a virtual screening database with 81 targets across 11 target classes. Active molecules were taken from BindingDB (Liu et al., 2006), with the activity cut-off selected dynamically on a per target basis. Bauer et al. removed compounds possessing a 1000-fold weaker target affinity than the most potent inhibitor for the respective target. Decoys were taken exclusively from ZINC (Irwin and Shoichet, 2005), with 30 decoys selected per active compound.

Similarly to DUD-E, the authors designed their tool to avoid the introduction of well-known and described biases into the decoy sets, i.e., analogue bias and artificial enrichment. Physicochemical properties of both active ligand and decoys are matched to limit the analogue bias, while the authors proposed a quantitative measure to assess the risk of including false negative compounds as decoys, known as the latent actives in the decoy set (LADS) score (Vogel et al., 2011; Bauer et al., 2013), based functional class fingerprints. Property similarity and LADS score are then simultaneously optimised to produce the final decoy set. While

they demonstrated improved property-matching compared to DUD-E, bias still remains among the decoys.

MUV. Whilst the use of putative inactives is widespread and the methods adopted to ensure decoys are not binders are fairly effective, experimentally verified inactives remain the gold standard (Lagarde et al., 2015a). However, using such inactives only overcomes one of the problems for a benchmarking set. The Maximum Unbiased Validation data sets (MUV, Rohrer and Baumann, 2009) were designed with these two factors in mind. MUV is based on PubChem (Souvorov et al., 2007) bioactivity data from 17 targets, each with 30 actives and 15,000 decoys. This presents an order of magnitude increase in the number of decoys per active compound compared to the other data sets discussed. Actives were selected from confirmatory screens and were chosen to be maximally spread based on simple descriptors (such as basic atom counts) and embedded in decoys, which were selected from a primary screen for the same target.

The MUV data sets were designed to avoid analogue bias and artificial enrichment, which produce overly optimistic predictions of virtual screening performance. The result is a very challenging benchmark that minimises the effects of artificial enrichment in evaluating virtual screening methods, in particular ligand-based methods. Some papers of have questioned the appropriateness of using MUV as a structure-based virtual screening benchmark (Ragoza et al., 2017; Tiikkainen et al., 2009), due to the use of cell-based assays for some of the targets and limited attention to structure-based considerations, among other factors. Indeed, the only approaches that have shown meaningful predictive power on MUV are ligand-based methods, and these have required including substantial amounts of target-specific data in the training set, and have often used large external data sets without any regard to the overlap between training and test sets (Ramsundar et al., 2015; Wu et al., 2018).

MUBD. The Maximal Unbiased Benchmarking Data Sets (MUBD) seek to extend the principles of the MUV data set to new targets, in particular where putative, rather than true, inactives must be used. As such, the principles adopted can be used to create a data set suitable for both LBVS and SBVS, simultaneously

addressing the relative lack of LBVS data sets and minimising the bias often found in SBVS benchmark sets. In particular, Xia et al. (2014) built decoy data sets targeting the G protein-coupled receptors (GPCRs). They have since expanded the concept to other families (Xia et al., 2015; Xia et al., 2018). In the case of MUBD-GPCR (Xia et al., 2014), active ligands were taken from GLL (Gatica and Cavasotto, 2012), an existing virtual screening benchmark for GPCRs. Diversity between active ligands was ensured by enforcing a maximum similarity based on MACCS (Molecular ACCess System) fingerprints (McGregor and Pallai, 1997).

Decoys from ZINC were matched with six physicochemical properties such that no compound has physicochemical properties outside of the bounds of the active ligands, while a maximum fingerprint similarity to any active compounds was enforced to prevent the introduction of potential active structures into the decoy set. Compounds were further filtered to ensure the similarity of physicochemical properties was maximised while maintaining a random spatial distribution of the decoys around the active ligands, coupled with targeting an optimal embedding of actives in the decoys by fingerprints. For each active, 39 decoys were chosen.

In contrast to the other data sets described above, each MUBD only includes targets for a specific family or collection of proteins, resulting in a highly targeted scoring function. Such a set can be very helpful when the goal is to learn a model for a novel target within that family, or to assess selectivity of compounds within a protein family.

ChEMBL set. The final SBVS data set we will discuss is one generated from ChEMBL by Riniker and Landrum (Riniker and Landrum, 2013), following Heikamp and Bajorath (Heikamp and Bajorath, 2011). They selected a set of 50 human targets from ChEMBL version 14. They chose actives that had at least 10 μM potency, had a molecular weight under 700 g/mol, and did not have metal ions. The actives were down-sampled using the RDKit (Landrum, 2006) diversity picker to select the 100 most diverse compounds for each target. For each active, two decoys were randomly selected from the ZINC database subject to a minimum similarity using a simple atom-count fingerprint (ECFP0). This yielded a total of

10,000 decoys that were shared across all targets. This data set has been analysed in less detail than DUD-E, but it is likely to suffer from artificial enrichment due to the lack of physicochemical property matching of decoys to actives, and the use of the same decoys across all targets.

Data Splitting Methodologies

Once a suitable data set has been selected, there are numerous possible ways of splitting data to form training and validation sets. Ideally, in any machine learning experiment one should keep a held-out test set of data that most closely represents the real-world challenge for the data set. In the majority of machine learning applications, a random split of data within each class is appropriate. However, for structure-based virtual screening, more careful consideration needs to be given in light of both the specific problem (binding to one target is not the same as binding to another) and the use-cases (often a prospective screen for a novel target).

Intra-target splitting. Intra-target splitting in its basic form is comparable to a random split of data within each class. Both training and test sets contain data from all targets (or a single target), i.e. each target has its ligands both in training and test sets. Such an approach mimics experiments on targets for which there are already known ligands. Assessing models in this way does not allow conclusions to be made about the ability for a model to generalise to novel targets, which is highly relevant in most use-cases, and is likely to lead to overly optimistic assessments of scoring functions due to the known-biases that exist within many benchmarks unless further precautions are taken, such as chemotype splitting or time splitting (described below).

Cross-target splitting. In contrast, in cross-target splitting there is no overlap of targets between training and test data. As a result, all testing is performed on unseen targets, more closely mimicking many real-world scenarios. This is more challenging, as different proteins typically have different binding modes. This normally results in substantial differences between the active compounds between

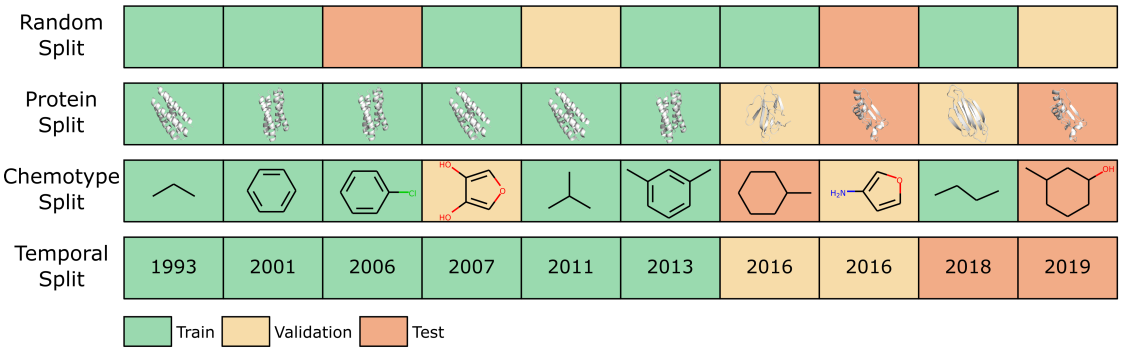


Figure 1.10: Representation of different methodologies for splitting a data set into train, validation, and test sets.

the targets, while substantial commonalities will normally exist for a collection of actives for a single target that bind in the same pocket.

Several publications have demonstrated the additional challenge of cross-target splitting compared to intra-target splits. Wu et al. (2018) achieved almost perfect performance on DUD-E when performing a random training/test split across all targets (intra-target splitting), allowing training and test sets to contain examples from the same targets. Wójcikowski et al. (2017) saw performance in cross-validation almost triple when splitting their data either randomly across all targets (intra-target splitting) or on a single target basis, as opposed to keeping all examples for a given target in the same fold (cross-target splitting).

Many proteins are highly similar and, as a result, have substantial overlap of active ligands. Consequently, cross-target splitting alone is insufficient to ensure that one has a robust method to assess the predictive power of models. There are several different ways of splitting data to ensure dissimilarity between training and validation/test sets (Figure 1.10).

Protein sequence similarity. Proteins with high sequence similarity often have similar structures and function, and hence similar ligands will often bind to these proteins. To remove this source of bias, targets can be clustered during cross validation, ensuring highly similar targets are in the same fold, or can be removed from a potential test set if a maximum similarity threshold is met with any target in the training set. In the experimental setup of Ragoza et al. (2017),

a threshold of 80% sequence similarity from a global sequence alignment was set for both cross-validation and the construction of test sets.

Binding site similarity. Binding sites can be extremely similar, even for targets with relatively low sequence similarity. In most cases, this would result in substantial overlap in activity between two targets that would not be identified by protein sequence similarity. To avoid this, Ragoza et al. (2017) assessed their CNN protocol by performing ProBiS (Konc and Janežič, 2010) structural alignment on the binding sites of all pairs of targets from the training and proposed test sets, removing any targets for which a significant alignment was found using the default ProBiS parameters. To illustrate the level of redundancy between commonly used benchmarks, these two steps combined resulted in reduction in potential independent test sets from 50 initially to a 13 target subset of the Riniker and Landrum ChEMBL set, and from 17 initially to a 9 target subset of the MUV set (Ragoza et al., 2017).

Chemotype splitting. Scaffold, or chemotype, splitting separates the samples based on the two-dimensional structural frameworks of the ligands, such as Bemis-Murcko scaffolds (Bemis and Murcko, 1996). Since scaffold splitting attempts to separate structurally different molecules into different subsets, it offers a greater challenge for learning algorithms than the random split (Wu et al., 2018). Scaffold splits provide a stronger test of a given model's ability to generalise compared with random splitting (Gomes et al., 2017), and is most applicable for intra-target splitting.

Temporal splitting. Finally, if time information is known, training, validation, and test sets can be constructed on the basis of this information (temporal or time splitting, Wu et al., 2018; Sheridan, 2013). This allows retrospective studies to more accurately reflect prospective ones, and is more challenging than a random split (Gomes et al., 2017). However, it is frequently not possible to easily incorporate, since time information is often not available. This idea is most readily applicable to intra-target splitting, but could also be adopted in the case of a cross-target split, where all data from training targets must have been published before any of the ligands from the test targets.

1.4.2 *De Novo* Design

De novo design is a computational technique to generate novel molecules with a desired property profile (Nicolaou and Brown, 2013). It offers a potentially complementary approach to virtual screening, which requires the screening of increasingly large virtual compound libraries, with recent reports assessing $10^8 - 10^9$ molecules (Lyu et al., 2019; Stein et al., 2020; Gorgulla et al., 2020). However, this still represents a small fraction of chemical space, which is estimated to contain between $10^{23} - 10^{60}$ molecules (Polishchuk et al., 2013). Due to the complex, multi-objective nature of drug discovery, library sizes are being driven increasingly larger to find suitable molecules. *De novo* design attempts to explore chemical space via search or optimisation to reduce the number of molecules explicitly generated. Due to the size of the relevant subset of chemical space, *de novo* design algorithms offer the potential to explore this space more completely despite assessing orders of magnitude fewer compounds (Brown et al., 2019).

Computational methods for generating molecules broadly fall into two main categories. Traditionally, algorithms have utilised rules-based transformations (e.g. Böhm, 1992b), often with great success (e.g. Besnard et al., 2012). More recently, deep learning-based methods have been proposed, displaying promising results (e.g. Zhavoronkov et al., 2019). In this section, we will briefly discuss both approaches.

Rules-Based Transformations

Prior to the introduction of deep learning-based methods, generative algorithms employed a variety of handcrafted rules governing molecule design. Early methods attempted to design molecules for a specific binding site via the combination of a library of molecular fragments (e.g. LUDI, Böhm, 1992b; Böhm, 1992a; SPROUT, Gillet et al., 1993; Gillet et al., 1994) or atom-by-atom design of molecules (e.g. Nishibata and Itai, 1991; Rotstein and Murcko, 1993).

Rules-based systems were also employed to transform a starting structure into novel compounds. This was often achieved via a genetic search algorithm (e.g.

Venkatasubramanian et al., 1994; Brown et al., 2004) to evolve compounds using multiple transformations to maximise an objective function.

Generative systems employing rules-based transformations have been applied to many design scenarios in drug discovery, including scaffold hopping (e.g. Maass et al., 2007; Vainio et al., 2013), fragment linking (e.g. Thompson et al., 2008; Dey and Caflisch, 2008)

An example demonstrating the success of such an approach can be seen in Besnard et al. (2012). Using a genetic-based approach coupled with Bayesian probabilistic activity models (Paolini et al., 2006), they evolved donepezil, an approved acetylcholinesterase inhibitor drug, into a dual dopamine D2 inverse agonist and D4 agonist which was able to penetrate the blood brain barrier.

However, there are several limitations to genetic-based approaches. The first is the requirement to manually specify rules or heuristics for compound mutations. Second, it is claimed that discrete optimisation methods struggle to effectively search large areas of chemical space because it is not possible to guide the search with gradients (Gómez-Bombarelli et al., 2018).

Rules-based transformations also form the basis of matched molecular pair analysis (MMPA, Griffen et al., 2011; Dossetter et al., 2013). MMPA mines a database for pairs of molecules differing by a single molecular transformation. The properties of the molecules in such pairs are compared to understand the affect of the particular molecular transformation. This can then be used during compound design or optimisation to improve the properties of a molecule in a data-driven way. However, the same structural transformation may have vastly different effects depending on the specific project (Hajduk and Sauer, 2008; Warner et al., 2012).

Generative Modelling using Deep Learning

Deep generative modeling of molecules is a nascent field which began with the molecular autoencoder of Gómez-Bombarelli et al. (2018). In this work, they demonstrated that a variational autoencoder (VAE, Kingma and Ba, 2015) could be used to encode the SMILES string (Weininger, 1988) representation of molecules

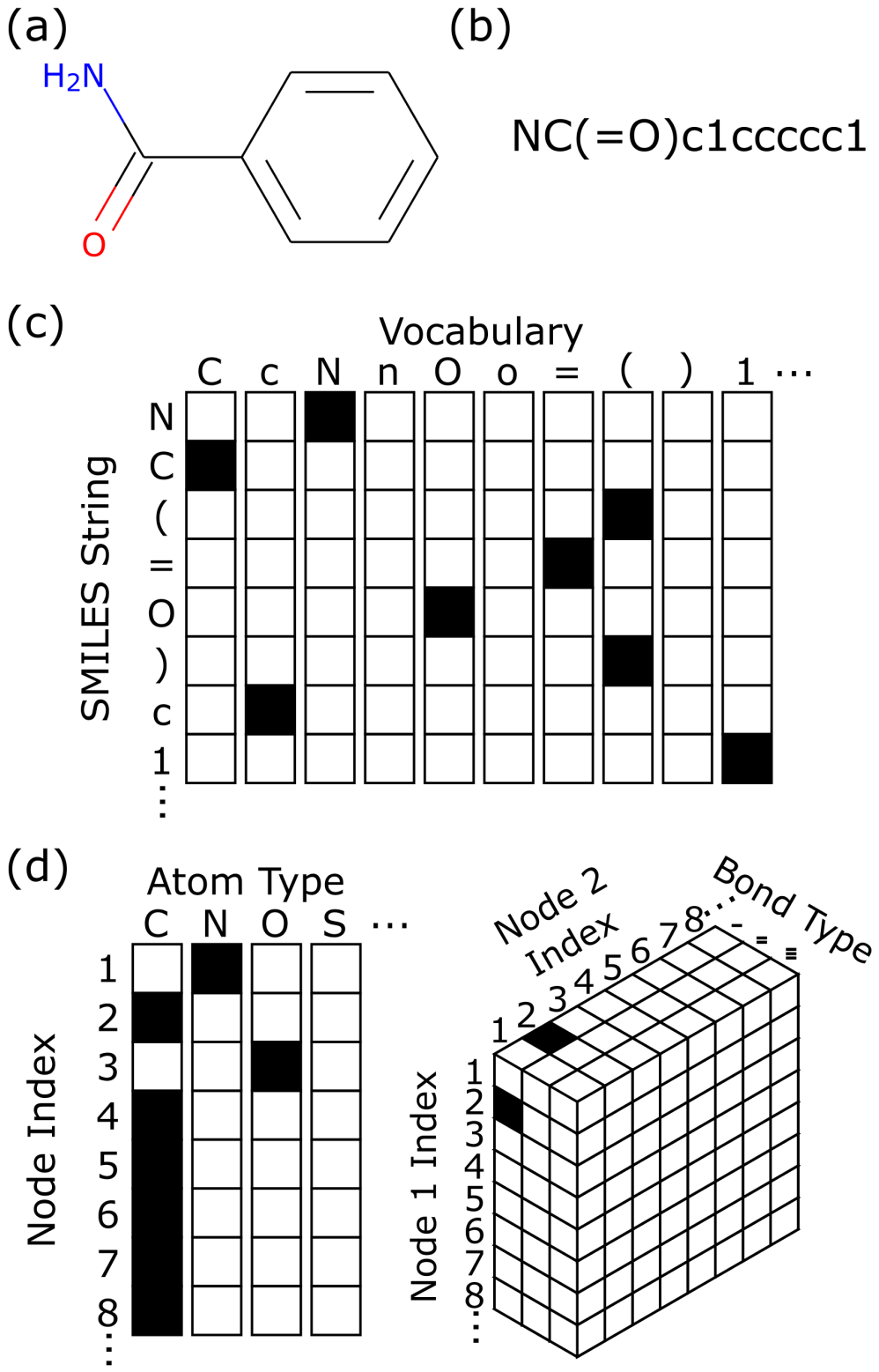


Figure 1.11: (a) Molecular graph of benzamide, (b) Canonical SMILES representation of benzamide, (c) one-hot representation of the SMILES string for benzamide, (d) one-hot tensor representation of benzamide's molecular graph. Note only partial representations of (c) and (d) are shown for illustrative purposes.

into a continuous representation (known as the latent space) that could then be used by a decoder to generate molecules. The benefit of such a method is the ability to generate novel compounds by traversing the lower dimension latent space rather than optimising chemical structures directly.

Since then, there has been a significant number of deep generative models for generative chemical matter using several different deep learning architectures and input formats. In this section, we will describe the key developments. Elton et al. (2019) provides a more exhaustive discussion of recently published deep generative models for molecule design.

Initially, models followed Gómez-Bombarelli et al. (2018) and utilised SMILES strings within a variety of contexts (e.g. Olivecrona et al., 2017; Segler et al., 2018). This allowed language models, such as recurrent neural networks (e.g. Hochreiter and Schmidhuber, 1997) to be used directly on a text-based representation of molecules.

Strings are not a natural way to describe molecules and quickly graph-based generative methods began to be used (see Section 1.2.2), outperforming SMILES-based methods at several tasks (e.g. Jin et al., 2018; Liu et al., 2018). While the use of graphs offer clear benefits over SMILES, the ability to leverage methods development for natural language processing and the significantly lower computational demands of SMILES means that both formats are still widely used.

Computational Design Tasks

There are two main use cases for *de novo* molecular design: distributional learning and goal-based design (Brown et al., 2019). In distributional learning, given a training set of molecules, a model is tasked with generating new molecules that follow the same chemical distribution. In goal-based design, the challenge is to generate molecules that satisfy a predefined goal.

Numerous machine learning models have been proposed for distributional learning (e.g. Gómez-Bombarelli et al., 2018; Jin et al., 2018). Two benchmarks have been proposed to assess such methods (Brown et al., 2019; Polykovskiy et al., 2020). Distributional learning is an important problem, not least to explore the

nature of generative models (e.g. Arús-Pous et al., 2019) and as a precursor for latent space-based optimisation or search techniques (e.g. Gómez-Bombarelli et al., 2018). However, the practical use in drug discovery is limited to specific scenarios (e.g. Segler et al., 2018), with the majority of the drug discovery process falling into the goal-based paradigm.

Many approaches for goal-directed design have utilised a reinforcement learning approach to find molecules that maximise a given scoring function (e.g. Guimaraes et al., 2017; Popova et al., 2018; You et al., 2018a). Often these methods have adopted a two-stage approach, first training their model for distributional learning before applying a reinforcement learning step (Olivecrona et al., 2017; Popova et al., 2018). This has the effect of speeding up the reinforcement learning process and initialising the model in the space of chemically reasonable molecules.

A common pitfall of both reinforcement learning and genetic algorithms is over-exploiting the scoring function, often leading to chemically unrealistic molecules. Indeed, both Gómez-Bombarelli et al. (2018) and Sanchez-Lengeling et al. (2017) added additional terms to the original reward function in an attempt to improve the chemistry of the generated molecules. While this approach can provide benefit, it is likely necessary to optimise the additional terms for every goal.

A further limitation of reinforcement learning approaches is the reliance on a scoring function that is guaranteed to be imperfect for any biological end-point. While this is known to have significant impact on the success of reinforcement learning, Yang et al. (2020a) is one of the few works in molecule generation to consider the error of the predictive model, with most methods assessing the generated molecules using the same scoring function used to guide the learning process (e.g. Popova et al., 2018).

A different approach for many goal-directed tasks is to reframe the problem as that of chemical transformation. This approach has been utilised in cheminformatics in the form of matched molecular pair analysis (Griffen et al., 2011; Dossetter et al., 2013), but was pioneered for deep learning models by Jin et al. (2019b). In this paradigm, the goal is to learn the chemical transformation mapping a molecule

to one, or several, compounds with the desired properties based on a corpus of paired molecules. While this approach is limited by the availability of such a set of molecular pairs, it is a useful formulation for a number of design tasks, as we will explore further in Chapters 2, 3, and 5.

1.5 Challenges for Machine Learning in Drug Discovery

There are many challenges for machine learning in drug discovery, spanning all areas, including data, algorithmic, political, and practical. Here, we will touch on each briefly.

The first is the dependency on expensive (both in time and cost) experimental data for training and validation. This contrasts with the successes of deep learning in games such as Go (Silver et al., 2016) or chess (Silver et al., 2018), where training data can be perfectly generated in simulations. This motivates the development of methods that can learn from small quantities of data (e.g. few shot learning, Altae-Tran et al., 2017) or effectively utilise other available data (e.g. transfer learning, Pan and Yang, 2010; meta-learning, Maudsley, 1979). Further algorithmic challenges arise from the nature of biological and chemical data, both in terms of the format of such data (e.g. graphs, Section 1.2.2) as well as the inherent noise.

A key challenge is how we quantify success. Prevailing human-led processes are far from infallible (see Section 1.3.2), but it is not currently possible to quantify medicinal chemistry success (Green et al., 2018). In light of this, what is the bar for algorithmic success? Several have cautioned not to set the barrier for computational approaches too high (Green et al., 2018).

Finally, realising the full impact of machine learning methods will require considerable resources to be invested. Experimental validation on real-world drug discovery projects is a critical next step to assess the contribution of machine learning in medicinal chemistry and identify areas requiring improvement.

1.6 Thesis Outline

In this chapter, we have highlighted the challenges facing drug discovery and motivated the application of machine learning as a partial solution. We have discussed several key machine learning methods with applications to drug discovery. In this thesis, we describe several novel methodologies leveraging deep learning for applications to pre-clinical drug discovery.

In Chapter 2, we propose a generative approach to the design of molecular linkers which incorporates basic 3D information. In large scales tests, we find that our method substantially outperforms a database-based approach, the previous *de facto* approach to this problem. Through a series of case studies, we demonstrate the application of our approach to scaffold hopping, fragment linking and PROTAC design.

In Chapter 3, we extend our framework for linker design to incorporate physically-meaningful 3D structural information, providing a richer prior for the generative process. In addition, we demonstrate that our method can be applied to molecular elaboration tasks, such as R-group design, by changing only the training set with no other modifications to the methodology necessary.

In Chapter 4, we turn our attention to predictive modelling, and structure-based virtual screening. We find that the advances in CNN methods for general computer vision tasks are applicable to SBVS. In addition, we propose two techniques to incorporate domain-specific knowledge into this framework. First, we show that limitations in docking necessitate the use of multi-pose scoring and demonstrate the benefits of an average scoring policy. Second, we propose a transfer learning approach to construct protein family-specific models, utilising knowledge of the differences between protein families.

In Chapter 5, we investigate how a generative approach can be used to improve the training and benchmarks sets employed in SBVS. We propose a deep learning method that generates decoys to a user's preferred specification in order to control decoy bias or construct sets with a defined bias. We show that our approach significantly reduces the bias contained in such sets. We validate that our generated molecules are more challenging for docking approaches to separate from actives

than previous decoys. In addition, we show that CNN-based SBVS methods can be trained on such compounds.

Finally, in Chapter 6, we summarise the results of this work. We discuss the main conclusions and describe future work that might follow from this thesis.

What I cannot create, I do not understand.

— Richard P. Feynman

2

Deep Generative Models for 3D Linker Design

Contents

This chapter is based on work described in the following publication:

Fergus Imrie, Anthony R. Bradley, Mihaela van der Schaar, and Charlotte M. Deane (2020). Deep Generative Models for 3D Linker Design. *Journal of Chemical Information and Modeling*, 60(4): 1983–1995

An open-source implementation of the method described in this chapter is

available at https://github.com/oxpig/DeLinker.

2.1 Preface

Rational compound design remains a challenging problem for both computational methods and medicinal chemists alike. As outlined in Chapter 1, computational methods to generate molecules employing machine learning have begun to show promising results (e.g. Zhavoronkov et al., 2019) and have the potential to transform how new molecules are discovered.

However, their use is far from commonplace in drug discovery projects (Schneider and Clark, 2019; Grebner et al., 2020). Early methods were not directly applicable to most design tasks in drug discovery, with a key barrier to broad adoption the lack of flexibility and control (Ståhl et al., 2019). In particular, most generative models built molecules from scratch (e.g. Segler et al., 2018), while methods that required a starting compound to be specified did not allow the user to select which components of the molecule could be altered (e.g. Jin et al., 2018). Further, methods did not allow other constraints to be placed on the generative process, such as 3D information from either the protein structure or other ligands. This meant that control of the design process was largely limited to the selection of the training set for supervised methods or specification of the reward function in the case of reinforcement learning (Li et al., 2020b). This made it challenging, or simply impossible, to integrate domain knowledge or prior beliefs in existing frameworks.

These limitations motivated the method described in this chapter. We sought to design a method that was naturally applicable to a wide variety of scenarios in drug discovery. Crucially, this meant our method should not design entire molecules from scratch, but include one or more starting substructures in all generated molecules, as is typically desired in the hit-to-lead and lead optimisation stages of drug discovery. In addition, most existing frameworks completely ignored 3D structural information, which is crucial to successful compound design. We wanted to develop a method that included such information to allow molecules to be designed for a particular binding site.

To achieve these aims, we developed a graph-based deep generative model combining state-of-the-art machine learning techniques with structural knowledge. Our method ("DeLinker") takes two molecular fragments or partial structures and designs a molecule incorporating both. The generation process is protein context-dependent, utilising the relative distance and orientation between the partial structures. Despite this minimal parametrisation of the structural information, the 3D information is vital to successful compound design and we have demonstrated its impact on the generation process and the limitations of omitting such information.

In a large-scale evaluation, DeLinker designed 60% more molecules with high 3D similarity to the original molecule than a database baseline. When considering the more relevant problem of longer linkers with at least five atoms, the outperformance increased to 200%. We have demonstrated the effectiveness and applicability of this approach on a diverse range of design problems: fragment linking, scaffold hopping, and proteolysis targeting chimera (PROTAC) design. As far as we are aware, this is the first molecular generative model to incorporate 3D structural information directly in the design process.

2.2 Introduction

Drug design is an iterative process that requires potential compounds to be optimised for specific properties, ranging from binding affinity to pharmacokinetics. This process is challenging, in part due to the size of the search space (Polishchuk et al., 2013) and discontinuous nature of the optimisation landscape (Stumpfe and Bajorath, 2012). Typically molecule design is undertaken by human experts, and therefore is a subjective process (Lajiness et al., 2004).

Machine learning models for molecule generation have been proposed as an alternative to human-led design and rules-based transformations (such as Besnard et al., 2012). As discussed in Chapter 1, generative models have typically adopted either the SMILES string representation of molecules (e.g. Gómez-Bombarelli et al., 2018; Popova et al., 2018) or, more recently, graph representations (e.g. Li et al., 2018; Jin et al., 2018). Existing generative models have primarily been used in two

ways. First, methods have been developed to generate molecules that follow the same distribution as the training set, whether a general set of molecules (Gómez-Bombarelli et al., 2018) such as ZINC (Sterling and Irwin, 2015) or ChEMBL (Gaulton et al., 2016), or a more focussed one such as inhibitors for a particular protein target (Segler et al., 2018; Gupta et al., 2018). Second, generative models have been proposed to perform molecular optimisation, taking an input molecule and attempting to modify one, or several, chemical properties, typically subject to a similarity constraint (Jin et al., 2019b; Jin et al., 2019a; Zhou et al., 2019).

While substantial progress has been made for these two problems, current methods have inherent limitations, in particular for structure-based design. Only one approach to date has attempted to include any three dimensional (3D) information in the generative process (Skalic et al., 2019a), despite its importance for designing potent and selective compounds. In this work, Skalic et al. proposed a SMILES-based model for generating molecules from 3D representations. A shape variational autoencoder using convolutional neural networks (CNNs) was coupled with a shape captioning network consisting of a separate CNN used to condition a recurrent neural network (RNN). In this formulation, 3D information was only provided implicitly to seed the RNN, and the method did not allow further control over generated compounds. As a result, their generative model frequently changed the entire molecule, and recovered fewer than 2% of the seed molecules. This is undesirable in many practical settings, such as the design problems described below.

Fragment-based drug discovery (FBDD) has become an increasingly important tool for finding hit compounds, in particular for challenging targets and novel protein families. FBDD utilises smaller than drug-like compounds (typically <300 Da) to identify low potency, high-quality leads, which are then matured into more potent, drug-like compounds. One common way of maturing fragment hits is through a linking strategy, joining fragments together that bind to distinct sites via a linker. It is crucial for successful fragment linking that a linker does not disturb the original binding poses of each fragment (Ichihara et al., 2011; Bienstock,

2015). Thus compound suggestions have strong 3D constraints, determined by the binding mode of the fragments.

Scaffold hopping, though a distinct problem, shares some characteristics with fragment linking. The aim of scaffold hopping is to discover structurally novel compounds starting from a known active compound by modifying the central core structure of the molecule (Böhm et al., 2004). Such a change can result in much improved molecular properties, such as solubility, toxicity, synthetic accessibility, affinity, and selectivity (Böhm et al., 2004; Langdon et al., 2010).

Numerous computational methods have been proposed for fragment linking or scaffold hopping (e.g. Böhm, 1992b; Maass et al., 2007). However, almost all methods published to date rely exclusively on a database of candidate fragments from which to select a linker, with the differences between approaches arising solely from how the database is searched, how the linked compounds are scored, or the contents of the database itself. As a result, these methods are inherently constrained to a set of predetermined rules or examples, limiting exploration of chemical space. In addition, they can only incorporate additional structural knowledge (e.g. the fragment's binding mode) via filtering or search mechanisms.

Current machine learning-based molecule generation methods have not been designed to effectively handle the structure-based design tasks of fragment linking and scaffold hopping. These scenarios require proposed molecules to contain specific substructures, with the goal to design a molecule that maintains the binding mode of the original compound or fragments.

Numerous challenges using SMILES-based methods have been previously noted, largely arising from using language to represent inherently graph-based objects (Elton et al., 2019; Jin et al., 2018). In particular, the grammar makes working with existing structures especially difficult, and has even led to alternative language-based representations being proposed (O'Boyle and Dalke, 2018; Krenn et al., 2019). We therefore do not believe that design tasks requiring proposed molecules to contain specific substructures naturally suit a SMILES-based representation, although this remains an active area of research (Arús-Pous et al., 2020).

Graph-based methods have become more widely used, in particular for molecular optimisation (Jin et al., 2019b; Jin et al., 2019a; Zhou et al., 2019). Due to the nature of these tasks, the shapes of the suggested molecules often differ greatly from the starting points and many of the proposed changes are R-group modifications. In addition, most methods do not afford a high level of control over the generated molecule. Li et al. (2020b) recently proposed a scaffold-based molecular generator that designs molecules retaining particular scaffolds as their core structures. This allows greater control over generated molecules than most existing methods. However, none of the graph-based methods to date incorporate 3D structural information, with only one SMILES-based approach including any 3D information in the generative process (Skalic et al., 2019a).

Finally, Ståhl et al. (2019) introduced a fragment-based reinforcement learning approach for multiparameter optimisation. At each step, their model selects a fragment contained in the molecule and replaces it with a similar one. This typically leads to minor changes in the compounds, and could be utilised for scaffold hopping. However, their model is not able to perform fragment linking and is not designed for tasks requiring 3D shape optimisation.

In this chapter, we introduce the first graph-based deep generative method that incorporates 3D structural information directly into the design process. Our method takes as input two molecular fragments and designs a molecule incorporating both substructures, either generating or replacing the linker between them. This allows our method to handle design tasks such as fragment linking and scaffold hopping effectively. The generation process integrates 3D structural information, specifically the distance between the fragments and their relative orientations. This 3D information is vital to successful compound design, and we demonstrate the limitations of omitting such information, both quantifying its impact in large-scale assessments and empirically showing how our model uses the structural information.

We first demonstrate the effectiveness of our proposed deep generative approach over a database method through large-scale computational assessments. We show that our method, DeLinker, designs 60% more compounds with high 3D similarity

to the original molecule compared to a database-based approach on an independent test set. DeLinker outperforms the database approach by 200% when the evaluation is restricted to linkers with at least five atoms. We then apply our method to several case studies encompassing fragment linking, scaffold hopping, and PROTAC design. DeLinker frequently recovers the experimental end point, even in cases where the linker was not present in the training set, and produces many novel designs with high 3D similarity to the original molecules.

2.3 Methods

The method takes two fragments and their relative position and orientation and generates or replaces the linker between them. This is achieved by building new molecules in an iterative manner "bond-by-bond" from a pool of atoms that can be initialised with partial structures (Figure 2.1). In this framework, the user is able to control the generation process by specifying both the substructures that should be linked and the maximum length of the linker between them. The starting substructures are always retained in the generated molecule, with changes only from the specified exit vectors. In addition, 3D structural information in the form of the distance and angle between the starting substructures is provided to the model to inform the design process. Molecules are encoded by a set of 14 permitted atom types, and our model enforces simple atomic valency rules via a masking procedure to ensure chemical validity. This is the only chemical knowledge incorporated directly into our model; all other decisions required to generate molecules are learnt through a supervised training procedure.

2.3.1 Generative Process

The generative process is illustrated in Figure 2.1 and is similar to Liu et al. (2018) in that our method builds molecules "bond-by-bond" in a breadth-first manner. Generation is initialised with two fragments or substructures that are to be linked together with structural information providing the distance and angle between the substructures. The fragments are converted to a graph representation, where atoms

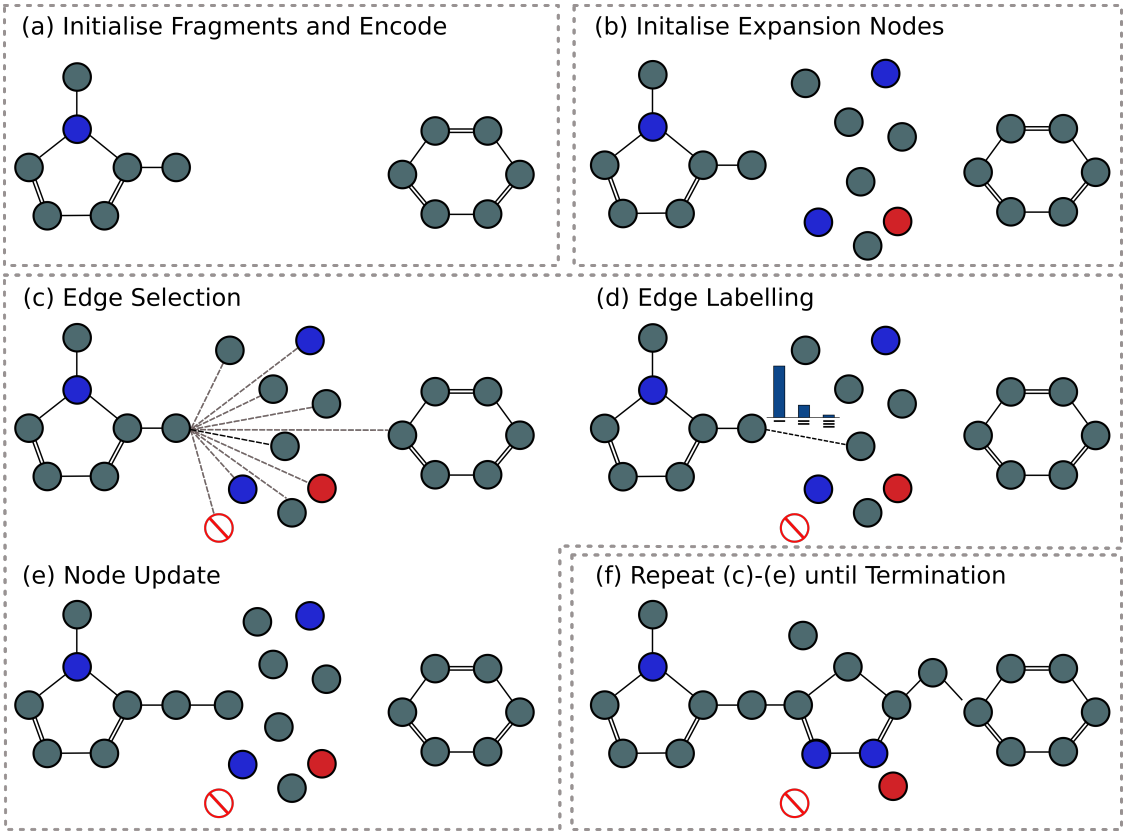


Figure 2.1: Overview of the generation process. The initial fragments (a) are iteratively expanded "bond-by-bond" (c)-(e) to produce a molecule including both fragments (f). Atoms are represented by nodes in a graph, with the colour of the nodes representing different atoms types, while bonds are represented by edges, with different edge types for single, double and triple bonds.

and bonds are represented by nodes and edges, respectively. Each node is associated with a hidden state, $\boldsymbol{z}_v$, and label, $\boldsymbol{l}_v$, representing the atom type of the node. A list of the 14 permitted atom types can be found in Appendix A. The graph is passed through an encoder network, a standard gated graph neural network (GGNN, Li et al., 2016, see Chapter 1 for more information), and the hidden states of the nodes are updated to incorporate their local environment (Figure 2.1a).

Next, a set of expansion nodes are initialised at random, with hidden states $\boldsymbol{z}_v$ drawn from the h-dimensional standard normal distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, where h is the length of the hidden state (Figure 2.1b). The nodes are then labelled with an atom type according to their hidden state, $\boldsymbol{z}_v$, and the structural information by sampling from the softmax output of a learned mapping f. Here, f is implemented

as a linear classifier but could be any function mapping a node's hidden state to an atom type. The number of expansion nodes determines the maximum length of the linker, and is a parameter chosen by the user.

The new molecule is constructed from this set of nodes via an iterative process consisting of edge selection, edge labelling, and node update (Figure 2.1c-e). At each step, we consider whether to add an edge between one of the nodes, v, and another node in the graph. v is chosen according to a deterministic first-in-first-out queue that is initialised with the exit vectors of each fragment. When a node is connected to the graph for the first time, it is added to the queue. New edges are added to node v until an edge to the stop node is selected. The node then becomes "closed" with no additional edges with that node permitted.

All possible edges between the node v and other nodes in the graph are considered (Figure 2.1c), subject to basic valency constraints. A single-layer neural network assesses the candidate edges using a feature vector. The feature vector for the edge between node v and candidate node u is given by

$$\phi_{v,u}^t = [t, \boldsymbol{s}_v^t, \boldsymbol{s}_u^t, d_{v,u}, \boldsymbol{H}^0, \boldsymbol{H}^t, \boldsymbol{D}]\,,$$

where $\boldsymbol{s}_v^t = [\boldsymbol{z}_v^t, \boldsymbol{l}_v]$ is the concatenation of the hidden state of node v after t steps and its atomic label, $d_{v,u}$ is the graph distance between v and u, $\boldsymbol{H}^0$ is the average initial representation of all nodes, $\boldsymbol{H}^t$ is the average representation of nodes at generation step t, and $\boldsymbol{D}$ represents the 3D structural information.

As such, when choosing which edge to add to the graph, the model utilises (1) local information about the nodes, (2) global information regarding the unlinked fragments and the current graph state, and (3) 3D structural information.

Once a node u has been selected, the edge between v and u is labelled as either a single, double, or triple bond (subject to valency constraints) by another single layer neural network taking as input the same feature vector $\phi_{v,u}^t$ (Figure 2.1d).

Finally, the hidden states of all nodes are updated according to a GGNN (Figure 2.1e). At each step, we discard the current hidden states $\boldsymbol{s}_{\mathcal{G}}^t := \{\boldsymbol{s}_v^t : v \in \mathcal{G}\}$ and compute new representations $\boldsymbol{s}_{\mathcal{G}}^{t+1}$ taking their (possibly changed) neighborhood

into account. Note that $\boldsymbol{s}_{\mathcal{G}}^{t+1}$ is computed from $\boldsymbol{s}_{\mathcal{G}}^{0}$ rather than $\boldsymbol{s}_{\mathcal{G}}^{t}$. This means that the state of each node is independent of the generation history of the graph and depends only on the current state of the graph.

Steps c-e in Figure 2.1 are repeated for each node in the queue, until the queue is empty, at which point the generation process terminates. At termination (Figure 2.1f), all unconnected nodes are removed and the largest connected component is returned as the generated molecule. We note that the stereochemistry of generated molecules is not assigned during the generative process.

Multimodal Encoder-Decoder Setup. Our goal is to learn a multimodal mapping from unconnected fragments to connected molecules. During training, we utilised a data set of paired fragments and molecules and trained our model in a supervised manner to reconstruct known linkers. While in this data set there may be a unique molecule associated with two fragments, in practice there are many ways to link two fragments. As such, given a new pair of starting points, a model should be able to generate a diverse set of output compounds.

To this end, we took inspiration from Jin et al. (2019b) and augmented the basic encoder-decoder model with a low-dimensional latent vector $\boldsymbol{z}$ to explicitly encode the multimodal aspect of the distribution of suitable linkers. The generative mapping is converted from $F : X \mapsto Y$ to $F : (X, \boldsymbol{z}) \mapsto Y$, where X represents the starting substructures and Y the connected molecule, with latent code $\boldsymbol{z}$ drawn from a prior distribution, chosen to be the standard normal distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

There are two challenges in learning this mapping. First, as shown in the image domain (Zhu et al., 2017), the latent codes are often ignored by the model unless they are forced to encode meaningful variations. Second, the latent codes should be suitably regularised so that the model does not produce invalid outputs. That is, the generated molecule $F(X, \boldsymbol{z})$ should belong to the domain of the target molecule Y (i.e. connected and able to satisfy the structural constraints provided) given a latent code drawn from the prior distribution. We overcame both of these challenges through our training procedure, where we derived $\boldsymbol{z}$ during training from

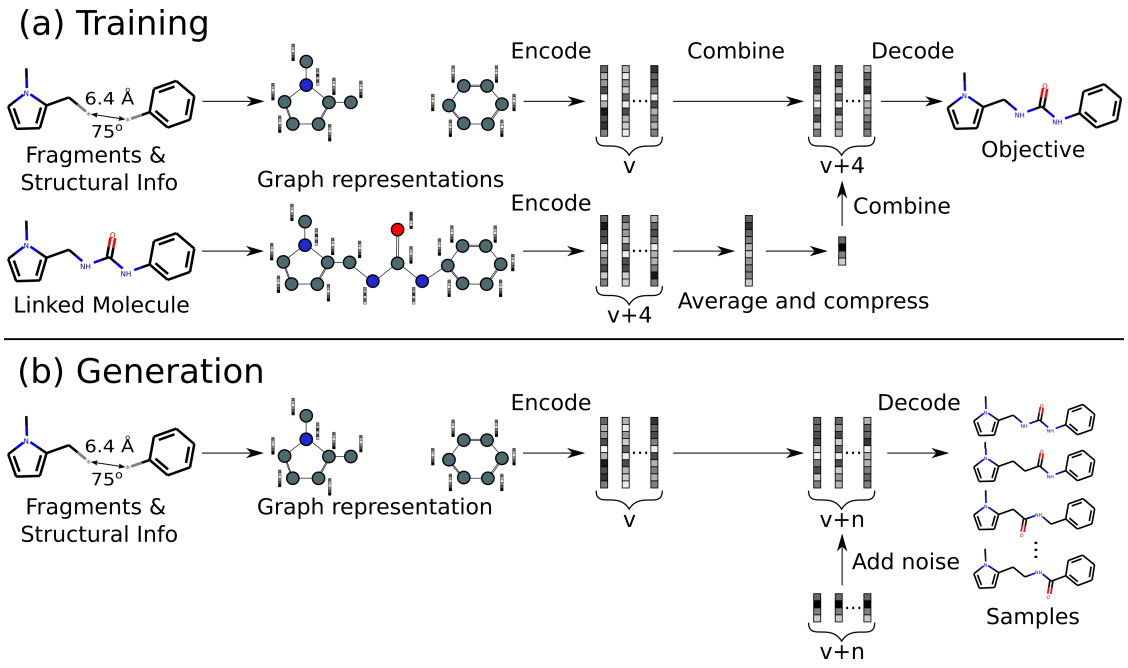


Figure 2.2: Illustration of training and generation procedures. (a) Pairs of fragments and linked molecules are provided as input. The model is trained to reproduce the linked molecule from a combination of the encodings of the fragments and linked molecule. (b) At generation time, the model is given only the unlinked fragments and structural information, and is able to sample a diverse range of linked molecules by combining the encoding of the fragments with random noise.

the embedding of the linked molecule, but regularised the latent vector to follow a standard normal distribution so that we can sample $\boldsymbol{z}$ during generation.

Training. We trained our generative model under a variational autoencoder (VAE) framework on a collection of fragment-molecule pairs (Figure 2.2). For a given pair of fragments X and linked molecule Y, the model is trained to reconstruct Y from $(X, \boldsymbol{z})$, while enforcing the standard regularisation constraint on both $\boldsymbol{z}$ and the encoding of X, $\boldsymbol{z}_X := \{\boldsymbol{z}_v : v \in X\}$.

To encode meaningful variations, the latent code $\boldsymbol{z}$ is derived via a learnt mapping from the average of the node embeddings of the ground truth molecule Y, the linked molecule. Crucially, $\boldsymbol{z}$ is constrained to be a low dimensional vector to prevent the model from ignoring input X and degenerating to an autoencoder for Y. The decoder is trained to reconstruct Y when taking as input a combination of the low dimensional vector $\boldsymbol{z}$ and the node embeddings $\boldsymbol{z}_X$ of the unlinked fragments X (Figure 2.2).

The training objective is similar to the standard VAE loss, including a reconstruction loss and a Kullback-Leibler (KL) regularisation term:

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_{KL}\mathcal{L}_{KL}\,.$$

The reconstruction loss, $\mathcal{L}_{recon}$, is composed of two cross-entropy loss terms, resulting from the error in predicting the atom types and in reconstructing the sequence of steps required to produce the target molecule.

The KL regularisation loss, $\mathcal{L}_{KL}$, contains two terms, one for the encoding of the unlinked fragments X, and the other for the low dimensional vector $\boldsymbol{z}$ derived from the linked molecule Y. These terms are the standard VAE terms capturing the KL divergence between the encoder distributions and the standard Gaussian prior.

We performed limited hyperparameter tuning, measuring performance via the validation loss and not generative performance directly. We found that overall the model was fairly robust to the choice of hyperparameters. Full details of the model architecture and hyperparameters can be found in Appendix A.

2.3.2 Data Sets

There have only been a limited number of examples of successful fragment linking or scaffold hopping reported. As such, for training and large-scale evaluation, we constructed sets of fragment-molecule pairs using standard transformations from matched-molecular pair analysis (Hussain and Rea, 2010).

ZINC. To construct our training set, we used the subset of ZINC (Sterling and Irwin, 2015) selected at random by Gómez-Bombarelli et al. (2018) that contains 250 000 molecules. We constructed possible fragmentations of each molecule by enumerating all double cuts of acyclic single bonds that were not within functional groups, the same procedure adopted by Hussain and Rea (2010). Fragmentations satisfying basic criteria regarding the number of atoms in the linker and fragments were retained, removing trivial and unrealistic scenarios (see Appendix A for further details).

The remaining fragment-molecule pairs were filtered for several 2D properties, namely synthetic accessibility (Ertl and Schuffenhauer, 2009), ring aromaticity, and pan-assay interference compounds (PAINS, Baell and Holloway, 2010), to remove unwanted examples. Full details of the property filters can be found in Appendix A.

By filtering the training set for specific 2D properties, we are also able to assess whether the model is able to learn to generate linkers with certain properties implicitly from the data alone. Since these properties are not input explicitly into the model, these could easily be tailored to a specific project or other requirements.

To provide structural information, we generated 3D conformers for the ZINC set using RDKit (Landrum, 2006), adopting the filtering and sampling procedure proposed by Ebejer et al. (2012). We took the lowest energy conformation as the reference 3D structure for each molecule.

These preprocessing and filtering steps resulted in a data set of 418 797 example fragment elaborations, with linkers of between three and twelve atoms. We selected 800 fragment-molecule pairs at random for model validation (400) and testing (400), and used the remainder to train our model, ensuring no overlap between the molecules in the training and held-out sets.

CASF. A major limitation of the ZINC data set is the use of generated comformers, as opposed to experimentally verified active ones. To address this, we used the CASF-2016 data set (Su et al., 2019), also known as the PDBbind core set, which consists of 285 protein-ligand complexes from PDBbind with high-quality crystal structures from a diverse set of proteins, to construct an independent test set. We followed the same preprocessing procedure as for the ZINC data set (except for conformer generation), resulting in a set of 309 examples.

We performed large-scale evaluations of our method on both the held-out ZINC test data and the CASF data set. For each example, we generated 250 molecules from each pair of unlinked fragments, assuming the linker length was equal to the linker length of the original molecule.

2.3.3 Assessment Metrics

We assessed the generated molecules with a range of 2D and 3D metrics. As is standard in the assessment of models for molecule generation (Brown et al., 2019), we first checked the generated molecules for validity, uniqueness, and novelty. We then determined if the generated linkers were consistent with the 2D property filters used to produce the training set. In addition, we recorded in how many cases the original molecule used to produce the fragments was recovered by the generation process.

Molecules that passed the 2D property filters were assessed on the basis of their 3D shape. Conformers of the generated molecules and the original molecule were compared using two distinct methods: (i) a shape and colour similarity score (SC_{RDKit}), and (ii) root-mean-square deviation (RMSD).

The shape and colour similarity score (SC_{RDKit}) uses two RDKit functions, based on the methods described in Putta et al. (2005) and Landrum et al. (2006). The colour similarity function scores two 3D conformers against each other based on the overlap of their pharmacophoric features, while the shape similarity measure is a simple volumetric comparison between the two conformers. Each produces a score between 0 (no match) and 1 (perfect match), which are averaged to produce a final score between 0 and 1. Scores above 0.7 indicate a good match, while scores above 0.9 suggest an almost perfect match. An illustration of several conformers and their similarity scores can be seen in Figure 2.3.

SC_{RDKit} can either be calculated by comparing only the atoms of the starting fragments (SC_{RDKit} Fragments), or by comparing the entire generated molecule to the original molecule (SC_{RDKit} Molecule). The first measure assesses how closely the conformations of the fragments match, whereas the second also incorporates whether or not the generated linker matches the original (Figure 2.3). Our method is trained to output a diverse range of linkers and not to map exactly to a previously observed linker. However, in the case of scaffold hopping, this metric is important as typically the new linker should match the shape and pharmacophoric features of the original core (Langdon et al., 2010)[1].

[1]In some cases it might be desirable to deviate from this, for example, to design linkers that

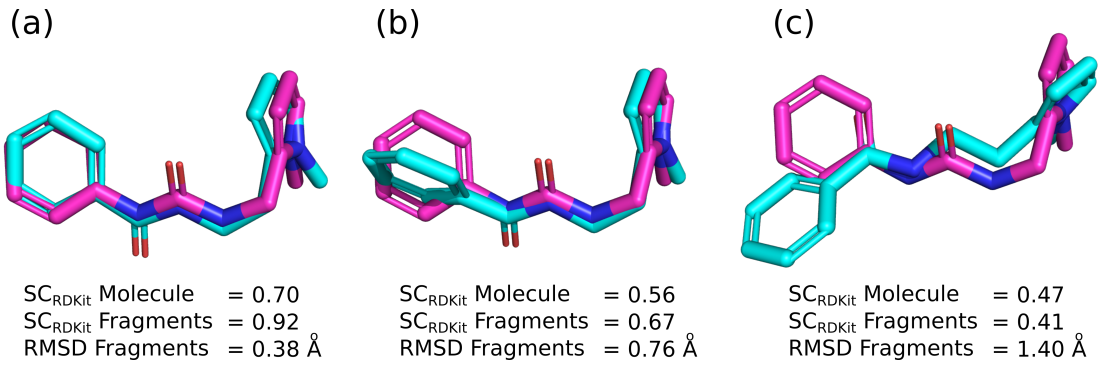


Figure 2.3: Examples of the 3D metrics used to assess the similarity of conformers. The reference conformer is shown in magenta, while conformers of the generated molecule are shown in cyan. (a) represents very strong alignment by both fragment-based metrics, but lower similarity by SC_{RDKit} Molecule due to the different linker. (b) shows modest similarity by all three metrics, while (c) represents poor similarity by all three measures.

RMSD between the coordinates of atoms in the starting fragments in the original and generated molecule can be calculated to give a different measure of 3D similarity (RMSD Fragments). A perfect match has an RMSD of 0Å, with a higher figure indicating greater deviation. An RMSD of below 0.5Å suggests an almost perfect match, while an RMSD above 1.0Å corresponds to a poor match given the alignment procedure and number of heavy atoms. An illustration of several conformations and their RMSDs can be seen in Figure 2.3. Due to the need to match specific atoms, RMSD can only be (reliably) calculated between the atoms of the fragments that are linked, and not the entire molecule.

For each proposed molecule, we generated 3D conformers using RDKit (Landrum, 2006), adopting the filtering and sampling procedure proposed by Ebejer et al. (2012), and scored all conformers. The score for each similarity measure was the best score among all generated conformers for a particular molecule.

2.3.4 Comparison to Other Methods

Several traditional methods exist for linking fragments or replacing the core of a molecule (e.g. Böhm, 1992b; Maass et al., 2007). Almost all methods rely on a database from which to select linkers. As a baseline with which to compare our method, we created a set of all linkers from the training data and sampled from

make new interactions or to enhance selectivity.

this set, joining the linker in one of the two possible orientations at random. This setup ensures that both methods are constructed using the same data, and allows us to assess whether the generated molecules have better shape complementarity than using linkers from the database, while still obeying 2D chemical constraints.

Liu et al. (2018) proposed constrained graph variational autoencoders to generate molecular graphs. They sought to generate arbitrary molecules that conformed to the distribution observed in the training data. Similarly to our method, they assumed a sequential ordering of graph extension steps, constructing molecules in a "bond-by-bond" manner. To compare their work to our method, it was necessary to modify their implementation to perform the molecular design tasks described.

Our work differs from the model of Liu et al. (2018) in three key ways. First, our model takes as input structural information in the form of the distance and angle between the starting substructures and uses this to augment the feature vector used in the generation process. Second, our model is designed to be trained for molecular translation, in particular taking as input unlinked substructures and generating linked molecules. While others have developed methods for molecular translation (Jin et al., 2019b; Zhou et al., 2019), none have taken two unlinked substructures as input. Third, we have modified the training procedure from a standard encoder-decoder framework to explicitly encode the set of possible linkers. The training process combines encodings of both the unlinked substructures and the linked molecule, as opposed to solely the starting point (Figure 2.2). In addition, there are several minor architectural differences between DeLinker and their model (further details can be found in Appendix A). The impact of these changes is detailed in Tables A.2 and A.3.

2.3.5 Experimental Setup

In all of our experiments, we used the same training set derived from the ZINC data set to train DeLinker and construct the database for the Database baseline. In both of the large-scale evaluations on the held-out ZINC test data and the CASF data set, we generated 250 linkers for each pair of starting fragments to be

assessed for both DeLinker and the Database. The number of atoms in the linker was set equal to the linker length of the original molecule for both DeLinker and the Database baseline in each of the large-scale evaluations. We also demonstrated the applicability of DeLinker using case studies from fragment linking, scaffold hopping and PROTAC design. There are minor differences in the evaluation of our generative method for the case studies. These deviations are detailed in Section 2.4 with the appropriate case study.

Generated molecules were first assessed by the 2D metrics described above. Molecules that passed the 2D property filters were assessed on the basis of their 3D shape as described above. We reported the proportion of molecules that pass the 2D metrics that meet 3D similarity thresholds.

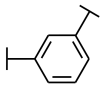
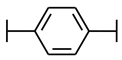
2.4 Results

We demonstrate DeLinker, a deep generative method that designs a molecule incorporating two starting substructures using 3D structural information. We first checked the impact of the structural information and then assessed our generative method in three experiments: (i) large-scale validation on ZINC (generated conformers), (ii) large-scale validation on CASF (experimentally determined active conformations), and (iii) three case studies covering fragment linking (Trapero et al., 2018), scaffold hopping (Kamenecka et al., 2009), and PROTAC design (Farnaby et al., 2019).

2.4.1 Importance of Structural Information

To assess the importance of including structural information, we empirically examined its impact on the generation process (Table 3.1). We considered three almost identical fragment-molecules pairs based on ZINC7670105 from the held-out ZINC test set (see Section 2.3). In all three cases, the starting substructures remained constant, but the substitution pattern of the benzene linker differed. This resulted in the distance and angle between the fragments changing, but no other differences between the input data to our model. We generated 1000 linkers with a maximum of

Table 2.1: Impact of structural information on generated ring substitution patterns. The generated compounds closely followed the true substitution pattern, with only the structural information provided differing between examples.

True substitution pattern	Proportion with substitution pattern		
	Ortho	Meta	Para
 Ortho	91.8%	8.2%	0.0%
 Meta	0.0%	83.1%	16.9%
 Para	0.0%	1.8%	98.2%

six atoms for each set of structural information and assessed the substitution pattern of generated molecules that contained a six-membered ring as the linker (Table 3.1).

DeLinker generated a high number of six-membered rings in all three cases (33%-54%). Most rings were generated with the para structural information. This is consistent with chemical knowledge since there are fewer possibilities given these structural constraints. The generated molecules closely followed the substitution pattern of the molecule used to calculate the structural information, with between 83% and 98% of the rings produced following the same pattern (Table 3.1). The effect of the structural information on the performance of DeLinker in a large-scale evaluation is discussed below and can be found in Tables A.2 and A.3.

2.4.2 Large-Scale Validation

Validation on ZINC. We next evaluated our method on the held-out test set from the ZINC data set, consisting of 400 pairs of fragments. We primarily compared DeLinker to a method based on database lookup ("Database"). The Database samples linkers from the same set of data used to train our method, joining the

Table 2.2: 2D performance metrics for molecules generated by DeLinker, our deep generative model, compared to a Database baseline on the held-out ZINC test set and the independent CASF data set.

	ZINC		CASF		CASF ≥ 5 atoms	
Metric	Database	DeLinker	Database	DeLinker	Database	DeLinker
Valid	100.0%	98.4%	99.0%	95.5%	98.4%	94.7%
Unique	38.8%	44.2%	43.0%	51.9%	58.3%	72.9%
Novel	0.0%	39.5%	0.0%	51.0%	0.0%	68.7%
Recovered	78.0%	79.0%	42.8%	53.7%	14.9%	29.8%
Pass 2D filters	97.0%	89.8%	95.0%	81.4%	93.6%	71.7%

fragments in one of the two possible orientations at random. This setup ensures that both methods are constructed using the same data, and allows a direct comparison to be made between database lookup and our deep learning-based generative approach.

We generated 250 linkers for each pair of fragments, resulting in 100 000 generated molecules to be assessed for both DeLinker and the Database (see Section 2.3 for details). For the evaluation of the ZINC test set, the number of atoms in the linker was set equal to the linker length of the original molecule. This is an easier test for both methods than if the linker length was assumed to be unknown, but allows us to assess whether the two methods presented are able to generate molecules that possess desired 2D chemical properties and high 3D structural similarity.

DeLinker generated a high proportion of valid molecules that passed the 2D chemical property filters (Table 2.2) and substantially outperformed the Database method by all 3D similarity measures (Table 2.3). DeLinker displayed a similar improvement over the graph-based molecular generative model of Liu et al. (2018), which performed broadly comparably to the Database method (Tables A.2 and A.3). Further metrics and an ablation study showing the effects of including different structural information can be found in Tables A.2 and A.3. Without any structural information, the deep generative model performed similarly to the Database method by the 3D similarity measures (Table A.3). Including only the distance between the fragments substantially improved performance, with further benefit from including the angle between fragments.

A molecule is deemed "valid" if it contains both starting fragments (i.e. the fragments have been linked) and its SMILES representation can be parsed by RDKit (Landrum, 2006) (i.e. satisfies atomic valency rules). The small proportion of invalid molecules produced by DeLinker (Table 2.2) was due to the fragments remaining unlinked, rather than failing atomic valency. This is a design choice by the deep learning system, and is beneficial in reducing the number of unsuitable linkers suggested.

A fundamental benefit of our deep generative method over any database is evident in the proportion of novel linkers. The Database method is unable to suggest linkers not in the database, and thus 0% of the proposals were novel. In contrast, DeLinker proposed a linker not in the training set in around 40% of suggestions, despite the training set of linkers containing over 5 000 unique linkers. Examples of novel linkers proposed by DeLinker are shown in Figure A.1.

Both methods recovered over 75% of the original molecules (Table 2.2), demonstrating that they are able to sample from the distribution of linkers effectively. However, this is in part due to the chemical redundancy of molecules in ZINC. Indeed, all of the original linkers in the held-out ZINC test set were present in the training set.

DeLinker was able to learn the 2D filters implicitly, although it produced slightly fewer molecules passing these filters than the Database method (Table 2.2). Both methods had high success rates of 95% or above for all of the individual filters (Table A.2).

For all of the 3D measures at all thresholds assessed, DeLinker produced a substantially higher proportion of linkers with the required 3D similarity than the Database (Table 2.3). In particular, at the highest levels of similarity, DeLinker generated over 80% more molecules scoring > 0.9 for SC_{RDKit} Fragments and over 60% more molecules with an RMSD < 0.5Å.

Performance of both methods is impacted by the length of the generated linkers, and in particular the number of short (three/four atoms) linkers in the test set (Table A.1), where there are a limited number of possibilities. The degree of outperformance of DeLinker over the Database increased substantially when only

Table 2.3: 3D performance metrics for molecules generated by DeLinker, our deep generative model, compared to a Database baseline on the held-out ZINC test set and the independent CASF data set. See Section 2.3.3 for a description of the metrics.

	ZINC		CASF		CASF $\geq$ 5 atoms	
Metric	Database	DeLinker	Database	DeLinker	Database	DeLinker
SC_{RDKit} Molecule						
>0.7	33.5%	47.1%	14.9%	22.3%	7.8%	16.3%
>0.8	8.5%	14.2%	3.3%	5.2%	1.1%	3.6%
>0.9	1.3%	1.8%	0.5%	0.8%	0.3%	0.8%
SC_{RDKit} Fragments						
>0.7	60.2%	71.3%	28.4%	39.1%	24.2%	38.7%
>0.8	24.7%	35.8%	8.7%	12.7%	6.1%	12.3%
>0.9	4.5%	8.2%	1.4%	2.3%	0.5%	1.6%
RMSD Fragments						
<1.00Å	46.9%	58.6%	19.4%	28.1%	14.6%	26.6%
<0.75Å	20.5%	30.0%	7.1%	10.2%	4.3%	9.3%
<0.50Å	5.7%	9.3%	2.0%	3.1%	0.9%	2.4%

considering linkers with at least five atoms (Table A.4). In this setting, DeLinker generated around 190% more molecules scoring > 0.9 for SC_{RDKit} Fragments and 130% more molecules with RMSD Fragments < 0.5Å or SC_{RDKit} Molecule > 0.9.

Validation on CASF. We saw similar performance when we evaluated the methods on the CASF data set (Tables 2.2 and 2.3). Both methods found producing 3D similar molecules more challenging for the CASF set than for the held-out ZINC set. Two possible explanations are the lower molecular similarity and use of experimentally-determined structures in the evaluation. The average Tanimoto similarity of the Morgan fingerprints (radius 2, 1024 bits, Morgan, 1965; Rogers and Hahn, 2010) of the 250 most similar starting fragments in the training set to each starting point in the test set was 0.36 for the held-out ZINC set but only 0.26 for the CASF set. However, our method was still frequently able to generate compounds with high similarity to the original molecule (Table 2.3).

In particular, DeLinker generated around 60% more molecules than the Database at the highest 3D similarity threshold (> 0.9 SC_{RDKit} Fragments and SC_{RDKit} Molecule, < 0.5Å RMSD Fragments). When restricting the evaluation to linkers with at least five atoms, the degree of outperformance substantially increased, with

DeLinker producing 200% more molecules that scored > 0.9 by SC_{RDKit} Fragments than the Database (Table 2.3).

DeLinker recovered 54% of the original linkers, compared to only 43% for the Database method, while around 50% of molecules generated by DeLinker were novel. The proportion recovered was lower than in the evaluation on ZINC, however, this set is more challenging with an average length of the true linker 5.9 atoms, compared to 4.9 for the held-out ZINC test set and 4.7 for the ZINC training set. In addition, only around 70% of the true linkers were present in the training set, providing an upper bound for the Database method. Similarly to the ZINC set, DeLinker substantially outperformed the Database method for longer linkers; DeLinker recovered around 30% of molecules with a linker of at least five atoms, twice as many as the Database method that only recovered 15% (Table 2.2).

As previously noted, a fundamental limitation of a database method is an inability to generate linkers that are not present in the database. Despite being trained on the same database of linkers, DeLinker has learnt to extrapolate from this set to novel linkers. The following is an example of when this is crucial for successful compound design.

Dequalinium is a nanomolar binder (Ki: 70 nM) of chitinase A (PDB ID: 3ARP, Figure 2.4b, Pantoom et al., 2011). One possible fragmentation of the dequalinium-chitinase complex is shown in Figure 2.4a. To recover dequalinium from these fragments requires joining them with a decane linker, which is not present in the training set of linkers and thus the Database is unable to recover the original molecule. We generated 250 molecules with DeLinker, which included several highly similar novel linkers. The five most similar measured by SC_{RDKit} Fragments are shown in Figure 2.4c. While DeLinker did not recover the decane linker within 250 generated compounds, simple chain linkers that closely resemble the true decane linker are prevalent. We compared this to an exhaustive search of linkers in the Database of the same length as the true decane linker (790 unique molecules). None of the Database generated molecules are highly similar to dequalinium (Figure 2.4d), with only one molecule with SC_{RDKit} Fragments > 0.7. In contrast, DeLinker

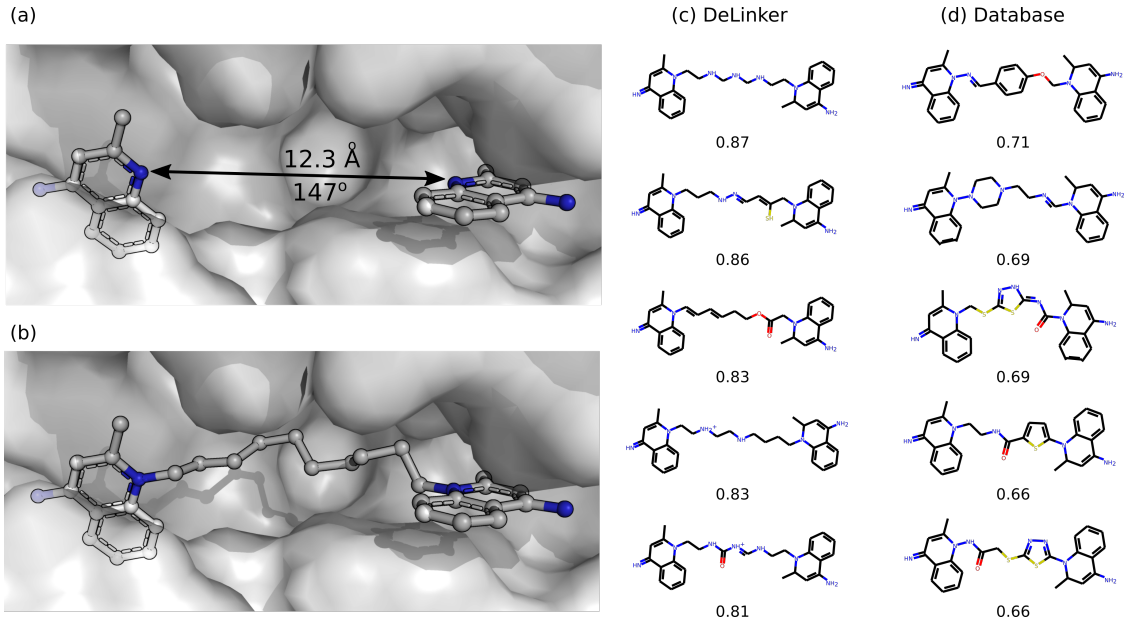


Figure 2.4: Comparison of DeLinker with an exhaustive Database search. A fragmention of dequalinium (PDB ID: 3ARP, (b)) is shown in (a). The most 3D similar molecules by SC_{RDKit} Fragments proposed by DeLinker and the Database method are shown in (c) and (d), respectively, together with the 3D similarity score. (c) DeLinker was able to produce several very similar molecules, despite limited sampling (250 samples). (d) Exhaustive search of the database was not able to recover the original molecule or produce any highly similar molecules.

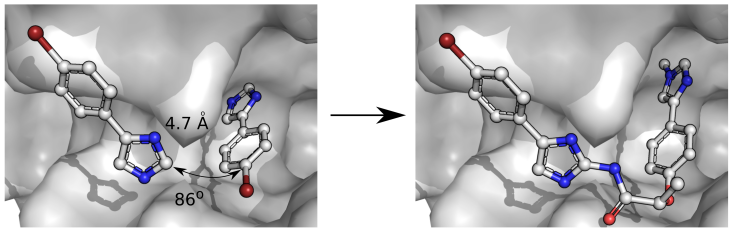
generated 34 unique molecules with SC_{RDKit} Fragments > 0.7. This illustrates the importance of *de novo* design and the limitations of any database-based solution.

Finally, we showed the applicability of our method in three diverse examples from the literature, covering fragment linking (Trapero et al., 2018), scaffold hopping (Kamenecka et al., 2009), and PROTAC design (Farnaby et al., 2019). Due to the availability of independent experimental structural data for both the initial and optimised complexes, this represents the most realistic evaluation, albeit with a limited sample size.

2.4.3 Fragment Linking Case Study

Trapero et al. (2018) considered both growing and linking strategies to create potent inhibitors of inosine 5-monophosphate dehydrogenase (IMPDH, UniProt: G7CNL4), a tuberculosis drug target. Linking proved most successful, with the authors identifying several promising compounds, the most potent with more than 1000-fold

(a)



(b)

Metric	Database	DeLinker
SC_{RDKit} Fragments		
>0.70	681	1115
>0.80	129	301
>0.85	6	18
Vina Score		
<-6.0	61	144
<Lead	51	123
<-6.5	32	80
<-7.0	14	31

Figure 2.5: Fragment linking case study. (a) Left: The inital fragment hits (PDB 5OU2). Right: The most potent experimentally verified linked molecule of Trapero et al. (2018) (PDB ID: 5OU3). DeLinker recovered this and the two other experimentally verified active molecules. (b) 3D similarity metrics and AutoDock Vina minimised affinities. Unique ligands with SC_{RDKit} Fragments > 0.8 were docked with AutoDock Vina using a local minimization. DeLinker produced more than twice as many molecules than the Database method with better Vina scores than the most potent reported binder (Lead).

improvement in affinity over the initial fragment hits. Three direct elaborations of the initial fragments were reported (compounds 29-31 in Table 4 of Trapero et al., 2018), with structures of both the initial fragments (PDB ID: 5OU2) and most potent linked compound (PDB ID: 5OU3) available (Figure 2.5a).

In previous experiments, we chose the linker length based on the number of atoms in the linker of the original molecule. To reflect prospective use more accurately, we assumed the linker length was unknown and generated 1000 linkers for each length between three and eleven atoms, inclusively. We assessed the generated linkers using the same criteria as before.

Both DeLinker and the Database method recovered all three experimentally validated compounds. However, DeLinker identified more than twice as many unique compounds as the Database with high 3D similarity (> 0.8 SC_{RDKit} Fragments) to the initial fragments (301 vs. 129, Figure 2.5b and Table A.5). The compounds meeting the above 3D similarity threshold were docked with AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013) via a local minimisation after alignment with the starting fragments. This allows us to understand whether the proposed molecules are complementary to the active site and are able to maintain the binding mode of the original fragments. Around 40% of molecules generated by both DeLinker and the Database were scored better than the most potent experimentally validated compound. As a result, DeLinker suggested more than twice as many

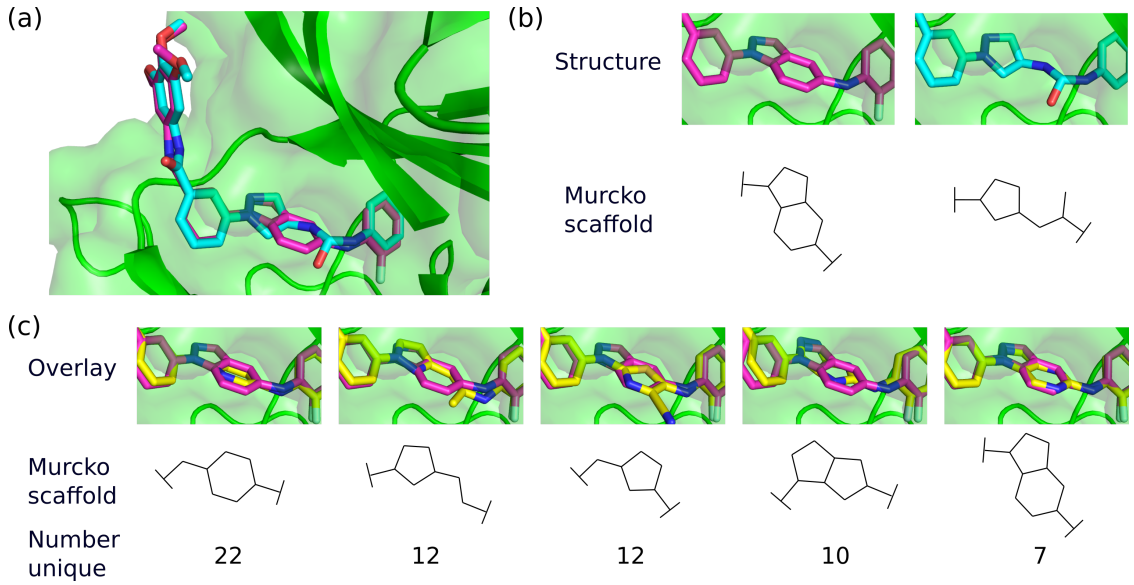


Figure 2.6: Scaffold hopping case study. (a) Overlay of the indazole (PDB ID 3FI3, magenta carbons) and aminopyrazole (PDB ID 3FI2, cyan carbons) structures, with JNK3 shown in green. DeLinker recovered both active molecules, despite neither linker being in the training set. (b) Structures of the indazole (left) and aminopyrazole (right) linkers, and their Murcko scaffolds. (c) Overlay of the indazole compound (PDB ID 3FI3, magenta carbons) and example linkers (yellow carbons) from several highly 3D similar scaffolds.

unique compounds as the Database with better docking scores than the active compound (123 vs. 51, Figure 2.5b).

2.4.4 Scaffold Hopping Case Study

Kamenecka et al. (2009) designed JNK3-selective (UniProt: P53779) inhibitors that had > 1000-fold selectivity over p38 (UniProt: Q16539), another closely related mitogen-activated protein kinase family member. Starting with an indazole class of compounds, they were not able to establish significant selectivity for JNK3 over p38. However, changing scaffolds led to an aminopyrazole linker that afforded compounds with > 2800-fold selectivity. The two inhibitors displayed nearly identical binding mode (RMSD 0.33 Å, Figure 2.6a) and affinity for JNK3 (indazole: IC50 12 nM, aminopyrazole: IC50 25 nM), but significantly different binding affinity to p38 (indazole: IC50 3.2 nM, aminopyrazole IC50 3.6 μM).

Starting with the indazole-based inhibitor (PDB ID: 3FI3), we explored the ability of our method to change molecular scaffold, in particular towards the

aminopyrazole-based inhibitor (PDB ID: 3FI2). We generated 5 000 linkers with both eight and nine atoms, and assessed the generated linkers using the same criteria as before. In particular, we focussed on the diversity of molecular scaffolds proposed by DeLinker that satisfied the 3D structural information and could adopt a highly similar conformation to the original indazole-based inhibitor.

Of the 10 000 compounds generated by DeLinker, there were 2 688 unique compounds that satisfied the 2D chemical filters (Table A.6). 699 of these had an SC_{RDKit} Fragments score above 0.75, of which 627 were not in the training set (89.7% novel). These compounds covered 182 unique generic Murcko scaffolds (Bemis and Murcko, 1996). Five of the most common are shown in Figure 2.6b, together with an example linker and the number of unique linkers generated with the same generic Murcko scaffold that also met the 3D similarity threshold. The examples from all five scaffolds show almost perfect overlap with the indazole linker, while maintaining the conformation of the remainder of the molecule.

In addition, DeLinker recovered both the indazole- and aminopyrazole-based linkers, despite neither being present in the training set.

2.4.5 PROTAC Case Study

Farnaby et al. (2019) developed PROTAC degraders of the BAF ATPase subunits SMARCA2 (UniProt: P51531) and SMARCA4 (UniProt: P51532) using a bromodomain ligand and recruitment of the E3 ubiquitin ligase VHL (UniProt: P40337). They first designed a PROTAC by combining known binders of SMARCA2/4 and E3 ubiquitin ligase VHL using poly(ethylene glycol)-based linkers (PDB ID: 6HAY, Figure 2.7a). The linker was then optimised to improve interactions with the lipophilic face created in part by Y98 of the VHL protein. In particular, they designed the linker to mimic the conformation observed in the ternary complex structure, resulting in improved molecular recognition (PDB ID: 6HAX). This was confirmed by the two crystal structures displaying near identical ternary complexes (Figure 2.7b).

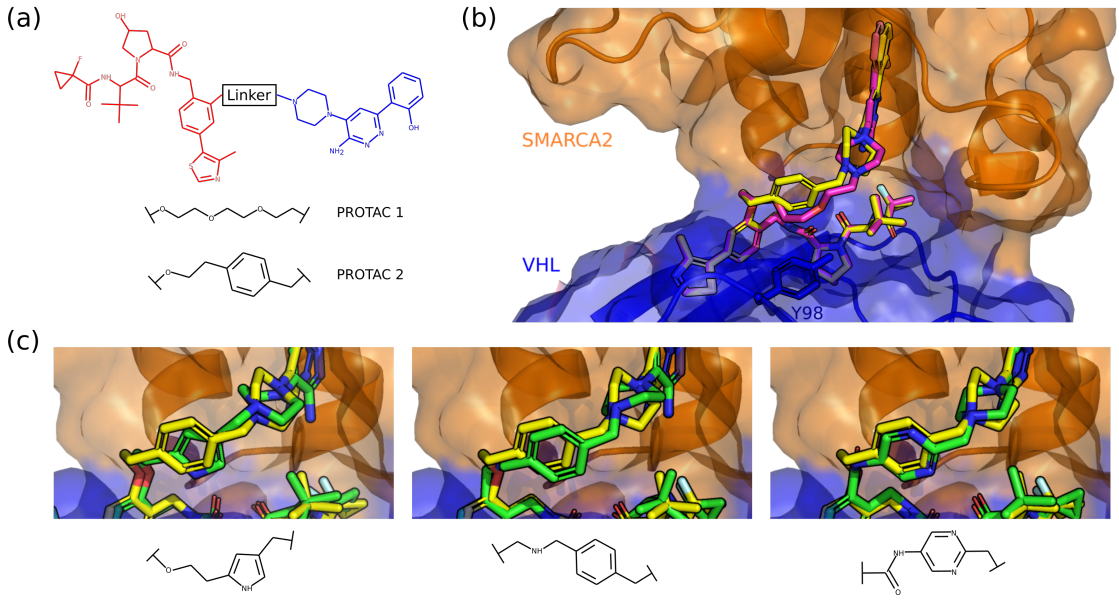


Figure 2.7: PROTAC design case study. (a) Two-dimensional chemical structures of PROTAC 1 and PROTAC 2. (b) Overlays of ternary crystal structures of PROTAC 1 (PDB ID 6HAY, magenta carbons) and PROTAC 2 (PDB ID 6HAX, yellow carbons), with SMARCA2 shown in orange, VHL in blue. (c) Overlays of three linkers with different scaffolds produced by DeLinker (green carbons); all three accurately recapitulate the linker geometry observed in PROTAC 2 (yellow carbons). None of these linkers were present in the training set.

We investigated the ability of our model to design alternative linkers to the known polyethylene glycol-based linker (PDB ID: 6HAY) that could maintain the same conformation observed in the ternary complex. We generated 5 000 linkers with a maximum of either nine or ten atoms. There were almost 3 000 unique linkers that passed the 2D chemical filters (Table A.7).

Due to the size and complexity of the PROTAC, we generated conformers constraining the two starting substructures (Figure 2.7a) to adopt poses close to their known binding conformation, removing any high energy poses. DeLinker produced 2150 unique compounds with SC_{RDKit} Fragments > 0.85, of which three novel linkers that accurately recapitulate the linker geometry observed in PROTAC 2 are shown in Figure 2.7c. In all three cases, the aromatic systems perfectly align with that of PROTAC 2, and are likely to fulfil the goal of improving interactions with the lipophilic face compared to PROTAC 1. In particular, the pyrrole-based linker (Figure 2.7c, left) appears to be making an NH-π interaction with the

Y98 residue, possibly improving the CH-π interaction being made by the benzene in PROTAC 2. When conformations of these structures were minimised using AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013), the compound with the pyrrole-based linker scored comparably to PROTAC 2 (-14.17 vs. -14.32) with both scoring substantially better than PROTAC 1 (-12.66). However we note that AutoDock Vina does not model electrostatics explicitly and thus does not capture NH-π or CH-π interactions. The other two compounds shown (Figure 2.7c, center, right) scored better than PROTAC 2 (-14.81 and -14.33 respectively), as did almost 20% of all the minimised compounds (536).

For each of the three case studies, the top 20 molecules generated by DeLinker that met the 3D similarity threshold (SC_{RDKit} Fragments $>$ 0.80) ranked by AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013) score are shown in Figures A.2-A.4.

2.5 Discussion

We have developed a graph-based deep generative method for fragment linking or scaffold hopping that integrates 3D structural information, utilising the relative distance and orientation between the starting substructures in the design process. Unlike previous attempts at computational fragment linking or scaffold hopping, our method does not rely on a database of fragments from which to select a linker but instead designs one given the fragments provided and 3D information.

Through two large-scale assessments, we have demonstrated that our generative method is able to learn to produce a distribution of linkers that matches the constraints present in the training set, while being able to generalise to novel linkers that satisfy both 2D and 3D constraints. In addition, the generated molecules consistently have high 3D similarity to both the initial fragments and the original molecules, outperforming a database baseline by 60% in the evaluation on CASF, increasing to 200% when restricting the evaluation to linkers with at least five atoms.

Finally, through three case studies, we have shown that our method can be applied to fragment linking, scaffold hopping, and PROTAC design. In the fragment

linking example, our method reproduced all of the reported potent molecules using only the crystal data of the initial fragment hits. In addition, in docking-based evaluation, many of the generated molecules were scored more highly than the original hits, while maintaining similar binding modes. In the scaffold hopping case study, our method reproduced both the starting and final molecule, while suggesting many other scaffolds with high 3D similarity to the initial crystal data. Finally, in the PROTAC design case study, our method suggested a range of novel linkers that met the design goal of maintaining the linker geometry of PROTAC 1, while improving interactions with the lipophilic face created in part by residue Y98 of the VHL protein.

Our generative model can be readily combined with previous methods for fragment linking or scaffold hopping (e.g. Böhm, 1992b; Maass et al., 2007). This can be achieved by using the compounds generated by DeLinker to augment the database that is used as input for the search and filtering methods adopted by existing database-based methods.

As is frequently the case with machine learning-based generative models, some of the molecules generated by DeLinker might have unstable functional groups or be hard to synthesise. We have demonstrated that our method was able to learn to produce molecules that frequently passed the 2D filters (which included a measure of synthetic accessibility) used to construct the training set through implicit supervision alone, thus altering the training set composition would likely reduce the number of undesirable molecules generated. The properties of generated molecules could also potentially be improved through direct supervision with reinforcement learning (Sanchez-Lengeling et al., 2017) or filtered via a post-processing approach (Sumita et al., 2018).

As far as we are aware, this is the first molecular generative model to incorporate 3D structural information directly in the design process. Currently the only 3D information utilised by the model is the distance between the fragments or starting substructures and their relative orientations. This provides explicit constraints for a given compound, but only implicit information about the shape of the binding

site. Despite this minimal parametrisation, there is a substantial impact on the generated molecules.

Extending our method to use additional structural information that incorporates further constraints from the protein is a direction for future research. This could be achieved directly by providing our method with a richer representation of the protein-ligand binding site, or indirectly through the combination of reinforcement learning (Williams, 1992) and existing methods from structure-based drug discovery (e.g. Trott and Olson, 2010; Imrie et al., 2018). Both directions promise substantial benefits for structure-based molecular generative methods.

While we have shown that our method can be applied to fragment linking, scaffold hopping, and PROTAC design, we believe that our framework is general and readily extendable to other design tasks. Applications of DeLinker to other design tasks such as fragment growing, R-group optimisation, and macrocycle design would be interesting avenues for further studies.

In the following chapter, we propose an extension of the DeLinker framework that uses a convolutional neural network to incorporate physically-meaningful 3D structural information, providing a richer prior for the generative process. In addition, we demonstrate that our method can be applied to molecular elaboration tasks, such as R-group design, by changing only the training set with no other modifications to the methodology necessary.

The key to growth is the introduction of higher dimensions of consciousness into our awareness.

— Vilayat Inayat Khan, *Toward the One*

3

Deep Generative Design with 3D Pharmacophoric Constraints

Contents

This chapter contains work described in the following publication:

Fergus Imrie, Anthony R. Bradley, and Charlotte M. Deane (2020). Constrained Deep Generative Linker Design Using 3D Structural Priors. *NeurIPS Workshop on Machine Learning for Molecules.*

A manuscript describing the work presented in this chapter for submission to a journal is under preparation. The experiments for scaffold elaboration in

Sections 3.4.3 and 3.4.4 were performed in collaboration with Thomas Hadfield.

3.1 Preface

In the previous chapter, we described DeLinker, a graph-based deep generative method that integrates 3D structural information into the generation process. We demonstrated the application of our method for fragment linking, scaffold hopping, and PROTAC design. While the distance and angle between the two starting substructures is a powerful constraint on the possible set of linkers, it is far from a comprehensive representation of the binding site and is not directly applicable to other design tasks, such as scaffold elaboration.

In this chapter, we propose an extension of the DeLinker framework that uses a convolutional neural network to incorporate physically-meaningful 3D representations of molecules and target pharmacophores, providing a richer prior for the generative process. In addition, we demonstrate that our method can equally be applied to scaffold elaboration, such as R-group design or fragment growing. This greatly extends the applicability of our method in hit-to-lead and lead optimisation scenarios.

The 3D pharmacophoric information results in improved generation and allows greater control of the design process. In multiple large-scale evaluations, we show that including 3D pharmacophoric constraints results in substantial improvements in the quality of generated molecules. On a challenging test set derived from PDBbind, our model improves the proportion of generated molecules with high 3D similarity to the original molecule by over 250% and recovers around ten times more of the true linkers and R-groups compared to the baseline DeLinker method. Our approach is general-purpose, readily modifiable to alternate 3D representations, and can be incorporated into other generative frameworks.

3.2 Introduction

As described in Chapter 1, drug design optimises molecules through a multi-step, iterative process in order to achieve a desired biological response. The size of the search space (Polishchuk et al., 2013) and discontinuous nature of the optimisation landscape (Stumpfe and Bajorath, 2012) are two key factors contributing to the difficulty of this problem and, as a result, currently molecular design is typically led by human experts.

Machine learning models for molecule generation (e.g. Gómez-Bombarelli et al., 2018; Segler et al., 2018; Jin et al., 2018) offer an alternative approach to human-led design or rules-based transformations (such as Brown et al., 2004; Besnard et al., 2012), as discussed in Chapter 1. Despite recent success (Zhavoronkov et al., 2019), for these methods to be broadly adopted in drug discovery, more control over the generative process is required, including the ability to incorporate prior knowledge.

In the hit-to-lead (or lead generation) and lead optimisation stages of drug discovery, the goal is to improve one, or several, properties. This is typically achieved by modifying an existing molecule rather than designing a compound from scratch. Such modifications can be broadly categorised into one of two scenarios: linker design and scaffold elaboration.

As discussed in Chapter 2, linker design is a general problem in drug discovery capturing a wide range of tasks where the goal is to design a molecular scaffold that incorporates two (or more) specific substructures. Scaffold hopping (Böhm et al., 2004; Langdon et al., 2010), fragment linking (Ichihara et al., 2011; Bienstock, 2015), and PROTAC design (Troup et al., 2020; Li and Liu, 2020) are three key applications that can all be considered as linker design. Examples of these design tasks are shown in Figure 3.1a-c.

In contrast to linker design which is tasked with discovering molecular cores, scaffold elaboration proposes molecules incorporating these privileged substructures. Scaffold elaboration covers a broad range of medicinally important scenarios, such as R-group optimisation (Guha, 2013) and fragment growing (Bienstock, 2015; Lamoree and Hubbard, 2017) (Figure 3.1d-e). R-group optimisation is utilised in

almost all drug discovery projects to improve the potency, selectivity, and other properties of a molecule during lead optimisation and characterise the structure-activity relationship (SAR) of a molecular series, while growing is the primary method for elaborating fragment hits.

Recently, several deep learning methods have been proposed to address these design challenges. In Chapter 2, we described the first application of deep learning for molecular linker design ("DeLinker", Imrie et al., 2020), reporting substantial improvement over a database-based approach, the previous *de facto* computational method for this task, by including basic structural information. Yang et al. (2020b) proposed an alternative model ("SyntaLinker") based on the transformer architecture and a SMILES-based representation. Their model did not incorporate structural information but instead included 1D molecular patterns capturing factors such as the shortest linker bond distance.

Deep learning approaches have also been proposed for scaffold elaboration. Graph-based approaches were proposed by Lim et al. (2020) and Li et al. (2020b). The scaffolds employed in both methods do not have explicit attachment points. As such, these methods are primarily applicable to the general generation of molecules with a privileged scaffold or substructure, rather than tasks such as R-group design. In contrast, Arús-Pous et al. (2020) developed a preprocessing formulation to permit a SMILES-based approach that requires specific attachment points to be defined.

In both linker design and scaffold elaboration, some knowledge about the desired modification is typically available (Papadatos and Brown, 2013); this can either be derived from the protein binding site in the case of structure-based design (Anderson, 2003), or from other molecules in ligand-based drug discovery (Sliwoski et al., 2014). In either case, this information has strong 3D dependencies which should be taken into account. However, currently this information, which is crucial to successful compound design, is typically not utilised by generative models (Xia et al., 2020).

None of the existing machine learning models for linker design or scaffold elaboration effectively utilise structural information, with DeLinker the only framework explicitly incorporating any 3D information in the form of the distance between

the starting substructures and their relative orientations. While this minimal parametrization had a substantial impact on the quality of the generated molecules (Imrie et al., 2020), much of the key information about the characteristics of the binding site is not taken into account in the generative process.

There have been several recent approaches proposed to generate molecules from 3D representations. Skalic et al. (2019a) generated molecules from a 3D representation of a seed ligand. However, their approach requires a known active molecule, only provides 3D information implicitly to seed their model, and offers no further control over generated compounds. As a result, their generative model recovered fewer than 2% of seed molecules. This idea was extended in Skalic et al. (2019b) to generate the ligand representation from the protein target. While this alleviates the need for a known active, it is not possible to use prior knowledge to influence the ligand representation. In concurrent work to this chapter, Ragoza et al. (2020) and Masuda et al. (2020) both adopt an autoencoder framework to generate atomic densities, before using a fitting procedure to convert the continuous grids to discrete molecular structures.

However, all prior approaches utilising 3D representations attempt to generate entire molecules and do not readily incorporate expert knowledge. While this is arguably the end-goal for molecular design, in practice this limits the applicability of such methods. In particular, it prevents their use in later stage drug discovery where there is significant prior knowledge that should inform compound design.

In this chapter, we propose DeLinker-3D, a graph-based generative model that uses a convolutional neural network (CNN) to incorporate physically-meaningful 3D structural information, here provided as 3D pharmacophores (Schaller et al., 2020), a general and widely-used representation in cheminformatics. Our model is applicable to a wide variety of design tasks in the hit-to-lead and lead optimisation stages of drug discovery, covering linker design and scaffold elaboration. Importantly, the richer representation of the binding site readily and naturally allows the incorporation of domain knowledge and significantly improves the quality of generated compounds. On a challenging test set derived from PDBbind, our model improves the proportion

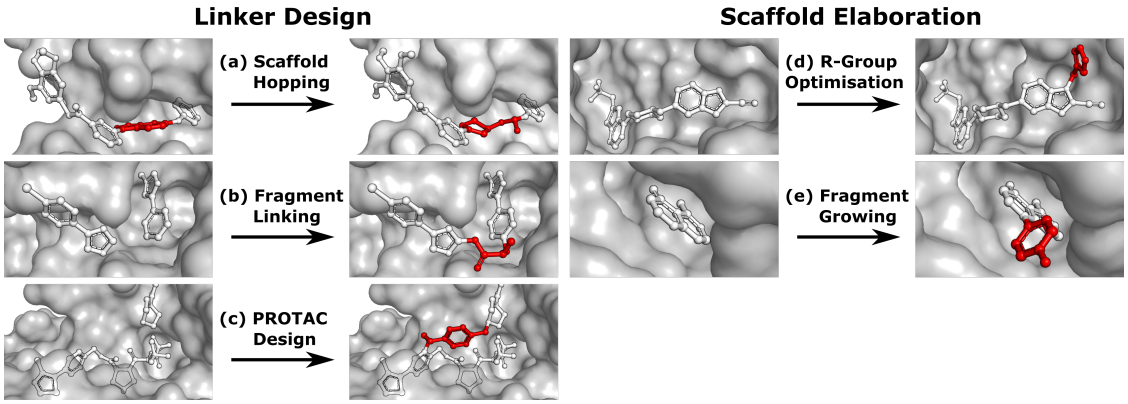


Figure 3.1: Design tasks considered in this chapter. (a)-(c) cover linker design, specifically (a) scaffold hopping, (b) fragment linking, and (c) PROTAC design. (d)-(e) scaffold elaboration, namely (d) R-group optimisation and (e) fragment growing. Components of the ligand that are modified or added in the design process are shown in red.

of generated molecules with high 3D similarity to the original molecule by over 250%. In addition, DeLinker-3D recovers around ten times more of the original molecules compared to the baseline DeLinker method.

3.3 Methods

In this chapter we describe DeLinker-3D, a deep learning approach combining GNNs and CNNs for molecular linker design and scaffold elaboration. We extend current molecular generative methods to incorporate physically-meaningful 3D structural information, enabling prior knowledge to be readily incorporated and allowing greater control of the generative process by domain experts. Our underlying model is based on the DeLinker framework described in Chapter 2, which built on the generative process introduced by Liu et al. (2018) that constructs molecules "bond-by-bond" in a breadth-first manner. Here we outline the generative process and describe how 3D structural information is incorporated (Figure 3.2).

3.3.1 Generative Process

To perform the generative tasks considered in this chapter, DeLinker-3D takes as input (i) the chemical structure of either the substructures that are to be linked or the molecular scaffold that is to be elaborated and (ii) a 3D structure of the partial

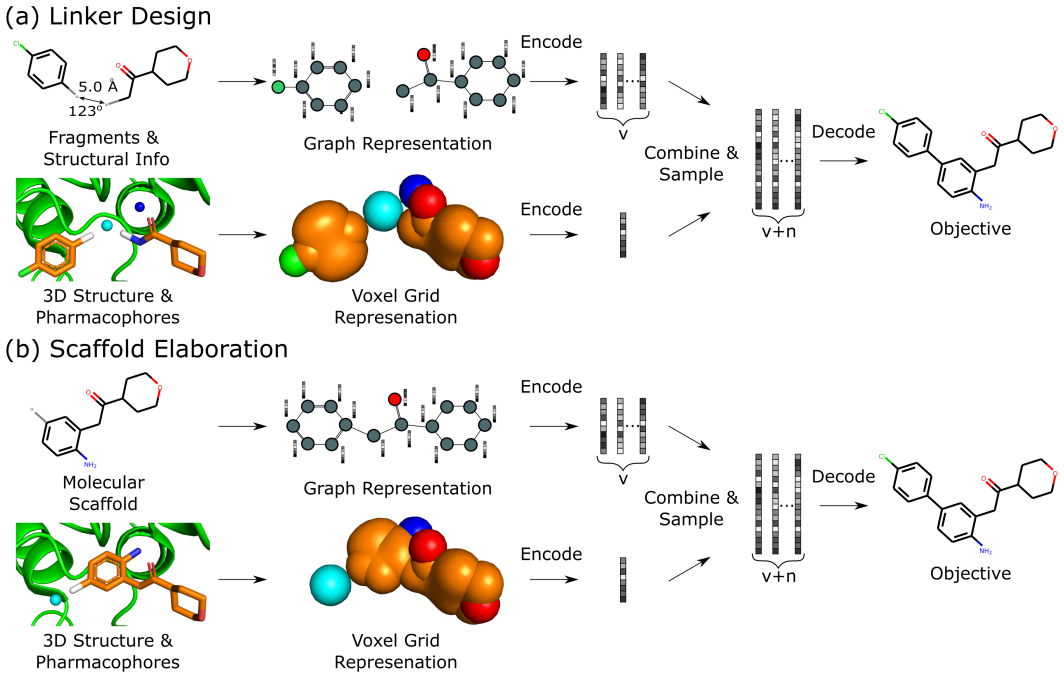


Figure 3.2: Overview of DeLinker-3D. The starting structures and 3D pharmacophore map are converted into a graph representation and a voxel grid, respectively. These are fed into GNN and CNN encoders, respectively. The featurisations are combined and decoded by a GNN-based decoder.

molecule and the desired pharmacophoric features. The input to DeLinker-3D can be seen in Figure 3.2 for both linker design and scaffold elaboration.

Pharmacophores are a widely-used representation in cheminformatics (Schaller et al., 2020). They are designed to capture the key chemical interactions that allow ligands to bind to macromolecular targets, such as hydrogen bonds, charges, or lipophilic contacts. In this chapter, we utilise 3D pharmacophores. These pharmacophores can be derived both from other molecules (ligand-based) and inferred or proposed based on the protein target of interest (structure-based), making this representation broadly applicable.

Due to their prevalence and importance in drug discovery, the pharmacophores included in our representation were hydrogen bond donors, hydrogen bond acceptors, and aromatic systems. Our framework is readily extendable to additional pharmacophores, or alternate structural representations.

To generate new molecules, first, a graph representation of the starting substructure(s) is constructed and nodes are encoded using a gated graph neural

network (GGNN, Li et al., 2016). The 3D structure of the starting molecular fragment(s) and desired pharmacophores is voxelised to construct a 3D grid, with atoms and pharmacophores adopting a Gaussian representation centered at their input coordinates (Sunseri and Koes, 2020, Figure 3.2). The voxel grid representation is passed into a 3D convolutional neural neural network composed of three $3 \times 3 \times 3$ convolutional layers with ReLU activation, each followed by a $2 \times 2 \times 2$ max pooling layer, with the final convolutional layer followed by a global max pooling operation. A fully-connected layer then produces the 3D structural encoding.

For linker design, the distance and angle between the starting substructures has been shown to provide a useful constraint (Chapter 2). However, this representation is not readily extendable to scaffold elaboration, and thus this information is only provided for linker design. The 3D structural encoding is concatenated with the distance and angle information (in the case of linker design) and a 1D count vector representing the number of each pharmacophoric feature that should be present in the generated molecule. This forms the structural information, $\boldsymbol{D}$, used by the decoder to generate molecules.

From these embeddings, molecules are generated in the same way as described in Chapter 2. The decoding process is initialised with the node encodings together with a set of expansion nodes whose feature vectors are drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Each node is labeled with an atom type sampled from a classifier applied to the concatenation of the node encoding and the structural information, $\boldsymbol{D}$.

Molecules are constructed iteratively "bond-by-bond" from this set of nodes. After each step, the node encodings are updated by a decoder GGNN. Edges and their edge types are chosen based on the feature vector for the (possible) edge between node v and candidate node u given by

$$\phi_{v,u}^t = [t, \boldsymbol{s}_v^t, \boldsymbol{s}_u^t, d_{v,u}, \boldsymbol{H}^0, \boldsymbol{H}^t, \boldsymbol{D}] \,,$$

where $\boldsymbol{s}_v^t = [\boldsymbol{z}_v^t, \boldsymbol{l}_v]$ is the concatenation of the hidden state of node v after t steps ($\boldsymbol{z}_v^t$) and its atomic label ($\boldsymbol{l}_v$), $d_{v,u}$ is the graph distance between v and u, $\boldsymbol{H}^0$

is the average initial representation of all nodes, $\boldsymbol{H}^t$ is the average representation of nodes at generation step t, and $\boldsymbol{D}$ represents structural information.

Our model is trained using the same loss function as DeLinker (Chapter 2), which is similar to the standard VAE loss, including a reconstruction loss and a Kullback-Leibler (KL) regularisation term:

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda_{KL}\mathcal{L}_{KL}\,.$$

No extra terms are included to regularise the CNN encoding. We use the same hyperparameters for training as discussed in Chapter 2. For additional details regarding the underlying model see Chapter 2.

3.3.2 Data Sets

In line with the previous chapter, due to the lack of experimental data, we constructed sets for training and evaluation from general molecular data sets using standard transformations from matched-molecular pair analysis (Hussain and Rea, 2010). For both linker design and scaffold elaboration, we used the same underlying data and adopted the same process for constructing datasets, with the main difference the transformation used. For linker design, we enumerated all double cuts of acyclic single bonds that were not within functional groups, while for scaffold elaboration we performed single cuts.

The training sets were derived from the subset of ZINC (Sterling and Irwin, 2015) selected at random by Gómez-Bombarelli et al. (2018) using the fragmentation procedure described above. For linker design, this results in c. 418,000 fragment-molecule pairs and is the same training set as in Chapter 2, while for scaffold elaboration there are c. 427,000 examples.

To evaluate our method, we constructed test sets for linker design and scaffold elaboration from CASF-2016 (Su et al., 2019) and the PDBbind Refined Set (Liu et al., 2017) (v. 2019) using the same fragmentation procedure used to construct the training set. For both of the CASF and PDBbind test sets, we only retained examples with elaborations containing at least 5 atoms. In addition, for the

PDBbind test sets, we ensured that the molecular elaboration was unique and was not present in the training set. As a result, the CASF test sets contain 188 and 237 examples for linker design and scaffold elaboration, respectively, while the PDBbind test sets contain 311 and 295 examples, respectively. Due to the stricter inclusion criteria, the PDBbind test sets represent a significantly more challenging test than the CASF sets and should better capture the ability of a method to extrapolate to new linkers and elaborations.

3.3.3 Evaluation Metrics

We adopted a similar evaluation procedure as in Chapter 2, assessing the generated molecules with a range of 2D and 3D metrics. After first checking the generated molecules for validity, uniqueness, and novelty, we then determined if the generated examples were consistent with the 2D property filters used to produce the training set. While there are likely a number of molecules that would fulfil the desired criteria of the user, the original molecule is one of the "true" correct answers and represents the best single ground truth. As a result, a primary evaluation metric was the the recovery rate, which measures in how many cases the original molecule was recovered by the generation process.

Molecules which passed the 2D property filters were assessed on the basis of their 3D shape. We calculated 3D similarity by scoring conformers of the generated molecules against the original molecule using the same 3D shape and colour score utilised in Imrie et al. (2020) based on the methods described in Putta et al. (2005) and Landrum et al. (2006). For both linker design and scaffold elaboration, we primarily assessed the 3D complementarity of the generated molecular component (i.e. the linker or R-group) with the reference structure (SC_{RDKit} Generated). This score ranges between 0 (no match) and 1 (perfect match). Scores above 0.6 indicate a good match, while scores above 0.9 suggest an almost perfect match.

The focus of our analysis is based on SC_{RDKit} Generated as it captures the chemical differences between the proposed molecules. However, for linker design we also calculated the 3D metrics utilised in Chapter 2 (SC_{RDKit} Molecule, SC_{RDKit}

Fragments, RMSD Fragments) to ensure that the proposed linkers satisfy the basic structural constraints. We did not use these metrics in the scaffold elaboration experiments since the conformation of the molecular core is typically largely unaffected by its side chains (Malhotra and Karanicolas, 2017).

For each proposed compound, we generated 3D conformers using RDKit (Landrum, 2006), adopting the filtering and sampling procedure proposed by Ebejer et al. (2012). To calculate SC_{RDKit} Generated, we generated conformers in a constrained manner, biasing conformations towards those that maintained the conformation of the starting structure(s) but removing high energy conformers. We then scored all conformers, taking the best score as the final score for a particular molecule.

3.3.4 Comparison to Other Methods

For both linker design and scaffold elaboration, we compared DeLinker-3D to the original version of DeLinker (Imrie et al., 2020) and a version of the DeLinker method which is provided with the number of each pharmacophoric feature that should be present in the generated linker ("DeLinker-Counts"). The difference between DeLinker-3D and these two baselines is the structural information, $\boldsymbol{D}$, included in the feature vector, $\phi^t_{v,u}$. This comparison allowed us to assess directly the impact of (1) not including pharmacophoric information (DeLinker), (2) including such information as a 1D count vector (DeLinker-Counts), and (3) providing pharmacophoric constraints as a physically-meaningful 3D structural representation (DeLinker-3D).

We also compared our results to recent deep learning methods for these design problems. For linker design, we compared our method to SyntaLinker (Yang et al., 2020b), while for scaffold elaboration, we benchmarked against Arús-Pous et al. (2020) ("REINVENT"). Both methods adopt a SMILES-based formulation and neither framework incorporates 3D information in the design process. In both cases, we retrained these models on the training sets described above using the open-source implementations to ensure a fair comparison between the methods

tested. We adopted the same settings and hyperparameters described in the original publications.

3.3.5 Experimental Setup

In all of our experiments, we used the same training sets (one for linker design and one for scaffold elaboration) derived from the ZINC data set to train all of the models considered. In the evaluations on the data sets derived from CASF and PDBbind, we generated 250 molecules for each example for each of the methods considered. For the DeLinker-based models, the number of atoms was set equal to the number of atoms in the original molecule. In the case of SyntaLinker, the model was provided with the shortest linker bond distance. We also demonstrate the applicability of DeLinker-3D to scaffold elaboration using an R-group optimisation case study.

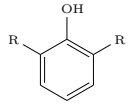
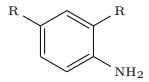
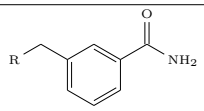
3.4 Results

We validated the ability of our deep generative model (DeLinker-3D) to perform linker design and scaffold elaboration using 3D pharmacophoric information. Through the use of several canonical examples, we demonstrated the impact of the pharmacophoric constraints on the generated molecules. We report a significant improvement in the quality of generated molecules in large-scale evaluations on test sets derived from CASF and PDBbind, further demonstrating the importance of including pharmacophoric information. Finally, we have demonstrated our approach in an R-group optimisation case study derived from the literature.

3.4.1 Importance of Pharmacophoric Constraints

We assessed the impact of pharmacophoric constraints on the generation process empirically using two canonical examples for linker design and one example for scaffold elaboration (Table 3.1). The examples were all chosen from the PDBbind test sets (see Section 3.3.2) and therefore none of the target elaborations were included in the training set. We generated 1000 molecules for each example with the three DeLinker-based methods.

Table 3.1: Impact of pharmacophoric constraints. Including 3D pharmacophoric information (DeLinker-3D), substantially improves the ability to generate molecules with desired interaction patterns.

Design Task	Target Elaboration	Method	Recovered	No. Geometric Isomers	No. Include Pharmacophores
Linker Design		DeLinker	No	0	14
		DeLinker-Counts	No	1	12
		DeLinker-3D	Yes	41	148
		DeLinker	No	0	13
		DeLinker-Counts	Yes	16	19
		DeLinker-3D	Yes	47	60
Scaffold Elaboration		DeLinker	No	0	0
		DeLinker-Counts	No	0	0
		DeLinker-3D	Yes	3	7

Only DeLinker-3D was able to recover both of the canonical examples for linker design, with the original version of DeLinker not generating the correct linker in either case. The difference between the methods is further exemplified when considering geometric isomers with the same chemical structure but possibly different substitution patterns of the exit vectors and substituent. DeLinker-3D frequently generated linkers matching the chemical structure of the linker (41-47), while DeLinker did not produce a single geometric isomer. When considering molecules which included the desired pharmacophoric pattern of the examples (aromatic ring with correct substituent group) the improved performance of DeLinker-3D continued, with a significantly larger proportion of the generated molecules containing the desired pharmacophoric features when the 3D information was provided (60-148) compared to not providing this information (13-14, DeLinker) or providing only 1D pharmacophore counts (12-19, DeLinker-Counts).

The most notable difference in generated molecules occurred in the phenol example, where only one geometric isomer was generated by DeLinker and DeLinker-Counts combined. This contrasts with DeLinker-3D generating 41 such molecules. This is a particularly difficult example for both DeLinker and DeLinker-Counts due to the presence of a donor-acceptor group, but illustrates the necessity of adopting a 3D representation.

Table 3.2: Linker design. PDBbind set results (see Section 3.3.3 for definitions of the metrics).

Metric	SyntaLinker	DeLinker	DeLinker-Counts	DeLinker-3D
Valid	11.4%	96.8%	90.2%	92.8%
Unique	93.5%	84.7%	77.1%	76.3%
Novel	54.1%	82.5%	86.8%	87.6%
Recovered	0.3%	2.3%	10.0%	22.2%
Pass 2D filters	81.3%	64.2%	60.4%	63.8%
SC_{RDKit} Generated				
>0.6	11.4%	9.9%	17.9%	25.2%
>0.7	5.7%	4.3%	9.1%	13.5%
>0.8	2.6%	1.7%	4.1%	6.0%
>0.9	1.1%	0.4%	1.2%	1.7%

The scaffold elaboration example proved challenging for all methods, primarily due to the size of the elaboration. Only DeLinker-3D recovered the 3-methyl-benzamide elaboration, with no other method generating a single geometric isomer. In addition, DeLinker-3D was the only method to generate any elaborations containing the desired functionality of an aromatic system with an amide side-chain.

These examples demonstrate the importance of including pharmacophoric constraints for both linker design and scaffold elaboration. In all cases, it was only possible to consistently generate molecules with specific pharmacophoric profiles by including 3D pharmacophoric information.

3.4.2 Linker Design Experiments on Large Test Sets

DeLinker-3D substantially outperformed all other methods on both the CASF and PDBbind test sets, with significant improvements in both the number of true linkers recovered and the proportion of generated molecules with high SC_{RDKit} Generated. This was achieved with limited impact on the uniqueness of the generated molecules and their ability to pass basic 2D chemical filters (Tables 3.2, B.1, Figure 3.3).

In comparison, SyntaLinker performed poorly in particular as measured by the 2D metrics, producing weaker results than were reported in its original publication

(Yang et al., 2020b). SyntaLinker produced a low proportion of valid molecules and recovered only 0.3% of the original molecules. Due to the comparatively weak results, we focus the remainder of our analysis on the three DeLinker-based methods. For further discussion on SyntaLinker, see Section B.1.

The proportion of valid molecules generated by the DeLinker methods was high in all cases (>85%) with similar proportions of novel molecules proposed (69-71% on CASF, Table B.1; 83-88% on PDBbind, Table 3.2). As is expected, as more structural information is provided to the model, the proportion of unique molecules decreased due to the constraints on the generative process. However, 58% and 76% of the molecules produced DeLinker-3D on CASF and PDBbind, respectively, were unique, demonstrating that the model still samples from chemical space and has not experienced mode collapse, degrading to a single or small number of solutions.

Incorporating pharmacophoric information substantially increases the recovery rate of the original molecules. On the CASF set, DeLinker-3D recovered 50%, compared to 30% for DeLinker and 42% for DeLinker-Counts. The PDBbind set is particularly challenging with DeLinker only able to recover 2.3% of the original molecules, while a database-based method would not be able to recover any, due to there being no overlap with the training set. Including the count of each pharmacophore present in the original linker increased the proportion recovered to 10% (DeLinker-Counts). Crucially, providing this information as a 3D structural representation offered a significant benefit over simply providing the pharmacophore counts. On the PDBbind test set, DeLinker-3D recovered 22.2% of the original molecules, almost ten times as many as DeLinker and more than twice as many as DeLinker-Counts (Table 3.2).

A significant improvement is also seen when assessing the 3D similarity of the generated linkers to the original ones. DeLinker-3D improved the proportion of molecules with high structural similarity (SC_{RDKit} Generated > 0.8) by 250% and 45% compared to DeLinker and DeLinker-Counts, respectively, on the PDBbind test set, with similar improvements on the CASF set (Table B.1).

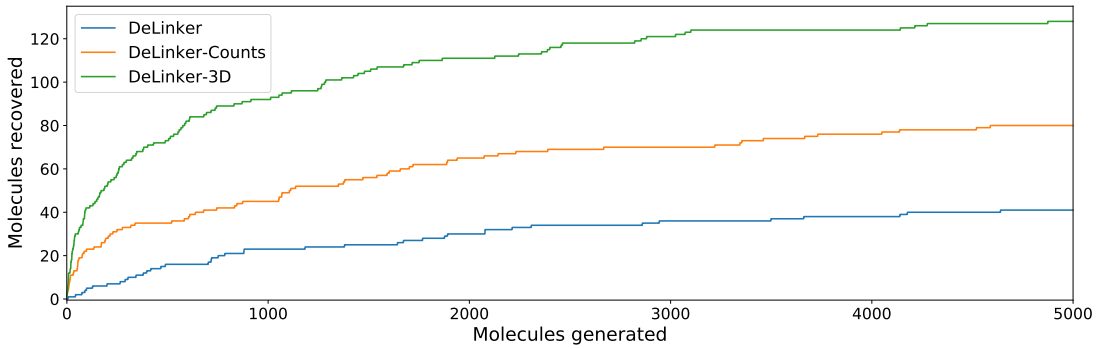


Figure 3.3: Linker design. Number of original molecules recovered as the number of generated molecules is increased. DeLinker-3D recovered the significantly more of the original molecules than both baselines for any number of samples generated.

In addition to SC_{RDKit} Generated, we also calculated the 3D metrics employed in Chapter 2, namely SC_{RDKit} Molecule, SC_{RDKit} Fragments, and RMSD Fragments. These metrics primarily capture whether the molecular linker allows the original substructures to adopt similar conformations, with the chemical features of the linker having limited to no effect on this score. The linkers generated by DeLinker-3D showed a substantial improvement on CASF compared to both DeLinker and DeLinker-Counts (Table B.2), while performing similarly on PDBbind (Table B.3).

As previously stated, these metrics primarily assess whether the linker can allow the starting substructures to adopt the required conformation. The additional information regarding the desired linker chemistry may well not improve this, even when linker quality is substantially improved.

To investigate whether the improvement in recovery rate is due to the number of linkers generated, we generated 5000 examples for each pair of starting fragments and assessed in how many cases the true linker was recovered (Figure 3.3). Due to sampling constraints, it was not possible to include SyntaLinker in this analysis. The improvement in recovery rate of DeLinker-3D persisted even as substantially more linkers were generated. After several thousand examples, the rate of recovery of additional linkers decreased significantly for all methods, but remained the highest for DeLinker-3D. While increasing the number of samples further would likely yield more linkers being recovered, this effect is likely to be relatively small

Table 3.3: Scaffold elaboration. PDBbind set results (see Section 3.3.3 for definitions of the metrics).

Metric	REINVENT	DeLinker	DeLinker-Counts	DeLinker-3D
Valid	99.9%	99.9%	100.0%	99.4%
Unique	23.7%	86.7%	80.3%	74.8%
Novel	2.1%	70.2%	78.3%	77.2%
Recovered	0.0%	1.4%	4.4%	14.9%
Pass 2D filters	98.8%	56.1%	48.6%	52.2%
SC_{RDKit} Generated				
>0.6	4.5%	3.3%	5.5%	13.8%
>0.7	1.1%	1.0%	1.7%	5.2%
>0.8	0.3%	0.2%	0.5%	1.0%
>0.9	0.0%	0.0%	0.2%	0.1%

unless orders of magnitude more samples were generated. Figure 3.3 is strong validation that DeLinker-3D generates better linkers rather than simply producing similar molecules to DeLinker.

3.4.3 Scaffold Elaboration Experiments on Large Test Sets

Large-scale assessments on the CASF and PDBbind test sets demonstrate that DeLinker-3D can effectively perform scaffold elaboration, with similar trends as the linker design experiments (Tables 3.3, B.4 and Figure 3.4).

Almost all molecules generated by the DeLinker models are deemed valid since chemical valency is enforced during generation, with the very small number of invalid molecules due to no elaboration being proposed (Table 3.3). The majority of molecules generated were unique, with uniqueness decreasing from 87% for DeLinker to 73% for DeLinker-3D, as more constraints are provided. This was in line with expectations and mirrors the linker design experiments, with all methods proposing a high proportion of novel R-groups (35-43% on the CASF set, 70-78% on the PDBbind set).

In line with the performance for linker design, including 3D pharmacophoric information resulted in a substantially higher proportion of the true elaborations

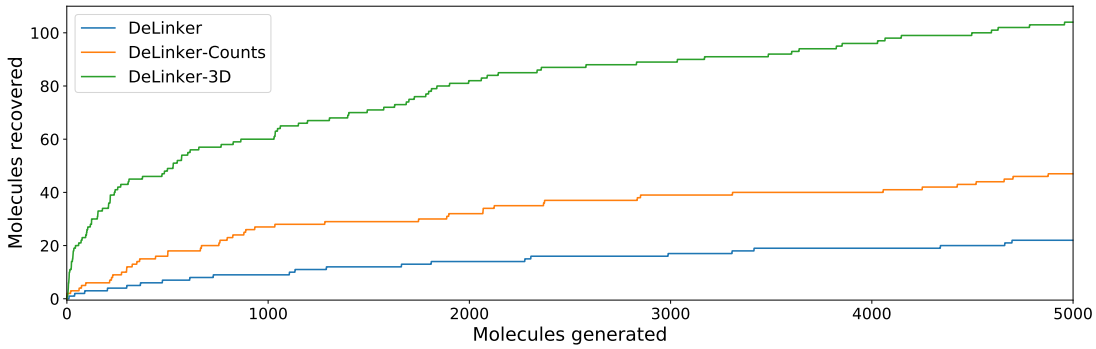


Figure 3.4: Scaffold elaboration. Number of original molecules recovered as the number of generated molecules is increased. DeLinker-3D recovered the significantly more of the original molecules than both baselines for any number of samples generated.

being recovered. On the CASF test set, DeLinker-3D recovered 69% of the ground truth molecules compared to 47% for DeLinker and 60% for DeLinker-Counts (Table B.4). On the PDBbind set, DeLinker-3D recovered 15% of the original elaborations, an increase of ten-fold compared to DeLinker (1.4%, Table 3.3). This performance persisted as more molecules were generated (Figure 3.4). When 5000 elaborations were generated for each scaffold, DeLinker-3D recovered 35% of the original molecules compared to 16% when the 3D information was removed (DeLinker-Counts) and only 7% when no pharmacophoric information was included (DeLinker).

Finally, there was a substantial improvement in the 3D similarity of the generated molecules to the original ones. On the PDBbind set, of the elaborations which passed the 2D filters, 14% of those generated by DeLinker-3D obtained an SC_{RDKit} Generated score of greater than 0.6 compared to 3% and 6% obtained by DeLinker and DeLinker-Counts, respectively. Moreover, DeLinker-3D was more often able to generate elaborations which were highly similar to the original, with 1.0% of elaborations obtaining a score of above 0.8, twice as many as DeLinker-Counts.

REINVENT produced significantly fewer novel elaborations than all of the DeLinker models, with only 2-3% of generated elaborations not contained in the training set (Table 3.3, B.4). As such, REINVENT did not recover any of the original elaborations in the PDBbind test set, while on the CASF test set REINVENT recovered only 25% of the original elaborations compared to 69% for

DeLinker-3D. In addition, DeLinker-3D generated a substantially higher proportion of elaborations that were highly similar to the original molecule than REINVENT, with 1.0% scoring above 0.8 compared to 0.3%. As measured by SC_{RDKit} Generated, REINVENT performed similarly to DeLinker and was outperformed by DeLinker-Counts, consistent with the level of structural information provided to the models.

3.4.4 R-Group Optimisation Case Study

Borkin et al. (2016) developed a thienopyrimidine class of compounds to block the protein–protein interaction between menin and mixed lineage leukemia (MLL) fusion proteins. This interaction plays an important role in acute leukemias with MLL translocations, making this an important drug target. The authors' previous work (Borkin et al., 2015) had led to the identification of a highly potent menin–MLL inhibitor (IC_{50}=31 nM, GI_{50}[1]=0.55 μM, PDB ID: 4X5Z) but required further improvement of cellular activity and drug-like properties to develop compounds with potential therapeutic value. This was achieved via structure-based optimisation of substituents introduced to the indole ring (Figure 3.5a).

Following optimisation of several positions, the most potent compound displayed almost a 7-fold improvement in affinity in MLL-AF9 cells (GI_{50}=83 nM, PDB ID: 5DB3, Figure 3.5b, right), while other highly potent compounds demonstrated favourable drug-like properties, such as significant improvements in selectivity, reduced lipophilicity, and bioavailability.

The most significant modification to the original compound was the optimisation of the hydrogen bond interactions with Glu363 and Glu366 on menin. The indole nitrogen in the original molecule was involved in a hydrogen bond with the side chain of Glu363 but was partially solvent exposed and was not forming interactions with Glu366. This led the authors to explore a variety of substituents containing hydrogen bond donors. Two potent substitutions were an acetamide group (Figure 3.5b, left) and 4-methylpyrazole (Figure 3.5b, right).

[1]Half maximal cell growth inhibition

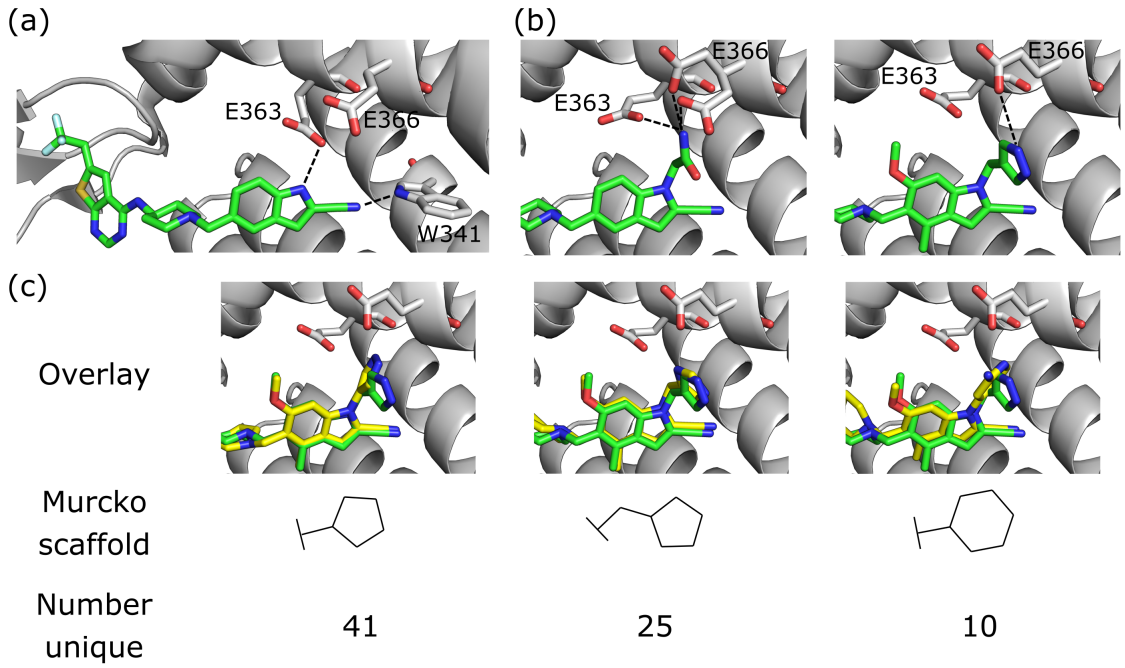


Figure 3.5: R-group optimisation case study. (a) Crystal structure (PDB ID 4X5Z) of the initial complex bound to menin. (b) Structure of two of the most potent optimised compound (PDB IDs left 5DB2, right 5DB3). The dashed lines represent key interactions. (c) Overlay of the most potent optimised compound (green carbons, PDB ID 5DB3) and several compounds generated by DeLinker-3D (yellow carbons) that make similar hydrogen bonding interactions.

We investigated the ability of DeLinker-3D to propose R-groups that met the design hypothesis described in Borkin et al. (2016). In particular, we sought to design both aromatic and non-aromatic hydrogen bond donor groups that were able to make similar interactions to the R-groups that were experimentally tested.

We derived 3D pharmacophoric profiles from the ligands in PDB IDs 5DB2 and 5DB3 to serve as input to DeLinker-3D. For the pharmacophoric profile derived from 5DB2, we generated 1000 R-groups with a maximum of four, five, and six atoms, whilst for the pharmacophoric profile derived from 5DB3 we generated 1000 molecules with a maximum of five, six, and seven atoms.

DeLinker-3D successfully recovered both of the experimentally-verified R-groups while generating many alternative molecules that could form similar interactions with menin. All methods were able to recover the acetamide R-group (Figure 3.5b, left). However, DeLinker-3D produced substantially more examples that matched the pharmacophoric profile (455) compared to both DeLinker-Counts (327) and

DeLinker (103). All methods were also able to recover the methylpyrazole R-group, although this was only generated once by DeLinker and DeLinker-Counts, compared to 61 times by DeLinker-3D. In addition, 237 of the elaborations generated by DeLinker-3D contained a aromatic system with a donor group linked to the indole via a methyl group compared to 50 for DeLinker and 11 for DeLinker-Counts.

We next sought to assess the alternatives to the methylpyrazole R-group (Figure 3.5b, right) that were proposed by DeLinker-3D. To validate the molecules proposed by DeLinker-3D, we docked the generated molecules containing an aromatic system and at least one donor group using GOLD (Verdonk et al., 2003) and checked whether the docked pose formed hydrogen bonding interactions with Glu363 or Glu366. Three elaborations, together with their Murcko scaffolds, are shown in Figure 3.5c (yellow carbons) overlayed with the methylpyrazole R-group (green carbons). All of the examples appear to fit within the pocket and were able to form similar interactions with Glu363 or Glu366, consistent with the stated design hypothesis.

3.5 Discussion

We have developed a method that combines GNNs with CNNs to incorporate 3D pharmacophoric constraints into molecular generation. Our approach allows prior knowledge to be used to control the design process and is readily extendable to alternate 3D structural representations.

We have demonstrated the applicability of our approach to both linker design and scaffold elaboration, two general tasks in the hit-to-lead and lead optimisation stages of drug discovery.

The experimental results show that our model significantly outperforms previous methods for these problems and demonstrates the power of including pharmacophoric constraints as a 3D representation as opposed to a 1D count vector.

While the quality of the generated compounds has increased significantly, the problem of selecting which ones should be explored further remains a key consideration. Successful application of generative models relies on their successful integration into the broader drug discovery toolbox.

While the focus of our work is generating molecules with specific 3D characteristics, we do not directly assign atomic coordinates during generation. The direct generation of 3D molecular structures is an exciting development (Gebauer et al., 2018; Gebauer et al., 2019), but has not yet been applied to drug-like molecules nor are existing methods directly applicable to the settings considered in this chapter. Extending our framework to generate atomic coordinates directly is an avenue for future work. Similarly, while we have shown encoding graph- and voxel-based representations separately is effective, unifying both with a single encoder that is 3D-aware could provide further benefit.

We believe that our method will allow greater synergy between human design hypotheses and machine learning-based molecular design.

In the following chapter, we turn our attention from generative modelling to predictive modelling, in particular structure-based virtual screening. We introduce our method, DenseFS, that combines advances in CNN methods for general computer vision tasks with techniques to incorporate domain-specific knowledge.

We are more alike, my friends,
than we are unalike.

— Maya Angelou, *Human Family*

4

Structure-Based Virtual Screening with Convolutional Neural Networks

Contents

This chapter is based on work described in the following publication:

Fergus Imrie, Anthony R. Bradley, Mihaela van der Schaar, and Charlotte M. Deane (2018). 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

An open-source reimplementation of the method described in this chapter is available at https://github.com/oxpig/DenseFS.

4.1 Preface

As described in Chapter 1, convolutional neural networks have been the primary driver of the enormous progress in computer vision over the last few years. This success has sparked the use of CNNs across a diverse range of application areas to great success, including several applications of CNNs to the *in silico* evaluation of protein-ligand complexes.

In this chapter, we have explored their use specifically in the context of structure-based virtual screening (SBVS), although the techniques employed are highly transferable to other problems in protein-ligand scoring. In doing so, we primarily sought to answer two questions. Firstly, are advances in CNN methods for general computer vision tasks applicable to SBVS and protein-ligand scoring more generally? Secondly, how can domain-specific knowledge be incorporated into this framework to improve such methods?

We have addressed the first of these questions by exploring the impact of adopting a more recent CNN architecture and also examining how combining multiple trained models in an ensemble effects the final predictions. To tackle the second question, we proposed two techniques to incorporate domain-specific knowledge. First, we demonstrated that the uncertainty and inaccuracy of docking necessitate the use of multi-pose scoring and have adopted an average scoring policy. Secondly, we utilised knowledge of the differences between protein families to propose a transfer learning approach to construct protein family-specific models.

An in-depth empirical study showed that our approach substantially outperformed recent benchmarks, with an ablation study demonstrating that the improvement of each of our proposed changes was somewhat orthogonal. In particular, our results suggest that the continued improvements in machine learning architectures for computer vision are applicable to SBVS.

4.2 Introduction

Drug discovery requires finding molecules that interact with targets with high affinity and specificity. While ultimately this is determined through experimental assays, computational techniques are frequently used to reduce the cost and improve the hit-rate of experimental verification. Successful applications of virtual screening in drug discovery processes are being increasingly reported (e.g. Kiss et al., 2008), however current methods still show relatively weak predictive power in many settings (Scior et al., n.d.; Li et al., 2014b; Wójcikowski et al., 2017).

As discussed in Chapter 1, traditional approaches have typically used experimental data to parametrise a physically inspired function (e.g. Böhm, 1994; Friesner et al., 2004). While interpretable, these techniques are inherently limited in their ability to capture complex interactions due to the use of rigid functional forms. Many machine learning-based scoring functions reuse the features of traditional approaches (e.g. Durrant and McCammon, 2011; Wójcikowski et al., 2017), but exploit the greater flexibility in model structure to produce better representations of the same input data (Li et al., 2015). However, this can lead to overfitting and often results in a loss of interpretability. In addition, the use of specific features, such as descriptors (Durrant and McCammon, 2011; Ballester and Mitchell, 2010) or fingerprints (Wu et al., 2018), both biases the model to the choice of features and leads to an unnecessary loss of information through the elimination or approximation of the raw structural data. For these reasons, following the work of Ragoza et al. (2017), we have adopted an approach that minimises initial featurisation of input data.

Due to the importance of spatial configurations for physical interactions, determination of binding can be reframed as a computer vision problem. Ragoza et al. (2017) showed that a fully convolutional neural network, taking as input only spatial and atom type information, can outperform empirical and feature-based machine learning approaches at virtual screening, while Jiménez et al. (2018) exhibited state-of-the-art performance at binding affinity prediction using a similar approach.

Both of these methods were based largely on early CNN models used in the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC, or ImageNet,

Russakovsky et al., 2015). Since its introduction in 2010, this competition has been the source of many of the substantial advances not just in computer vision, but in machine learning more broadly, several of which are described in Chapter 1. All entries in the first two years of the challenge had error rates of over 25% at the image classification task, while the winner of the 2017 edition, SENets (Hu et al., 2017), achieved a 2.3% error rate. These advances have been successfully applied to other areas outside of traditional computer vision tasks, such as medical imaging (Drozdzal et al., 2018; Li et al., 2017; Yu et al., 2017). However, limited use has been made in cheminformatics (Goh et al., 2017) and none that we are aware of for the study of protein-ligand interactions. We examine the applicability of modern CNNs to SBVS by utilising a densely connected convolutional neural network architecture (DenseNet, Huang et al., 2016).

A major challenge in virtual screening is the heterogeneity of binding between different targets arising from the structural diversity of proteins. Interactions of one target are not necessarily indicative of interactions of another target. However, proteins can be grouped into families, with proteins belonging to the same family having similar structures and physicochemical properties. As a result, it has been shown that in most cases a targeted scoring function will outperform a universal model (Ross et al., 2013; Wang et al., 2015b). We investigate how we can improve predictions for a target using information from its protein family but not the target itself. This mimics the investigation of a novel target or one for which data is difficult to obtain.

Transfer learning is a general class of machine learning methods which improve performance on a new task by exploiting information that has already been learnt on a different, but often related, task. Finetuning is a transfer learning technique that allows a general model, trained on one data set, to be re-purposed for a new task by re-training a part (or all) of the model on a data set specific to the new task. For example, Tajbakhsh et al. (2017) showed that finetuning a model for medical imaging that was originally trained on general image data outperformed training solely on the medical data, in particular when limited examples were available. We

use this technique to create protein family-specific models, and compare these to see if such a targeted scoring function outperforms a universal model.

In this chapter, we first show how recent advances in computer vision can be applied to SBVS by adopting a CNN model based on the DenseNet architecture (Huang et al., 2016). We then investigate the optimal number of poses to use as input. Finally, we detail how transfer learning can be used to construct models for specific protein families. We conduct an in-depth empirical study into the number of family members required to adopt family-specific models, and for the first time present guidelines for the expected benefit from collecting additional data.

4.3 Methods

To develop our virtual screening tool, we utilised a clustered cross-validation of our training set (DUD-E) to optimise our network and investigate other choices in the procedure. We then trained a final model on the full training set, and evaluated our methods on two independent test sets (a subset of ChEMBL and MUV). These sets have been filtered to ensure that targets are sufficiently dissimilar from targets in the training set (see below).

4.3.1 Input Format

Ligand poses for all actives and decoys were generated using AutoDock Vina (Trott and Olson, 2010), specifically the smina (Koes et al., 2013) implementation. Ligands were docked against a reference receptor within a box centered around a reference ligand with 8 Å of padding using smina's default arguments for exhaustiveness and sampling [1]. We followed the approach described in Ragoza et al. (2017) and discretised the docked protein-ligand structures into a grid format to act as the input for the CNN. A schematic of the input featurisation process is shown in Figure 4.1. The grid used was a 24 $Å^3$ cube, with a resolution of 0.5 Å, centered around the binding site. Each point of the 3D grid has 34 information channels,

[1] We thank the authors of Ragoza et al. (2017), in particular David Ryan Koes, for providing docked poses for the datasets employed in this chapter.

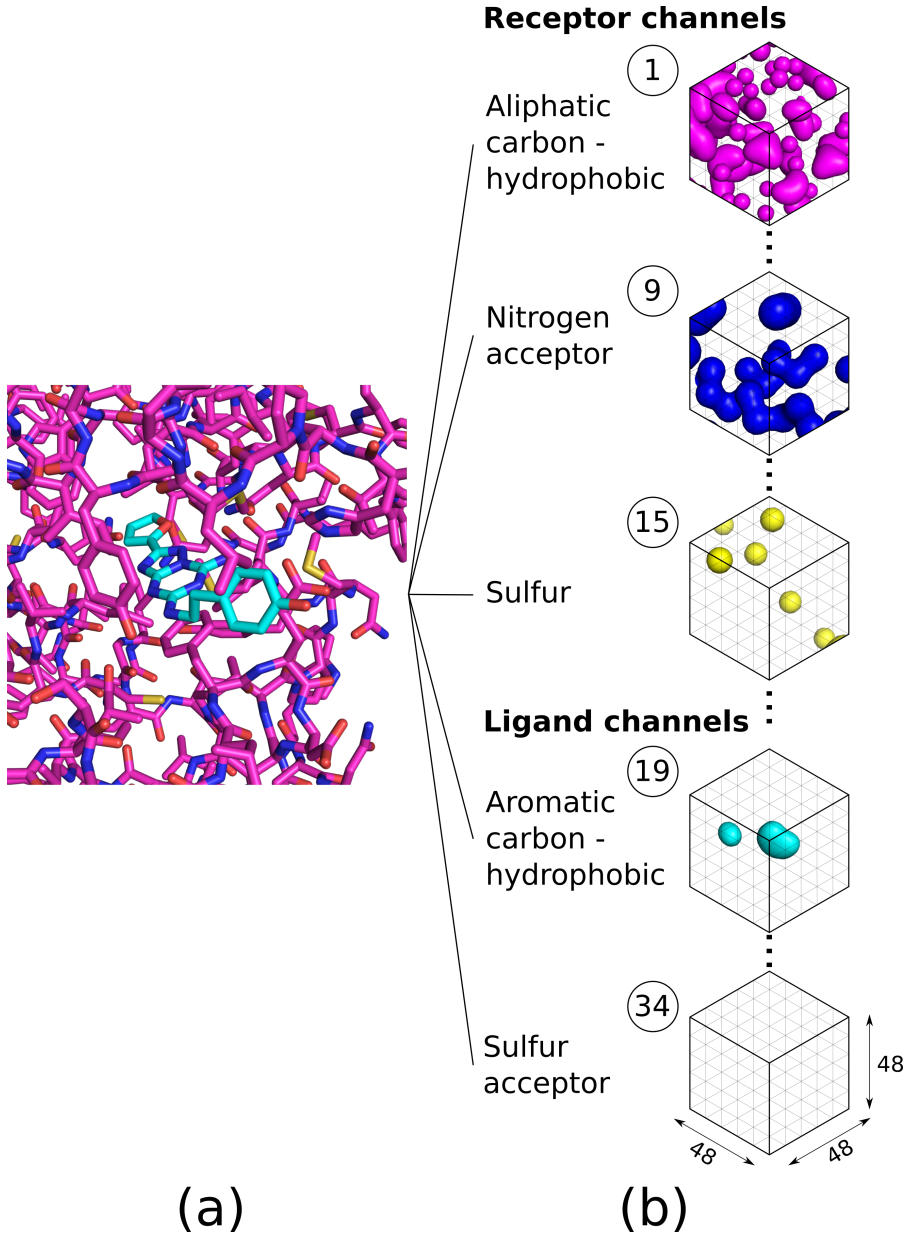


Figure 4.1: Input featurisation for PDB ID 3EML (ligand ZMA). (a) The protein-ligand complex is cropped to a 24 Å^3 box, centered on the ligand. The ligand is shown with carbons in cyan, the receptor with carbons in magenta, and the heteroatoms are coloured with standard colouring. (b) The complex is decomposed into information channels, one for each atom type (Table 4.1), and divided into voxels with a resolution of 0.5 Å. Atoms within each channel have a Gaussian representation. We show this for five of the 34 channels, and visualise the final voxel grid as an isosurface. The resulting atom type channels are concatenated to produce the (34, 48, 48, 48) input tensor for the CNNs.

Table 4.1: Atom typing scheme. The input format included 34 information channels, one for each atom type, 18 corresponding to ligand atoms and 16 to receptor atoms.

Receptor atom types	Ligand atom types
AliphaticCarbonXSHydrophobe	AliphaticCarbonXSHydrophobe
AliphaticCarbonXSNonHydrophobe	AliphaticCarbonXSNonHydrophobe
AromaticCarbonXSHydrophobe	AromaticCarbonXSHydrophobe
AromaticCarbonXSNonHydrophobe	AromaticCarbonXSNonHydrophobe
-	Bromine
Calcium	-
-	Chlorine
-	Fluorine
-	Iodine
Iron	-
Magnesium	-
Nitrogen	Nitrogen
NitrogenXSAcceptor	NitrogenXSAcceptor
NitrogenXSDonor	NitrogenXSDonor
NitrogenXSDonorAcceptor	NitrogenXSDonorAcceptor
-	Oxygen
OxygenXSAcceptor	OxygenXSAcceptor
OxygenXSDonorAcceptor	OxygenXSDonorAcceptor
Phosphorus	Phosphorus
Sulfur	Sulfur
-	SulfurAcceptor
Zinc	-

corresponding to distinct heavy atoms on either the protein (16 channels) or ligand (18). A list of permitted atom types, based on smina atom types, is provided in Table 4.1 and is consistent with the work of Ragoza et al. (2017). Atoms have a Gaussian representation within the van der Waals radius that is quadratically smoothed to zero at 1.5x the van der Waals radius from the input coordinates of a given atom. The input format provides no additional information beyond spatial coordinates and atom type, and does not explicitly include bond order or hydrogens; the information provided in the input format is a comprehensive representation of the binding site of a single, static, docked protein-ligand complex, up to the chosen grid resolution and atom typing scheme.

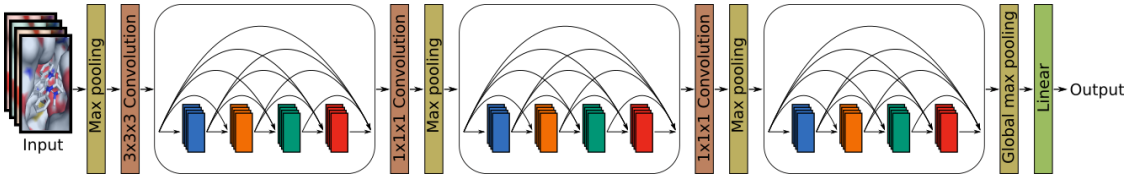


Figure 4.2: Schematic of the DenseNet architecture used in our model.

4.3.2 Model Description

Model architecture.

We based our model on the DenseNet architecture, introduced by Huang et al. (2016). DenseNets have achieved state-of-art performance on several computer vision tasks, while exhibiting substantially improved parameter efficiency (Huang et al., 2016). The key architectural difference in a DenseNet compared with other convolutional networks is the fashion in which layers are connected (Figure 1.5). Instead of each layer receiving as input solely the output of the previous layer (standard connectivity), within each dense block a layer receives the output of all prior layers within that block (dense connectivity). The ability for the output of a layer to skip the next allows feature maps of different depths and hence complexities to form, and for new feature maps to be learnt from a combination of existing maps of differing complexities. Furthermore, the dense connections improve gradient flow during backpropagation (Rumelhart et al., 1986), allowing deeper models to be trained effectively (Huang et al., 2016).

Our model contained three dense blocks, with four convolutional blocks within each (Figure 4.2). The convolutional blocks consisted of a batch normalisation layer, a convolutional layer with a $3 \times 3 \times 3$ kernel, followed by a rectified linear unit. Between each dense block, we included a $1 \times 1 \times 1$ convolutional layer followed by a $2 \times 2 \times 2$ max pooling layer. The first convolutional layer contained 32 filters, after which each $3 \times 3 \times 3$ convolutional layer contained 16 filters, while the $1 \times 1 \times 1$ convolutional layers matched the number of input features. A schematic of our network architecture can be seen in Figure 4.2.

Training

Our CNN models were defined and trained using the Caffe deep learning framework (Jia et al., 2014), with the MolGridDataLayer input format from the gnina fork (Ragoza et al., 2017). [2] All networks were trained using stochastic gradient descent (SGD) to minimise the multinomial logistic loss. We trained our model with a batch size of 16 for 25,000 iterations (corresponding to between 12 and 14 epochs of the active molecules during cross-validation). Due to the substantial class imbalance, we used oversampling and forced each batch to contain an equal number of positive and negative examples (Buda et al., 2017). Following Ragoza et al. (2017), we used a learning rate of 0.01, momentum of 0.9, an inverse learning rate decay with power = 1 and gamma = 0.001, and weight decay of 0.001. Input structures were augmented via random translations of up to 2 Å and random rotations.

Test time scoring

We trained our CNN models only on the top-ranked AutoDock Vina pose for each complex. Many of these poses will be incorrect and restricting evaluation in the same way at test time would result in scoring inaccurate poses for many of the active molecules (Figure C.1). At test time, any scoring system should ideally display a robustness to the poses selected, i.e. a similar score should be obtained if the poses assessed are similar. In particular, compounds, especially decoys, should not be ranked highly based on a single pose. Ragoza et al. (2017) demonstrated improvement over scoring only the top-ranked Vina pose by scoring all poses and using the maximum as the final score. We investigated averaging the scores of the top n ranked poses, where we determined n through cross-validation, and compared this to using the maximum.

Protein family-specific models

It has been shown that in most cases a scoring function constructed for a specific protein family will outperform a universal model (Ross et al., 2013; Wang et al.,

[2] We have since released an open-source reimplementation using the libmolgrid package (Sunseri et al., 2019) which is available at https://github.com/oxpig/DenseFS.

2015b). As such, we constructed protein family-specific models using transfer learning by finetuning our universal model on data from targets belonging to the target's family.

Finetuning describes the technique where a model is pre-trained on an initial, usually larger, training set, before the model is trained further on a second, targeted, set. This allows general features to be formed in the initial training stage, before the parameters become more specific during the second training stage. Deep learning models are prone to overfitting without sufficient training data. In many settings, data for a similar or related task is readily available but numerous high quality examples for the specific task are not. Thus, if training data is limited to only closely related targets, the examples would likely be more informative, but the risk of overfitting would be higher. In the other extreme, if all available data is included in the training set, then related, but not overly informative examples (such as those from a structurally dissimilar target), would make up the majority of the training set and thus non-specific representations would be learnt. Finetuning can be used to overcome these challenges.

Selecting the training set for finetuning is a non-trivial problem, especially given that we do not permit any overlap of targets between training and test sets, restricting training to off-target data. Proteins can be clustered into families where members typically have similar 3D structures and function. There are several widely used methods available to categorise proteins using combinations of sequence, structural, and functional information, such as SCOP (Andreeva et al., 2014) and CATH (Dawson et al., 2017). We have used the target protein classes provided in the works of Mysinger et al. (2012), Riniker and Landrum (2013), and Rohrer and Baumann (2009) to cluster proteins, and refer to these as the protein family. For all three data sets, these are consistent with the second level of ChEMBL's protein target classification. After pre-training on the entire training set, we investigated finetuning models for specific protein families using training sets constructed from targets belonging to the same family only. We examined the effect this had for different families and a varying number of family members.

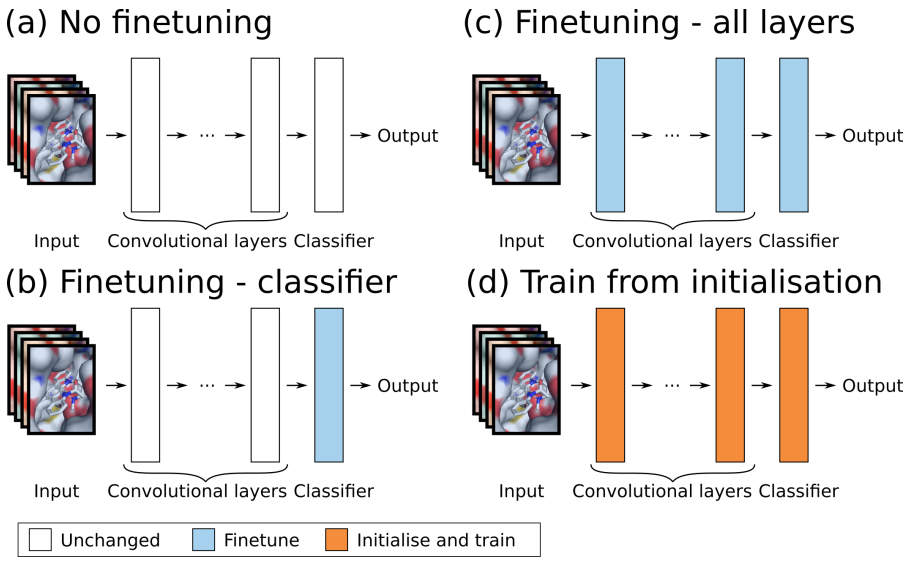


Figure 4.3: (a) - (d) Illustration of the different training regimes adopted to construct family-specific models. White corresponds to layers of the model that have been trained on all training data; blue to layers that have been trained first on all training data, and then finetuned on data from a specific protein family; orange to layers that have been trained only on data from a specific protein family.

We investigated the two extremes of finetuning by training only the classifier (final layer) and freezing the parameters in all other layers, or letting the parameters of all layers of the model train (Figure 4.3). We finetuned for five epochs of the active molecules. We reduced the learning rate by a factor of 10 to 0.001 when finetuning only the classifier, and a factor of 20 to 0.0005 when finetuning all layers of the model. As is custom, we adopted a variable learning rate when finetuning all layers of the model, with the classifier's learning rate 10x higher than the convolutional layers. We compared these approaches to not finetuning the trained model, as well as training a new model from initialisation (i.e. from scratch) only on data from a given protein family (Figure 4.3).

4.3.3 Model Evaluation

The choice of evaluation method should closely reflect the practical situation and desired outcome. This is discussed in more detail in Chapter 1. To evaluate our SBVS method, we adopted the following approach, in line with the work of Ragoza et al. (2017). We first assessed the performance of our models by a 3-fold cross-

validation on DUD-E (Mysinger et al., 2012), and used the results to optimise our protocol. During cross-validation, we clustered proteins by sequence similarity using CD-HIT (Huang et al., 2010) and ensured that targets with >80% sequence similarity were included in the same fold to avoid training and testing on overly similar targets. Wu et al. (2018) demonstrated almost perfect performance on DUD-E when performing a random training/test split across all targets, allowing training and test sets to contain examples from the same targets. Wójcikowski et al. (2017) saw performance in cross-validation almost triple when splitting their data either randomly across all targets ("horizontal split") or on a per-target basis, as opposed to keeping all examples for a given target in the same fold ("vertical split"). We did not permit any overlap of targets between training and validation sets, meaning that all testing is performed on unseen targets. The clustered approach we adopted represents a further refinement of the vertical split, and overall our cross-validation procedure more closely mimics the screening of a novel target.

The final protocol was then evaluated on two independently constructed test sets, a subset of the ChEMBL database (Bento et al., 2014) curated by Riniker and Landrum (Riniker and Landrum, 2013), following Heikamp and Bajorath (Heikamp and Bajorath, 2011), and the maximum unbiased validation (MUV, Rohrer and Baumann, 2009) data set, which is based on PubChem bioactivity data. Following the work of Ragoza et al. (2017), the independent sets have been further refined to avoid artificially enhancing performance using the following steps: (i) a global sequence alignment between all targets from the training and proposed test sets was performed, removing any test target with more than 80% sequence similarity with a training target, (ii) ProBiS (Konc and Janežič, 2010) structural alignment on the binding sites of all pairs of targets from the training and proposed test sets was performed, removing any targets for which a significant alignment was found using the default ProBiS parameters. This resulted in a 13 target subset of the Riniker and Landrum ChEMBL set (from 50 initially), and a 9 target subset of the MUV set (from 17). Further details on all the datasets employed in this chapter can be found in Chapter 1.

The models were evaluated with respect to two global metrics and four local ones. The global metrics used were area under curve (AUC) of the receiver operating characteristic (ROC) curve (Hanley and McNeil, 1982) and the precision recall curve (PRC). As noted by previous works (Davis and Goadrich, 2006; Wu et al., 2018), while AUC ROC and AUC PRC are highly correlated, substantial performance differences can emerge in the case of high class imbalance. We argue similarly to the work of Wu et al. (2018) that AUC PRC is a more informative metric given the extent of class imbalance inherent in virtual screening. Random performance has an expected AUC ROC of 0.5, whereas the expected AUC PRC of a random classifier is equal to the class imbalance (e.g. if there were 50 decoys for each active, the expected AUC PRC for a random classifier would be 0.02). We present AUC ROC due to its wide use, interpretability, and to allow direct comparison to other work. To measure early enrichment, we reported the ROC enrichment (Jain and Nicholls, 2008; Nicholls, 2008). ROC enrichment measures the ratio of true positive rate (TPR) to false positive rate (FPR) at a given FPR. We assessed the enrichment factor (EF) at 0.5%, 1%, 2%, and 5%. The maximum possible ROC enrichment depends on the FPR threshold (e.g. at an FPR of 2%, the highest possible ROC enrichment is 50), while random performance has an expected enrichment factor of 1 at any FPR threshold.

4.3.4 Visualisation

We took a machine learning approach to visualisation and replaced the final layer of the model (the classifier) with a $1 \times 1 \times 1$ convolutional layer with the same parameters as the classifier. This transformed the final layer from a global classifier into a regional classifier and allowed specific regions of the image to be assessed. A score is produced for each sub-section, rather than a single score for the entire complex. A limitation of this technique is that our models have not been trained to assess subregions of the complex, but rather the complex as a whole. However, this technique provides a region-based assessment without modifying the protein-ligand complex at marginal additional computational cost.

4.3.5 Comparison to Previous Work

In structure-based virtual screening, traditional approaches have typically used experimental data to parametrise a physically inspired scoring function (e.g. Böhm, 1994; Friesner et al., 2004). Machine learning approaches have either reused the features of traditional approaches, or calculated new descriptors from docked protein-ligand complexes (Durrant and McCammon, 2011; Ballester and Mitchell, 2010; Wójcikowski et al., 2017). Recently, several attempts to learn features relevant for binding in an end-to-end manner, rather than manually selecting them, have shown promise. Ragoza et al. (2017) and Wallach et al. (2015) both used shallow CNN architectures for virtual screening. CNNs have also been used for binding affinity prediction (Jiménez et al., 2018; Gomes et al., 2017).

Specifically, Ragoza et al. (2017) proposed a neural network for protein-ligand scoring consisting of three $3 \times 3 \times 3$ convolutional layers (with 32, 64, and 128 filters respectively), each preceded by a $2 \times 2 \times 2$ max pooling layer. They scored all docked poses using a single, universal model, and took the maximum as the final score. To our knowledge, their approach is the best performing CNN model, and has been shown to outperform both empirical scoring functions and feature-based machine learning approaches at virtual screening (Ragoza et al., 2017). We have therefore used this approach as the benchmark. Our method utilises the same input format, but differs in four key ways from all previous approaches. Firstly, we adopted the densely-connected neural network architecture described above. Secondly, at test time our model scored each protein-ligand complex by averaging over an ensemble of docked poses. Thirdly, we used transfer learning to construct protein family-specific models for each of the four major protein classes in DUD-E. Finally, we employed an ensemble of models. The effect of each of these changes is detailed in a full ablation study (Table 4.3) and discussed below.

4.4 Results

4.4.1 Cross-Validation on DUD-E

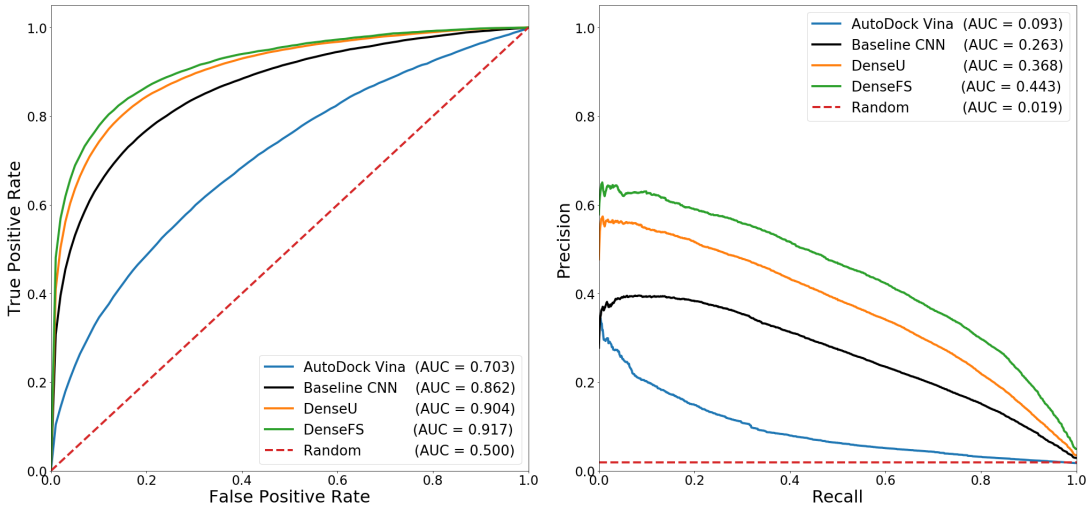
We assessed performance using a clustered cross-validation on DUD-E. We present a final version of our method that incorporates the DenseNet architecture, average test time scoring, and combines an ensemble of protein family-specific models ("DenseFS"). Our approach achieved state-of-the-art performance on the DUD-E benchmark, recording average per-target AUC ROC of 0.917, AUC PRC of 0.443, and 0.5% ROC enrichment factor of 79.3 (Table 4.2).

We compared DenseFS to the CNN model described by Ragoza et al. (2017) ("Baseline CNN"). We also compared to the AutoDock Vina scoring function. Summary results from cross-validation on DUD-E are shown in Figures 4.4 and 4.5, and Table 4.2. For each of the CNN methods, we carried out three replicas with different random seeds. DenseFS exhibited around a 70% improvement in AUC PRC and 0.5% ROC enrichment over the Baseline CNN, with around a 400% gain over AutoDock Vina. Compared with the Baseline CNN, our method achieved a higher AUC ROC for 97 of the 102 targets (95%), and AUC PRC for 95 of the 102 targets (93%).

For both the Baseline CNN and DenseFS, substantial performance differences emerged between targets, in particular between different protein families (Figure 4.5, Tables C.1 - C.5). DenseFS exhibited the most predictive power for kinases followed by proteases, while the largest improvement in performance compared to the Baseline CNN was on the nuclear proteins family. Performance on GPCRs and the remaining targets (denoted as Other in Figure 4.5) was lower compared to the families for which the data set contained more targets, albeit still much better than random. The ordering of performance by family precisely matched the overall number of targets in DUD-E, highlighting again the importance of the presence of data from the same protein family in the training set. However, while beneficial, this is not a requirement. DenseFS achieved an AUC ROC of 0.943 on DUD-E target DYR, an oxidoreductase, which shared no family members with the training

Table 4.2: Mean AUC ROC, AUC PRC and ROC enrichment across targets in the DUD-E data set for our method, DenseFS, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseFS
AUC ROC	0.703	0.862	0.917
AUC PRC	0.093	0.263	0.443
0.5% EF	15.017	44.521	79.321
1% EF	10.383	30.652	47.986
2% EF	7.135	19.724	28.408
5% EF	4.726	10.595	13.744



(a) **(b)**

Figure 4.4: Average per-target ROC (a) and PRC (b) plots comparing our methods (DenseFS and DenseU) with the Baseline CNN and the AutoDock Vina scoring function on the DUD-E data set during cross-validation. Our methods outperformed the Baseline CNN by 6.4% and 4.9% with respect to AUC ROC, and 68.4% and 39.9% with respect to AUC PRC.

set. More generally, DenseFS exhibited substantial predictive power for the "other" category, achieving an average AUC ROC of 0.865 (Table S5).

We performed an ablation study on the four key changes made between the Baseline CNN and DenseFS (Table 4.3). All four changes had a material positive impact both on a standalone basis, but also when added to any pre-existing combination of changes. This suggests that each factor works somewhat independently. For the remainder of this section, we discuss the individual effect and implications

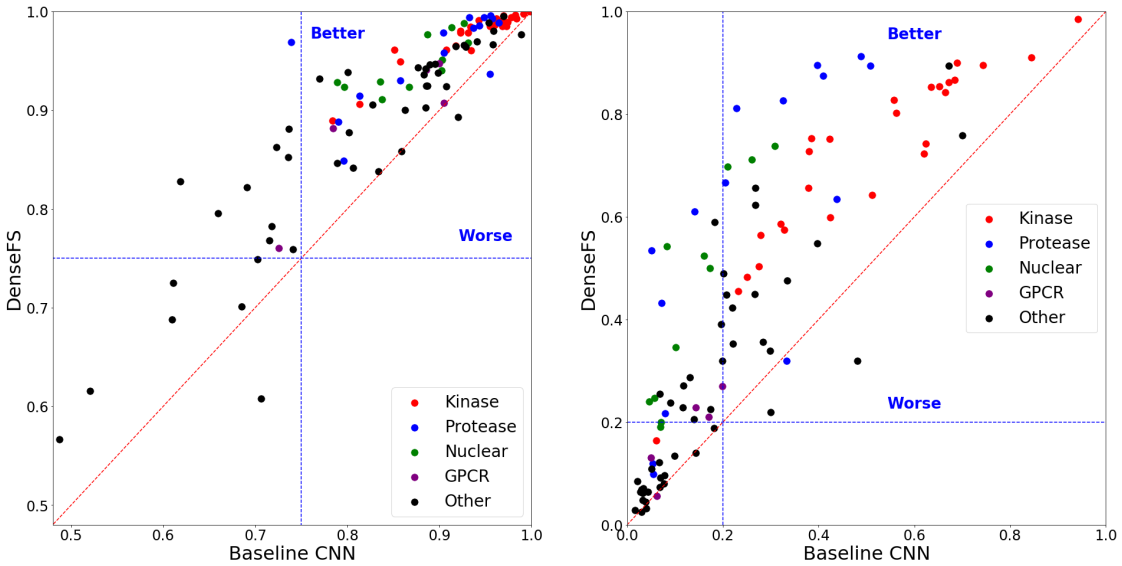


Figure 4.5: Performance of DenseFS compared to the Baseline CNN during cross-validation on the DUD-E data set. We directly compare performance on each target in DUD-E with respect to AUC ROC (left) and AUC PRC (right). Points above the diagonal represent targets for which DenseFS outperforms the Baseline CNN (and vice versa). Guidelines are set at 0.75 for AUC ROC, and 0.2 for AUC PRC. This corresponds to approximately a 10x improvement over random for AUC PRC. DenseFS achieved a higher AUC ROC for 97 of the 102 targets (95%), and AUC PRC for 95 of the 102 targets (93%).

of each of these advantages.

Advantage 1. Model architecture

We found substantial benefit from changing the model topology to a DenseNet architecture and adopting a deeper model, shown in Table 4.3. Changing the model architecture alone was responsible for a 25% increase in AUC PRC compared to the Baseline CNN (c. 37% of the overall improvement). This shows the suitability of reformulating SBVS as a computer vision problem, and highlights how advances in computer vision tasks provide improvements in this setting.

Our final model contained more convolutional layers and more overall features than the Baseline CNN. That a more expressive model led to improved performance adds further evidence to the understanding that binding is governed by complex relationships and the factors that determine activity are not readily summarised. Our findings that a deeper model improved performance is somewhat contrary to

the optimization analysis performed by Ragoza et al. (2017), where cross-validation suggested a shallower and narrower model led to an improved representation of the data. However, in this case the network architecture was optimised on a different, much smaller, data set and a different task (pose prediction). Given the similarities with virtual screening, we expect that if larger data sets were utilised then deeper models would yield benefit for other problems in protein-ligand scoring (pose prediction and binding affinity prediction) [3].

Hyperparameter tuning is necessary to extract maximum performance from a model, and suboptimal choices can lead to substantially weaker results (Breuel, 2015). Despite adopting largely the same hyperparameter settings determined by Ragoza et al. (2017) for a different network, our ablation study showed that changing the network architecture from a traditional CNN to a DenseNet resulted in a 20.8% - 36.5% increase in AUC PRC.

Advantage 2. Test time scoring

We see in Figure 4.6 that averaging the top n ranked poses outperformed scoring only the top ranked Vina pose or the highest scored pose for almost all values of n, with the best performance exhibited by taking $n = 9$. AutoDock Vina generated 15 poses on average for each of the active molecules in DUD-E; averaging the top 9 highest scored poses corresponds to averaging the top 60% of poses. This intuitively agrees with our motivation of averaging over a group of poses that are more likely to be close to a native pose. Averaging across almost all of the complexes would be likely to include many inaccurate or unrealistic poses, whereas only averaging across a small number of poses would still be susceptible to an outlier, either with high or low score. We found that the optimum choice of n was consistent across the CNN approaches tested. Overall, adopting average scoring instead of a max scoring policy led to improvements in AUC PRC of between 7.6% and 14.4% (Table 4.3).

[3]Indeed, Francoeur et al. (2020) show that a variant of our DenseNet provides performance improvements in binding affinity and pose prediction compared to their previous CNN architectures when trained on a larger dataset.

Table 4.3: Ablation study of the primary improvements in our final protocol (DenseFS) compared to the Baseline CNN. Each of the four main changes had a material positive impact in any possible protocol, with the change of model architecture and use of family-specific models the most important.

Model Architecture	Average Scoring	Protein Family Models	Ensemble	AUC PRC	Gain (%)	0.5% EF	Gain (%)
✓	✓	✓	✓	**0.443**	**68.4%**	**79.321**	**78.2%**
✓	✓	✓		0.421	60.1%	75.351	69.2%
✓		✓	✓	0.407	54.8%	72.995	64.0%
✓	✓		✓	0.394	49.8%	69.647	56.4%
✓		✓		0.384	46.0%	67.903	52.5%
✓	✓			0.368	39.9%	64.888	45.7%
✓			✓	0.359	36.5%	62.713	40.9%
	✓	✓	✓	0.357	35.7%	64.040	43.8%
	✓	✓		0.348	32.3%	62.010	39.3%
		✓	✓	0.336	27.8%	59.262	33.1%
✓				0.330	25.5%	56.819	27.6%
		✓		0.326	24.0%	56.771	27.5%
	✓		✓	0.298	13.3%	52.289	17.4%
	✓			0.285	8.4%	49.484	11.1%
			✓	0.278	5.7%	48.481	8.9%
				0.263	**0.0%**	**44.521**	**0.0%**

Combining these two modifications produces a universal scoring function consisting of a single model. We compared this approach, denoted as DenseU, with the Baseline CNN, and found an improvement in average AUC PRC of 39.9% (Figure 4.4), with DenseU achieving a higher AUC PRC for 90 of the 102 targets (88%).

Advantage 3. Protein family-specific models

The benefits of finetuning a separate model for each of the four major protein families represented in DUD-E can be seen in Table 4.3 and Figures 4.7, C.2, and C.3. In our ablation study, adopting protein family-specific models instead of a universal model led to average improvements in AUC PRC of 18.3% - 24.0% (Table 4.3).

We subdivided DUD-E into the following five groups by the classifications provided: kinase (26 targets), protease (15), nuclear (11), GPCR (5), and other (45). We found finetuning improved performance for all four families, but due to the different amounts of data available, different training regimes were optimal. Allowing all parameters to vary (Figure 4.3c) for kinases, proteases and nuclear

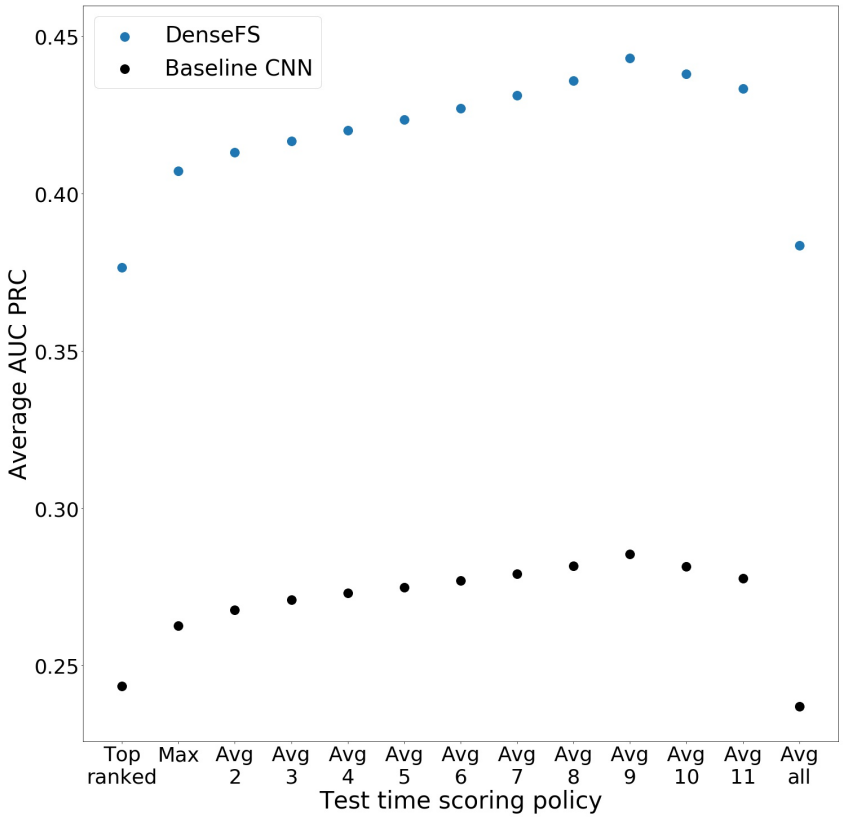


Figure 4.6: Average AUC PRC across targets in the DUD-E data set for different test time scoring policies. For both the Baseline CNN and DenseFS, the optimum n to average across was nine.

proteins led to better results over retraining solely the final layer or no finetuning at all, whereas for GPCRs we found that freezing the convolutional filters that were trained on a mix of protein family data and only finetuning the final layer on GPCR data was optimal (Figure 4.3b).

We believe the different behaviour is due to the small number and diversity of GPCR targets. If additional GPCR data were available, we anticipate that allowing more parameters to vary would outperform finetuning only the classifier. However, given we witnessed improvement from finetuning all layers with very limited data for the other families, it is also possible that the diversity of the GPCR targets means that it is more important to retain representations learnt from other protein families.

In order to gain a better understanding of the impact of protein family data on the construction of CNN models, we examined the effect of artificially reducing the amount of family-specific data used both in initial training and subsequent finetuning

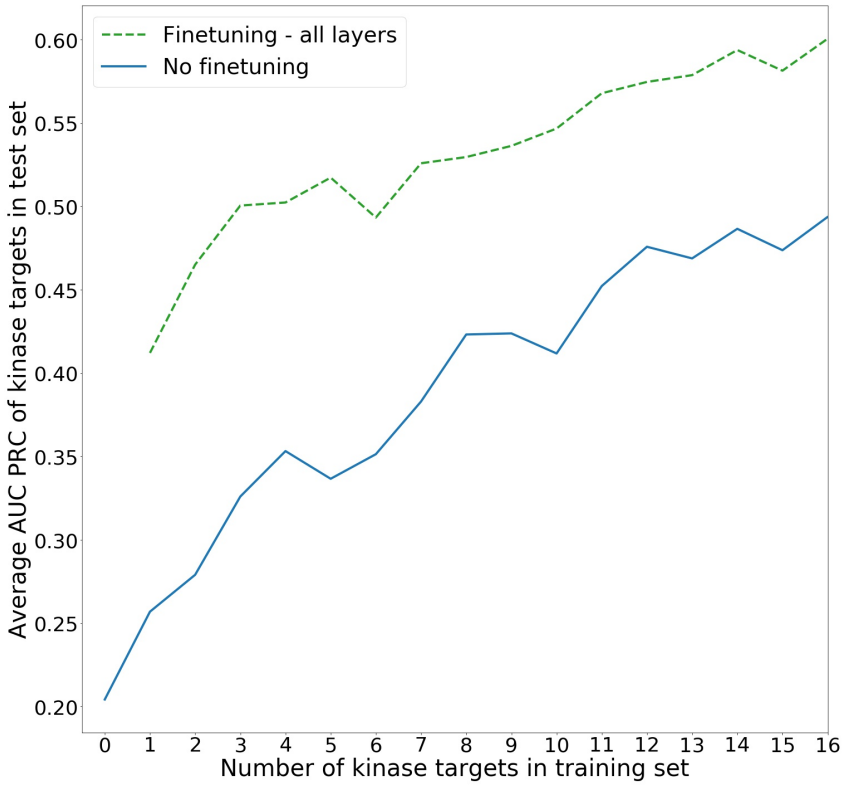


Figure 4.7: Average AUC PRC of kinase targets for varying number of kinases in the training set. We compared finetuning all layers to no finetuning (training protocols displayed in Figure 4.3). The blue line (solid) represents no finetuning, while the green line (dashed) shows the effect of finetuning all layers on only kinase data. Finetuning all layers outperformed no finetuning, even with limited data, and performance continued to improve as more targets were added to the training data.

(Figures 4.7, C.2, C.3). This allowed us to assess whether additional data had a positive impact on performance, or if further examples beyond a point were largely redundant. We trained copies of the Baseline CNN with varying numbers of targets from the protein family included in the training set. We then finetuned these models as previously described in Figure 4.3, and compared with training from initialisation.

Firstly, we found that finetuning improved performance even with very limited number of family members for all families (Figures 4.7, C.2, and C.3). Considering the largest family, this effect persisted regardless of the number of kinases included in the training set, with a 20-50% increase in performance as a result of finetuning all layers of our model, depending on the number of targets included (Figure 4.7). Secondly, ensuring family members are present in the training set is crucially important. The average AUC PRC across kinase targets more than doubled when

all available family data was present compared to including no kinase data in the training set. To confirm that this effect was not simply due to a smaller training set, we removed targets at random and did not witness the same reduction in predictive power compared to removing the same number of kinases. Finally, we saw that performance continued to improve as more targets were included in the training set, and the rate of improvement remained fairly constant as targets were added. This suggests that adding more kinases to the training set would further improve performance and that even with 16 kinase targets in the training set we have not reached learning saturation.

Advantage 4. Ensemble

Ensemble methods exploit the predictions of multiple models to improve performance. We combined the predictions of the three replicas by averaging the scores produced by each of the models. This provided a small, but appreciable, improvement, with increases in average AUC PRC of 3.4% - 11.0%.

4.4.2 Independent Test Sets

We assessed performance on the independent test sets by training models using the entirety of the DUD-E data set. We present the same version of our method, DenseFS, and again compared to the Baseline CNN and the AutoDock Vina scoring function.

ChEMBL

Consistent with cross-validation results, DenseFS substantially outperformed the Baseline CNN and AutoDock Vina (Table 4.4). DenseFS achieved an average AUC PRC of 0.214 and 0.5% ROC enrichment factor of 47.6, representing improvements of 40.8% and 33.6%, respectively, over the Baseline CNN.

As anticipated, the ChEMBL test set was more challenging for all methods by all metrics. However, the models retained substantial predictive power, and the results show an ability to generalise to targets and data sources beyond the

Table 4.4: Mean AUC ROC, AUC PRC and ROC enrichment across targets in the ChEMBL test set for our method, DenseFS, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseFS
AUC ROC	0.656	0.788	0.838
AUC PRC	0.039	0.152	0.214
0.5% EF	9.293	35.626	47.587
1% EF	7.505	23.244	30.732
2% EF	5.324	14.933	18.940
5% EF	4.288	8.417	9.979

training set, suggesting that the improvements seen in cross-validation were not simply a result of overfitting to DUD-E.

On a per-target basis, DenseFS outperformed the Baseline CNN by the largest margin on targets belonging to one of the four specific protein families (kinase, protease, nuclear, GPCR), in particular the three protease targets. Our method demonstrated more modest improvements in performance for targets in the "other" category. Overall, DenseFS achieved a higher AUC ROC for 11 of the 14 targets (79%), and AUC PRC for 13 of the 14 targets (93%), compared to the Baseline CNN.

MUV

The summary results in Table 4.5 show that none of the methods tested had any meaningful predictive power on MUV. The only target for which any of the methods demonstrated an ability to discriminate actives from decoys was human cAMP-dependent protein kinase (MUV target ID 548, PDB ID 3poo). All three CNN methods achieved an AUC ROC of around 0.8 and AUC PRC of c. 0.01 (5x greater than random).

The MUV data set is constructed differently from the other data sets used, using PubChem data rather than ChEMBL. In addition, cell-based assays are used for some of the targets. These factors undoubtedly make the data set a tougher challenge, and some have even questioned the appropriateness of using MUV as an SBVS benchmark (Ragoza et al., 2017; Tiikkainen et al., 2009). The only

Table 4.5: Mean AUC ROC, AUC PRC and ROC enrichment across targets in the MUV test set for our method, DenseFS, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseFS
AUC ROC	0.546	0.507	0.534
AUC PRC	0.003	0.003	0.003
0.5% EF	0.000	2.018	2.407
1% EF	1.204	1.798	3.207
2% EF	1.769	1.521	2.222
5% EF	1.323	1.607	1.338

approaches that have shown meaningful predictive power on MUV are ligand-based methods (Ramsundar et al., 2015; Wu et al., 2018). Furthermore, these have required including target-specific data in the training set, and have often used large external data sets without any regard to the overlap between training and test sets.

Despite the weak performance on the MUV data set, as a result of the clustered cross-validation procedure on DUD-E and the performance on the ChEMBL independent test set, we are confident that our model is learning genuinely useful information about protein-ligand interactions, as opposed to simply artifacts of the construction of DUD-E or overfitting to specific examples. However, substantial further improvements will be required in order to capture more accurately the factors determining binding.

4.4.3 Visualisation

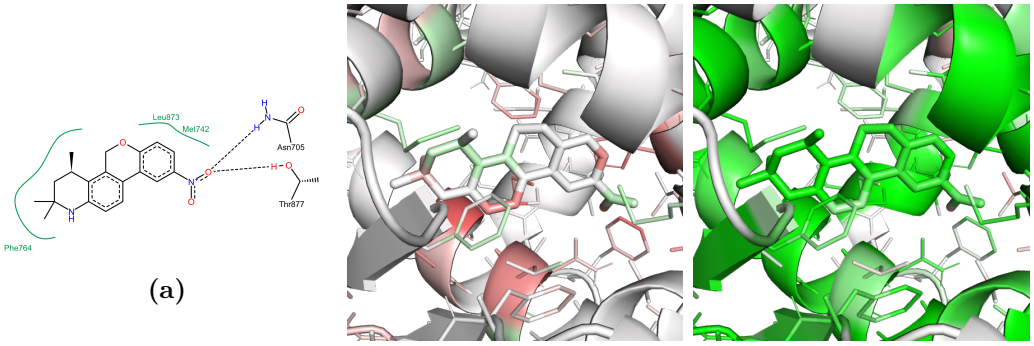
Understanding the components of a protein-ligand complex that govern interactions would greatly assist with both finding initial hits and lead optimisation. In particular, the ability to interpret the CNN's output beyond a simple score is crucial. In order to visualise the drivers of the CNN models' predictions, we replaced the final fully connected layer of a trained model with a $1 \times 1 \times 1$ convolutional layer with the same parameters. This transformed the global classifier into a regional-based classifier at almost no additional computational cost.

In Figure 4.8, we present the analysis of our visualisation procedure for an example protein-ligand complex, the ChEMBL293409 ligand docked against the human androgen receptor (DUD-E target ANDR, PDB ID 2am9), a known active. In Figure 4.8a, we display a 2D diagram of the complex, annotated by PoseView (Stierand and Rarey, 2010). In Figures 4.8b and 4.8c, we coloured the complex according to the regional scores, based on a threshold of 0.5. Scores below 0.5 were coloured in red, whereas scores above 0.5 were coloured in green, with the intensity depending on the magnitude of the difference. We analysed the predictions of the Baseline CNN and DenseFS for this complex.

The Baseline CNN assigned the complex a score of 0.34. While the Baseline CNN scored one of the oxygens in the nitronium ion slightly favourably due to the interaction with the alcohol on the threonine 877 residue, it scored the interaction with the amine on the asparagine 705 residue unfavourably. The Baseline CNN scored the remainder of the molecule overall fairly neutrally for binding. In contrast, DenseFS scored the complex at 0.91, scoring both of the interactions with the nitronium ion highly. In addition, the majority of the ring structure was scored favourably, due to the proximity with the phenylalanine 764, leucine 873, and methionine 742 residues. The bottom two carbons of the left hand benzene ring (in relation to the orientation in Figure 4.8) were scored less favourably than the remainder of the ring structure by both models. This suggests an area of the molecule that could be altered to increase the affinity of the compound.

4.5 Discussion

We have presented a deep learning approach that gives substantial improvement over previous work in virtual screening on both DUD-E and an independent test set, producing state-of-the-art results by all metrics assessed. On DUD-E, our method exhibited around a 70% improvement in AUC PRC and 0.5% ROC enrichment over the Baseline CNN of Ragoza et al. (2017), achieving a higher AUC PRC for 95 of the 102 targets (93%). On the independent ChEMBL set, our method



(a) (b) (c)

Figure 4.8: A visualisation of the known active CHEMBL293409 ligand (4.8a) docked against the DUD-E target ANDR. 4.8b and 4.8c depict the results of the visualisation procedure for the Baseline CNN and DenseFS respectively. Areas of green indicate a score for that region above 0.5, whereas red represents a score below 0.5, with the intensity depending on the magnitude of the difference. The Baseline CNN assigned the complex a overall score of 0.34, while DenseFS scored the complex at 0.91.

outperformed the Baseline CNN on 13 of the 14 targets, resulting in an average increase in AUC PRC of over 40%.

The performance of our method further reinforces the power of approaches that adopt minimal input beyond a spatial representation and limited atom typing of the 3D complex structures. All improvements were obtained by better utilising the same input structures and format. Our approach differed in four key ways from Ragoza et al. (2017), each of which contributed to the improved performance in a somewhat independent manner (Table 4.3).

First, we showed that recent advances in computer vision can be applied to virtual screening by using a densely-connected CNN architecture. This highlights the suitability of re-framing virtual screening as a computer vision problem. We did not perform an extensive review of choices for our deep learning network, and note that several architectures (Chen et al., 2017; Hu et al., 2017) now report improved accuracy on ImageNet. We anticipate that further improvement could be obtained by applying the current state-of-the-art techniques. Adopting the DenseNet architecture resulted in the largest improvement in performance and changing this alone was responsible for a 20.8% - 36.5% increase in AUC PRC during cross-validation (30.4% - 53.4% of overall improvement).

Docking introduces substantial noise to the data sets due to the inaccuracy of many docked poses (Figure C.1). We demonstrated that using an average scoring protocol instead of max scoring provided a 7.6% - 14.4% increase in AUC PRC (11.1% - 21.1% of overall improvement). The benefit of this approach results from reducing the reliance on any single pose, and eliminating the ability for a molecule to be ranked highly from a single pose alone. We expect that with more accurate docked poses the impact of an average scoring protocol over a max or single pose scoring protocol would be reduced, although we would expect some benefit to persist.

Finetuning a universal model on subsets of the available data allowed us to construct protein family-specific models, as opposed to a single universal model, and resulted in average improvements to AUC PRC of 18.3% - 24.0% (26.7% - 35.1% of overall improvement). This let our models form different representations to capture physicochemical nuances exhibited by different families. In our investigation into the importance of family-specific data, we found that (i) very limited family data was required before a family-specific model outperformed a universal one; (ii) the presence of proteins from the same family in the training set is crucial for the methods tested to have high predictive power, although our models continued to exhibit predictive power even when this was not the case; and (iii) continuing to add further examples provided appreciable benefit, even for the largest family present in the training set. This suggests that more data is required to exploit CNN-based methods fully. Future work could investigate the extent to which this effect persists, or if at some point learning saturation is reached and further data becomes redundant.

Finally, combining the predictions of three models (trained with different random seeds) in an ensemble improved average AUC PRC by 3.4% - 11.0% (5.0% - 16.1% of overall improvement). While this is a relatively minor benefit, this technique consistently improved predictions, despite using only three models in our ensemble. A major drawback is that the additional computation required directly scales with the number of models used; however, this is somewhat mitigated by the fully parallelisable nature of additional models, both during training and at test time.

While our methods represent a substantial improvement over the state-of-the-art on the DUD-E and ChEMBL benchmarks, and show the promise of a CNN-based approach, the limited predictive power on the MUV data set underscores the challenges still faced in virtual screening.

Despite the impressive performance of deep learning methods for virtual screening both retrospectively and prospectively (e.g. Stecula et al., 2020; Adeshina et al., 2020), two recent studies have suggested that inadequacies in benchmarks can allow machine learning methods to perform strongly by utilising features that are known not to be important for binding (Sieg et al., 2019; Chen et al., 2019). In particular, both publications show limited degradation in the performance of some structure-based machine learning methods when all information about the protein receptor is removed, both in training and at test time. This arises, in part, due to the decoy molecules in current benchmarks being biased in basic chemical properties.

In the following chapter, we propose a deep learning-based generative approach, DeepCoy, to address this limitation of decoys molecules and propose new decoys molecules for commonly-used structure-based benchmarking sets with substantially reduced bias.

Imitation is the sincerest of flattery.

— Charles Caleb Colton, *Lacon: or, Many things in few words*

5

Generating Property-Matched Decoy Molecules Using Deep Learning

Contents

This chapter is based on work described in the following publication:

Fergus Imrie, Anthony R. Bradley, and Charlotte M. Deane (2021). Generating Property-Matched Decoy Molecules Using Deep Learning. *Bioinformatics*, doi: 10.1093/bioinformatics/btab080

An open-source implementation of the method described in this chapter is

available at https://github.com/oxpig/DeepCoy.

5.1 Preface

In Chapter 4, we discussed how additional data improved the performance of structure-based virtual screening models. This suggests that new and improved training and benchmarking sets are required to harness the full power of deep learning models in drug discovery. Size is far from the only consideration in dataset construction; it is essential for method development that training and benchmarking sets enable robust evaluation of proposed models and do not contain biases that are unrepresentative of the general data generating process (Verdonk et al., 2004). Crucially, conclusions regarding the relative predictive power of methods in the proposed evaluation should be representative of their relative real-world performance.

As noted in Section 4.5, recent studies have shown that the decoy molecules in commonly used structure-based virtual screening datasets are biased (Sieg et al., 2019; Chen et al., 2019). This means that it is possible for methods to exploit these biases to separate actives and decoys rather than learn to perform molecular recognition. This is a fundamental issue that has the potential to prevent methods generalising and hinders virtual screening method development by obfuscating benchmarking results.

The decoy molecules contained in benchmarking sets have so far been selected from virtual compound libraries (e.g. Mysinger et al. 2012). However, a common criticism of such libraries is the lack of molecular complexity compared to bioactive compounds (Dandapani and Marcaurelle, 2010; Méndez-Lucio and Medina-Franco, 2017). Adopting bioactive compounds as decoys for other targets has been proposed (Chen et al., 2019), although due to the limited number of such compounds (for example, ChEMBL contains bioactivities on fewer than 2 million compounds, Mendez et al., 2019) it is unlikely that suitable decoys will be contained in this set for many active molecules.

Inspired by the success of machine learning models for other design tasks such as the DeLinker methodology described in Chapters 2 and 3, in this chapter we

explore a generative approach to decoy design. We propose a deep learning method (DeepCoy) that generates decoys to a user's preferred specification in order to control decoy bias or construct sets with a defined bias. We validated DeepCoy using two established benchmarks, DUD-E (Mysinger et al., 2012) and DEKOIS 2.0 (Bauer et al., 2013). For all 102 DUD-E targets and 80 of the 81 DEKOIS 2.0 targets, our generated decoy molecules more closely matched the active molecules' physicochemical properties while introducing no discernible additional risk of false negatives. The DeepCoy decoys improved the Deviation from Optimal Embedding (DOE, Vogel et al., 2011) score by an average of 81% and 66%, respectively, decreasing from 0.166 to 0.032 for DUD-E and from 0.109 to 0.038 for DEKOIS 2.0. Further, the generated decoys are harder to distinguish than the original decoy molecules via docking with Autodock Vina (Trott and Olson, 2010), with virtual screening performance falling from an AUC ROC of 0.70 to 0.63. We believe that this substantial reduction in bias will benefit the development and improve generalisation of structure-based virtual screening methods.

5.2 Introduction

As described in Chapters 1 and 4, virtual screening is a computational approach that is often used in early stage drug discovery to help find molecules that interact with protein targets with high affinity and specificity. Numerous prospective applications of virtual screening have been reported, reducing the cost and improving the hit-rate of experimental verification (e.g. Lyu et al., 2019; Liu et al., 2019).

There are a variety of datasets available for retrospectively benchmarking virtual screening methods. These sets consist of a collection of active and inactive molecules for a range of protein targets. Frequently used examples for structure-based virtual screening (SBVS) are DUD (Huang et al., 2006) and DUD-E (Mysinger et al., 2012), DEKOIS (Vogel et al., 2011; Bauer et al., 2013), and MUV (Rohrer and Baumann, 2009). These datasets are described in more detail in Chapter 1.

While experimentally-verified inactives represent the gold standard for dataset construction (Lagarde et al., 2015b), suitable inactive molecules are often not

available. As such, using presumed inactives is typically necessary in SBVS datasets (Réau et al., 2018). There are efforts to construct sets using only known inactives (e.g. Rohrer and Baumann, 2009; Tran-Nguyen et al., 2020); however, these are relatively limited in size and breadth of protein targets and are not yet suitable for training general-purpose SBVS models using modern machine learning methods.

Bias in virtual screening datasets can be split into three main types: artificial enrichment, analogue bias, and false negative bias (Réau et al., 2018). Artificial enrichment captures the performance that can be attributed to the differences in chemical space between the active and decoy molecules. Analogue bias arises from limited diversity of the active molecules, while false negative bias describes the risk of active compounds being present in the decoy set, which could lead to an underestimation of the screening performance. It is crucial that benchmarking sets minimise these bias (e.g. Sieg et al., 2019).

To achieve this, decoys are typically selected to match the chemical properties of active molecules while simultaneously ensuring structure mismatching to minimise the chance of decoys being binders ("property-matched decoys", e.g. Mysinger et al., 2012; Adeshina et al., 2020).

Alternative approaches for decoy construction have also been proposed. For example, one criticism of property-matched decoys is that they inherently struggle to capture the chemical diversity present in screening libraries (Li et al., 2020a), and thus several reports have used property-unmatched decoys selected at random from a representative dataset (e.g. Sun et al., 2016). In other publications, actives from other targets have been adopted to produce an "actives as decoys" set (e.g. Chen et al., 2019).

However, SBVS methods should be able to discriminate between actives and inactives on the basis of structural information alone and not depend on exploiting differences in physicochemical properties between actives and inactives. This can only be ensured if the physicochemical properties of decoys in benchmarking sets match those of the actives (Verdonk et al., 2004; Nicholls, 2008).

Property matching arbitrary actives is challenging and, despite improvements, still leads to substantial differences in molecular properties between actives and decoys (Chaput et al., 2016). It has been shown that on several widely-used datasets it is possible to discriminate actives from inactives from these properties alone (Wallach and Heifets, 2018; Sieg et al., 2019). Hence closer matching is required to ensure retrospective testing is not over-optimistic (e.g. Verdonk et al., 2004).

In recent years, many machine learning methods have been trained and evaluated on these datasets (e.g. Wójcikowski et al., 2017; Imrie et al., 2018). The reported results show that these methods substantially outperform other methodologies such as empirical and knowledge-based scoring functions at SBVS.

Concerningly, some reports have suggested that a key driver of the retrospective performance of machine learning-based systems is hidden biases in the training data, such as physicochemical differences, and have questioned the extent to which such methods are learning to perform molecular recognition (Sieg et al., 2019; Chen et al., 2019). While prospective successes have demonstrated that such methods can be useful (e.g. Stecula et al., 2020; Adeshina et al., 2020), both Sieg et al. (2019) and Chen et al. (2019) emphasise the need for improved validation on unbiased datasets.

The challenges of decoy design are in part due to the inherent limitations of matching to an explicit, fixed database of potential decoys. While virtual libraries such as ZINC (Sterling and Irwin, 2015) have grown considerably, they still represent only a tiny fraction of potential drug-like chemical space (Polishchuk et al., 2013) and are insufficient for closely matching core chemical properties of many active molecules.

Wallach and Lilien (2011) pioneered the use of a generative approach to construct virtual decoy sets for the original DUD (Huang et al., 2006) targets with tighter property matching than the decoys selected from ZINC. They used a rules-based algorithm employing a library of chemical building blocks and bridges to iteratively generate possible decoys. However, their method ignored synthetic feasibility and, despite clear improvements in property matching, has not been widely adopted.

Machine learning models for molecule generation have been proposed as an alternative to human-led design and rules-based transformations and have shown great promise in several molecular design tasks (e.g Segler et al., 2018; Zhavoronkov et al., 2019).

In this chapter, we describe DeepCoy, a deep learning method using graph neural networks, to generate decoy molecules. DeepCoy takes as input an active molecule and generates property-matched decoy molecules. This eliminates the need to use a database to search for molecules and allows decoys to be generated for the requirements of a particular active molecule and the user's specification.

The properties can be chosen by the user depending on their objective, and in this chapter we demonstrate the ability of DeepCoy to learn to produce decoy molecules with different sets of matched properties, highlighting the flexibility of our approach. We validated our generative model using two established SBVS benchmarks, DUD-E and DEKOIS 2.0. For all 102 DUD-E targets and 80 of the 81 DEKOIS 2.0 targets, our generated decoy molecules more closely matched the physicochemical properties deemed by the respective datasets to be non-informative for binding, improving property matching as measured by DOE score by 81% and 66% for DUD-E and DEKOIS 2.0, respectively.

Finally, we demonstrate that the generated decoys are harder to distinguish from active molecules than the original decoy molecules with docking using Autodock Vina (Trott and Olson, 2010). This ability to substantially reduce bias will benefit the development and improve generalisation of structure-based virtual screening methods.

5.3 Methods

This chapter describes a novel approach using deep learning to propose molecules that match a set of features provided by the user. We achieve this with a generative model using graph neural networks. Our model makes no underlying assumptions regarding the nature of the properties that are to be matched, and relies only on a training set of paired molecules exhibiting the desired similarities.

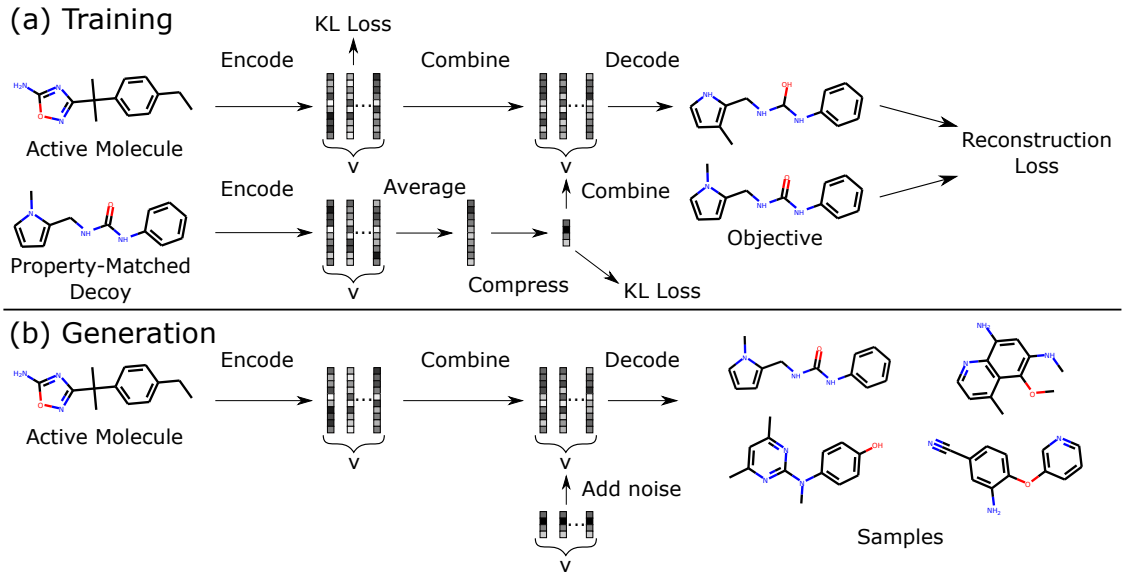


Figure 5.1: Illustration of training and generation procedures. (a) Pairs of structurally dissimilar molecules with similar physicochemical properties are provided as input. The model is trained to convert one molecule into the other from a combination of the encodings of both molecules. (b) At generation time, the model is given only the active molecule and is able to sample a diverse range of property-matched decoys by combining the encoding of the active molecule with random noise.

5.3.1 Generative Model

In order to generate decoys we use an adapted version of Imrie et al. (2020), described in Chapter 2, which was designed for linker generation. Imrie et al. (2020) builds on the generative process introduced by Liu et al. (2018) that constructs molecules "bond-by-bond" in a breadth-first manner. The most substantial differences with the DeLinker model (Chapter 2) are the input data and goal of the generative process.

DeepCoy takes an active molecule as input and generates a new molecule that has similar physicochemical properties but is structurally dissimilar. This is achieved by building new molecules in an iterative manner "bond-by-bond" from a pool of atoms. In this framework, the user is able to control the maximum number of heavy atoms in the molecules and, if desired, specific heavy atoms or partial substructures.

Minimal chemical knowledge is directly incorporated in our model; this takes the form of a set of permitted atom types and basic atomic valency rules which ensure the chemical validity of generated molecules. The model is required to learn all other decisions required to generate molecules.

Our method learns through a supervised training procedure using pairs of molecules (Figure 5.1). Inspired by Jin et al. (2019b), we frame decoy generation as a multimodal graph-to-graph translation problem. We train DeepCoy to convert graphs of active molecules into property-matched decoys under an augmented variational autoencoder setting, employing standard gated-graph neural networks (Li et al., 2016) in both the encoder and decoder. DeepCoy implicitly learns which properties to keep constant and is not explicitly told which properties to match, nor their values. This provides a highly flexible framework, and makes it possible to learn from pairs of molecules without quantifying their similarity.

We employed a training objective similar to the standard VAE loss, including a reconstruction loss and a Kullback-Leibler (KL) regularisation term:

$$\mathcal{L}_{Total} = \mathcal{L}_{recon} + \lambda_{KL}\mathcal{L}_{KL}\,.$$

The reconstruction loss, $\mathcal{L}_{recon}$, is composed of two terms resulting from the error in predicting the atom types and in reconstructing the sequence of steps required to produce the target molecule.

To improve the quality of generated molecules, we adopted a novel loss function that deviates from a standard cross entropy loss for the sequence of actions adopted in Chapter 2 and by Liu et al. (2018). Instead of each step in the generative processes having equal importance, we reweighted the probabilities of actions by the frequencies of the induced subgraphs across the training set of molecules, leading to the revised cross-entropy loss:

$$\mathcal{L} = -\overline{f} * \log\left(\frac{p(x_j)f(x_j)}{\sum_i p(x_i)f(x_i)}\right)\,,$$

where $p(x_i)$ is the probability of choosing action x_i, $f(x_i)$ is the reciprocal frequency of the induced local subgraph by taking action x_i, the sum is over all permitted actions, and $\overline{f}$ is the average of f over all permitted actions. This has the effect of reducing the chance of introducing local subgraphs that are not present in the training set. We observe that this change does not meaningfully affect the novelty of generated molecules compared to the standard cross-entropy loss.

For a more detailed description of the model, see Chapter 2 and Appendix D.

5.3.2 Training Set

We constructed pairs of molecules to train our model from the 250 000 molecule subset of ZINC (Sterling and Irwin, 2015) selected at random by Gómez-Bombarelli et al. (2018) as follows.

We first characterised compounds by their physicochemical properties. The properties can be selected by the user and we demonstrate the effectiveness of our framework using multiple sets of properties (described in Section 5.3.4). Pairs of molecules were constructed to satisfy the following criteria: (1) identical heavy atom count and counts of specific heavy atoms (C, N, O, S, Cl, F), (2) high similarity in property-space, and (3) low structural similarity. We measured similarity in property-space using the Euclidean distance between normalised property values and structural similarity by the Tanimoto similarity between the Morgan fingerprints (radius 2, 1024 bits, Rogers and Hahn, 2010).

In order to create training sets for our large scale benchmarking experiments (see Section 5.3.4), we set the maximum permitted structural similarity between a pair of molecules at 0.15 and the maximum distance in property space to 0.20 for the assessment on DUD-E and 0.07 for DEKOIS 2.0. The thresholds were set to ensure roughly equal training set sizes and the differences were as a result of the different sets of properties to unbias. This resulted in a training set of 131,199 pairs for DUD-E and 103,170 for DEKOIS 2.0. We selected 1000 pairs for model validation, and used the remainder to train our model.

5.3.3 Assessment

Several metrics have been proposed to assess artificial enrichment and the risk of false negatives introduced by using putative decoy molecules. Vogel et al. (2011) proposed the deviation from optimal embedding score (DOE score) and the doppelganger score to assess the quality of physicochemical matching of decoys and risk of introducing latent active molecules, respectively. These metrics are our primary way of assessing the generated decoy molecules.

The DOE score measures the quality of the embedding of actives and decoys in chemical space by employing a series of receiver operating characteristic curves (ROC curves) for each active calculated using the physicochemical properties of interest. The DOE score is the average absolute difference between these ROC curves and a random distribution. An optimal embedding of actives and decoys achieves a DOE score of zero, while complete separation in physicochemical space results in an DOE score of 0.5.

The doppelganger score captures the structural similarity between actives and their most structurally related decoys. We generated functional fingerprints (similar to FCFP6) using RDKit (Landrum, 2006) for all compounds and evaluated the structural similarity between actives and decoys using the Tanimoto coefficient. For each decoy molecule, its doppelganger score is the maximum similarity across all actives. For each target, we report the mean doppelganger score over all decoys and the maximum structural similarity between an active and a decoy.

An alternate way to quantify the physicochemical property matching is via predictive models trained on such properties. Bias can be measured using machine learning performance directly (Sieg et al., 2019) or a measure of bias can be derived from such models (Wallach and Heifets, 2018). We assessed bias using both approaches. First, we trained 1-nearest neighbour (1NN) and random forest (RF) models on all possible subsets of the physicochemical properties deemed non-informative for binding. We adopted 10-fold cross-validation on a per-target basis and assessed performance via the area under the ROC curve (AUC ROC), following Sieg et al. (2019). In addition, we calculated AVE (Wallach and Heifets, 2018) using the same properties and cross-validation splits..

We also considered the virtual screening performance of docking using AutoDock Vina (Trott and Olson, 2010), specifically the smina (Koes et al., 2013) implementation. Ligands were docked against the reference receptor within a box centered around the reference ligand with 8 Å of padding. We used smina's default arguments for exhaustiveness and sampling. We focussed our analysis on performance as measured by AUC ROC.

5.3.4 Large Scale Benchmarking Experiments

We assessed our method using two of the most popular SBVS datasets, DUD-E (Mysinger et al., 2012) and DEKOIS 2.0 (Bauer et al., 2013).

We trained a separate model for each of the datasets to demonstrate the flexibility of our method to learn to match different sets of properties. For DEKOIS 2.0, we used the same eight properties employed to construct the dataset (Bauer et al., 2013). To assess whether our framework extends to a higher dimensional property space, we trained our model to match twenty-seven properties for our assessment on DUD-E, instead of only the original six properties selected by Mysinger et al. (2012). Training set construction is described in Section 5.3.2 and a complete list of physicochemical properties is provided in the Section D.2. Despite training for this broader array of properties and selecting the final decoys for the DUD-E set based on all 27 properties, we report results calculated using the original six DUD-E properties, unless otherwise stated. Not selecting the DeepCoy set optimally with respect to the original six DUD-E properties will result in inferior performance of DeepCoy, but will allow us to evaluate how our method performs when required to unbias a larger number of properties.

We filtered the active molecules in both datasets to exclude those containing rare atom types outside of the scope of our model (c. 1% of actives, see Appendix D for a list of permitted atom types). This led to no actives for DUD-E target FPPS due to the presence of phosphorus in all active molecules. To address this and demonstrate the flexibility of our generative approach, we trained a separate model for this target (see Appendix D for more details). For each active, we generated 1000 candidate decoys using DeepCoy. We then selected final decoy sets using a similar pipeline to DEKOIS 2.0. Generated molecules were initially filtered by the difference in heavy atom counts and maximum doppelganger score using an iterative procedure until at least 100 candidate decoys remained. The final decoys were selected from these candidate decoys in a greedy manner based on the sum of the normalised property difference and LADS score (Bauer et al., 2013). While this greedy selection policy is likely not optimal, we adopted it primarily due to

its simplicity. We then compared the generated decoy sets to the original decoy sets using the metrics described in Section 5.3.3.

5.4 Results

We assessed our ability to generate property-matched decoy molecules with varying requirements through two widely-used SBVS datasets, DUD-E and DEKOIS 2.0. For both sets, we generated new decoy molecules and compared these to the original set, assessing the generated molecules with respect to the same physicochemical properties used to select the original decoys. We show that:

- DeepCoy generated decoys substantially improve property matching compared to the original database decoys.
- DeepCoy generated decoys do not introduce additional risk of false negatives.
- DeepCoy generated decoys are harder to distinguish from active molecules than the original DUD-E decoys with docking using AutoDock Vina, despite being as structurally dissimilar from the active molecules as the original decoys.

Our results demonstrate that our framework is an alternative to database approaches for selecting property-matched decoy molecules, while offering full flexibility to the user regarding choice of specific properties and how to choose the final decoys from the generated molecules.

5.4.1 Physicochemical Property Matching

Across both DUD-E and DEKOIS 2.0, our generated decoy molecules more closely matched the physicochemical properties deemed by the respective datasets to be non-informative for binding than the original decoys (see Appendix D for a full list of properties).

When selecting decoys based on the same properties as the original datasets, our generated decoys improved the DOE score by an average of 81% and 66%,

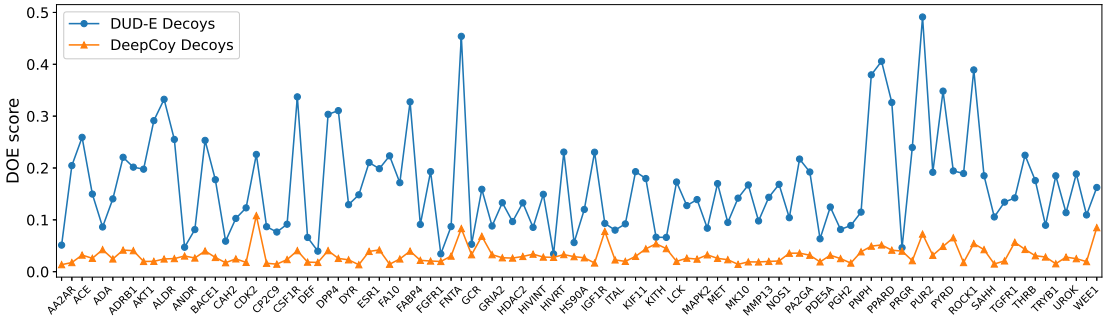


Figure 5.2: DOE scores of the original DUD-E set (blue) compared to the DeepCoy generated decoys (orange). For all targets, the DeepCoy generated decoys have lower DOE score (lower is better), with the average DOE score decreasing by 81% from 0.166 to 0.032. The x-axis displays each DUD-E target in the same order as they appear in the DUD-E database (http://dude.docking.org/targets). The targets with even indices are not labeled on the x-axis due to space limitations.

respectively, decreasing from 0.166 to 0.032 for DUD-E and 0.109 to 0.038 for DEKOIS 2.0. In this setting, the DOE score was improved by using DeepCoy generated decoys for all 102 DUD-E targets (Figure 5.2) and 80 of the 81 DEKOIS 2.0 targets (Figure D.1). The only DEKOIS 2.0 target that did not show an improvement in DOE score had DOE scores below 0.04, corresponding to an almost perfect embedding for both the DeepCoy and original decoy molecules. Finally, DeepCoy generated decoys achieved a DOE score below 0.1, indicating a close to optimal embedding (Bauer et al., 2013), for 101 of the 102 DUD-E and 79 of the 81 DEKOIS 2.0 targets, while the original decoys only met this threshold for 32 DUD-E and 48 DEKOIS 2.0 targets.

We selected our final decoy set for DUD-E using all 27 properties, rather than just the six used to construct the original dataset. The average DOE score of this set was 0.045, a comparable improvement of 73%, outperforming the original decoys for 98 of the 102 targets (Figure D.2). Importantly, the DeepCoy decoys experienced no drop in performance when all 27 properties were included in the calculation of DOE score, with an average score of 0.041 (Figure D.3). In contrast, the original decoys experienced a substantial decline to 0.222, proving matching this larger set is non-trivial. This demonstrates the ability of DeepCoy to scale successfully to a high-dimensional property space to unbias.

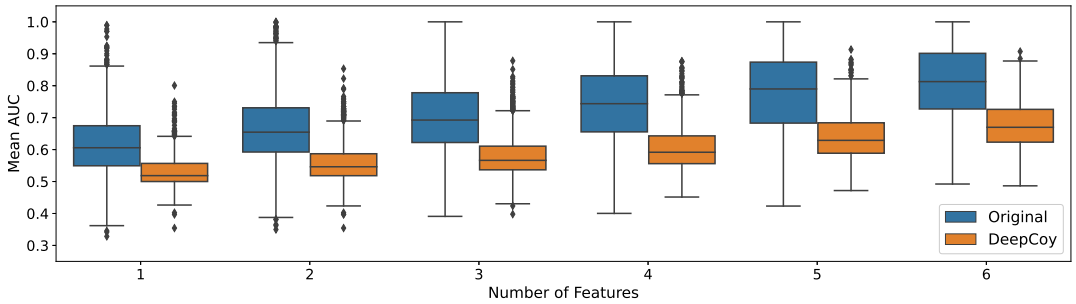


Figure 5.3: Results of the machine-learning based assessment of physicochemical property matching on DUD-E. Random forests were trained to predict whether a compound was an active or a decoy based on the unbiased features. Virtual screening performance was assessed by AUC ROC for the original DUD-E decoys and DeepCoy generated decoys. The DeepCoy generated decoys resulted in a reduction in the median per-target AUC ROC using all 6 features from 0.81 to 0.67 indicating a substantial reduction in bias.

A similar improvement can be seen when assessing property matching via the ability of machine learning models to predict whether a compound is an active or a decoy when trained on the physicochemical properties deemed non-informative for binding (Figures 5.3, D.4). On the DUD-E set, using all 6 features, the median (average) AUC ROC decreased from 0.66 (0.66) to 0.55 (0.56) and 0.81 (0.80) to 0.67 (0.68) for the 1-nearest neighbour and random forest models, respectively, for the DeepCoy decoys compared to the original set. A similar reduction was observed when using any combination of the physicochemical properties (Figure 5.3).

Assessing bias using AVE also demonstrated a significant reduction in bias with a reduction in median AVE (using all 6 features) of 72% from 0.17 to 0.05 (Figure D.5). As noted by Sieg et al. (2019), while there is a notable correlation between AVE and machine learning performance, AVE does not always explain high predictive performance (Figure D.6).

However, even with the much improved property matching of the DeepCoy decoys, there remains some signal in the physicochemical properties. This is in part due to the high level of similarity between many of active molecules in DUD-E, a factor that should be controlled for when constructing the dataset to ensure low levels of bias (Wallach and Heifets, 2018). This is exemplified by the DUD-E target SAHH. DeepCoy decoys substantially reduced the DOE for the DUD-E properties

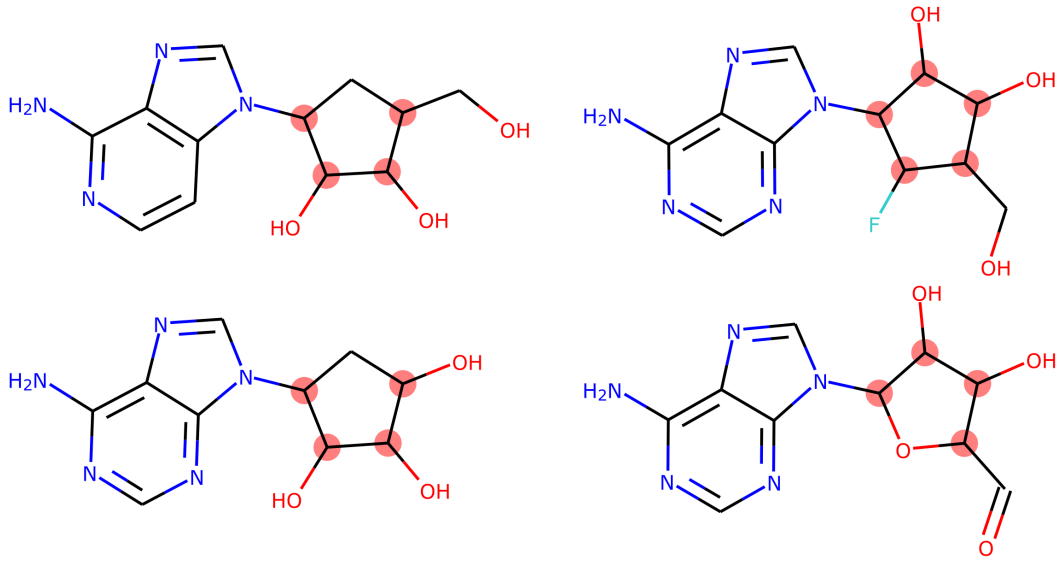


Figure 5.4: Four representative active ligands for DUD-E target SAHH. The 63 active molecules for SAHH have high levels of structural similarity, with all sharing similar fused rings systems. These ligands all have at least four stereocenters (highlighted in red, stereochemistry not shown), a property shared by over half of the active molecules for this target.

to 0.11 (original decoys: 0.19). However, when assessing the decoys using the larger set of 27 properties, it became very challenging to unbias the decoy set (DeepCoy DOE: 0.29, original DOE: 0.34) due to high levels of similarity within the active set. All 63 active molecules for SAHH contain a similar fused ring system, while around half of the active molecules have 4 stereocenters (Figure 5.4). The considerable structural similarity, coupled with the high number of stereocenters for molecules of this size, was the primary cause of the poor DOE scores and is highly challenging to overcome via better decoy selection alone.

5.4.2 False Negative Bias

It is crucial that the improvement in property matching achieved by DeepCoy was not as a result of increasing the similarity between the active and decoy molecules, risking increasing false negative bias.

The average doppelganger score (Vogel et al., 2011), a measure of the structural similarity between actives and decoys, remained consistent on the DUD-E set at 0.26 for the DeepCoy decoys and 0.25 for the original decoys, while the average

maximum doppelganger score per target fell from 0.37 for the original decoys to 0.34 for the generated decoys. We saw similar results for the DEKOIS set; the average doppelganger score fell slightly (DeepCoy: 0.22, Original: 0.25), while there was a significant drop in maximum doppelganger score from 0.44 to 0.30 when using the DeepCoy decoys.

These results strongly suggest that the decoys generated by DeepCoy should not carry an increased risk of false negative bias compared to the original decoys.

5.4.3 Structure-Based Virtual Screening

We further validated the quality of our generated decoys by docking the DUD-E set. Several publications have shown that most docking scoring functions are influenced by basic physicochemical properties (e.g. Chaput et al., 2016). In particular, Wallach and Lilien (2011) showed that property mismatching can lead to an arbitrary increase *or* decrease in virtual screening performance of docking methods. Thus docking performance cannot be used alone to evaluate decoy molecules.

However, overall, better quality decoys should be harder to distinguish from active molecules, in particular if such decoys also more closely match the physicochemical properties of the active molecules and do not display an increased risk of false negatives.

The virtual screening performance of AutoDock Vina on the DUD-E set fell to an average per-target AUC ROC of 0.63 for the DeepCoy generated decoys compared to 0.70 for the original decoy molecules. There was a relatively high correlation between the per-target docking performance using the original and DeepCoy decoys (Pearson's R: 0.56, Figure D.11) driven by the active molecules, which are common between both sets. However, for 86 of the 102 targets, the DeepCoy decoys led to a lower AUC ROC than the original decoys.

The decrease in the discriminative power of SBVS is likely driven by the closer property matching of the generated decoys, consistent with other studies (e.g. Vogel et al., 2011). This further reinforces the need for unbiased benchmarking sets, even for non-machine learning based scoring functions. For example, the original decoys

for IGF1R resulted in a DOE score of 0.23, indicating a large mismatch between the active and decoy molecules. When this set was docked, Vina performed well with an AUC ROC of 0.81. In contrast, the DeepCoy generated decoys gave a DOE score of 0.02, a c. 90% reduction, and had a lower AUC ROC of 0.56. The inability for DeepCoy generated decoys to be easily separated from active molecules via docking together with the lack of additional risk of false negative is further validation of the suitability of these molecules for testing SBVS methods.

Deep learning-based SBVS methods (e.g. Ragoza et al., 2017) have become increasingly popular due to their strong empirical performance. As discussed in Section 5.2, it has been suggested that, for models trained on DUD-E, a driver of this could be dataset biases (Sieg et al., 2019; Chen et al., 2019). To assess SBVS methods with an external validation set, Ragoza et al. (2017) utilised a subset of the datasets curated from ChEMBL (Bento et al., 2014) by Riniker and Landrum (2013), selecting the targets to ensure that models were evaluated on dissimilar binding sites to those in the training set.

Such an external test set should be more representative of real-world use and should not share biases with the training set. However, likely due to the use of decoy molecules from ZINC, the ChEMBL targets share similar biases to DUD-E as measured by the same metrics as our assessment of DUD-E and DEKOIS 2.0 (Figures D.7-D.9). In particular, random forests trained on the unbiased physicochemical properties of compounds in DUD-E achieved high virtual screening performance on the ChEMBL test sets (average AUC ROC DUD-E features 0.70, larger feature set 0.84, Figure D.10). In contrast, when trained on the DeepCoy decoys, the RF model had limited discriminative power on the ChEMBL test sets (average AUC ROC DUD-E features 0.54, larger feature set 0.57). We thus caution against using these datasets as external validation for models trained on DUD-E due to the similar physicochemical biases.

Since there is limited bias between the version of DUD-E employing DeepCoy decoys and the ChEMBL test sets, we can be more confident that predictive power on the ChEMBL test sets of models trained using DeepCoy decoys arises from

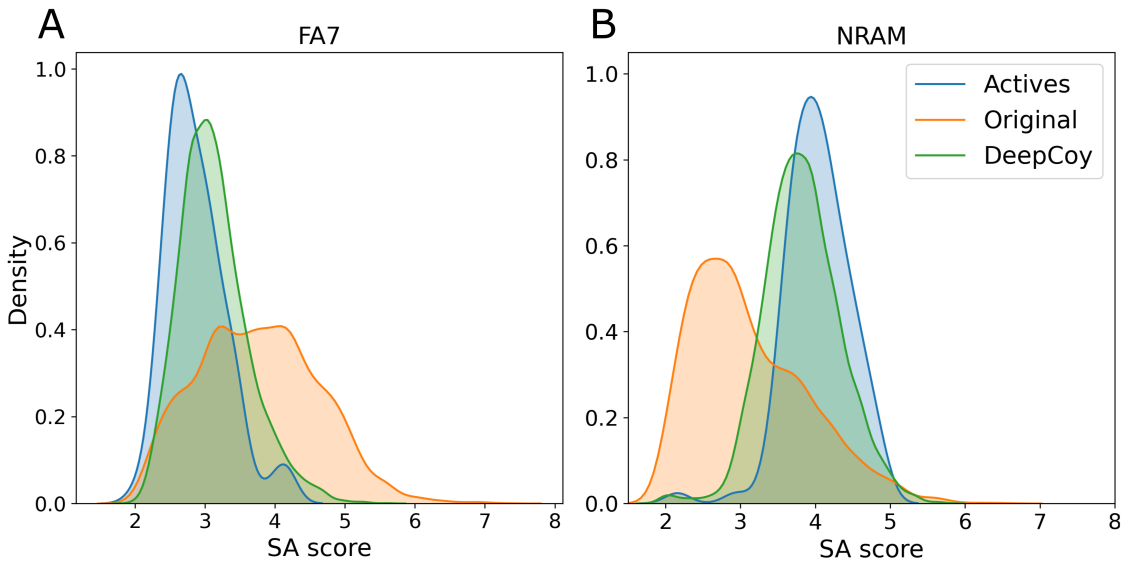


Figure 5.5: Synthetic accessibility (SA) scores for the active molecules (blue), original DUD-E decoys (orange), and DeepCoy generated decoys (green) for DUD-E targets FA7 (A) and NRAM (B). The DeepCoy generated decoys much more closely match SA scores of the active molecules than the original DUD-E decoys for both targets.

the model having learnt meaningful features. We trained the convolutional neural network architectures of Ragoza et al. (2017) and Imrie et al. (2018) on the version of DUD-E employing DeepCoy decoys (see Section D.3 for more details). All of the CNN-based models outperformed AutoDock Vina on the ChEMBL test sets (Table D.1). In particular, both gnina (Ragoza et al., 2017) and DenseU (Imrie et al., 2018) improved early enrichment by around 50% (1.0% ROC EF 11.0 and 11.2, respectively) compared to AutoDock Vina (7.5), while the performance of DenseFS (Imrie et al., 2018) improved by 110% (16.0). These results demonstrate that DeepCoy decoys can be used to train complex SBVS models.

5.4.4 Synthesisability of Generated Decoys

A primary reason for selecting decoys from a virtual library of molecules is their high chance of synthesisability and the ability to purchase such compounds. However, for retrospective screening, or indeed training machine-learning models, decoys do not necessarily need to be synthetically feasible, but should be chemically possible (Yuriev, 2014).

A common criticism of molecules generated using *de novo* design methods is that they are not synthetically accessible. We assessed the synthetic feasibility of molecules using the synthetic accessibility score (SA score, Ertl and Schuffenhauer, 2009). SA score ranges from 1 (easy to make) to 10 (very difficult to make), with the majority of bioactive molecules falling between 2.5 and 4.5. The generated decoys have not been optimised for SA score nor selected based on this property. Despite this, the decoys generated by DeepCoy are, on average, relatively synthetically accessible, with an average SA score on the DEKOIS 2.0 set of 3.55 compared to 3.21 for the original decoys and 3.13 for the active molecules.

SA score is broadly a measure of molecular complexity, but with no regards to the precise functionality nor whether a given molecule should bind to a given target. Thus decoys should match the SA score (or a similar metric) of the active molecules, otherwise molecular complexity could become a distinguishing factor between actives and decoys.

As such, when generating decoys for DUD-E we included SA score as one of the properties to unbias. The DeepCoy decoys (average SA score: 3.27) more closely matched the SA score of the active molecules (2.99) than the original decoys (3.41). We further demonstrate the effect this has on the SA score of decoy molecules by examining FA7, the median performing target (measured by DOE score) for the original decoy molecules, and NRAM, a target for which the active molecules have relatively high SA scores. The distributions of SA scores for FA7 and NRAM are shown in Figure 5.5 (mean SA score FA7 actives 2.9, NRAM actives 4.0). The DeepCoy decoys much more closely matched the SA score of the actives molecules of both targets than the original decoys, which did not match the SA score of the actives molecules in either case. This exemplifies the mismatch between SA scores of active and decoy molecules for some targets in DUD-E and demonstrates the adaptability of our generative framework.

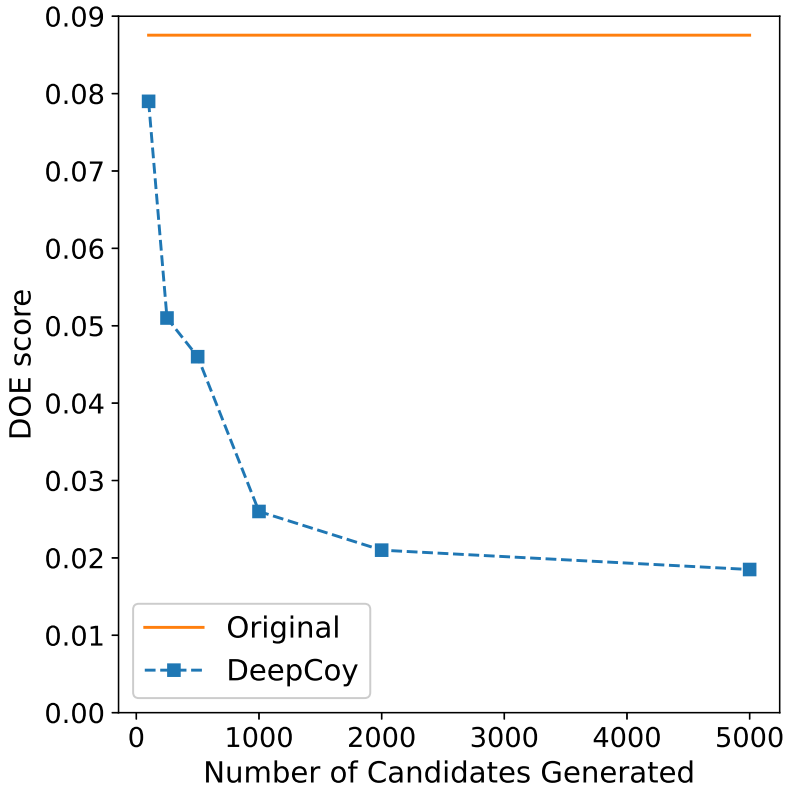


Figure 5.6: The effect on the DOE score of the final decoy set as the number of candidate decoys generated by DeepCoy is varied for DEKOIS 2.0 target P38-alpha. In all cases, 30 decoys per active molecule are chosen. The DOE score for the DeepCoy generated decoys decreases rapidly as more candidates are generated, before slowing after 2000 potential decoys are generated. Even with only 100 candidates, the DOE score for the DeepCoy decoys is lower than the original decoys.

5.4.5 Effect of Number of Generated Candidate Decoys per Active

We investigated how the number of candidate decoys generated per active with DeepCoy affects the quality of the final decoy set. Ideally as few candidates would be generated as possible; however, generating more candidates is likely to lead to a higher quality final decoy set. This creates a trade-off between quality and computational requirements.

To explore this, we used the DEKOIS 2.0 target P38-alpha. This target achieved median performance as measured by DOE score with the original decoys, with a DOE score of 0.088 and doppelganger score of 0.22. We constructed multiple decoy

sets by varying the number of candidate decoys generated by DeepCoy between 100 and 5000 per active molecule and selecting the best 30 as described previously.

Even generating only 100 candidates per active, the DOE score of the DeepCoy decoys was 0.079, representing an improvement over the original decoys of around 10%. As more candidates are generated, this difference rapidly increases (Figure 5.6), with a DOE score of 0.026 when 1000 candidates are generated, a 70% reduction compared to the original decoys. This continues to improve as more candidates are generated, albeit at a slower rate, reaching a score of 0.019 when 5000 are generated. The mean doppelganger score also decreased from 0.26 with 100 candidate per active to 0.23 when 5000 candidates were generated.

While there is a clear dependence between the quality of the final decoy set and the number of candidates generated, DeepCoy generated molecules outperformed the original decoys even when a very limited number of candidates were generated. Unlike a database approach where the maximum performance is limited by the dataset, in our framework the user can decide the desired level of property matching and risk of false negatives, generating additional candidate decoys until this is reached.

5.5 Discussion

We have developed a graph-based deep learning method for generating property-matched decoy molecules for virtual screening. Unlike almost all virtual screening benchmarks, our method does not rely on a database molecules from which to select decoys but instead designs ones that are tailored to the active molecule.

We validated our generative model using two established structure-based virtual screening benchmarks, DUD-E and DEKOIS 2.0. For all 102 DUD-E targets and 80 of the 81 DEKOIS 2.0 targets, our generated decoy molecules more closely matched the physicochemical properties deemed by the respective datasets to be non-informative for binding, while introducing no additional false negative bias.

In particular, our generated decoys decreased the average DOE score from 0.166 to 0.032 for DUD-E and 0.109 to 0.038 for DEKOIS 2.0, an improvement of 81% and

66%, respectively. In addition, we demonstrated that they are no easier to distinguish than the original decoy molecules via docking with smina/Autodock Vina.

We believe that this substantial reduction in bias will benefit the development and improve generalisation of structure-based virtual screening methods. Currently, methods can perform well on retrospective benchmarks without performing molecular recognition by simply learning underlying biases (Wallach and Heifets, 2018; Sieg et al., 2019; Chen et al., 2019). Thus it is unclear if improvements are genuine or due to more closely capturing these biases. However, when such models were trained on DeepCoy decoys which have limited bias and do not share significant bias with the test set, they displayed substantial predictive power (Table D.1). While our generated decoys might contain new biases of which we are currently unaware, these results together with recent prospective successes (e.g. Stecula et al., 2020; Adeshina et al., 2020) is good evidence that such methods can learn to perform molecular recognition.

DeepCoy represents a novel approach to solve this problem, exhibiting substantial benefit over previous database-based methods. Our framework is highly customisable by the user and can naturally be combined with database search. While experimentally-verified inactives should be used whenever possible, this is not practically feasible apart from for limited-size benchmarking sets (e.g. Rohrer and Baumann, 2009). As such, effective decoys are crucial to the development of structure-based virtual screening methods.

There will come a time when you believe everything is finished. That will be the beginning.

— Louis L'Amour, *Lonely on the Mountain*

6

Conclusions & Future Work

Contents

Drugs are key therapeutic treatments for a wide variety of medical conditions and are an essential building block of a functioning health system (World Health Organization, 2010). However, developing new drugs is a challenging process, with each new drug on average costing $1.5-3 billion, (Avorn, 2015; DiMasi et al., 2016) and taking over ten years to develop (Paul et al., 2010). Deep learning methods have been successfully applied to a number of problems in other fields. However, such methods are typically not immediately applicable to key problems in drug discovery. This thesis has focused on developing deep learning methodologies for the pre-clinical drug discovery. In this chapter, we summarise the main contributions of this thesis and provide several directions for future work.

6.1 Molecule Generation for Hit-To-Lead and Lead Optimisation

In Chapter 2, we described DeLinker, a graph-based deep generative method for fragment linking or scaffold hopping that integrates 3D structural information, utilising the relative distance and orientation between the starting substructures in the design process. Unlike previous methods for computational fragment linking or scaffold hopping, our model does not rely on a database of fragments from which to select a linker but instead designs one given the fragments provided and 3D information.

Through two large-scale assessments, we demonstrated that DeLinker generates molecules that have high 3D similarity to both the initial fragments and the original molecules, substantially outperforming a database baseline, the previous *de facto* method. We further validated our method through three case studies, demonstrating applications to fragment linking, scaffold hopping, and PROTAC design. As far as we are aware, this is the first molecular generative model to incorporate 3D structural information directly in the design process.

However, the only 3D information utilised by the model was the distance between the fragments or starting substructures and their relative orientations. This provided explicit constraints for a given compound, but only implicit information about the shape of the binding site. Despite this minimal parametrisation, there is a substantial impact on the generated molecules.

In Chapter 3, we extended our method to incorporate physically-meaningful 3D structural information using a convolutional neural network, providing a richer prior for the generative process. In addition, we demonstrated that our method can be applied to molecular elaboration tasks, such as R-group design. In particular, we utilised 3D pharmacophores, a general and widely-used representation in cheminformatics. This representation is both physically-meaning and broadly applicable, since pharmacophores can be derived from other molecules (ligand-based) or proposed based on the protein target of interest (structure-based).

The experimental results showed that our model significantly outperformed previous methods for both linker design and scaffold elaboration. In particular, we demonstrated the power of including pharmacophoric constraints as a 3D representation as opposed to a 1D count vector. In addition, we demonstrated our method in an R-group optimisation case study. These results validate the effectiveness of our model as a tool for use in the hit-to-lead and lead optimisation stages of the drug discovery process.

There are two immediate directions for future work. First, in a typical fragment screen there are normally many hits that could be linked or elaborated. Enumerating all possible combinations, together with a set of candidate pharmacophores, quickly leads to combinatorial explosion. While a brute force approach has some merit, in particular given the speed of compound generation, this approach is clearly suboptimal and would result in either limited coverage, or a large number of generated compounds to assess. Initially, we expect medicinal chemists to propose a small number of possible combinations as input for our generative model. However, this risks missing many interesting cases that should have been prioritised. Learning which fragment hits are most likely to lead to promising candidates would be an interesting avenue to explore.

Second, alternative structural representations could be explored. Due to their prevalence and importance in drug discovery, we included hydrogen bond donors, hydrogen bond acceptors, and aromatic systems in our pharmacophoric representation. However, other pharmacophores are medicinally relevant and could be readily incorporated into our model. Further, we adopted a voxel representation of the 3D features. This choice, while practical, has limitations, such as a lack of rotational equivariance. Methods that do not require voxelisation (e.g. Schütt et al., 2017) might be better suited. The use of such approaches could also enable the 3D structure to be directly output by the generative method (e.g. Gebauer et al., 2019). Finally, the structural representation adopted represents a static view of protein-ligand binding. A method to incorporate dynamic information into the generative process would be an interesting development.

6.2 Structure-Based Virtual Screening

In Chapter 4, we described DenseFS, a CNN-based deep learning approach for structure-based virtual screening. We demonstrated substantial improvements in predictive power compared to previous work in assessments on both DUD-E and an independent test set, producing state-of-the-art results by all metrics assessed. On DUD-E, our method exhibited around a 70% improvement in AUC PRC and 0.5% ROC enrichment over the Baseline CNN of Ragoza et al. (2017), achieving a higher AUC PRC for 95 of the 102 targets. On the independent ChEMBL set, our method outperformed the Baseline CNN on 13 of the 14 targets, resulting in an average increase in AUC PRC of over 40%.

We achieved this by combining advances in machine learning with techniques to incorporate domain-specific knowledge. First, we utilised a modern CNN architecture, namely a DenseNet (Huang et al., 2016). We did not perform an extensive review of possible choices for our deep learning network and anticipate that further improvement could be obtained by applying the current state-of-the-art techniques. Adopting the DenseNet architecture resulted in the largest improvement in performance and changing this alone was responsible for a 20.8% - 36.5% increase in AUC PRC during cross-validation (30.4% - 53.4% of overall improvement). We also demonstrated that combining the predictions of three models (trained with different random seeds) in an ensemble improved average AUC PRC by 3.4% - 11.0% (5.0% - 16.1% of overall improvement). While this is a relatively minor benefit, this technique consistently improved predictions, despite using only three models in our ensemble.

We also proposed two changes to the virtual screening protocol to incorporate domain-specific knowledge. Docking introduces substantial noise to the data sets due to the inaccuracy of many docked poses. We adopted an average scoring protocol to combat this noise. This provided a 7.6% - 14.4% increase in AUC PRC (11.1% - 21.1% of overall improvement) by reducing the reliance on any single pose, and eliminating the ability for a molecule to be ranked highly from a single pose alone.

Finally, we demonstrated how transfer learning could be used to construct protein family-specific models by finetuning a universal model on subsets of the training data. Using a family-specific, as opposed to a single, universal model, allowed our models to learn different representations capturing physicochemical nuances exhibited by different families, resulting in average improvements in AUC PRC of 18.3% - 24.0% (26.7% - 35.1% of overall improvement). In addition, we demonstrated that only a limited quantity of data was required before a family-specific model outperformed a universal one, suggesting that this approach both can and should be adopted in the majority of scenarios.

Recent applications of CNNs to the scoring of protein-ligand complexes have resulted in state-of-the-art performance on existing benchmarks not only in structure-based virtual screening as described in Chapter 4, but also on pose prediction (Hochuli et al., 2018; Mahmoud et al., 2020) and binding affinity prediction (Jiménez et al., 2018). This is very promising as the application of these techniques is nascent and substantial room for improvement exists.

In addition, encouraging results in prospective experiments have begun to be reported, such as Sunseri et al. (2019) in Drug Design Data Resource (D3R) Grand Challenge 3 (Gaieb et al., 2019), where a CNN-based method provided best-in-class performance on several virtual screening tasks, and Stecula et al. (2020), who identified five low-micromolar inhibitors targeting aspartate N-acetyltransferase with a CNN-based approach despite the lack of a protein structure or high sequence identity homologous templates. However, the true potential of CNNs will only be understood following numerous real-world prospective screens in live drug discovery projects and further methodological advances. We believe that the development of CNNs for protein-ligand scoring will rely on integrating domain-specific knowledge with the advances in computer vision (Imrie et al., 2021a).

As discussed in Chapter 1, the prominence of machine learning methods in cheminformatics will necessitate new and improved training and benchmarking sets in order to train more powerful models. The datasets available for structure-based methods remain small (c. 23,000 actives across 102 targets in DUD-E; around

18,000 protein-ligand complexes in PDBbind v.2019) compared to computer-vision datasets (up to 300 million examples). Our findings support this and highlight the need for additional data, even in data-rich protein families. Continuing to add further examples provided appreciable benefit, even for the largest family present in the training set. This suggests that CNN-based methods will continue to learn from larger training sets. While we have not addressed the challenge of proposing a larger dataset in this thesis, this is a key area for future work.

In Chapter 5, we proposed a deep learning method, DeepCoy, that generates decoys to a user's preferred specification in order to remove such biases or construct sets with a defined bias. Unlike almost all virtual screening benchmarks, our method does not rely on a database molecules from which to select decoys but instead designs ones that are tailored to the active molecule. We validated DeepCoy using two established benchmarks, DUD-E and DEKOIS 2.0. For all 102 DUD-E targets and 80 of the 81 DEKOIS 2.0 targets, our generated decoy molecules more closely matched the active molecules' physicochemical properties while introducing no discernible additional risk of false negatives. The DeepCoy decoys improved the Deviation from Optimal Embedding (DOE) score by an average of 81% and 66%, respectively. Furthermore, we validated that our generated decoys are no easier to distinguish than the original decoy molecules via docking with Autodock Vina and demonstrated that CNN-based methods, such as those discussed in Chapter 4, can exhibit substantial predictive power when trained on DeepCoy decoys.

The pipeline used by DeepCoy (Imrie et al., 2021b) to construct decoy molecules largely follows existing methodology, with the crucial exception of the introduction of our learning-based framework to replace a database search. While our work has greatly improved the ability to meet the stated desiderata of decoy molecules, there has been limited change to the broader pipeline for constructing molecular decoys since property matching was introduced by Huang et al. (2006) almost 15 years ago.

Substantial improvements in benchmarking, and in particular training sets for machine learning models, are likely to require developments in the assumptions underlying decoy construction. An example of a recent such development was

the extension of the DUD-E methodology to match the 3D shape and charge distribution by Adeshina et al. (2020). Designing decoy molecules specifically to address weaknesses of models during training or probe methods in a rigorous assessment could be an exciting avenue to explore. Such a setting could be made feasible by a learning-based framework such as DeepCoy.

6.3 Closing Remarks

Despite their importance as therapeutic treatments, current methods for drug discovery have frequently been called unsustainable (Moors et al., 2014; Ernst & Young, 2017) due to the cost and length of time required to develop new treatments. Deep learning offers the potential to transform the drug discovery process; however, existing techniques are often not directly applicable to drug discovery. In this thesis, we have proposed several novel deep learning-based methods encompassing a range of challenges in preclinical drug discovery. Our methods encompass both generative (Chapters 2 and 3) and predictive modelling (Chapter 4), as well as proposing a generative method to improve predictive modelling (Chapter 5). The generative models described in Chapters 2 and 3 are applicable within both ligand- and structure-based paradigms. Finally, the proposed methods cover almost the full spectrum of preclinical drug discovery, namely hit finding (Chapters 4 and 5), hit-to-lead (Chapter 2), and lead optimisation (Chapter 3), with an emphasis on the discovery of potent bioactive inhibitors. Despite the rapid advances over the past few years, deep learning methods and applications to drug discovery remain nascent. Future development of computational methods and advances in automation promises to further transform drug discovery.

Appendices

A

Deep Generative Models for 3D Linker Design

Contents

A.1 DeLinker Implementation Details

A.1.1 Atom Types

There are 14 permitted atom types: carbon, nitrogen (N^-, N, N^+), oxygen (O^-, O, O^+), fluorine, chlorine, bromine, iodine, and sulphur (maximum valence 2, 4, or 6).

A.1.2 Network Architecture

Following Liu et al. (2018), both the encoder and decoder are standard gated graph neural networks (GGNN, Li et al., 2016), which propagate messages for 7 steps, and have residual connections between odd numbered time steps.

We implemented the function f, which maps the hidden state of a node to its atom type, as a linear classifier with attention from the node's hidden vector to one of the node types. The attention mechanism is similar to Bahdanau et al. (2015) and allows the label for a given node to depend on the hidden states of the other expansion nodes.

Similarly, we augmented the edge selection and edge labelling step by adding attention between the feature vectors for all candidate edges. This allows the score of a candidate edge to depend on the other possible edges.

We trained the model with a learning rate of 0.001 for 10 epochs using the Adam optimiser.

A.1.3 Hyperparameter Search.

We performed a limited hyperparameter search of the following parameters (final parameters in bold):

- Learning rate: 0.01, **0.001**, 0.0001
- Batch size: 8, **16**
- Hidden state dimension: 16, **32**, 50, 100
- Encoding dimension: 2, **4**, 8
- λ_{KL}: 0.1, **0.3**, 0.6

The model was fairly robust to the choice of hyperparameters. Performance was measured via the validation reconstruction loss and not generative performance.

A.2 Data Curation

Fragment-molecule pairs for the ZINC (Sterling and Irwin, 2015) and CASF (Su et al., 2019) sets were constructed as follows. First all possible fragmentations of each molecule were produced by enumerating all double acyclic single bond cuts (Hussain and Rea, 2010). These we then filtered to remove trivial and unrealistic

situations using the following constraints: (i) minimum linker length: 3 atoms, (ii) minimum fragment size: 5 atoms, (iii) linker fewer heavy atoms than either fragment, (iv) minimum path length between fragments: 2 atoms.

The remaining fragment-molecule pairs were filtered for several 2D properties: (i) the synthetic accessibility (SA) score (Ertl and Schuffenhauer, 2009) of the molecule must be lower than the fragments with exit vectors represented by dummy atoms, (ii) the molecule must pass pan-assay interference (PAINS, (Baell and Holloway, 2010)) filters, and (iii) rings must either be saturated aliphatic or aromatic (according to RDKit valency rules). PAINS filters were implemented by SMARTS substructure searching with RDKit (Landrum, 2006), using the RDKit version of the Saubern et al. (2011) translation of the original PAINS (Baell and Holloway, 2010) as specified at `https://github.com/rdkit/rdkit/blob/master/Data/Pains/wehi_pains.csv` (Accessed: 02/06/2019). Any molecules containing atom types outside of the permitted atom types were excluded.

A.3 Training Set Composition

Table A.1: Distribution of number of atoms contained in the original linkers in the datasets utilised. The average linker length in the CASF set (5.9) is around one atom longer than the ZINC training set (4.7), validation (4.7) and test set (4.9).

Linker	ZINC			CASF
Length	Train	Valid	Test	Test
3	28.7%	30.7%	26.2%	25.2%
4	21.6%	22.7%	17.7%	13.9%
5	19.5%	15.7%	22.0%	10.7%
6	17.8%	16.5%	21.0%	13.3%
7	7.7%	10.3%	7.3%	12.0%
8	3.1%	2.5%	3.5%	9.1%
9	1.3%	1.3%	2.0%	4.8%
10	0.3%	0.3%	0.3%	3.2%
11	0.0%	-	-	5.5%
$\geq$12	0.0%	-	-	2.3%

A.4 Additional Results

Table A.2: Ablation study for DeLinker, our deep generative method on the ZINC data set. We show the effect on the 2D metrics of removing all of the structural information ("No info") and including only the distance information ("Distance") compared to our full protocol ("DeLinker"), the database baseline ("Database"), and a graph-based baseline ("CGVAE", Liu et al., 2018). See Section A.2 for a description of the 2D property filters.

Metric	Database	CGVAE	No Info	Distance	DeLinker
Valid	100.0%	88.9%	97.0%	98.6%	98.4%
Unique	38.8%	58.8%	51.2%	47.3%	44.2%
Novel	0.0%	51.0%	36.2%	37.6%	39.5%
Recovered	78.0%	65.8%	74.5%	78.3%	79.0%
Pass 2D filters	97.0%	85.9%	89.9%	90.2%	89.8%
Pass SA filter	97.8%	90.0%	95.1%	95.5%	95.3%
Pass ring filter	100.0%	93.2%	95.2%	94.5%	94.8%
Pass PAINS filter	99.2%	96.1%	97.8%	98.4%	97.9%

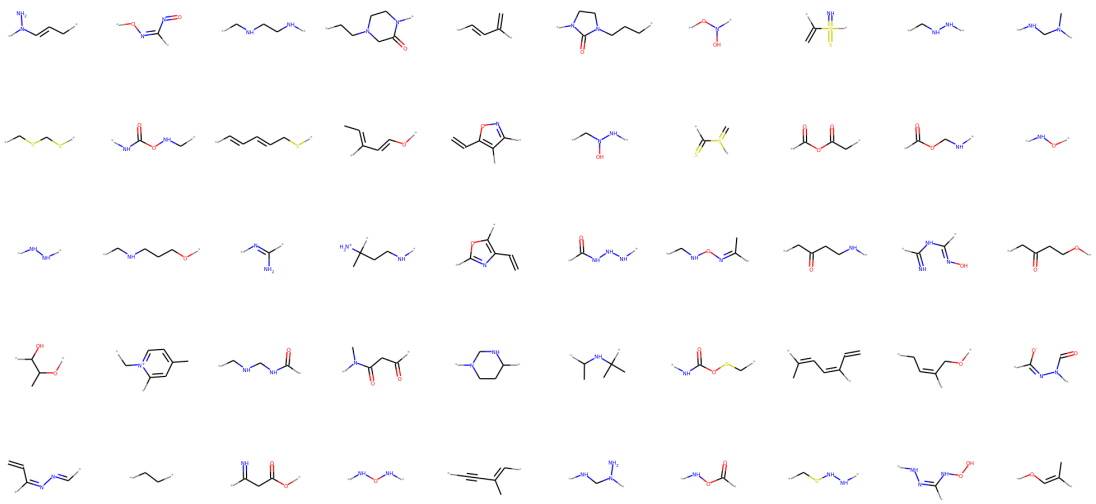


Figure A.1: A random sample of 50 novel linkers generated by DeLinker during testing on the held-out ZINC data set.

Table A.3: Ablation study for DeLinker, our deep generative method, on the ZINC data set. We show the effect on the 3D metrics of removing all of the structural information ("No info") and including only the distance information ("Distance") compared to our full protocol ("DeLinker") that includes both distance and angle information, the database baseline ("Database"), and a graph-based baseline ("CGVAE", Liu et al., 2018). See Section 2.3.3 for a description of the metrics.

Metric	Database	CGVAE	No Info	Distance	DeLinker
SC_{RDKit} Molecule					
>0.7	35.5%	35.4%	37.6%	43.2%	47.1%
>0.8	8.5%	7.2%	9.2%	11.8%	14.2%
>0.9	1.3%	0.7%	1.1%	1.5%	1.8%
SC_{RDKit} Fragments					
>0.7	60.2%	64.1%	64.4%	69.1%	71.3%
>0.8	24.7%	26.3%	27.7%	33.4%	35.8%
>0.9	4.5%	4.2%	5.0%	7.0%	8.2%
RMSD Fragments					
<1.00	46.9%	51.0%	50.9%	56.6%	58.6%
<0.75	20.5%	21.6%	22.4%	27.8%	30.0%
<0.50	5.7%	4.8%	5.6%	7.9%	9.3%

Table A.4: 2D and 3D metrics for molecules generated by DeLinker, our *de novo* deep generative model, compared to a Database baseline on the held-out ZINC test set. See Section A.2 for a description of the 2D property filters and Section 2.3.3 for a description of the 3D metrics.

	ZINC		ZINC $\geq$ 5 atoms	
Metric	Database	DeLinker	Database	DeLinker
Valid	100.0%	98.4%	100.0%	98.1%
Unique	38.8%	44.2%	53.6%	61.0%
Novel	0.0%	39.5%	0.0%	49.4%
Recovered	78.0%	79.0%	67.0%	67.0%
Pass 2D filters	97.0%	89.8%	96.4%	84.1%
SC_{RDKit} Molecule				
>0.7	33.5%	47.1%	21.3%	37.1%
>0.8	8.5%	14.2%	3.5%	9.4%
>0.9	1.3%	1.8%	0.4%	1.0%
SC_{RDKit} Fragments				
>0.7	60.2%	71.3%	51.5%	66.7%
>0.8	24.7%	35.8%	16.8%	30.3%
>0.9	4.5%	8.2%	2.1%	6.0%
RMSD Fragments				
<1.00Å	46.9%	58.6%	39.1%	55.1%
<0.75Å	20.5%	30.0%	14.2%	26.9%
<0.50Å	5.7%	9.3%	3.0%	6.9%

Table A.5: Fragment linking case study. 2D and 3D Metrics for DeLinker and the Database baseline.

Metric	Database	DeLinker
Valid	100.0%	98.7%
Unique	30.7%	56.4%
Novel	0.0%	58.5%
Recovered	100.0%	100.0%
Pass 2D filters	97.8%	74.0%
SC_{RDKit} Fragments		
>0.7	681	1115
>0.8	129	301
>0.9	6	18

Table A.6: Scaffold hopping case study. 2D and 3D metrics for DeLinker. Compounds with SC_{RDKit} Fragments>0.80 were docked with AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013).

Metric	DeLinker
Valid	99.5%
Unique	63.7%
Novel	88.5%
Recovered	100.0%
Pass 2D filters	51.4%
SC_{RDKit} Fragments	
>0.70	1928
>0.75	699
>0.80	114
>0.85	9
Vina Score	
<-7	105
<-8	69
<Aminopyrazole	33
<-9	26
<-10	3
<Indazole	0

Table A.7: PROTAC design case study. 2D and 3D metrics for DeLinker. Compounds with SC_{RDKit} Fragments>0.80 were docked with AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013).

Metric	DeLinker
Valid	96.0%
Unique	62.1%
Novel	95.9%
Recovered	0.0%
Pass 2D filters	61.8%
SC_{RDKit} Fragments	
>0.70	2930
>0.80	2930
>0.85	2150
Vina Score	
<-12	2930
<PROTAC 1	2927
<-13	2845
<-14	1160
<PROTAC 2	536
<-15	34

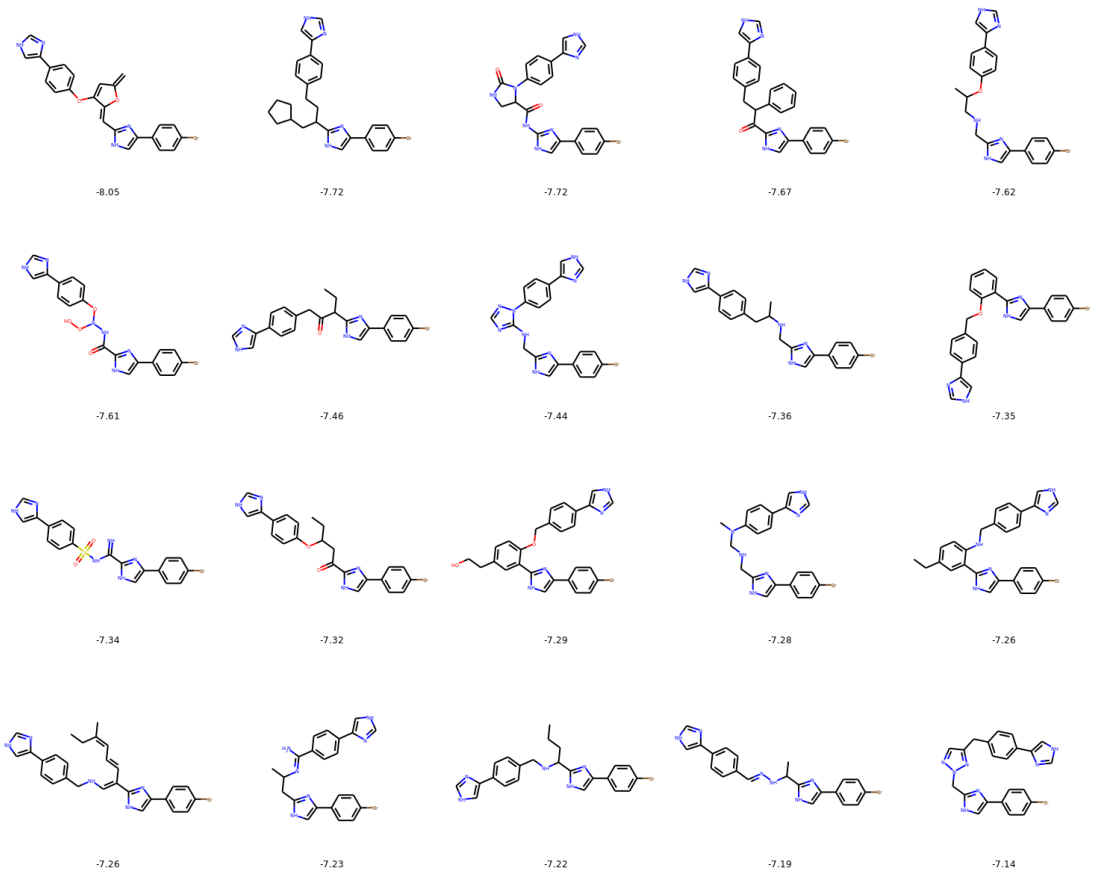


Figure A.2: Fragment linking case study. The top 20 molecules generated by DeLinker that met the 3D similarity threshold ranked by AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013) score. Labels are the docking score from minimizing the aligned molecular conformer according to the Vina energy function.

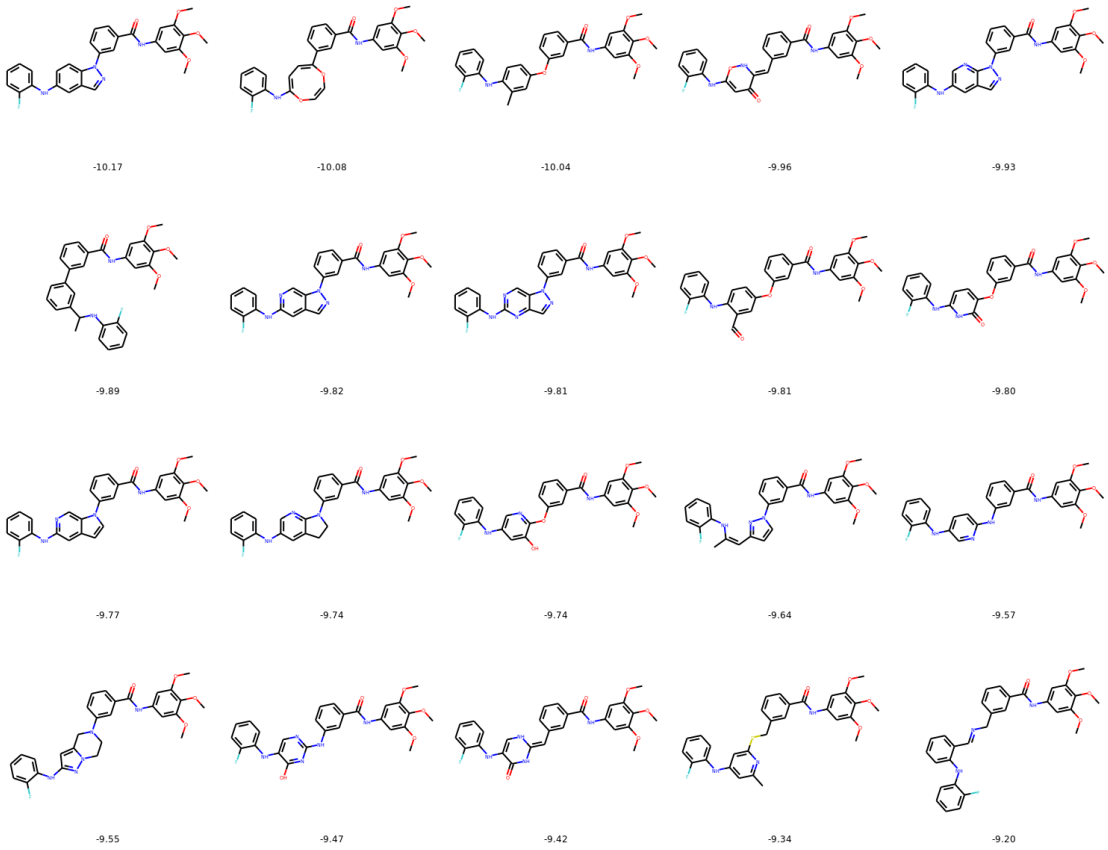


Figure A.3: Scaffold hopping case study. The top 20 molecules generated by DeLinker that met the 3D similarity threshold ranked by AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013) score. Labels are the docking score from minimizing the aligned molecular conformer according to the Vina energy function.

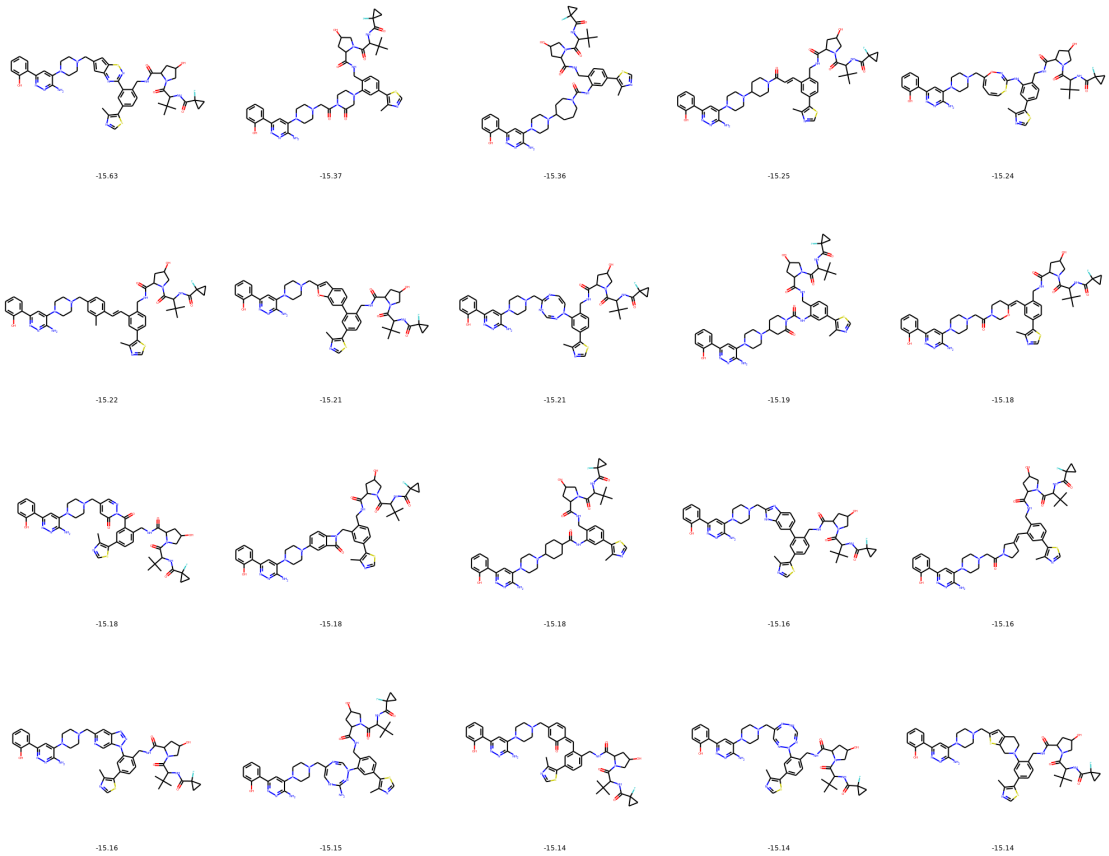


Figure A.4: PROTAC design case study. The top 20 molecules generated by DeLinker that met the 3D similarity threshold ranked by AutoDock Vina (Trott and Olson, 2010; Koes et al., 2013) score. Labels are the docking score from minimizing the aligned molecular conformer according to the Vina energy function.

B

Deep Generative Design with 3D Pharmacophoric Constraints

Contents

B.1 Comparison to SyntaLinker

SyntaLinker (Yang et al., 2020b) is a transformer-based model for linker design that utilises a SMILES-based representation. The results in Table B.1 are not directly comparable with the original publication since SyntaLinker was both trained and evaluated on different data sets in Yang et al. (2020b). In Chapter 3, we ensured a fair comparison between all methods by training and assessing all methods on the same data sets.

SyntaLinker produced substantially weaker results than were reported in its original publication (Yang et al., 2020b). In particular, SyntaLinker produced a very low proportion of valid molecules on both the CASF (Table B.1) and PDBbind (Table 3.2) test sets. The poor validity of generated molecules is largely due to the sampling method used by SyntaLinker, which employs beam search (Lowerre, 1976)

to generate SMILES strings. When sampling only the most likely sequences, the validity of generated molecules is relatively high; when the number of sequences sampled is increased, the validity falls significantly. In addition, we note that this sampling procedure limits the number of molecules that can be generated by SyntaLinker. However, as a result of using beam search, almost all of the SMILES strings generated by SyntaLinker correspond to unique molecules.

Furthermore, on the CASF set SyntaLinker recovered 8% of the original molecules compared to 30% for DeLinker and 50% for DeLinker-3D, while on the PDBbind set SyntaLinker only recovered 0.3% of the original molecules compared to 2.3% and 22.2% for DeLinker and DeLinker-3D, respectively. The 3D shape similarity of the molecules generated by SyntaLinker was significantly lower than DeLinker-3D for both the CASF (Table B.1) and PDBbind (Table 3.2) test sets.

B.2 Additional Results

Table B.1: CASF set results for linkers with ≥5 atoms.

Metric	SyntaLinker	DeLinker	DeLinker-Counts	DeLinker-3D
Valid	9.8%	94.7%	86.0%	89.6%
Unique	95.1%	72.9%	58.6%	58.2%
Novel	54.4%	68.7%	68.4%	71.1%
Recovered	7.5%	29.8%	41.5%	50.0%
Pass 2D filters	80.5%	71.7%	71.7%	68.6%
SC_{RDKit} Generated				
>0.6	13.9%	25.0%	44.7%	53.7%
>0.7	7.5%	14.3%	32.7%	39.5%
>0.8	4.6%	7.9%	22.3%	26.3%
>0.9	2.1%	3.2%	12.0%	14.4%

Table B.2: Linker Design. Alternative 3D similarity metrics for CASF test set for linkers with ≥5 atoms.

Metric	SyntaLinker	DeLinker	DeLinker-Counts	DeLinker-3D
SC_{RDKit} Molecule				
>0.7	13.9%	16.3%	22.5%	25.5%
>0.8	3.1%	3.6%	7.0%	7.2%
>0.9	0.2%	0.8%	2.9%	3.0%
SC_{RDKit} Fragments				
>0.7	39.4%	38.7%	40.8%	43.0%
>0.8	13.8%	12.3%	14.7%	15.4%
>0.9	1.5%	1.6%	3.2%	3.8%
RMSD Fragments				
<1.00Å	29.4%	26.6%	28.1%	30.7%
<0.75Å	11.1%	9.3%	10.6%	12.9%
<0.50Å	2.0%	2.4%	4.3%	4.7%

Table B.3: Linker Design. Alternative 3D similarity metrics for PDBbind test set.

Metric	SyntaLinker	DeLinker	DeLinker-Counts	DeLinker-3D
SC_{RDKit} Molecule				
>0.7	12.3%	12.3%	14.7%	15.9%
>0.8	2.5%	1.9%	3.2%	3.7%
>0.9	0.1%	0.0%	0.2%	0.2%
SC_{RDKit} Fragments				
>0.7	34.8%	32.1%	33.1%	31.6%
>0.8	10.5%	9.4%	9.6%	9.8%
>0.9	2.0%	1.1%	0.8%	1.2%
RMSD Fragments				
<1.00Å	26.6%	23.1%	24.3%	23.4%
<0.75Å	11.0%	7.8%	8.1%	8.0%
<0.50Å	3.1%	1.3%	1.2%	1.4%

Table B.4: Scaffold elaboration. CASF set results.

Metric	REINVENT	DeLinker	DeLinker-Counts	DeLinker-3D
Valid	99.9%	99.9%	99.8%	99.1%
Unique	27.4%	56.6%	39.1%	29.1%
Novel	3.2%	43.4%	40.9%	34.5%
Recovered	25.3%	47.3%	59.9%	68.8%
Pass 2D filters	98.2%	72.6%	75.1%	80.0%
SC_{RDKit} Generated				
>0.6	16.0%	13.7%	27.6%	43.9%
>0.7	9.3%	7.6%	19.6%	32.3%
>0.8	5.2%	4.0%	12.9%	20.9%
>0.9	1.8%	2.0%	6.0%	9.4%

C

Structure-Based Virtual Screening with Convolutional Neural Networks

Contents

C.1 Example of a Failure Case of Docking

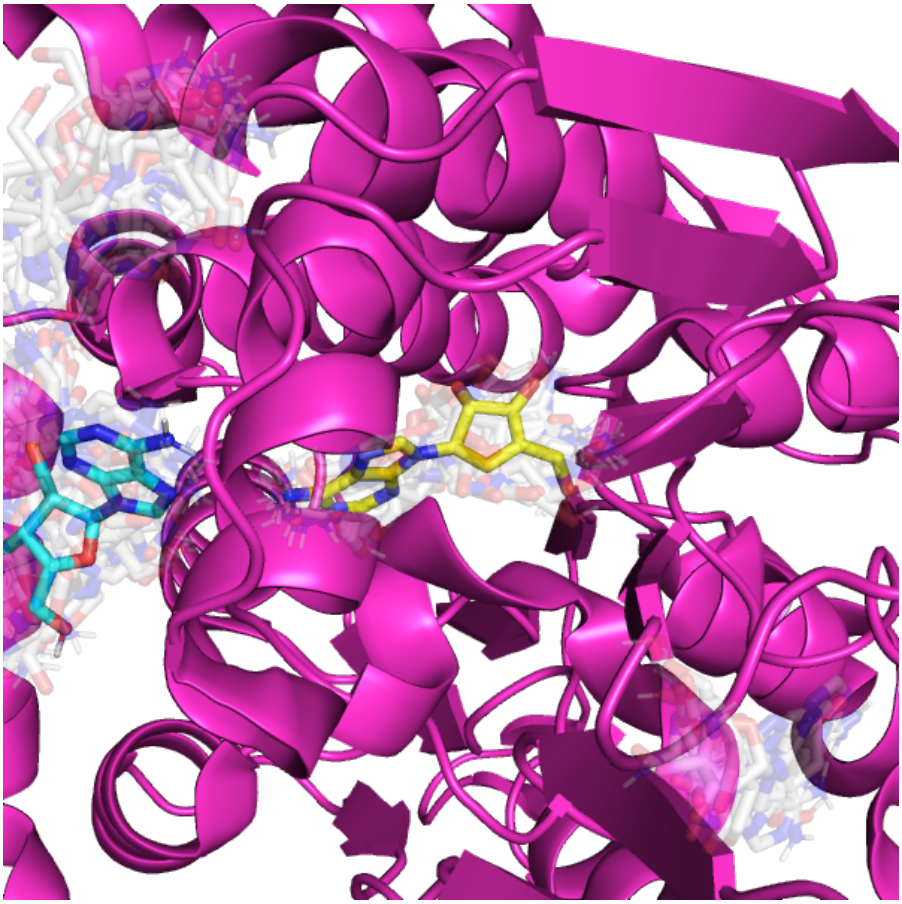


Figure C.1: Docked poses of the ligand CHEMBL300406 into DUD-E target SAHH. The highest ranked pose generated by AutoDock Vina is shown with carbons in cyan, while the crystal pose (PDB ID 1LI4) is shown with carbons in yellow. Other docked poses are shown with carbons in white. The heteroatoms for all poses are coloured with standard colouring. The inaccuracy of many of the docked poses is evident, and there is a substantial difference of binding modes between the highest ranked Vina pose and the crystal structure.

C.2 Additional Results

Table C.1: Mean AUC ROC, AUC PRC and ROC enrichment across kinase targets in the DUD-E dataset for our methods, DenseFS and DenseU, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseU	DenseFS
AUC ROC	0.751	0.937	0.963	0.977
AUC PRC	0.116	0.505	0.607	0.712
0.5% EF	20.148	97.889	116.215	136.355
1% EF	13.160	59.530	67.033	75.831
2% EF	8.824	33.967	37.650	41.383
5% EF	5.649	15.570	16.868	17.960

Table C.2: Mean AUC ROC, AUC PRC and ROC enrichment across protease targets in the DUD-E dataset for our methods, DenseFS and DenseU, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseU	DenseFS
AUC ROC	0.711	0.893	0.942	0.957
AUC PRC	0.080	0.252	0.453	0.590
0.5% EF	10.242	38.699	77.665	108.453
1% EF	7.513	28.289	48.961	61.282
2% EF	6.149	19.283	29.525	34.251
5% EF	4.601	10.991	14.580	15.791

Table C.3: Mean AUC ROC, AUC PRC and ROC enrichment across nuclear targets in the DUD-E dataset for our methods, DenseFS and DenseU, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseU	DenseFS
AUC ROC	0.720	0.872	0.919	0.948
AUC PRC	0.129	0.140	0.275	0.449
0.5% EF	21.833	14.661	39.792	72.708
1% EF	14.747	13.351	29.845	48.092
2% EF	9.669	10.920	20.818	30.112
5% EF	5.861	7.889	12.095	15.055

Table C.4: Mean AUC ROC, AUC PRC and ROC enrichment across GPCR targets in the DUD-E dataset for our methods, DenseFS and DenseU, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseU	DenseFS
AUC ROC	0.663	0.840	0.888	0.888
AUC PRC	0.032	0.125	0.171	0.179
0.5% EF	3.033	17.098	24.919	24.531
1% EF	2.920	15.565	20.111	22.493
2% EF	2.688	13.703	15.919	17.651
5% EF	2.522	9.155	11.350	11.575

Table C.5: Mean AUC ROC, AUC PRC and ROC enrichment across other targets in the DUD-E dataset for our methods, DenseFS and DenseU, compared to the Baseline CNN and the AutoDock Vina scoring function.

Metric	Vina	Baseline CNN	DenseU	DenseFS
AUC ROC	0.673	0.808	0.854	0.865
AUC PRC	0.081	0.171	0.247	0.266
0.5% EF	13.310	25.972	41.549	44.362
1% EF	9.499	20.660	28.176	30.273
2% EF	6.362	14.463	18.406	19.742
5% EF	4.203	8.411	9.961	10.545

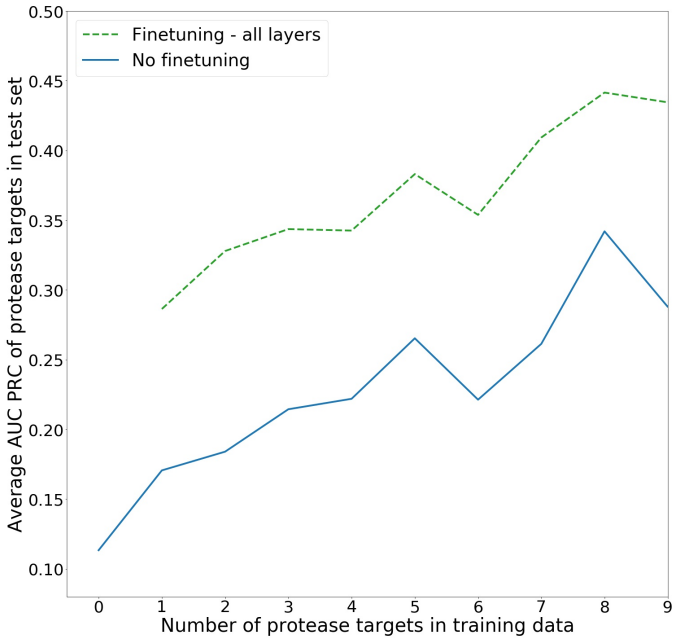


Figure C.2: Average AUC PRC of protease targets for varying number of protease in the training set, comparing finetuning all layers to no finetuning.

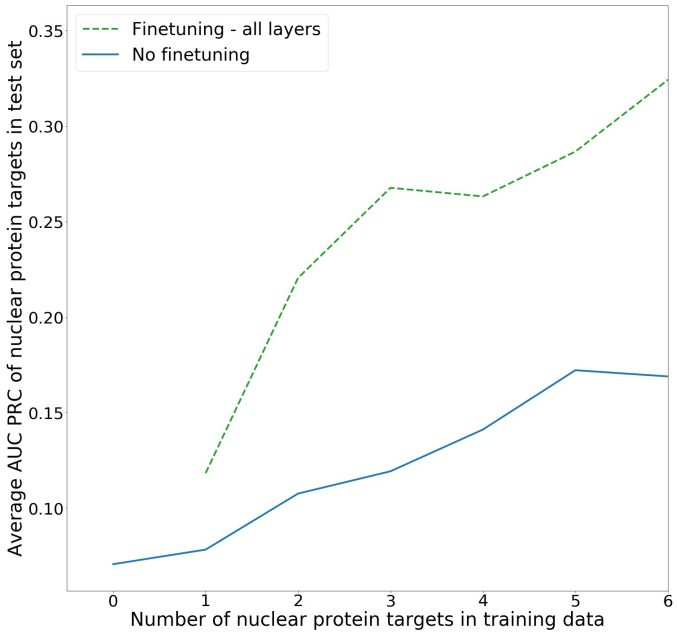


Figure C.3: Average AUC PRC of nuclear targets for varying number of nuclear proteins in the training set, comparing finetuning all layers to no finetuning.

Table C.6: Cross-validation DUD-E AUC ROC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	0.636	0.905	**0.932**	0.908
abl1	0.770	0.934	0.966	**0.985**
ace	0.521	0.739	0.942	**0.969**
aces	0.764	0.834	0.828	**0.838**
ada	0.536	0.887	0.912	**0.925**
ada17	0.694	0.937	0.966	**0.983**
adrb1	0.709	0.900	0.942	**0.947**
adrb2	0.698	0.886	0.939	**0.941**
akt1	0.753	0.974	0.985	**0.989**
akt2	0.780	0.982	0.986	**0.994**
aldr	**0.726**	0.609	0.681	0.688
ampc	0.595	0.521	**0.623**	0.616
andr	0.603	0.789	0.849	**0.928**
aofb	**0.768**	0.611	0.711	0.725
bace1	0.703	0.858	**0.937**	0.930
braf	0.852	0.983	0.991	**0.997**
cah2	0.565	0.487	**0.573**	0.567
casp3	0.643	0.813	0.883	**0.914**
cdk2	0.686	0.858	0.927	**0.949**
comt	0.607	0.736	0.832	**0.853**
cp2c9	0.611	0.886	0.911	**0.925**
cp3a4	0.590	0.899	0.924	**0.938**
csf1r	0.669	0.931	0.968	**0.979**
cxcr4	0.535	0.725	0.750	**0.760**
def	0.735	0.929	0.943	**0.964**
dhi1	**0.762**	0.686	0.682	0.701
dpp4	0.608	0.796	0.811	**0.849**
drd3	0.738	0.785	0.878	**0.882**
dyr	0.741	0.877	0.935	**0.943**
egfr	0.630	0.968	0.976	**0.985**
esr1	0.802	0.903	0.943	**0.951**
esr2	0.771	0.902	**0.948**	0.940
fa10	0.823	0.905	0.938	**0.958**
fa7	0.898	0.965	0.980	**0.989**
fabp4	0.784	0.789	0.840	**0.846**
fak1	0.802	0.970	0.984	**0.996**
fgfr1	0.689	0.968	0.982	**0.990**
fkb1a	0.690	0.660	0.784	**0.796**
fnta	0.615	0.941	0.965	**0.970**
fpps	0.234	0.970	0.992	**0.995**
gcr	0.589	0.867	0.913	**0.923**
glcm	0.448	**0.706**	0.594	0.608
gria2	0.735	0.828	0.889	**0.906**
grik1	0.585	0.736	0.871	**0.881**
hdac2	0.843	0.918	0.959	**0.965**
hdac8	0.809	0.954	0.986	**0.988**
hivint	0.683	0.890	0.938	**0.946**
hivpr	0.678	0.790	0.857	**0.888**
hivrt	0.658	0.715	0.759	**0.768**
hmdh	0.757	0.927	0.961	**0.966**
hs90a	0.258	**0.920**	0.884	0.893

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	0.544	0.770	0.926	**0.932**
igf1r	0.814	0.973	0.979	**0.985**
inha	0.695	**0.859**	0.845	0.858
ital	0.575	0.895	0.938	**0.947**
jak2	0.754	0.992	0.994	**0.998**
kif11	**0.846**	0.741	0.728	0.759
kit	0.765	0.923	0.951	**0.980**
kith	0.673	0.800	0.933	**0.938**
kpcb	0.757	0.813	0.895	**0.906**
lck	0.781	0.943	0.969	**0.991**
lkha4	0.889	0.955	**0.966**	0.937
mapk2	0.852	0.851	0.944	**0.961**
mcr	0.514	0.797	0.846	**0.923**
met	0.797	0.978	0.983	**0.994**
mk01	0.834	0.923	0.960	**0.978**
mk10	0.721	0.907	0.932	**0.961**
mk14	0.722	0.918	0.953	**0.965**
mmp13	0.630	0.957	0.985	**0.993**
mp2k1	0.527	0.784	0.847	**0.890**
nos1	0.549	0.691	0.820	**0.822**
nram	0.506	0.619	0.815	**0.828**
pa2ga	0.615	0.806	0.830	**0.842**
parp1	0.852	0.863	0.886	**0.900**
pde5a	0.650	0.885	0.929	**0.942**
pgh1	0.631	0.718	0.772	**0.782**
pgh2	0.764	0.802	0.864	**0.877**
plk1	0.628	0.934	0.951	**0.961**
pnph	0.857	0.907	0.904	**0.924**
ppara	0.857	0.926	0.965	**0.988**
ppard	0.756	0.913	0.960	**0.984**
pparg	0.777	0.931	0.961	**0.969**
prgr	0.636	0.836	0.881	**0.929**
ptn1	0.828	0.883	0.919	**0.936**
pur2	0.860	0.958	0.957	**0.966**
pygm	0.590	0.703	0.740	**0.749**
pyrd	0.824	0.885	0.889	**0.903**
reni	0.620	0.943	0.974	**0.986**
rock1	0.701	0.955	0.975	**0.985**
rxra	0.802	0.887	0.955	**0.977**
sahh	0.745	**0.989**	0.954	0.977
src	0.629	0.960	0.975	**0.986**
tgfr1	0.894	0.999	0.998	**1.000**
thb	0.812	0.838	0.883	**0.911**
thrb	0.737	0.904	0.948	**0.978**
try1	0.775	0.956	0.984	**0.996**
tryb1	0.702	0.932	0.967	**0.994**
tysy	0.851	0.959	0.973	**0.980**
urok	0.747	0.948	0.989	**0.994**
vgfr2	0.755	0.962	0.980	**0.993**
wee1	0.957	0.984	0.991	**0.993**
xiap	0.721	0.723	0.849	**0.863**

Table C.7: Cross-validation DUD-E AUC PRC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	0.025	0.143	0.206	**0.229**
abl1	0.105	0.671	0.760	**0.862**
ace	0.017	0.051	0.286	**0.534**
aces	**0.113**	0.077	0.073	0.080
ada	0.021	0.202	0.380	**0.490**
ada17	0.246	0.409	0.785	**0.875**
adrb1	0.038	0.170	0.197	**0.210**
adrb2	0.045	0.199	0.244	**0.270**
akt1	0.054	0.623	0.594	**0.742**
akt2	0.201	0.621	0.596	**0.723**
aldr	0.063	0.043	**0.067**	0.065
ampc	0.024	0.028	**0.065**	0.064
andr	0.139	0.057	0.105	**0.247**
aofb	**0.071**	0.032	0.059	0.061
bace1	0.035	0.079	0.183	**0.217**
braf	0.127	0.690	0.778	**0.900**
cah2	0.017	0.016	0.028	**0.029**
casp3	0.030	0.072	0.158	**0.432**
cdk2	0.067	0.250	0.417	**0.483**
comt	0.019	0.033	**0.049**	0.048
cp2c9	0.026	0.284	0.320	**0.356**
cp3a4	0.020	0.299	0.333	**0.339**
csf1r	0.024	0.328	0.510	**0.575**
cxcr4	0.012	**0.062**	0.061	0.056
def	0.065	0.397	**0.552**	0.548
dhi1	**0.046**	0.039	0.033	0.032
dpp4	0.017	0.053	0.088	**0.120**
drd3	0.039	0.050	**0.145**	0.131
dyr	0.049	0.267	0.413	**0.449**
egfr	0.032	0.665	0.749	**0.842**
esr1	0.184	0.172	0.336	**0.500**
esr2	0.109	0.161	0.333	**0.524**
fa10	0.224	0.438	0.561	**0.635**
fa7	0.162	0.325	0.678	**0.826**
fabp4	**0.233**	0.143	0.151	0.140
fak1	0.213	0.653	0.743	**0.853**
fgfr1	0.053	0.423	0.624	**0.752**
fkb1a	0.032	0.030	0.062	**0.069**
fnta	0.017	0.268	0.578	**0.623**
fpps	0.006	0.183	0.511	**0.590**
gcr	0.078	0.070	0.131	**0.199**
glcm	0.012	**0.030**	0.026	0.025
gria2	0.066	0.051	0.102	**0.109**
grik1	0.026	0.091	0.210	**0.237**
hdac2	0.125	0.700	0.745	**0.759**
hdac8	0.161	0.671	0.885	**0.894**
hivint	0.026	0.198	0.288	**0.319**
hivpr	0.029	0.055	0.070	**0.098**
hivrt	0.037	0.078	0.088	**0.097**
hmdh	0.050	0.196	0.371	**0.391**
hs90a	0.011	0.139	0.189	**0.206**

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	0.023	0.335	0.449	**0.476**
igf1r	0.113	0.744	0.793	**0.895**
inha	0.070	0.182	0.156	**0.188**
ital	0.018	0.131	0.268	**0.287**
jak2	0.160	0.844	0.844	**0.910**
kif11	**0.250**	0.038	0.042	0.045
kit	0.041	0.275	0.323	**0.503**
kith	**0.331**	0.117	0.262	0.271
kpcb	**0.289**	0.060	0.148	0.164
lck	0.066	0.558	0.698	**0.828**
lkha4	0.151	0.333	**0.370**	0.319
mapk2	0.129	0.278	0.407	**0.564**
mcr	0.056	0.046	0.088	**0.241**
met	0.092	0.684	0.753	**0.866**
mk01	0.074	0.385	0.604	**0.753**
mk10	0.052	0.425	0.515	**0.598**
mk14	0.053	0.321	0.461	**0.586**
mmp13	0.029	0.397	0.786	**0.895**
mp2k1	0.014	0.233	0.368	**0.455**
nos1	0.016	0.068	0.229	**0.255**
nram	0.014	0.021	**0.088**	0.084
pa2ga	0.024	0.069	**0.100**	0.091
parp1	0.129	0.099	0.116	**0.134**
pde5a	0.050	0.220	0.392	**0.423**
pgh1	0.041	0.067	0.109	**0.121**
pgh2	**0.259**	0.117	0.197	0.228
plk1	0.027	0.379	0.565	**0.656**
pnph	0.093	0.174	0.216	**0.225**
ppara	0.089	0.260	0.449	**0.711**
ppard	0.043	0.210	0.462	**0.697**
pparg	0.058	0.308	0.520	**0.738**
prgr	0.070	0.069	0.124	**0.190**
ptn1	0.273	0.221	0.328	**0.353**
pur2	0.076	0.208	0.375	**0.448**
pygm	0.026	0.068	0.067	**0.073**
pyrd	0.197	**0.299**	0.208	0.219
reni	0.056	0.141	0.449	**0.610**
rock1	0.041	0.380	0.486	**0.727**
rxra	0.334	0.083	0.279	**0.543**
sahh	0.153	**0.481**	0.282	0.319
src	0.026	0.635	0.759	**0.853**
tgfr1	0.115	0.942	0.930	**0.985**
thb	0.262	0.101	0.203	**0.346**
thrb	0.041	0.205	0.446	**0.667**
try1	0.045	0.489	0.685	**0.912**
tryb1	0.073	0.229	0.462	**0.811**
tysy	0.196	0.268	0.569	**0.656**
urok	0.049	0.507	0.784	**0.894**
vgfr2	0.127	0.562	0.701	**0.802**
wee1	**0.720**	0.512	0.667	0.642
xiap	0.059	0.034	0.071	**0.071**

Table C.8: Cross-validation DUD-E 0.5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	2.075	16.459	32.780	**43.568**
abl1	18.785	127.072	146.225	**165.746**
ace	1.429	7.619	37.143	**90.714**
aces	**21.634**	5.004	6.475	6.181
ada	2.151	22.222	74.552	**101.075**
ada17	53.759	71.303	154.511	**163.534**
adrb1	1.619	**31.309**	25.911	28.340
adrb2	5.195	**31.169**	28.283	28.571
akt1	3.413	128.100	107.622	**145.392**
akt2	42.735	104.274	104.274	**145.299**
aldr	**13.836**	7.547	12.998	12.579
ampc	0.000	1.389	9.722	**12.500**
andr	26.766	2.726	11.896	**31.970**
aofb	8.197	3.279	**8.197**	**8.197**
bace1	4.240	8.245	22.379	**43.110**
braf	18.421	134.211	159.211	**182.895**
cah2	0.000	2.033	5.285	**5.691**
casp3	1.010	2.020	20.875	**76.768**
cdk2	14.376	49.049	73.714	**74.419**
comt	4.878	3.252	**8.130**	4.878
cp2c9	3.333	50.556	52.778	**58.333**
cp3a4	1.198	51.497	**62.275**	56.287
csf1r	0.000	67.470	85.542	**98.795**
cxcr4	0.000	0.000	**8.333**	0.000
def	7.843	76.471	**87.582**	82.353
dhi1	4.242	**4.848**	2.828	1.212
dpp4	0.375	10.882	18.261	**25.891**
drd3	6.276	6.555	**29.289**	22.176
dyr	6.061	48.773	78.499	**85.714**
egfr	6.273	130.996	149.692	**165.314**
esr1	29.765	18.277	54.308	**88.251**
esr2	21.798	15.259	51.045	**87.193**
fa10	27.933	68.901	85.785	**99.814**
fa7	15.789	38.012	130.409	**161.404**
fabp4	**63.830**	31.206	21.277	17.021
fak1	38.000	132.000	145.333	**176.000**
fgfr1	12.950	69.544	113.189	**146.763**
fkb1a	3.604	0.601	**4.204**	3.604
fnta	1.689	59.459	117.905	**125.338**
fpps	0.000	20.392	123.137	**131.765**
gcr	19.380	2.584	15.762	**27.132**
glcm	0.000	0.000	**3.704**	0.000
gria2	**16.456**	0.844	10.549	8.861
grik1	2.000	17.333	**43.333**	42.000
hdac2	11.892	135.856	134.775	**137.297**
hdac8	28.235	141.961	170.196	**171.765**
hivint	4.000	33.333	50.000	**56.000**
hivpr	2.985	**8.085**	4.726	7.463
hivrt	5.341	12.661	15.826	**17.211**
hmdh	5.882	15.686	54.510	**64.706**
hs90a	0.000	8.333	31.061	**34.091**

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	2.174	62.319	68.841	**73.913**
igf1r	20.270	154.955	155.405	**175.676**
inha	13.953	18.605	15.504	**27.907**
ital	0.000	15.572	**42.822**	40.876
jak2	31.776	172.586	178.193	**188.785**
kif11	**44.828**	2.874	5.747	5.172
kit	3.614	51.004	50.201	**74.699**
kith	**63.158**	17.544	32.749	35.088
kpcb	**53.333**	8.395	19.259	16.296
lck	12.381	110.794	134.444	**159.524**
lkha4	14.118	**54.118**	48.235	44.706
mapk2	15.842	48.185	64.686	**100.990**
mcr	8.511	0.000	3.546	**19.149**
met	15.663	142.972	138.956	**169.880**
mk01	2.532	65.823	129.114	**156.962**
mk10	11.538	71.154	99.359	**113.462**
mk14	10.035	58.362	76.471	**100.692**
mmp13	4.196	66.900	149.534	**172.378**
mp2k1	0.000	57.778	75.000	**88.333**
nos1	2.000	23.333	**50.000**	48.000
nram	0.000	0.000	**11.565**	10.204
pa2ga	0.000	6.122	8.844	**10.204**
parp1	**20.472**	10.892	11.811	15.354
pde5a	14.070	45.226	69.179	**74.874**
pgh1	7.179	8.547	19.487	**23.590**
pgh2	**49.655**	18.851	36.169	39.080
plk1	0.000	66.667	111.321	**122.642**
pnph	19.608	13.072	**41.176**	35.294
ppara	5.898	37.176	69.169	**126.005**
ppard	0.833	23.333	69.167	**117.500**
pparg	5.785	41.598	83.333	**132.231**
prgr	15.700	7.281	12.059	**18.430**
ptn1	47.692	35.897	**51.282**	46.154
pur2	0.000	2.667	56.000	**84.000**
pygm	0.000	4.329	2.597	**5.195**
pyrd	34.234	**63.063**	31.231	28.829
reni	9.709	8.414	77.670	**122.330**
rock1	4.000	71.333	90.667	**134.000**
rxra	61.069	2.036	30.534	**83.969**
sahh	19.048	40.212	24.339	**41.270**
src	3.435	135.242	152.290	**169.847**
tgfr1	13.534	187.970	187.970	**196.992**
thb	44.660	11.003	36.893	**67.961**
thrb	0.434	21.837	80.694	**126.681**
try1	1.782	83.742	123.088	**179.510**
tryb1	12.162	32.883	63.514	**144.595**
tysy	38.532	25.076	97.859	**104.587**
urok	3.704	97.531	148.148	**167.901**
vgfr2	28.362	106.764	131.540	**153.056**
wee1	**142.574**	92.409	141.914	122.772
xiap	**6.061**	0.000	2.694	2.020

Table C.9: Cross-validation DUD-E 1% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	2.905	18.534	26.141	**30.913**
abl1	13.260	72.560	81.584	**88.398**
ace	1.786	6.190	36.310	**56.786**
aces	**16.336**	5.740	7.064	8.389
ada	1.075	26.882	47.312	**59.140**
ada17	30.263	49.123	78.759	**85.714**
adrb1	3.644	25.641	24.022	**29.150**
adrb2	3.030	24.387	22.799	**28.571**
akt1	4.437	81.001	73.151	**89.420**
akt2	26.496	64.957	67.521	**82.906**
aldr	11.321	6.289	10.063	**11.950**
ampc	0.000	3.472	10.417	**14.583**
andr	18.216	3.717	10.781	**26.394**
aofb	**6.557**	1.639	6.284	**6.557**
bace1	3.180	8.481	20.141	**27.562**
braf	17.763	76.974	85.526	**96.053**
cah2	0.000	1.355	**3.862**	3.455
casp3	1.515	5.724	19.024	**45.455**
cdk2	9.514	33.122	45.877	**48.626**
comt	4.878	1.626	**5.691**	4.878
cp2c9	2.500	34.444	38.056	**40.833**
cp3a4	1.796	38.124	41.916	**44.311**
csf1r	1.205	45.582	54.618	**61.446**
cxcr4	0.000	2.500	**5.833**	5.000
def	6.863	40.850	**48.366**	47.059
dhi1	3.636	**5.253**	2.424	1.818
dpp4	0.563	9.193	14.759	**16.698**
drd3	5.021	6.764	**21.757**	18.828
dyr	6.926	35.642	52.237	**58.009**
egfr	5.535	74.785	80.935	**86.716**
esr1	19.060	17.406	39.426	**56.136**
esr2	14.441	17.439	39.237	**58.856**
fa10	19.926	44.631	50.528	**56.983**
fa7	13.158	37.719	77.778	**85.965**
fabp4	**31.915**	23.404	19.149	23.404
fak1	21.000	83.667	82.667	**92.000**
fgfr1	10.072	52.998	70.024	**79.856**
fkb1a	3.604	0.901	4.505	**6.306**
fnta	2.365	41.385	67.624	**71.284**
fpps	0.000	29.804	77.255	**85.882**
gcr	12.403	3.876	11.757	**20.930**
glcm	0.000	0.617	**2.469**	1.852
gria2	12.025	1.688	**12.236**	11.392
grik1	3.000	13.667	25.333	**29.000**
hdac2	13.514	71.351	71.532	**74.054**
hdac8	25.294	79.804	86.471	**87.059**
hivint	2.000	29.667	44.000	**48.000**
hivpr	2.985	5.846	5.286	**8.396**
hivrt	2.967	11.573	11.177	**12.760**
hmdh	4.118	22.353	43.137	**45.294**
hs90a	0.000	12.121	**23.864**	22.727

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	3.261	32.971	**42.029**	41.304
igf1r	15.541	85.586	84.009	**91.892**
inha	9.302	**19.380**	18.605	18.605
ital	0.000	13.869	35.766	**40.146**
jak2	15.888	88.474	92.835	**96.262**
kif11	**36.207**	3.448	3.736	4.310
kit	3.614	35.141	36.145	**56.627**
kith	**33.333**	12.281	23.977	26.316
kpcb	**33.333**	6.667	16.543	15.556
lck	10.000	67.937	76.190	**84.762**
lkha4	15.294	37.255	**38.824**	32.941
mapk2	15.842	33.993	42.244	**61.386**
mcr	5.319	0.355	6.738	**25.532**
met	10.241	77.510	80.321	**88.554**
mk01	3.797	39.662	72.152	**82.278**
mk10	6.731	50.641	57.372	**60.577**
mk14	7.266	39.562	47.578	**59.862**
mmp13	3.671	45.105	81.119	**89.860**
mp2k1	0.000	37.500	41.667	**47.500**
nos1	2.000	17.333	**34.333**	34.000
nram	0.000	0.680	**12.585**	9.184
pa2ga	1.020	5.782	**8.503**	6.122
parp1	**16.339**	9.974	12.270	14.764
pde5a	9.548	30.988	44.975	**47.236**
pgh1	6.667	8.205	14.872	**15.385**
pgh2	**31.724**	14.330	24.751	26.897
plk1	0.000	50.000	65.723	**73.585**
pnph	11.765	19.935	26.144	**26.471**
ppara	6.434	28.418	47.096	**76.408**
ppard	0.833	22.500	52.500	**79.583**
pparg	5.785	32.851	54.545	**72.727**
prgr	11.604	5.461	11.832	**18.089**
ptn1	29.231	29.231	33.333	**36.923**
pur2	2.000	12.000	44.667	**56.000**
pygm	2.597	**7.792**	5.195	6.494
pyrd	22.523	**36.937**	25.526	27.027
reni	4.854	12.621	59.223	**71.845**
rock1	5.000	47.333	53.667	**76.000**
rxra	35.115	2.545	28.499	**56.489**
sahh	14.286	**80.952**	29.630	34.921
src	4.962	76.399	81.934	**87.786**
tgfr1	10.526	96.491	95.990	**100.000**
thb	33.010	12.298	25.890	**37.864**
thrb	1.302	27.693	49.241	**69.414**
try1	2.450	54.417	74.239	**92.650**
tryb1	7.432	24.775	47.297	**85.135**
tysy	23.853	33.945	60.550	**65.138**
urok	4.321	55.556	81.893	**93.827**
vgfr2	17.848	64.874	75.061	**84.352**
wee1	72.277	64.356	**81.518**	79.208
xiap	**9.091**	0.000	4.040	5.051

Table C.10: Cross-validation DUD-E 2% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	3.216	16.805	20.712	**22.303**
abl1	10.773	39.319	42.634	**45.028**
ace	1.607	4.286	25.298	**35.536**
aces	**11.148**	6.770	6.439	7.395
ada	0.538	23.118	29.749	**33.333**
ada17	16.729	31.736	40.445	**45.019**
adrb1	3.441	18.758	18.489	**21.053**
adrb2	2.597	17.821	18.038	**20.563**
akt1	5.631	43.970	43.402	**47.270**
akt2	13.248	38.319	41.311	**47.863**
aldr	6.918	5.241	8.700	**8.805**
ampc	1.042	3.125	9.028	**10.417**
andr	10.409	3.408	9.170	**19.703**
aofb	**6.967**	1.366	4.918	5.738
bace1	2.827	7.420	**18.198**	18.021
braf	12.171	41.996	45.724	**48.684**
cah2	0.102	1.050	**3.049**	2.947
casp3	2.525	6.987	15.320	**25.758**
cdk2	5.920	21.670	27.766	**29.915**
comt	2.439	1.626	**5.285**	4.878
cp2c9	2.083	20.972	25.278	**27.917**
cp3a4	1.198	25.549	**27.046**	26.647
csf1r	3.313	29.719	35.040	**38.554**
cxcr4	0.000	**9.167**	6.250	8.750
def	5.392	23.366	**29.085**	27.941
dhi1	3.939	**4.495**	2.323	2.121
dpp4	0.938	7.286	10.663	**11.351**
drd3	4.184	5.962	**16.109**	15.586
dyr	5.628	23.160	31.385	**33.333**
egfr	4.705	41.175	43.204	**44.742**
esr1	11.619	15.318	27.720	**34.595**
esr2	10.627	14.941	27.384	**34.469**
fa10	13.128	26.660	30.571	**33.147**
fa7	12.719	26.901	42.105	**44.737**
fabp4	**15.957**	13.830	13.830	**15.957**
fak1	11.000	44.833	43.500	**46.500**
fgfr1	7.194	35.372	39.928	**44.245**
fkb1a	2.252	0.901	3.904	**5.405**
fnta	1.943	27.055	38.091	**39.274**
fpps	0.000	26.667	43.922	**49.412**
gcr	7.946	4.974	10.401	**15.891**
glcm	0.000	1.852	**2.469**	1.852
gria2	8.544	3.165	11.498	**11.709**
grik1	3.000	9.833	16.333	**19.500**
hdac2	11.622	37.117	37.838	**38.378**
hdac8	17.353	41.471	44.510	**45.294**
hivint	1.000	22.167	29.500	**31.500**
hivpr	2.985	4.944	5.597	**9.888**
hivrt	2.226	**9.545**	9.248	9.496
hmdh	3.235	18.137	28.529	**29.412**
hs90a	0.000	13.447	17.235	**18.182**

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	3.261	17.935	24.457	**25.543**
igf1r	9.459	44.257	43.694	**46.622**
inha	8.140	**17.054**	12.403	16.279
ital	0.000	13.017	25.182	**28.102**
jak2	9.813	47.040	47.664	**48.598**
kif11	**21.552**	2.299	3.736	3.017
kit	4.518	23.092	23.594	**38.554**
kith	18.421	10.234	19.883	**20.175**
kpcb	**20.000**	4.691	14.074	14.444
lck	6.786	37.063	40.952	**44.405**
lkha4	15.000	25.490	**27.941**	23.235
mapk2	12.376	19.637	28.383	**35.644**
mcr	3.191	0.532	6.383	**24.468**
met	9.940	41.667	43.775	**45.783**
mk01	3.165	25.738	38.397	**43.671**
mk10	5.769	29.006	31.410	**32.692**
mk14	6.142	25.260	29.844	**34.775**
mmp13	3.497	30.653	43.706	**46.591**
mp2k1	0.833	21.806	24.028	**24.167**
nos1	1.000	12.167	20.000	**21.000**
nram	0.000	0.850	8.844	**10.204**
pa2ga	0.510	5.442	6.122	**6.122**
parp1	12.795	9.383	11.549	**13.878**
pde5a	8.040	20.184	27.554	**29.271**
pgh1	5.128	6.838	**11.026**	**11.026**
pgh2	20.000	10.920	16.590	**20.115**
plk1	1.887	29.874	39.623	**42.925**
pnph	7.353	**20.098**	18.301	19.118
ppara	6.166	20.643	29.491	**42.761**
ppard	1.667	17.153	32.083	**45.417**
pparg	5.269	23.691	33.815	**40.393**
prgr	8.362	5.119	10.694	**17.235**
ptn1	17.692	18.718	21.923	**22.308**
pur2	1.000	15.667	30.333	**31.000**
pygm	1.299	7.576	6.277	**8.442**
pyrd	13.063	**23.724**	18.619	20.721
reni	3.398	13.592	36.246	**40.291**
rock1	3.000	27.833	33.000	**43.500**
rxra	20.229	3.181	23.410	**33.969**
sahh	10.317	**46.032**	24.074	30.159
src	4.008	40.999	43.575	**45.611**
tgfr1	9.023	49.248	48.872	**50.000**
thb	20.874	11.165	18.447	**22.330**
thrb	3.362	23.030	30.369	**38.069**
try1	2.227	32.665	41.500	**47.996**
tryb1	5.743	17.455	31.194	**46.284**
tysy	15.138	27.523	37.156	**38.991**
urok	5.556	30.144	43.724	**47.840**
vgfr2	10.636	37.979	41.443	**45.232**
wee1	38.119	41.584	44.059	**46.535**
xiap	**7.071**	0.168	5.051	6.061

Table C.11: Cross-validation DUD-E 5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
aa2ar	2.739	10.871	**12.794**	11.826
abl1	7.624	17.348	17.790	**18.674**
ace	1.000	3.357	13.857	**16.714**
aces	6.667	6.799	6.049	**7.108**
ada	0.860	12.688	13.620	**13.763**
ada17	7.406	15.789	16.967	**18.722**
adrb1	3.158	10.499	12.740	**13.522**
adrb2	3.117	9.899	13.189	**14.026**
akt1	4.505	18.840	18.817	**19.386**
akt2	7.692	18.348	18.632	**19.829**
aldr	4.403	3.774	**5.241**	5.031
ampc	1.667	2.639	4.861	**5.417**
andr	5.130	4.040	6.716	**13.011**
aofb	**5.574**	1.913	3.880	4.426
bace1	2.686	6.031	**12.933**	11.802
braf	7.763	18.465	19.254	**19.474**
cah2	0.569	1.070	2.249	**2.520**
casp3	2.424	5.556	9.798	**12.121**
cdk2	4.228	11.247	13.672	**15.560**
comt	1.951	2.439	5.203	**6.341**
cp2c9	2.667	11.056	13.056	**13.500**
cp3a4	1.916	12.854	13.533	**13.772**
csf1r	2.289	14.458	16.988	**17.952**
cxcr4	0.000	**9.667**	8.000	8.500
def	5.490	12.484	14.183	**15.098**
dhi1	**3.455**	3.253	2.364	2.000
dpp4	0.938	5.153	6.604	**7.467**
drd3	3.598	4.840	**10.028**	10.000
dyr	3.983	12.323	15.036	**15.238**
egfr	3.247	17.823	18.266	**18.635**
esr1	6.580	10.044	14.917	**15.875**
esr2	7.139	10.064	15.005	**15.640**
fa10	7.486	12.886	14.898	**15.866**
fa7	11.404	14.971	18.012	**18.596**
fabp4	6.383	5.816	**8.369**	8.085
fak1	6.000	18.467	18.600	**19.800**
fgfr1	4.892	16.691	18.177	**18.993**
fkb1a	1.441	1.381	4.685	**5.405**
fnta	1.858	14.347	17.173	**17.466**
fpps	0.000	16.157	19.529	**20.000**
gcr	4.496	5.168	8.915	**11.240**
glcm	0.000	**2.963**	2.593	2.593
gria2	5.190	5.738	9.662	**10.127**
grik1	3.000	6.267	8.867	**9.200**
hdac2	8.757	15.892	16.216	**16.432**
hdac8	9.059	17.373	18.549	**18.588**
hivint	1.800	12.200	14.267	**14.400**
hivpr	2.910	3.980	5.572	**8.172**
hivrt	2.730	6.034	6.291	**6.825**
hmdh	2.941	11.686	16.235	**16.471**
hs90a	0.000	8.939	9.697	**10.682**

Target	Vina	Baseline CNN	DenseU	DenseFS
hxk4	1.522	7.754	**13.043**	**13.043**
igf1r	6.486	18.468	18.333	**19.189**
inha	4.186	**11.008**	8.217	7.907
ital	0.584	9.051	13.723	**14.307**
jak2	5.981	19.377	19.252	**19.813**
kif11	**11.379**	2.471	2.931	2.931
kit	3.133	12.369	14.337	**17.952**
kith	9.474	6.316	11.930	**13.684**
kpcb	9.481	4.494	10.074	**10.667**
lck	4.857	16.429	17.492	**19.000**
lkha4	10.588	14.863	**15.882**	12.471
mapk2	8.515	10.759	14.719	**17.624**
mcr	1.915	1.915	6.170	**13.617**
met	6.747	17.992	18.675	**19.277**
mk01	6.076	14.937	16.793	**18.481**
mk10	4.231	12.436	13.910	**14.808**
mk14	4.048	13.356	14.983	**16.228**
mmp13	3.042	15.548	18.660	**19.476**
mp2k1	0.333	10.444	10.556	**11.500**
nos1	1.400	6.600	9.867	**10.600**
nram	0.000	1.224	5.578	**6.939**
pa2ga	0.612	4.694	4.762	**5.714**
parp1	7.323	7.769	9.121	**9.882**
pde5a	4.623	10.754	14.322	**14.874**
pgh1	3.282	4.923	6.188	**7.077**
pgh2	9.701	7.540	10.238	**10.621**
plk1	1.887	14.843	17.170	**17.736**
pnph	6.863	**12.418**	10.327	11.373
ppara	6.649	12.851	15.567	**18.606**
ppard	2.917	11.333	16.000	**18.750**
pparg	4.380	13.485	16.612	**17.934**
prgr	5.051	4.300	8.077	**11.945**
ptn1	9.538	10.205	11.641	**12.308**
pur2	5.200	14.800	14.267	**15.200**
pygm	1.299	**6.061**	5.368	5.714
pyrd	8.108	11.291	10.450	**12.432**
reni	3.495	11.262	17.735	**18.835**
rock1	2.400	13.867	16.733	**18.200**
rxra	9.924	4.326	14.707	**16.947**
sahh	6.984	**19.153**	13.968	18.413
src	2.977	17.672	18.295	**18.969**
tgfr1	9.173	19.950	19.900	**20.000**
thb	10.291	9.256	10.356	**12.039**
thrb	3.514	12.495	14.577	**17.527**
try1	3.252	16.347	18.396	**19.733**
tryb1	4.054	11.982	15.721	**19.730**
tysy	10.459	15.963	17.125	**17.798**
urok	4.815	14.650	19.095	**19.630**
vgfr2	6.259	16.985	18.142	**19.609**
wee1	16.040	18.746	19.010	**19.604**
xiap	**4.242**	0.404	3.771	3.232

Table C.12: Independent ChEMBL set AUC ROC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	0.411	0.652	0.681	**0.784**
10498	0.578	0.722	0.733	**0.807**
10752	0.733	0.830	0.852	**0.872**
11279	0.536	0.841	0.811	**0.857**
11359	0.672	0.817	0.815	**0.826**
11534	0.569	0.793	0.815	**0.895**
11631	0.668	**0.828**	0.816	0.816
12670	0.718	0.895	0.901	**0.925**
12968	**0.645**	0.545	0.541	0.621
18061	**0.740**	0.679	0.703	0.712
20014	0.791	0.910	0.927	**0.947**
219	0.516	0.744	0.843	**0.888**
276	0.852	**0.871**	0.861	0.870
28	0.762	**0.910**	0.901	0.907

Table C.13: Independent ChEMBL set AUC PRC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	0.014	0.045	0.031	**0.155**
10498	0.026	0.094	0.070	**0.189**
10752	0.079	0.204	**0.278**	0.247
11279	0.010	**0.049**	0.038	0.043
11359	0.030	0.174	0.200	**0.217**
11534	0.015	0.091	0.073	**0.207**
11631	0.024	0.123	**0.178**	0.165
12670	0.048	0.259	0.360	**0.395**
12968	**0.024**	0.011	0.012	0.014
18061	0.054	0.033	0.032	**0.035**
20014	0.046	0.485	0.573	**0.581**
219	0.020	0.032	**0.117**	0.115
276	0.071	0.266	0.295	**0.327**
28	0.088	0.263	0.273	**0.304**

Table C.14: Independent ChEMBL set 0.5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	2.000	14.000	6.667	**38.000**
10498	4.000	28.667	18.667	**44.000**
10752	18.000	51.333	**62.667**	56.000
11279	0.000	4.000	**6.000**	0.000
11359	6.000	43.333	50.667	**62.000**
11534	2.000	26.000	20.000	**42.000**
11631	4.000	28.667	**42.667**	40.000
12670	10.000	60.667	78.667	**86.000**
12968	**8.000**	0.000	1.333	2.000
18061	**14.000**	6.000	8.667	8.000
20014	12.000	98.000	109.333	**114.000**
219	10.101	9.428	**26.936**	22.222
276	16.000	64.667	65.333	**82.000**
28	24.000	64.000	62.667	**70.000**

Table C.15: Independent ChEMBL set 1% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	1.000	9.000	4.667	**23.000**
10498	3.000	17.667	12.000	**26.000**
10752	12.000	33.333	**36.333**	35.000
11279	0.000	5.000	**5.333**	4.000
11359	7.000	27.667	34.000	**37.000**
11534	1.000	16.667	14.667	**29.000**
11631	6.000	19.333	28.667	**31.000**
12670	10.000	40.333	46.000	**52.000**
12968	**4.000**	0.333	1.000	1.000
18061	**12.000**	5.667	7.333	9.000
20014	8.000	57.667	65.333	**67.000**
219	7.071	7.744	21.549	**25.253**
276	14.000	43.000	42.667	**48.000**
28	20.000	42.000	40.333	**43.000**

Table C.16: Independent ChEMBL set 2% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	0.500	6.667	4.667	**19.000**
10498	3.000	11.667	8.667	**16.000**
10752	7.000	19.500	21.500	**19.500**
11279	0.000	**6.000**	5.167	4.000
11359	5.500	20.833	20.167	**21.500**
11534	1.500	10.833	10.167	**17.000**
11631	4.000	13.667	19.333	**21.000**
12670	8.000	24.167	27.833	**30.000**
12968	**3.000**	0.500	0.833	2.000
18061	**8.000**	5.667	6.333	7.000
20014	6.000	32.333	36.833	**37.500**
219	4.040	5.556	14.141	**16.667**
276	11.000	26.500	26.833	**29.500**
28	13.000	**25.167**	24.167	24.500

Table C.17: Independent ChEMBL set 5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
10378	1.200	4.133	3.467	**9.600**
10498	3.000	7.000	6.333	**9.000**
10752	5.600	**9.667**	**9.667**	9.000
11279	0.000	**6.400**	4.333	5.400
11359	3.800	10.000	10.333	**10.600**
11534	1.600	7.600	6.667	**10.200**
11631	4.200	10.200	10.267	**10.800**
12670	5.600	13.000	13.800	**14.800**
12968	**4.600**	0.333	0.933	1.000
18061	**5.400**	4.800	4.667	5.000
20014	5.600	14.733	17.000	**17.200**
219	3.030	3.838	8.215	**10.707**
276	8.200	13.467	12.667	**13.600**
28	8.200	12.667	12.333	**12.800**

Table C.18: Independent MUV set AUC ROC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	0.571	**0.695**	0.604	0.607
548	0.457	0.800	0.783	**0.835**
600	**0.577**	0.541	0.574	0.524
689	**0.592**	0.504	0.535	0.470
692	0.411	0.415	**0.435**	0.349
832	**0.607**	0.369	0.473	0.513
846	**0.654**	0.350	0.556	0.492
852	**0.512**	0.311	0.483	0.430
859	0.535	0.577	0.584	**0.588**

Table C.19: Independent MUV set AUC PRC for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	0.002	**0.003**	0.002	**0.003**
548	0.002	**0.011**	0.009	0.010
600	0.004	0.003	**0.004**	0.003
689	0.003	**0.005**	**0.005**	0.003
692	0.002	0.002	0.002	0.002
832	**0.004**	0.002	0.002	0.002
846	0.003	0.002	0.003	**0.004**
852	**0.002**	0.001	**0.002**	**0.002**
859	0.002	0.002	**0.003**	**0.003**

Table C.20: Independent MUV set 0.5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	0.000	0.000	0.000	**8.333**
548	0.000	**11.494**	4.598	0.000
600	0.000	0.000	**6.667**	0.000
689	0.000	**6.667**	**6.667**	**6.667**
692	0.000	0.000	0.000	0.000
832	0.000	0.000	0.000	0.000
846	0.000	0.000	2.222	**6.667**
852	0.000	0.000	0.000	0.000
859	0.000	0.000	0.000	0.000

Table C.21: Independent MUV set 1% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	**4.167**	1.389	1.389	**4.167**
548	0.000	**10.345**	4.598	**10.345**
600	0.000	1.111	**3.333**	**3.333**
689	3.333	3.333	3.333	3.333
692	0.000	0.000	0.000	0.000
832	**3.333**	0.000	0.000	0.000
846	0.000	0.000	**3.333**	**3.333**
852	0.000	0.000	0.000	0.000
859	0.000	0.000	2.899	**4.348**

Table C.22: Independent MUV set 2% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	**2.083**	0.694	1.389	**2.083**
548	0.000	**8.046**	4.598	6.897
600	**5.000**	1.111	2.222	1.667
689	1.667	1.667	**2.222**	1.667
692	0.000	0.000	0.000	0.000
832	**5.000**	0.000	0.000	0.000
846	0.000	0.000	2.222	**3.333**
852	0.000	0.000	0.000	0.000
859	2.174	2.174	2.899	**4.348**

Table C.23: Independent MUV set 5% ROC EF for Vina, Baseline CNN, and our CNN models.

Target	Vina	Baseline CNN	DenseU	DenseFS
466	0.833	**1.389**	1.111	0.833
548	0.000	**6.667**	5.977	4.138
600	**2.667**	1.556	2.444	1.333
689	**2.000**	1.556	1.333	0.667
692	0.000	**1.111**	0.222	0.667
832	**3.333**	0.000	0.000	0.667
846	0.667	0.444	**1.556**	1.333
852	**0.667**	0.000	0.000	**0.667**
859	1.739	1.739	**2.609**	1.739

D

Generating Property-Matched Decoy Molecules Using Deep Learning

Contents

D.1 Additional DeepCoy Model Details

D.1.1 Atom Types

In line with both Liu et al. (2018) and Imrie et al. (2020), 14 atom types are permitted in the standard version of our model: carbon, nitrogen (N^-, N, N^+), oxygen (O^-, O,

O^+), fluorine, chlorine, bromine, iodine, and sulphur (maximum valence 2, 4, or 6).

All active molecules for DUD-E target FPPS contained phosphorus, and thus for this target we included phosphorus as an additional atom type (with maximum valence 5).

D.1.2 Phosphorus Training Set

The 250 000 molecule subset of ZINC (Sterling and Irwin, 2015) selected at random by Gómez-Bombarelli et al. (2018) used to train the standard versions of DeepCoy contains only 123 compounds with phosphorus. In order to construct a more suitable training set, we extracted all phosphorus-containing compounds from ZINC (Sterling and Irwin, 2015) and followed the same procedure and thresholds described in Section 5.3.2 to construct a training set. This process resulted in 109,061 pairs that were used to train our model. We used identical hyperparameters as the other versions of DeepCoy.

D.1.3 Network Architecture

Following Liu et al. (2018), both the encoder and decoder utilise standard gated graph neural networks (GGNN, Li et al., 2016), which propagate messages for 7 steps and have residual connections between odd numbered time steps.

The neural network mapping the hidden state of a node to its atom type is implemented as a linear classifier, following Liu et al. (2018).

D.1.4 Hyperparameters

We trained the model with a learning rate of 0.001 for 10 epochs using the Adam optimiser and a batch size of 8, selecting the model from the epoch with the lowest validation loss. The dimension of the latent space was 100 and the dimension of the vector used to encode the target molecule during training and sampled from a $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ distribution when generating new molecules was 8. λ_{KL} was set at 0.3.

We performed limited hyperparameter optimisation. However, we found that the model required a larger latent space dimension than in Imrie et al. (2020), with the

dimension of the latent space matching Liu et al. (2018). We believe this is due to the additional number of steps required to construct a full molecule compared to a partial structure as well as the importance of the input molecule for the generation task.

D.2 Physicochemical Properties to Unbias

This section contains a list of properties that were unbiased. We calculated the values for all properties using RDKit (Landrum, 2006). This will likely cause minor discrepancies with the property values calculated in the construction of both DUD-E (Mysinger et al., 2012) and DEKOIS 2.0 (Bauer et al., 2013). Indeed, values of the metrics used to assess property-matching and structural similarity were similar but not identical to the values reported by Bauer et al. (2013). All values reported here are calculated in the same way and thus directly comparable.

D.2.1 DUD-E

DUD-E (Mysinger et al., 2012) selected the following six properties to match: molecular weight, log P, number of hydrogen bond acceptors, number of hydrogen bond donors, number of rotatable bonds, and net charge.

D.2.2 DEKOIS 2.0

DEKOIS 2.0 (Bauer et al., 2013) was constructed by matching the following eight properties: molecular weight, log P, number of hydrogen bond acceptors, number of hydrogen bond donors, number of rotatable bonds, number of aromatic rings, positive charge, and negative charge.

D.2.3 All Properties

In our experiments using the DUD-E data set, we trained our model to match a substantially larger number of properties to assess whether DeepCoy could handle higher-dimensional restrictions.

Chaput et al. (2016) focused their analysis of the DUD-E data set on nine properties, largely overlapping with the original DUD-E properties. In particular,

they noted deviation between the DUD-E actives and decoys for the embranchment count and polar surface area. We included these properties as the number of chiral centers (Chaput et al. 2016 reported a correlation of 0.97 with embranchment count) and topological polar surface area (TPSA).

MUV (Rohrer and Baumann, 2009) was constructed to unbias the following 17 properties: simple counts of all atoms, heavy atoms, boron, bromine, carbon, chlorine, fluorine, iodine, nitrogen, oxygen, phosphorus, and sulfur atoms, number of hydrogen bond acceptors, number of hybond donors, logP, number of chiral centers, and number of rings.

Combining the properties used to construct DUD-E, DEKOIS 2.0, and MUV, together with the two additional properties from Chaput et al. (2016), synthetic accessibility (Ertl and Schuffenhauer, 2009), and quantitive estimation of druglikeness (QED, Bickerton et al., 2012) yields 27 unique physicochemical properties.

D.3 Deep Learning-Based SBVS

As further validation, we used the DeepCoy decoys to both train and test deep learning-based SBVS methods. Specifically, we used the DUD-E set with both the original decoys and the DeepCoy decoys and trained the convolutional neural networks proposed by Ragoza et al. (2017) ("gnina") and Imrie et al. (2018) ("DenseU" and "DenseFS"). For an external set, we used the subset of the datasets curated from ChEMBL (Bento et al., 2014) by Riniker and Landrum (2013) adopted by Ragoza et al. (2017). This subset was selected by Ragoza et al. (2017) to ensure that models were evaluated on targets dissimilar to those in the training set. This was achieved by ensuring a maximum global sequence similarity of 80% with any training target and also removing any target which had a significant structural alignment of its binding site with any training target according to ProBiS (Konc and Janežič, 2010) (using the default ProBiS parameters).

As discussed in Section 5.4.3, the ChEMBL targets share similar biases to the original DUD-E, but shared limited bias with the DeepCoy decoys. As such, it is not

appropriate to compare the performance of models trained or evaluated on the DUD-E and DeepCoy decoys. In addition, limited conclusions about the relative quality of the decoy molecules can be made based on these experiments as it is challenging to conclude if models trained on DUD-E have learnt bias or meaningful features.

D.3.1 Implementation Details

Following Ragoza et al. (2017) and Imrie et al. (2018), we trained the CNN models only on the top-ranked AutoDock Vina pose for each complex. Models were trained using stochastic gradient descent with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.001. Models were trained for the same number of iterations with the same batch sizes described in Ragoza et al. (2017) and Imrie et al. (2018). We employed the same data augmentation scheme as Ragoza et al. (2017). We did not perform any hyperparameter tuning. We trained three replicas of the models using different random seeds. The performance of the random seeds was averaged in the case of gnina and DenseU, while the predictions were combined in an ensemble for DenseFS. Models were implemented using PyTorch (Paszke et al., 2019) and libmolgrid (Sunseri and Koes, 2020) for molecular gridding and code can be found at https://github.com/oxpig/DenseFS.

D.3.2 Train DUD-E, Test ChEMBL

Table D.1 shows the performance of AutoDock Vina, gnina (Ragoza et al., 2017) and DenseU (Imrie et al., 2018) on the ChEMBL test sets trained on the original version of DUD-E and the version of DUD-E employing DeepCoy decoys.

While there is a significant reduction in the predictive power of the CNN-based methods when trained on DeepCoy decoys, a random forest model trained on the unbiased features experiences a larger decrease in performance (average AUC ROC all features: DUD-E 0.84, DeepCoy 0.57).

As discussed in Section 5.4.3, due to the shared bias in the original DUD-E dataset and ChEMBL, it is challenging to conclude whether models have learnt bias or meaningful features. However, since the version of DUD-E employing

DeepCoy decoys does not contain these biases, we can be more confident that any predictive power on the ChEMBL test set arises from learning meaningful features. When trained on the version of DUD-E employing DeepCoy decoys, the CNN-based models outperform both AutoDock Vina and the random forest model trained on the unbiased features.

D.3.3 Train ChEMBL, Test DUD-E

We trained the same CNN models using the ChEMBL targets and rescored the docked poses of the DUD-E set. In line with AutoDock Vina, gnina had lower virtual screening performance when assessed on the DeepCoy decoys compared to the original decoy molecules. However, while DenseU experienced a marginal drop in performance as measured by average AUC ROC, early enrichment and AUC PRC saw modest increases (Table D.2).

Notably, AutoDock Vina experienced the largest fall in predictive performance, despite being a linear combination of five energy terms representing protein-ligand interactions and the number of rotatable bonds in the ligand. The virtual screening performance as measured by AUC ROC fell by 0.07 from 0.70 to 0.63 compared to smaller drops for gnina (reduction of 0.05) and DenseU (reduction of 0.02). This was also true for the other performance metrics.

In addition, we generated DeepCoy decoys for the ChEMBL test sets. The average DOE score for the ChEMBL test targets was 0.173 and 0.170, calculated using the original DUD-E properties or all properties, respectively. Using DeepCoy decoys the average DOE scores fell to 0.033 and 0.026, respectively, indicating a substantial reduction in bias.

We repeated the above experiment, replacing the original decoys in the ChEMBL test sets with DeepCoy decoys and trained CNN-based methods on these sets. As expected, the performance of both CNN-based methods when trained on DeepCoy decoys fell for the original DUD-E set, since the shared bias between the training and test set has been removed (Table D.3). The performance on the DeepCoy version of DUD-E improved, likely due to the mismatch of unbiased properties

being removed and the lack of independence between decoys in the training and test set. This suggests that even when decoys exhibit minimal bias on a per-target basis, independent test sets should be employed for validation, in line with accepted practice for model evaluation (e.g. Sieg et al., 2019; Ballester, 2019; Adeshina et al., 2020)

Due to the limited training set size (only 14 protein targets with 100 actives per target) and the bias shared between the ChEMBL targets and the original DUD-E set, we believe that limited conclusions can be drawn from these experiments.

D.4 Additional Results

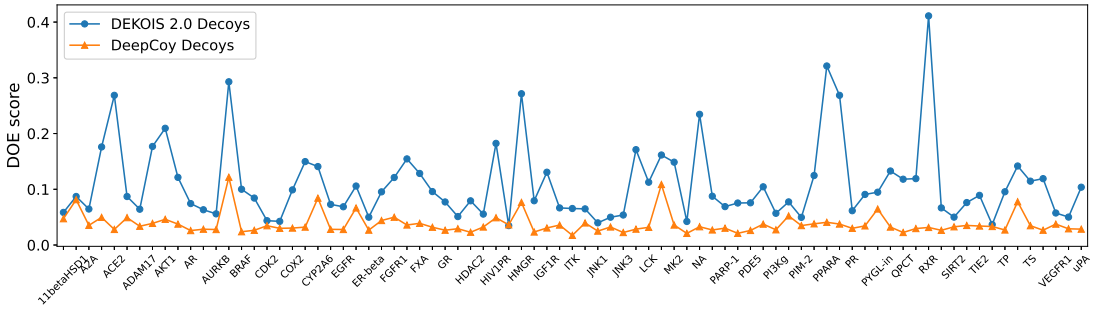


Figure D.1: DOE scores of the original DEKOIS 2.0 set (blue) compared to the DeepCoy generated decoys (orange). The x-axis displays each DEKOIS 2.0 target in the same order as they appear in the DEKOIS 2.0 database (http://www.dekois.com/). The targets with even indices are not labeled on the x-axis due to space limitations.

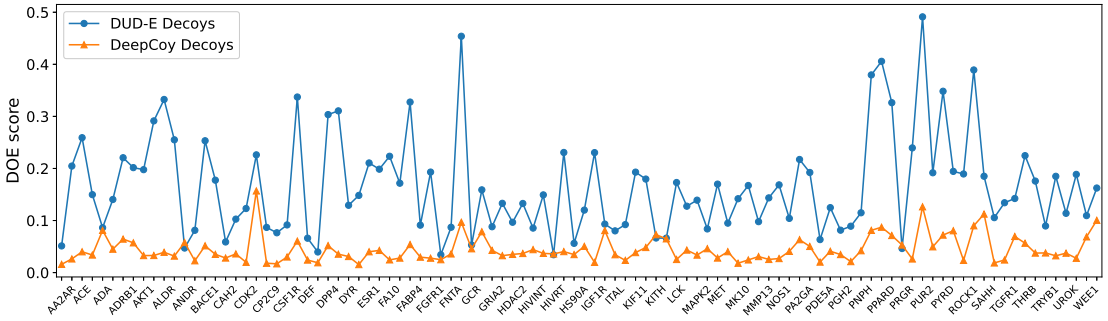


Figure D.2: DOE scores of the original DUD-E set (blue) compared to the final DeepCoy generated decoys (orange) that were selected based on a larger number of properties to unbias, calculating DOE score using only the original DUD-E properties. The x-axis displays each DUD-E target in the same order as they appear in the DUD-E database (http://dude.docking.org/targets). The targets with even indices are not labeled on the x-axis due to space limitations.

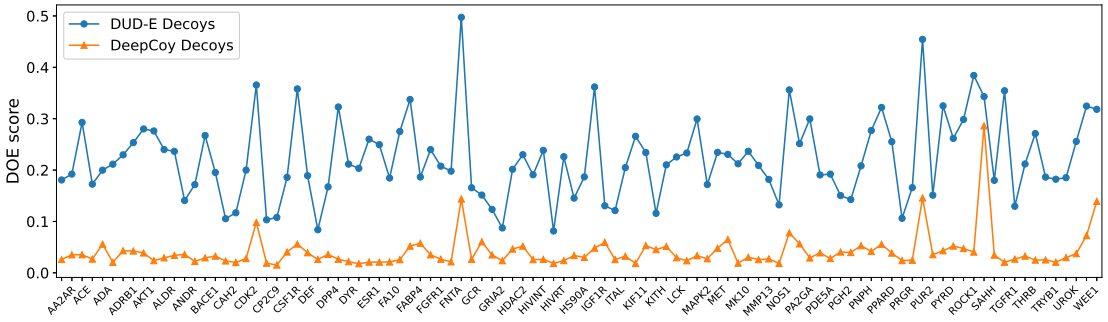


Figure D.3: DOE scores of the original DUD-E set (blue) compared to the final DeepCoy generated decoys (orange) that were selected based on a larger number of properties to unbias, calculating DOE score using all 27 properties to unbias. The x-axis displays each DUD-E target in the same order as they appear in the DUD-E database (http://dude.docking.org/targets). The targets with even indices are not labeled on the x-axis due to space limitations.

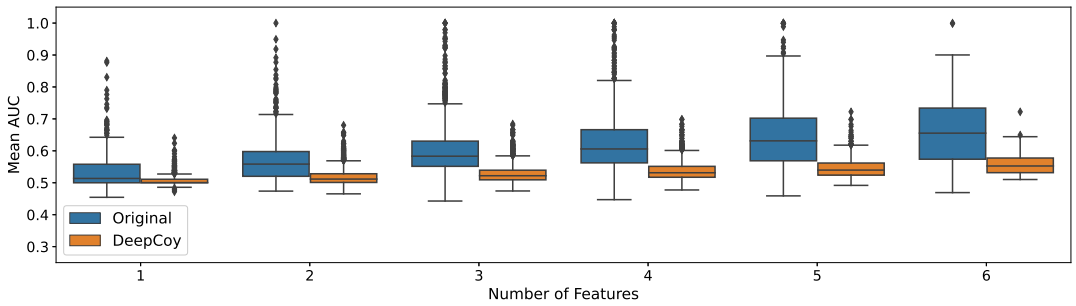


Figure D.4: Results of the machine learning-based assessment of physicochemical property matching on DUD-E. A 1-nearest neighbour model was trained to predict whether a compound was an active or a decoy based on the unbiased features. Virtual screening performance was assessed by AUC ROC for the original DUD-E decoys and DeepCoy generated decoys. The DeepCoy generated decoys resulted in a reduction in the median per-target AUC ROC using all 6 features from 0.66 to 0.55, indicating a substantial reduction in bias.

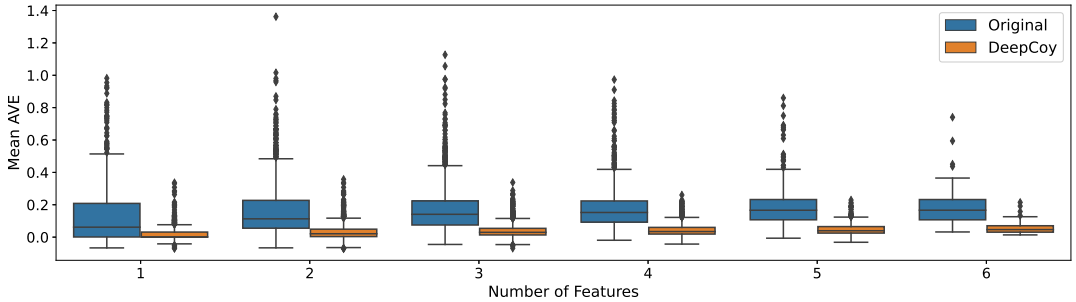


Figure D.5: Results of the AVE assessment of physicochemical property matching on DUD-E. The DeepCoy generated decoys resulted in a 72% reduction in the median AVE using all 6 features decreasing from 0.17 to 0.05, indicating a substantial reduction in bias.

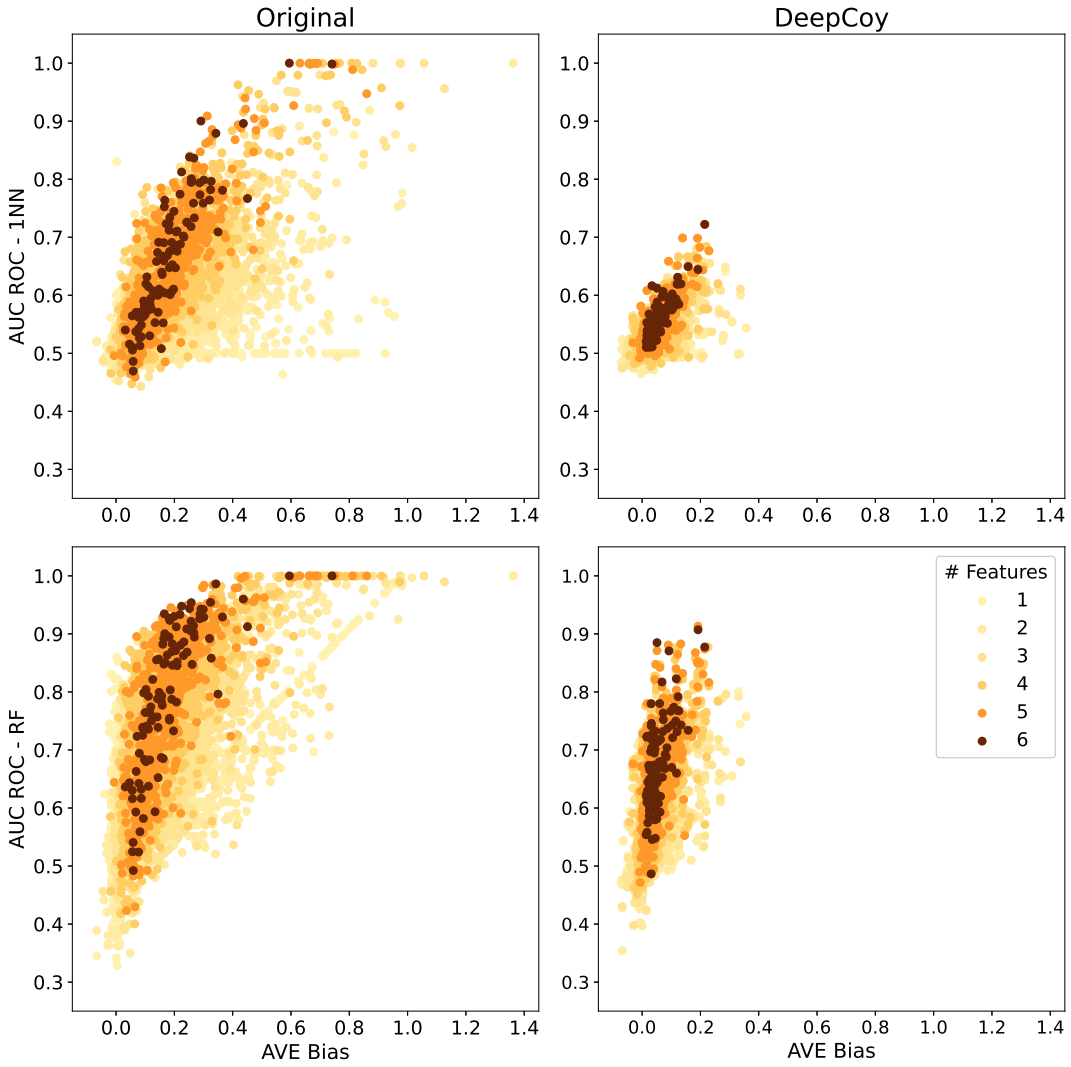


Figure D.6: Comparison of machine learning-based assessment and AVE assessment of physicochemical property matching on DUD-E.

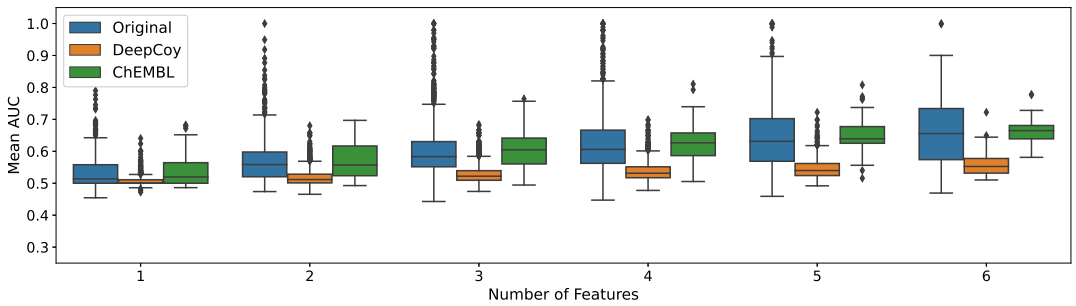


Figure D.7: Results of the machine learning-based assessment of physicochemical property matching on DUD-E and the ChEMBL test sets. A 1-nearest neighbour model was trained to predict whether a compound was an active or a decoy based on the unbiased features. Virtual screening performance was assessed by AUC ROC.

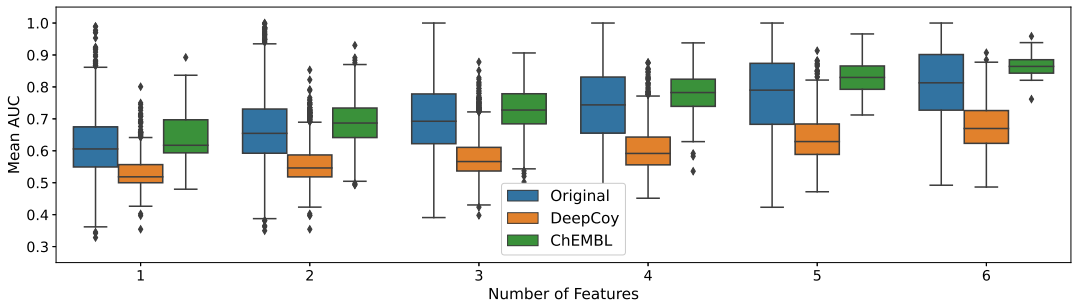


Figure D.8: Results of the machine learning-based assessment of physicochemical property matching on DUD-E and the ChEMBL test sets. A random forest model was trained to predict whether a compound was an active or a decoy based on the unbiased features. Virtual screening performance was assessed by AUC ROC.

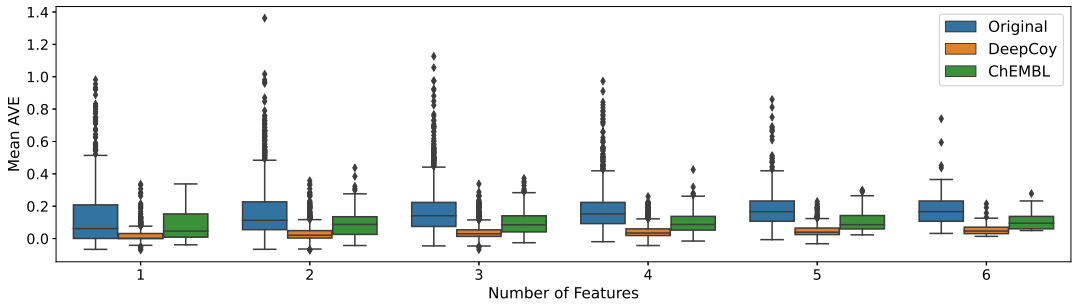


Figure D.9: Results of the AVE assessment of physicochemical property matching on DUD-E and the ChEMBL test sets.

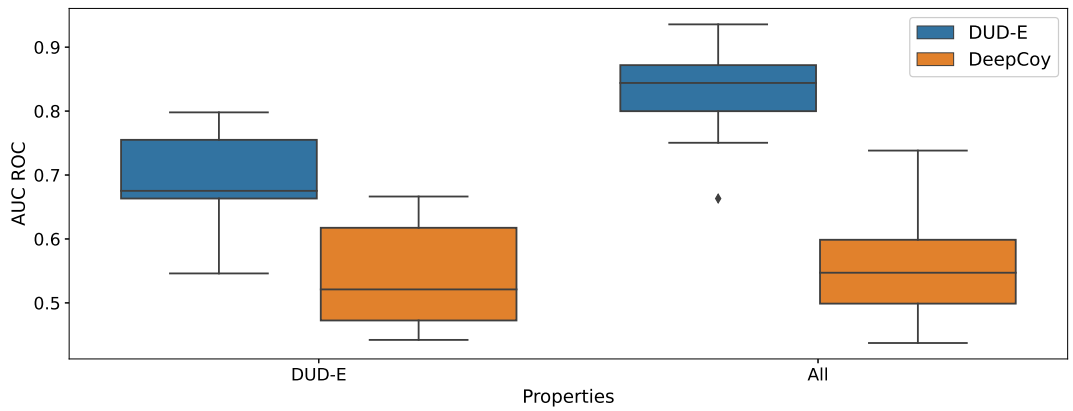


Figure D.10: Results of the machine learning-based assessment of physicochemical property matching on the ChEMBL test sets. A random forest model was trained on the DUD-E targets to predict whether a compound was an active or a decoy based on the unbiased features. Virtual screening performance was assessed by AUC ROC.

Table D.1: SBVS performance on ChEMBL test sets when trained on DUD-E with either the original or DeepCoy decoys. We used the smina (Koes et al., 2013) implementation of AutoDock Vina (Trott and Olson, 2010). "gnina" refers to the CNN model of Ragoza et al. (2017) and "DenseU", "DenseFS" to the universal and protein family-specific CNN models of Imrie et al. (2018), respectively.

	AutoDock	gnina		DenseU		DenseFS	
Metric	Vina	Original	DeepCoy	Original	DeepCoy	Original	DeepCoy
AUC ROC	0.66	0.79	0.74	0.80	0.76	0.82	0.77
AUC PRC	0.04	0.14	0.07	0.17	0.07	0.21	0.11
0.5% ROC EF	9.29	30.29	14.77	36.97	14.01	44.02	23.58
1.0% ROC EF	7.51	21.34	11.01	23.56	11.20	27.37	16.01
2.0% ROC EF	5.32	14.47	7.92	15.33	8.71	17.69	10.26
5.0% ROC EF	4.29	8.42	5.36	8.23	5.99	9.35	6.35

Table D.2: SBVS performance on DUD-E with either the original or DeepCoy decoys when trained on the ChEMBL test sets. We used the smina (Koes et al., 2013) implementation of AutoDock Vina (Trott and Olson, 2010). "gnina" refers to the CNN model of Ragoza et al. (2017) and "DenseU" to the universal CNN model of Imrie et al. (2018).

	AutoDock Vina		gnina		DenseU	
Metric	Original	DeepCoy	Original	DeepCoy	Original	DeepCoy
AUC ROC	0.70	0.63	0.77	0.72	0.75	0.73
AUC PRC	0.09	0.06	0.08	0.07	0.07	0.09
0.5% ROC EF	15.02	8.54	7.57	6.88	8.37	12.60
1.0% ROC EF	10.39	6.36	6.92	6.11	7.01	9.90
2.0% ROC EF	7.14	4.72	6.26	5.42	5.72	7.52
5.0% ROC EF	4.73	3.21	5.14	4.39	4.32	5.32

Table D.3: SBVS performance on DUD-E with either the original or DeepCoy decoys when trained on the ChEMBL test sets with the original decoys replaced with DeepCoy decoys. We used the smina (Koes et al., 2013) implementation of AutoDock Vina (Trott and Olson, 2010). "gnina" refers to the CNN model of Ragoza et al. (2017) and "DenseU" to the universal CNN model of Imrie et al. (2018).

	AutoDock Vina		gnina		DenseU	
Metric	Original	DeepCoy	Original	DeepCoy	Original	DeepCoy
AUC ROC	0.70	0.63	0.70	0.82	0.70	0.84
AUC PRC	0.09	0.06	0.04	0.14	0.05	0.21
0.5% ROC EF	15.02	8.54	3.49	19.56	6.46	35.91
1.0% ROC EF	10.39	6.36	3.32	14.50	5.26	24.61
2.0% ROC EF	7.14	4.72	3.23	10.99	4.45	16.39
5.0% ROC EF	4.73	3.21	2.83	7.28	3.56	9.14

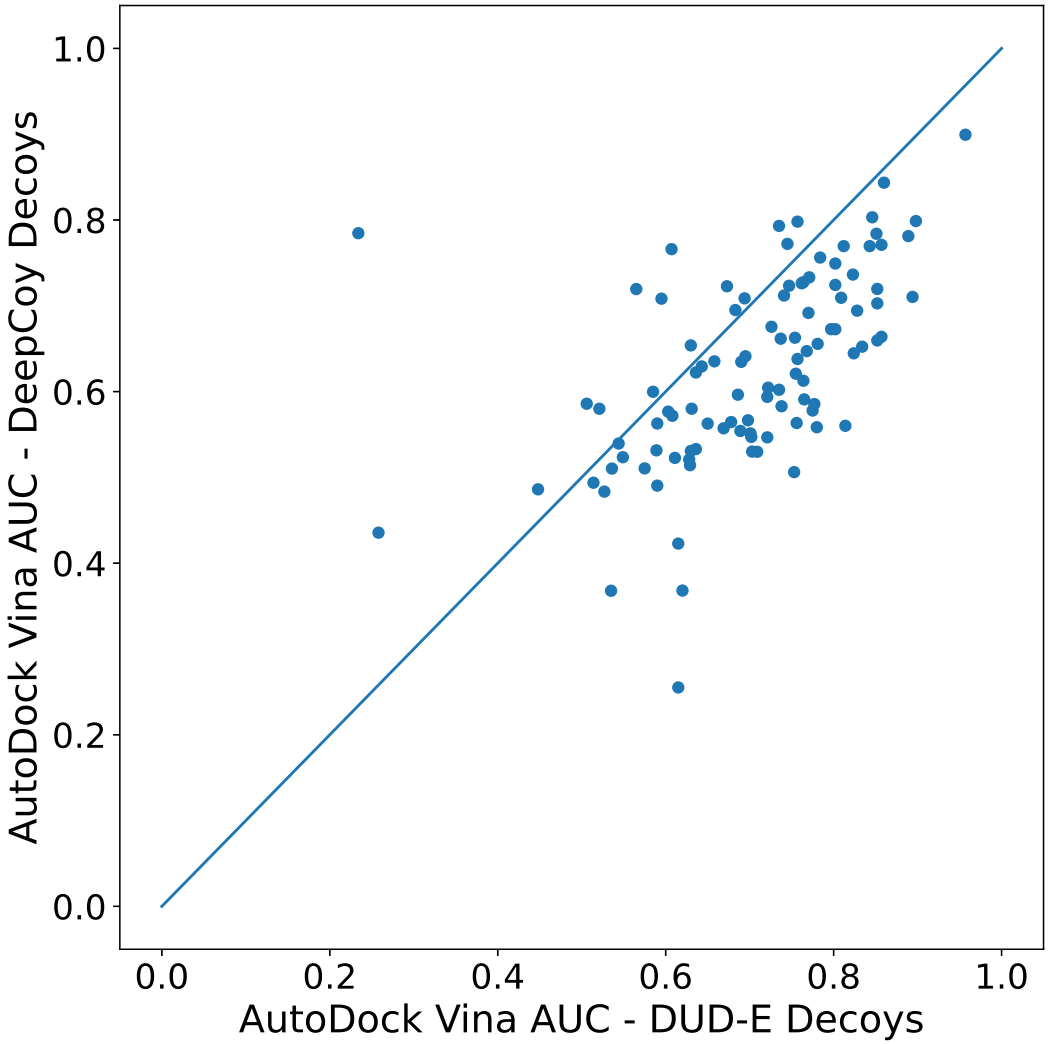


Figure D.11: Virtual screening performance of docking using AutoDock Vina as measured by AUC ROC. The average AUC ROC decreased from 0.70 for the original DUD-E set to 0.63 for the DeepCoy generated decoys.

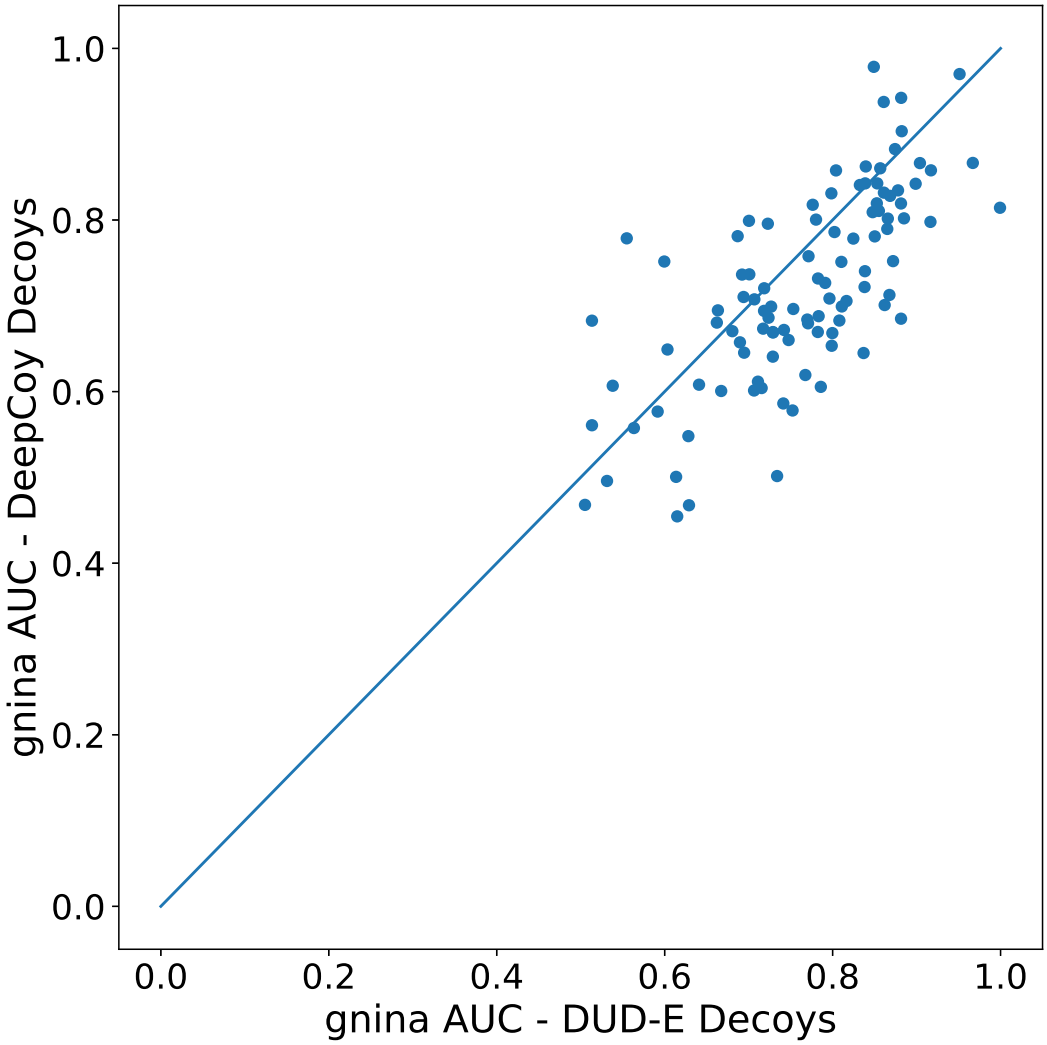


Figure D.12: Virtual screening performance of gnina (Ragoza et al., 2017) as measured by AUC ROC. The average AUC ROC decreased from 0.77 for the original DUD-E set to 0.72 for the DeepCoy generated decoys.

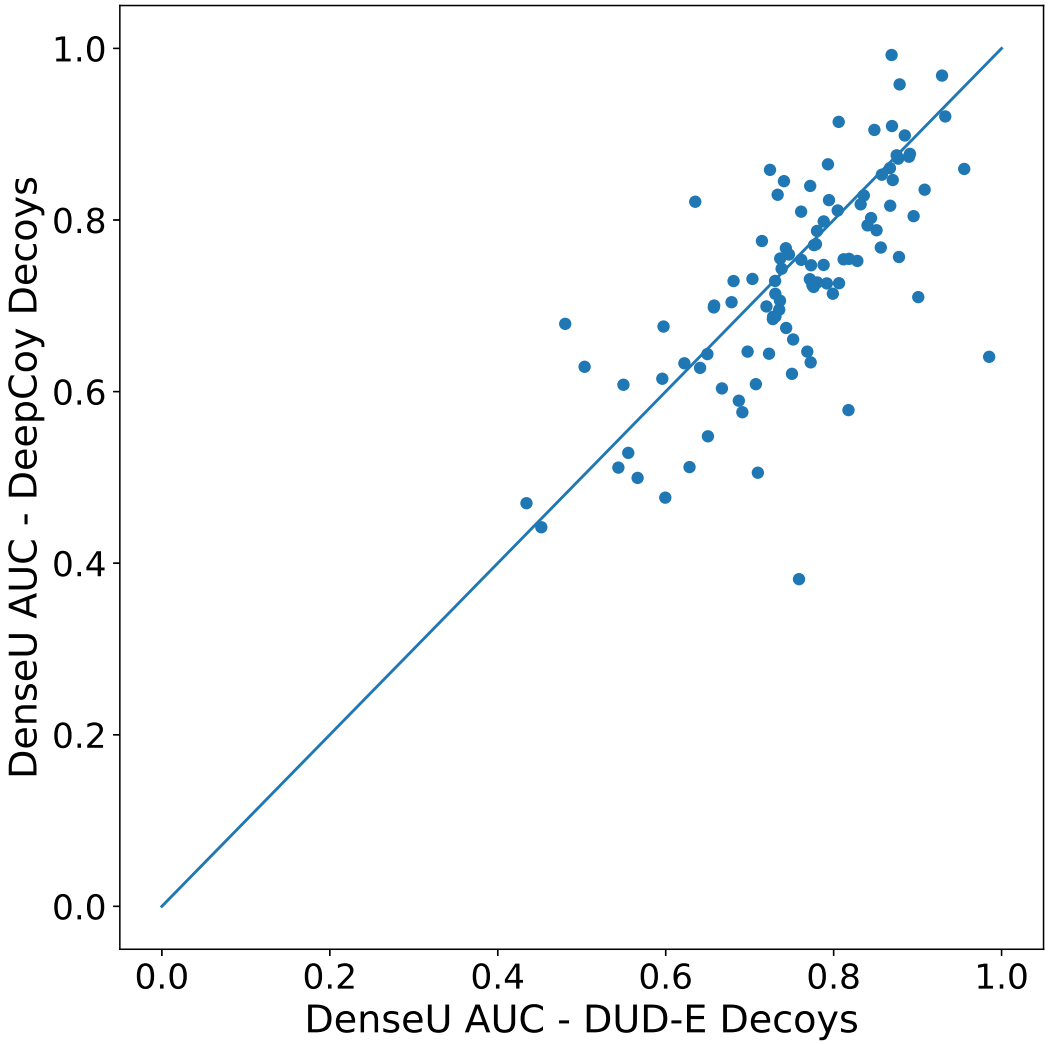


Figure D.13: Virtual screening performance of DenseU (Imrie et al., 2018) as measured by AUC ROC. The average AUC ROC decreased from 0.75 for the original DUD-E set to 0.73 for the DeepCoy generated decoys.

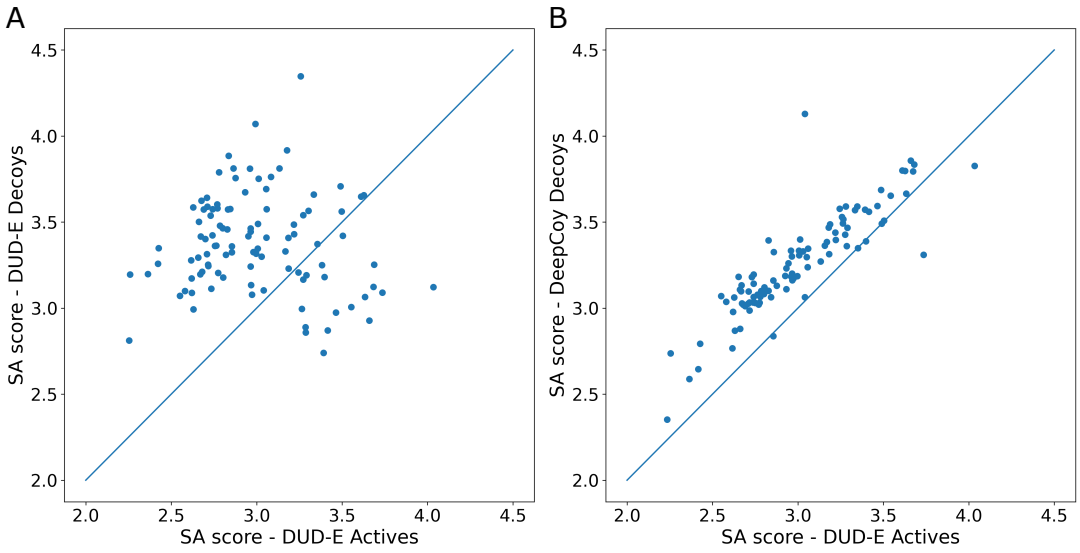


Figure D.14: Average per-target Synthetic accessibility (SA) score of the original DUD-E decoys (A) and the DeepCoy decoys (B) compared to the actives. The average SA score of DeepCoy decoys is highly correlated to the actives (Pearson's R: 0.88), while there is no correlation between the original DUD-E decoys and the active molecules (Pearson's R: -0.07). In addition, the average SA score of the DeepCoy decoys (3.27) more closely matches the SA score of the active molecules (2.99) than the original decoys (3.41).

References

Adeshina, Yusuf O., Eric J. Deeds and John Karanicolas (2020). 'Machine Learning Classification can Reduce False Positives in Structure-Based Virtual Screening'. In: *Proc. Natl. Acad. Sci.* 117.31, pp. 18477–18488. DOI: 10.1073/pnas.2000585117.

Ain, Qurrat Ul, Antoniya Aleksandrova, Florian D. Roessler and Pedro J. Ballester (2015). 'Machine-Learning Scoring Functions to Improve Structure-Based Binding Affinity Prediction and Virtual Screening'. In: *WIREs Comput. Mol. Sci.* 5.6, pp. 405–424. DOI: https://doi.org/10.1002/wcms.1225.

Aldeghi, Matteo, Alexander Heifetz, Michael J. Bodkin, Stefan Knapp and Philip C. Biggin (2016). 'Accurate Calculation of the Absolute Free Energy of Binding for Drug Molecules'. In: *Chem. Sci.* 7 (1), pp. 207–218. DOI: 10.1039/C5SC02678D.

Aldeghi, Matteo, Alexander Heifetz, Michael J. Bodkin, Stefan Knapp and Philip C. Biggin (2017). 'Predictions of Ligand Selectivity from Absolute Binding Free Energy Calculations'. In: *J. Am. Chem. Soc.* 139.2, pp. 946–957. DOI: 10.1021/jacs.6b11467.

Altae-Tran, Han, Bharath Ramsundar, Aneesh S. Pappu and Vijay Pande (2017). 'Low Data Drug Discovery with One-Shot Learning'. In: *ACS Cent. Sci.* 3.4, pp. 283–293. DOI: 10.1021/acscentsci.6b00367.

Anderson, Amy C. (2003). 'The Process of Structure-Based Drug Design'. In: *Chem. Biol.* 10.9, pp. 787 –797. DOI: 10.1016/j.chembiol.2003.09.002.

Andersson, Shalini, Alan Armstrong, Annika Björe, Sue Bowker, Steve Chapman, Rob Davies, Craig Donald, Bryan Egner, Thomas Elebring, Sara Holmqvist, Tord Inghardt, Petra Johannesson, Magnus Johansson, Craig Johnstone, Paul Kemmitt, Jan Kihlberg, Pernilla Korsgren, Malin Lemurell, Jane Moore, Jonas A. Pettersson, Helen Pointon, Fritiof Pontén, Paul Schofield, Nidhal Selmi and Paul Whittamore (2009). 'Making Medicinal Chemistry More Effective—Application of Lean Sigma to Improve Processes, Speed and Quality'. In: *Drug Discovery Today* 14.11, pp. 598 –604. DOI: 10.1016/j.drudis.2009.03.005.

Andreeva, Antonina, Dave Howorth, Cyrus Chothia, Eugene Kulesha and Alexey G. Murzin (2014). 'SCOP2 Prototype: A New Approach to Protein Structure Mining'. In: *Nucleic Acids Res.* 42.D1, pp. D310–D314. DOI: 10.1093/nar/gkt1242.

Arús-Pous, Josep, Atanas Patronov, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen and Ola Engkvist (2020). 'SMILES-Based Deep Generative Scaffold Decorator for De-Novo Drug Design'. In: ChemRxiv preprint:chemrxiv.11638383.v1. DOI: 10.26434/chemrxiv.11638383.v1.

Arús-Pous, Josep, Thomas Blaschke, Silas Ulander, Jean-Louis Reymond, Hongming Chen and Ola Engkvist (2019). 'Exploring the GDB-13 Chemical Space Using Deep Generative Models'. In: *J. Cheminf.* 11.1, p. 20. DOI: 10.1186/s13321-019-0341-z.

Avorn, Jerry (2015). 'The $2.6 Billion Pill — Methodologic and Policy Considerations'. In: *N. Engl. J. Med.* 372.20, pp. 1877–1879. DOI: 10.1056/NEJMp1500848.

Baell, Jonathan B. and Georgina A. Holloway (2010). 'New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays'. In: *J. Med. Chem* 53.7, pp. 2719–2740. DOI: 10.1021/jm901137j.

Bahdanau, Dzmitry, Kyunghyun Cho and Yoshua Bengio (2015). 'Neural Machine Translation by Jointly Learning to Align and Translate'. In: *International Conference on Learning Representations (ICLR)*.

Ballester, Pedro J. (2019). 'Selecting Machine-Learning Scoring Functions for Structure-Based Virtual Screening'. In: *Drug Discovery Today: Technologies* 32-33, pp. 81 –87. DOI: 10.1016/j.ddtec.2020.09.001.

Ballester, Pedro J and John B O Mitchell (2010). 'A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking'. In: *Bioinformatics* 26.9, pp. 1169–1175. DOI: 10.1093/bioinformatics/btq112.A.

Battaglia, Peter, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende and koray kavukcuoglu (2016). 'Interaction Networks for Learning about Objects, Relations and Physics'. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett. Vol. 29. Curran Associates, Inc., pp. 4502–4510.

Bauer, Matthias R., Tamer M. Ibrahim, Simon M. Vogel and Frank M. Boeckler (2013). 'Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets'. In: *J. Chem. Inf. Model.* 53.6, pp. 1447–1462. DOI: 10.1021/ci400115b.

Bemis, Guy W. and Mark A. Murcko (1996). 'The Properties of Known Drugs. 1. Molecular Frameworks'. In: *J. Med. Chem* 39.15, pp. 2887–2893. DOI: 10.1021/jm9602928.

Bento, A. Patricia, Anna Gaulton, Anne Hersey, Bissan Al-Lazikani, David Michalovich, Jon Chambers, Louisa J. Bellis, Mark Davies, Shaun McGlinchey, Yvonne Light and John P. Overington (2011). 'ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery'. In: *Nucleic Acids Res.* 40.D1, pp. D1100–D1107. DOI: 10.1093/nar/gkr777.

Bento, A. Patrícia, Anna Gaulton, Anne Hersey, Louisa J. Bellis, Jon Chambers, Mark Davies, Felix A. Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, Michal Nowotka, George Papadatos, Rita Santos and John P. Overington (2014). 'The ChEMBL Bioactivity Database: An Update'. In: *Nucleic Acids Res.* 42.D1, pp. 1083–1090. DOI: 10.1093/nar/gkt1031.

Berman, Helen M., John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov and Philip E. Bourne (2000). 'The Protein Data Bank'. In: *Nucleic Acids Res.* 28.1, pp. 235–242. DOI: 10.1093/nar/28.1.235.

Besnard, Jérémy, Gian Filippo Ruda, Vincent Setola, Keren Abecassis, Ramona M. Rodriguiz, Xi-Ping Huang, Suzanne Norval, Maria F. Sassano, Antony I. Shin, Lauren A. Webster, Frederick R. C. Simeons, Laste Stojanovski, Annik Prat, Nabil G. Seidah, Daniel B. Constam, G. Richard Bickerton, Kevin D. Read, William C. Wetsel, Ian H. Gilbert, Bryan L. Roth and Andrew L. Hopkins (2012). 'Automated Design of Ligands to Polypharmacological Profiles'. In: *Nature* 492, pp. 215–220. DOI: 10.1038/nature11691.

Bickerton, G. Richard, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan and Andrew L. Hopkins (2012). 'Quantifying the Chemical Beauty of Drugs'. In: *Nat. Chem.* 4.2, pp. 90–98. DOI: 10.1038/nchem.1243.

Bienstock, Rachelle J. (2015). 'Computational Methods for Fragment-Based Ligand Design: Growing and Linking'. In: *Fragment-Based Methods in Drug Discovery*. Ed. by Anthony E. Klon. New York, NY: Springer New York, pp. 119–135. DOI: 10.1007/978-1-4939-2486-8_10.

Blundell, Tom L., Harren Jhoti and Chris Abell (2002). 'High-Throughput Crystallography for Lead Discovery in Drug Design'. In: *Nat. Rev. Drug Discovery* 1.1, pp. 45–54. DOI: 10.1038/nrd706.

Böhm, Hans-Joachim (1992a). 'LUDI: Rule-Based Automatic Design of New Substituents for Enzyme Inhibitor Leads'. In: *J. Comput.-Aided Mol. Des.* 6.6, pp. 593–606. DOI: 10.1007/BF00126217.

Böhm, Hans-Joachim (1992b). 'The Computer Program LUDI: A New Method for the De Novo Design of Enzyme Inhibitors'. In: *J. Comput.-Aided Mol. Des.* 6.1, pp. 61–78.

Böhm, Hans Joachim (1994). 'The Development of a Simple Empirical Scoring Function to Estimate the Binding Constant for a Protein-Ligand Complex of Known Three-Dimensional Structure'. In: *J. Comput.-Aided Mol. Des.* 8.3, pp. 243–256. DOI: 10.1007/BF00126743.

Böhm, Hans-Joachim, Alexander Flohr and Martin Stahl (2004). 'Scaffold Hopping'. In: *Drug Discovery Today: Technol.* 1.3, pp. 217–224.

Bojchevski, Aleksandar, Oleksandr Shchur, Daniel Zügner and Stephan Günnemann (2018). 'NetGAN: Generating Graphs via Random Walks'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 610–619.

Bollini, Mariela, Emilse S. Leal, Natalia S. Adler, María G. Aucar, Gabriela A. Fernández, María J. Pascual, Fernando Merwaiss, Diego E. Alvarez and Claudio N. Cavasotto (2018). 'Discovery of Novel Bovine Viral Diarrhea Inhibitors Using Structure-Based Virtual Screening on the Envelope Protein E2'. In: *Front. Chem.* 6.March, pp. 1–10. DOI: 10.3389/fchem.2018.00079.

Borkin, Dmitry, Shihan He, Hongzhi Miao, Katarzyna Kempinska, Jonathan Pollock, Jennifer Chase, Trupta Purohit, Bhavna Malik, Ting Zhao, Jingya Wang, Bo Wen, Hongliang Zong, Morgan Jones, Gwenn Danet-Desnoyers, Monica L. Guzman, Moshe Talpaz, Dale L. Bixby, Duxin Sun, Jay L. Hess, Andrew G. Muntean, Ivan Maillard, Tomasz Cierpicki and Jolanta Grembecka (2015). 'Pharmacologic Inhibition of the Menin-MLL Interaction Blocks Progression of MLL Leukemia In Vivo'. In: *Cancer Cell* 27.4, pp. 589 –602. DOI: 10.1016/j.ccell.2015.02.016.

Borkin, Dmitry, Jonathan Pollock, Katarzyna Kempinska, Trupta Purohit, Xiaoqin Li, Bo Wen, Ting Zhao, Hongzhi Miao, Shirish Shukla, Miao He, Duxin Sun, Tomasz Cierpicki and Jolanta Grembecka (2016). 'Property Focused Structure-Based Optimization of Small Molecule Inhibitors of the Protein–Protein Interaction between Menin and Mixed Lineage Leukemia (MLL)'. In: *J. Med. Chem.* 59.3, pp. 892–913. DOI: 10.1021/acs.jmedchem.5b01305.

Breuel, Thomas M. (2015). 'The Effects of Hyperparameters on SGD Training of Neural Networks'. In: *arXiv preprint:1508.02788*.

Bronstein, M. M., J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst (2017). 'Geometric Deep Learning: Going beyond Euclidean data'. In: *IEEE Signal Process. Mag.* 34.4, pp. 18–42. DOI: 10.1109/MSP.2017.2693418.

Brown, David (2007). 'Unfinished Business: Target-Based Drug Discovery'. In: *Drug Discovery Today* 12.23, pp. 1007 –1012. DOI: 10.1016/j.drudis.2007.10.017.

Brown, Nathan, Marco Fiscato, Marwin H. S. Segler and Alain C. Vaucher (2019). 'GuacaMol: Benchmarking Models for De Novo Molecular Design'. In: *J. Chem. Inf. Model.* 59.3, pp. 1096–1108. DOI: 10.1021/acs.jcim.8b00839.

Brown, Nathan, Ben McKay, François Gilardoni and Johann Gasteiger (2004). 'A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules'. In: *J. Chem. Inf. Comput. Sci.* 44.3, pp. 1079–1087. DOI: 10.1021/ci034290p.

Buda, Mateusz, Atsuto Maki and Maciej A. Mazurowski (2017). 'A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks'. In: *arXiv preprint:1710.05381*.

Burley, Stephen K, Helen M Berman, Charmi Bhikadiya, Chunxiao Bi, Li Chen, Luigi Di Costanzo, Cole Christie, Ken Dalenberg, Jose M Duarte, Shuchismita Dutta, Zukang Feng, Sutapa Ghosh, David S Goodsell, Rachel K Green, Vladimir Guranović, Dmytro Guzenko, Brian P Hudson, Tara Kalro, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Periskova, Andreas Prlić, Chris Randle, Alexander Rose, Peter Rose, Raul Sala, Monica Sekharan, Chenghua Shao, Lihua Tan, Yi-Ping Tao, Yana Valasatava, Maria Voigt, John Westbrook, Jesse Woo, Huanwang Yang, Jasmine Young, Marina Zhuravleva and Christine Zardecki (2019). 'RCSB Protein Data Bank: Biological Macromolecular Structures Enabling Research and Education in Fundamental Biology, Biomedicine, Biotechnology and Energy'. In: *Nucleic Acids Res.* 47.D1, pp. D464–D474. DOI: 10.1093/nar/gky1004.

Chaput, Ludovic, Juan Martinez-Sanz, Nicolas Saettel and Liliane Mouawad (2016). 'Benchmark of Four Popular Virtual Screening Programs: Construction of the Active/Decoy Dataset Remains a Major Determinant of Measured Performance'. In: *J. Cheminf.* 8.1, p. 56. DOI: 10.1186/s13321-016-0167-x.

Chen, I-Jen and Roderick E. Hubbard (2009). 'Lessons for Fragment Library Design: Analysis of Output from Multiple Screening Campaigns'. In: *J. Comput.-Aided Mol. Des.* 23.8, pp. 603–620. DOI: 10.1007/s10822-009-9280-5.

Chen, Lieyang, Anthony Cruz, Steven Ramsey, Callum J. Dickson, Jose S. Duca, Viktor Hornak, David R. Koes and Tom Kurtzman (2019). 'Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening'. In: *PLOS ONE* 14, pp. 1–22. DOI: 10.1371/journal.pone.0220113.

Chen, Yunpeng, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan and Jiashi Feng (2017). 'Dual Path Networks'. In: *arXiv preprint:1707.01629*. DOI: 10.1007/s11042-017-4764-0.

Cheng, Tiejun, Xun Li, Yan Li, Zhihai Liu and Renxiao Wang (2009). 'Comparative Assessment of Scoring Functions on a Diverse Test Set'. In: *J. Chem. Inf. Model.* 49.4, pp. 1079–1093. DOI: 10.1021/ci9000053.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk and Yoshua Bengio (2014). 'Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation'. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

Cox, Oakley B., Tobias Krojer, Patrick Collins, Octovia Monteiro, Romain Talon, Anthony Bradley, Oleg Fedorov, Jahangir Amin, Brian D. Marsden, John Spencer, Frank von Delft and Paul E. Brennan (2016). 'A poised fragment library enables rapid

synthetic expansion yielding the first reported inhibitors of PHIP(2), an atypical bromodomain'. In: *Chem. Sci.* 7 (3), pp. 2322–2330. DOI: 10.1039/C5SC03115J.

Craik, David J., David P. Fairlie, Spiros Liras and David Price (2013). 'The Future of Peptide-based Drugs'. In: *Chem. Biol. Drug Des.* 81.1, pp. 136–147. DOI: 10.1111/cbdd.12055.

Dandapani, Sivaraman and Lisa A. Marcaurelle (2010). 'Grand Challenge Commentary: Accessing New Chemical Space for 'Undruggable' Targets'. In: *Nat. Chem. Biol.* 6.12, pp. 861–863. DOI: 10.1038/nchembio.479.

Davis, Jesse and Mark Goadrich (2006). 'The Relationship Between Precision-Recall and ROC Curves'. In: *ICML*, pp. 233–240. DOI: 10.1145/1143844.1143874.

Dawson, Natalie L., Tony E. Lewis, Sayoni Das, Jonathan G. Lees, David Lee, Paul Ashford, Christine A. Orengo and Ian Sillitoe (2017). 'CATH: An Expanded Resource to Predict Protein Function Through Structure and Sequence'. In: *Nucleic Acids Res.* 45.D1, pp. D289–D295. DOI: 10.1093/nar/gkw1098.

De Cao, Nicola and Thomas Kipf (2018). 'MolGAN: An Implicit Generative Model for Small Molecular Graphs'. In: *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*.

Deng, J., W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei (2009). 'ImageNet: A large-scale hierarchical image database'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Deng, Wei, Curt Breneman and Mark J. Embrechts (2004). 'Predicting Protein–Ligand Binding Affinities Using Novel Geometrical Descriptors and Machine-Learning Methods'. In: *J. Chem. Inf. Comput. Sci.* 44.2, pp. 699–703. DOI: 10.1021/ci034246+.

Dey, Fabian and Amedeo Caflisch (2008). 'Fragment-Based De Novo Ligand Design by Multiobjective Evolutionary Optimization'. In: *J. Chem. Inf. Model.* 48.3, pp. 679–690. DOI: 10.1021/ci700424b.

DiMasi, Joseph A., Henry G. Grabowski and Ronald W. Hansen (2016). 'Innovation in the Pharmaceutical Industry: New Estimates of R&D Costs'. In: *Journal of Health Economics* 47, pp. 20 –33. DOI: 10.1016/j.jhealeco.2016.01.012.

DiMasi, Joseph A, Ronald W Hansen and Henry G Grabowski (2003). 'The Price of Innovation: New Estimates of Drug Development Costs'. In: *J. Health Econ.* 22.2, pp. 151 –185. DOI: 10.1016/S0167-6296(02)00126-1.

Doman, Thompson N., Susan L. McGovern, Bryan J. Witherbee, Thomas P. Kasten, Ravi Kurumbail, William C. Stallings, Daniel T. Connolly and Brian K. Shoichet (2002). 'Molecular Docking and High-Throughput Screening for Novel Inhibitors of Protein Tyrosine Phosphatase-1B'. In: *J. Med. Chem.* 45.11, pp. 2213–2221. DOI: 10.1021/jm010548w.

Dossetter, Alexander G., Edward J. Griffen and Andrew G. Leach (2013). 'Matched Molecular Pair Analysis in Drug Discovery'. In: *Drug Discovery Today* 18.15, pp. 724 –731. DOI: 10.1016/j.drudis.2013.03.003.

Drozdzal, Michal, Gabriel Chartrand, Eugene Vorontsov, Mahsa Shakeri, Lisa Di Jorio, An Tang, Adriana Romero, Yoshua Bengio, Chris Pal and Samuel Kadoury (2018). 'Learning Normalized Inputs for Iterative Estimation in Medical Image Segmentation'. In: *Med. Image Anal.* 44, pp. 1–13. DOI: 10.1016/j.media.2017.11.005.

Du, Xing, Yi Li, Yuan-Ling Xia, Shi-Meng Ai, Jing Liang, Peng Sang, Xing-Lai Ji and Shu-Qun Liu (2016). 'Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods'. In: *Int. J. Mol. Sci.* 17.2, p. 144. DOI: 10.3390/ijms17020144.

Durrant, Jacob D. and J. Andrew McCammon (2010). 'NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein-Ligand Complexes'. In: *J. Chem. Inf. Model.* 50.10, pp. 1865–1871. DOI: 10.1021/ci100244v.

Durrant, Jacob D. and J. Andrew McCammon (2011). 'NNScore 2.0: A Neural-Network Receptor-Ligand Scoring Function'. In: *J. Chem. Inf. Model.* 51.11, pp. 2897–2903. DOI: 10.1021/ci2003889.

Ebejer, Jean-Paul, Garrett M. Morris and Charlotte M. Deane (2012). 'Freely Available Conformer Generation Methods: How Good Are They?' In: *J. Chem. Inf. Model.* 52.5, pp. 1146–1158. DOI: 10.1021/ci2004658.

Eldridge, M D, C W Murray, T R Auton, G V Paolini and R P Mee (1997). 'Empirical Scoring Functions: I. The Development of a Fast Empirical Scoring Function to Estimate the Binding Affinity of Ligands in Receptor Complexes'. In: *J. Comput.-Aided Mol. Des.* 11.5, pp. 425–445. DOI: Doi10.1023/A:1007996124545.

Elton, Daniel C., Zois Boukouvalas, Mark D. Fuge and Peter W. Chung (2019). 'Deep Learning for Molecular Design—A Review of the State of the Art'. In: *Mol. Syst. Des. Eng.* 4 (4), pp. 828–849. DOI: 10.1039/C9ME00039A.

Erlanson, Daniel A., Stephen W. Fesik, Roderick E. Hubbard, Wolfgang Jahnke and Harren Jhoti (2016). 'Twenty Years On: The Impact of Fragments on Drug Discovery'. In: *Nat. Rev. Drug Discovery* 15, pp. 605–619.

Ernst & Young (2017). 'Beyond Borders: Staying the Course'. In: *Biotechnology Report.*

Ertl, Peter and Ansgar Schuffenhauer (2009). 'Estimation of Synthetic Accessibility Score of Drug-Like Molecules Based on Molecular Complexity and Fragment Contributions'. In: *J. Cheminf.* 1.1, p. 8. DOI: 10.1186/1758-2946-1-8.

Farnaby, William, Manfred Koegl, Michael J. Roy, Claire Whitworth, Emelyne Diers, Nicole Trainor, David Zollman, Steffen Steurer, Jale Karolyi-Oezguer, Carina Riedmueller, Teresa Gmaschitz, Johannes Wachter, Christian Dank, Michael Galant, Bernadette Sharps, Klaus Rumpel, Elisabeth Traxler, Thomas Gerstberger, Renate Schnitzer, Oliver Petermann, Peter Greb, Harald Weinstabl, Gerd Bader, Andreas Zoephel, Alexander Weiss-Puxbaum, Katharina Ehrenhöfer-Wölfer, Simon Wöhrle, Guido Boehmelt, Joerg Rinnenthal, Heribert Arnhof, Nicola Wiechens, Meng-Ying Wu, Tom Owen-Hughes, Peter Ettmayer, Mark Pearson, Darryl B. McConnell and Alessio Ciulli (2019). 'BAF Complex Vulnerabilities in Cancer Demonstrated via Structure-Based PROTAC Design'. In: *Nat. Chem. Biol.* 15.7, pp. 672–680. DOI: 10.1038/s41589-019-0294-6.

Fout, Alex, Jonathon Byrd, Basir Shariat and Asa Ben-Hur (2017). 'Protein Interface Prediction using Graph Convolutional Networks'. In: *Advances in Neural Information Processing Systems.* Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Vol. 30. Curran Associates, Inc., pp. 6530–6539.

Fox, Sandra, Shauna Farr-Jones, Lynne Sopchak, Amy Boggs, Helen Wang Nicely, Richard Khoury and Michael Biros (2006). 'High-Throughput Screening: Update on Practices and Success'. In: *J. Biomol. Screening* 11.7, pp. 864–869. DOI: 10.1177/1087057106292473.

Francoeur, Paul G., Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B. Iovanisci, Ian Snyder and David R. Koes (2020). 'Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design'. In: *J. Chem. Inf. Model.* 60.9, pp. 4200–4215. DOI: 10.1021/acs.jcim.0c00411.

Friesner, Richard A., Jay L. Banks, Robert B. Murphy, Thomas A. Halgren, Jasna J. Klicic, Daniel T. Mainz, Matthew P. Repasky, Eric H. Knoll, Mee Shelley, Jason K. Perry, David E. Shaw, Perry Francis and Peter S. Shenkin (2004). 'Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy'. In: *J. Med. Chem.* 47.7, pp. 1739–1749. DOI: 10.1021/jm0306430.

Friesner, Richard A., Robert B. Murphy, Matthew P. Repasky, Leah L. Frye, Jeremy R. Greenwood, Thomas A. Halgren, Paul C. Sanschagrin and Daniel T. Mainz (2006). 'Extra Precision Glide: Docking and Scoring Incorporating a Model of Hydrophobic Enclosure for Protein-Ligand Complexes'. In: *J. Med. Chem.* 49.21, pp. 6177–6196. DOI: 10.1021/jm051256o.

Fukushima, Kunihiko (1980). 'Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position'. In: *Biol. Cybern.* 36.4, pp. 193–202. DOI: 10.1007/BF00344251.

Gaieb, Zied, Conor D. Parks, Michael Chiu, Huanwang Yang, Chenghua Shao, W. Patrick Walters, Millard H. Lambert, Neysa Nevins, Scott D. Bembenek, Michael K. Ameriks, Tara Mirzadegan, Stephen K. Burley, Rommie E. Amaro and Michael K. Gilson (2019). 'D3R Grand Challenge 3: Blind Prediction of Protein–Ligand Poses and Affinity Rankings'. In: *J. Comput.-Aided Mol. Des.* 33.1, pp. 1–18. DOI: 10.1007/s10822-018-0180-4.

Gatica, Edgar A. and Claudio N. Cavasotto (2012). 'Ligand and Decoy Sets for Docking to G Protein-Coupled Receptors'. In: *J. Chem. Inf. Model.* 52.1, pp. 1–6. DOI: 10.1021/ci200412p.

Gau, David, Taber Lewis, Lee Mcdermott, Peter Wipf, David Koes and Partha Roy (2017). 'Structure-Based Virtual Screening Identifies Small Molecule Inhibitor of the Profilin1-Actin Interaction'. In: *J. Biol. Chem.* 293.7, pp. 2606–2616. DOI: 10.1074/jbc.M117.809137.

Gaulton, Anna, 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 (2016). 'The ChEMBL Database in 2017'. In: *Nucleic Acids Res.* 45.D1, pp. D945–D954. DOI: 10.1093/nar/gkw1074.

Gebauer, Niklas, Michael Gastegger and Kristof Schütt (2019). 'Symmetry-Adapted Generation of 3D Point Sets for the Targeted Discovery of Molecules'. In: *Advances in Neural Information Processing Systems 32*. Vol. 32, pp. 7566–7578.

Gebauer, Niklas W. A., Michael Gastegger and Kristof T. Schütt (2018). 'Generating Equilibrium Molecules with Deep Neural Networks'. In: *NeurIPS Workshop on Machine Learning for Molecules and Materials*.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats and Yann N. Dauphin (2017). 'Convolutional Sequence to Sequence Learning'. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research 70. Ed. by Doina Precup and Yee Whye Teh, pp. 1243–1252.

Gillet, Valerie, A. Peter Johnson, Pauline Mata, Sandor Sike and Philip Williams (1993). 'SPROUT: A Program for Structure Generation'. In: *J. Comput.-Aided Mol. Des.* 7.2, pp. 127–153. DOI: 10.1007/BF00126441.

Gillet, Valerie J., William Newell, Paulina Mata, Glenn Myatt, Sandor Sike, Zsolt Zsoldos and A. Peter Johnson (1994). 'SPROUT: Recent Developments in the

De Novo Design of Molecules'. In: *J. Chem. Inf. Comput. Sci.* 34.1, pp. 207–217. DOI: 10.1021/ci00017a027.

Gilmer, Justin, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals and George E. Dahl (2017). 'Neural Message Passing for Quantum Chemistry'. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 1263–1272.

Goh, Garrett B, Charles Siegel, Abhinav Vishnu, Nathan O Hodas and Nathan Baker (2017). 'Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models'. In: *arXiv preprint:1706.06689*.

Gohlke, Holger, Manfred Hendlich and Gerhard Klebe (2000). 'Knowledge-Based Scoring Function to Predict Protein-Ligand Interactions'. In: *J. Mol. Biol.* 295.2, pp. 337–356. DOI: 10.1006/jmbi.1999.3371.

Gomes, Joseph, Bharath Ramsundar, Evan N. Feinberg and Vijay S. Pande (2017). 'Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity'. In: *arXiv preprint:1703.10603*.

Gómez-Bombarelli, Rafael, 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 (2018). 'Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules'. In: *ACS Cent. Sci.* 4.2, pp. 268–276. DOI: 10.1021/acscentsci.7b00572.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio (2014). 'Generative Adversarial Nets'. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., pp. 2672–2680.

Gorgulla, Christoph, Andras Boeszoermenyi, Zi-Fu Wang, Patrick D. Fischer, Paul W. Coote, Krishna M. Padmanabha Das, Yehor S. Malets, Dmytro S. Radchenko, Yurii S. Moroz, David A. Scott, Konstantin Fackeldey, Moritz Hoffmann, Iryna Iavniuk, Gerhard Wagner and Haribabu Arthanari (2020). 'An Open-Source Drug Discovery Platform Enables Ultra-Large Virtual Screens'. In: *Nature* 580.7805, pp. 663–668. DOI: 10.1038/s41586-020-2117-z.

Grebner, Christoph, Hans Matter, Alleyn T. Plowright and Gerhard Hessler (2020). 'Automated De Novo Design in Medicinal Chemistry: Which Types of Chemistry Does a Generative Neural Network Learn?' In: *J. Med. Chem.* 63.16, pp. 8809–8823. DOI: 10.1021/acs.jmedchem.9b02044.

Green, Clive P, Ola Engkvist and Garry Pairaudeau (2018). 'The Convergence of Artificial Intelligence and Chemistry for Improved Drug Discovery'. In: *Future Med. Chem.* 10.22, pp. 2573–2576. DOI: 10.4155/fmc-2018-0161.

Griffen, Ed, Andrew G. Leach, Graeme R. Robb and Daniel J. Warner (2011). 'Matched Molecular Pairs as a Medicinal Chemistry Tool'. In: *J. Med. Chem.* 54.22, pp. 7739–7750. DOI: 10.1021/jm200452d.

Gu, Jiuxiang, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai and Tsuhan Chen (2018). 'Recent

Advances in Convolutional Neural Networks'. In: *Pattern Recognition* 77, pp. 354–377. DOI: 10.1016/j.patcog.2017.10.013.

Guedes, Isabella A., Felipe S. S. Pereira and Laurent E. Dardenne (2018). 'Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges'. In: *Front. Pharmacol.* 9, p. 1089. DOI: 10.3389/fphar.2018.01089.

Guha, Rajarshi (2013). 'On Exploring Structure–Activity Relationships'. In: *In Silico Models for Drug Discovery*. Ed. by Sandhya Kortagere. Totowa, NJ: Humana Press, pp. 81–94. DOI: 10.1007/978-1-62703-342-8_6.

Guimaraes, Gabriel Lima, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias and Alán Aspuru-Guzik (2017). 'Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models'. In: *CoRR* arXiv preprint:1705.10843.

Gupta, Anvita, Alex T. Müller, Berend J. H. Huisman, Jens A. Fuchs, Petra Schneider and Gisbert Schneider (2018). 'Generative Recurrent Networks for De Novo Drug Design'. In: *Mol. Inf.* 37.1-2, p. 1700111. DOI: 10.1002/minf.201700111.

Hajduk, Philip J. and Jonathan Greer (2007). 'A Decade of Fragment-Based Drug Design: Strategic Advances and Lessons Learned'. In: *Nat. Rev. Drug Discovery* 6.3, pp. 211–219. DOI: 10.1038/nrd2220.

Hajduk, Philip J. and Daryl R. Sauer (2008). 'Statistical Analysis of the Effects of Common Chemical Substituents on Ligand Potency'. In: *J. Med. Chem.* 51.3, pp. 553–564. DOI: 10.1021/jm070838y.

Halevy, Alon, Peter Norvig and Fernando Pereira (2009). 'The Unreasonable Effectiveness of Data'. In: *IEEE Intell. Syst.* 24, pp. 8–12.

Hamilton, Will, Zhitao Ying and Jure Leskovec (2017). 'Inductive Representation Learning on Large Graphs'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Vol. 30. Curran Associates, Inc., pp. 1024–1034.

Hanley, A.J. and J.B. McNeil (1982). 'The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve'. In: *Radiology* 143, pp. 29–36. DOI: 10.1148/radiology.143.1.7063747.

Hansch, Corwin, Peyton P. Maloney, Toshio Fujita and Rober M. Muir (1962). 'Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients'. In: *Nature* 194.4824, pp. 178–180. DOI: 10.1038/194178b0.

Harrison, Richard K. (2016). 'Phase II and phase III failures: 2013-2015'. In: *Nat. Rev. Drug Discovery* 15, pp. 817–818.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun (2015). 'Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification'. In: *ICCV* 2015 Inter, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun (2016). 'Deep Residual Learning for Image Recognition'. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heikamp, Kathrin and Jürgen Bajorath (2011). 'Large-Scale Similarity Search Profiling of ChEMBL Compound Data Sets'. In: *J. Chem. Inf. Model.* 51.8, pp. 1831–1839. DOI: 10.1021/ci200199u.

Hillisch, Alexander, Luis Felipe Pineda and Rolf Hilgenfeld (2004). 'Utility of Homology Models in the Drug Discovery Process'. In: *Drug Discovery Today* 9.15, pp. 659 –669. DOI: 10.1016/S1359-6446(04)03196-4.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long Short-Term Memory'. In: *Neural Comput.* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Hochuli, Joshua, Alec Helbling, Tamar Skaist, Matthew Ragoza and David Ryan Koes (2018). 'Visualizing Convolutional Neural Network Protein-Ligand Scoring'. In: *J. Mol. Graphics Modell.* 84, pp. 96 –108. DOI: 10.1016/j.jmgm.2018.06.005.

Hodge, Victoria J. and Jim Austin (2004). 'A Survey of Outlier Detection Methodologies'. In: *Artif. Intell. Rev.* 22.2, pp. 85–126. DOI: 10.1007/s10462-004-4304-y.

Hopkins, Andrew L., Colin R. Groom and Alexander Alex (2004). 'Ligand Efficiency: A Useful Metric for Lead Selection'. In: *Drug Discovery Today* 9.10, pp. 430 –431. DOI: 10.1016/S1359-6446(04)03069-7.

Hu, Jie, Li Shen and Gang Sun (2017). 'Squeeze-and-Excitation Networks'. In: *arXiv preprint:1709.01507*.

Huang, Gao, Zhuang Liu, Kilian Q. Weinberger and Laurens van der Maaten (2016). 'Densely Connected Convolutional Networks'. In: *arXiv preprint:1608.06993*. DOI: 10.1109/CVPR.2017.243.

Huang, Niu, Brian K. Shoichet and John J. Irwin (2006). 'Benchmarking Sets for Molecular Docking'. In: *J. Med. Chem.* 49.23, pp. 6789–6801. DOI: 10.1021/jm0608356.

Huang, Ying, Beifang Niu, Ying Gao, Limin Fu and Weizhong Li (2010). 'CD-HIT Suite: A Web Server for Clustering and Comparing Biological Sequences'. In: *Bioinformatics* 26.5, pp. 680–682. DOI: 10.1093/bioinformatics/btq003.

Huggins, David J., Philip C. Biggin, Marc A. Dämgen, Jonathan W. Essex, Sarah A. Harris, Richard H. Henchman, Syma Khalid, Antonija Kuzmanic, Charles A. Laughton, Julien Michel, Adrian J. Mulholland, Edina Rosta, Mark S. P. Sansom and Marc W. van der Kamp (2019). 'Biomolecular Simulations: From Dynamics and Mechanisms to Computational Assays of Biological Activity'. In: *WIREs Comput. Mol. Sci.* 9.3, e1393. DOI: 10.1002/wcms.1393.

Hughes, J. P., S. Rees, S. B. Kalindjian and K. L. Philpott (2011). 'Principles of Early Drug Discovery'. In: *Br. J. Pharmacol.* 162.6, pp. 1239–1249. DOI: 10.1111/j.1476-5381.2010.01127.x.

Hummel, Gerd, Ulrich Reineke and Ulf Reimer (2006). 'Translating peptides into small molecules'. In: *Mol. BioSyst.* 2 (10), pp. 499–508. DOI: 10.1039/B611791K.

Hussain, Jameed and Ceara Rea (2010). 'Computationally Efficient Algorithm to Identify Matched Molecular Pairs (MMPs) in Large Data Sets'. In: *J. Chem. Inf. Model.* 50.3, pp. 339–348. DOI: 10.1021/ci900450m.

Ichihara, Osamu, John Barker, Richard J. Law and Mark Whittaker (2011). 'Compound Design by Fragment-Linking'. In: *Mol. Inf.* 30.4, pp. 298–306. DOI: 10.1002/minf.201000174.

Imrie, Fergus, Anthony R. Bradley and Charlotte M. Deane (2021a). 'Chapter 8 Virtual Screening with Convolutional Neural Networks'. In: *Artificial Intelligence in Drug Discovery*. Ed. by Nathan Brown. The Royal Society of Chemistry, pp. 151–183. DOI: 10.1039/9781788016841-00151.

Imrie, Fergus, Anthony R. Bradley and Charlotte M. Deane (2021b). 'Generating property-matched decoy molecules using deep learning'. In: *Bioinformatics*. DOI: 10.1093/bioinformatics/btab080.

Imrie, Fergus, Anthony R. Bradley, Mihaela van der Schaar and Charlotte M. Deane (2018). 'Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data'. In: *J. Chem. Inf. Model.* 58.11, pp. 2319–2330. DOI: 10.1021/acs.jcim.8b00350.

Imrie, Fergus, Anthony R. Bradley, Mihaela van der Schaar and Charlotte M. Deane (2020). 'Deep Generative Models for 3D Linker Design'. In: *J. Chem. Inf. Model.* 60.4, pp. 1983–1995. DOI: 10.1021/acs.jcim.9b01120.

Inglese, James, Ronald L. Johnson, Anton Simeonov, Menghang Xia, Wei Zheng, Christopher P. Austin and Douglas S. Auld (2007). 'High-Throughput Screening Assays for the Identification of Chemical Probes'. In: *Nat. Chem. Biol.* 3.8, pp. 466–479. DOI: 10.1038/nchembio.2007.17.

Ioffe, Sergey and Christian Szegedy (2015). 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 448–456.

Irwin, John J. and Brian K. Shoichet (2005). 'ZINC - A Free Database of Commercially Available Compounds for Virtual Screening'. In: *J. Chem. Inf. Model.* 45.1, pp. 177–182. DOI: 10.1021/ci049714+.

Itti, L., C. Koch and E. Niebur (1998). 'A Model of Saliency-Based Visual Attention for Rapid Scene Analysis'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11, pp. 1254–1259. DOI: 10.1109/34.730558.

Jain, Ajay N. and Anthony Nicholls (2008). 'Recommendations for Evaluation of Computational Methods'. In: *J. Comput.-Aided Mol. Des.* 22.3-4, pp. 133–139. DOI: 10.1007/s10822-008-9196-5.

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama and Trevor Darrell (2014). 'Caffe: Convolutional Architecture for Fast Feature Embedding'. In: *arXiv preprint:1408.5093*. DOI: 10.1145/2647868.2654889.

Jiménez, José, Miha Škalič, Gerard Martínez-Rosell and Gianni De Fabritiis (2018). 'KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks'. In: *J. Chem. Inf. Model.* 58.2, pp. 287–296. DOI: 10.1021/acs.jcim.7b00650.

Jin, Wengong, Regina Barzilay and Tommi Jaakkola (2019a). 'Hierarchical Graph-to-Graph Translation for Molecules'. In: arXiv preprint:1907.11223.

Jin, Wengong, Regina Barzilay and Tommi S. Jaakkola (2018). 'Junction Tree Variational Autoencoder for Molecular Graph Generation'. In: *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research 80, pp. 2323–2332.

Jin, Wengong, Kevin Yang, Regina Barzilay and Tommi Jaakkola (2019b). 'Learning Multimodal Graph-to-Graph Translation for Molecule Optimization'. In: *International Conference on Learning Representations (ICLR)*.

Jones, Gareth, Peter Willett, Robert C Glen, Andrew R Leach and Robin Taylor (1997). 'Development and Validation of a Genetic Algorithm for Flexible Docking'. In: *J. Mol. Biol.* 267.3, pp. 727 –748. DOI: 10.1006/jmbi.1996.0897.

Jumper, John, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Anna Potapenko, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov,

Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Martin Steinegger, Michalina Pacholska, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli and Demis Hassabis (2020). 'High Accuracy Protein Structure Prediction Using Deep Learning'. In: *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*.

Kamenecka, Ted, Jeff Habel, Derek Duckett, Weimin Chen, Yuan Yuan Ling, Bozena Frackowiak, Rong Jiang, Youseung Shin, Xinyi Song and Philip LoGrasso (2009). 'Structure-Activity Relationships and X-ray Structures Describing the Selectivity of Aminopyrazole Inhibitors for c-Jun N-terminal Kinase 3 (JNK3) over p38'. In: *J. Biol. Chem.* 284.19, pp. 12853–12861. DOI: 10.1074/jbc.M809430200.

Kaplan, Warren, Veronika J Wirtz, Aukje Mantel-Teeuwisse, Pieter Stolk, Béatrice Duthey and Richard Laing (2013). *Priority Medicines for Europe and the World 2013 Update*. https://www.who.int/medicines/areas/priority_medicines/MasterDocJune28_FINAL_Web.pdf. Accessed: 2020-12-10.

Kearnes, Steven, Kevin McCloskey, Marc Berndl, Vijay Pande and Patrick Riley (2016). 'Molecular Graph Convolutions: Moving Beyond Fingerprints'. In: *J. Comput.-Aided Mol. Des.* 30.8, pp. 595–608. DOI: 10.1007/s10822-016-9938-8.

Keserű, György M., Daniel A. Erlanson, György G. Ferenczy, Michael M. Hann, Christopher W. Murray and Stephen D. Pickett (2016). 'Design Principles for Fragment Libraries: Maximizing the Value of Learnings from Pharma Fragment-Based Drug Discovery (FBDD) Programs for Use in Academia'. In: *J. Med. Chem.* 59.18, pp. 8189–8206. DOI: 10.1021/acs.jmedchem.6b00197.

Khanna, Ish (2012). 'Drug Discovery in Pharmaceutical Industry: Productivity Challenges and Trends'. In: *Drug Discovery Today* 17.19, pp. 1088 –1102. DOI: 10.1016/j.drudis.2012.05.007.

Kim, Sunghwan, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang and Stephen H. Bryant (2015). 'PubChem Substance and Compound Databases'. In: *Nucleic Acids Res.* 44.D1, pp. D1202–D1213. DOI: 10.1093/nar/gkv951.

Kingma, Diederik P. and Jimmy Ba (2015). 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kipf, Thomas N. and Max Welling (2017). 'Semi-Supervised Classification with Graph Convolutional Networks'. In: *International Conference on Learning Representations (ICLR)*.

Kiss, R, B Kiss, A Konczol, F Szalai, I Jelinek, V Laszlo, B Noszal, A Falus and G M Keseru (2008). 'Discovery of Novel Human Histamine H4 Receptor Ligands by Large-Scale Structure-Based Virtual Screening'. In: *J. Med. Chem.* 51.11, pp. 3145–3153. DOI: 10.1021/jm7014777.

Klambauer, Günter, Sepp Hochreiter and Matthias Rarey (2019). 'Machine Learning in Drug Discovery'. In: *J. Chem. Inf. Model.* 59.3, pp. 945–946. DOI: 10.1021/acs.jcim.9b00136.

Klebe, Gerhard (2006). 'Virtual Ligand Screening: Strategies, Perspectives and Limitations'. In: *Drug Discovery Today* 11.13, pp. 580 –594. DOI: 10.1016/j.drudis.2006.05.012.

Koes, David Ryan, Matthew P. Baumgartner and Carlos J. Camacho (2013). 'Lessons Learned in Empirical Scoring with smina From the CSAR 2011 Benchmarking Exercise'. In: *J. Chem. Inf. Model.* 53.8, pp. 1893–1904. DOI: 10.1021/ci300604z.

Konc, Janez and Dušanka Janežič (2010). 'ProBiS Algorithm for Detection of Structurally Similar Protein Binding Sites by Local Structural Alignment'. In: *Bioinformatics* 26.9, pp. 1160–1168. DOI: 10.1093/bioinformatics/btq100.

Kramer, Jeffrey A., John E. Sagartz and Dale L. Morris (2007). 'The Application of Discovery Toxicology and Pathology Towards the Design of Safer Pharmaceutical Lead Candidates'. In: *Nat. Rev. Drug Discovery* 6. Review Article, pp. 636–649.

Krenn, Mario, Florian Häse, AkshatKumar Nigam, Pascal Friederich and Alán Aspuru-Guzik (2019). 'SELFIES: A Robust Representation of Semantically Constrained Graphs with an Example Application in Chemistry'. In: *CoRR* abs/1905.13741.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'ImageNet Classification with Deep Convolutional Neural Networks'. In: *NIPS*, pp. 1097–1105. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

Kryshtafovych, Andriy, Torsten Schwede, Maya Topf, Krzysztof Fidelis and John Moult (2019). 'Critical Assessment of Methods of Protein Structure Prediction (CASP)—Round XIII'. In: *Proteins: Structure, Function, and Bioinformatics* 87.12, pp. 1011–1020. DOI: 10.1002/prot.25823.

Kurosawa, Gene, Yasushi Akahori, Miwa Morita, Mariko Sumitomo, Noriko Sato, Chiho Muramatsu, Keiko Eguchi, Kazuki Matsuda, Akihiko Takasaki, Miho Tanaka, Yoshitaka Iba, Susumu Hamada-Tsutsumi, Yoshinori Ukai, Mamoru Shiraishi, Kazuhiro Suzuki, Maiko Kurosawa, Sally Fujiyama, Nobuhiro Takahashi, Ryoichi Kato, Yoshikazu Mizoguchi, Mikihiro Shamoto, Hiroyuki Tsuda, Mototaka Sugiura, Yoshinobu Hattori, Shuichi Miyakawa, Ryoichi Shiroki, Kiyotaka Hoshinaga, Nobuhiro Hayashi, Atsushi Sugioka and Yoshikazu Kurosawa (2008). 'Comprehensive Screening for Antigens Overexpressed on Carcinomas via Isolation of Human mAbs that may be Therapeutic'. In: *Proc. Natl. Acad. Sci. U. S. A.* 105.20, pp. 7287–7292. DOI: 10.1073/pnas.0712202105.

Lagarde, Nathalie, Jean-François Zagury and Matthieu Montes (2015a). 'Benchmarking Data Sets for the Evaluation of Virtual Ligand Screening Methods: Review and Perspectives'. In: *J. Chem. Inf. Model.* 55.7, pp. 1297–1307. DOI: 10.1021/acs.jcim.5b00090.

Lagarde, Nathalie, Jean-François Zagury and Matthieu Montes (2015b). 'Benchmarking Data Sets for the Evaluation of Virtual Ligand Screening Methods: Review and Perspectives'. In: *J. Chem. Inf. Model.* 55.7, pp. 1297–1307. DOI: 10.1021/acs.jcim.5b00090.

Lahana, Roger (1999). 'How Many Leads From HTS?' In: *Drug Discovery Today* 4.10, pp. 447 –448. DOI: 10.1016/S1359-6446(99)01393-8.

Lajiness, Michael S., Gerald M. Maggiora and Veerabahu Shanmugasundaram (2004). 'Assessment of the Consistency of Medicinal Chemists in Reviewing Sets of Compounds'. In: *J. Med. Chem.* 47.20, pp. 4891–4896. DOI: 10.1021/jm049740z.

Lamoree, Bas and Roderick E. Hubbard (2017). 'Current Perspectives in Fragment-Based Lead Discovery (FBLD)'. In: *Essays Biochem.* 61.5, pp. 453–464. DOI: 10.1042/EBC20170028.

Landrum, Greg (2006). *RDKit: Open-Source Cheminformatics*. http://www.rdkit.org/.

Landrum, Gregory A., Julie E. Penzotti and Santosh Putta (2006). 'Feature-map Vectors: A New Class of Informative Descriptors for Computational Drug Discovery'. In: *J. Comput.-Aided Mol. Des.* 20.12, pp. 751–762. DOI: 10.1007/s10822-006-9085-8.

Langdon, Sarah R., Peter Ertl and Nathan Brown (2010). 'Bioisosteric Replacement and Scaffold Hopping in Lead Generation and Optimization'. In: *Mol. Inf.* 29.5, pp. 366–385. DOI: 10.1002/minf.201000019.

Lazebnik, S., C. Schmid and J. Ponce (2006). 'Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories'. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2, pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.

Leach, Andrew R., Valerie J. Gillet, Richard A. Lewis and Robin Taylor (2010). 'Three-Dimensional Pharmacophore Methods in Drug Discovery'. In: *J. Med. Chem.* 53.2, pp. 539–558. DOI: 10.1021/jm900817u.

Lecun, Y., L. Bottou, Y. Bengio and P. Haffner (1998). 'Gradient-Based Learning Applied to Document Recognition'. In: *Proc. IEEE* 86.11, pp. 2278–2324. DOI: 10.1109/5.726791.

LeCun, Yann, Patrick Haffner, Léon Bottou and Yoshua Bengio (1999). 'Object Recognition with Gradient-Based Learning'. In: *Shape, Contour and Grouping in Computer Vision*. London, UK, UK: Springer-Verlag, pp. 319–.

Lee, Chen-Yu, Patrick W. Gallagher and Zhuowen Tu (2016). 'Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, pp. 464–472.

Leelananda, Sumudu P. and Steffen Lindert (2016). 'Computational Methods in Drug Discovery'. In: *Beilstein J. Org. Chem.* 12. Ed. by Helge B. Bode, pp. 2694–2718. DOI: 10.3762/bjoc.12.267.

Li, Hongjian, Kwong-Sak Leung, Man-Hon Wong and Pedro J. Ballester (2014a). 'Substituting Random Forest for Multiple Linear Regression Improves Binding Affinity Prediction of Scoring Functions: Cyscore as a Case Study'. In: *BMC Bioinf.* 15.1, p. 291. DOI: 10.1186/1471-2105-15-291.

Li, Hongjian, Kwong-Sak Leung, Man-Hon Wong and Pedro J. Ballester (2015). 'The Importance of the Regression Model in the Structure-Based Prediction of Protein-Ligand Binding'. In: *Computational Intelligence Methods for Bioinformatics and Biostatistics. CIBB 2014. Lecture Notes in Computer Science* 8623, pp. 219–230. DOI: 10.1007/978-3-319-24462-4_19.

Li, Hongjian, Kam-Heung Sze, Gang Lu and Pedro J. Ballester (2020a). 'Machine-Learning Scoring Functions for Structure-Based Virtual Screening'. In: *WIREs Comput. Mol. Sci.*, e1478. DOI: 10.1002/wcms.1478.

Li, Jie and Jieqing Liu (2020). 'PROTAC: A Novel Technology for Drug Development**'. In: *ChemistrySelect* 5.42, pp. 13232–13247. DOI: 10.1002/slct.202003162.

Li, Xiaomeng, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu and Pheng Ann Heng (2017). 'H-DenseUNet: Hybrid Densely Connected UNet for Liver and Liver Tumor Segmentation from CT Volumes'. In: *arXiv preprint:1709.07330*.

Li, Yan, Li Han, Zhihai Liu and Renxiao Wang (2014b). 'Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results'. In: *J. Chem. Inf. Model.* 54.6, pp. 1717–1736. DOI: 10.1021/ci500081m.

Li, Yibo, Jianxing Hu, Yanxing Wang, Jielong Zhou, Liangren Zhang and Zhenming Liu (2020b). 'DeepScaffold: A Comprehensive Tool for Scaffold-Based De Novo Drug Discovery Using Deep Learning'. In: *J. Chem. Inf. Model.* 60.1, pp. 77–91. DOI: 10.1021/acs.jcim.9b00727.

Li, Yujia, Daniel Tarlow, Marc Brockschmidt and Richard Zemel (2016). 'Gated Graph Sequence Neural Networks'. In: *International Conference on Learning Representations (ICLR)*.

Li, Yujia, Oriol Vinyals, Chris Dyer, Razvan Pascanu and Peter Battaglia (2018). 'Learning Deep Generative Models of Graphs'. In: *CoRR* arXiv preprint:1803.03324.

Lim, Jaechang, Sang-Yeon Hwang, Seokhyun Moon, Seungsu Kim and Woo Youn Kim (2020). 'Scaffold-based molecular design with a graph generative model'. In: *Chem. Sci.* 11 (4), pp. 1153–1164. DOI: 10.1039/C9SC04503A.

Liu, Qi, Miltiadis Allamanis, Marc Brockschmidt and Alexander Gaunt (2018). 'Constrained Graph Variational Autoencoders for Molecule Design'. In: *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pp. 7795–7804.

Liu, Shengchao, Moayad Alnammi, Spencer S. Ericksen, Andrew F. Voter, Gene E. Ananiev, James L. Keck, F. Michael Hoffmann, Scott A. Wildman and Anthony Gitter (2019). 'Practical Model Selection for Prospective Virtual Screening'. In: *J. Chem. Inf. Model.* 59.1, pp. 282–293. DOI: 10.1021/acs.jcim.8b00363.

Liu, Tiqing, Yuhmei Lin, Xin Wen, Robert N. Jorissen and Michael K. Gilson (2006). 'BindingDB: A Web-Accessible Database of Experimentally Determined Protein-Ligand Binding Affinities'. In: *Nucleic Acids Res.* 35.suppl_1, pp. D198–D201. DOI: 10.1093/nar/gkl999.

Liu, Zhihai, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li and Renxiao Wang (2017). 'Forging the Basis for Developing Protein–Ligand Interaction Scoring Functions'. In: *Acc. Chem. Res.* 50.2, pp. 302–309. DOI: 10.1021/acs.accounts.6b00491.

Lloyd, S. (1982). 'Least Squares Quantization in PCM'. In: *IEEE Trans. Inf. Theory* 28.2, pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

Lowerre, Bruce T. (1976). 'The Harpy Speech Recognition System.' PhD thesis. USA.

Lyu, Jiankun, Sheng Wang, Trent E. Balius, Isha Singh, Anat Levit, Yurii S. Moroz, Matthew J. O'Meara, Tao Che, Enkhjargal Algaa, Kateryna Tolmachova, Andrey A. Tolmachev, Brian K. Shoichet, Bryan L. Roth and John J. Irwin (2019). 'Ultra-Large Library Docking for Discovering New Chemotypes'. In: *Nature* 566.7743, pp. 224–229. DOI: 10.1038/s41586-019-0917-9.

Ma, Junshui, Robert P. Sheridan, Andy Liaw, George E. Dahl and Vladimir Svetnik (2015). 'Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships'. In: *J. Chem. Inf. Model.* 55.2, pp. 263–274. DOI: 10.1021/ci500747n.

Maas, Andrew L., Awni Y. Hannun and Andrew Y. Ng (2013). 'Rectifier Nonlinearities Improve Neural Network Acoustic Models'. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Maass, Patrick, Tanja Schulz-Gasch, Martin Stahl and Matthias Rarey (2007). 'Recore: A Fast and Versatile Method for Scaffold Hopping Based on Small Molecule Crystal Structure Conformations'. In: *J. Chem. Inf. Model.* 47.2, pp. 390–399. DOI: 10.1021/ci060094h.

Maaten, Laurens van der, Eric Postma and H. Herik (2007). 'Dimensionality Reduction: A Comparative Review'. In: *J. Mach. Learn. Res.* 10.

Macalino, Stephani Joy Y., Vijayakumar Gosu, Sunhye Hong and Sun Choi (2015). 'Role of Computer-Aided Drug Design in Modern Drug Discovery'. In: *Arch. Pharm. Res.* 38.9, pp. 1686–1701. DOI: 10.1007/s12272-015-0640-5.

Madhavi Sastry, G., Matvey Adzhigirey, Tyler Day, Ramakrishna Annabhimoju and Woody Sherman (2013). 'Protein and Ligand Preparation: Parameters, Protocols, and Influence on Virtual Screening Enrichments'. In: *J. Comput.-Aided Mol. Des.* 27.3, pp. 221–234. DOI: 10.1007/s10822-013-9644-8.

Mahmoud, Amr H., Matthew R. Masters, Ying Yang and Markus A. Lill (2020). 'Elucidating the Multiple Roles of Hydration for Accurate Protein-Ligand Binding Prediction via Deep Learning'. In: *Commun. Chem.* 3.1, p. 19. DOI: 10.1038/s42004-020-0261-x.

Malhotra, Shipra and John Karanicolas (2017). 'When Does Chemical Elaboration Induce a Ligand To Change Its Binding Mode?' In: *J. Med. Chem.* 60.1, pp. 128–145. DOI: 10.1021/acs.jmedchem.6b00725.

Masuda, Tomohide, Matthew Ragoza and David Ryan Koes (2020). *Generating 3D Molecular Structures Conditional on a Receptor Binding Site with Deep Generative Models*.

Maudsley, DB (1979). 'A Theory of Meta-Learning and Principles of Facilitation'. In: *University of Toronto*.

Mayr, Andreas, Günter Klambauer, Thomas Unterthiner and Sepp Hochreiter (2016). 'DeepTox: Toxicity Prediction using Deep Learning'. In: *Frontiers in Environmental Science* 3, p. 80. DOI: 10.3389/fenvs.2015.00080.

McGregor, Malcolm J. and Peter V. Pallai (1997). 'Clustering of Large Databases of Compounds: Using the MDL "Keys" as Structural Descriptors'. In: *J. Chem. Inf. Model.* 37.3, pp. 443–448. DOI: 10.1021/ci960151e.

Mendez, David, 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 (2019). 'ChEMBL: Towards Direct Deposition of Bioassay Data'. In: *Nucleic Acids Res.* 47.D1, pp. D930–D940. DOI: 10.1093/nar/gky1075.

Mitchell, Tom M (1997). *Machine learning*. McGraw-Hill.

Miyazawa, Sanzo and Robert L. Jernigan (1985). 'Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation'. In: *Macromolecules* 18.3, pp. 534–552. DOI: 10.1021/ma00145a039.

Méndez-Lucio, Oscar and José L. Medina-Franco (2017). 'The Many Roles of Molecular Complexity in Drug Discovery'. In: *Drug Discovery Today* 22.1, pp. 120 –126. DOI: 10.1016/j.drudis.2016.08.009.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg and Demis Hassabis (2015). 'Human-Level Control Through Deep Reinforcement Learning'. In: *Nature* 518.7540, pp. 529–533. DOI: 10.1038/nature14236.

Moors, Ellen H.M., Adam F. Cohen and Huub Schellekens (2014). 'Towards a Sustainable System of Drug Development'. In: *Drug Discovery Today* 19.11, pp. 1711 –1720. DOI: 10.1016/j.drudis.2014.03.004.

Morgan, H. L. (1965). 'The Generation of a Unique Machine Description for Chemical Structures - A Technique Developed at Chemical Abstracts Service.' In: *J. Chem. Doc.* 5.2, pp. 107–113. DOI: 10.1021/c160017a018.

Murphy, Kevin (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Murray, Christopher W. and David C. Rees (2009). 'The Rise of Fragment-Based Drug Discovery'. In: *Nat. Chem.* 1. Perspective, pp. 187–192.

Myers, Scott and Ann Baker (2001). 'Drug Discovery—An Operating Model for a New Era'. In: *Nat. Biotechnol.* 19.8, pp. 727–730. DOI: 10.1038/90765.

Mysinger, Michael M., Michael Carchia, John J. Irwin and Brian K. Shoichet (2012). 'Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking'. In: *J. Med. Chem.* 14, pp. 6582–6594.

Nair, Vinod and Geoffrey E. Hinton (2010). 'Rectified Linear Units Improve Restricted Boltzmann Machines'. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. USA: Omnipress, pp. 807–814.

Nelson, Aaron L., Eugen Dhimolea and Janice M. Reichert (2010). 'Development Trends for Human Monoclonal Antibody Therapeutics'. In: *Nat. Rev. Drug Discovery* 9.10, pp. 767–774. DOI: 10.1038/nrd3229.

Nicholls, Anthony (2008). 'What Do We Know and When Do We Know It?' In: *J. Comput.-Aided Mol. Des.* 22.3-4, pp. 239–255. DOI: 10.1007/s10822-008-9170-2.

Nicolaou, Christos A. and Nathan Brown (2013). 'Multi-Objective Optimization Methods in Drug Design'. In: *Drug Discovery Today: Technologies* 10.3, e427 –e435. DOI: 10.1016/j.ddtec.2013.02.001.

Nishibata, Yoshihiko and Akiko Itai (1991). 'Automatic Creation of Drug Candidate Structures Based on Receptor Structure. Starting Point for Artificial Lead Generation.' In: *Tetrahedron* 47.43, pp. 8985 –8990. DOI: 10.1016/S0040-4020(01)86503-0.

O'Boyle, Noel and Andrew Dalke (2018). 'DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures'. In: DOI: 10.26434/chemrxiv.7097960.v1.

Odolczyk, Norbert, Janine Fritsch, Caroline Norez, Nathalie Servel, Melanie Faria Da Cunha, Sara Bitam, Anna Kupniewska, Ludovic Wiszniewski, Julien Colas, Krzysztof Tarnowski, Danielle Tondelier, Ariel Roldan, Emilie L. Saussereau, Patricia Melin-Heschel, Grzegorz Wieczorek, Gergely L. Lukacs, Michal Dadlez, Grazyna Faure, Harald Herrmann, Mario Ollero, Frédéric Becq, Piotr Zielenkiewicz and Aleksander Edelman (2013). 'Discovery of Novel Potent ΔF508-CFTR Correctors that Target the Nucleotide Binding Domain'. In: *EMBO Mol. Med.* 5.10, pp. 1484–1501. DOI: 10.1002/emmm.201302699.

Olivecrona, Marcus, Thomas Blaschke, Ola Engkvist and Hongming Chen (2017). 'Molecular De-Novo Design Through Deep Reinforcement Learning'. In: *J. Cheminf.* 9.1, p. 48. DOI: 10.1186/s13321-017-0235-x.

Ou-Yang, Si-sheng, Jun-yan Lu, Xiang-qian Kong, Zhong-jie Liang, Cheng Luo and Hualiang Jiang (2012). 'Computational Drug Discovery'. In: *Acta Pharmacol. Sin.* 33.9, pp. 1131–1140. DOI: 10.1038/aps.2012.109.

Pan, S. J. and Q. Yang (2010). 'A Survey on Transfer Learning'. In: *IEEE Trans. Knowl. Data. Eng.* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

Panchagnula, Ramesh and Narisetty Sunil Thomas (2000). 'Biopharmaceutics and Pharmacokinetics in Drug Research'. In: *Int. J. Pharm.* 201.2, pp. 131 –150. DOI: 10.1016/S0378-5173(00)00344-6.

Pantoom, Supansa, Ingrid R. Vetter, Heino Prinz and Wipa Suginta (2011). 'Potent Family-18 Chitinase Inhibitors: X-ray Structures, Affinities, and Binding Mechanisms'. In: *J. Biol. Chem.* 286.27, pp. 24312–24323. DOI: 10.1074/jbc.M110.183376.

Paolini, Gaia V., Richard H. B. Shapland, Willem P. van Hoorn, Jonathan S. Mason and Andrew L. Hopkins (2006). 'Global Mapping of Pharmacological Space'. In: *Nat. Biotech.* 24.7, pp. 805–815. DOI: 10.1038/nbt1228.

Papadatos, George and Nathan Brown (2013). 'In silico Applications of Bioisosterism in Contemporary Medicinal Chemistry Practice'. In: *WIREs Comput. Mol. Sci.* 3.4, pp. 339–354. DOI: 10.1002/wcms.1148.

Papadatos, George, Anna Gaulton, Anne Hersey and John P. Overington (2015). 'Activity, Assay and Target Data Curation and Quality in the ChEMBL Database'. In: *J. Comput.-Aided Mol. Des.* 29.9, pp. 885–896. DOI: 10.1007/s10822-015-9860-5.

Pason, Lukas P. and Christoph A. Sotriffer (2016). 'Empirical Scoring Functions for Affinity Prediction of Protein-ligand Complexes'. In: *Molecular Informatics* 35.11-12, pp. 541–548. DOI: 10.1002/minf.201600048.

Paszke, Adam, 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 (2019). 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*, pp. 8026–8037.

Paul, Steven M., Daniel S. Mytelka, Christopher T. Dunwiddie, Charles C. Persinger, Bernard H. Munos, Stacy R. Lindborg and Aaron L. Schacht (2010). 'How to Improve R&D Productivity: The Pharmaceutical Industry's Grand Challenge'. In: *Nat. Rev. Drug Discovery* 9, pp. 203–214.

Polishchuk, P. G., T. I. Madzhidov and A. Varnek (2013). 'Estimation of the Size of Drug-Like Chemical Space Based on GDB-17 Data'. In: *J. Comput.-Aided Mol. Des.* 27.8, pp. 675–679. DOI: 10.1007/s10822-013-9672-4.

Polykovskiy, Daniil, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik and Alex Zhavoronkov (2020). 'Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models'. In: *Front. Pharmacol.* 11, p. 1931. DOI: 10.3389/fphar.2020.565644.

Popova, Mariya, Olexandr Isayev and Alexander Tropsha (2018). 'Deep Reinforcement Learning for De Novo Drug Design'. In: *Sci. Adv.* 4.7, eaap7885. DOI: 10.1126/sciadv.aap7885.

Putta, Santosh, Gregory A. Landrum and Julie E. Penzotti (2005). 'Conformation Mining: An Algorithm for Finding Biologically Relevant Conformations'. In: *J. Med. Chem* 48.9, pp. 3313–3318. DOI: 10.1021/jm0490661.

Ragoza, Matthew, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri and David Ryan Koes (2017). 'Protein-Ligand Scoring with Convolutional Neural Networks'. In: *J. Chem. Inf. Model.* 57.4, pp. 942–957. DOI: 10.1021/acs.jcim.6b00740.

Ragoza, Matthew, Tomohide Masuda and David Ryan Koes (2020). 'Learning a Continuous Representation of 3D Molecular Structures with Deep Generative Models'. In: *NeurIPS Workshop on Machine Learning for Structural Biology*.

Ramesha, Chakkodabylu S (2000). 'How Many Leads From HTS? – Comment'. In: *Drug Discovery Today* 5.2, pp. 43 –44. DOI: 10.1016/S1359-6446(99)01444-0.

Ramsundar, Bharath, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding and Vijay Pande (2015). 'Massively Multitask Networks for Drug Discovery'. In: *arXiv preprint:1502.02072*. DOI: https://arxiv.org/abs/1502.02072.

Réau, Manon, Florent Langenfeld, Jean-François Zagury, Nathalie Lagarde and Matthieu Montes (2018). 'Decoys Selection in Benchmarking Datasets: Overview and Perspectives'. In: *Front. Pharmacol.* 9, p. 11. DOI: 10.3389/fphar.2018.00011.

Riniker, Sereina and Gregory A. Landrum (2013). 'Open-Source Platform to Benchmark Fingerprints for Ligand-Based Virtual Screening'. In: *J. Cheminf.* 5.5, p. 26. DOI: 10.1186/1758-2946-5-26.

Rogers, David and Mathew Hahn (2010). 'Extended-Connectivity Fingerprints'. In: *J. Chem. Inf. Model.* 50.5, pp. 742–754. DOI: 10.1021/ci100050t.

Rohrer, Sebastian G. and Knut Baumann (2009). 'Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data'. In: *J. Chem. Inf. Model.* 49.2, pp. 169–184. DOI: 10.1021/ci8002649.

Rosa, Arianna Carolina, Mitsunobu Mio, Ioanna Andreadou and Vadim V. Sumbayev (2020). 'Editorial: The Challenge of New Therapeutic Approaches for Unmet Therapeutic Needs'. In: *Front. Pharmacol.* 11, p. 1341. DOI: 10.3389/fphar.2020.01341.

Rosenblatt, Frank (1958). 'The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.' In: *Psychol. Rev.* 65.6, p. 386.

Ross, Gregory A., Garrett M. Morris and Philip C. Biggin (2013). 'One Size Does Not Fit All: The Limits of Structure-Based Models in Drug Discovery'. In: *J. Chem. Theory Comput.* 9.9, pp. 4266–4274. DOI: 10.1021/ct4004228.

Rotstein, Sergio H. and Mark A. Murcko (1993). 'GenStar: A Method for De Novo Drug Design'. In: *J. Comput.-Aided Mol. Des.* 7.1, pp. 23–43. DOI: 10.1007/BF00141573.

Roughley, Stephen D. and Allan M. Jordan (2011). 'The Medicinal Chemist's Toolbox: An Analysis of Reactions Used in the Pursuit of Drug Candidates'. In: *J. Med. Chem.* 54.10, pp. 3451–3479. DOI: 10.1021/jm200187y.

Ruder, Sebastian (2016). 'An overview of gradient descent optimization algorithms'. In: *CoRR* abs/1609.04747.

Rumelhart, David E., Geoffrey E. Hinton and Ronald J. Williams (1986). 'Learning Representations by Back-Propagating Errors'. In: *Nature* 323.6088, pp. 533–536. DOI: 10.1038/323533a0.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei (2015). 'ImageNet Large Scale Visual Recognition Challenge'. In: *Int. J. Comput. Vis.* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Salt, David W., Nihat Yildiz, David J. Livingstone and Chris J. Tinsley (1992). 'The Use of Artificial Neural Networks in QSAR'. In: *Pestic. Sci.* 36.2, pp. 161–170. DOI: 10.1002/ps.2780360212.

Sanchez-Gonzalez, Alvaro, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell and Peter Battaglia (2018). 'Graph Networks as

Learnable Physics Engines for Inference and Control'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, pp. 4470–4479.

Sanchez-Lengeling, Benjamin, Carlos Outeiral, Gabriel L. Guimaraes and Alan Aspuru-Guzik (2017). 'Optimizing Distributions over Molecular Space. An Objective-Reinforced Generative Adversarial Network for Inverse-Design Chemistry (ORGANIC)'. In: ChemRxiv preprint:chemrxiv.5309668.v3. DOI: 10.26434/chemrxiv.5309668.v3.

Santurkar, Shibani, Dimitris Tsipras, Andrew Ilyas and Aleksander Madry (2018). 'How Does Batch Normalization Help Optimization?' In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett. Curran Associates, Inc., pp. 2483–2493.

Saubern, Simon, Rajarshi Guha and Jonathan B. Baell (2011). 'KNIME Workflow to Assess PAINS Filters in SMARTS Format. Comparison of RDKit and Indigo Cheminformatics Libraries'. In: *Mol. Inf.* 30.10, pp. 847–850. DOI: 10.1002/minf.201100076.

Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini (2009). 'The Graph Neural Network Model'. In: *IEEE Trans. Neural Netw.* 20.1, pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

Schaller, David, Dora Šribar, Theresa Noonan, Lihua Deng, Trung Ngoc Nguyen, Szymon Pach, David Machalz, Marcel Bermudez and Gerhard Wolber (2020). 'Next Generation 3D Pharmacophore Modeling'. In: *WIREs Comput. Mol. Sci* 10.4, e1468. DOI: 10.1002/wcms.1468.

Schindler, Christina E. M., Hannah Baumann, Andreas Blum, Dietrich Böse, Hans-Peter Buchstaller, Lars Burgdorf, Daniel Cappel, Eugene Chekler, Paul Czodrowski, Dieter Dorsch, Merveille K. I. Eguida, Bruce Follows, Thomas Fuchß, Ulrich Grädler, Jakub Gunera, Theresa Johnson, Catherine Jorand Lebrun, Srinivasa Karra, Markus Klein, Tim Knehans, Lisa Koetzner, Mireille Krier, Matthias Leiendecker, Birgitta Leuthner, Liwei Li, Igor Mochalkin, Djordje Musil, Constantin Neagu, Friedrich Rippmann, Kai Schiemann, Robert Schulz, Thomas Steinbrecher, Eva-Maria Tanzer, Andrea Unzue Lopez, Ariele Viacava Follis, Ansgar Wegener and Daniel Kuhn (2020). 'Large-Scale Assessment of Binding Free Energy Calculations in Active Drug Discovery Projects'. In: *J. Chem. Inf. Model.* 60.11, pp. 5457–5474. DOI: 10.1021/acs.jcim.0c00900.

Schlichtkrull, Michael, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov and Max Welling (2018). 'Modeling Relational Data with Graph Convolutional Networks'. In: *The Semantic Web*. Ed. by Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai and Mehwish Alam. Cham: Springer International Publishing, pp. 593–607.

Schneider, Gisbert and Hans-Joachim Böhm (2002). 'Virtual Screening and Fast Automated Docking Methods'. In: *Drug Discovery Today* 7.1, pp. 64 –70. DOI: 10.1016/S1359-6446(01)02091-8.

Schneider, Gisbert and David E. Clark (2019). 'Automated De Novo Drug Design: Are We Nearly There Yet?' In: *Angew. Chem., Int. Ed.* 58.32, pp. 10792–10803. DOI: 10.1002/anie.201814681.

Schütt, Kristof, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko and Klaus-Robert Müller (2017). 'SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions'. In: *Advances in Neural Information Processing Systems 30*. Vol. 30, pp. 991–1001.

Scior, Thomas, Andreas Bender, Gary Tresadern, Jose L. Medina-Franco, Karina Martínez-Mayorga, Thierry Langer, Karina Cuanalo-Contreras and Dimitris K. Agrafiotis (n.d.). 'Recognizing Pitfalls in Virtual Screening: A Critical Review'. In: *J. Chem. Inf. Model.* 4 (), pp. 867–881. DOI: 10.1021/ci200528d.

Scott, Duncan E., Anthony G. Coyne, Sean A. Hudson and Chris Abell (2012). 'Fragment-Based Approaches in Drug Discovery and Chemical Biology'. In: *Biochemistry* 51.25, pp. 4990–5003. DOI: 10.1021/bi3005126.

Segler, Marwin H. S., Thierry Kogej, Christian Tyrchan and Mark P. Waller (2018). 'Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks'. In: *ACS Cent. Sci.* 4.1, pp. 120–131. DOI: 10.1021/acscentsci.7b00512.

Senior, Andrew W., Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu and Demis Hassabis (2020). 'Improved Protein Structure Prediction Using Potentials from Deep Learning'. In: *Nature* 577.7792, pp. 706–710. DOI: 10.1038/s41586-019-1923-7.

Shekhar, Chandra (2008). 'In Silico Pharmacology: Computer-Aided Methods Could Transform Drug Development'. In: *Chem. Biol.* 15.5, pp. 413 –414. DOI: 10.1016/j.chembiol.2008.05.001.

Shen, Yelong, Xiaodong He, Jianfeng Gao, Li Deng and Gregoire Mesnil (2014). 'Learning Semantic Representations Using Convolutional Neural Networks for Web Search'. In.

Sheridan, Robert P. (2013). 'Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction.' In: *J. Chem. Inf. Model.* 53.4, pp. 783–790. DOI: 10.1021/ci400084k.

Shuker, Suzanne B., Philip J. Hajduk, Robert P. Meadows and Stephen W. Fesik (1996). 'Discovering High-Affinity Ligands for Proteins: SAR by NMR'. In: *Science* 274.5292, pp. 1531–1534. DOI: 10.1126/science.274.5292.1531.

Siedlecki, Pawel, Regine Garcia Boy, Tanja Musch, Bodo Brueckner, Sandor Suhai, Frank Lyko and Piotr Zielenkiewicz (2006). 'Discovery of Two Novel, Small-Molecule Inhibitors of DNA Methylation'. In: *J. Med. Chem.* 49.2, pp. 678–683. DOI: 10.1021/jm050844z.

Sieg, Jochen, Florian Flachsenberg and Matthias Rarey (2019). 'In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening'. In: *J. Chem. Inf. Model.* 59.3, pp. 947–961. DOI: 10.1021/acs.jcim.8b00712.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel and Demis Hassabis (2016). 'Mastering the Game of Go with Deep Neural Networks and Tree Search'. In: *Nature* 529.7587, pp. 484–489. DOI: 10.1038/nature16961.

Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel,

Timothy Lillicrap, Karen Simonyan and Demis Hassabis (2018). 'A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go Through Self-Play'. In: *Science* 362.6419, pp. 1140–1144. DOI: 10.1126/science.aar6404.

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel and Demis Hassabis (2017). 'Mastering the Game of Go Without Human Knowledge'. In: *Nature* 550.7676, pp. 354–359. DOI: 10.1038/nature24270.

Sippl, Manfred J. (1990). 'Calculation of Conformational Ensembles from Potentials of Mean Force: An Approach to the Knowledge-Based Prediction of Local Structures in Globular Proteins'. In: *J. Mol. Biol.* 213.4, pp. 859 –883. DOI: 10.1016/S0022-2836(05)80269-4.

Skalic, Miha, José Jiménez, Davide Sabbadin and Gianni De Fabritiis (2019a). 'Shape-Based Generative Modeling for De Novo Drug Design'. In: *J. Chem. Inf. Model.* 59.3, pp. 1205–1214. DOI: 10.1021/acs.jcim.8b00706.

Skalic, Miha, Davide Sabbadin, Boris Sattarov, Simone Sciabola and Gianni De Fabritiis (2019b). 'From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design'. In: *Mol. Pharm.* 16.10, pp. 4282–4291. DOI: 10.1021/acs.molpharmaceut.9b00634.

Sliwoski, Gregory, Sandeepkumar Kothiwale, Jens Meiler and Edward W. Lowe (2014). 'Computational Methods in Drug Discovery'. In: *Pharmacol. Rev.* 66.1. Ed. by Eric L. Barker, pp. 334–395. DOI: 10.1124/pr.112.007336.

Souvorov, Alexandre, David Landsman, David J. Lipman, Deanna M. Church, Dennis A. Benson, Donna R. Maglott, Edwin Sequeira, Eugene Yaschenko, Gregory D. Schuler, Grigory Starchenko, James Ostell, Karl Sirotkin, Kathi Canese, Kim D. Pruitt, Lewis Y. Geer, Lukas Wagner, Martin Shumway, Michael DiCuccio, Michael Feolo, Oleg Khovayko, Roman L. Tatusov, Ron Edgar, Scott Federhen, Stephen H. Bryant, Steven T. Sherry, Tanya Barrett, Tatiana A. Tatusova, Thomas L. Madden, Vadim Miller, Vyacheslav Chetvernin, Wolfgang Helmberg, Yuri Kapustin and David L. Wheeler (2007). 'Database Resources of the National Center for Biotechnology Information'. In: *Nucleic Acids Res.* 36.suppl_1, pp. D13–D21. DOI: 10.1093/nar/gkm1000.

Stahl, Martin, Wolfgang Guba and Manfred Kansy (2006). 'Integrating Molecular Design Resources Within Modern Drug Discovery Research: The Roche Experience'. In: *Drug Discovery Today* 11.7, pp. 326 –333. DOI: 10.1016/j.drudis.2006.02.008.

Ståhl, Niclas, Göran Falkman, Alexander Karlsson, Gunnar Mathiason and Jonas Boström (2019). 'Deep Reinforcement Learning for Multiparameter Optimization in De Novo Drug Design'. In: *J. Chem. Inf. Model.* 59.7, pp. 3166–3176. DOI: 10.1021/acs.jcim.9b00325.

Stecula, Adrian, Muhammad S. Hussain and Ronald E. Viola (2020). 'Discovery of Novel Inhibitors of a Critical Brain Enzyme Using a Homology Model and a Deep Convolutional Neural Network'. In: *J. Med. Chem.* 63.16, pp. 8867–8875. DOI: 10.1021/acs.jmedchem.0c00473.

Stein, Reed M., Hye Jin Kang, John D. McCorvy, Grant C. Glatfelter, Anthony J. Jones, Tao Che, Samuel Slocum, Xi-Ping Huang, Olena Savych, Yurii S. Moroz, Benjamin Stauch, Linda C. Johansson, Vadim Cherezov, Terry Kenakin, John J. Irwin, Brian K. Shoichet, Bryan L. Roth and Margarita L. Dubocovich

(2020). 'Virtual discovery of Melatonin Receptor Ligands to Modulate Circadian Rhythms'. In: *Nature* 579.7800, pp. 609–614. DOI: 10.1038/s41586-020-2027-0.

Sterling, Teague and John J. Irwin (2015). 'ZINC 15 – Ligand Discovery for Everyone'. In: *J. Chem. Inf. Model.* 55.11, pp. 2324–2337. DOI: 10.1021/acs.jcim.5b00559.

Stierand, Katrin and Matthias Rarey (2010). 'Drawing the PDB: Protein-Ligand Complexes in Two Dimensions'. In: *ACS Med. Chem. Lett.* 1.9, pp. 540–545. DOI: 10.1021/ml100164p.

Stumpfe, Dagmar and Jürgen Bajorath (2012). 'Exploring Activity Cliffs in Medicinal Chemistry'. In: *J. Med. Chem.* 55.7, pp. 2932–2942. DOI: 10.1021/jm201706b.

Su, Minyi, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li and Renxiao Wang (2019). 'Comparative Assessment of Scoring Functions: The CASF-2016 Update'. In: *J. Chem. Inf. Model.* 59.2, pp. 895–913. DOI: 10.1021/acs.jcim.8b00545.

Sumita, Masato, Xiufeng Yang, Shinsuke Ishihara, Ryo Tamura and Koji Tsuda (2018). 'Hunting for Organic Molecules with Artificial Intelligence: Molecules Optimized for Desired Excitation Energies'. In: *ACS Cent. Sci.* 4.9, pp. 1126–1133. DOI: 10.1021/acscentsci.8b00213.

Sun, Chen, Abhinav Shrivastava, Saurabh Singh and Abhinav Gupta (2017). 'Revisiting Unreasonable Effectiveness of Data in Deep Learning Era'. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Sun, Huiyong, Peichen Pan, Sheng Tian, Lei Xu, Xiaotian Kong, Youyong Li, Dan Li and Tingjun Hou (2016). 'Constructing and Validating High-Performance MIEC-SVM Models in Virtual Screening for Kinases: A Better Way for Actives Discovery'. In: *Sci. Rep.* 6.1, p. 24817. DOI: 10.1038/srep24817.

Sunseri, Jocelyn, Jonathan E. King, Paul G. Francoeur and David Ryan Koes (2019). 'Convolutional Neural Network Scoring and Minimization in the D3R 2017 Community Challenge'. In: *J. Comput.-Aided Mol. Des.* 33.1, pp. 19–34. DOI: 10.1007/s10822-018-0133-y.

Sunseri, Jocelyn and David R. Koes (2020). 'libmolgrid: Graphics Processing Unit Accelerated Molecular Gridding for Deep Learning Applications'. In: *J. Chem. Inf. Model.* 60.3, pp. 1079–1084. DOI: 10.1021/acs.jcim.9b01145.

Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke and Alexander Alemi (2017). 'Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning'. In.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich (2014). 'Going Deeper with Convolutions'. In: *arXiv preprint:1409.4842*. DOI: 10.1109/CVPR.2015.7298594.

Tajbakhsh, Nima, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway and Jianming Liang (2017). 'Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?' In: *IEEE Trans. Med. Imaging* 35.5, pp. 1299–1312. DOI: 10.1109/TMI.2016.2535302.

Thompson, David C., R. Aldrin Denny, Ramaswamy Nilakantan, Christine Humblet, Diane Joseph-McCarthy and Eric Feyfant (2008). 'CONFIRM: Connecting Fragments Found in Receptor Molecules'. In: *J. Comput.-Aided Mol. Des.* 22.10, p. 761. DOI: 10.1007/s10822-008-9221-8.

Tiikkainen, Pekka, Pekka Tiikkainen, Patrick Markt, Patrick Markt, Gerhard Wolber, Gerhard Wolber, Johannes Kirchmair, Johannes Kirchmair, Simona Distinto,

Simona Distinto, Antti Poso, Antti Poso, Olli Kallioniemi and Olli Kallioniemi (2009). 'Critical Comparison of Virtual Screening Methods Against the MUV data set'. In: *J. Chem. Inf. Model.* 49.10, pp. 2168–2178. DOI: 10.1021/ci900249b.

Tran-Nguyen, Viet-Khoa, Célien Jacquemard and Didier Rognan (2020). 'LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening'. In: *J. Chem. Inf. Model.* DOI: 10.1021/acs.jcim.0c00155.

Trapero, Ana, Angela Pacitto, Vinayak Singh, Mohamad Sabbah, Anthony G. Coyne, Valerie Mizrahi, Tom L. Blundell, David B. Ascher and Chris Abell (2018). 'Fragment-Based Approach to Targeting Inosine-5-monophosphate Dehydrogenase (IMPDH) from Mycobacterium tuberculosis'. In: *J. Med. Chem* 61.7, pp. 2806–2822. DOI: 10.1021/acs.jmedchem.7b01622.

Trott, Oleg and Aj Olson (2010). 'AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization and Multithreading'. In: *J. Comput. Chem.* 31.2, pp. 455–461. DOI: 10.1002/jcc.21334.AutoDock.

Troup, Robert I, Charlene Fallan and Matthias G J Baud (2020). 'Current Strategies for the Design of PROTAC Linkers: A Critical Review'. In: *Explor. Target. Anti-Tumor Ther.* 1.5, pp. 273–312. DOI: 10.37349/etat.2020.00018.

U.S. Food and Drug Administration (2018a). *2017 New Drug Therapy Approvals*. https://www.fda.gov/downloads/AboutFDA/CentersOffices/OfficeofMedicalProductsandTobacco/CDER/ReportsBudgets/UCM591976.pdf. Accessed: 2018-10-23.

U.S. Food and Drug Administration (2018b). *The Drug Development Process - Step 3: Clinical Research*. https://www.fda.gov/ForPatients/Approvals/Drugs/ucm405622.htm. Accessed: 2018-10-30.

Vainio, Mikko J., Thierry Kogej, Florian Raubacher and Jens Sadowski (2013). 'Scaffold Hopping by Fragment Replacement'. In: *J. Chem. Inf. Model.* 53.7, pp. 1825–1835. DOI: 10.1021/ci4001019.

Venkatasubramanian, V., K. Chan and J.M. Caruthers (1994). 'Computer-Aided Molecular Design Using Genetic Algorithms'. In: *Comput. Chem. Eng.* 18.9. Comput. Chem. Eng., pp. 833 –844. DOI: 10.1016/0098-1354(93)E0023-3.

Verdonk, Marcel L., Valerio Berdini, Michael J. Hartshorn, Wijnand T. M. Mooij, Christopher W. Murray, Richard D. Taylor and Paul Watson (2004). 'Virtual Screening Using Protein-Ligand Docking: Avoiding Artificial Enrichment'. In: *J. Chem. Inf. Comput. Sci.* 44.3, pp. 793–806. DOI: 10.1021/ci034289q.

Verdonk, Marcel L., Jason C. Cole, Michael J. Hartshorn, Christopher W. Murray and Richard D. Taylor (2003). 'Improved Protein–Ligand Docking using GOLD'. In: *Proteins: Struct., Funct., Bioinf.* 52.4, pp. 609–623. DOI: 10.1002/prot.10465.

Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps and David Silver (2019).

'Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning'. In: *Nature* 575.7782, pp. 350–354. DOI: 10.1038/s41586-019-1724-z.

Vogel, Simon M., Matthias R. Bauer and Frank M. Boeckler (2011). 'DEKOIS: Demanding Evaluation Kits for Objective in Silico Screening — A Versatile Tool for Benchmarking Docking Programs and Scoring Functions'. In: *J. Chem. Inf. Model.* 51.10, pp. 2650–2665. DOI: 10.1021/ci2001549.

Wallach, Izhar, Michael Dzamba and Abraham Heifets (2015). 'AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery'. In: *arXiv preprint:1510.02855*.

Wallach, Izhar and Abraham Heifets (2018). 'Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization'. In: *J. Chem. Inf. Model.* 58.5, pp. 916–932. DOI: 10.1021/acs.jcim.7b00403.

Wallach, Izhar and Ryan Lilien (2011). 'Virtual Decoy Sets for Molecular Docking Benchmarks'. In: *J. Chem. Inf. Model.* 51.2, pp. 196–202. DOI: 10.1021/ci100374f.

Wang, Lingle, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K. Dahlgren, Jeremy Greenwood, Donna L. Romero, Craig Masse, Jennifer L. Knight, Thomas Steinbrecher, Thijs Beuming, Wolfgang Damm, Ed Harder, Woody Sherman, Mark Brewer, Ron Wester, Mark Murcko, Leah Frye, Ramy Farid, Teng Lin, David L. Mobley, William L. Jorgensen, Bruce J. Berne, Richard A. Friesner and Robert Abel (2015a). 'Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field'. In: *J. Am. Chem. Soc.* 137.7, pp. 2695–2703. DOI: 10.1021/ja512751q.

Wang, Renxiao, Yipin Lu and Shaomeng Wang (2003). 'Comparative Evaluation of 11 Scoring Functions for Molecular Docking'. In: *J. Med. Chem.* 46.12, pp. 2287–2303. DOI: 10.1021/jm0203783.

Wang, Yu, Yanzhi Guo, Qifan Kuang, Xuemei Pu, Yue Ji, Zhihang Zhang and Menglong Li (2015b). 'A Comparative Study of Family-Specific Protein-Ligand Complex Affinity Prediction Based on Random Forest Approach'. In: *J. Comput.-Aided Mol. Des.* 29.4, pp. 349–360. DOI: 10.1007/s10822-014-9827-y.

Warner, Daniel J., Matthew H. Bridgland-Taylor, Clare E. Sefton and David J. Wood (2012). 'Prospective Prediction of Antitarget Activity by Matched Molecular Pairs Analysis'. In: *Mol. Inf.* 31.5, pp. 365–368. DOI: 10.1002/minf.201200020.

Weininger, David (1988). 'SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules'. In: *J. Chem. Inf. Comput. Sci.* 28.1, pp. 31–36. DOI: 10.1021/ci00057a005.

Willett, Peter, John M. Barnard and Geoffrey M. Downs (1998). 'Chemical Similarity Searching'. In: *J. Chem. Inf. Model.* 38.6, pp. 983–996. DOI: 10.1021/ci9800211.

Williams, Ronald J. (1992). 'Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning'. In: *Mach. Learn.* 8.3, pp. 229–256. DOI: 10.1007/BF00992696.

Williams, Sarah CP (2013). 'Small Nanobody Drugs Win Big Backing From Pharma'. In: *Nat. Med.* 19.11, pp. 1355–1356. DOI: 10.1038/nm1113-1355.

Wójcikowski, Maciej, Pedro J. Ballester and Pawel Siedlecki (2017). 'Performance of Machine-Learning Scoring Functions in Structure-Based Virtual Screening'. In: *Sci. Rep.* 7.March, p. 46710. DOI: 10.1038/srep46710.

Wong, Chi Heem, Kien Wei Siah and Andrew W Lo (2018). 'Estimation of Clinical Trial Success Rates and Related Parameters'. In: *Biostatistics* 20.2, pp. 273–286. DOI: 10.1093/biostatistics/kxx069.

World Health Organization (2010). *Monitoring the Building Blocks of Health Systems: A Handbook of Indicators and Their Measurement Strategies*. World Health Organization.

Wu, Zhenqin, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing and Vijay Pande (2018). 'MoleculeNet: A Benchmark for Molecular Machine Learning'. In: *Chem. Sci.* 9, pp. 513–530. DOI: 10.1039/c7sc02664a.

Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang and Philip S. Yu (2019). 'A Comprehensive Survey on Graph Neural Networks'. In: *CoRR* abs/1901.00596.

Wyatt, Paul G., Andrew J. Woodhead, Valerio Berdini, John A. Boulstridge, Maria G. Carr, David M. Cross, Deborah J. Davis, Lindsay A. Devine, Theresa R. Early, Ruth E. Feltell, E. Jonathan Lewis, Rachel L. McMenamin, Eva F. Navarro, Michael A. O'Brien, Marc O'Reilly, Matthias Reule, Gordon Saxty, Lisa C. A. Seavers, Donna-Michelle Smith, Matt S. Squires, Gary Trewartha, Margaret T. Walker and Alison J.-A. Woolford (2008). 'Identification of N-(4-Piperidinyl)-4-(2,6-dichlorobenzoylamino)-1H-pyrazole-3-carboxamide (AT7519), a Novel Cyclin Dependent Kinase Inhibitor Using Fragment-Based X-Ray Crystallography and Structure Based Drug Design'. In: *J. Med. Chem.* 51.16, pp. 4986–4999. DOI: 10.1021/jm800382h.

Xia, Jie, Hongwei Jin, Zhenming Liu, Liangren Zhang and Xiang Simon Wang (2014). 'An Unbiased Method To Build Benchmarking Sets for Ligand-Based Virtual Screening and its Application To GPCRs'. In: *J. Chem. Inf. Model.* 54.5, pp. 1433–1450. DOI: 10.1021/ci500062f.

Xia, Jie, Terry-Elinor Reid, Song Wu, Liangren Zhang and Xiang Simon Wang (2018). 'Maximal Unbiased Benchmarking Data Sets for Human Chemokine Receptors and Comparative Analysis'. In: *J. Chem. Inf. Model.* 58.5, pp. 1104–1120. DOI: 10.1021/acs.jcim.8b00004.

Xia, Jie, Ermias Lemma Tilahun, Eyob Hailu Kebede, Terry-Elinor Reid, Liangren Zhang and Xiang Simon Wang (2015). 'Comparative Modeling and Benchmarking Data Sets for Human Histone Deacetylases and Sirtuin Families'. In: *J. Chem. Inf. Model.* 55.2, pp. 374–388. DOI: 10.1021/ci5005515.

Xia, Xiaolin, Jianxing Hu, Yanxing Wang, Liangren Zhang and Zhenming Liu (2020). 'Graph-Based Generative Models for De Novo Drug Design'. In: *Drug Discovery Today: Technol.* DOI: 10.1016/j.ddtec.2020.11.004.

Yang, Kevin, Wengong Jin, Kyle Swanson, Dr.Regina Barzilay and Tommi Jaakkola (2020a). 'Improving Molecular Design by Stochastic Iterative Target Augmentation'. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. Virtual: PMLR, pp. 10716–10726.

Yang, Yuyao, Shuangjia Zheng, Shimin Su, Chao Zhao, Jun Xu and Hongming Chen (2020b). 'SyntaLinker: automatic fragment linking with deep conditional transformer neural networks'. In: *Chem. Sci.* 11 (31), pp. 8312–8322. DOI: 10.1039/D0SC03126G.

You, Jiaxuan, Bowen Liu, Zhitao Ying, Vijay Pande and Jure Leskovec (2018a). 'Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation'. In:

Advances in Neural Information Processing Systems. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett. Vol. 31. Curran Associates, Inc., pp. 6410–6421.

You, Jiaxuan, Rex Ying, Xiang Ren, William Hamilton and Jure Leskovec (2018b). 'GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models'. In: *International Conference on Machine Learning (ICML)*. Proceedings of Machine Learning Research 80, pp. 5708–5717.

Yu, Dingjun, Hanli Wang, Peiqiu Chen and Zhihua Wei (2014). 'Mixed Pooling for Convolutional Neural Networks'. In: *Rough Sets and Knowledge Technology*. Ed. by Duoqian Miao, Witold Pedrycz, Dominik Ślęzak, Georg Peters, Qinghua Hu and Ruizhi Wang. Cham: Springer International Publishing, pp. 364–375.

Yu, Yuhai, Hongfei Lin, Jiana Meng, Xiaocong Wei, Hai Guo and Zhehuan Zhao (2017). 'Deep Transfer Learning for Modality Classification of Medical Images'. In: *Information* 8.3, p. 91. DOI: 10.3390/info8030091.

Yuriev, Elizabeth (2014). 'Challenges and Advances in Structure-Based Virtual Screening'. In: *Future Medicinal Chemistry* 6.1, pp. 5–7. DOI: 10.4155/fmc.13.186.

Zhang, Shuxing, Alexander Golbraikh and Alexander Tropsha (2006). 'Development of Quantitative Structure–Binding Affinity Relationship Models Based on Novel Geometrical Chemical Descriptors of the Protein–Ligand Interfaces'. In: *J. Med. Chem.* 49.9, pp. 2713–2724. DOI: 10.1021/jm050260x.

Zhang, Z., P. Cui and W. Zhu (2020). 'Deep Learning on Graphs: A Survey'. In: *IEEE Trans. Knowl. Data Eng.*, pp. 1–1. DOI: 10.1109/TKDE.2020.2981333.

Zhavoronkov, Alex, Yan A. Ivanenkov, Alex Aliper, Mark S. Veselov, Vladimir A. Aladinskiy, Anastasiya V. Aladinskaya, Victor A. Terentiev, Daniil A. Polykovskiy, Maksim D. Kuznetsov, Arip Asadulaev, Yury Volkov, Artem Zholus, Rim R. Shayakhmetov, Alexander Zhebrak, Lidiya I. Minaeva, Bogdan A. Zagribelnyy, Lennart H. Lee, Richard Soll, David Madge, Li Xing, Tao Guo and Alán Aspuru-Guzik (2019). 'Deep Learning Enables Rapid Identification of Potent DDR1 Kinase Inhibitors'. In: *Nat. Biotechnol.* 37.9, pp. 1038–1040. DOI: 10.1038/s41587-019-0224-x.

Zheng, Heping, Jing Hou, Matthew D Zimmerman, Alexander Wlodawer and Wladek Minor (2014). 'The Future of Crystallography in Drug Discovery'. In: *Expert Opin. Drug Discovery* 9.2, pp. 125–137. DOI: 10.1517/17460441.2014.872623.

Zheng, Xiliang, LinFeng Gan, Erkang Wang and Jin Wang (2013). 'Pocket-Based Drug Design: Exploring Pocket Space'. In: *AAPS J.* 15.1, pp. 228–241. DOI: 10.1208/s12248-012-9426-6.

Zhou, Hongyi and Jeffrey Skolnick (2011). 'GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction'. In: *Biophys. J.* 101.8, pp. 2043–2052. DOI: 10.1016/j.bpj.2011.09.012.

Zhou, Jie, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu and Maosong Sun (2018). 'Graph Neural Networks: A Review of Methods and Applications'. In: *CoRR* abs/1812.08434.

Zhou, Zhenpeng, Steven Kearnes, Li Li, Richard N. Zare and Patrick Riley (2019). 'Optimization of Molecules via Deep Reinforcement Learning'. In: *Sci. Rep.* 9.1, p. 10752. DOI: 10.1038/s41598-019-47148-x.

Zhu, Jun-Yan, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang and Eli Shechtman (2017). 'Toward Multimodal Image-to-Image

Translation'. In: *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 465–476.