THE UNIVERSITY OF CHICAGO

ARTIFICIAL INTELLIGENCE AND HIGH-PERFORMANCE COMPUTING FOR ACCELERATING STRUCTURE-BASED DRUG DISCOVERY

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCE
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
AUSTIN ROBERT CLYDE

CHICAGO, ILLINOIS
AUGUST 2022

For my great grandmother, Bridget.

"We cannot understand without wanting to understand, that is, without wanting to let something be said...Understanding does not occur when we try to intercept what someone wants to say to us by claiming we already know it."

Hans-Georg Gadamer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I want to express my deepest appreciation to my committee members Eric Jonas, Arvind Ramanathan, and my advisor Rick Stevens for their dedication, commitment, and advice. In particular, I am deeply indebted to Rick Stevens for his expansive and always in the future ideas, critical and encouraging words, and his endless commitment to involving students in the future of science, high-performance computing, and thought. Many ideas in this text came from discussions with Arvind Ramanathan, whose time and guidance was formative and inspirational. Throughout the process, Arvind served nearly every role from advisor and mentor to friend and collaborator.

I thank Monisha Ghosh for welcoming me to her research group during my undergraduate years and sharing with me the excitement of research (and Nita Yack for her gracious understanding of an undergrad's ineptitude for recording work hours). I thank Fangfang Xia for his mentorship, and in particular for a pivotal conversation about Alan Aspuru-Guzik's December 2018 paper. I thank Dr. Thomas Brettin for his mentorship and comments on my research, as well as managing all the various supports for my research. I thank Prof. Peter Coveney for his collaboration with me throughout my Ph.D., insights on expanding my work, in particular the application of fine-grain free energy methods, and integrating me into the European HPC world. I thank Prof. Shantenu Jha for his endless collaboration and guidance, especially with the development of high-performance workflow techniques.

I thank Prof. Sheila Jasanoff for her mentorship and guidance. We met during a short virtual mentorship meeting during the Science and Democracy Network, and ever since has been a guide in expanding my work beyond disciplinary boundaries. I thank Prof. Cristina Lafont for her mentorship and guidance on the Ph.D. process and academic job market. I thank Prof. Jeffery Bishop for welcoming me to the Saint Louis University philosophy community, and his advice as philosopher-scientist. I thank John Chodera for sharing his time, laboratory, and passion for statistical biophysics. I thank John Karanicholas for sharing his

laboratory with me for a week a few summers ago. In particular, our conversations motivated me to seek out a framework not based on screening but basic principles in medicinal chemistry. I thank Nadeem Vellore and Janssen Pharmaceuticals for their collaboration. I thank my peers for their support and insights into various projects: Ashka Shah, Max Zvyagin, Xiaotian Duan, and many others.

I thank my grandparents for nurturing my curiosity and for their care. They have been understanding and supportive throughout this somewhat foreign process in a way that I am deeply thankful for. While you might not see how the contents of this dissertation arose from your care, I hope you trust me enough to believe it. Finally, I thank Diane DiGravio, Karen Savella and Joann Hamm for fostering my argumentative drive and curiosity.

Finally, research from this dissertation drew from many projects across institutions. I aw knowledge their support below:

Research was supported by the DOE Office of Science through the National Virtual Biotechnology Laboratory, a consortium of DOE national laboratories focused on response to COVID-19, with funding provided by the Coronavirus CARES Act. This has been funded in part with Federal funding by the NCI-DOE Collaboration established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health, Cancer Moonshot Task Order No. 75N91019F00134 and under Frederick National Laboratory for Cancer Research Contract 75N91019D00024. This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health. This project has been funded in part with federal funds from the National Cancer Institute, National Institutes of Health, under Contract No. HHSN261200800001E. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

This research used resources at the Spallation Neutron Source and the High Flux Isotope Reactor, which are DOE Office of Science User Facilities operated by the Oak Ridge National Laboratory, managed by UT-Battelle LLC for DOE's Office of Science. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under contract DE-AC02-06CH11357. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. Additional data storage and computational support for this research project has been generously supported by the following resources: Petrel Data Service at the Argonne Leadership Computing Facility (ALCF), Frontera at the Texas Advanced Computing Center (TACC), Comet at the San Diego Supercomputing Center (SDSC). The work leveraged data and computing infrastructure produced in other projects, including: ExaLearn and the Exascale Computing Project (DOE Contract DE-AC02- 06CH11357); Parsl: parallel scripting library (NSF 1550588); funcX: distributed function as a service platform (NSF 2004894); Globus: data services for science (authentication, transfer, users, and groups Polymer Property Predictor Database (NIST 70NANB19H005 and NIST 70NANB14H012).

This work was supported or used resources from UK MRC Medical Bioinformatics project (grant no. MR/L016311/1), the UK Consortium on Mesoscale Engineering Sciences (UKCOMES grant no. EP/L00030X/1) and the European Commission for the EU H2020 CompBioMed2 Centre of Excellence (grant no. 823712). Access to SuperMUC-NG, at the Leibniz Supercomputing Centre in Garching, was made possible by a special COVID-19 allocation award from the Gauss Centre for Supercomputing in Germany. I am grateful for the contributions of OpenEye Scientific for academic access to their software tool kits, NVIDIA for access to resources and expertise, especially Tom Gibbs, Cerebras for access to resources and their expertise, especially Andy Hock, Nanome, and Acellera.

ABSTRACT

Traditional techniques for discovering novel drugs are too slow for 21st challenges, from precision oncology to emerging global pandemics. The COVID-19 Pandemic demonstrated the unequivocal need for rapidly deployable drug discovery capabilities as a matter of national biopreparedness and biosecurity. The challenge is, though, that drug discovery is an immense and complex interdisciplinary field drawing from cheminformatics, bioinformatics, biophysics, machine learning, and high-performance computing among others. To accelerate the screening process of new molecules, researchers are applying developments from artificial intelligence (AI) to the problem; however, the direct application of traditional AI methods overlooks the essential complexities of drug discovery, ranging from protein-conformation flexibility to unique statistical properties of virtual ligand screening. This dissertation presents a unique approach to AI for drug discovery based on tightly integrating insights from biochemistry and biophysics, driving a more accurate and more interpretable drug discovery system, all while leveraging the same accelerating advances from AI. These cross-cutting contributions from AI and HPC workflows illustrate orders of magnitude speedup for computational virtual drug screening, novel workflow designs for high-fidelity screening pipelines which are more accurate than traditional docking, new sampling strategies for exploring novel chemotypes, and complementary workflow analysis techniques which directly links actionable and interpretable goals (such as the design of drugs) with quantitative cost functions. These methodological developments are realized in a case study discovering and validating a novel SARS-CoV-2 3CL-Main Protease inhibitor with a K_i of 2.9 μM (95% CI 2.2, 4.0).

CHAPTER 1

INTRODUCTION

As of August 2022, the COVID-19 pandemic—caused by the SARS-CoV-2 virus—has killed over 6.4 million people worldwide and infected over 8% of the global population Organization [19]. Within just a few months during the spring and summer of 2022, Monkey Pox—a disease caused by the monkeypox virus from the same genus as variola virus, *Orthopoxvirus*—has already spread globally with over 30,000 cases identified. Even moving beyond the landscape of salient global pandemics, in 2019, it is estimated that between 4 and 6 million people died from disease caused by antimicrobial resistant (AMR) bacteria ([Murray et al., 2022]). Beyond pathogens, cancer is a leading killer with over 10 million deaths global. All of these diseases can be solved through small-molecule therapies, in theory.

Drug discovery is a huge business—and its only getting bigger with artificial intelligence and machine learning (AI/ML) (fig. 1.1). AI and drug discovery is expected to be a 12 billion dollar industry in 2026 [Factors, 2021]. AI in drug discovery is tackling drug design, drug development, predictive analytic, research risk assessment, and clinical tracking [Smalley, 2017]. Growth in ML research combined with a boom in high performance computing (HPC) has lead to new computing paradigms such as the utilization of graphics processing units (GPU) across research domains. As more nonstandard computing devices, architectures, and learning algorithms intersect drug discovery [Chen et al., 2020], it seems ripe to ask where is computational drug discovery heading?[1]

Yet, the scientific community has only explored a tiny fraction of small molecules. The enumeration and exploration chemicals is no new question: in 1875, Caley published a short note on his enumeration of alkanes utilizing a tree structure [Cayley, 1875]. Though Caley's

1. As a preliminary note, our aim is to develop a computational theory of the various computational practices employed in drug discovery. This text has its telescope pointed at a discipline outside of computer science. As with any understanding via a telescope from our land to a distant one, the understanding of medicinal chemistry developed here is best seen as anthropology. There is an inside story of medicinal chemistry—one which we claim to disclose—but we rather only can disclose via our telescope.

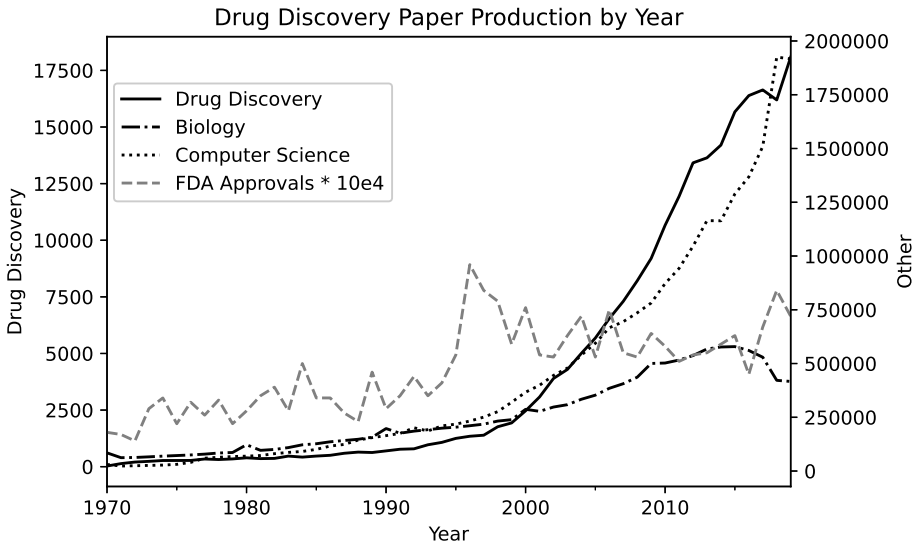


Figure 1.1: **Paper production and drug discovery.** (left axis) Drug discovery yearly paper production compared to (right axis) paper production in computer science and biology and FDA drug approvals. Drug discovery is rapidly growing at pace with computer science, rather than biology. Data was compiled using Microsoft academic graph and FDA data [Sinha et al., 2015, Kass-Hout et al., 2016].

enumeration ended up having a few errors, it is a very early account of treating chemical space as a structured mathematical object [Rains and Sloane, 1999]. The design space of small molecule—chemical space—is vast and estimated to consist of about 10^{60} molecules [Bohacek et al., 1996b]. Recent research has highlighted that moving beyond the standard chemotypes found in chemsimistry and biology textbooks and lab stocks is extremely fruitful for finding highly potent and selective inhibitors [Jia et al., 2019, Lyu et al., 2022, 2019]. This motivates an immediate need for efficient and automated exploration for synthesis and assay development for various applications, including drug discovery and materials design. Computational enumeration of chemical space is a long-studied problem since the early ages of computing [Cernak, 2018]. The current state of the art projects have enumerated around 2 billion drug-like compounds, and GDB has around 166 billion compounds of up to 17 atoms of C, N, O, S, and halogens [Patel et al., 2020, Ruddigkeit et al., 2012].

Given this unimaginably dense design space, the question is *how to identify drugs which*

can, with the help of other medical countermeasures and social controls, prevent disease, be personalised to reduce side-effects and increase efficacy, and be readily available cheaply, in real-time, and globally distributed? No pharmaceutical company may share all of these goals, but there are academic endeavours which have shown promise both in the scientific and social axis of this vision. During the COVID-19 pandemic, two academic group have identified small-molecular inhibitors such as the National Virtual Biotechnology Laboratory and the COVID Moonshot Project [Clyde et al., 2021d, Achdout et al., 2020c]. Both groups were able to identify lead compounds within 9 months of the beginning of the pandemic. Although the transition from drug lead to downstream toxicity, animal model, and human studies is a challenging and more difficult road (a open area for improvement—no doubt), this feat should invite further perspective and analysis on how this was possible?

These two groups, of which I was entrenched in both, pursued approaches to drug discovery unlike those found in tightly sealed corporate research and development offices. They focused on leveraging a large and interdisciplinary community of scientists from private, public, and academic laboratories [Clyde, 2022a]. They pursued computational techniques capable of leveraging the United States Department of Energy supercomputing infrastructure [Buchanan and Streiffer, 2020, Bhati et al., 2021] and, globally, the extra cycles of personal computers forming an decentralized "exascale" supercomputer [Zimmerman et al., 2021]. Lastly, they were both founded on the ideal of fully open and global science. Computational drug discovery has great potential for democratizing an industry facing many societal pressures.

1.1 Pharmaceutical pipeline

The drug development process is vast and spans disciplines. It spans initial pharmaceutical campaigns, testing, clinical trials, to FDA approval, and generally refered to as the the pharmaceutical pipeline or funnel. The first stage of the pharmaceutical pipeline is known

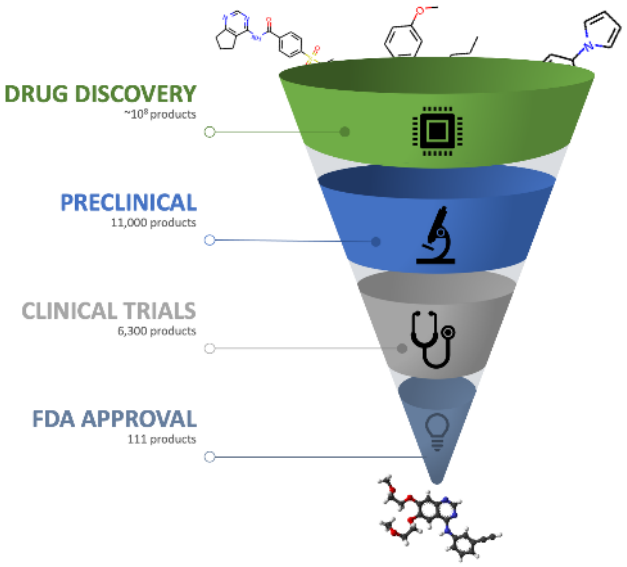


Figure 1.2: **The pharmaceutical pipeline.** Drug discovery is represented as funnel, starting with known compounds and filtering down through various stages. Numbers of products in each pipeline stage are from 2017 [Long, 2017].

as lead generation, lead discovery, or simply drug discovery. The goal of drug discovery is what it seems: to find promising compounds to solve a particular medicinal problem. At the end of the discovery stage of the pipeline, compounds are called *leads*. After the drug discovery process, sets of leads move to the next stage of preclinical testing. We will study in depth what it means for a drug to be considered a lead—for now we characterize such a moment as a compound being *promising*.

Preclinical testing involves *in-vivo* and *in-vitro* testing. For example, in cancer drug development, compounds in the drug discovery phase can be sent to laboratories for *in-vitro* cell line inhibition testing [Holbeck et al., 2017]. Drugs which showed promising inhibition *in-vitro* in cell line studies will be moved to preclinical toxicology studies and further inhibition studies. These may involve more complex *in-vivo* experiments such as organoid models, animal testing, and patient derived xenograft models (PDX) [Caldwell et al., 2001]. The final stages include clinical trials and FDA approval which is outside our scope; however, it is important to understand these stages are the bottleneck (both the slowest stages with the

fewest amount of compounds in the pipeline).

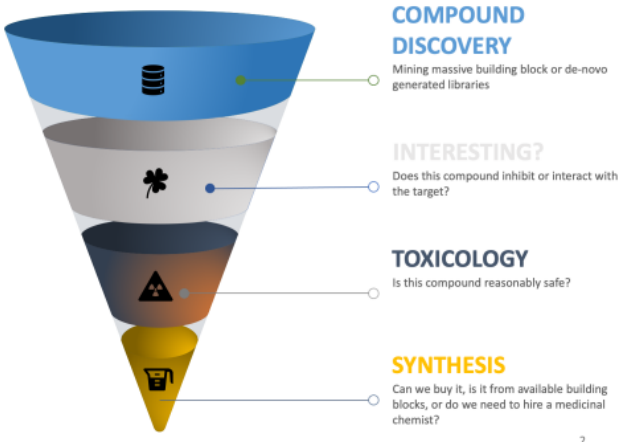


Figure 1.3: **Early drug discovery broken down into components.** The first step of the pharmaceutical pipeline, drug discovery, expanded into four specific filters outlining the initial steps to generate leads for preclinical or *in-vitro* testing.

1.2 Outline

In this dissertation, I will focus on the technical developments to accelerate and expand these drug discovery programs, particularly in AI and high-performance computing (HPC). I focus on drug discovery proper—the first stages of drug discovery–discovery and interest (fig. 1.3). While preclincal testing and later stages are essential, it is outside of my scope. In the following chapters, I will outline new approaches to scaling and applying these methods as part of a broader goal. I envision these systems as part of a basic biosecurity surveillance program capable of monitoring the environment, identifying threats, and queuing wet-lab automated testing so that when epidemiology officials declare a disease spreading uncontrollable, a treatment already is known and ready.

Artificial intelligence (AI) is a form of computational problem solving that a aims to (1) mimic some human notion of intelligence (whether it be the ability to naturally use language or identify objects in the world) and (2) scale these capabilities to process vast data outside the window reasonable by a single person. Recently, advances in deep learning

have revitalized AI's research program through breakthroughs in natural language processing [Brown et al., 2020], game-playing [Silver et al., 2016], protein-folding [Cramer, 2021], code-generation [Chen et al., 2021], and image-classification [He et al., 2016, Chai et al., 2021]. Although none of these models represent paradigmatic solutions to the problem[2], these models are, at a minimum, surprisingly adept and capable at convincing one that they are nearing solutions. In particular engineering domains, scientific modeling is increasingly merging with the technical and structural ideas presented in AI papers broadly encapsulating the idea of "AI for science" [Stevens et al., 2020]. If modeling offers little scientific insight from a theoretical perspective, what then can we make of AI for science (AI4S)?

One can raise the question that the science in the ideas percolating around AI4S might be a different kind of science. This dissertation is by no means an attempt to characterize the present status of computational scientific research nor fortune-tell its future; however, it is certainly a case study in AI4S. I will offer at moments a perspective on the broader ambition on how one can connect semantic meaning such as that sought in traditional science with the often overly-used quip that deep learning is a black box.[3] There are theoretical and practical assumptions carried with computational tools, and, similarly, the computational frame—a kind of bias required to force a scientific problem through the structure of a AI problem—does impact, affect, and alter the science [Anderson, 2008]. This co-production of the sciences and its theory alongside AI4S, its algorithmic assumptions, and the tools will be directly discussed at various moments (in particular in Chapter 9) [Jasanoff, 2004].

Given the interdisciplinary required for approaching computational drug discovery, the first two chapters of this dissertation outline background information for computer science

2. A paradigmatic solution would most likely come with some theory fitting the present ideas around what it means to solve a scientific challenge. None of these models present any theoretical breakthroughs within the terms of the problems they attempt to solve. Furthermore, many of these models require great set up, specification, and are limited in scope within those problem areas.

3. See forthcoming article "The Black Box Civic Epistemology: AI Senselessness and Democratic Participation."

minded researchers to understand the intricacies for drug discovery. Chapter 2 provides an interdisciplinary overview of the biology and biophysical background for drug discovery. While the chapter mainly aimed to review concepts such as different models of induced-fit protein-ligand binding and free-energy perturbation methods, it aims to concisely connect a computationally-minded reader into the various theoretical levers biophysics and biology have to offer.

Chapter 3 presents an overview of virtual ligand screening (VLS) as a computational challenge. Recent advances in computational screening techniques are outlined, including an overview of the different paradigms of drug discovery such as ligand and structure based drug discovery. Chapter 3 presents a contribution to ligand based drug design for precision oncology while the rest of this dissertation is devoted to structure-based drug design.

Chapter 4 outlines the different workflow designs for AI in HPC drug discovery workflows with a focus applying AI to structure based drug design. It begins with the simple idea of using surrogate models to replace slow and less-accurate code from traditional VLS workflows. This small building block is connected with different design ideas ranging from generative modeling to database searches.

Chapter 5 demonstrates precisely how chemical structures are can be used in deep neural network (DNN) based surrogate models. The challenge for training surrogate models is that the standard statistical assumptions stable neural network training relies on are not applicable for the highly-skewed distributions in drug discovery. This chapter showcases different techniques that can be applied to neural network training for improving the accuracy of speed up and illustrates various trade-offs of different chemical featurization techniques.

Chapter 6 showcases how increasingly accurate and more costly computational simulation techniques can be combined with surrogate models to achieve more accurate screening workflows without increasing the overall computational cost.

Chapter 7 removes the assumption that virtual screening problems maintain no structure

over the screening space by arguing that chemical space can be thought of as a multi-hypergraph partitioned by basic theoretical tools from cheminformatics such as Bemis-Murcko scaffold decomposition and electrostatics.

Chapter 8 presents new workflow evaluation metrics which couple business decisions such as spending for purchasing compounds directly with model performance and high-performance computing measures. Together, this presents a complete model of exactly how a model's performance is coupled to throughput and hit-identification as well as how computational characteristics of an HPC platform will scale those metrics.

Finally, chapter 9 summarizes various opportunities for future work such as utilizing advances in large-language models for improving the transferability of disparate data sets, improving the calibration and uncertain quantification, connecting active learning and automatic laboratories with drug discovery workflows, and finally generative drug design techniques.

CHAPTER 2

INTRODUCTION TO THE BIOPHYSICS AND BIOCHEMSITRY OF DRUG DISCOVERY

This chapter outlines drug discovery with a focus on the biological components. The first section will address the basics of cheminformatics and medicianl chemistry, including the mathematical set up. The second second outlines the role of proteins in disease. The third section explores the thermodyanmics of protein-ligand binding. The final section outlines the basics of molecular dynamics simulations and binding free energy estimation.

2.1 Medicinal and Bio- Chemistry

Small molecules act on proteins in cells. Through the introduction of a small molecule into a cell, the behavior of proteins can be modulated. In this way, biological response (BR) can be considered a function of chemical structure (C), $BR = f(C)$ [Hansch, 1976]. The magnitude and specificity of this action are determined by attributes of the compound such as its shape and flexibility, polarity, and physical properties (such as solubility) [Reymond and Awale, 2012]. Shape and flexibility refer to the range of 3D *conformations* that a compound can take on, where flexibility refers to the likelihood of multiple stable states or an overall instability. Polarity refers to the distribution of charges of chemical groups composing of a small molecule, leading to a molecule have an *electric dipole moment*—an uneven distribution of charges. The idea of *quantitative structure activity relationship* is to characterize how changes in these parameters lead to changes in biological response. For example, given two compounds C_1 and C_2, we might decompose the change activity around these attributes of the structure,

$$\Delta BR = f(C_2) - f(C_1) \approx f(\Delta(\text{sterics}) + \Delta(\text{polarity}) + \Delta(\text{hydrophoic}) + C_1) \qquad (2.1)$$

This is a core idea of medicinal chemistry, that small changes in chemical structure should produce discernible and small changes in response. A driving hypothesis of medicinal chemistry is that a specific small molecule ligand can be found for any binding sites [Schuffenhauer et al., 2006]. While other tools and disciplines have unique ways of ascertaining properties of compounds that may elicit a biological response, such as X-ray crystallography, experimental data from a series of perturbed compounds is an important paradigm we will consider in the design of computational drug discovery systems, specifically in chapters 7 and 8.

Colloquially in drug discovery, when we refer to small molecules we are referring to small organic compounds with qualitative drug-likeness properties[1]. The most commonly used rule is Lipinski's Rule of Five (not five rules, but all numbers are a multiple of five) [Lipinski et al., 1997]:

- No more than 5 hydrogen bond donors (total number of N-H and O-H bonds)
- No more than 10 hydrogen bond acceptors (N or O counts)
- A molecular mass less than 500 daltons
- An octanol-water partition-coefficient (logP) less than or equal to 5. LogP measures the relationship between lipophilicity and hydrophilicity and can be experimentally determined by mixing a compound in a a system consisting of water and octanol and calculating the ratio of the concentration of the compound in the two partitions of the mixture.

Although drug-likeness can be codified into filters, recent advances in large-scale chemical space exploration have demonstrated that it may be advantageous to loosen restrictions to prevent particular biases in our knowledge of chemistry drive our exploration of spaces [Jia et al., 2019, Lyu et al., 2019, Kaplan et al., 2022].

1. It should be noted drug-likeness is a somewhat contested term which can represent various different filters [Walters, 2012]. In general, the concept of drug-likeness is meant to exclude compounds which are merely too large to ever be permeable to cells, not soluble, or clearly toxic.

2.1.1 Cheminformatics and graph theory

In computer science, a graph G is a collection of objects or nodes V and the relationship between those objects, edges, $E \subseteq V \times V$. Two nodes $v_1, v_2 \in V(G)$ are connected if and only if $(v_1, v_2) \in E(G)$. Two graphs are said to be isomorphic if they are structurally identical. Graphs are used across chemical informatics from database matching to hashing. A molecular structure can be represented by identifying nodes of a graph with the atoms of the molecule and the edges with the bonds. Molecular graphs are therefore said to be colored and weighted since nodes have elemental types and charges and edges are associated with their bond type and aromaticity. Aromatic bonds occur in flat, cyclic, and conjugated rings systems that obey Huckel's rule, a delocalized conjugated π-system, most commonly an arrangement of alternating single and double bonds. Molecular graphs typically do not include hydrogens unless explicitly stated.

Molecules can be represented with a wide variety of identifiers ranging from generic names to International Union of Pure and Applied Chemistry (IUPAC) nomenclature to registry systems which assign formal names to compounds such as CAS Registry Number [Hall and Kier, 2001, Wigh et al., 2022]. Computationally, molecules can be represented by graphs, adjacency matrices (connectivity tables), or Simplified Molecular-Input Line-Entry Specification (SMILES), among others.

SMILES is arguably the most widely used computational representation in virtual screening, and has increased in usage with AI due to its similarity to sentences [Wigh et al., 2022]. Atoms in a SMILES string are represented by chemical symbols, where hydrogens are typically implicit. Water can be represented as just `O`. Bonds are represented with a "-" for a single bond, "=" for a double bond, "#" for a triple bond, "$" for a quadruple bond, and ":" for a aromatic bond. Single bonds are usually implicit. Branching can be indicated with parentheses, so for example isopropyl alcohol is written `CC(C)O`. The first two carbons are bonded, and the second carbon to the third, but the parenthesis indicates that the oxygen

is bond bonded to the third carbon but to the second. Ring systems are identified with numeral tags so `CCCC` is a chain of carbons but `C1CCCC1` is a six member ring. A second ring would increase the count.

Alternatives to SMILES strings have been proposed such as SELFIES [Krenn et al., 2019] since SMILES generally can be permuted. There canonicalization methods which can *generally* standardize a SMILES string into a common ordering [Jochum and Gasteiger, 1977]. SELFIES, for example, uses a formal grammar to derive words which represent semantically valid graphs. A challenge with other methods which introduce unique words is that the vocabulary of tokens becomes non-standard as it requires knowing in advance the set of compounds to be represented to build the vocabulary whiles SMILES has a fixed vocabulary from the elements and conjugations.

2.1.2 Descriptors and similarity

Computationally, chemical similarity can be computed using traditional graph similarity metrics. In generally, similarity in terms of chemistry is defined in terms of properties or descriptors. Molecular similarity or 2D similarity is defined in terms of graph similarity metrics. 3D similarity refers to particular 3D distance measures between graphs with the additional property that nodes have 3D coordinates.

Before deep learning, computational inference problems relied on classical machine learning models based on tabular, vectorized data. Molecular descriptors were created to vectorize molecules into a numerical and thereby commutable representation [Pastor et al., 2000, Yap, 2011, Todeschini and Consonni, 2008]. Various software packages exist open-source with various degrees of customizability. *MOrdred* offers a simple Python interface [Moriwaki et al., 2018]. Molecular descriptors can be used readily with nearly any machine learning technique or deep learning models.

Most descriptor packages consist of hundreds of routines for computing various molecular

features, such as molecular weight, number of acid or base groups, or charge. Each of these smaller computations is bundled together in a large vector. Thus, applied over a large molecular library, a familiar table emerges. Each row is representative of a sample, a molecule, and each column is a particular and explainable feature. This method is preferred for use with classical machine learning techniques as vector 1D features are often required for random forests or linear models.

It should be noted that molecular descriptors are distinct from of molecular fingerprints. Molecular fingerprints, another alternative for featurization, are bit vectors based on hashing different molecular neighborhoods together. Molecular fingerprints originated for use in databases as a surrogate for molecular similarity. Unlike fingerprints, molecular descriptors are explainable, where fingerprints individual bits are relatively opaque.

The most commonly used 2D similarity measure is fingerprint similarity. A fingerprint is a bit vector derived from a molecular graph. Two bit vectors can be compared with metrics such as tanimoto similarity (also known as the Jaccard index). Given two bit vectors X and Y, the tanimoto similarity is

$$T_s(X, Y) = \frac{\Sigma_i(X_i \wedge Y_i)}{\Sigma_i(X_i \vee Y_i)}. \tag{2.2}$$

In order to derive vectors from graphs, we will introduce the idea of message-passing on graphs. This presentation of message-passing for fingerprints is non-typical. It is based on a formulation common in graph neural networks, although the framework is general enough for arbitrary non-neural network functions.

To begin, we must expand our conception of elements of $\mathcal{M}$ beyond colored and weighted non-directed graphs to be an object which consists of this graph along with metadata for each edge and node. Consider a graph with possible node and edge data as $G = (V, E, \mathcal{A}, \mathcal{B})$. Here each graph consists of a set of nodes that have an element and charge type, V, a set of bonds which have a weight, E, a set of *features* for each node $\mathcal{A}$, and a set of *features* for each node $\mathcal{B}$. We think of $\mathcal{A}$ and $\mathcal{B}$ as functions from $V \to \mathbb{R}^m$ (so given a node, $\mathcal{A}$ returns

a vector of data). The data can be anything from additional details of atom type such as molecular weight to more complex details such as bond angles.

The problem can framed as we want to use some arbitrary function $f : \mathcal{A} \to \mathbb{R}^n$ and the structure of the graph G to reduce the graph down to a single vector. Given this framework, we will explore message passing as a technique to propagate information along graph edges (see [Zhou et al., 2018] for an overview of neural networks utilizing graphs as input). We consider a generalization here for homogeneous graphs. Heterogeneous graphs can be used to indicate different types of edges, that have different types of features (for double or single bonds, for example). That is, there is only one type of edge for propagating node features.

There are three operators in this framework. The idea is that for each each node, it sends its neighbors a message with the $\textsc{message}(\cdot)$ function, which a vector of nodes features to another vector (possibly of a different size), say $\mathbb{R}^n \to \mathbb{R}^m$. Each node then has a mailbox of messages. For each node, it applies a reduction (or aggregation function) to all messages in its mailbox, $\textsc{reduce}(\cdot)$. Examples of a reduce function might be a summation, product, or other more complex function which can take a variable number of vectors and reduce them to a single vector. Finally, there is an update step which replaces the state of data for each node $\mathcal{A}$ with a new state, $\textsc{update}(\cdot)$. If we use $\mathcal{N}_j$ to be the set of neighbors for node j, $h_i^{(0)} \in \mathbb{R}^n$ to be the state of $\mathcal{A}(i)$ for node i at time step 0, and $h_i^{(1)} \in \mathbb{R}^m$ for the state after one step of message passing, we write:

$$h_i^{(1)} = \textsc{update}(h_i^{(0)}, \textsc{reduce}(\{\textsc{message}(h_j^{(0)}) \mid j \in \mathcal{N}_j\})). \tag{2.3}$$

The last step after as many desired stages of message passing is completed is to reduce globally so that the graph is compressed into a vector. This is typically done with summation or averaging, written as $h^{(l+1)}$ with no subscript

$$h^{(l+1)} = \textsc{agg}_{v_i \in V} h_i^{(l)} \qquad \in \mathbb{R}^m \tag{2.4}$$

where m is the length of the vector after all the various message passing routines were run.

This generalizes to graph neural networks which we will briefly outline. The simplest setting is a graph convolutional layer (GCN) [Kipf and Welling, 2016]. For this layer, we set the message function to the identity function, the reduction function equal to summation, and the update function to a linear or dense neural network layer, which is to say $\text{UPDATE}(h_i, \widehat{h_j}) := \sigma(W\widehat{h_j})$ where W is a weight or parameter matrix adjusted during the training process with dimension m by m, $b \in \mathbb{R}^m$ is a bias term, σ is an activation function, and $\widehat{h_j} \in \mathbb{R}^m$ is the reduction of messages received by node i. We can write this as

$$h_i^{(l+1)} = \sigma(W^{(l)}[\sum_{v_j \in \mathcal{N}(v_i)} h_j^{(l)}]) \tag{2.5}$$

meaning a graph convolution layer adds all the node feature vectors from neighbors and applies a weight matrix. Notice in this model formulation, as written, unless self edges are explicit in the model (i.e., $(u, u) \in E$), it is possible the current features of node u are not used to update the features at the next layer $l+1$. This behavior is dependent on the software package being utilized and may or may not be desired. In this setting, though, we can only ever get node-specific features, not global (i.e., we can only ever develop per atom features as h is sub-scripted with nodes via i). Thus, just as in 2-dimensional convolution networks, a global pooling operation is required such as sum or average. Via global node pooling, we can create a feature vector or intermediate representation of the entire graph, rather than just a set of node-specific vectors through message passing alone. This is analogous to combining the bit vectors from the circular or Morgan fingerprints [Duvenaud et al., 2015]. These intermediate graph representations, or fingerprints, can then be used as features in any dense neural network to predict the scores. In the case of topological fingerprints, different kernels or functions are used during messaging process, or messaging may be modified to collect nodes with a particular graph distance (number of edges between), but the general

procedure is the same.

2.1.3 Chemical space

Since the early ages of artificial intelligence and computing, dating back to 1960s, chemists have imagined that computers would be able to make great progress on the basis of medicinal chemistry's hypothesis [Lederberg et al., 1969]. Given the scientific paradigm of small-molecule modulation of cellular activity and the belief that there exists a small molecule which can be selective to every target, drug discovery is well posed as a computational screening problem (discussed in the following chapter in detail). The basis of screening problem requires a well-defined design space. For us, this space is called *chemical space*—the space or set of all molecules.

Even with the restrictions of a molecular weight less than 500, no more than four rings, no more than 30 atoms, and only using hydrogen, carbon, oxygen, and sulfur, there are over 10^{63} compounds [Reymond, 2015]. This number is a vast underestimate of a reasonable design space in organic chemistry at large [Kirkpatrick and Ellis, 2004]. With the immense and innumerable number of possible chemicals, computational chemists refers to the density and vastness of chemical space. By referring to the range of potential chemicals as a space, we invoke the mathematical concept of a set with some added structure. We will discuss this added structure in chapter 7. For now, we denote chemical space as a mere set of graphs where the elements are all possible chemicals

$$\mathcal{M} = \{G = (V, E, \mathcal{A}, \mathcal{B}) \quad \text{s.t.} \quad G \text{ is a valid chemical structure}\} \tag{2.6}$$

Given the immense size of this space, it is effectively impossible to enumerate and store such a list. In theory the set is computable and enumerable since there exists a finite-time algorithm which can determine if a graph G is a possible small-molecule or not; however, the

lifespan of a human and limit on physical memory means that it is practically not possible to enumerate. This does not mean efforts have been made to enumerate large chemical libraries computationally. For example, the GDB-17 enumerates 166 billion possible compounds using C, N, O, S, up to 17 atoms. Pfizer has enumerated over 10 trillion compounds [Hu et al., 2011].

One further limitation of theoretical enumeration of compounds as graphs in $\mathcal{M}$ is that not all compounds have known pathways for synthesis. This limitation has been a major critique of the field of generative drug design—using AI models to generate novel possible compounds since they are in some sense "useless" if there is no method to create it [Farrant, 2020, Walters and Barzilay, 2021]. Chemical synthesis is the complex physical process of using reagents and reactants to take chemical building blocks to a desired compound. This is a complex and detailed process which can introduce impurities into mixtures and even create hazards for chemists. Furthermore, even if a chemical is synthesizable, the process may not be amenable to large-scale batch chemistry which is necessary for ultimate commercialization and wide scale availability of compounds. Advances from artificial intelligence in this area have made progress on retro-synthesis—finding a possible pathway for synthesis given a particular end target compound [Segler et al., 2018, Struble et al., 2020].

Enumerated libraries of compounds which have known building blocks and reaction pathways do exist and are commonly used. One library, called the Synthetically Accessible Virtual Inventory (SAVI), has over 1.75 billion compounds enumerated with the reaction steps and required building blocks [Patel et al., 2020]. Chemical vendors such as Mcule or Enamine offer libraries with over 31 billion possible compounds, where they promise a 70% materialization of compounds in the library [Shivanyuk et al., 2007]. There is great benefit to working off of libraries that are synthesizable as any hits identified can immediately be purchased for experimental evaluation. We will hastily denote the set $\mathcal{M}_{\mathcal{S}} \subseteq \mathcal{M}$ as

$$\mathcal{M}_{\mathcal{S}} = \{G \in \mathcal{M} \quad \text{s.t.} \quad G \text{ has a known synthesis pathway}\}. \tag{2.7}$$

In theory $\mathcal{M}_{\mathcal{S}}$ may actually not be computable in the computability theoretic sense since there may exist non-finite synthesis pathways. Thus, given a compound it may be indeterminable if it is in $\mathcal{M}_{\mathcal{S}}$ in finite time. Regardless, in practice, given finite number of reaction types and finite steps, the set is enumerable [Patel et al., 2020].

2.2 Proteins and Disease

Proteins are polymers of amino acids, and they perform a variety of functions within cells. Proteins differ primarily through different sequences of amino acids. Proteins are amino acid polymers with a somewhat-fixed vocabulary of around 20 amino acids [Buonfiglio et al., 2015]. Proteins range in size, but an average range is around 200 amino acids. Proteins serve as channels to transport material between cells, as signals to communicate within and between cells, and provide structure to cells, among many other functions. Proteins are created through the process of transcription and translation. Transcription is the copying of a segment of DNA into RNA and then into mRNA. Translation is the process involving ribosomes which translate the mRNA into sequence of amino acids.

Protein structure is related to protein function [Orengo et al., 1999]. Protein's primary structure is considered the 2D sequence of amino acids. Secondary structure is the initial folding of segments of the sequence into common motifs, such as α-helices or β-sheets. Protein ternary structure is the stabilization of the overall structure through side-chain interactions of amino acids. Proteins can also form complexes with each other, of the same or different protein, sometimes called a quaternary structure. For example, type II deoxyribonuclease forms a dimer–joining up with a second copy of itself. Proteins can also have intrinsically disordered regions which form no know stable structure [Oldfield and Dunker, 2014]. Even the most inherently structured proteins though still exhibit flexibiltiy, and often flexibiltiy is related to function—such as the opening and closing a channel.

Proteins can cause disease through many different mechanisms. For example, some pro-

teins can misfold causing a cascade of misfolding among other proteins, such as with sickle cell disease [Valastyan and Lindquist, 2014]. Cancer medications typically target kinases and inhibit their activity, leading to the death of tumors [Zhang et al., 2009]. Proteins of bacteria may be targeted with antibiotics. Non-human proteins may also be targeted in human treatments of viral diseases, such as with Emtricitabine/tenofovir which inhibit the action of viral reverse transcriptases—proteins which integrate viral RNA into human DNA.

A great challenge in drug discovery target identification—finding a protein which, if drugged, eliminates the disease. There are many aspects of target selection beyond merely locating disease causing proteins such as selectivity of a protein, other functions a protein may have outside of disease, and the ability for virus, bacteria, or cancer to mutate maintaing the protein function but evading the small molecule therapy. The region of a protein which performs its main function is called the active site. Regions of a protein that may fit a small molecule for binding is called a binding site or binding pocket. Sometimes binding for a drug may not occur in the active site, but rather in a different binding site. The action of the drug occurs through inducing a structural change in the protein which eliminates the active site's action. While target identification is an immense challenge, it not studied deeply in this dissertation. Instead, we will be addressing situations where a protein of interest is settled, a 3D structure is known, and a binding site or active site of interest is identified.

2.3 Protein-Ligand Binding

Before we dive into the computational frameworks, it is important to have a working theory for what it means for a drug to have activity and characterize what observable there are to measure this activity. This section covers the basics of biochemistry and cell biology to understand the rest of this text.

Lock and Key Theory of Drug Activity

Drugs work through their mechanism of action (MOA). The MOA for a drug is sometimes unknown, for instance lithium, Li^+, is a common treatment for psychological disorders. It is speculated that lithium up-regulates serotonin in healthy individuals, but the mechanisms still remains unknown [Massot et al., 1999]. The most common MOA for drugs in inhibition of a protein or enzyme with a known pathway or function. Aspirin is one of the only non-steroidal anti-inflammatory drug (NSAID) which has a known MOA. Aspirin binds irreversibly to COX-1 and changes the enzymatic activity of COX-2 [Warner and Mitchell, 2002].

The basic accepted theory for drug activity is attributed to Emil Fischer in 1894. Emil Fischer explained the specific action of an enzyme (protein) on a substrate (drug) through analogy to a lock (enzyme) and a key (drug). Essentially, the analogy poses proteins as containing special locks, and drugs are keys. If you obtain the correct key for a certain lock, the drug will bind to the protein.[2] Once a drug binds or interacts with a protein, if the activity decreases then the drug is referred to as *antagonist*. Conversely, if activity increases then the drug is an *agonist*.

The lock and key model is rather simplistic, and has seen as many changes over time—just as traditional keys themselves have. Variations of the lock and key model include the *induced fit model* which supposes some changes in the active site occur in the presence of a compound [Jorgensen, 1991]. A common example is D3-dopamine receptor (fig. 2.1, a G coupled protein receptor (GPCR), which has a druggable binding site which is *cryptic* [Ferruz et al., 2018]. In analogy, a cryptic binding site (or pocket) is a lock which is hidden until a key approaches and then appears. Familiar to computer scientists, one can cast the

2. To avoid confusion, binding can be covalent (reversible or irreversible) or non-covalent. In this text, we deal only with non-covalent drugs—but one should be aware it is possible for drugs to bind covalent to proteins and this may affect calculations if one is unaware of the possibility. See [Kalgutkar and Dalvie, 2012] for more information on covalent drug binding and covalent drug discovery.

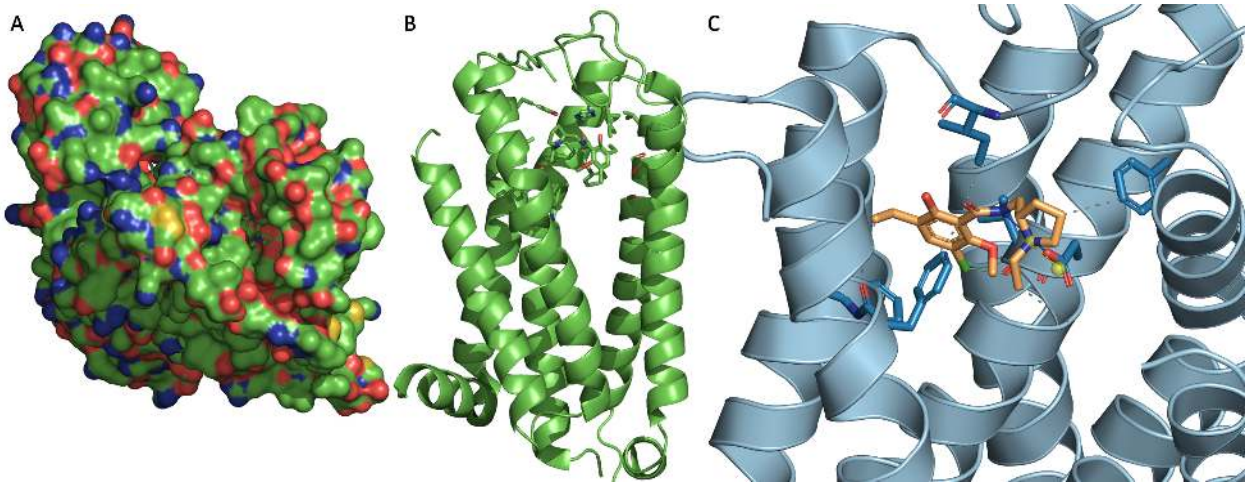


Figure 2.1: **Dopamine D3 receptor in complex with an antagonist drug situated in a cryptic pocket** (structure from Ferruz et al. [2018], PDB ID 3PBL). (A) Surface view of structure colored by electrostatics. Inside the pocket, far in, the ligand is situated. (B) A cartoon view of the protein oriented to illustrate this GPCR protein is a trans-membrane protein. [Ferruz et al., 2018]. (C) Structural interactions between the protein and ligand are illustrated by PLIP [Salentin et al., 2015]. The curly members are β-sheets which intersect the membrane, the orange structure is a ligand, and the blue residues sticking off the β-sheets are the functional components of the amino acids which exhibit interactions with the ligand (determined computationally by PLIP).

induced fit lock and key model into a Markov state model, where there are certain states and requirements to move between them. For example, in the case of Dopamine D3-receptor, [Ferruz et al., 2018] found five global states, with transitions from state 1 to state 2, a clique between states 2, 3 and 4, and transitions from state 4 (which is the bound ligand state) to state five (off-pathway).

2.3.1 Models of Binding

We assume that binding is non-covalent in this dissertation. This means that we assume no ligand is forming a bond to the protein, which is used in certain drug therapies such as Bruton's tyrosine kinase inhibitor ibrutinib [Weichert and Gmeiner, 2015, Boike et al., 2022]. Here we present a basic overview of theoretical concepts for protein-ligand binding theromodyamics and kinetics before discussing the computational aspects.

Consider a ligand L and a protein L in free (unbound) form. We consider the equilibrium between these two:

$$L + P \rightleftharpoons LP \quad \text{and} \quad K_b = \frac{[LP]}{[L][P]} = \frac{1}{K_d} \tag{2.8}$$

where K_b is known as the binding constant or binding affinity in M^{-1}, square brackets indicate concentrations in molarity, and K_d is the dissociation constant. This is a thermodynamic system as there is solute, solvent, and ions that exchange heat and interact. This is written as a equilibrium equation since ligands may bind and unbind constantly—hence, we investigate what is the stable equilibrium of the concentration of bound and unbound complexes. This can be quantified as Gibbs free energy (G), a thermodynamic potential [Ravindra and Winter, 2003], which relates to the equilibrium constants through the absolute protein-ligand binding free energy,

$$\Delta G^0 = \mu_{LP} - \mu_L - \mu_P = -k_b T \ln(C^0 K_b), \tag{2.9}$$

where $\mu_L P$, μ_L, and μ_P are the chemical potentials between the complex, ligand, and protein respectively, k_b is the Boltzmann constant, T is temperature, and C^0 is the concentration of reactants.

2.4 Computational Modeling of Protein-Ligand Binding

Computationally, the question is how to represent a whole system of biological macromolecules such as a protein with a small molecule ligand with a large number of water molecules and ions (fig. 2.1C).

2.4.1 Biophysical modeling with molecular dynamics

During the course of molecular dynamics (MD) simulations, we sample the state of a system as time progresses. The basic intuition stems from Newtown's equation $F = ma$, where we

can write for each atom

$$m_i \frac{\partial^2 r_i}{\partial t^2} = f_i \qquad f_i = -\frac{\partial U(r_i)}{\partial r_i} \tag{2.10}$$

where r_i is the position of atom, m_i is the mass, f_i is the force, and U is the potential energy. Next, to develop the simple algorithm called the Verlet Algorithm [Grubmüller et al., 1991], we introduce that particles also have atomic momenta p_i in terms of kinetic energy $K(p_i) = \sum \frac{|p_i|^2}{2} m_i$. Therefore the overall energy for the system is $E = \sum_i (K(p_i) + U(r_i))$. The Verlet algorithm may be written as

$$p_i(t + \frac{1}{2}\delta t) = p_i(t) + \frac{1}{2}\delta t f_i t(t) \tag{2.11}$$

$$r_i(t + \delta t) = r_i(t) + \delta t p_i(t + \frac{1}{2}\delta t) m_i^{-1} \tag{2.12}$$

$$p_i(t + \delta t) = p_i(t + \frac{1}{2}\delta t) + \frac{1}{2}\delta t f_i(t + \delta t). \tag{2.13}$$

How are forces computed? Consider the Born-Oppenheimer (BO) approximation to the energy potential $\mathcal{E}_{BO}$ [Ponder and Case, 2003]. For now we will leave this as a black box, since we use force-fields in molecular dynamics simulations. Force-fields are approximates to the quantum mechanical BO potential. Given bond length between two particles b and the energy minimum b_{eq},

$$\mathcal{E}_{BO}(b) = E_b + \frac{1}{2} K^{(\text{bond})} (b - b_{eq})^2 \tag{2.14}$$

where $K^{(\text{bond})}$ is the harmonic force and E_b is the offset. Using these approximations, we

can represent the potential energy with a force field as

$$U(r_1, \ldots, r_n) = \sum_{\text{bonds}} \frac{1}{2} K_i^{(\text{bond})} (b_i - b_{eq,i})^2 \tag{2.15}$$

$$+ \sum_{\text{angles}} \frac{1}{2} K_i^{(\text{angle})} (\theta_i - \theta_{eq,i})^2 \tag{2.16}$$

$$+ \sum_{\text{dihedrals}} \sum_{i} K_i^{(\text{dihedral})} \cos(m_i \phi_i) \tag{2.17}$$

$$\sum_{\text{non-bonded pairs} i,j} \left(\frac{q_i q_j}{\epsilon r_{ij}} + 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]\right). \tag{2.18}$$

This is the fundamental aspect of obtaining force-field parameters for molecular dynamics simulations. For more information on molecular dynamics simulations and the derivation of the energy functions see Allen et al. [2004].

2.4.2 Free energy of binding estimation

There are two primary methods for binding free estimation that will be used in later chapters: Molecular Mechanics Poisson-Boltzmann Surface Area (MM-PBSA) and Molecular and Mechanics generalized Born Surface Area (MM-GBSA) as well as free energy perturbations (FEP). The most basic approximation method is the linear response approximation (LRA). Broadly these are all methods of end-state methods. MM-PGBSA and MM-GBSA compute absolute binding free energy, that is they return a values for ΔG. FEP is a relative free energy of binding which provides an estimate of the difference between two difference ligands.

The simplest form of computing absolute ΔG is linear interaction energy model (LIE) [Åqvist et al., 1994]. LRA is estimated from the following approximation

$$\Delta G = \alpha[\langle E_{\text{ele}}^{L-S} \rangle_{\text{PL}} - \langle E_{\text{ele}}^{L-S} \rangle_{\text{L}}] + \beta[\langle E_{\text{vdw}}^{L-S} \rangle_{\text{PL}} - \langle E_{\text{vdw}}^{L-S} \rangle_{\text{PL}}] \tag{2.19}$$

where E_{vdw}^{L-S} is the van der Waals interactions between the ligand and solution and α and β are parameters. The brackets indicate an average over the simulation time. The way in which this average is computed from snapshots of a MD simulation can be a simple geometric average or include advanced sampling techniques such as the Multistate Bennett Acceptance Ratio (MBAR) methods [Shirts and Chodera, 2008].

MM-GBSA and MM-PBSA are extremely similar methods [Kollman et al., 2000]. The only difference between the two is the use of a particular solvation energy term based on Poisson-Boltzman or Born Surface Area. I will outline the procedure for MM-GBSA and the same procedure applies for MM-PBSA. In practice, three short simulations are carried out in solvent (water and ions) of the unbound protein, unbound ligand, and the complex. The binding free energy is estimated

$$\Delta G_{\text{bind}} = \langle G_{PL} \rangle - \langle G_P \rangle - \langle G_L \rangle \tag{2.20}$$

where

$$G = E_{\text{bond}} + E_{\text{elec}} + E_{\text{vdw}} + G_{\text{pol}} + G_{\text{np}} - TS \tag{2.21}$$

with G_{pol} and G_{np} are the polar and non-polar contribution, calculated from the Poisson-Boltzman equation or Born model. Sometimes only one simulation is used where the appropriate energies are obtained by removing atoms from the other components,

$$\Delta G_{\text{bind}} = \langle G_{PL} - G_P - G_L \rangle_{PL}. \tag{2.22}$$

Finally, there are alchemical relative free energy techniques such as FEP [Zwanzig, 1954]. These methods compute a relative free binding energy estimate $\Delta\Delta G$, based on analysis of simulation trajectories of a thermodynamic path from two states (fig. 2.2). The name alchemical comes from the fact that during the transition between these states, non-standard chemistries may be used. These approaches use a transition coupling parameter λ that moves

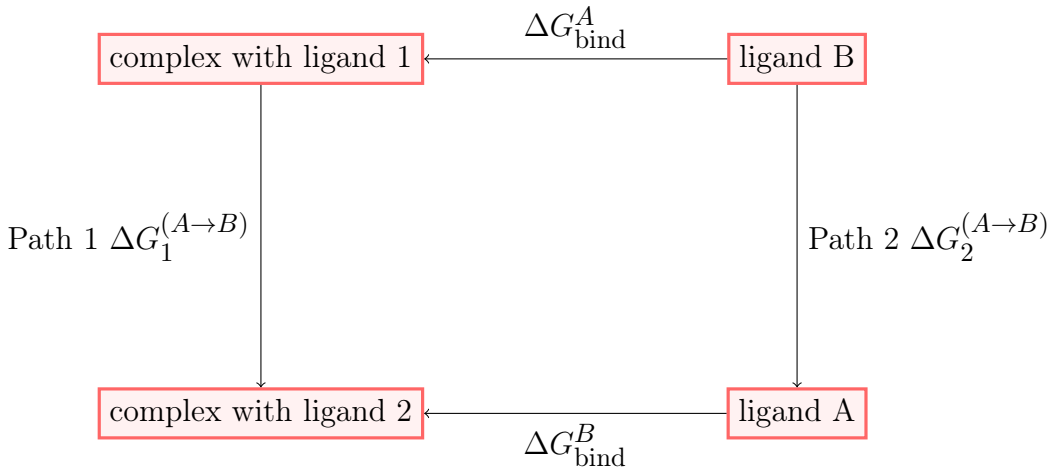


Figure 2.2: **Thermodynamic cycle for FEP.** The cycle is used to compute relative free energy between the binding of ligand A versus the binding of ligand B. Path 1 focuses on the transformation of the complex from ligand 1 to ligand 2, and path 2 focuses on the transformation of ligand 1 to ligand 2 in solution.

from the initial state, 0, to 1. The relative free energy is computed by

$$\Delta\Delta G = \Delta G_2^{(A\to B)} - \Delta G_1^{(A\to B)} = \Delta G^{B}_{\text{bind}} - \Delta G^{A}_{\text{bind}} \tag{2.23}$$

These methods are considerered to be more accurate than MM-GBSA or MM-PGBSA methods.

While the details of these methods are not essential for understanding the rest of this work, they do appear in great detail in chapter 6 and chapter 9.

2.5 Conclusion

The main theoretical contribution of medicinal chemistry to this work is the idea that small molecules are a design space which can be studied under perturbation. In some ways, the relativity smoothness of binding free energy with respect to small perturbations of a drug is a deep insight that will be explored later. We presented the notion of chemical space through the lens of graph theory, which lead to a unique presentation of tradditional

cheminformatics concepts. Finally, molecular dynamics simulations offer insight into the binding thermodyanmics of proteins and ligands in complex, offering a unqiue tool modulate the functions of proteins—at the root of many diseases.

CHAPTER 3

INTRODUCTION TO COMPUTATIONAL DRUG DISCOVERY

This chapter will focus directly on the topic of the remaining dissertation: virtual screening. The first section will cover some basics of computational screening broadly (not just for drugs). The second section will cover the history of virtual drug screening and the predominant ideologies (structure-based drug design and ligand-based drug design). This section will also feature some of my work in ligand-based drug design in particular for personalized cancer therapy. This will only be a minor example as this dissertation is focused on structure based drug design. The third section will focus on protein-ligand docking and the computational details of it (such as binding site identification, conformer generation, scoring function, available tools, and their computational performance characteristics).

3.1 Computational Screening

Computational screening problems include a wide variety of problems outside of drug discovery and is a general naive approach to inverse-design problems. The general idea of inverse design problems is that there exists some domain of physical objects (such as materials, small-molecules, manufacturing processes, computer architectures, etc.) that exhibit some set of properties (elasticity, solubility, efficiency, layout, etc.). The forward problem is considered property prediction, that is, using a computational model to ascertain quantitative properties from a computational representation of the physical object. We will discuss the forward problem of property prediction with molecules later in this chapter with techniques such as computational docking, which serve to approximate the property of binding free energy between a protein and small-molecule. The inverse problem asks given particular properties such a high likelihood of binding, what should the molecule be? The most basic set-up involves some function F which takes a object from the design space p and predicts

some observable property d_{obs}

$$d_{\text{obs}} = F(p). \tag{3.1}$$

Thus given some design goal such as maximization of a property, the inverse problem is

$$p^* = \arg\max F(p). \tag{3.2}$$

In this problem set up, it is clear that computational screening is the naive approach since computational screening performs a forward inference of properties over a large enumerated subset of the design space. Given then this list of property predictions over the space, the inverse problem is solved by merely selecting from this list the objects which satisfy the design goals. Furthermore, such an approach is naive as it does not consider any relationship between objects in the design space—an assumption which is most often not true. Objects in a design space are typically deeply related and these can be examined by looking at $\frac{\partial F}{\partial p}$, but of course often times it is not obvious how to differentiate a function with respect to physical objects. The computational limitations of screening this way relate to the size of the design space, n, and the efficiency of the approximation of F. Because we cannot directly control the size of the design space (at least until some more algebraic properties of the design space are articulated in chapter 7), we will focus in the next few chapters on accelerating the function approximation of F with machine learning and AI techniques. There are non-naive approaches to inverse design such as direct generation of the object using generative computational techniques. These will be outlined in chapters 7 and 9. For now, we consider computational screening.

3.2 Computational Drug Discovery

Drug discovery, as the first step of the process, is our focus (fig. 1.3). We start with simple compound discovery—in order to "discover" compounds, one must have a listing of some

compounds.[1] As with any in-the-world field of study, there are various theories of common practices situated with their own workflows, assumptions, and best practices. The main division in drug design is ligand based drug design versus structure based drug design. Ligand based drug design focuses on finding molecular motifs to design better drugs, while structure based drug design focuses on utilizing the drug target and 3D structure to guide the ligand design. The three common practices in drug discovery which utilize both design idealogies are *fragment based drug discovery* (FBDB), *high throughput virtual screening* (HTVS), and *scaffold based drug discovery*. In practice, these methods intersect in various ways, and often times other techniques all together are used such as natural product or patent and literature searching [Bleicher et al., 2003].

Fragment based drug discovery generates hits by identifying small chemical fragments that show activity.[2] Active fragments are identified by high throughput screening techniques such as solution NMR, computational modeling, or literature searches [Erlanson et al., 2016, Davies and Tickle, 2011]. New screening methods are being developed such as high throughput X-ray crystallography [Douangamath et al., 2020]. Fragment libraries are commonly available by chemical vendors and serve as the starting point.[3] Once identified, these active fragments are then combined together or grown into a full compound where activity is increased or *refined* through *iterative design*. The result is an active lead generated from an iterative design process based on a active fragment identified through some sort of computational or experimental search.

Structure based drug design focuses on utilizing structural data throughout the process

1. A listing is importantly completely enumerated. The most common example one encounters is a dataset (i.e. CSV file with SMILES strings or chemical formulas). A listing is not a function which could enumerate compounds if called (generator), or a combinatorial rule yet to be expanded. We will touch later on discovery in section 3 and means for chemical space.

2. A rule of 3 can be used to determine is a compound is a *fragment*: a molecular weight < 300, ClogpP < 3, and less than 3 hydrogen acceptors, donors, and rotatable bonds [Congreve et al., 2003].

3. One such fragment library for fragment screening is offered by chemical vendor Enamine, Ltd.: `https://enamine.net/compound-libraries/fragment-libraries`

[Anderson, 2003]. The essential structural data is the protein target. We deal only with protein drug targets. Protein structural information is determined experimentally, or can be estimated computationally using homology or even AI methods [Hillisch et al., 2004, Service, 2020]. Once the protein structure is obtained, the binding site can be analyzed to identify similar targets. If similar targets have known inhibitors, those inhibitors may serve as a starting point. When a protein structure is available, computational techniques such as protein-ligand docking can be utilized to score compounds based on their structural fit. Once some compounds are assessed, they can be tested *in-vitro* through biological assays [Vogel, 2002]. If the assays indicate binding, then crystallography can be used to determine the structure of the hit in complex with the protein. Based on the resulting complex structure, another iterative design loop can occur to increase the strength of the binding or alter the biochemical interactions observed.

Ultimately, most hit generation techniques in drug discovery contain at least one screening stage. Within the big design practices, searching is the common bottleneck. Screens, scientific searches, are a fundamental experimental design [Wass, 2010]; however, its popularity ought not justify its use. In chapter 3, we will develop a different perspective of computational drug discovery which moves away from screening as the main computational tool.

High throughput screening has a simple formal set up. Let $\mathcal{M}$ be chemical space. For now, an element in $\mathcal{M}$ is interpreted as a 2-dimensional structural representation. Let $X \subseteq \mathcal{M}$ be a subset of molecules, the screening library, and let $f : X \to \mathbb{R}$ be our screening function where $f(m)$ for $m \in X$ returns a real-valued score. The general goal of the computational leaning drug hunter can be expressed simply,

$$m^* = \arg\min_{m \in X} f(m). \tag{3.3}$$

The goal is to determine the best (or best set) of hits or produce a ranking over X. In

practice, the difficulty is in picking the correct scoring regime f, and ensuring the library size, $|X|$, is tractable. Currently, the only known solution to this optimization problem is pure dicrete search. Naively, the computational runtime of equation 3.3 is the product of library size and the runtime of f. If the ranking we compute ends up being equivalent to the ground truth ranking, then we have a solution to finding drugs—so long as it appears in the library X. HTVS can be done directly on a set of compounds [Lyu et al., 2019], or can be done on fragments for a fragment screen [Kawatkar et al., 2009].

As result, two main arguments appear regarding research directions for improving our ability to discover drugs. We need more accurate virtual screening methods (i.e. new ways of trying to accurately compute a ranking or scoring function f), and new ways of expanding the size of the screening library X (either through speeding-up the function f or discovering new sets of compounds). It should be noted generating large subsets of $\mathcal{M}$ is not trivial and is an active area of research [Ruddigkeit et al., 2012]. $\mathcal{M}$, restricted to just "drug-like" molecules, is conservatively estimated to contain 10^{33} distinct compounds [Polishchuk et al., 2013]. This number is theoretical. In reality, chemical databases have enumerated much smaller portions of $\mathcal{M}$. PubChem has enumerated approximately 100 million distinct chemical structures [Kim et al., 2019a], Enamine Real Database contains over 2 billion enumerated compounds [4], SAVI contains over 15 billion enumerated compounds [Patel et al., 2020], and GDB-17 contains 166 billion compounds [Ruddigkeit et al., 2012].[5]

Besides determining whether or not a drug might be active, there are two other criteria hits must pass in order to continue as a lead for experimental characterization: safety and synthesis. Drugs are designed for human use ultimately—and so designing safe and non-toxic drugs is essential. Rather than waiting until preclinical testing to determine exact

4. `https://enamine.net/compound-collections/real-compounds/real-database`

5. Enamine and SAVI contain compounds with estimated synthesis pathways. Enamine Real Library has an 80% synthesis success rate. GDB-17 contains no synthesis guarantees or estimates and is an exhaustive combinatorial search over structures with very specific constraints (such as *leq* 17 atoms).

safety parameters of a compound, there are early stage computational techniques to estimate the safety of a compound. One such technique studies the so-called ADME properties: absorption, distribution, metabolism and elimination [Kassel, 2004]. Synthesis is another important and necessary topic. If any compounds in the library do not have a defined synthesis pathway, one must be found in order to continue testing the drug. Both of these topics are outside the scope of our study, and we bundle into "downstream" tasks for a hit once it is found.

3.3 Virtual Drug Screening

In this section, we cover studies in virtual screening for ligand based drug design and fragment based drug design. We outline an example of utilizing cancer-genomic data and experimental assays in cell-lines to identify drugs that likely inhibit tumor growth. This is ligand based drug design since no structural information is used, no protein targets are identified. Rather, we work directly on the idea that the screening problem is designed as $d_{\text{obs}} = F(t, d)$ where F is the observed response of cell t with the application of drug d. Then, in the following subsection I outline an example of fragment based drug design based on work with the the COVID Moonshot is a non-profit, open-science consortium of scientists [Achdout et al., 2020a].

3.3.1 Ligand-based drug design

While the main focus of this work is in SBDD, we will briefly outline an application of LBDD utilizing deep learning techniques. The crux of precision oncology and virtual drug screening lies in the relationship between molecular structure and cancer genetics. Untangling this relationship has historically consisted of various genome wide association studies and quantitative structure-activity models. While both have provided the field an in depth view of cancer and the agents that cure the issue for many, the protocols lacked the throughput

to study the association between the compound and specific cancer cell line [Csermely et al., 2013]. It is well studied that cancer genetics may play a role in drug efficacy even across similar cancer types [Candelaria et al., 2005]. In order to understand this relationship, drug screening assays such as the NCI-60 Human Tumor Cell Line Screen were created to drive the precision oncology search for the link [Grever et al., 1992]. Recently, deep learning has been applied to the problem as a means of modeling the interaction between cell genetics and drug properties [Manica et al., 2019, Cortés-Ciriano and Bender, 2019, Xia et al., 2018, Chang et al., 2018].

Classical machine learning models such as random forest have been used in typing cancer genetics and even in single agent drug response models [Menden et al., 2013a]; however, few have shown success across a wide range of cancer cell lines and diverse set of drugs. Deep learning provides a more natural interface as higher dimensional datasets can be applied and reduced to a lower dimensional representation, where the choice of representation of the data is still important [Coates et al., 2011].

Before deep learning became tractable both computationally and on the combined datasets, Jang *et. al* systematically analyzed CCL drug sensitivity modeling as a classical machine learning task, by exploring the multitude of feature modalities, algorithms, prediction targets and more [Jang et al., 2014]. Given that they had a small set of drugs available between the studies (138 compound), the models Jang *et. al* studied did not featurize the drugs in a continuous embedding; rather they were one hot encoded whether present or not. Overall, in explaining the predictive variance between the over 110,000 models tested, they ranked the factors for variance among models, finding genetic features and the particular compound being predicted to be very explanatory, while the algorithm being considerably less important. Their results were further in line with other work on more general learning problems involving genetic features [Shi et al., 2010].

While many drug feature representations such as SMILES based encoders [Manica et al.,

2019] or pharmacophore models [Skalic et al., 2019] are commonly used in deep learning models, we chose to only compare two variants of molecular descriptors: Mordred and Dragon7, which are some of the more common classical feature representations [Moriwaki et al., 2018, Mauri et al., 2006]. Deep learning models may be moving towards more natural representations like graphical representations; however, even the integrated data set does not offer the sort of drug diversity seen in the papers effectively using these newer representations [Dutil et al., 2018].

Across the recent works, the decision to use RNA-seq in conjunction with images, smiles, or fingerprints was a choice dictated by the desired model architecture, explainability of the model, or resources available; however, these choices do not span the entire space of representations available for cells nor drugs. Cortés-Ciriano and Bender found introducing convolutional neural networks (CNNs) for the drug image alongside Morgan fingerprints improved performance of cell line sensitivity predictions with a small effect size [Cortés-Ciriano and Bender, 2019]. *Manica et al.* compared fingerprints to the use of SMILES strings, showing an RMSE improvement from 0.122 ± 0.010 their deep baseline to 0.104 ± 0.005 in IC50 predictions with the benefit of an explainability mechanism for improved mechanism of action (MOA) and perturbation study [Manica et al., 2019]. *Xia et al.* found the use of proteome, expression, and molecular fingerprints to perform the best when studying combinations of drug pairs; however, proteome data is not available at scale of this study [Xia et al., 2018]. Chang et al. utilized a virtual docking technique in conjunction with CNNs to achieve $r^2 > 0.84$ on 244 drugs from GDSC [Chang et al., 2018]. The goal of these models is to provide insight into which drugs and cells should be tested in downstream drug development analysis with, for example, PDX models or organoids. While some evidence exists cell line analysis does not correlate with response in tissue, scoping the target correctly allows interpretation and use outside a pure precision medicine approach [Johnson et al., 2001].

In this work, we curated a collection of cancer cell line screens from four different data

sources. We use the Genomics of Drug Sensitivity (GDSC) [Yang et al., 2013], NCI-60 Human Tumor Cell Lines Screen (NCI-60) [Grever et al., 1992], the Cancer Cell Line Encyclopedia (CCLE) [Barretina et al., 2012], Cancer Therapeutics Response Portal (CTRP) [Klijn et al., 2014], and the Genentech Cell Line Screening Initiative (gSCI) [Rees et al., 2015]. Each dataset consists of a panel of cancer cell lines tested of drugs. The datasets span size and specificity, as the NCI-60 contains the widest set of drugs, and the smallest set of cells. The datasets individually overlap both on some cells and drugs.

Cancer types across the datasets were not reported uniformly and contained some missing data. Using an auto-encoder on RNA-seq data from the Genomic Data Commons [Grossman et al., 2016], we were able to generate type-like labels for all RNA-seq data in our uniform dataset. Due to the imbalanced representation of various cancer types, the data was limited the 21 most prevalent cancer types represented in the combined data frames. We applied a standard cancer type clustering technique to determine this source similarity metric (figure 3.1).

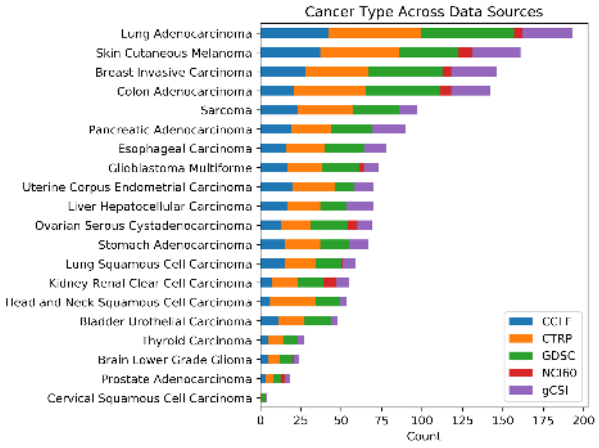


Figure 3.1: **Counts of cells based on type from each dataset used in the training data.** Each data source contains other types not included, but we limit ourselves to the top 21 and top 6 cancer types. The types were determined based on clustering of RNA-seq. While the method is not the same as a somatic tissue type or diagnosis, the type provides an indication of the diversity between data sources when attempting to balance data across classes.

Data Preprocessing

RNA-seq profiles are not provided in standard format by CTRP, gSCI, and GDSC; however, there is a mapping from cells in CCLE and NCI-60 to obtain RNA-seq profiles. After mapping and aligning to the set of drugs from each data frame, the cell response metric was standardized. The NCI-60 reports a dose-response metric based on cell growth after drug application.

The gene expression datasets for cancer cell lines, generated using RNA-seq and mutation data, were collected from the following sources: NCI-60, CCLE, and GDSC. The CTRP and gCSI drug response datasets were generated using the cell lines from CCLE dataset. Hence, for those, we used the gene expression and mutation data from the matching cell lines in the CCLE dataset. We refer to two featurization for cells, the RNA-seq for the expression data and single nucleotide polymorphisms (SNPs) for the mutation data. The SNP data was prepared as counts rather than binary indictor of presence or absence.

The gene expression values were represented as fragments per kilobase of transcript per million (FPKM) values. To create varying gene lists, three datasets were created. The original contains a standard set of genes by an inner join of all the data sets (19,000 features), onco-gene set, and a LINCS1000 set. The LINCS1000 set was derived from the gene in The Library of Integrated Network-Based Cellular Signatures (LINCS) 1000 gene set [Koleti et al., 2017]. The onco-gene set was created from a list of 2054 genes derived from the following three sources: i) 976 "landmark" human genes from high-throughput gene expression assay used in The Library of Integrated Network-Based Cellular Signatures (LINCS) 1000 study, ii) 470 high-confidence cancer genes identified in GDSC1000 study [Iorio et al., 2016], and iii) 1020 genes considered to be cancer genes by OncoKB [Wong et al., 2013].

The genes were filtered based on the genes int he respective dataset list and the FPKM

values were transformed into log TPM values by

$$\log\left(\text{FPKM} \cdot 106/\text{Sum of all FPKM values}\right),$$

and batch effect procresssed according to a few different methods.

When batch effects are in the data frame, the biological signal is not as strong as the effect coming from the various batches (figure 3.2. This effects the downstream analysis such as differential expression and predictive modeling resulting in bias and unpredictable behavior. In order to manage batch effects between the different RNA-seq profiling, we tested three approaches: whole frame scaling, source scaling, and combat scores. *Whole frame scaling*, or applying no batch effect handling method, merges the combined RNA-seq expression values and scaling each feature to unit norm. *Source scaling* scales each feature to unit norm by the source rather than the combined data frame. *Combat scaling* come from combat algorithm from Johnson et al. [2007].

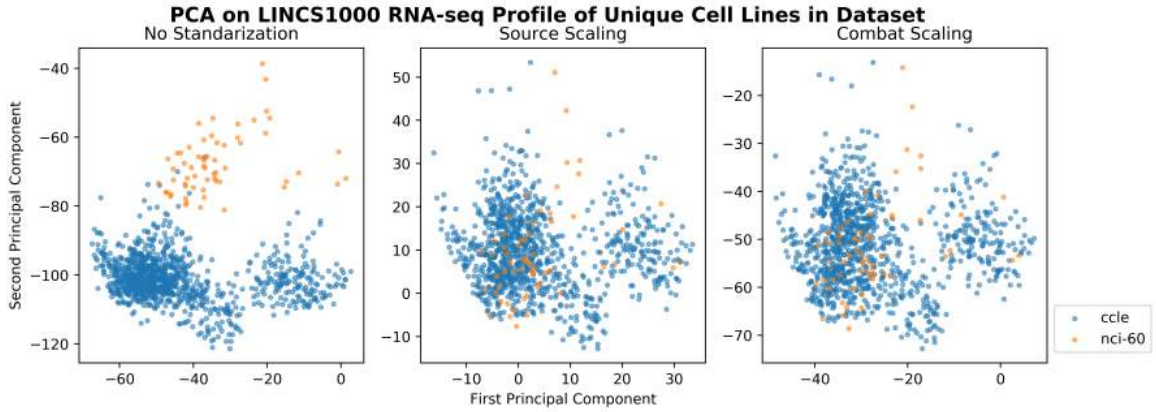


Figure 3.2: **First two components from PCA on CCLE, GDC, and NCI-60 cell lines from our combined data frame.** Without standardization, some batch effects are clear between NCI-60 and CCLE cell lines. The two methods tested seem to eliminate an obvious skew towards various batches.

Target Processing

Ideally, the target metric chosen correlates to growth inhibition used in the NCI-60 dataset for drug response measure given the NCI-60 is the largest data source used. The growth response data for NCI-60 cell lines is prepared by applying the compound at five different dosages to the tumor [Boyd and Paull, 1995]. By staining the cancer cells and measuring the absorbency with automated plate reader, they obtained the absorbency at time-zero, T_z, the control absorbency, C, and the absorbency after the application of the drug, T_i. Growth response used in the datasets is computed as Percentage Growth Inhibition (PGI) by

$$\text{PGI} = \begin{cases} 100\frac{T_i - T_z}{C - T_z} & T_i \geq T_z \\ 100\frac{T_i - T_z}{T_z} & T_i < T_z \end{cases}. \tag{3.4}$$

The PGI value is the target prediction value of most learning problems related to the NCI-60 dataset.

From the PGI values across the different dosages, three values are computed related to the compound and cell. First, Growth inhibition of 50% (GI50) is calculated from setting the PGI to 50% and solving for T_i in Equation 3.4. The Total Growth Inhibition (TGI) is calculated by finding the concentration of the drug where $T_i = T_z$. Lastly the 50% lethal concentration (LC50) is the concentration of drug resulting in a 50% reduction in the measured protein at the end of the drug treatment as compared to that at the beginning, which is the T_i dosage so $(Ti - Tz)/Tz = -0.5$.

However, this metric requires each sample is associated with a cell, drug, and a dosage. In order to remove the dependency on dosage as the interest in cell lines is skewed towards drug screening and further downstream precision medicine where dosage is not a preliminary question, we formulate a dose-independent drug response metric for the datasets provided by some data sources such as CCLE.

Source	r^2 fit	E_∞	HS
CCLE	0.71 ± 1.35	0.31 ± 0.35	1.93 ± 1.50
CTRP	0.70 ± 0.36	0.33 ± 0.38	2.01 ± 1.53
GDSC	0.63 ± 0.40	0.40 ± 0.39	1.88 ± 1.53
NCI-60	0.80 ± 43.16	0.36 ± 0.36	1.93 ± 1.47
gCSI	0.82 ± 0.31	0.34 ± 0.33	1.90 ± 1.28

Table 3.1: **Dose independent fitting results.** E_∞ and HS are parameters from the fit, and r^2 is from the result of the hill curve fit on a per-drug , per-cell basis. The NCI-60's large standard deviation comes from a few extreme outliers that were removed.

For each dataset, we fit the dose dependent response to a hill curve with E_∞, EC50, and hill sloop binding cooperatively (HS) (table 3.1). From the fit curve, we computed AUC - area under growth curve for a fixed dose range between 4 and 10 $-log_{10}(M)$, IC50 - drug dose to have 50% growth, EC50se - standard error of the estimated EC50, R2fit - R2 score between the unclipped, real growths and fitted growth values, AUC1 - area under growth curve for the measured dose range in a study, AAC1 - area above growth curve for the measured dose range in a study, and DDS1 - drug sensitivity score [Yadav et al., 2015].

While all the metrics are highly correlated, in this paper we study AUC for a fixed dose range across all studies from 4 to $10 - \log_{10}(M)$. A value of 1 for AUC indicates a cell does not respond at dosages while a value of 0 indicates a cell responds completely at all dosages. Given most cell lines do not respond to a particular drug, the predictive target AUC is highly skewed (Figure 3.3).

Molecular Representations

The representation of molecules plays a crucial role in response prediction for it directly indicates what kind and how much information will given into the deep learning models during training and inference.

We can divide the representations into engineered (knowledge-based) or non-engineered ones based on whether the featurization process involves the domain knowledge on molecular

chemistry; alternatively, from the learning and data presentation point of view, the molecular representations can be either euclidean (vectors, tensors, voxels, etc.)or geometric (meshes, graphs, point clouds, etc.) ones.

The most basic but yet still effective representation of molecules is SMILES string. Introduced in Anderson et al. [1987] in the late 1980s, SMILES (simplified molecular-input line-entry system) converts molecules into unique and human-understandable ASCII strings based on a given set of rules, which can be then processed with the methods and models in natural language processing. In our experiment, we used the canonical SMILES string, encoded them based on characters, then fed the vectors into deep learning models. Successful results has been demonstrated [Goh et al., 2017, Kwon and Yoon, 2017, Hirohara et al., 2018] with this approach despite the fact that this vector representation requires minimal domain knowledge.

Molecular fingerprint is a widely used method of encoding molecules based on the presence or absence of particular substructures. There are different ways to implement such substructure encoding, some of them are non-exhaustive like MACCS, which only has 166 structural key, while the others are exhaustive but differs in searching patterns (circular-, path-, or tree-based searching) and substructure specifications (like the number of atoms in substructure). In our experiment, we used ECFP[Rogers and Hahn, 2010a], one of the most commonly used fingerprint, with different substructure size and vector dimensions to search for the most effective ECFP features for drug response.

Molecular descriptor is a more broad definition, which usually requires more domain knowledge and feature-engineering to generate. Generally speaking, any numeric representation that can "describe" the molecules in a certain way, is a descriptor. For the sake of better performance, a good set of descriptors often covers a wide range of molecular properties, some of which are rather complex and demands understanding of chemistry at a very deep level to design. In our experiment, we have tried Mordred [Moriwaki et al., 2018] and

Dragon descriptors, both of which are highly popular and widely used in other works. In this work, we limit the exploration to molecular descriptors from MOrdered and Dragon7 code bases.

DNN Models

Unlike classical algorithms studied in previous sweeps over this problem, deep neural networks are highly parameterized over architecture and training strategy [Zou et al., 2018]. A study of greater scale could be done just exploring the question of architecture and training strategy for a single set of features and data. Arbitrarily, we selected three model architectures based on experience from training models on the data, with the hope the three models are different enough to capture any interesting variance between them.

We present three model architectures, a deep model, a model with a differential dropout scheme (inspired by the observations from [Zheng et al., 2014]), and a model with a sigmoid channel gated attention mechanism. The standard deep model is a simple ReLU based multi-layer perceptron (MLP). The differential dropout model is the same model except the dropout rates decrease to zero towards the output layer. The multiplicative channel gating is a two tower MLP, one for the genetic features of the cell and one for the drug features. The towers are combined using a multiplicative channel gate.

Multiplicative Channel Gating

Attention mechanisms are used successfully in natural language processing (NLP) applications. Attention layers in NLP take as input a set of keys, values, and queries and attend to certain parts of the sequence to highlight important parts of the prediction [Vaswani et al., 2017a]. Image attention expands the use case to images. Image attention has been used for image captioning where the model "attends" to a region in the image to predict word used to caption that part of the image [Xu et al., 2015]. We consider a self-attention mechanism

between linear layers of a network, where an activation function (sigmoid, tanh, softmax, etc.) is applied to one channel and multiplied onto another, effectively attending to certain values of the activation. We employ a variety of generic activation functions such as sigmoid, tanh, or softmax. Unlike a standard self-attention implementation such as the one used in Monica et. al [Manica et al., 2019], we do not enforce the use of softmax which can dimension the value of the gradients for non-sequence data. By using a sigmoid or tanh function, each channel is gated with the possibility of decreasing the value, though there is no restriction on how much information passes through the layer, unlike a softmax function.

Model Training

Three training methods were used across all models. Training was performed using Pytorch [Paszke et al., 2017]. All models were trained on Summit, where each model used a single NVIDIA Volta V100. The models were dispatched for training and scoring using the Cancer Distributed Learning Environment (CANDLE) Supervisor [Wozniak et al., 2018a]. Two optimizers, stochastic gradient descent (SGD) and Adam optimizer were tested with the learning rate initially set to 8e-4 for both[Kingma and Ba, 2014]. The learning rate was reduced by a quarter when the validation loss did not decrease for over 20 epochs. The batch size was not varied and set to 512. Each model was set to train for 400 epochs, and stop early if validation loss stopped decreasing after 10 epochs. We trained using Huber loss (Smooth L1 loss) with $\delta = 1$ [Huber, 1964].

Besides the standard vanilla training procedure, two other training variations were used. Due to the high imbalance of positive leads (AUC ≤ 0.5), an imbalanced data sampling strategy was used for each batch or a weighted loss function [Byrd and Lipton, 2018, Lauron and Pabico, 2016]. Imbalanced data sampling fills each batch with a balanced number of responders and non-responders. The drawback of this method is the model sees some data more than others which may lead to over-fitting on the smaller class. Loss weighting

is generally used for classification where different weights are applied to different target predictions. As the models in this study are trained for a regression task, loss weighting for this context involves multiplying the loss of each batch by $1 - y$, the target, in order to produce stronger gradients for non-responders and weaker gradients for responders—the idea coming from the the responders, having few examples, not being able to influence the gradient enough to push the model towards learning the smaller class. Neither strategy was tested in conjunction with the other.

Model Evaluation

Prior works have utilized strict and non-strict partitioning for both cell lines and drugs in the training and testing set. Manica et al. marks the distinction between a strict and lenient split by cell and drug identity in the training set [Manica et al., 2019]. Chang et al. used a leniant split for the validation data [Chang et al., 2018]. Given our dose-independent prediction target, a specific cell and specific drug constitutes a single training example only in terms of model training and testing.

Outside of pure model performance testing, there are two use cases we targeted: drug screening and precision medicine screening. For drug screening, panels are often performed against the set collection of cell lines to determine if a drug should move on to PDX models for further testing. In this case, we partition the training and test set by unique molecules, and use a 3-fold cross validation where each fold consist of entirely different molecules. For precision medicine, there is a list of approved drugs or known agents and the medical question is regarding which treatment to offer to a cell. In this case, the 3-fold cross validation is done over the unique cell lines, where each fold consists of unique cells not in any other fold. For the sake of testing more models and feature combinations, we did not perform a completely strict model evaluation of unseen drugs and cells as the use case is outside the scope of this research.

While this problem is posed as a regression problem as the AUC values are continuous on $0 < AUC \leq 1$, viewing the problem as classification links it directly to application of the model. Due to the extreme skew of the training/testing distributions, r^2 and root mean squared error (RMSE) may not represent whether or not the model performs well in the region of interest (the model may minimize residuals over a large mass of the data, ignoring the residuals separating interesting response from no response). By artificially selecting a cutoff, 0.5, we can determine for the case of drug screening or precision medicine whether or not the model is learning to distinguish a responder cell/drug or a non-responder. The balance between type I and type II error is a calculation that is use-case specific and the selection of the feature set and model from our sweep will require consideration of this trade-off. We further justify this unusual approach to classification by the result from Jang *et. al* indicating discretized target variables created less performant models from [Jang et al., 2014]

Analysis of Results

In order to test thoroughly the various proposed feature and training combinations, we tested the full combination space. Given 198 feature sets entailing various scaling, cell profile types, and drug representations, we ran each feature set on each model, with three different training strategies, top 6 and top 21 datasets, and three cross validation folds. In total, we ran 35,320 models on Summit. We gathered the results, and concatenates the predictions on each training fold from the model to estimate average statistics for the model.

At this scale of models, some models (15%) did not converge or provide reasonable results. Incorporating those failures into the overall performance of a model would not encapsulate the possible performance given hyper-parameter tuning and detailed work with a given configuration. We present models ranked by balanced accuracy and MCC to illustrate classification performance with post-processing binning and r^2 scores to illustrate the

predictive power of the regression view of the problem.

After gathering data, we removed models we believed to have not converged or over fit based on having one of the following criteria: TPR or FPR of 1.0, negative or zero r^2 score, or a loss outside the 85th quantile. This removed 2045 models from the data for analysis.

In the following sections we highlight comparisons between the drug validation models and the cell validation models. The drug validations did not perform nearly as well as the cell validation models. We believe drug validation is a harder test on the model as the test is based on drug discovery rather than preclinical screening for a fixed set of drugs. For metrics like Mathew correlation coefficient (MCC) and r^2 score we report 99th percentile, and for root mean squared error we report (0.01) percentile. We report multiple metrics as different purposes may arise for these models. We chose to report at 99th percentile versus best models found, as it is unclear with limited resources a researcher could reproduce the best model, and we aim to discuss approaches to the problem that can be taken out of the box and be performant.

Results

We report on the results of models based on predictions on their assigned validation set and metrics. In terms of understanding the parameter space, we ran a series of classical machine learning predictors trained on the hyperparameter configuration to predict our four key metrics: RMSE, r^2, balanced accuracy, and MCC (Figure 3.3). The results indicate standard deep learning hyperparameters to play the most important role in variance, rather than specific feature information, as optimizer, model architecture, training strategy, and dropout all are more predictive using SNPs or the featurization of drugs or cells.

The inclusion of SNPs improves performance on cell validation split models in RMSE, r^2, MCC, and balanced accuracy (Table 3.2). There was no calculated metric in our test suite which reported a worse score when SNPs were included (at 100% and 99% percentiles). For

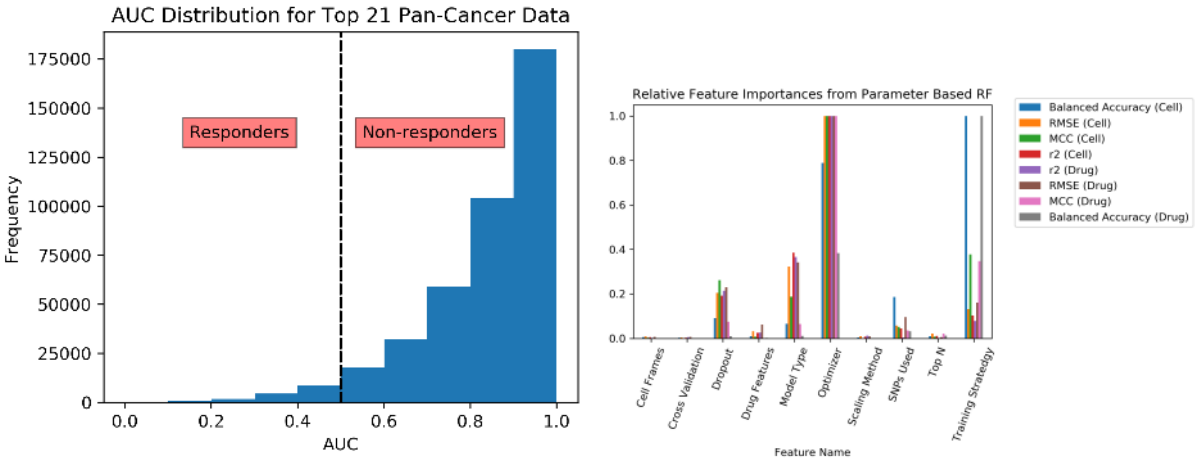


Figure 3.3: **Distributions of labels and feature importance.** *(Left)* The AUC distribution of pan-cancer data frame. With this scheme, the training distribution has 3.86% responders. While our cutoff of 0.5 is arbitrary, learning to distinguish this slice along with a regression-based training strategy will prevent standard regression metrics from appearing much better than they are on the skewed portion of the dataset. *(Right)*Relative feature importance for hyperparameters used in model training for predicting validation metric using decision trees. The r^2 for those models were > 0.9 on cell validation metrics, and > 0.6 for drug validation metrics. Optimizer, model type, and dropout were among the top three features, though the training strategy very important when predicting balanced accuracy.

the on cell validation method, the independent t-test shows significant improvement RMSE and r^2 scores with the inclusion of SNP features ($p < 10^{-5}$), and less significant improvement for balanced accuracy and MCC ($p < 0.17$ and $p < 0.02$ respectively). For drug validation models, the RMSE and r^2 are again significantly different with the inclusion of SNP features ($p < 0.003$ and $p < 0.0005$) and not as significant for balanced accuracy and MCC ($p < 0.01$ and $p < 0.3$).

Scaling methods did not seem to have a large effect on the variance between models besides source scaling seemed to improve measures slightly (Table 3.3. According to an ANOVA the null hypothesis of similarity among scaling methods is not rejected for either validation method ($p < 0.14$). Cell frame splits did not have any noticeable differences amongst them). According to an ANOVA test for fit, there is no significant difference among cell validation methods ($p < 0.31$) and the drug validation method shows a possible

difference but nothing significant ($p < 0.06$).

Chemical informatics package used to features the molecules did not have a significant effect on the validation scores (Table 3.5). For on cell validation models, no reported metric rejects the hypothesis that all featurization are the same for $p < 0.09$; however, for on drug validation RMSE and r^2 scores are effected by featurization to an extent ($p < 0.0001$). We applied post hoc analysis to these two metrics using John Turkey's HSD analysis which shows Dragon7 and MOrdred slightly different on r^2 and RMSE ($p < 0.05$). The use of both MOrdred and Dragon7 however is significantly different from just using Dragon7 on both metrics ($p < 0.01$).

Validation Method	SNPs Included	RMSE	r^2 score	Balanced Accuracy	MCC
On Cell	False	0.086	0.675	0.889	0.553
	True	0.083	0.712	0.896	0.577
On Drug	False	0.108	0.448	0.792	0.460
	True	0.107	0.495	0.807	0.487

Table 3.2: **Metric comparison at 99-percentile of grouped by the model validation strategy and the inclusion of SNPs.** Rows do not represent a single model with those three metrics, rather a model exists with one of those metrics. A t-test for difference between group means shows a significant difference in r^2 scores $p = 1.9e-18$ and 0.0005 for cell and drug validation methods respectively.

Validation Method	RNA-seq Scaling Method	RMSE	r^2 score	Balanced Accuracy	MCC
On Cell	Combat	0.084	0.708	0.893	0.568
	None	0.084	0.703	0.893	0.565
	Source Scaled	0.084	0.710	0.891	0.568
On Drug	Combat	0.107	0.484	0.797	0.478
	None	0.110	0.470	0.795	0.474
	Source Scaled	0.108	0.494	0.796	0.473

Table 3.3: **Metric comparison at 99-percentile of grouped by the model validation strategy and the scaling method used.** Rows do not represent a single model with those three metrics, rather a model exists with one of those metrics.

There was a small detectable difference between the models. The differential dropout

Validation Method	RNA-seq Feature Set	RMSE	r^2 score	Balanced Accuracy	MCC
	lincs1000	0.084	0.708	0.895	0.567
On Cell	oncogenes	0.084	0.706	0.891	0.574
	oncogenes & lincs1000	0.085	0.708	0.894	0.568
	lincs1000	0.108	0.478	0.796	0.482
On Drug	oncogenes	0.107	0.478	0.788	0.463
	oncogenes & lincs1000	0.108	0.487	0.810	0.478

Table 3.4: **Metric comparison at 99-percentile of grouped by the model validation strategy and the RNA-seq feature set used.** Rows do not represent a single model with those three metrics, rather a model exists with one of those metrics.

Validation Method	Drug Featurization	RMSE	r^2 score	Balanced Accuracy	MCC
	Dragon7 Descriptors	0.084	0.706	0.894	0.570
On Cell	Mordred Descriptors	0.084	0.708	0.889	0.568
	Mordred and Dragon7 Descriptors	0.084	0.709	0.896	0.567
	Dragon7 Descriptors	0.108	0.478	0.793	0.466
On Drug	Mordred Descriptors	0.107	0.473	0.811	0.481
	Mordred and Dragon7 Descriptors	0.107	0.491	0.793	0.471

Table 3.5: **Metric comparison at 99-percentile of grouped by the model validation strategy and the drug featurization method used.** Rows do not represent a single model with those three metrics, rather a model exists with one of those metrics.

strategy seemed to improve model performance though between the two tested dropout rates the RMSE 1-percentile were both 0.084 and for drug validation 0.108 and 0.107 (Table 3.6).

Training strategies had a unexpected impact on the data. In the cell validation split models and the drug validation split models, imbalanced sampling produced the lowest validation loss scores but also the lowest validation r^2 scores

We find the improvement in RMSE and decrease in r^2 score to be an interesting result, and this supports one's instinct about the data–that it is widely too imbalanced for most metrics to effectively explain model predictions.

Validation Method	Model	RMSE	r^2 score	Balanced Accuracy	MCC
	Differential Dropout	0.083	0.712	0.884	0.573
On Cell	Sigmoid Channel Gating	0.084	0.708	0.887	0.561
	Standard Deep Model	0.085	0.702	0.897	0.568
	Differential Dropout	0.106	0.488	0.792	0.488
On Drug	Sigmoid Channel Gating	0.108	0.480	0.796	0.470
	Standard Deep Model	0.108	0.484	0.801	0.474

Table 3.6: Metric comparison at 99-percentile, grouped by model used for training.

Validation Method	Training Strategy	RMSE	r^2 score	Balanced Accuracy	MCC
	Class Balanced Sampling	0.089	0.685	0.899	0.585
On Cell	Loss Weighting	0.084	0.710	0.780	0.550
	Vanilla Training	0.084	0.710	0.781	0.548
	Class Balanced Sampling	0.112	0.466	0.825	0.499
On Drug	Loss Weighting	0.107	0.484	0.689	0.450
	Vanilla Training	0.107	0.485	0.693	0.453

Table 3.7: **Metric comparison at 99 percentile for training strategy.**

Discussion

We first comment on the overall model performance of a few models created in this study. Second, we evaluate prior claims that feature representation should matter as the primary explanation of variance between models. We finally evaluate the three significant findings to consider in future research when building DNN models on CCL sensitivity predictions such as the drug featurization, training strategies, and the inclusion of SNPs.

In general, we found the best models to perform quite well on the validation data in both regression and classification contexts. There are models with validation scores on unseen cells of $r^2 > 0.7$ and balanced accuracy $> 90\%$, and there are models with validation on unseen drugs with $r^2 > 0.55$ and balanced accuracy $> 84\%$. The cross study performance of the two best cell validation models (Table 3.9) are on par with several other studies, though not obviously more predictive on a single drug basis. These model is also top ranked if a simple average is taken across the cross study measures as well. Models validated with

unseen cells outperform models trained with unseen drugs in, however the performance varies across the studies where (Figure 3.5). Given classical ML methods were generally capable of single drug predictions, we expected the cell validation performance to be better than the drug validation. Drug validation is a harder problem given the diversity of chemical space—a model that is largely successful on th drug validation problem can be used as a virtual screening tool for compounds. While it may appear the CCLE samples performed better with drug validation, this is an anomaly given at most two CCLE drugs appeared in that validation set.

Considering the case of a classification problem is often more actionable than the regression case for drug discovery and virtual screening tasks. The specific task of the model would be to filter models molecules based on their expected performance on CCLs which can be done at massive scale and quickly on a GPU, where inference times for these models well outperform standard virtual docking libraries (approx. 5000 samples per second, less if featurizing molecule on the fly). While other techniques such as virtual docking and simulation experiments can perform similar tasks by rank-ordering large virtual libraries, we propose the screening results from these models' predictions can be another measure designers of CCL panels and initial drug datasets can employ. The use of these models in the creation of new CCL panels would hopefully be able to balance the distribution of responders versus non-responders further. Given the models are regressors, a cutoff can be selected both at training time for using the balanced data sampling strategy and used afterwards for a sensitivity detection cutoff (we used 0.5, for example). Type I error (false positives) is a more costly error in the precision medicine task where the assignment of an incorrect drug is a wasted treatment opportunity, while type II error (false negatives) is costly in the virtual library screening where missing a potential lead at an early stage is a costly error. Of course, both errors should be considered for either task, but we aim to highlight the diversity of models possible just by training a large collection of DNN models. In Figure 3.4 we show

post-processing cutoff analysis against the errors. By alerting the regression cutoff to 0.3, one can achieve $> 90\%$ balanced accuracy on the validation screening problem. We believe the cost analysis of virtual screening models is beneficial to researchers, where the choice of model used in screening should come from the task at hand, not necessarily the best state of the art performance. Further work can be done to introduce uncertainty quantification into these models using techniques in [Lakshminarayanan et al., 2017].

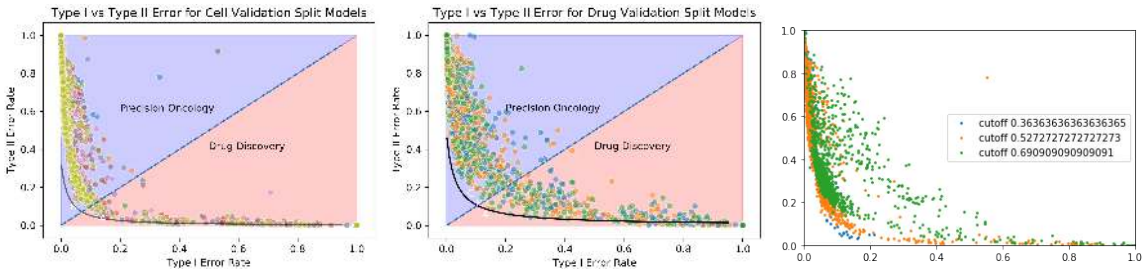


Figure 3.4: **Error trade-off.** type I error (false positive rate) vs type II error (false negative rate) trade off for cell validated models (left) and drug validated models. Cut-off analysis is presented (right) for drug validated models, where the boundary of the error trade-off curve can be shifted with post-processing for even finer control over virtual screening error.

We now move on to discuss the variance of models predictive performance. While the prior classical assessment viewed the input data as the most significant point of variance, our results indicate the parameterization of the deep learning problem itself to be the largest explanation of variance. Based on the results, we see the transition from classical machine learning to deep learning will require a shift in focus from pure-feature engineering to deep learning hyperparameter optimization. Although we did not test discordant feature sets (all genetic features were super sets of each other, and the drug features techniques were both descriptor and fingerprint methods), one would expect to see a large variance across them, though the predictive signal seems stable across all input features. This implies better representations exist for drugs and cells, or perhaps the features we have selected are optimal and a smaller set can be created.

While the hyperparameter decision tree ranked non-feature related aspects higher than the cell and drug featurizations, the result from the percentile and ANOVA analysis should

not be understated, attributing significant to use of SNP data. For instance, the quantile analysis shows a greater difference in MCC on cell validation for the inclusion of SNPs than the differences between MCC for the various model architectures; yet, the decision tree feature importance for the cell validation ranks model architecture as a more useful predictor than the inclusion of SNP frames. We believe this should be read a probability difference, rather than a discrepancy. The decision tree is attempting to model the entire distribution of models we trained, so the fact that the optimizer has a big impact is only due to the fact that SGD is more likely to not converge or converge to the optimum point without hyperparameter optimization while Adam is likely to converge to the same point. In a sense, for researchers getting into deep learning, time spent understanding hyperparameters of deep neural networks has a higher probability of paying off than feature engineering initially. The quantile analysis, however, does indicate there are pay-offs to exploring feature sets, only if one is willing to pay the computational cost of hyperparameter sweeps or optimization.

The results from this study indicate the choice of handling batch effect, general architecture, drug descriptors and modalities, and even the genetic features provided do not seem to impact the performance of the best models. The case of model architecture, dropout, and optimizers this result is not unexpected, as in general optimizers and dropout strategies should have minimal effect on the overall model capacity. Given the various model architectures were designed to have similar number of parameters, it seemed they were not differentiable in the case of general well performance. Batch effect handling, for example, employs a standard neural network to reduce the batch effect in the case of the combat scores; though this would be possible to occur within the first few layers of the network. To check, we present the by study scores of the models in the validation set split by the batch effect standardization procedure (Table 3.8) which again do not seem to indicate a noticeable effect. Other aspects such as dropout and optimizer do not have a global effect on models as we trained many models, however there are smaller effects such as the Adam optimizer

generally performing better than SGD but this may be due to the lack of global parameter optimization, adjusting factors such as batch size and learning rate as it is well studied that SGD requires careful selection of learning schedule [Zeiler, 2012].

As a minor point, the choice of RNAseq genes to use in the model was not an important consideration for model variance. There is that no superset of LINCS1000 gene set clearly outperformed any other set. This is, however, by design of the LINCS1000 gene set, as the selection of genetic features was created to recover most information contained in the transcriptome [Keenan et al., 2018]. Given the ability of the network to learn at such a high drop out rate, the learned weights should have optimal sparse networks. Following a node pruning strategy as a surrogate of feature selection [Jianchang Mao et al., 1994], we performed magnitude tuning on a single model to validate this hypothesis by pruning the initial layer of the network to 99.99% while fine tuning the training with SGD optimizer. The result shows only two of the 127 non-zero weighted features belong to cellular representation while the remaining are drug representations.

The SNPs improved performance all around, when looking at aggregated test statistics. The results suggest that SNPs improved regression metric performance as well as classification metric performance, regardless of the validation strategy. It is known that NCI-60 cell line show over 100x mutation signal compared to other cell line panels. In order to examine if this artifact affected the results, we again break down the performance by cross study. We see that breakdown by study in the analysis further illustrates complexities around this type of study. The results in figure 3.6 indicate however the inclusions of SNPs do not increase the performance across every study when aggregated from validation data. The cross-study results indicate that the larger mutation profile of the NCI-60 is not skewing the result indicating SNPs boost model performance on the cell validation split.

Both MOrderd and Dragon7 are fingerprint and descriptor based methods, with a great overlap between the two in terms of chemiformatics information found. An interesting com-

ponent is the discrepancy between the cell and drug validation measures. When broken down by cross study, GDSC predictive performance is hurt severely when validating on drugs, and general predictions on unseen drugs is less powerful than unseen cells. We believe this is an indication that the models are not learning a useful representation of drugs, and further analysis of different drug modalities should be undertaken, such as images and fingerprints in [Cortés-Ciriano and Bender, 2019].

		Balanced Accuracy				r^2 Score			
validation	scaling	All	CCLE	GDSC	NCI-60	All	CCLE	GDSC	NCI-60
On Cell	Combat	0.893	0.914	0.912	0.902	0.708	0.729	0.619	0.749
	None	0.893	0.918	0.906	0.904	0.703	0.73	0.617	0.743
	Source Scaled	0.891	0.91	0.904	0.899	0.71	0.727	0.617	0.747
On Drug	Combat	0.797	0.945	0.799	0.854	0.484	0.762	0.378	0.607
	None	0.795	0.944	0.794	0.854	0.47	0.761	0.388	0.598
	Source Scaled	0.796	0.946	0.808	0.839	0.501	0.748	0.397	0.602

Table 3.8: **Breakdown of data standardization techniques across the different studies in the validation data.**

	Best Regression Model				Best Classification Model			
	r^2 score	RMSE	Balanced Accuracy	MCC	r^2 score	RMSE	Balanced Accuracy	MCC
All	0.73	0.078	0.69	0.47	0.56	0.104	0.91	0.59
CCLE	0.70	0.086	0.71	0.53	0.68	0.087	0.90	0.69
GDSC	0.57	0.098	0.74	0.53	0.52	0.106	0.87	0.60
NCI60	0.76	0.075	0.68	0.45	0.56	0.103	0.92	0.58

Table 3.9: **Individual metrics for cell split validation for two of most useful models.** The regression model is a differential dropout model with an initial dropout rate of 0.45, trained with the top 21 cancer type samples using loss weighting on the samples. Cells were featurized with SNPs and RNAseq from the LINCS1000 subset, and chemicals were featurized by Dragon7 descriptors. The classification model is trained only on the top 6 cancer types and is the standard MLP model with a dropout rate of 0.2. Cells were featurized with SNPs and RNAseq from the LINCS1000 subset, and chemicals were featurized by MOrdred descriptors. Both models used the Adam optimizer.

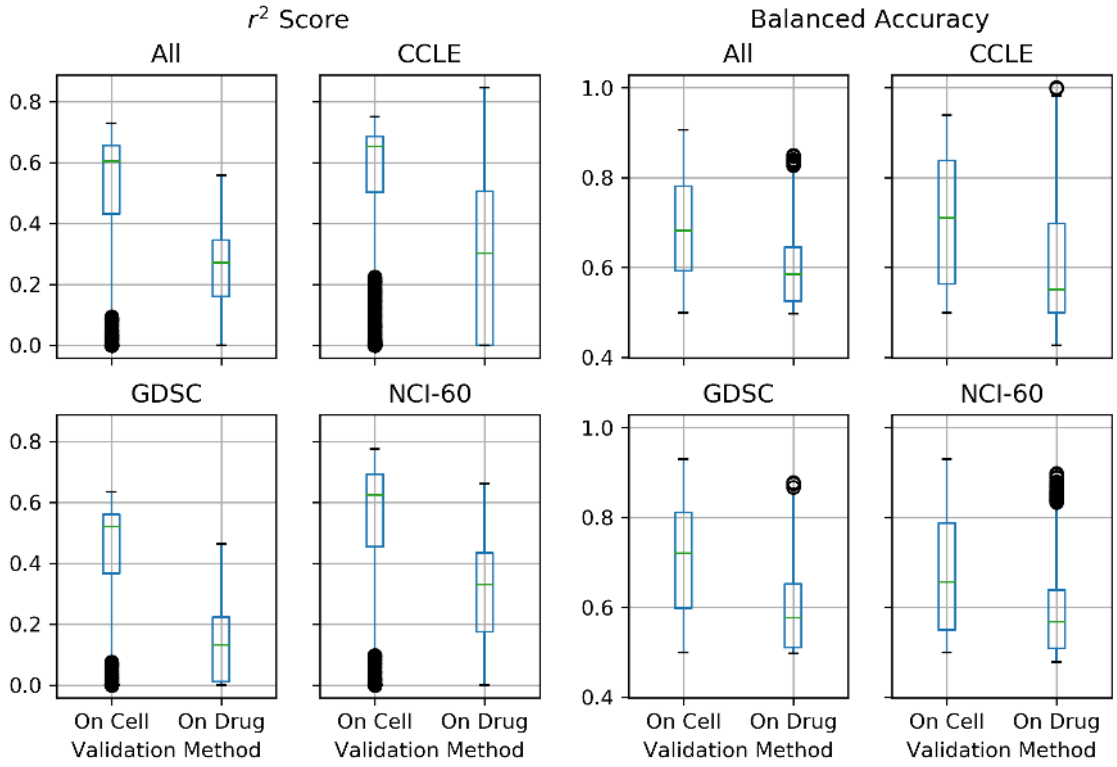


Figure 3.5: **Comparison of the models' performances between validation methods over all converged trained models broken down by sample's originating study.**

3.4 Structure-based drug design

There are many different approaches to structure-based drug design, such as expert design. Experts may use prior knowledge of drugs to design drugs while looking at the particular pocket of a protein. Advances in virtual reality for example allow scientists to perform expert-guided design in 3D. Here, we focus on a computational tool, protein-ligand docking, which is a fast approximation to score the overall viability of drug. There are two main forces looked at, shape and color. Shape refers to the physical question: does a conformation, or 3D position of the molecule, fit in the pocket of the protein? Color refers to chemical and biophysical properties such as pharmacaphores or electrostatics which evaluate the stability or likelihood that a drug would bind.

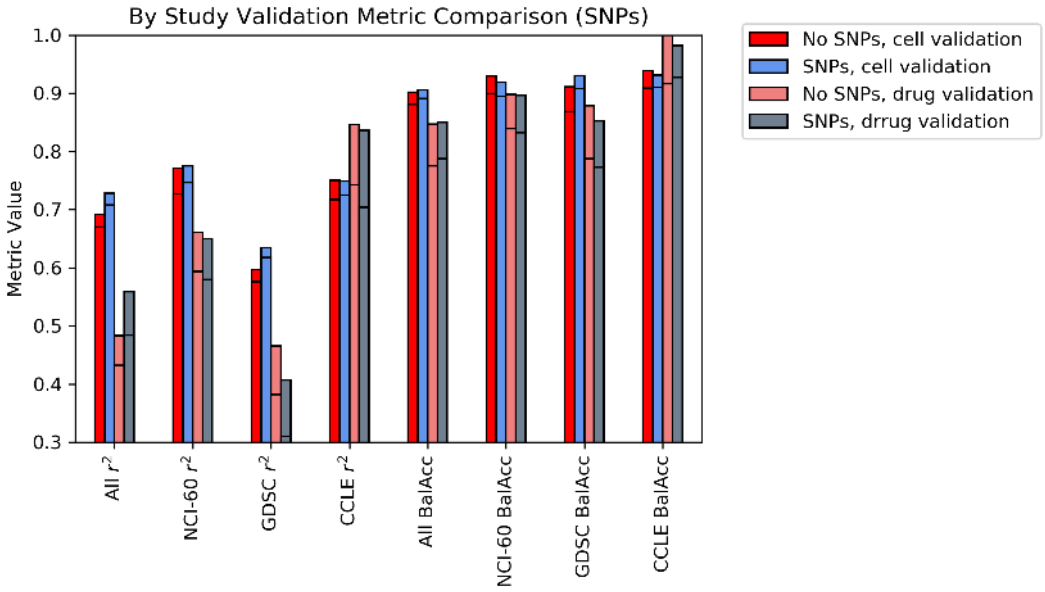


Figure 3.6: **Cross study analysis of SNPs, 98th percentile and max for each score.**

3.4.1 Protein-ligand docking

Classical docking studies involve a compound database to screen, protein annotated with a binding region, and a scoring function (see figure 3.7 for an overview of docking). Compound databases are readily available in various flavors. Compounds selected for VS library size should leave some room for optimization, being libraries of lead-like or fragment-like molecules [Oprea, 2000, Verdonk et al., 2004, Klebe, 2006]. An important but sometimes overlooked preparation for docking is enumerating 3D conformations of compounds in the library (B). It should be noted that even in databases that provide 3D conformations, the sampling density of those poses may not be sufficient for rigid-structure docking (the most common type). Without the correct ligand conformation enumerated from the database, the pose may be incorrect, and this is one of the leading causes of error in computational docking [Warren et al., 2006]. The protein binding region's preparation and the location is an important consideration but outside the scope of this paper (see [Le Guilloux et al., 2009] for a discussion of protein pocket detection). General caveats to docking are not discussed in detail here (see [Cole et al., 2005] for references); however, scoring is considered the pinnacle

to the success of uHTVS computational docking for virtual screening while pose prediction is considered acceptable for most uses [Kitchen et al., 2004].

3.4.2 Scoring Functions

Scoring functions are programs—the scoring function determines which pose is selected from exhaustive search and how the resulting molecule and pose rank amongst the particular dataset. While some docking protocols expand the typical exhaustive mapping of a scoring function over possible positions, the distinction between protocols boils down to the scoring function [Cross et al., 2009]. The scoring function was initially intended solely to detect when the ligand was adequately positioned in the pocket (pose prediction) and provide a general assessment of activity (virtual screening); however, with the rise of virtual screening, the scoring function was tasked with a third role: rank-ordering compound libraries from their selected pose binding affinity ranking [Guedes et al., 2018, Clyde et al., 2020c]. Thus, scoring functions are used to score poses to determine the most likely pose of the molecule, the magnitude of which is used to provide a sense of active versus inactive ligands, and lastly, to rank order sets of libraries.

Given the central role of scoring functions, the entire enterprise of molecular docking pragmatically and theoretically rests on these scoring functions' speed and accuracy. Some scoring functions can be slow and particularly well suited for pose prediction, while others may be fast, and only capable of decoy detection [Ballester and Mitchell, 2010]. The scoring function itself considers a ligand pose and is typically statically implemented for a receptor or protein (given it is costly to compute the initial pharmacophores and solvation energies of the protein [Mcgann et al., 2003b]). Consider function f_{receptor} to be a scoring function prepared with a particular protein pocket. We associate the score of a compound $c \in \mathcal{M}$ as

$$S(c) = \max_{\tau \in \text{valid poses}} f_{\text{receptor}}(\tau) \tag{3.5}$$

as a stand-in property to rank the "goodness" of molecules in a database (some authors call this value the predicted binding affinity [Guedes et al., 2018]).

There are many scoring functions used, such as DOCK, GOLD, FlexX. There is research into using the combination, or consensus score, as a more reliably and interpretable metric, such as CScore [Clark et al., 2002, Clyde et al., 2021c]. There is also a flexible based docking protocol, reducing the need for ensembles against static protein receptors [Meiler and Baker, 2006, Razzaghi-Asl et al., 2015]. A general overview of scoring functions, and outlining the differences amongst them for sampling speed is available here [Moitessier et al., 2008, McGaughey et al., 2007].

Given this framework, there are a few options to consider for developing surrogate ML/AI models. The most straightforward approach would be to model the scoring function f_{receptor} and create an ML model as the heart of the docking algorithm; in practice, this is a considerable challenge primarily due to encoding a 3-D ligand pose inside of a protein pocket, which would need to be very sensitive to the position (to satisfy the pose prediction requirement of the scoring function). Of course, there is the option of ignoring the idea of scoring functions and utilizing an ML/AI model to generate the pose directly without the exhaustive search framework—such models tend to suffer from (i) their specificity to a particular scoring function protocol used for the training data [6], (and the imposition of a docking protocol), and (ii) how docking the objective is nearly cast to mimic the physical processes involved in protein-ligand binding, which is usually represented as a multi-objective optimization problem. This dual role of the scoring function imposes on ML/AI model's an undue expectation—the expectation to act as a useful surrogate to model to the actual protein-ligand docking process, and, to find the correct pose immediately (many approaches utilize reinforcement learning/ other techniques to model this). We will discuss in the following the section the different

6. particular scoring function protocol refers to the method for combining the protein pocket information with the theory to produce f_{receptor}, and is not referring to f_{receptor} in specifics (difference between a Gaussian versus discrete scoring function, for example)

workflows that arise.

3.4.3 Active sites

Molecules are keys in our colloquial lock and key model. Fully specifying the key is rather simple in that sense, as the key is a global entity fully specified by structural formulas for molecules. But what about the protein? Molecules are physically much smaller than most proteins found in the human body. In Fig. 2.1 A, is the location of the lock obvious? If one sees the hidden molecule, maybe-so; but, suppose the molecule was not present in the structure we obtained. In this case, we need to identify the active site (lock) of the protein computationally (or experimentally if such resources are available) [English et al., 2001].

Active sites are classified into two categories, binding sites and catalytic sites. We focus on binding sites. Binding sites are regions of the protein where ligands or other compounds form a stable interaction with the compound. Non-covalent interactions are classified into four main forces, electrostatic interactions, Van der Waals force, hydrogen bonding, and hydrophobic interactions.

Computational techniques for binding site detection. A common tool freely available for use is FPocket [Le Guilloux et al., 2009]. These tools work via an important concept in computational drug discovery—simplification to known models. Given the protein structure, a simple model is created using alpha spheres [Liang et al., 1998]. An alpha sphere is a sphere which touches four atoms when overlain with a protein structure, and contains no internal atoms. Pockets or cavities where a molecule may situate around many protein atoms are thus detected by locating spheres with large enough radii but small enough to be very close to many atoms of the protein.

Certain classes of proteins have well defined and studied active sites based on the protein's main function. Kinase are a large class of proteins which transfer phosphate groups from phosphate-donating molecules to specific molecules. In order to achieve this function, the

phosphate-donating molecule (adenosine triphosphate, ATP) must form stable interactions in an active site.

Conformer Generation

Docking programs treat each conformer of a compound as a rigid body. Hence it is essential to generate an ensemble of conformers, each of which is put through rigid body optimization. Generating conformers is typically done prior to docking as this reduces the run time since it only needs to be done once, and can be used regardless of the protein active site under exploration. For example, OpenEye uses OMEGA which uses a library of to identify and enumerate rotatable bonds and flexible rings. These flexible aspects of the ligand are enumerated into different structures, and conformers with clashes or high strain and discarded, while the remaining are clustered are returned. The level of sampling is a hyperparameter, but typically around 200 conformers for each ligand are generated.

Search

Given a particular conformer and a scoring function, exhaustive search is performed where the rigid ligand is attempted to be placed in various positions in the active site. A particular grid spacing is chosen such as 1. The possible rotations and translations for the ligand are enumerated that fit the bounding box of the active site. This leads to an enumeration of scoring. The first step leads to the elimination of poses that clash with particular protein atoms within that box. In particular, this leads to the creation of what is called a negative image. The negative image of the protein is used to quickly filter which poses are possible on a shape basis. Internally, when FRED constructs a scoring function, it uses a small library of compounds with full exhaustive search where this filtering is not used. This leads to then creating a set of poses that are not possible. Top scoring poses, those which do not clash, are used to create a density field, which is converted into a Gaussian density distribution. Each

pose is than scored against this density map, and a cut-off is used to filter. After this initial negative image filtering, millions of poses or just a few hundred remain. Finally, these poses are fully enumerated in 3D, and the scoring function is applied. Finally, optimization is used from the top 100 poses. Essentially, for each remaining pose, the grid spacing is shrunk to say 0.5. Finally, the best pose is returned.

Gaussian potentials

The most common scoring techniques used in traditional high-throughput docking are based on Gaussian potentials [Mcgann et al., 2003b]. Given the active site, a precomputed grid is constructed and then smoothed. All interactions in Chemgauss scoring, for example—the most common scoring function used in this dissertation as part of OpenEye Scientific's FRED docking program—are constructed with step functions. Step functions are used because intermolecular forces are often computed with distance cut-offs to increase tractability; however, smoothing via Gaussian potentials allows these step functions to be less sensitive to small deviations in ligand position. For example, each heavy atom in the protein is assigned a penalty score and cut-off for forces such as van der Waals.

3.5 Conclusion

Drug design is a vast field with many different approaches. First, we explored a complete case study of ligand-based drug design for precision oncology, where cell express is encoded into vectors along side drug encoding and a model is trained on experimental data. Second, we outline the basics of protein-ligand docking through the lens of computational screening. Irrespective of ligand or structure based drug design, both paradigms are computational screening where scores are produced for each object of the design space. In later chapters, we will address how to then sample from this space to increase the likelihood that score reflects the true underlying likelihood.

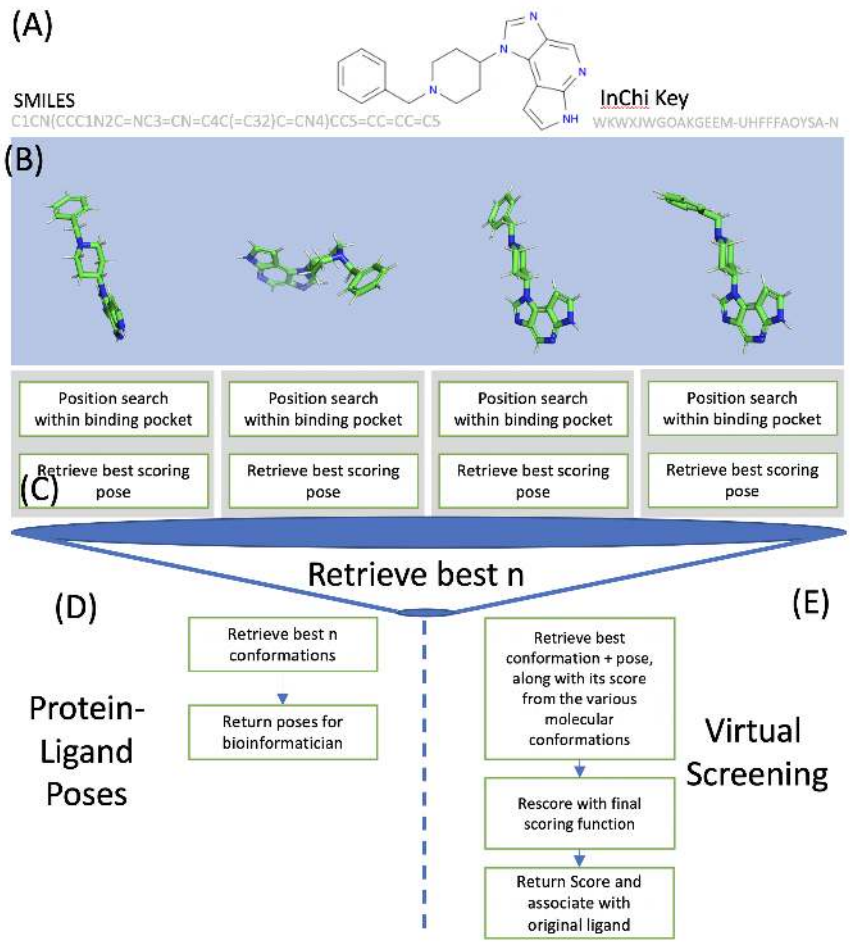


Figure 3.7: **A typical docking overview is broken down into theoretical components.** (A) Compound libraries are often prepared using 2D identifiers such as SMILES or Inchi Keys, though they can be found in 3D formats such as SDF or MOL2. (B) *Conformer generation* creates an ensemble of low energy 3D conformations sampled from the 2/3D compound in the library. Also, in this step, one can increase the ensemble size by enumerating stereoisomers if desired. (C) *docking*: here, poses are optimized within the protein pocket and the final scores for each conformer in the ensemble. (D and E) the analysis begins by retrieving the top scoring ensemble conformer (sometimes top n) for downstream analysis. A typical final step is taking the best scoring pose and score and annotating the original starting structure with that associated score and pose only (not the whole ensemble of scores).

CHAPTER 4

AI AND VIRTUAL SCREENING HPC WORKFLOWS

This chapter focuses on how AI can be introduced into the basic VLS pipeline introduced in the previous chapter through the concept of surrogate models. This will include details on modeling as a data science problem protein-ligand docking, the kinds of workflows that are possible with this application of AI, and a simple uHTVS workflow case study for SARS-CoV-2 3CL-Main Protease.

4.1 AI for Virtual Screening

Virtual screening as previously introduced is the process of applying a computational scoring kernel to a series of objects in a design space. In this section, I illustrate how AI models can be used to replace mechanistic force-field based scoring functions in order to accelerate the process. Given recent growth in hardware accelerators and graphics processing units (GPUs), replacing traditional code with accelerated models applicable to new hardware is a promising area for speeding up the process of virtual ligand screening [Gupta, 2021].

4.1.1 Model Types and Featurizations

Deep learning applied to classical cheminformatics property predictions is a growing area of research [Schütt et al., 2018, Bartók et al., 2017]. Besides finding novel data sources for research in deep learning architectures, high-throughput methods for small molecule property predictions have been used across materials engineering, biology, and pharmaceutical development. This work focuses primarily on small drug-like molecules and the properties essential for pharmaceutical lead discovery and virtual screening. Models for drug discovery are targeted for screening large libraries of compounds on hardware accelerators, opening up the door to typically unscreened regions of chemical space.

However, molecular data has posed a significant challenge from the deep learning perspective: how does one featurize a molecule? Molecules have no obvious featurization technique. A few classes of featurization methods are currently used today: descriptors/fingerprints, graphs, SMILES, 3D wave fields, and 3D point cloud. Descriptors and fingerprints are classical in the sense that they create vectors to represent each molecule based on various chemical informatics properties or shape-kernels. Vectors are typically used to work effortlessly with standard machine learning techniques such as linear regressions or random forests. It was generally believed operating on the graph structure of the molecule itself, with an architecture similar to convolution neural networks (CNN), would prove to be successful, and these models have shown impressive results through various complex architecture inventions. While these methods perform well, the implementations are young and not widely available for users with native framework implementations or hardware accelerators. In the following details, we outline the models and corresponding featurization techniques.

Descriptors

Before deep learning, computational inference problems relied on classical machine learning models based on tabular, vectorized data—a far cry from the nebulous representation molecules exists. Molecular descriptors were created to vectorize molecules into a numerical and thereby commutable representation [Pastor et al., 2000, Yap, 2011, Todeschini and Consonni, 2008]. Various software packages exist open-source with various degrees of customizability. *MOrdred* offers a simple Python interface [Moriwaki et al., 2018]. Molecular descriptors can be used readily with nearly any machine learning technique or deep learning models.

Most descriptor packages consist of hundreds of routines for computing various molecular features, such as molecular weight, number of acid or base groups, or charge. Each of these smaller computations is bundled together in a large vector. Thus, applied over a

large molecular library, a familiar table emerges. Each row is representative of a sample, a molecule, and each column is a particular and explainable feature. This method is preferred for use with classical machine learning techniques as vector 1D features are often required for random forests or linear models.

It should be noted that molecular descriptors are distinct from of molecular fingerprints. Molecular fingerprints, another alternative for featurization, are bit vectors based on hashing different topological neighborhoods of the molecule together. Molecular fingerprints originated for use in databases as a surrogate for molecular similarity. Unlike fingerprints, molecular descriptors are explainable, where fingerprints individual bits are relatively opaque.

Graphs

Representing molecules as graphs is a natural concept to most (scientists widely use 2D depictions of molecules) [Clark et al., 2006, Ebalunode and Zheng, 2009]. Graphs internally represent molecules in standard cheminformatics packages [Babel, 2010, Landrum et al., 2006, OEChem, 2012a]. Graphs are at the core of molecular and neural fingerprints, aggregating hashes from nodes via edges [Duvenaud et al., 2015]. Here we introduce graphs as an added structure to inference problems with molecules.

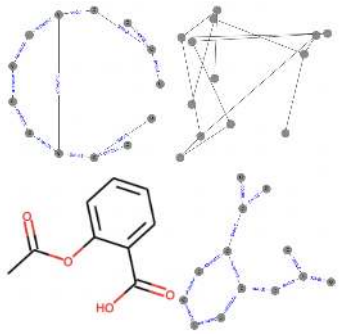


Figure 4.1: **Aspirin as a graph.** Example of molecule Aspirin represented as a graph with different node 2D positioning for illustration.

Sequences, images, and new techniques

Descriptors, sequences, voxelizations, and graphs are among the most common mechanisms for coding drug discovery models [Elton et al., 2019]. While not having appeared in the literature thus far, images are a novel but intriguing featurization technique for molecular docking. As discussed, docking is an exercise, at least in part, in understanding shape complesmentarity (lock-and-key view of protein-ligand inhibition [Tripathi and Bankaitis, 2017]).

A straightforward featurization method has been widely ignored—2D image depictions. 2D depictions of molecules are used widely across most domains working with molecules, from chemistry textbooks to whiteboards across medicinal chemists' offices. From the 2D depiction of a molecule, chemists can generally identify significant properties such as H-acceptors, ballpark the molecule weight, and even determine if a molecule might bind to a protein. Unlike graph structure, this featurization method can utilize off the shelf convolutional neural networks (CNNs), which have historically dominated deep learning research under the wing of computer vision. In computer vision, CNN research has typically relied on ImageNet, a vast collection of labeled images [Deng et al., 2009]. While various approaches have used 3D voxelizations of a molecule, these typically require large sets of parameters with small datasets. Using 2D images, we can initialize our models with pretrained weights that are typically scale and rotation invariant under image classification. The task itself for transfer learning with image models seems to be not relevant, as upper layers of models such as ResNet have been shown to learn a basic understanding of images such as color differences and lines [He et al., 2016].

In conjugation with attention-based models, images can be utilized for inferring explanations for predictions [Nam et al., 2017]. For example, if the model is trained on predicting the number of hydrogen acceptors for a given compound from the image, the attention network should highlight the hydrogen acceptors atoms themselves (see figure 4.2).

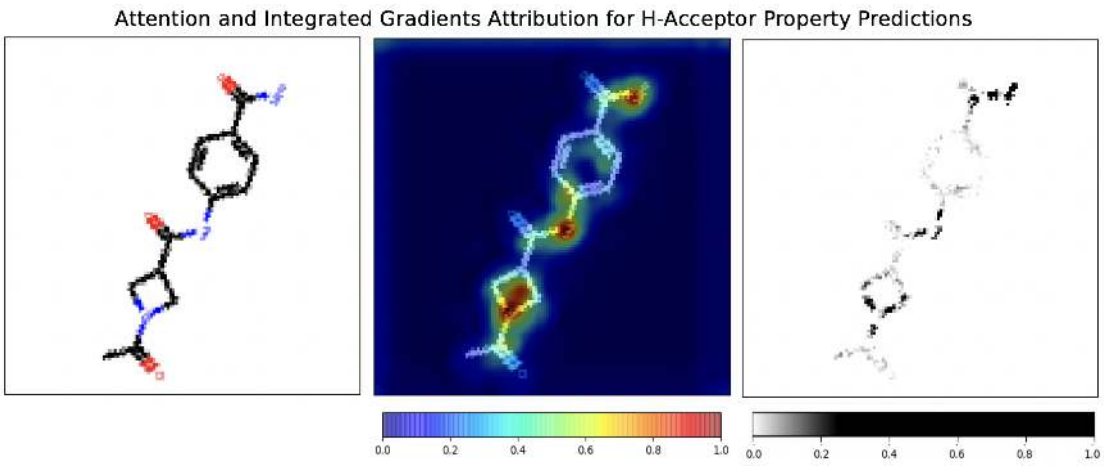


Figure 4.2: **Attention-based deep neural network inference over molecular image depictions**. (left) 2D image depiction of ZINC70817879. This 128 by 128 pixel image is used as the input features for a modified ResNet-101 model. The model is trained to predict the number of H-Acceptors in a compound. (center) Attention values by pixel from a single head attention layer used in the model. Values near 1 indicate the model attended to the region, while closer to 0 indicates the opposite. (right) Integrated gradients feature attribution where 0 indicates less contribution to prediction and 1 is the most contribution to the prediction [Sundararajan et al., 2017]. Both the attention and integrated gradient methods of attribution show the model correctly using H-Acceptors to predict 3 H-acceptors.

4.2 Taxonomy of Workflows

There are a few workflows (fig. 4.3) to consider which integrate AI/ML methods into uHTVS for protein-ligand docking: surrogate models for scoring, surrogate models for pose prediction, or end-to-end in-depth learning solutions. In this work, we outline the building blocks for A-C and leave D as an example of a complicated workflow utilizing the pieces from the models we explore. In chapter 5, the development of surrogate models for protein-ligand docking is specified in detail. In chapter 6, workflows which use filtering in a hierarchical method are outlined. In chapter 8, the analysis of filter models is outlined. Workflows D and C are not studied in detail in this dissertation, and more details can found in references [Xu et al., 2020, Li et al., 2020].

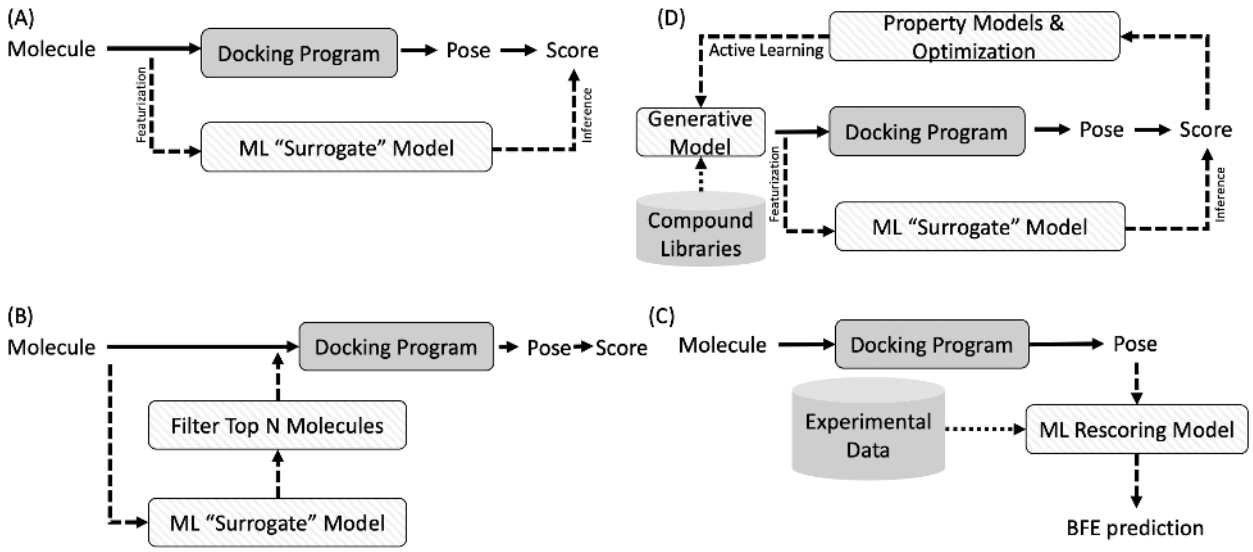


Figure 4.3: Common ML/docking virtual screening workflows. We show docking programs as taking an input of a molecule, with an implicit protein receptor, and outputting a pose, which then has an associated score. (A) Surrogate ML docking model: a model is trained to predict the score of the best pose. (B) a trained ML surrogate model is used to filter the molecular database for which traditional docking is used (surrogate with filtering). (C) Regular docking is performed to find correct poses, but a machine learning scoring function is used to re-score the poses, in this case, using experimental data to predict a binding free energy estimate (BFE). (D) A generative workflow where a generative model produces new compounds can be used in standard docking or with another surrogate model. Those scores are then used to try to optimize the model to produce higher scoring compounds.

4.3 Case Study: uHTVS for Discovery of a Novel SARS-CoV-2 3CL-Main Protease

The ongoing novel coronavirus pandemic (COVID-19) has resulted in over 200 million infections and more than 4 million deaths worldwide[1]. Although vaccines against the COVID-19 causative agent, the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), are being deployed [Knoll and Wonodi, 2021, Le et al., 2020], the discovery of drugs which can inhibit various SARS-CoV-2 proteins remains essential for treating patients [Cao, 2020, Wu et al., 2020]. Leveraging existing coronavirus treatments developed for severe acute respiratory syndrome (SARS) and middle eastern respiratory syndrome (MERS) [Yang et al.,

1. https://covid19.who.int

2006], as well as broad international collaborations, researchers have quickly determined structures for over 15 viral proteins, including inhibitor/lead bound structures and fragment-based screening for several non-structural proteins (NSP) such as the main protease (3C-like protease/M^{pro}), adenine diphosphate ribosyl-transferase (ADRP/NSP3), endoribonuclease (NSP15), and helicase (NSP13) [Lubin et al., 2020], all playing crucial roles in viral replication. Together, these collaborations have significantly accelerated the design and development of antiviral treatments targeting SARS-CoV-2 [Rosas-Lemus et al., 2020, Shyr et al., 2020].

Of these proteins M^{pro} is an attractive drug target mainly because it plays a critical role in viral replication and does not have any closely related homologs within the human genome [Hegyi and Ziebuhr, 2002, Pillaiyar et al., 2016]. Drug discovery efforts have resulted in discovering/re-purposing small molecules based on their ability to inhibit other coronavirus M^{pro} from middle east respiratory syndrome(MERS) and severe acute respiratory syndrome(SARS); however, it has been a challenge to identify non-covalent inhibitors for SARS-CoV-2 M^{pro} mainly due to the intrinsic flexibility of the primary binding site [Kneller et al., 2020c].

High throughput virtual screening (HTVS) is a common step of drug-discovery, enabling rapid, low-cost screening of significantly larger compound libraries than feasible in experimental studies [Zhang et al., 2008]. A number of efforts have focused on creating open HTVS infrastructure, taking advantage of cloud computing platforms or supercomputing resources to support large-scale ligand docking across various protein targets [Acharya et al., 2020]. These platforms have leveraged open-source toolkits such as AutoDock/AutoDock-VINA (for molecular docking) [Trott and Olson, 2010] in conjunction with molecular modeling (MM) and molecular dynamics (MD) simulation engines to capture 'modes' of interaction between a protein target and specific compounds from compound-libraries (e.g., ZINC [Irwin and Shoichet, 2005a], MCULE [Kiss et al., 2012]). Of these approaches, the COVID-Moonshot

project, using crowdsourced design strategies, high-throughput experimental screening, MD simulations and ML were able to identify both covalent and non-covalent inhibitors against M^{pro} which demonstrated viral inhibition *in vitro* [Achdout et al., 2020a].

We describe our discovery of a non-covalent small molecule inhibitor for M^{pro} using our HTVS platform that employs supercomputing resources, ensemble docking strategies, high-throughput experimental screening, X-ray crystallography, and MD. Complementary to efforts that scaled crowdsourcing approaches [Achdout et al., 2020b] as well as HTVS across potentially O(billion) compounds [Acharya et al., 2020, Gorgulla et al., 2021], we used a library consisting of 6.5 million in-stock compounds from the MCULE library [Kiss et al., 2012]. Ensemble docking was carried out across available crystal structures for M^{pro}, from which O(1,000) top consensus-scoring compounds across two popular docking programs (Autodock-VINA [Trott and Olson, 2010] and OpenEye FRED [McGann, 2012]) were experimentally characterized. From these compounds, we discovered one molecule that inhibits M^{pro} with K_i of 2.9 μM and determined its room-temperature X-ray crystallographic structure to 1.8 Å resolution. Finally, we used μs-timescale atomistic MD simulations to characterize the binding mechanism to the M^{pro} active site, while altering the enzyme's overall conformational dynamics. Our workflow provides a scalable framework for the rapid discovery of viable lead molecules against SARS-CoV-2.

4.3.1 Methods

Molecular library generation

We use a set of on demand compounds from Mcule (ORD), which can be freely obtained on their website [Kiss et al., 2012]. ORD consists of compounds from Mcule listed as available on their website, Mcule Purchasable (Known Stock Amounts).

Molecular Docking Protocol with OpenEye Toolkit

The main protease structures screened included the following PDB structures: 7BQY, 6LU7, 6W63, 7C7P, and 7JU7. The receptors for OpenEye Chemgauss4 scoring were prepared using the known binding region of M^{pro} with the OpenEye Docking Toolkit[McGann, 2012].

For memory efficiency, conformer generation and tautomerization were performed on-the-fly. OMEGA [Hawkins et al., 2010] was used, sampling around 300–500 conformations for each ligand. When ligand-binding information was available in the receptor, HYBRID was used due to its increased pose prediction accuracy over FRED [McGann, 2012]. HYBRID and FRED have the same scoring function; however, HYBRID uses a heuristic to reduce the search space for ligand positioning. The best score from the ensemble of tautomers and conformers is chosen as the representative "docking score" for the chemical species.

Computational Workflow

OpenEye toolkit's FRED docking program was deployed on Frontera at TACC. Docking scores for the M^{pro} receptor were computed with individual runs per pocket. The docking protocol as described above requires the following steps for each compound: (1) load receptor into memory; (2) load compound data stored in MCULE [Kiss et al., 2012] database from disk; (3) run the specific docking protocol over the receptor/compound pair; and (4) write the resulting docking score to persistent storage.

High-throughput docking was implemented using RADICAL-Pilot (RP) and RAPTOR [Merzky et al., 2021]. RP is a pilot-enabled runtime system while RAPTOR is a scalable master/worker overlay developed to improve the execution performance of many, short-running tasks encoded as Python functions. The runs used between 128 and 7000 concurrent nodes. For each run, we measured throughput (the number of docking calls per hour) and resource utilization (the fraction of time that acquired nodes were kept busy). Resource utilization was dependent on the size of the run (number of compounds to dock, number of nodes to

use), and was typically above 90%.

SARS-CoV-2 M^{pro} expression and purification

A gene construct encoding M^{pro} (NSP5) from SARS-CoV-2 was cloned into plasmid pD451-SR (Atum, Newark, CA), which was developed in [Kneller et al., 2020c], and expressed and purified consistent with the protocols detailed in [Kneller et al., 2020b]. Protein purification supplies were purchased from Cytiva (Piscataway, New Jersey, USA). Briefly, authentic N-terminus is achieved by a NSP4-NSP5 autoprocessing sequence (SAVLQ↓SGFRK where the arrow indicates the scissile bond) flanked by maltose binding protein and M^{pro}. Following M^{pro}, a sequence encoding the human rhinovirus 3C (HRV-3C) cleavage site (SGVTFQ↓GP) is followed by a His6-tag. The N-terminal sequence is created by autocleavage during expression while the C-terminus is generated by HRV-3C treatment following Ni immobilized metal affinity chromatography.

Primary M^{pro} inhibition screen

Compounds were purchased from Mcule, Inc. as 10 mM stock solutions in DMSO and stored at -20 . The assays were performed in 40 μL total volume in black half area 96-well plates (Greiner PN 675076) at 25 . The assay buffer contained 20 mM Tris-HCl pH 7.3, 100 mM NaCl, 1 mM EDTA, and 2 mM reduced glutathione (6.15 mg added per 10 mL buffer fresh for each experiment) with 5% v/v final DMSO concentration. M^{pro} initial rates were measured using a previously established fluorescence resonance energy transfer (FRET) peptide substrate assay[Kuo et al., 2004]. The FRET substrate DABCYL-KTSAVLQ↓SGFRKM-E(EDANS) trifluoroacetate salt was purchased from Bachem (PN 4045664), dissolved to 10 mM in DMSO and stored in aliquots at -20 °C. 10 μL enzyme solution was dispensed into wells (250 nM final concentration), followed by 10 μL inhibitor solution (20 μM final concentration), centrifuged briefly, and incubated for 30 min. Reactions

were initiated by adding 20 μL substrate at 40 μM final concentration. Fluorescence was detected every 24 s by a Biotek Synergy H1 plate reader with an excitation wavelength of 336 nm and an emission wavelength of 490 nm, 6.25 mm read height, low lamp energy, and 3 measurements per data point. After background subtraction of the average of no enzyme negative controls, product formation was quantified using a 0.05 – 22 μM calibration curve of the free EDANS acid (Sigma PN A6517). Product concentrations were adjusted for inner filter absorbance effects with correction factors generated by comparing the fluorescence of 2 μM EDANS in solution with each concentration of substrate used to that with no substrate. Initial rates were determined for time points in the linear range by linear regression in Excel, residual activities were determined by normalizing candidate initial rates to the average of the positive controls, and z-scores were determined by dividing the difference between the candidate initial rate and average positive control initial rate by the standard deviation of the positive control initial rates. The Z' statistic for the plate was calculated using the published equation.

Peptide synthesis

The unlabelled M^{pro} substrate peptide AVLQ↓SGFRKK-amide and the isotopically labelled substrate and product peptide internal standards (A+7)VLQSGFRKK-amide and (A+7)VLQ-OH were synthesized by automated peptide synthesis using a Liberty PRIME^{TM} peptide synthesizer (CEM). Reagents were peptide synthesis or biotechnology grade. Amino acids were purchased from P3Bio, and Fmoc-[$^{13}C_3$, ^{15}N, D_3]-alanine (A+7) was previously synthesized at Los Alamos National Laboratory following published protocols [Lodwig and Unkefer, 1996]. Other purchased reagents were dimethylformamide (DMF), pyrrole (prepared as 20% v/v in DMF), and high performance liquid chromatography (HPLC)-grade acetonitrile (Alfa Aesar), diisopropylcarbodiimide and Oxyma Pure (AKScientific), N,N-diisopropyl ethyl amine (DIPEA), triisopropyl silane (TIPS), trifluoroacetic acid (TFA),

thioanisole, Rink amide resin, and octaethylenglycol-dithiol (Sigma Aldrich), dichloromethane and Optima mass spectrometry grade acetonitrile (Fisher Scientific).

Peptide syntheses were performed at 0.1 mM scale under argon on a Rink amide resin with 0.1 M DIPEA added to the Oxyma solution to prevent hydrolysis of acid labile side chain protecting groups, obtaining average yields for double couple cycles of >99%. For stable isotope labeled peptides only two equivalents of the labeled amino acid were used and coupling time was extended to 20 min at 90 °C. Peptides were deprotected with the following mixture: 1.25 mL TIPS, 0.625 mL thioanisole, 1.25 mL octaethyleneglycodithiol, and after 5 min TFA was added to a total volume of 25 mL. Solutions were filtered and the filtrate concentrated to 10 mL, followed by precipitation with ice cold ether and collection by centrifugation.

Peptides were purified to >98% by Waters HPLC workstation (2545 pump with 2998 photodiode array detector) with a Waters BEH 130, 5 μm, 19x150 mm C18 column and a linear gradient from 98:2 to 50:50 water:acetonitrile with 0.1% TFA at 20 mL/min. Absorbance at 215 nm was monitored and peaks were collected and lyophilized to yield a white fluffy solid. Peptide purity was analyzed by analytical HPLC and Thermo LTQ mass spectrometry with electrospray ionization in positive mode with a Waters BEH 130, 5 μm, 4.6x150 mm C18 column and a linear gradient from 96:2 to 60:40 water:acetonitrile with 0.1% TFA at 1.5 mL/min over 12 min.

Quantitative mass spectrometry M^{pro} inhibition assay

The quantitative mass spectrometry (MS) inhibition assay was performed as described for the FRET-based primary screen with some modifications. Round-bottom polypropylene 96-well plates (Corning PN 3365) were used with 150 nM final M^{pro} concentration and the unlabeled peptide substrate synthesized above. Five min after substrate addition, the assay was quenched 1:1 v:v with 2% formic acid in water with 2 μM of each internal standard

peptide from above, centrifuged 10 min at 4 °C, and the supernatant was diluted 1:9 v:v into 1% formic acid. Substrate and product peptides and internal standards were quantified by high-throughput MS using a Sciex 5500 QTRAP with a custom open port sampling interface (OPSI) [Van Berkel and Kertesz, 2015]. Samples were introduced as 2 μL droplets and the OPSI-MS analysis was performed using 10:90:0.1 v:v:v water:methanol:formic acid at 80 μL/min. Positive ion mode electrospray ionization parameters were CUR: 25, IS: 5000, TEM: 400, GS1: 90, GS2: 60, EP: 10, and CXP: 10. Optimized multiple reaction monitoring detection parameters were dwell: 50 msec, product DP: 100 and CE: 25, and substrate DP: 150 and CE: 34. The following mass-to-charge transitions were monitored: substrate AVLQSGFRKK, 566.9→722.3; (A+7)VLQSGFRKK, 570.4→722.3; product AVLQ, 430.3→260.3, and (A+7)VLQ, 437.3→260.3. Product formation and remaining substrate were quantified by dividing the peak area of the transitions by that of the corresponding internal standard transitions.

IC_{50} and K_i Value Determination

To determine the concentration at which a compound was able to achieve 50% inhibition of M^{pro} activity *in vitro* (IC_{50}), the FRET and quantitative MS assays described above were performed at 10 concentrations of inhibitor (0.56-100 μM) in triplicate with 150 nM enzyme. Initial rates, for FRET, or product formation in 5 min, for MS, were normalized to no inhibitor control (100% activity) and no enzyme control (0% activity), and nonlinear regression of the [Inhibitor] vs. normalized response IC_{50} equation was performed to fit the data using GraphPad Prism 9.0.0, yielding IC_{50} and its 95% confidence interval. To confirm the mechanism of inhibition and determine K_i, the FRET activity assay was performed at 8 concentrations of substrate (20-500 μM) and 4 concentrations of inhibitor (0-25 μM) in triplicate in two independent experiments. A global nonlinear regression was performed to fit the competitive inhibition equation to the entire data set using GraphPad Prism 9.0,

yielding K_M, K_i, V_{max}, and their associated 95% confidence intervals.

Crystallization

Crystallization reagents were purchased from Hampton Research (Aliso Viejo, California, USA). Crystallographic tools were purchased from MiTeGen (Ithaca, New York, USA) and Vitrocom (Mountain Lakes, New Jersey, USA). M^{pro} was concentrated to ∼5.0 mg/mL in 20 mM Tris pH 8.0, 150 mM NaCl, 1 mM TCEP, for crystallization. The presence of reducing agent such as TCEP is essential for preventing oxidation of the catalytic cysteine sidechain [Kneller et al., 2020d]. Conditions for growing crystalline aggregates of ligand-free M^{pro} were identified by high-throughput screen at the Hauptman-Woodward Research Institute [Luft et al., 2003] and reproduced locally using 22% PEG3350, 0.1 M Bis-Tris pH 6.5 in 20μL drops with 1:1 ratio of the protein:well solution using sitting-drop vapor diffusion with microbridges. Crystal aggregates of ligand-free sample were converted to microseeds with Hampton Research Seed Beads^{TM} and used for nucleating M^{pro} crystals in subsequent co-crystallization experiments. Lyophilized MCULE-5948770040 for co-crystallization was dissolved in 100% DMSO as a 50 mM stock stored at -20°C. MCULE-5948770040 was mixed with M^{pro} at 5:1 M ratio and allowed to incubate on ice for a minimum one hour. Crystals were grown in a 40 μL drop at a 1:1 mixture with 18% PEG3350, 0.1 M Bis-Tris pH 7.0 with 0.2 μL of 1:200 dilution microseeds and incubated at 14°C. A large crystal measuring $\sim 1 \times 0.5 \times 0.3$ mm suitable for room-temperature X-ray diffraction grew after 2 weeks.

Room-temperature X-ray data collection and structure refinement

The protein crystal was mounted using a MiTeGen (Ithaca, NY) room-temperature capillary system. X-rays for crystallography were generated from a Rigaku HighFlux HomeLab employing a MicroMax-007 HF X-ray generator and Osmic VariMax optics allowing diffraction images to be collected using an Eiger R 4M hybrid photon counting detector. Diffraction data

was reduced and scaled using Rigaku CrysAlis Pro software package. Molecular replacement was performed using the ligand-free room-temperature M^{pro} structure (PDB code 6WQF) [Kneller et al., 2020c] using Molrep [Winn et al., 2011]. Structure refinement was performed with Phenix.refine from Phenix suite [Adams et al., 2010] and COOT [Emsley and Cowtan, 2004] for manual refinement and Molprobity [Chen et al., 2010]. Data collection and refinement statistics are listed in Table S1. The structure and corresponding structure factors of the room temperature Mpro/MCULE-5948770040 complex have been deposited into the Protein Data Bank with the PDB accession code 7TLJ.

Molecular dynamics simulations of M^{pro} complex with MCULE-5948770040

The crystal structure of protein dimer was modeled with the AMBER molecular modeling package [Götz et al., 2012] with the amber.ff14sb force field parameters (for the protein) [Maier et al., 2015a] and with the GAFF parameters (for the ligand) [Wang et al., 2004a]. In order to better determine the partial charges for the ligand, quantum mechanical (QM) calculations were performed using NWChem [Valiev et al., 2010] based on the RESP method at B3LYP/6-31G* level of theory [Kozlowski, 2001], while all bonded parameters were taken from GAFF force field.

The systems (both the ligand-bound/LB and ligand-free/LF) were solvated using the TIP3P water model and counter ions were added to neutralize the charge. After equilibrating the systems using previously published protocols [Ramanathan et al., 2020], we carried out production runs using the OpenMM [Eastman et al., 2017a] simulation package on Nvidia V100 GPUs using the Argonne Leadership Computing Facility's (ALCF) computing clusters. Each time step was integrated with Langevin integrator at 310 K, 1 ps^{-1} friction coefficient, and 2 fs interval with fixed lengths being maintained for atomic bonds involving hydrogen atoms. System pressure was maintained at 1 atm with the MonteCarloBarostat. Nonbonded interactions were cut off at 1.0 nm and Particle Mesh Ewald (PME) was implemented for

long-range interaction. The simulations were run for 1 μs and 50 ps reporting interval (4 replicas).

Quantifying conformational transitions in the ligand-bound and ligand-free states of MPro with Anharmonic Conformational Analysis driven Autoencoders (ANCA-AE)

Conformational fluctuations within bio-molecular simulations (and specifically proteins) show significant higher order moments; these fluctuations may relate to protein function [Ramanathan et al., 2020, 2021]. To quantify such anharmonic fluctuations within our simulations, we used fourth-order statistics to describe atomistic fluctuations and to characterize the internal motions using a small number of anharmonic modes [Parvatikar et al., 2018]. We projected the original data (306 C^{α} atoms per chain - (x, y, z) coordinates) onto a 40 (or 50) dimensional space, depending on the set of simulations considered. Notably, for the ligand bound states (in both protomers), 40 dimensions covered about 95% of overall variance whereas we required 50 dimensions to cover 95% of the variance when we included the ligand bound states from just one protomer.

Given the significant non-linearity in the atomic fluctuations, we used an autoencoder to further delineate the intrinsic structure in the low-dimensional anharmonic space. Similar to approaches that use variational approximations to model molecular kinetics from MD simulations [Mardt et al., 2018], we used an autoencoder architecture consisting of a symmetric encoder and decoder network. The network is composed of a single dense layer with 32 dimensions, and an 8 dimensional latent space. We trained the network for 50 epochs using the RMSprop optimizer to minimize the mean-squared error (MSE) reconstruction loss with a learning rate of 0.001, weight decay of 0.00001, and a batch size of 64. We used ReLU activation in all places except the final reconstruction layer, where we used Tanh activation. A mixture of Gaussian (MoG) model was used to cluster the conformations in the low

dimensional landscape, similar to the approach outlined in [Ramanathan et al., 2011].

4.3.2 Results

HTVS of M^{pro} with on-demand molecular libraries

A docking screen against the main protease of SARS-CoV2, M^{pro}, was performed on an orderable on-demand compound library from Mcule [Kiss et al., 2012]. Given the intrinsic flexibility of the M^{pro}'s primary binding pocket consisting of the four conserved binding sites (S1', S1, S2 and S4) [Xue et al., 2008, Kneller et al., 2020c], we used five different crystal structures were used for an ensemble docking approach using PDB identifiers 6LU7[Jin et al., 2020], 6W63 [Mesecar, 2020], 7BQY [Jin et al., 2020], 7C7P [Qiao et al., 2020], and 7JU7 [Tan et al.]. In addition to the structural ensemble, we used the docking protocols and scoring functions from the OpenEye Scientific FRED [McGann, 2012] toolkit. In total, over 63 million docking scores were computed over the five structures, two compound libraries, and two protocols. The overall workflow is summarized in Fig. 4.4a. The workflows were deployed on HPC resources at the Argonne and Oak Ridge leadership computing facilities (ALCF/OLCF) using Theta and Summit supercomputers and using the Texas Advanced Computing Center (TACC; Frontera) and San Diego Supercomputing Center (SDSC; Comet). The resulting docking libraries (including scripts of preparation and docking) and the docking scores are available as a downloadable dataset [Clyde et al., 2021e]. The details of the computational performance and workflow optimization are described in the Supporting Information (SI) text (sections S1 and S2) and Fig. S1(a)-(c).

The resulting compounds were ranked based on the docking scores in conjunction with visual inspection and availability at the time, and selected compounds were ordered for experimental validation studies. Interestingly, docking score distributions across each of the structures were slightly different (summarized in Fig. 4.4b-c), and we therefore examined the top 0.1% of the overall distributions. Between receptors' respective docking, the highest

correlation coefficient is 0.85 (7BQY and 6LU7) and the lowest was 0.001 (6W63 and 7JU7; See SI Table S1). In fact, 6W63's docking result is an outlier with respect to the other four receptors, with the highest correlation coefficient of only 0.003. Given the variation amongst docking results between receptors, a consensus score was deemed necessary. A consensus score was created by taking the minimum over the available series. A minimum was chosen rather than an average, or other aggregation techniques, due to the nature of our docking protocol. OpenEye FRED has a wide range of scores, unbounded above or below. A small steric difference between receptors can cause a wide numerical discrepancy or even lack of a result. We see in Fig. 4.4(d) a significant difference between the correlation of consensus scores over using single samples (Table S1).

MCULE-5948770040 is a SARS-CoV-2 M^{pro} inhibitor

Based on the consensus scoring procedure above, 116 compounds from the Mcule database were selected for experimental screening using the top 20 from different M^{pro} crystal structures. Of these 116 compounds, five were not available for ordering, 15 were excluded due to pan assay interference compounds (PAINS) violations based on the substructure filters of Baell and Holloway [Baell and Holloway, 2010], and 72 were ultimately delivered. These compounds were subjected to a primary SARS-CoV-2 M^{pro} activity inhibition screen in which they were pre-incubated with the enzyme and the initial velocities of cleavage of a fluorescence resonance energy transfer (FRET) peptide were determined [Kuo et al., 2004]. The Z'-factor of the assay was 0.65, and the distribution of z-scores of compounds and positive (no inhibitor) and negative (no enzyme) controls is shown in Fig. 4.5A. At least 25% inhibition was observed for seven compounds, with MCULE-5948770040 resulting in the lowest residual activity at 20 μM (12%).

Mechanism of inhibition The concentration-dependence of MCULE-5948770040 *in-vitro* M^{pro} inhibition was measured at 40 μM substrate, giving an IC_{50} of 4.2 μM [95% confidence interval 3.8, 4.7] (Fig. 4.5B). An orthogonal quantitative high-throughput mass spectrometry-based endpoint assay was also performed at 40 μM unlabelled peptide substrate, giving a similar IC_{50} of 2.6 μM [95% CI 2.3, 2.9]. Initial rates measured at 20-500 μM substrate and 0-25 μM inhibitor were consistent with a competitive mechanism of inhibition with a K_i of 2.9 μM.

Room-temperature X-ray crystal structure of M^{pro} in complex with MCULE-5948770040

To elucidate the molecular basis of M^{pro} inhibition by the MCULE-5948770040 compound, a X-ray crystal structure of M^{pro} in complex with the compound was determined to 1.80 Å at near-physiological (room) temperature. The M^{pro}/MCULE-5948770040 complex crystallized as the biologically relevant homodimer with the protomers related by a two-fold crystallographic axis (Fig. 4.6). The tertiary fold is shown in Fig. 4.6. Each protomer consists of three domains (I-III). The substrate-binding cleft is formed at the interface of catalytic domains I (residues 8-101) and II (residues 102-184), whereas the α-helical domain III (residues 201-303) creates a dimerization interface [Zhang et al., 2020, Jin et al., 2020]. The substrate binding cleft lies on the surface of the enzyme and accommodates amino acid residues or inhibitor groups at positions P1'-P5 in subsites S1-S5, respectively [Sacco et al., 2020, Hoffman et al., 2020, Kneller et al., 2020a]. Subsites S1, S2, and S4 have well defined shapes while S1, S3, and S5 are surface-facing with poorly defined edges [Kneller et al., 2020e]. The non-canonical catalytic dyad composed of Cys145 and His41 lies deep within the substrate binding cleft poised for peptide bond cleavage between the C-terminal P1 and N-terminal P1 positions.

MCULE-5948770040 binds non-covalently to the active site of M^{pro}, occupying subsites

S1 and S2. The electron density for the inhibitor is unambiguous (Fig. 4.6a) enabling accurate determination of the protein-ligand interactions (Fig. 4.6b). The uracil P1 group of the ligand is situated in the S1 subsite driven by polar contacts. Nϵ2 of the His163 imidazole side chain makes a close 2.6Å H-bond with the carbonyl at position 4 of the uracil substituent. Notably, His163 was previously determined to be singly protonated on the Nδ1 by neutron crystallography [Kneller et al., 2020e], suggesting a possible rearrangement of the protonation state for His163 side chain upon ligand binding. The far end of the S1 subsite is formed such that the second protomer's N-terminal protonated amine creates H-bonds with the Glu166 sidechain, Phe140 main chain carbonyl, and a water molecule. The amide NH at position 3 of the ligand's P1 heterocycle is situated within hydrogen bonding distance with Glu166 and Phe140 although the geometry is unfavorable. The carbonyl and amide NH at positions 2 and 1 participate in water-mediated H-bonds with Ser1 and Asn142, respectively. M^{pro} features an oxyanion hole created by the main chain amide NH groups of Gly143, Ser144, and Cys145 at the S1 subsite base. A carbonyl linking the P1 uracil to the central piperazine linker of MCULE-5948770040 is positioned on the perimeter of the oxyanion hole forming a direct 2.8Å H-bond with Gly143 and a water-mediated contact with Cys145. Piperazine is located above the catalytic Cys145 side chain that was determined to be a deprotonated, negatively charged thiolate in the neutron structure of the ligand-free M^{pro}. The P2 dichlorobenzene substituent occupies the largely hydrophobic S2 subsite.

An overlay of the MCULE-5948770040 complex with the ligand-free joint neutron/X-ray crystal structure of M^{pro} [Kneller et al., 2020e] shows that the P2-dichlorobenzene group moves into the hydrophobic S2 pocket altering the position of the Met49 side chain and pushing out the P2-helix (residues 46-50) by as much as $\sim$2.6 Å (Fig. 4.6c). The Met49 terminal methyl is shifted $\sim$5.5 Å away and its Cα atom moves by $\sim$1.1 Å. Furthermore, the position of the P2-dichlorobenzene group is stabilized by the ß-ß stacking interactions with Gln189 and the imidazole side chain of catalytic His41 with the interatomic distances

of ∼3.8 Å. The Gln189 side chain amide is recruited from 3 Å away from its position in the ligand-free structure and the Cα atom shifts by almost 1 Å. Thus, the P2-dichlorobenzene is sandwiched between the side chains of these two residues. Interestingly, the binding of MCULE-5948770040 to the M^{pro} active site cleft causes the His41 side chain to flip and a χ2 angle rotation to create a favorable geometry for ß-ß stacking with P2-dichlorobenzene in the complex structure. Such change in the His41 conformation severs a conserved H-bond between the His41 Nδ1 and the conserved catalytic water molecule (H_2O_{cat}) normally seen in M^{pro} structures [Vuong et al., 2020, Rathnayake et al., 2020], but replaces it with a direct H-bond to the main chain carbonyl of His164 and results in recruitment of an additional water molecule from the bulk solvent to make an H bond with the His imidazole ring.

The computationally predicted binding pose of MCULE-5948770040 to the M^{pro} active site is in good agreement with the experimentally determined orientation (Fig. 4.6d). Only minor discrepancies in the piperazine linker and P2-dichlorobenzyl are present. The ligand's uracil group forms the same polar interactions in the docked pose as observed in the crystal structure. The piperazine linker is best modeled as a chair conformation in the crystal structure while the docked geometry scored highest with it adopting a twisted boat. P2-dichlorobenzene fits well into the S2 pocket as observed in the experimental configuration, despite a 180° rotation of the aromatic ring.

M^{pro}interacts with MCULE-5948770040 through conformational changes within the binding site

In order to understand how the molecule interacts with the M^{pro} binding site, we carried out μs-timescale atomistic molecular dynamics (MD) simulations (see Methods). For the ligand-bound (LB) state simulations, we modeled the ligand present in both M^{pro} protomers. Within the timescales of our simulations (μs timescales), we observed that the ligand stays bound to the primary binding site (for both chains A and B in the dimer). The protein also

does not undergo any large conformational changes (as seen from the root-mean squared deviations/ RMSD from the starting structure SI Fig. S4-S6).

We used the root mean squared fluctuations (RMSF) from ligand-free (LF) and ligand-bound (LB) M^{pro} simulations (Fig. 4.7A) to understand how the ligand impacts the conformational dynamics. For convenience, we considered each protomer individually (although the simulations were run with the ligand bound to both monomers in the active dimer form) and observed that several distinct regions across M^{pro} exhibit altered fluctuations. In all of our replicas, the RMSF in chain A of the dimer were slightly higher than in chain B. Distinct regions within M^{pro} respond to the ligand (rounded rectangles in Fig. 4.7A); these mostly consist of flexible loops surrounding the immediate vicinity of the binding pocket, corresponding to the sites (S1 - orange rounded rectangle and S2 - red rounded rectangle). Other regions surrounding the binding site (S3 – green and S4 – blue rounded rectangles respectively) also exhibit stabilization upon ligand binding. However, it is notable that not all regions exhibit stabilization within each protomer (e.g., region S4, the protomer chain A exhibits similar fluctuations to the ligand-free state). Interestingly, regions farther away from the binding site, including domain III of each protomer (R5 in Fig.4.7 (purple rounded rectangle) exhibit lower fluctuations in the LB simulations.

To elucidate the collective motions that are influenced by ligand binding, we used anharmonic conformational analysis enabled autoencoder (ANCA-AE; see Methods) to embed the conformational landscape spanned by the ligand-free and ligand-bound simulations in a low-dimensional manifold (summarized in Fig. 4.7b). Notably, our simulations can be embedded within an ten dimensional manifold (Fig. S7) which best explains conformational fluctuations undergone by the protein. Of these embeddings, the LF and LB simulations occupied distinct projections, with the LF-simulations sampling diverse conformational states (as quantified by the RMSD to the LB states). The predominant conformational changes in the LB simulations were confined to the binding pocket spanning S1-S4 and R5, shown

in Fig. 4.7c-d in protomer A, while we did not observe significant motions with respect to protomer B (shown in inset). The fluctuations observed were mostly a consequence of re-orienting the ligand within protomer A from its primary interaction site (P1,P2) to cover the complementary binding site of (P2,P4; orange rounded rectangle in Fig. 4.7c, labeled I $\rightarrow$Holo$_B$). Notably the P1-uracil forms new interactions with the S4 region (residues) while the P2-dichlorobenzene stays bound within the hydrophobic pocket. Although we do not observe such a ligand placement within the crystal structure, these ligand motions are prevalent across multiple replicas of our simulations and can form stable interactions between the uracil and protein side-chains. Corresponding to these changes, motions in domain III of the protein (purple cartoon insets in Fig. 4.7 c-d) are also suppressed, showing that this region may be stabilized upon ligand binding. We also examined the hydrogen bonding patterns between the ligand and protein from the LB-simulations (SI Fig. S8) and found that the hydrogen bonds between the ligand and the protein in protomer chain B are more stable than the hydrogen bonds in chain A.

4.3.3 Discussion

Our study demonstrates a 2.9μM potent M$^{\text{pro}}$ inhibitor discovered through HTVS. After computationally screening 6 million molecules, we located 100 promising compounds based on a consensus score across five different M$^{\text{pro}}$ crystal structures. Through X-ray crystallographic studies, we observed that the compound MCULE-5948770040 forms stable interactions within a hydrophobic pocket (S2) formed by the P2-dichlorobenzene group along with the P1-uracil group occupying the S1 site. Assay results also indicate that this molecule is a μM non-covalent inhibitor of this enzyme and can act as a competitive inhibitor. Our μs timescale simulations indicate fairly stable interactions between the protein and the ligand, which suggest that several regions of M$^{\text{pro}}$ – both in the vicinity of the binding site and distal from it – are impacted upon binding. This alters the conformational states accessed

by the protein as quantified by our ANCA-AE approach.

The combination of experimental validation with computational tools is essential to the rapid development of an inhibitor for M^{pro}. Without the ability to quickly obtain a compound and experimentally validate it, computational results alone will not be a solution. Virtual screens against M^{pro} have ranged from small libraries of existing approved pharmaceuticals to natural product libraries and billion-scale combinatorial libraries [Abo-Zeid et al., 2020, Gorgulla et al., 2021]. Studies such as [Marinho et al., 2020] use drug repurposing databases that are smaller (<50k) but have potential for faster lead to drug time. In this study, we aimed to balance library size and feasibility of validation, hence we opted for a 6M in-stock chemical library from Mcule. While [Lyu et al., 2019] used a approximately 120 million compound library to obtain hits for D4 dopamine receptor, we were able to use our approximately 6 million compound library without any advanced filtering post consensus scoring to locate a μM hit for M^{pro}.

Although a number of sub-μM and some nanomolar inhibitors are now available [Zhang et al., 2021b,a, Deshmukh et al., 2021, Morris et al., 2021, Rizzuti et al., 2021], our paper focused on the discovery of novel molecules from in stock molecules that could potentially inhibit M^{pro} activity. Further, several molecules from the COVID-moonshot project [Morris et al., 2021] also possess similar scaffolds (like the piperazine linker, or uracil) to MCULE-5948770040, which provides an indirect validation of how these fragments may be important for discovering additional inhibitors based on this molecule. Finally, we note that we have not tested this molecule for antiviral activity, which we plan to do as part of future work.

The collective conformational motions elucidated using ANCA-AE suggest an intrinsic asymmetry in how the ligand interacts with the two protomers. Our simulations point to a mechanism of complementary interactions and inter-domain motions whereby the ligand stabilizes the conformations of the loops around the binding site as well as a loop within domain III that is considerably far away from either binding sites. In order to elucidate if the

ligand binding affects either protomer separately, we also carried out simulations where the ligand was bound to only one of the two protomer units. Our analysis (SI text and Fig. S10-S11) of these simulations further indicate that the fluctuations in domain III of the protein are only affected from the ligand-bound chain. Taken together, our simulations suggest that the primary mechanism by which MCULE-5948770040 binds to and interacts with M^{pro} is by stabilizing the loops in and around the binding site. The binding of the ligand is asymmetric in the protomers; while it is stable in one of the protomers, it undergoes a slight conformational change (albeit stable) within the binding pocket while still maintaining the strong hydrophobic interactions within P2. Further, our analysis indicates that the hydrogen bonding patterns are different for the two chains, which also lends support to the idea that the collective motions as induced by the LB-states may indeed be different.

Computational protocols for HTVS vary widely across existing SARS-CoV-2 M^{pro} screens, from accurate but expensive simulations for MMGBSA/PBSA scoring to standard docking. Gorgulla et al. [Gorgulla et al., 2021] screened Enamine Real, a billion-scale combinatorial product library, against various SARS-CoV-2 targets using QuickVina W—a slightly less accurate but computationally efficient flavor of AutoDock Vina [Hassan et al., 2017]. Acharya et al. [Acharya et al., 2020] also screened Enamine Real with Autodock-GPU. In comparison one of the few other studies which involve assay and crystallographic experimental studies for M^{pro} lead generation [Achdout et al., 2020a], our work relies on a single computational workflow rather than community lead sourcing (where the methods of each contributor are not restricted). Rather than taking community input for prioritizing experimental leads, our work features an HTVS protocol based on ensemble docking with consensus scoring. Using the web portal to access hits from [Achdout et al., 2020a], we compared the difference between leads from domain experts with the library we used for screening and found that both groups arrived at structurally similar hits independently, with the same P1-Liner-P2 topology and interaction with M^{pro} S1 and S2 sites (SI Fig. S12). Between the two groups,

our best compounds share piperazine and the uracil groups.

The compound MCULE-5948770040 forms stable interactions with both the protomers as we observed from our X-ray crystallography and MD simulations. Our simulations provide insights into how the conformational fluctutations of the protein are altered in response to the ligand binding to the primary site. In fact, we observed that the fluctuations in region R5, which is over 20 away from the primary binding site in M^{pro}, are effected by the ligand binding. Further, the ligand's interaction with M^{pro} alters the conformational states accessed by the enzyme, notably along the substrate binding loops. Compared to other ligands that have been structurally characterized (as well as the substrate peptide), MCULE-5948770040 is much smaller and interacts stably with both the S1 and S2 sites within M^{pro}. Thus, a design strategy that targets the S1-S2 sites and mimics important features of the peptide side chains is sufficient for identifying inhibitors of M^{pro}. We have early indications that molecules structurally similar to MCULE-5948770040 also demonstrate inhibition, and efforts are currently underway to identify promising candidates, which we expect to publish shortly.

4.4 Conclusion

In this chapter, we outline how computational virtual screening problems can be turned into a deep learning problem. Given this notion of a surrogate model for docking, we provide a taxonomy of workflows for virtual ligand screening. Finally, we illustrate the power of throughput docking through a case study discovering and validating a novel SARS-CoV-2 inhibitor.

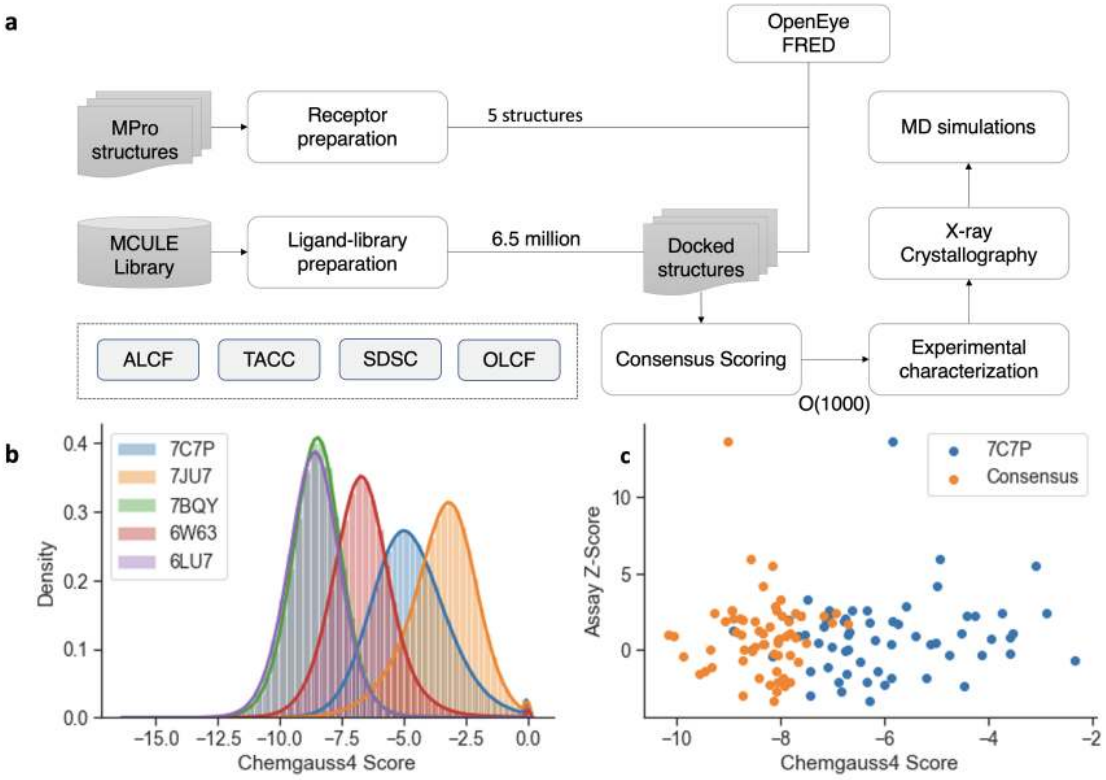


Figure 4.4: (A) Computational workflow used for screening on-demand chemical libraries against SARS-CoV2 M^{pro} with computational docking techniques. Four major supercomputing centers were utilized, namely Argonne Leadership Computing Facility (ALCF), Texas Advanced Computing Center (TACC), San Diego Supercomputing Center (SDSC), and Oak Ridge Leadership Computing Facility (OLCF). (B) The distribution of Chemgauss4 scores, from docking, from the docking a 6 million in-stock compound library. (C) The consensus scoring used shifted possible hits (higher Z-score is better) towards better scoring regions over just a single score from a single structure (7C7P is used for illustration). A lower consensus score implies a higher likelihood from the docking programs that the candidate compound will bind to the receptor.

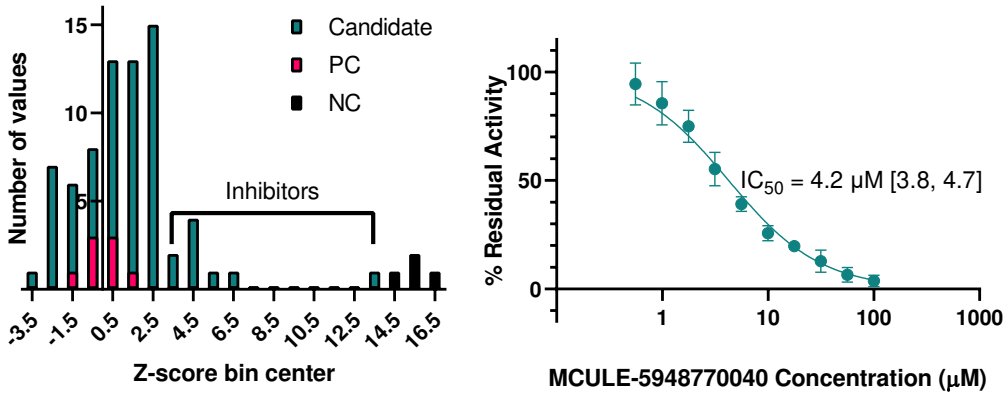


Figure 4.5: **Plate-based M^{pro} activity inhibition screening and hit confirmation.** (A) Histogram of z-scores of candidate inhibitors, no enzyme negative controls (NC), and no inhibitor positive controls (PC). (B) Inhibition of M^{pro} activity *in vitro* with increasing concentration of MCULE-5948770040. Initial rates are normalized to no inhibitor control (100% activity) and no enzyme control (0% activity). Error bars are standard deviation of two independent experiments, each performed in triplicate. Lines indicate the nonlinear regression of the [Inhibitor] vs. normalized response IC_{50} equation to the data with GraphPad Prism. Bracketed values indicate 95% confidence intervals from the regression.

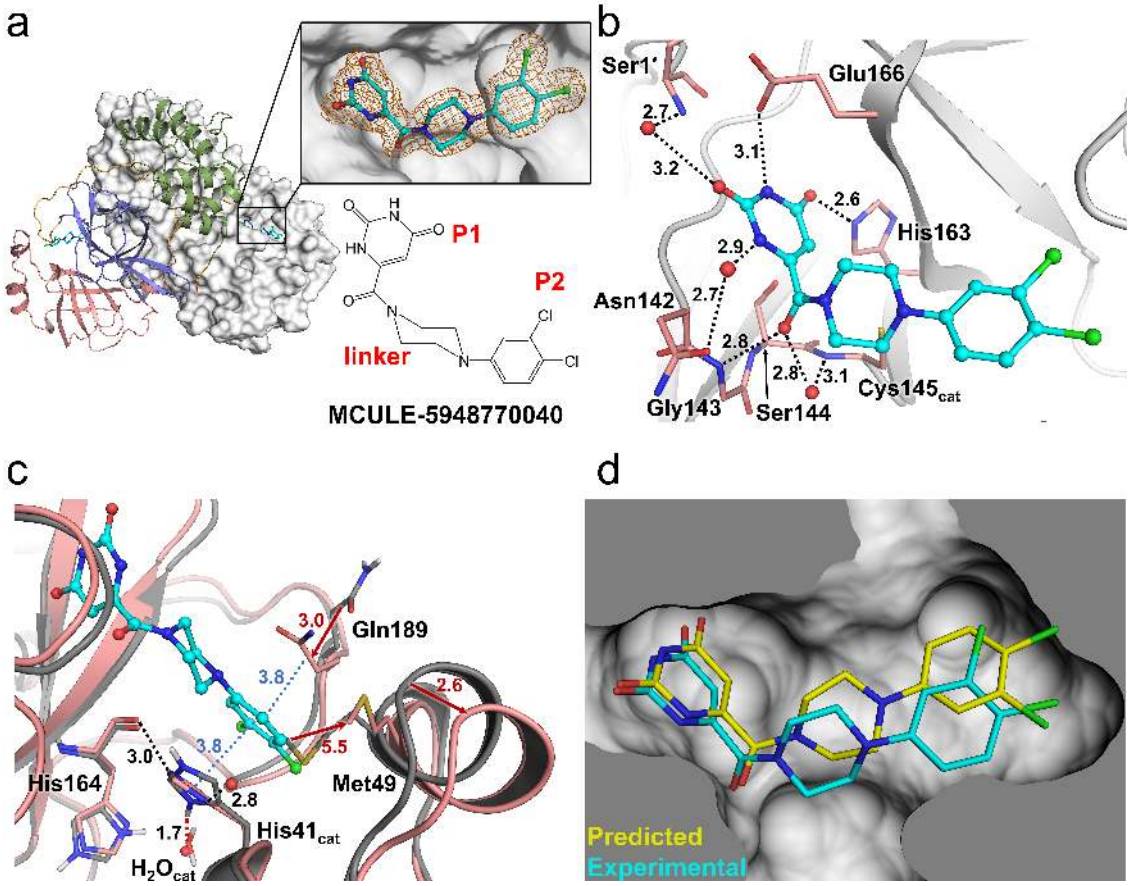


Figure 4.6: **Room-temperature X-ray crystal structure of M^{pro} in complex with MCULE-5948770040 and comparison with ligand-free and docked structures.** A) Overall M^{pro} homodimer in complex with MCULE-5948770040 (Cyan carbon ball and stick representation). One protomer is shown as a cartoon representation with domains I, II, and III in pink, purple, and green respectively and orange interdomain loops. The other protomer is shown as white surface. Insets show MCULE-5948770040 electron density (2Fo-Fc at 1.2σ as orange mesh) and 2D chemical diagram. B) Intermolecular interactions between M^{pro} (grey cartoon with salmon sticks) and the ligand. H-bonds are shown as black dashes. Distances in Å. C) Superposition of the M^{pro}/9MCULE-5948770040 complex (salmon) with ligand-free X-ray/neutron structure (grey, PDB code 7JUN). Red arrows indicate conformational shifts from ligand-free structure to complex structure. Blue dots show $\pi - \pi$ interactions with the P2-dichlorobenzene group. Red dashes represent a lost H-bond due to catalytic His41's imidazole side chain flip. D) Comparison of computationally predicted (yellow carbons) and experimentally determined (cyan carbons) pose of MCULE-5948770040 bound to M^{pro}.

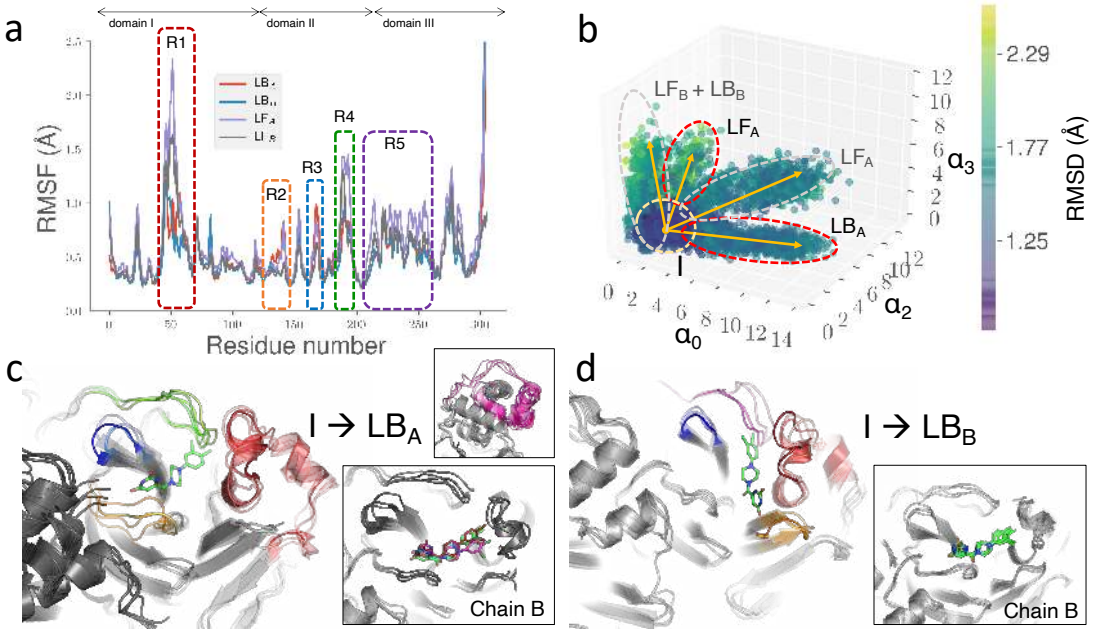


Figure 4.7: **Conformational changes upon MCULE-5948770040 binding to M^{pro} indicate changes within distinct regions, both close-to and farther-away from the primary binding site.** (a) RMS fluctuations of the LF- and LB-state of M^{pro} show several regions with decreased fluctuations that are highlighted within rounded rectangles. Although several regions within these regions are largely similar, amino-acid residues interacting with the ligand stabilize the binding site. (b) To further quantify the nature of these fluctuations, we characterized the collective motions which shows distinct conformational states sampled by the ligand-free (LF) and ligand-bound (LB) states. The yellow arrows indicate conformational transitions from the average structure towards the distinct conformational states (I, LF$_A$, LF$_B$, LB$_A$ and LB$_B$). These transitions are mapped in (c) I $\rightarrow$LB$_A$ and (d) I $\rightarrow$LB$_B$. (We show the I $\rightarrow$LF$_A$ and I $\rightarrow$LF$_B$). In each case, we observed that M^{pro} chain B of the dimer was more stable the chain A (insets). Regions highlighted in (a) show the motions undergone by the different regions of M^{pro}.

CHAPTER 5

SURROGATE MODELS FOR ACCELERATED DOCKING

This chapter will focus on my particular modeling efforts in building surrogate models for docking. This chapter will focus on the performance characteristics of docking with a focus on the COVID-19 inhibitor discovery [Clyde et al., 2021b, Wu et al., 2021]. *We will outline the datasets required, the models developed, the scale of their deployment, scaling characteristics, and error characteristics.*

As part of our drug discovery campaign for SARS-CoV-2 [Clyde et al., 2021d], we developed a database of docked protein-ligand across 15 protein targets and 12M compounds as well as the complexes' associated scores. The data preparation is outlined in the prior work. In brief, ligands were prepared using OpenEye Scientific OMEGA toolkit where 300-900 conformations were sampled for each ligand [OEChem, 2012a]. Receptors were prepared using the OEDOCK application. If the active site was unknown at the time, FPocket was used and the three highest scoring binding sites were used as an ensemble [Le Guilloux et al., 2009].

The database contains two related tasks. The first task is predicting a ligand's docking score to a receptor based on 2D structural information from the ligand. The second possible task is a pan-receptor model that encodes the protein target to use a single model across different ligands and targets. These tasks are distinct from other drug discovery datasets as this benchmark is focused directly on surrogate model performance over the baseline computational drug discovery method of docking. A different approach to applying machine learning to docking is the use ML models as a scoring function rather than the result of the optimization of the ligand conformation/position relative to the scoring function [Ragoza et al., 2017]. Other benchmarks are available to address to the gap between docking, and experimental binding free energy calculations such as DUD-E [Mysinger et al., 2012a].

The dataset we are releasing has three modes of representation, sequential, 2D or 3D, where the 3D data is a ligand conformation in an SDF file. 2D ligand data is available in a CSV file containing the molecule's purchasable name, a SMILES string, and its associated docking score in a particular complex.

The sequential dataframe includes maccs-keyDurant et al. [2002], ecfp2 Rogers and Hahn [2010b] , ecfp4, ecfp6 fingerprints, and descriptors. The models discussed in the rest of the main paper pertain to the 2D ligand structures (the associated 3D data is shared with the community for further developing 3D modeling techniques [Jiménez et al., 2018]).

The ligands available for each dataset are sorted into three categories ORD (orderable compounds from Mcule [Kiss et al., 2012]), ORZ (orderable compounds from Zinc [Sterling and Irwin, 2015]), and an aggregate collection which contains all the available compounds plus others (Drug Bank [Wishart et al., 2018], and Enamine Hit Locator Library [EDB]). Docking failures were treated as omissions in the data, which may be important consideration though typically, the number of omissions accounts for 1-2% at most of each sample.

The data is available here, `https://doi.org/10.26311/BFKY-EX6P`, and more information regarding persistence and usage is available on the data website [Clyde et al., 2021a].

5.1 Method

At a high level, surrogate models for protein-ligand docking aim to accelerate virtual ligand screening campaigns. A surrogate model seeks to replace the CPU-bound docking program with a trained model. In this case, surrogate models alone are not a viable solution to protein-ligand docking in general. ML surrogate models are based on gaussian statistics and generally perform well on predicting the central tendency of data, but not so at picking out the finer top or bottom 1%. We propose utilizing the ML to filter incoming ligands utilizing SPFD. Thus, the number of actual docking calculations is minimized compared to

the typical approach of docking the entire dataset. Due to model accuracy, the number of missed compounds is minimized as the fine-grained selection of a hit set comes from traditional docking and the model only needs to select a coarse set of hits rather than a fine set. In other words, a surrogate model is trained, and a cut-off is specific, say 1%. The model is run over the proposed library to screen, and the top 1% of ligands are then docked utilizing the program to have the exact scores and pose information as with typical docking. In this way, we do not see current surrogate models as a replacement for docking but rather as a mean of expanding their use over large virtual libraries. This model has a single hyperparameter, σ, which determines after running the surrogate model over the library which percentage of most promising predicted compounds we then dock utilizing traditional docking techniques.

5.1.1 Docking Pipeline

The training and testing datasets for these experiments were generated using 31 protein receptors, covering 9 diverse SARS-CoV-2 viral target protein conformations, that target (1) 3CLPro (main protease, part of the non-structural protein/ NSP-3), (2) papain like protease (PLPro), (3) SARS macrodomain (also referred to as ADP-ribosyltransferase, ADRP), (4) helicase (NSP13), (5) NSP15 (endoribonuclease), (6) RNA dependent RNA polymerase (RDRP, NSP7-8-12 complex), and (7) methyltransferase (NSP10-16 complex). For each of these protein targets, we identified a diverse set of binding sites along the protein interfaces using two strategies: for proteins that had already available structures with bound ligands, we utilized the X-ray crystallographic data to identify where ligand densities are found and defined a pocket bound by a rectangular box surrounding that area; and for proteins that did not have ligands bound to them, we used the FPocket toolkit that allowed us to define a variety of potential binding regions (including protein interfaces) around which we could define a rectangular box. This process allowed us to expand

the potential binding sites to include over 90 unique regions for these target proteins. We use the term target to refer to one binding site. The protocol code can be found here: `https://github.com/2019-ncovgroup/HTDockingDataInstructions`.

Two ligand libraries were prepared. The first was the orderable subset of the Zinc15 database (we refer to this as OZD) and the second was the orderable subset of the MCULE compound database (we refer to this as ORD). The generation of the orderable subsets was primarily a manual activity that involved finding all compounds that are either in stock or available to ship in three weeks across a range of suppliers. Consistent SMILE strings and drug descriptors for the orderable subsets of the Zinc15 and MCULE compound databases were generated as described by Babuji et al. [2020b]. Drug descriptors for the Zinc15 and MCULE compound databases can be downloaded from the nCOV Group Data Repository at `https://2019-ncovgroup.github.io`.

5.1.2 Data frame construction

We used the protein-ligand docking results between the prepared receptors and compounds in the OZD library to build machine learning (ML) data-frames for each binding site. The raw docking scores (the minimum Chemgauss4 score over the ensemble of conformers in a ligand-receptor docking simulation) were processed [OEChem, 2012a]. Because we were interested in determining strong binding molecules (low scores), we clipped all positive values to zero. Then, since we used the ReLu activation function at the output layer of the deep neural network, we transformed the values to positive by taking the absolute value of the scores. The processed docking scores for each compound to each binding site then served as the prediction target. The code for model training can be found here: `https://github.com/2019-ncovgroup/ML-Code`.

The features used to train the models were computed molecular descriptors. The molecular descriptors were computed as described by Babuji et al [2020]. The full set of molec-

ular features is derived from the 2D ligand structures. The molecular features consist of 2-D and 3-D descriptors where 3D-descriptors are computed from the 2D structure using high-performance kernels Moriwaki et al. [2018]. The feature set results in a total of 1,826 descriptors. The approximately 6 million docking scores per receptor and 1,826 descriptors were then joined into a data frame for each receptor.

5.1.3 Learning curves

We performed learning curve analysis with the 3CLPro receptor to determine the training behavior of the modelPartin et al. [2021]. A subset of 2M samples were obtained from the full set of 6M samples. The 2M sample dataset was split into train (0.8), validation (0.1), and test (0.1) sets. We trained the deep neural network on subsets of training samples, starting from 100K and linearly increasing to 1.6M samples (i.e., 80% of the full 2M set). Each model was trained from scratch and we used the validation set to trigger early stopping and the test to calculate measures of generalization performance such as the mean absolute error of predictions.

5.1.4 Model Details

The model was a fully connected deep neural network with four hidden layers (with neuron counts [250, 125, 60, 30, 1]), with dropout layers in between. The dropout rate was set to 0.1. Layer activation was done using the rectified linear unit activation function. The number of samples per gradient update (batch size) was set to 32. The model was compiled using mean squared error as the loss function and stochastic gradient descent (SGD) with an initial learning rate of 0.0001 and momentum set to 0.9 as the optimizer. The implementation was Python using Keras Chollet et al. [2015].

The model was trained by setting the initial number of epochs to 400. A learning rate scheduler monitored the validation loss and reduced the learning rate when learning stag-

nated. The number of epochs with no improvement after which the learning rate was reduced (patience) was set to 20. The factor by which the learning rate will be reduced was set to 0.75, and the minimum allowable learning rate was set to 10^{-9}. Early stopping was used to terminate training if after 100 epochs the validation loss did not improve.

Features were standardized by removing the mean and scaling to unit variance before the onset of training using the entire data frame (before the data frame was split into train and test partitions). The train and test partitions were based on a random 80:20 split of the input data frame. Hyperparameter optimization was performed (see section 5.2.7).

Inferencing was performed on Summit. The input was converted to Feather files using the python package feather, a wrapper around pyarrow.feather (see the Apache Arrow project at apache.org). Feather formatted files as input in our experience are read faster from disk than parquet, pickle, and comma-separated value formats.

5.2 Results

5.2.1 Identification of protein targets and binding receptors

A total of thirty one receptors representing 9 SARS-CoV-2 protein conformations were prepared for docking. These are illustrated in Figure 2 and listed in Table 1. The quality of the receptors reflect what was available at the time the receptor was prepared. For example, whereas the NSP13 (helicase) structure in Table 1was based on homology modeling, today there exists x-ray diffraction models.

5.2.2 Generation of training data

The results for the 3CLPro receptor demonstrate a normal distribution (Fig. 5.1). The best docking scores would be in the range of 12 to 18. The distribution of docking scores for the 3CLPro receptor is illustrative of the distributions for all the other receptors. As shown in

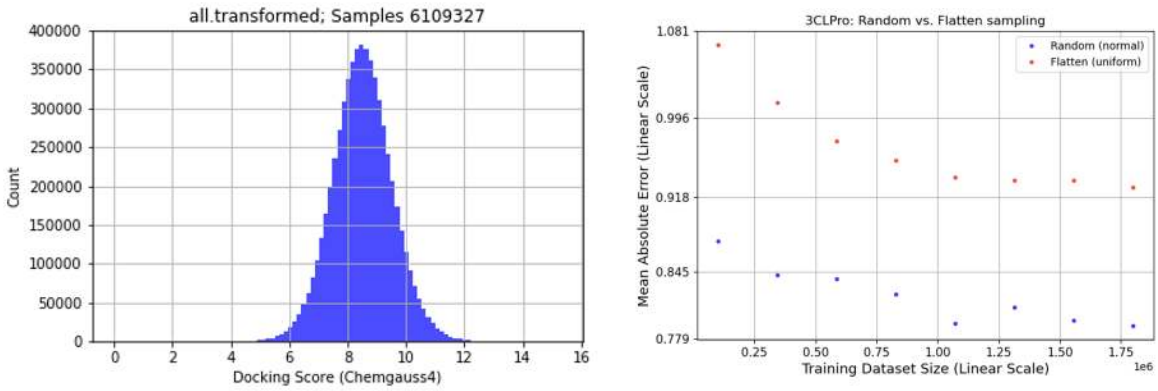


Figure 5.1: (left) **Histogram of protein-ligand docking of transformed docking scores for 3CL-$\mathbf{M_{pro}}$.** The distribution is from the ORZ dataset based on the transformed 2D scores. (right) **Learning curve between dataset size and MAE between random and flattened datasets.**

the figure, there are very few samples with good docking scores relative to the entire set of samples.

5.2.3 Sampling comparisons

We constructed a set of data frames to investigate the impact of the number of samples, sampling approach, and the choice of drug descriptors as features. The number of samples was further investigated using learning curves. Because we are interested in predicting docking scores in the tail of the distribution where the best docking scores exist, we explored two sampling approaches. Lastly, we investigated the impact of using the Mordred 3-D descriptors as features of the compounds.

Dataset	Count (samples)	Sampling method	Distribution (approximate)
100K-random	100,000	Random	Normal
100K-flatten	100,000	Flatten	Uniform
1M-random	1,000,000	Random	Normal
1M-flatten	1,000,000	Flatten	Uniform

Table 5.1: The four sampling approaches used to subset the approx. 6M docking scores for OZD.

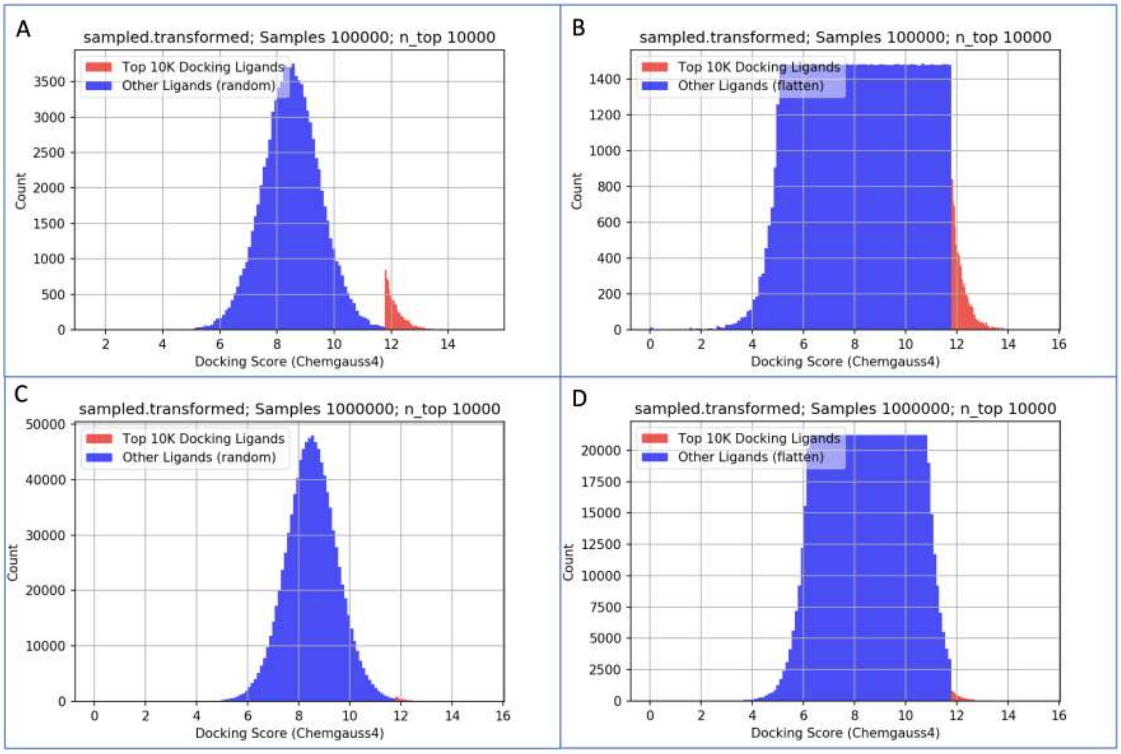


Figure 5.2: Docking score histograms for each of the four sampling a) 100K-random, b) 100K-flatten, c) 1M-random and d) 1M-flatten approaches used to generate a subset by sampling the full dataset of available scores (approximately six million samples).

We generated a dataset subset by sampling the approximately 6M samples in the OZD data complete data-frames. We examined four sampling approaches, differing by two parameters, as listed in Table 5.1: (1) the total number of samples drawn from the entire dataset (i.e., the count), and (2) the algorithm used to draw the samples (i.e., the sampling method).

Drawing samples at random preserves the original normal-shaped distribution (thus, the name Random). Alternatively, for a more balanced dataset, we sample scores with an alternative algorithm to create a roughly flattened, uniform-like distribution. To include the highly significant, top score samples, we retain the top ten thousand binding ligands. Figure 5.2 shows the histograms of the docking scores subset with each of the four sampling scenarios for 3CL-M_{pro}. The top ten thousand binding ligands are indicated in red. Note that the distribution of the full dataset can be roughly modeled as a normal distribution, as

1613 Features

Model	**epoch**	**val loss**	**val MAE**	**val r^2**
V5.1-100K-flatten-2	337	0.80	0.66	0.71
V5.1-100K-random-2	336	0.80	0.66	0.71
V5.1-1M-flatten-2	484	0.60	0.59	0.81
V5.1-1M-random-2	455	0.49	0.52	0.68

1826 Features

Model	**epoch**	**val loss**	**val MAE**	**val r^2**
V5.1-100K-flatten-2	313	0.97	0.74	0.85
V5.1-100K-random-2	330	0.81	0.67	0.71
V5.1-1M-flatten-2	462	0.60	0.59	0.81
V5.1-1M-random-2	456	0.52	0.54	0.67

Table 5.2: Impact of including Mordred 3-D descriptors in the training data for the different sampling strategies.

shown in Figure 5.1.

When examining the impact of including the Mordred 3-D descriptors in the feature set, we average the validation loss, validation MAE, and validation r^2 across the 31 models as we are interested in the aggregate performance of the models across the 31 receptors. Our analysis of the inclusion of the Mordred 3D descriptors is presented (Table 5.2). Our results show no significant advantage to including the 3D descriptors. The results show small improvements in the validation loss across all training data frames when using only the 2D descriptors. The results are mixed when considering validation r^2, with two smaller data frames performing slightly better and the two larger data frames performing marginally worse. While we do not consider the differences in most cases to be significant, we demonstrate that adding the extra training parameters in the form of 3D descriptors does not improve the training performance of the model.

When examining the impact of both the training set size (1M or 100K) and sample selection from either a random distribution or flattened distribution, we average the validation

loss, validation MAE, and validation r^2 for each trained receptor model that represents one of the thirty one different protein pockets. Table 5.2 shows the differences between the means. A negative value for the validation loss and validation MAE differences would indicate 1M samples achieved a higher quality model, and a positive value for the validation r^2 difference would indicate 1M samples achieved a higher quality model. The results indicate that 1M samples from a flattened distribution perform better than 100K samples for all three metrics, whereas 1M samples from a random distribution achieved better metrics for the validation loss and MAE. However, the 1M samples from a random distribution had a lower validation r^2.

To better understand the differences between the 1M data sets, the Pearson correlation coefficient was calculated between predicted and the observed values from the validation set for each pocket model. In the case of the v5.1-1M samples, the validation set had 200,000 samples. The mean of the PCC across the set of pocket models was calculated for each 1M data set and the V5.1-1M-random is 0.853 and the V5.1-1M-flatten is 0.914.

5.2.4 Learning curve analysis

To further explore the optimal sample size, we generated learning curves for the 3CLPro receptor model and assume 3CLPro will be indicative of other receptors. Using the entire dataset, which contains approximately 6M samples, imposes a significant computational burden for training a deep neural network model for each receptor and performing HP tuning. Regardless of the learning algorithm, supervised learning models are expected to improve generalization performance with increasing high-quality labeled data. However, the rate of model improvement achieved by adding more samples diminishes at specific sample sizes. The trajectory of generalization performance as a function training set size can be estimated using empirical learning curves.

The range at which the learning curve starts to converge indicates the sample size where

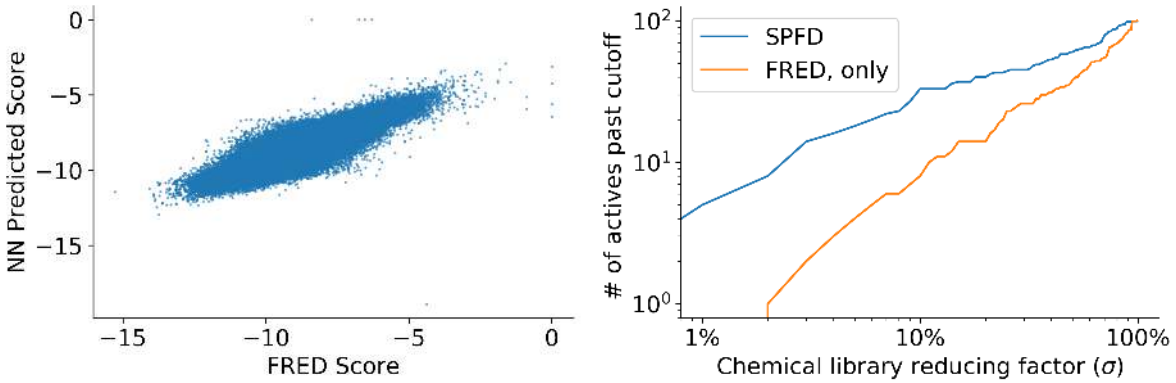


Figure 5.3: (left) Scatter plot illustrating correlation between the predicted scores and the FRED scores (for 3CL-main protease on a 100,000 random subset of orderable MCule molecules). (right) Detection of active compounds from NCATS (AC50$\leq$ 10μM) with SPFD (predicted with NN) and FRED (docking). SPFD detects all active compounds which FRED detects for 3CL-main protease and therefore is a faster alternative to regular docking without loss of active detection. This indicates the differences between the predictions and the actual FRED scores lean towards detecting actives.

the model begins to exhaust its learning capacity. Figure 5.1 shows the learning curve where the mean absolute error of predictions is plotted versus the training set size. The curve starts to converge at approximately 1M samples, implying that increasing sample size beyond this range is not expected to improve predictions.

5.2.5 Model Accuracy

FRED docking scores correlated (0.825) with the neural network predictions (see fig. 5.3). Furthermore, the variation between the NN and the actual FRED scores did not worsen the detection of active molecules. We utilized molecules from a set of 3CL-main protease screening data from National Center for Advancing Translational Sciences open data portal Brimacombe et al. [2020]. Molecules from this dataset with an AC50 of 10μM or less were considered active. Based on a filter cut-off, the NN was able to detect as many active compounds as FRED would (see fig. 5.3).

The observations of the data frame comparisons (sec. 4.2.3) and learning curves (sec. 4.2.4) show that the 1613 MOrdred 2D descriptors performed better without the inclusion

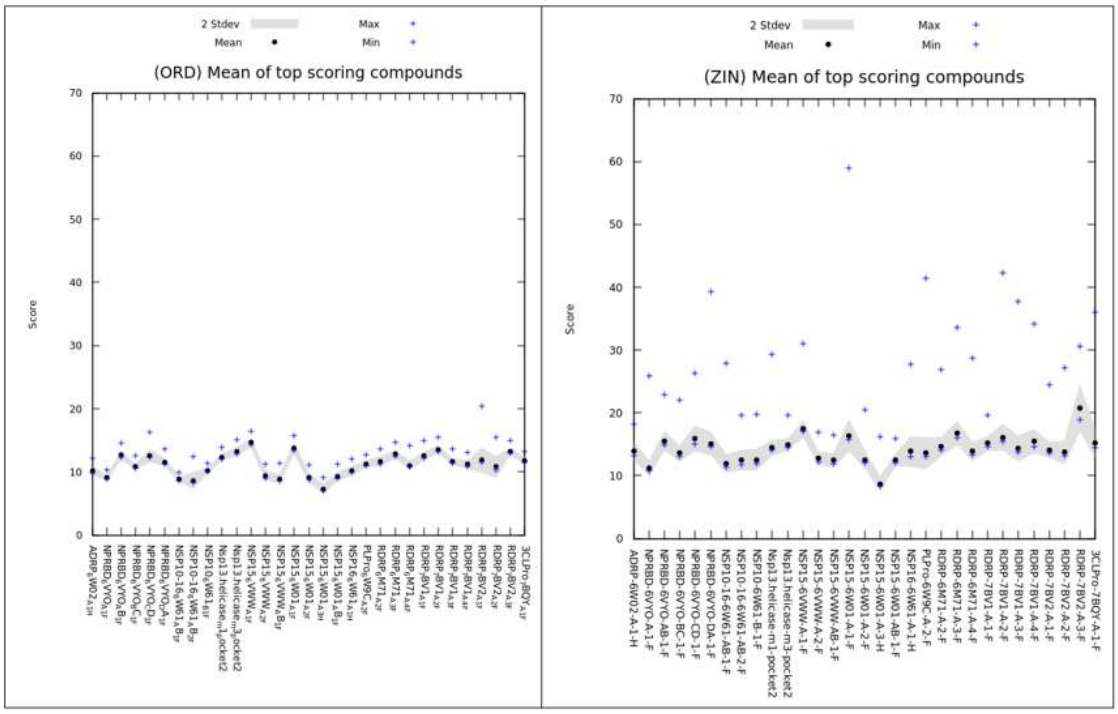


Figure 5.4: Comparison of the 31 receptor models with the 2000 best scoring compounds from ORD and ORZ.

of the 3D based descriptors (in total, 1826 features) in most cases. The 1M data frames performed better than the 100K data frames in most cases. The mean r^2 (0.825) of the 1M-flatten was higher than that of the 1M-random data frame (0.721).

5.2.6 Inference across 3.8 billion compounds

We divided the 4 billion compounds into 4 input data sets to enable better utilization of resources. ENA, G13, ZIN, OTH. We also constructed a set of compounds from the MCULE data set that could be easily purchased (organic synthesis already done). The MCULE subset was named ORD. The inferencing rate was approximately 50,000 samples per second per GPU, and all 6 GPUs per summit node were used.

We analyzed the results for each receptor by selecting the top 2000 scoring compounds, and computing mean, standard deviation, maximum, and minimum values. We present two examples of these results in Figure 5.4. Interestingly, the range represented by the maximum

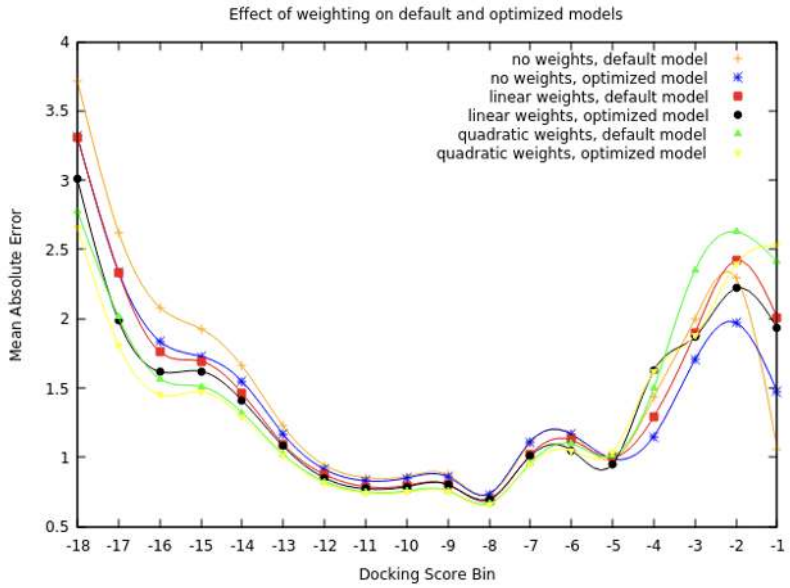


Figure 5.5: (left) **Effects of sample weighting strategies on the default and optimized model.** The docking score bins represent buckets where scores fall into and the $y - axis$ refers to the mean absolute error (MAE) of a model when using it to predict the docking scores. The different lines represent different optimization strategies between models.

and minimum predicted scores for the best 2000 scoring compounds is remarkably different between these two. In fact, ZIN was representative of the others (G13, ENA, and OTH). One working hypothesis is that the compounds in ORD are synthesizable, whereas compounds in the other sets are not necessarily synthesizable as these are virtual combinatorial libraries.

5.2.7 Model Hyperparameter Optimization

The CANDLE framework was subsequently used to tune the deep neural network for future training and screening activities Wozniak et al. [2018b]. The CANDLE compliant deep neural network was tuned in two phases. The first involved using two CANDLE hyperparameter optimization workflows - mrlMBO and GA. Each differs in the underlying ML techniques used to optimize the hyperparameters. The second phase involved implementing and testing new sample weighting strategies in an attempt to weight the samples at the good end of the distribution more heavily during training. Results of the GA and mlrMBO workflows

produced a model architecture that had a 6.6% decrease in the validation mean absolute error and a 2.8% increase in the validation R-squared metrics.

Efforts to decrease the error in the good tail of the distribution (where the docking scores are best) focused on adding sample weights to the model while training. We investigated linear and quadratic weighting strategies. We applied the weighting strategies to both the default model as well as the hyperparameter optimized model. The linear strategy weights the sample proportionally with the docking score, while the quadratic scales with the square of the docking score. These strategies generic in that they can be applied to basically any training target value. To analyze the impact of the weighting strategies, we computed the mean absolute error on bins of predicted scores with a bin interval of one. These results are presented in Figure 5.5.

5.3 Conclusion

We demonstrate an accelerated protein-ligand docking workflow with surrogate models, which is at least 10x faster than traditional docking with nearly zero loss of detection power. We utilize neural network models to learn a surrogate mode to the CPU-bound protein-ligand docking code. The surrogate model has a throughput over six orders of magnitude faster than the standard docking protocol. By combining these workflows, utilizing the surrogate model as a prefilter, we can gain a 10x speedup over traditional docking software without losing any detection ability (for hits defined as the best scoring 0.1% of a compound library). We utilize regression enrichment surfaces to perform this analysis in chapter 8 to analyze this case further. The regression enrichment surface plot is more illustrative than the typical accuracy metrics reported from deep learning practices. We released over 200 million 3D pose structures and associated docking scores across the SARS-CoV-2 protemome This 10x speedup means if a current campaign takes one day to run on library size L, one can screen ten times as many compounds in the same amount of time without missing leads. Given the

potential for 100x or even 1000x speedup for docking campaigns, we hope to advance the ability of surrogate models to filter at finer levels of discrimination accurately.

CHAPTER 6

TIERED-WORKFLOWS

This chapter will address my work with tiered-workflows—where cheap but inaccurate surrogates are used to screen compounds, sending a smaller subset to a more expensive but also more accurate kernel (such as simulation). The pipeline we discuss is called IMPECCABLE [Saadi et al., 2021, Bhati et al., 2021, Clyde et al., 2019]. *The first section will introduce the idea, the second section will outline the theoretical considerations (extending the mathematics from* [Woo et al., 2021]*, the third section will outline the results of the pipeline with some of the COVID-19 molecules, and finally the last section will discuss the performance characteristics.*

6.1 Background

Application of ML approaches to problems in ligand pose and affinity prediction is increasingly common and is included in top ranked entries in competitions such as the D3R and SAMPL community challenges [Rizzi et al., 2018, Sunseri et al., 2019]. Nonetheless, the performance of even best in class scoring functions remains below that of expensive simulation based alchemical BFE (see section 2.4) methods [Jimenez et al., 2018]. Ash and Fourches used MD simulations of 87 inhibitors to the ERK2 kinase to create descriptors which distinguish bioactive molecules more effectively than available Quantitative Structure–Activity Relationship (QSAR) models [Ash and Fourches, 2017].

The ABL1 kinase used as the target in our experiments to the fact that previous studies have used this system to test the ability of alchemical BFE methods to determine the impact of mutations on binding strength for a range of clinically approved drugs [Hauser et al., 2018]. This study considered 144 protein-ligand combinations and this database has been used as the basis of comparisons of multiple techniques. Aldeghi and coworkers [Aldeghi

et al., 2019] compared the performance of simulation based BFE techniques and a range of knowledge based estimators (including multiple ML models) finding that the two approaches were complementary and multiple techniques used in a consensus fashion achieved “remarkable accuracy".

Our work looks to extend the state of the art by incorporating much larger number of systems in order to refine both the simulation and BFE methodologies and the forcefields which under pin all biomolecular simulation. One aim of our framework is to enable the evaluation and enhancement of general small molecule forcefields for small molecule BFEs. Within the biological simulation community this effort is being led by the Open Force Field Initiative (openforcefield.org), who recently benchmarked a new forefield, SMIRNOFF99Frosst, using 43 host-guest complexes [Slochower et al., 2019].

In silico drug design presents intellectual challenges and is driven by clear imperatives of societal good and economic incentives. Recent methodological and infrastructural advances range from improved docking protocols [Liao et al., 2019, Gentile et al., 2020] to sophisticated multi-stage pipelines [Gorgulla et al., 2020], and scalable infrastructure [Vermaas et al., 2020] for drug discovery pipelines.

Walters and Wang [2020] presents new methods and infrastructure focused on virtual screening. Brute-force traditional docking is unlikely to be fast or sophisticated enough for *in silico* drug design. Enhancing the ability of traditional docking protocols to sample larger chemical space is critical. The recent studies that combine docking with machine learning methods Liao et al. [2019], Gentile et al. [2020] report up to 6000x increase in chemical space sampled [Gentile et al., 2020] — without notable loss of favorably docked entities. To overcome the limitation that single docking protocols are not universally reliable, Ref. [Gorgulla et al., 2020] introduces VirtualFlow —an open-source platform that supports several powerful docking programs on scalable infrastructure.

ML is used to improve computer-aided drug discovery. Jones et al. [2021] integrates

ML with binding free energy calculations. ML is also used to extend quantitative structure-activity relationships approaches. For example, AMPL [Minnich et al., 2020] provides extensible pipelines that support the building and sharing of ML models, along with automating model training to improve key pharma-relevant parameters predictions. AMPL highlights the trend towards sophisticated and open-source software platforms that can be deployed on diverse suitable scalable infrastructure as per requirements.

While clusters and clouds are pervasive and scalable platforms for pharma industry, HPC and leadership platforms continue to provide important and powerful platforms [Trager et al., 2016] for academia and government. Ref. [Smith and Smith, 2020] have recently developed drug discovery pipelines on supercomputers that integrate multiple molecular modeling techniques — temperature replica-exchange advanced sampling techniques with docking protocols. Similarly, Acharya et al. [2020] used ensemble-docking approaches to overcome limitations for single docking protocols. Vermaas et al. [2020] impressively used all ≈27500 GPUs on the Summit supercomputer for a sustained (average) molecular docking of ≈19000 compounds per second [Glaser et al., 2020] using Autodock-GPU [LeGrand et al., 2020]. Over one billion compounds were docked [Glaser et al., 2020] to two SARS-CoV-2 protein structures with full optimization of ligand position and 20 poses per docking, each in under 24 hours.

Spurred by, but not limited to COVID-19, the aforementioned recent publications reiterate the importance of both methodological and infrastructural enhancements. Consequently, our infrastructure is significant, as it: (i) provides scalable infrastructure for diverse workflows that are heterogeneous in distinct ways; (ii) delivers high-performance independent of the workflow and heterogeneity types, and similar to the highest values previously reported [Glaser et al., 2020]; and (iii) is developed using a common set of middleware building blocks which supports the flexible composition of diverse methods – ensemble docking, ML surrogates for docking, ML-enhanced advanced sampling and binding free energies of

differing granularity – into an integrated campaign with high end-to-end throughput.

6.2 Tiered workflows

Our approach relies upon the creation of workflows which combine expensive but accurate free energy calculations with fast ML models. While ultimately we intend to predict the affinity of compounds to wide ranges of disease-relevant kinases and clinically identified kinase mutants, we have initially focused on a subgoal; prioritizing the use of MD simulations to assign binding affinities to small molecule on a large set of small molecule drug candidates. Given a vast set of candidate drugs, what is the optimal ordering of simulating candidates to improve overall predictive screening performance using limited computer resource? Addressing this question is the basis of our prototype workflow (which initially assesses small molecule binding to a single kinase, ABL1, which is dysregulated in chronic myelogneous leukemia and has been the most widely studied kinase for the development of selective inhibitors), described below.

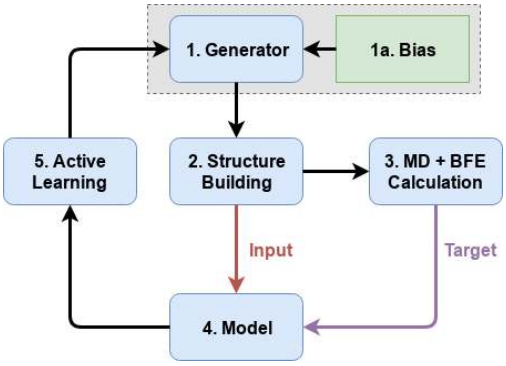


Figure 6.1: **Schematic overview of the integrated ML-MD workflow.**

Steps 1 & 2 The generation module samples from a known dataset (producing candidates as SMILES strings), but we will scale this to sampling from a variational autoencoder guided by a biasing filter. Bias is an optional module which restricts the generation module based on a particular subspace, dataset, or biochemical feature, allowing explicit filtering using

functions available in RDKit or OpenEye. 3D compound coordinates are generated from the SMILES, and docked into the pre-prepared protein conformation. The docking score is the first (and cheapest) binding strength estimate passed to the ML model.

Step 3 The structure of the protein-ligand complex is prepared for simulation using one of our chosen BFE protocols, which are executed to provide trajectories and binding free energy estimates (with associated uncertainties).

Step 4 Model is a deep neural network which predicts the binding free energy of a ligand. Initially the only input is the featurized SMILES string, though we will extend this to include topologies and trajectories.

Step 5 The Active Learning module ingests the SMILES, free energy estimate and Model output and returns information to the generator either in the form of the next sample or a space to continue sampling.

Execution of a prototype workflow requires the coordination not only of the overall workflow but multi-stage pipelines of molecular simulations. To support the scalable, adaptive and automated calculation of the binding free energies concurrently with ML method on HPC resources, we are developing workflow automation tools based on the RADICAL-Cybertools middleware building block approach [Balasubramanian et al., 2019]. This allows us to to attain both workflow flexibility and performance.

This workflow is executed on Summit (Oak Ridge National Laboratory), currently the world's fastest supercomputer. The NVIDIA Volta GPUs employed allow single OpenMM runs to generate 700+ nanoseconds of trajectory per day. However, the novel architecture of the system means tools that we have previously relied upon are currently unavailable. Consequently, our workflow has been adapted to make use of communication with a cluster running containers for docking and ligand preparation.

6.2.1 Molecular modelling

The starting point for physically based MD and BFE techniques is an atomistically detailed structure of the protein-ligand complex. The protein component of the system is typically based on an experimental crystal structure (although homology models may be used if none is available). In the experiments we perform here we use a single structure of the ABL1 kinase based on PDB: 4WA9. Compounds, initially represented as a SMILES string (i.e. in one dimension), are converted into three dimensional spatial coordinates and docked into the protein structure. For these processes we employ the OpenEye software suite, employing the hybrid approach for docking (which makes use of both known ligand engagement modes and physical interactions of the novel ligand with the receptor), in this study. Docking not only provides the coordinates of the protein-ligand complex but a score which provides a low cost, but imprecise, measure of binding strength.

The binding free energy of the protein-ligand interaction is approximated with one of two methods: (1) an energy minimized adjustment of the docked structure or (2) a 5 nanosecond simulation utilizing the molecular mechanics generalized Born surface area (MMGBSA) method. Subsequent work will integrate progressively more computationally costly binding free energy methodologies, such as alchemical relative and absolute free energy calculations, once software architecture support issues are solved, as the modular nature of the RADICAL EnTK workflows allow these additional models to be easily integrated at a later time.

Energy minimization, performed using the OpenMM package [Eastman et al., 2017a], is an extremely fast calculation that is likely to improve upon the docking score. MMGBSA is a slower and more accurate metric. Additionally, the uncertainty associated with the estimated BFE can be quantified. Previous work has demonstrated that the precision of MMGBSA-based BFE prediction can be refined to 0.5 kcal/mol [Sadiq et al., 2010].

6.2.2 Machine Learning

Given the various modes of target values, we use a multitask model with shared initial layer architecture and three independent loss functions and outputs [Ruder, 2017]. We make use of three primary target modes (this choice is arbitrary and can be expanded); docking score, energy minimization score, and MMGBSA BFE (from 5 nanosecond simulations). Samples are collected from the simulation pipeline continuous and consumed into the batch loader. Batches are homogeneous in property and trained without preference for one mode over another.

Two model types are trained in parallel; the first based on the ligand alone, the second using information from the protein-ligand complex. The ligand-only model does not incorporate features from the docked pose or the complex with the protein. Instead it is based on graph convolution of 2D graphs representing each compounds molecular structure [Coley et al., 2019]. In the complex based model, the three dimensional bound structure is transformed into a voxel grid of the ligand without the protein by the method discussed in [Skalic et al., 2019]. 3D convolution layers are used to produce a latent representation for use in downstream multitask regressors.

To effectively and efficiently sample the domain, the model's uncertainty in regions is quantified. Following Lakshminarayanan's approach to uncertainty estimation, the model's final layer outputs both $\mu(\vec{x})$ and $\sigma^2(\vec{x})$, and correspondingly altering the loss function to minimize the negative log-likelihood criterion [Lakshminarayanan et al., 2017]. We further follow their proposed adversarial training example and ensemble of models method using an ensemble of models trained in parallel.

6.3 Mathematics of tiered-workflows

In Woo et al. [2021], the mathematical problem of virtual screening with different granulates workflows is demonstrated. Given a library to screen L, and a series of functions with

increasing computational cost and increasing accuracy with respect to a target measure $\{f_i\}_{1,\ldots,n_f}$, let the initial set of hits be equal to the library, $X_0 = L$. Given a series of cut-off values $\{\lambda_i\}_{1,\ldots,n_f}$, we can write each stage of the pipeline as the following

$$X_i = \{x \in X_{i-1} | f_i(x) \geq \lambda_i\}. \tag{6.1}$$

Given a particular computational constraint, and the cost of screening each sample under each function c_i, we can write the computational cost as

$$C(\lambda_1, \ldots, \lambda_{n_f}) = \sum_{i}^{n_f} c_i \cdot |X_i|. \tag{6.2}$$

Finally, we may want to understand the discrepancy between each function. We can write the bias of each as having some bias b_i and some noise σ_i which is drawn from a normal distribution $\mathcal{N}(\sigma_i, b_i)$ such that each sample error x on a particular function f_i is $\widehat{\sigma_i} \sim \mathcal{N}(\sigma_i, b_i)$ so the overall discrepancy for a sample x which makes it through the whole pipeline is $\widehat{\sigma_{n_f}} \sim \mathcal{N}(\sigma_{n_f}, b_{n_f})$. Finally, for the pipeline setup, we demand that $\sigma_i \leq \sigma_{i-1}$ and $b_i \leq b_{i-1}$. We write the stage that a sample as an indicator function $\Gamma(x, i, \lambda_1, \ldots, \lambda_{n_f})$ equal to one if x is in X_i but not X_{i+1}. Thus, the error for each sample can be written as a function of λ_i given σ and b are fixed by the functions. Thus, one can think of these pipelines as an attempt to decrease the discrepancy of each sample while maintaing the particular computational cost

$$\lambda_1, \ldots, \lambda n_f = \underset{\lambda_1,\ldots,\lambda n_f}{\arg\min} \sum_{x \in X_0} \sum_{i=1}^{n_f} [\Gamma(x, i, \lambda_1, \ldots, \lambda_{n_f}) \mathbb{E}\delta(x)] \tag{6.3}$$

$$\text{subject to } C(\lambda_1, \ldots, \lambda_{n_f}) < \boldsymbol{C} \tag{6.4}$$

where $\boldsymbol{C}$ is the computational budget and $\delta(x)$ is the estimated discrepency for that sample

and stage.

Currently, there is no well known solution to this setup. In fact, the setup is more complicated because of relationships between samples set up in chapter 7. While this framework may not immediately present a clear solution, it offers a mental model for thinking through the idea of staged workflows.

6.4 Integration with HPC

In this section, we outline the performance and usage of this workflow through the use of RADAICAL-Cybertools.

6.4.1 RADICAL-Cybertools: Middleware Building Blocks

The RADICAL-Cybertools (RCT) software stack is used to support the adaptive, scalable and concurrent execution of heterogeneous tasks, where a task can vary from a single CPU or GPU node task, to an arbitrary large MPI task or of unspecified temporal duration. RCT are middleware building blocks [Turilli et al., 2019a] to develop efficient and effective workflow tools, designed to work as stand-alone systems, integrated among themselves or with third-party systems. RCT consists of three main components: Ensemble Toolkit [Balasubramanian et al., 2016, 2018] provides the ability to create and execute ensemble-based workflows/applications with complex coordination and communication but without the need for explicit resource management. It uses another RCT component—RADICAL-Pilot [Merzky et al., 2018] which provides resource management and task execution capabilities, which in turn use RADICAL-SAGA [Merzky et al., 2015] as an interoperable HPC batch-queue access layer. RCT provide scalable implementations of building blocks in Python and are currently used to support dozens of scientific applications on high-performance and distributed systems [Balasubramanian et al., 2019].

6.4.2 RADICAL-Pilot

Many scientific workloads are comprised of many tasks, where each task is an independent simulation or data processing analysis. The tasks might be heterogeneous, or the there might be distinct ensembles of homogeneous tasks. Traditionally, each task is submitted as an individual job, or MPI capabilities are used to execute multiple tasks as part of a multi-node single job. The former method suffers from unpredictable queue time for each job and the limited number of concurrent jobs that can be queued on HPC machines. The latter method is suitable to execute tasks that are homogeneous and have no dependencies, and relies on the fault tolerance of MPI. The execution of millions of tasks on heterogeneous HPC platforms requires scalable dynamic resource management and multi-level scheduling.

The Pilot abstraction [Turilli et al., 2018] solves these issues. This abstraction (i) uses a placeholder job to acquire resources via the local resource management system (LRMS), and (ii) decouples the initial resource acquisition from task-to-resource assignment. Once the pilot is scheduled via the LRMS, it can in turn schedule computational tasks on the available resources. This functionality allows for all the computational tasks to be executed directly on the resources, without being individually queued to the LRMS. Thus, this approach supports the requirements of task-level parallelism and high-throughput as needed by the science drivers.

RADICAL-Pilot (RP) is an implementation of the pilot abstraction, engineered to support scalable and efficient launching of heterogeneous tasks across different platforms. RP [Merzky et al., 2018] is a runtime system designed to decouple resource acquisition from task execution. As every pilot system, RP acquires resources by submitting a batch job, then bootstraps dedicated software components on those resources to schedule, place and launch application tasks, independent from the machine batch system [Turilli et al., 2018].

6.4.3 Integrated ML and MD Workflows on Summit

RADICAL Ensemble Toolkit (EnTK) is designed to support the concurrent execution of the RL based integrated and adaptive ML-MD workflow of Figure 6.1 as a set of concurrent and interacting computational pipelines. Each pipeline is composed of stages and each stage contains an arbitrary set of tasks. Tasks can execute concurrently while stages can execute only sequentially. These properties are insured by design, offering what we have called a Pipeline Stage Task (PST) model for the specification of computational workflows. It is important to note that 'task' here are not functions, methods or sub-processes of one of EnTK components. Task indicates instead a self-contained process (i.e., program) executed and managed by the operating system of the target resource. Consistently, tasks can be a single-threaded, multi-threaded or MPI program, and can use CPUs, GPUs or both within and across the compute nodes of a target machine.

Specified in PST, the workflow of Fig. 6.1 consists of a set of two pipelines: Pipeline 1 executes multiple stages, each with multiple MD simulations; Pipeline 2 executes a single stage with one or more ML tasks. Pipeline 1 and 2 execute concurrently, sharing their tasks' input/output files via a shared file-system. Initially, Pipeline 1 executes MD simulations to train the ML tasks of Pipeline 2. Once trained, Pipeline 2 uses ML to infer the initial state of the next MD simulations of Pipeline 2. Currently, we have implemented the training capabilities of the workflow. In a near future, we will develop also the inferring capabilities, which will then integrate with the RL loop.

6.5 Results

We discuss the many application- and platform-level issues that need to be addressed to execute our pipeline at scale. We also quantify performance bottlenecks and their determinant factors, a necessary indication on how to further improve the pipeline for its deployment in sustained production. Overall, our performance analysis is an indicator of challenges and

solution for pipelines that go beyond the specific requirements of our COVID-19 campaign.

6.5.1 WF1: Ensemble Consensus Docking

Compared to physics-based simulation methods, docking is a relatively inexpensive computational process. To increase the reliability of docking results, we prefer multiple docking protocols for the same ligand set and protease over individual docking scores. WF1 uses OpenEye and Autodock-GPU to leverage resource heterogeneity: the former executes on x86 architectures (e.g., Frontera); the latter on GPUs (e.g., Summit).

For each of the identified protein target[1] sites, WF1 iterates through a list of ligands and computes a docking score for that ligand-protein target pair. The score is written to disk and is used as filter to identify ligands with favorable docking results (score and pose). The docking call is executed as a Python function in OpenEye, and as a self-contained task process in AutoDock-GPU. In both cases, the RAPTOR framework is used for orchestration.

The duration of the docking computation depends on the type of CPU (OpenEye) or GPU (AutoDock-GPU) used, and the computational requirements of each individual protein target. We measure the docking time (seconds) and docking rate (docks/hr) of three use cases: (1) production runs for NVBL-Medical Therapeutics campaigns; and runs for largest achievable size on (2) Frontera and (3) Summit. Table 6.1 summarizes the parameterization and results of the experiments we performed for each use case.

WF1 assigns one pilot for each protein target to which a set of ligands will be docked. Within each pilot, one master task is executed for every ≈100 nodes. Each master iterates at different offsets through the ligands database, using pre-computed data offsets for faster access, and generating the docking requests to be distributed to the worker tasks. Each worker runs on one node, executing docking requests across the CPU cores/GPUs of that node.

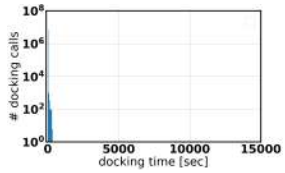
1. We define a protein target as a specific PDB file with a well defined binding site (according to how the specific molecular docking code requires) against which we dock the small molecule libraries.

Use Case	Platform	Application	Nodes	Pilots	Ligands [$\times 10^6$]	Utilization avg/steady	Docking Time [sec] min	max	mean	Docking Rate [$\times 10^6$/hr] min	max	mean
1	Frontera	OpenEye	128	31	205	90% / 93%	0.1	3582.6	28.8	0.2	17.4	5.0
2	Frontera	OpenEye	7650	1	126	79% / 95%	0.1	14958.8	61.5	20.1	35.8	25.2
3	Summit	AutoDock-GPU	1000	1	57	95% / 95%	0.1	263.9	36.2	10.9	11.3	11.1

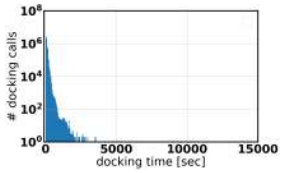
Table 6.1: WF1 use cases. For each use case, RAPTOR uses one pilot for each protein target, computing the docking score of a variable number of ligands to that protein target. OpenEye and AutoDock-GPU implement different docking algorithms and docking scores, resulting in different docking times and rates. Resource utilization is often impeded by the long tail docking time distributions which cause an expensive cooldown period. However, the steady state resource utilization is >=90% for all use cases.

Use Case 1

We assigned each of the 31 targets to a single pilot, i.e., to an independent job submitted to the HPC machine's batch-queue. Due to the different batch-queue waiting times, at most 13 concurrent pilots executed concurrently. With 13-way pilot concurrency, the peak throughput was $\approx 17.4 \times 10^6$ docks/hr. To keep an acceptable load on Frontera's shared filesystem, only 34 of the 56 cores available were used.



(a)



(b)

Figure 6.2: WF1, Use Case 1: Distribution of docking runtimes with the (a) shortest and (b) longest average docking time out of the 31 protein targets analyzed. The distributions of the docking runtimes all 31 protein targets have a long tail.

Figs. 6.2a and 6.2b show the distribution of docking times for protein targets with the shortest and longest average docking time, using the `Orderable-zinc-db-enaHLL` ligand database. All protein targets are characterized by long-tailed docking time distributions. Across the 31 protein targets, the min/max/mean docking times are 0.1/3582.6/28.8 seconds (Tab. 6.1), posing a challenge to scalability due to the communication and coordination overheads. The long tail distributions necessitate load balancing across available workers to maximize resource utilization and minimize overall execution time.

We addressed load balancing by: (i) communicating tasks in bulk so as to limit the communication frequency and therefore overhead; (ii) using multiple master processes to limit the number of workers served by each master, avoiding bottlenecks; (iii) using multiple concurrent pilots to partition the docking computations of the set of ligands.

Figs. 6.3a and 6.3b show the docking rates for the pilots depicted in Figs. 6.2a and 6.2b, respectively. As with dock time distributions, the docking rate behavior is similar across protein targets. It seems likely that rate fluctuations depend on the interplay of machine performance, pilot size, and specific properties of the ligands being docked, and the target protein target. We measure a min/max docking rate of $0.2/17.4 \times 10^6$ docks/hr with a mean of 5×10^6 docks/hr (Tab. 6.1).

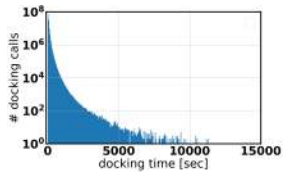
(a) (b)

Figure 6.3: WF1, Use Case 1: Docking rates for the protein target with (a) shortest and (b) longest average docking time.

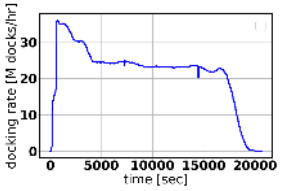
Use Case 2

Fig. 6.4a shows the distribution of docking times of approximately 126×10^6 ligands from the `mcule-ultimate-200204-VJL` library to a single protein target using OpenEye on Frontera. Note that the distribution is highly dependent on the protein target being used: for the specific protein target used in this run, we measure a min/max of 0.1/14985.8 seconds and a mean of 61.5 seconds (Tab. 6.1). The set of protein targets available to us varied in mean docking time from ≈3 to ≈70 seconds.

Fig. 6.4b shows the docking rate for a single pilot with 7650 compute (428,400 cores at 56 cores/node). Compared to Use Case 1, the rate does not fluctuate over time. After peaking at $\approx 35.8 \times 10^6$ docks/hr, the rate stabilizes at $\approx 25 \times 10^6$ docks/hr until the end of the

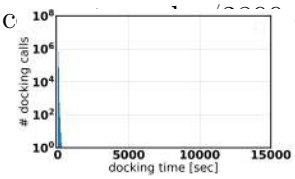


(a)

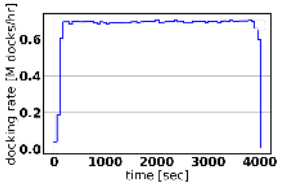


(b)

Figure 6.4: WF1, Use Case 2: (a) Distribution of docking time and (b) docking rate for a single protein target and 126×10^6 ligands. Executed with 158 masters, each using ≈50 c cores on Frontera.



(a)



(b)

Figure 6.5: WF1, Use Case 3: (a) Distribution of docking time and (b) docking rate for a single protein target and 57×10^6 ligands. A pilot is concurrently executed on Summit with 6000 GPUs.

execution (Tab. 6.1). Note that the long tail distribution of runtimes results in a long tail of docking calls and thus on a long "cooldown" phase. That phase ultimately lowers utilization from 92.3% in the steady-state (before cooldown starts) to a total average of 79.3%.

As discussed, the docking times depend on the protein targets used, and thus the docking rate inversely depends on that protein target choice. The range of rates is very wide: for the protein targets available to us, we observed a mean docking rate between $\approx 14 \times 10^6$ and $\approx 300 \times 10^6$.

Use Case 3

Figure 6.5a shows the distribution of the docking times of $\approx 57 \times 10^6$ ligands from the `mcule-ultimate-200- 204-VJL` database to a single protein target using AutoDock-GPU on Summit. The distribution has a min/max/mean of 0.1/263.9/36.2 seconds (Tab. 6.1). Compared to Use Case 1, Fig. 6.2, max docking time is shorter, but the mean is longer. Compared to Use Case 2, Fig. 6.4a, both max and mean are shorter. As observed, those differences are due to specific properties of the docked ligands and the target protein target.

Fig. 6.5b shows the docking rate for a single pilot with 1000 compute nodes, i.e., 6000

GPUs. Different from Use Case 1 and 2, the rate peaks very rapidly at $\approx 11 \times 10^6$ docks/hr and maintains that steady rate until the end of the execution. The cooldown phase is also very rapid. We do not have enough data to explain the observed sustained dock rate. As with Use Case 2, we assume an interplay between the scoring function and its implementation in AutoDock-GPU and specific features of the 57×10^6 docked ligands.

Different from OpenEye on Frontera, AutoDock-GPU bundles 16 ligands into one GPU computation in order to efficiently use the GPU memory, reaching an average docking rate of 11.1×10^6 docks/hr (Tab. 6.1). Currently, our profiling capabilities allow us to measure GPU utilization with 5% relative error. Based on our profiling, we utilized between 93 and 98% of the available GPU resources.

6.5.2 WF2: ML-Driven Enhanced Sampling

WF2 is an iterative pipeline composed of 4 stages. After the first iteration of the 4 stages is completed, if outliers were found, the next iteration starts simulating those outliers; otherwise, the simulation continues from where it stopped in the previous iteration. The pipeline stops after a predefined number of iterations.

We measured RCT overhead and resource utilization of WF2 to identify performance bottlenecks. We define RCT overhead as the time spent not executing at least one task. For example, the overhead includes the time spent bootstrapping environments before tasks execution, communicating between EnTK and RabbitMQ (RMQ), or between EnTK and RP while workloads wait to execute. Resource utilization is the percentage of time when resources (CPUs and GPUs) are busy running tasks.

The blue bars in Fig. 6.6 show RCT overheads for the first version of WF2 and how RCT overheads grew with iterations. WF2 may require a variable number of iterations. Thus, our goal was to reduce RCT overhead, and importantly, make it invariant of the number of iterations.

Initial analysis suggested multiple optimizations of WF2: some of these involved improving the deep learning model, the outlier detection, and RCT. For the latter, we improved the communication protocol between EnTK and RMQ, and we reduced the communication latency between EnTK and RMQ. We avoided sharing connections to RMQ among EnTK threads, reducing multiple concurrent connections, and reused communication channels whenever possible.

Fig. 6.6 (orange) shows the combined effects of improving DeepDriveMD and EnTK communication protocol which reduced the overheads by 57% compared to Fig. 6.6 (blue). However, they were still growing with the number of iterations. We moved our RMQ server to Slate, a container orchestration service offered by OLCF, which reduced the communication latency between EnTK and RMQ, as shown in Fig. 6.6 (green). The optimization allowed RCT overheads to be invariant up to 8 WF2 iterations.

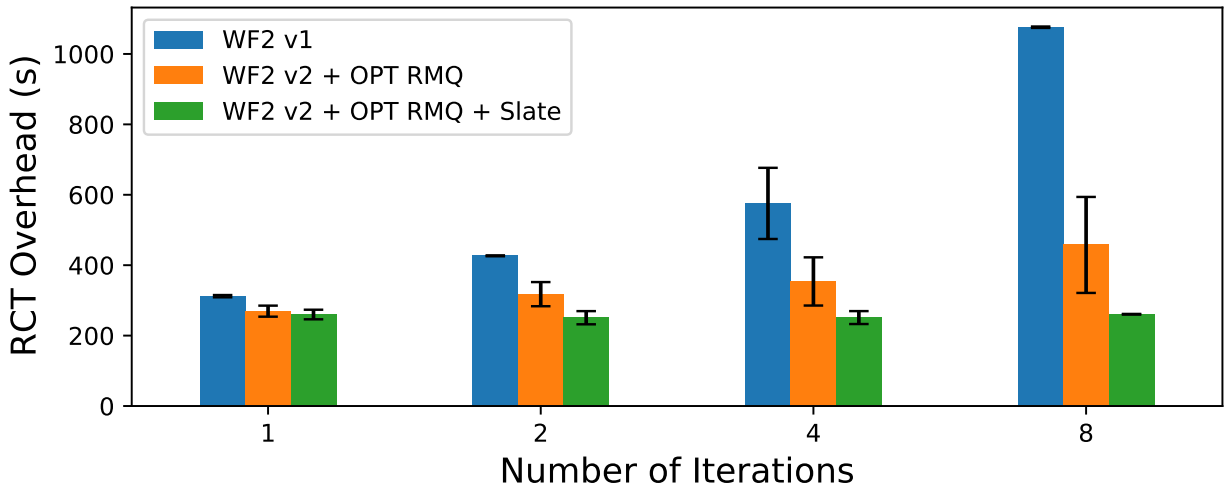


Figure 6.6: RCT overhead reduction with improved WF2, EnTK and RabbitMQ.

Fig. 6.7 depicts resource utilization for different (internal) RCT states as a time series. The region with "yellow, light blue, or green" colors represents unused resources; "dark" represents resource usage. Fig. 6.7 shows resource utilization of WF2, when executing four pipeline iterations on Summit with 20, 40, and 80 compute nodes. Note that most of the unused resources are CPU cores that are not needed by WF2. Overall, we measured 91%,

91%, 89% GPU utilization respectively. Across scales, Fig. 6.7 shows differences in the execution time of some of the pipeline stages but no relevant increase of the time spent without executing at least one task.

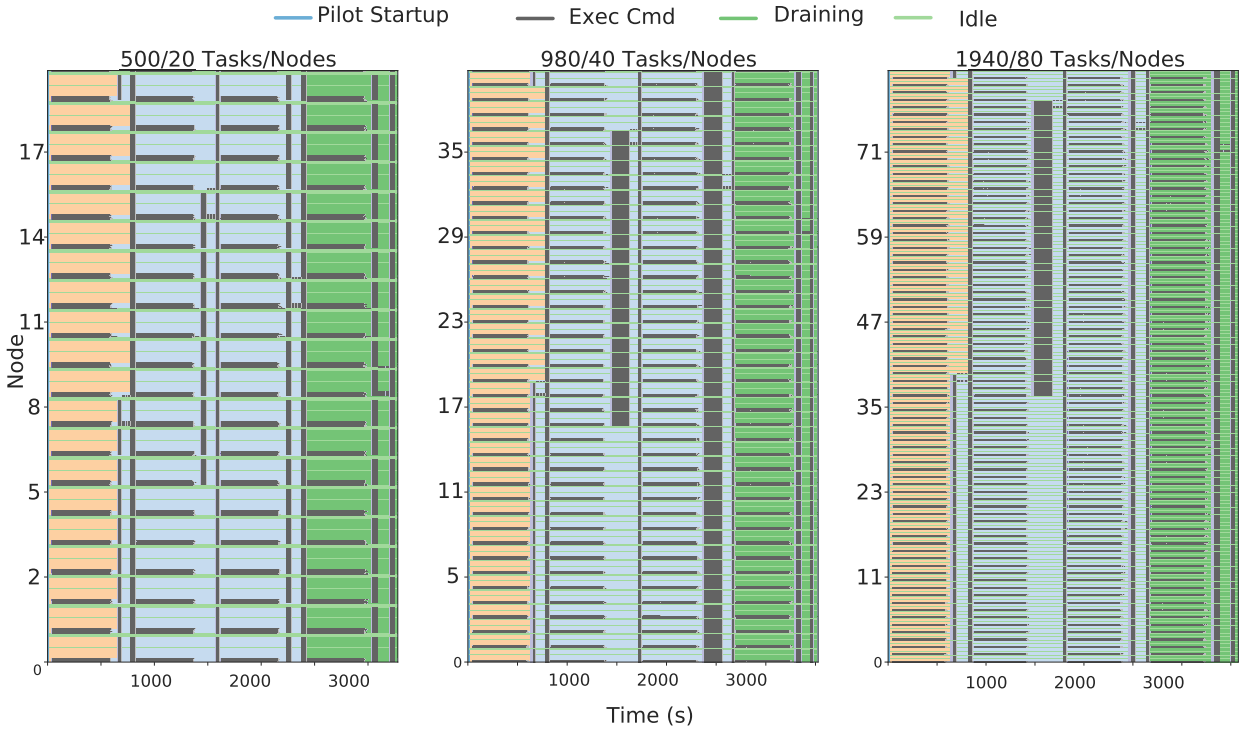


Figure 6.7: Timeline of RCT resource usage for WF2 when investigating weak scaling properties (from 20 to 80 nodes).

6.5.3 Hybrid WF3 & 4 Workflow

WF3 and WF4 are computationally intensive methods that cost several orders of magnitude more node-hours per ligand than WF1 [Saadi et al., 2021]. WF3 and WF4 both compute binding free energies but have workloads comprised of distinct tasks: GPU-based OpenMM and CPU-based NAMD tasks, respectively. Merging WF3 and WF4 into a single hybrid workflow allowed us to improve resource utilization by employing RP's unique capability to execute distinct tasks concurrently on CPU cores and GPUs. We evaluated that capability by measuring: (i) RCT overhead (as defined previously) as a function of scale; (ii) scalability as a function of problem and resource size; and (iii) resource utilization.

Fig. 6.8 compares RCT overhead to workflow time to completion (TTX) on 32 nodes for different task counts, representing different production workflow configurations. TTX in Fig. 6.8(c) illustrates concurrent execution of GPU and CPU tasks. The modest increase in TTX compared to Fig. 6.8(b) is likely due to interference from sharing resources across tasks, and some scheduler inefficiency. A careful evaluation and optimization will form the basis of further investigation. Fig. 6.8(d) plots the TTX for the *Hybrid-LB* scenario when the number of WF3 and WF4 tasks are selected to ensure optimal resource utilization. The number of WF3 tasks completed in Fig. 6.8(d) is twice the number of WF3 tasks completed in Fig. 6.8(c), with no discernible increase in TTX.

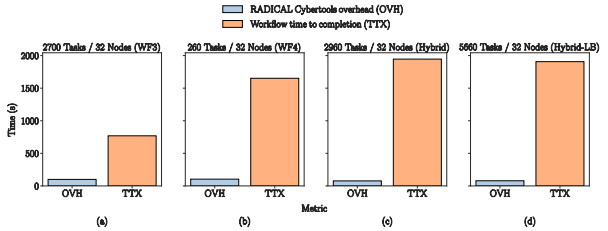


Figure 6.8: RCT Overhead in Hybrid Workflows.

Fig. 6.9 depicts RCT resource utilization for the configurations of Fig. 6.8(c) and Fig. 6.8(d). As with Fig. 6.7, "green space" represents unused resources; "dark space" represents resource usage. WF3 and WF4 have 4 and 3 stages, respectively, which can be discerned from black bars. Fig. 6.9(b) shows greater dark space and thus resource utilization than Fig. 6.9(a), representing the greater overlap of tasks on GPUs and CPUs due to workload sizing. Both have higher resource utilization than configurations in Fig. 6.8(a) and (b) due to concurrent CPU and GPU usage.

Fig. 6.10 shows the scalability of hybrid workflows with load balance enabled and up to 22640 tasks on 128 compute nodes on Summit. The left two panels show the comparison between 5660 GPU tasks and 5660 heterogeneous tasks (5400 GPU tasks + 260 CPU tasks). Note that RCT overhead is invariant between homogeneous and heterogeneous task placements and with proportionately increasing workloads and node counts (i.e., weak scaling).

In Figs. 6.8 and 6.10, RCT overhead varies from 3.8% to 11.5% of TTX but it should

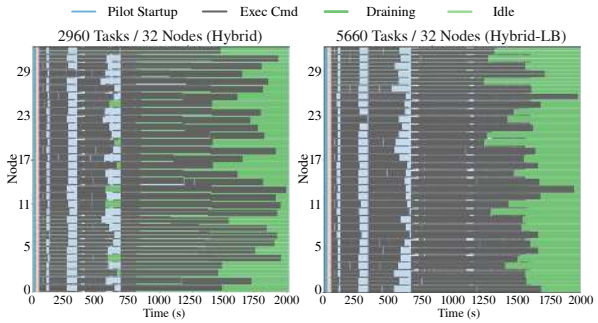


Figure 6.9: Timeline of RCT resource usage for Hybrid Workflows.

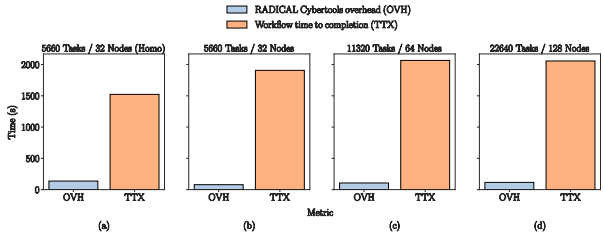


Figure 6.10: RCT Overhead in Hybrid Workflows at Scale.

be noted that task runtimes for these experiments are significantly shorter than those of production runs. RCT overhead arises from state transitions and data movements and is essentially invariant of task runtimes, which are reproduced with fidelity in our experiments. Thus, in production runs, RCT overhead is a significantly less proportion of TTX.

6.5.4 WF3–4: Enhancing Scale and Reliability

Driven by science results, and that WF3 & 4 are the "slowest" per ligand [Saadi et al., 2021], we need to increase the number of nodes and improve reliability across multiple platforms. We preview results for WF3; experience with WF4 and hybrid WF3–4 execution will be reported subsequently.

We run WF3 on 1000 compute nodes (+1 node for RCT), executing 6000 1-GPU tasks on 32 concurrent DVMs. Each DVM spawned ≈32 nodes and executed up to 192 tasks. Fig. 6.11 shows the utilization of the available resources across different stages of the execution. The pilot startup time (blue) is longer than when using a single DVM [Turilli et al., 2019b], mainly due to the 336 seconds spent on launching DVMs which, currently, is a sequential

process. Each task requires time to prepare the execution (purple), which mainly includes scheduling the task on a DVM, constructing the execution command, and process placement and launching by the DVM. The scheduling process takes longer than with a single DVM as it requires determining which DVM should be used. Further, constructing the execution command includes a 0.1s delay to let DVM finalize the previous task launching. As each operation is done sequentially per RP executor component, the 0.1s delay accounts for 600s alone.

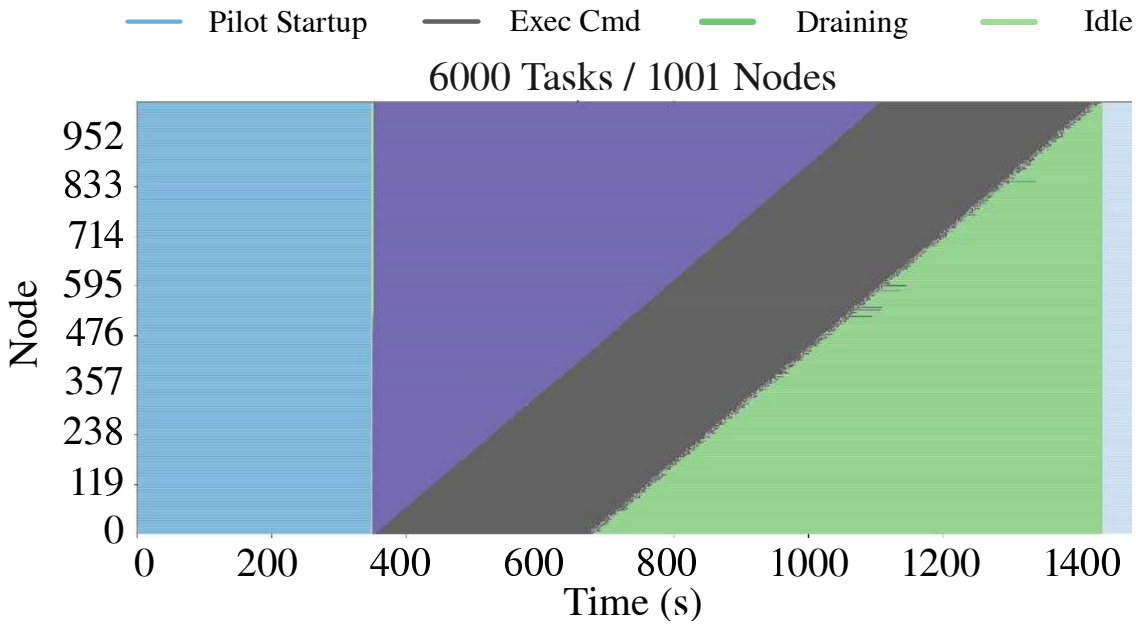


Figure 6.11: Timeline of RCT resource usage for WF3 using multi-DVM.

As with the other WF3–4 experiments, we reduced task runtimes to limit resource utilization while faithfully reproducing RCT overhead. In Fig. 6.11, Exec Cmd (task runtime) would be ten times longer for a production run. Thus, the overheads introduced by using multiple DVMs would have a lesser impact on the overall resource utilization.

We also run WF3 on 2000 compute nodes (+1 node for RCT), doubling the task and DVM number compared to the run with 1000 nodes. At that scale, we observed three main issues: (i) DVM startup failure; (ii) an internal failure of PRRTE; and (iii) lost DVM connectivity. The majority of tasks were successfully completed (11802 out of 12000), but

Table 6.2: WF3 use case. Test runs with RP/Flux integration

Use Case	Platform	# Nodes	# Tasks	# Failed Tasks	Flux Resource Utilization	Task Scheduling Rate	Execution Time
WF3	Lassen	128	512	0	88%	14.21 t/s	6m

those issues prevented RCT from handling their termination gracefully.

Table 6.3: HPC platforms used for the computational campaign. To manage the complexity arising from heterogeneity within and across platforms, requires middleware abstractions and design.

HPC Platform	Facility	Batch System	Node Architecture CPU	GPU	Workflows	Max # nodes utilized
Summit	OLCF	LSF	2 × POWER9 (22 cores)	6 × Tesla V100	WF1-4	2000
Lassen	LLNL	LSF	2 × POWER9 (22 cores)	4 × Tesla V100	WF2,3	128
Frontera	TACC	Slurm	2 × x86_64 (28 cores)	—	WF1	7650
Theta	ALCF	Cobalt	1 × x86_64 (64 cores)	—	WF1	256
SuperMUC-NG	LRZ	Slurm	2 × x86_64 (24 cores)	—	WF3-4	6000 (with failures)

6.6 Conclusion

In this chapter we demonstrate the scalability of heterogeneous workflows using stages of increasing computational cost and increasing accuracy with respect to binding free energy.

CHAPTER 7

SAMPLING STRATEGIES FOR CHEMICAL SPACE

This chapter will address the bootstrapping and sampling problem: how do you select what subset of molecules to get an initial dataset for modeling building on? Even if you have a model built, how do you sample chemical space given you cannot access the whole thing? This chapter will present the idea of using a hypergraph along different chemical-theoretic axes such as scaffolds and pharmacophores. This chapter will comment on the use of LLMs for storing this graph and accessing it on the fly.

7.1 Sampling Chemical Space

Molecules have seemed like a natural fit to deep learning's tendency to handle a complex structure through representation learning, given enough data. However, this often continuous representation is not natural for understanding chemical space as a domain and is particular to samples and their differences. First, we will explore current applications of off-the-shelf deep learning techniques for sampling a model trained on chemical representations to generate novel structures.

7.1.1 Generative drug design

Deep learning (DL) offers a new set of tools and algorithms for generating novel molecular pieces. With the introduction of generative models which can be sampled, such as variational autoencoders [Kingma et al., 2014], or generative adversarial networks [Goodfellow et al., 2014], de-novo molecular generation took hold as a practice in drug discovery [Olivecrona et al., 2017]. Molecules were embedded into a continuous representation and then given a decoder, sampled from continuous space—allowing property optimization and molecular generation based on some distance metric in the latent representation. These approaches

have had much success. We extend on this work by focusing on computational organization and enumeration specifically—seeking more structure than $\mathcal{N}(X, \Sigma)$ or $\mathbb{R}^n$.

7.1.2 Challenges

Given the significance and applications of molecular design, several approaches have explored how to organize the chemical design space, including the use of strings (e.g., SMILES), molecular/chemical descriptor data (e.g., Modred features), and molecular graphs for both representing and generating new molecular designs. Each representation presents some opportunities and challenges in capturing the complexity of the chemical landscape and has successfully designed new molecules for drug targets and new catalysts, and other materials. However, no single representation can sufficiently capture the diversity (in chemical species) and the statistical diversity in their representations. For e.g., while SMILES strings provide a convenient means to encode chemical information, two nearly identical molecules can have significantly different SMILES representations, presenting unique challenges when embedding them into a latent manifold. Similar observations can be made for other molecular representations.

Consider the set of drug-like molecules $\mathcal{M}$. $\mathcal{M}$ is not directly computable as it is a concept class for *molecules*. For computation, molecules require a computable representation, and this is the start of the difficulty. Representations are models of molecules which can be identified with a molecule. Graphs are a natural model of molecules, where nodes are atoms and edges are vertices Kearnes et al. [2016]. SMILES are another representation of molecules, which are a breadth-first search over the graph in a particular syntax. SMILES, unlike graphs, are not injective over molecules (if two SMILES strings are not equal, it does not imply the underlying molecules are not equivalent) O'Boyle [2012]. There are other representations which are less common such as point clouds, junction trees, or voxelization Elton et al. [2019]. We define R_X to be a general representation mapping from molecules to

some set X from $\mathcal{M}$.

Embeddings are distinct from representations. Embeddings are functions which take a representation X to embedding space Y. For instance, molecular fingerprints are an algorithm which takes graphs of molecules to $\mathbb{R}^n$ by utilizing a hashing function around the nodes or regions of a graph Stepišnik et al. [2021]. Node2vec models take graphs to $\mathbb{R}^n$. A simple variational autoencoder's encoder can take SMILES to a Gaussian unit ball $\mathcal{N}(X, \Sigma)$. The junction tree variational autoencoder takes a junction tree to a latent unit ball. In the later two examples, the idea of sampling from a normal unit ball is essential for maintaining the density of the sampling space—an important aspect of creating a generative model (see SI section 2 on sampling). Given a decoder, these embedding spaces can be sampled to produce potentially new molecules or molecules through a constrained optimization problem. The two embedding spaces so far have convenient distance metrics, denoted δ_Y.

While the generation methods in the previous subsection are very successful at certain property predictions and general optimization, they do not solve the enumerability problem. Both $\mathbb{R}^n$ and $\mathcal{N}(X, \Sigma)$ are continuous and not countable. In particular, every molecule has an open ball around it in embedding space of equivalent points which is a problem for enumerating discrete sets of molecules. In other words, if φ^{-1} is a decoder from an embedding $\mathbb{R}^n \to X$, and $\equiv$ is an equivalence relation on the representation X, there exists $y_1, y_2 \in \mathbb{R}^n$ and $\epsilon > 0$ such that $0 < \delta_{\mathbb{R}^n}(y_1, y_2) < \epsilon$ so

$$\varphi^{-1}(y_1) \not\equiv \varphi^{-1}(y_2).$$

In order to structure the embedding space to be conducive for enumeration, we must find an embedding space that is countable and discrete, just as Caley sought out by means of a tree.

7.2 An Algebra for Chemical Space

We focus on exploring a natural structure for representing chemical space as a structured domain: embedding drug-like chemical space into an enumerable hypergraph based on scaffold/fragment classes linked through an inclusion operator.

7.2.1 Motivation

Given the vastness of drug-like chemical space, how can we computationally explore it? In 1875, Caley published a short note on his enumeration of alkanes utilizing a tree structure Cayley [1875]. Though Caley's enumeration ended up having a few errors, it is a very early account of treating chemical space as a structured mathematical object Rains and Sloane [1999]. Over 100 years later, the ideas of enumerating structurally similar compounds and comparing their activity became known as quantitative structure relationship studies (QSAR/SAR). QSAR/SAR is the standard method in medicinal chemistry for taking an interesting chemical compound to an optimized and potent drug lead. In 1984, Klopman developed Computer-Automated Structure Evaluation (CASE), which "perform[s] automatically all operations related to the structure-activity analysis" Klopman [1984]. A success in its own right, CASE utilized the graph topology of molecules to generate QSAR studies or predict activity based on fragments. This graph structure naturally leads to studying subgraphs and their relations, such as decomposing the graph into a class of similar molecules sharing a framework (scaffold), linkers connecting rings, and sidechains Bemis and Murcko [1996]. Utilizing these ideas, various tool-kits and genetic algorithms have been designed to combine or grow molecular fragments into optimized drugs Lameijer et al. [2005], Cernak et al. [2016]. While these ideas in organization lay the framework for certain practices of medicinal chemistry, the methods do not address the problem of enumerating compounds in an organized way to find diverse chemical scaffolds.

A reemerging principle in small molecule-based property prediction models is the similar

property principle Johnson and Maggiora [1990]. Recall from chapter 2 this is a fundamental principle in medicinal chemistry. This principle has been widely applied in the context of determining quantitative structure-activity relationships (QSAR) in medicinal chemistry: how compounds and their activities (against a drug target) can potentially improve (or degrade) based on modifying certain chemical scaffolds (or addition/deletion of R-groups) Maggiora and Shanmugasundaram [2011], Guha [2011].

Molecular scaffolds are well defined through algorithms, decompose well into networks, and offer a general description of global properties (such as orientation in a protein binding region) Bemis and Murcko [1996], Scott and Chan [2020a]. Molecular scaffolds represent the core of a molecule, typically defined around the number of rings in the structure. Non-ring structures in molecules include linkers and sidechains which get collapsed in this representation to a single scaffold representative. In figure 7.1, we show a molecular scaffold decomposing into smaller scaffolds. In this way, we can take a graph or SMILES representation of a molecule and map it to this discrete embedding structure. The mapping into the scaffold structure is unique. As other authors rely on decoders to decode the embedding space, we will rely on decoders to sample the scaffold for the variety of molecules a part of it.

7.2.2 Orders

In this note, we explore a strange structure developed from a formal theory of molecules. The general setting is a well defined structure over sets of a domain, a well defined lift operator, but no downward operator besides statistical sampling in sets.

Consider a collection of sets, $\mathcal{P} = \{P_i\}$, from a universe X.

Definition 7.2.1 (Partial ordering)**.** A partial ordering $\preceq$ is a relation over a set, X, with the following properties:

1. (Reflexive) $x \leq X$, $\forall(x \in X)$.

2. (Antisymmetric) If $x \preceq y$ and $y \preceq x$ then $x = y$, $\forall(x \in X, y \in X)$.

3. (Transitive) If $x \preceq y$ and $y \preceq z$ then $x \preceq z$, $\forall(x \in X, y \in X, z \in X)$.

A *partially ordered set* (poset) is a set X with a partial ordering. We say two elements, say x and y, are not *comparable* if neither $x \preceq y$ or $y \preceq x$. If $\preceq$ extends over the whole set such that there are no non-comparable elements then $\preceq$ is a total or linear order. When $\preceq$ is total, X is a totally ordered set also called a *chain*. An element a is said to by b is $a \preceq b$, $a \neq b$, and there does not exist a distinct c such that $a \preceq c \preceq b$.

The most common example is the standard relation $\leq$ on the natural numbers (which is of course a total ordering). The following illustrates ordering sets of sets.

Example 7.2.2. Let $X = \{a, b, c\}$. Then $\mathscr{P}(X)$ is a partially ordered by inclusion. $\{a, b\} \subseteq \{a, b, c\}$ but $\{a\}$ is not comparable to $\{b\}$.

Definition 7.2.3 (Upper bound)**.** Let $Y \subseteq X$. An *upper bound* of Y is an element $x \in X$ such that for all $y \in Y$, $y \preceq x$. An upper bound s of the set Y is an *least upper bound*, *join*, or *supremum* if for all upper bounds u of S, $s \preceq u$.

The dual definition is stated for clarity, but should be clear by switching orderings in the previous definition.

Definition 7.2.4 (Lower bound)**.** Let $Z \subseteq X$. A *lower bound* of Z is an element $x \in X$ such that for all $z \in Z$, $x \preceq z$. An upper bound s of the set Z is a *greatest lower bound*, *meet*, or *infimum* if for all lower bounds u of S, $u \preceq s$.

Armed with the basics of ordering, we can start to inspect some ordered structures.

Definition 7.2.5 (Semilattice)**.** A partially ordered set $(X, \preceq)$ is a *upper semilattice* if every two-element subset has a join, or least upper bound. $(X, \preceq)$ is a *lower semilattice* if every two-element subset has a meet or greatest lower bound.

Definition 7.2.6 (Lattice)**.** A partially ordered set $(X, \preceq)$ is a lattice if it is both an upper and lower semilattice. A bounded lattice has a greatest element and a least element, sometimes represented by 1 and 0 respectively such that $0 \preceq x \preceq 1$ for all $x \in X$.

Definition 7.2.7 (Atomic)**.** Let L be a lattice with bottom element 0. An element $x \in L$ is an *atom* if $0 \prec x$ and there exists no element $y \in L$ such that $0 \prec y \prec x$. L is *atomic* if for every nonzero element $x \in L$, there exists an atom a of L such that $a \preceq x$. L is *atomistic* if every element of L is a supremum of atoms.

Definition 7.2.8 (Graded lattice)**.** A lattice $(L, \preceq)$ is called *graded* if there exists a function $r : L \to \mathbb{N}$ such that

1. r is compatible with the ordering such that $r(x) \preceq r(y)$ whenever $x \preceq y$
2. r respects covers, which is to say if y covers x then $r(y) = r(x) + 1$. The value of the rank function for an element is called its rank.

7.2.3 Scaffold math

The conceptual machinery for treating chemical space as a hypergraph structured through scaffolds is developed. There is an elegant statement of the principle of fragment-based drug design through the operations among scaffolds. Further, the framework developed provides intuitive concepts for understanding the diversity and size of chemical space explored or discussed by a model or computational research program. As a computational learning problem, we use transformer as seq2seq models to implement large graph navigation in practice.

7.2.4 Scaffold Embeddings

Utilizing the concept of scaffolds developed in section 2, we assume the operation `Scaffold` as a given oracle such that `Scaffold` is injective and defined for every molecule. We define

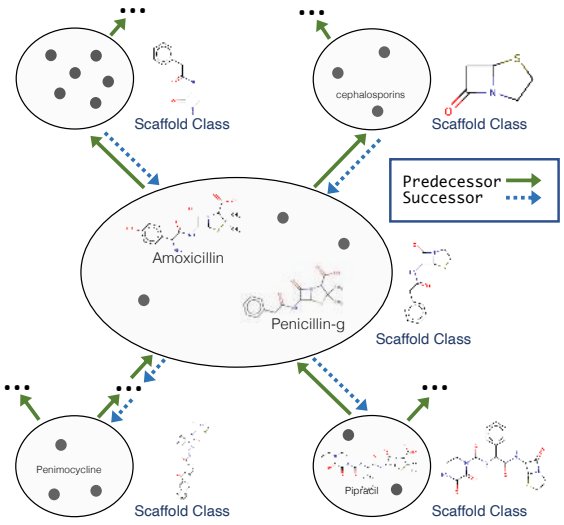


Figure 7.1: **Scaffold as classes over drug-like chemical space.** Every molecule (represented by dots or depiction inside circles) is inside a single scaffold class. Scaffold classes are related through common substructures, forming a hierarchy of classes. Penimocycline, for example, belongs to a scaffolding class from far Penicillin-g's or Amoxicillin's class, while Pipracil is a direct successor of the Penicillin-g class. The `Predecessor` function is defined via an algorithm, and the $\texttt{Successor}_\Phi$ function requires a generative model when working without data (i.e., given a single scaffold you cannot compute its successor unless you understand chemistry, thus have parameters Φ, but you can compute all of it is predecessors recursively without knowing how to generate new compounds).

$\mathcal{S}$ as the set of all scaffolds.

A hypergraph is a generalized graph where edges group more than two vertices. A hypergraph is n-regular when every vertex is contained in exactly n edges. Scaffolds as hypergraph edges over molecules form a 1-regular graph, as every molecule belongs to exactly one scaffold class, thus every vertex has degree 1 in the hypergraph. We denote the hypergraph as $\mathcal{H} = (\mathcal{M}, \mathcal{S})$.

Operations on scaffolds. We denote computational operations in `Monospace` font, and add a subscript Φ to represent parameters which may be required for the operations (i.e. $\texttt{Expand}_\Phi$).

1. $\texttt{Expand}_\Phi$ and `Scaffold`: Molecules and scaffolds represent two distinct types which can be converted back and forth (figure 7.2). Scaffold classes can be *expanded*, where

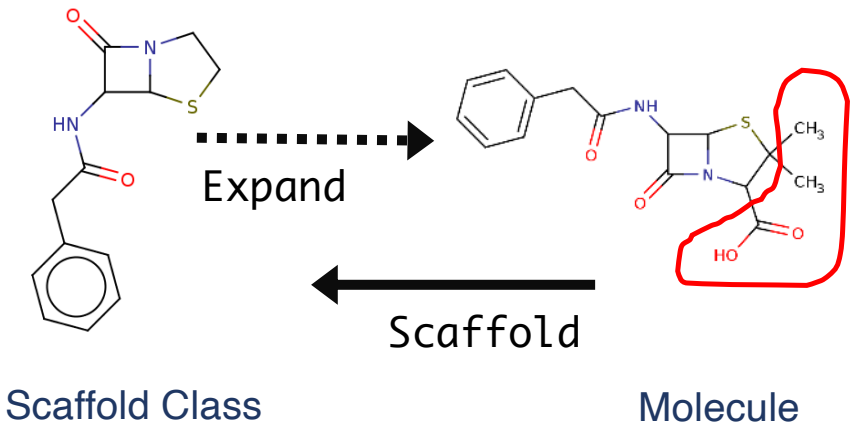


Figure 7.2: **Scaffold and molecule relation**. Scaffolds are the core or framework of a molecule, and they represent a class of molecules. Scaffolds, or scaffold classes as we often refer, group molecules together. A class can be extended by adding decorations to the scaffold, such as linkers and sidechains. Through the scaffold function, we obtain the scaffold of a molecule.

we envision zooming in, via the $\texttt{Expand}_\Phi$ model (i.e. $\texttt{Expand}_\Phi$: $\mathcal{S} \to \mathcal{M}$). Similarly, molecules can be taken to their scaffold via the program `Scaffold` (`Scaffold`: $\mathcal{M} \to \mathcal{S}$). We utilize RDKit to compute `Scaffold` via the MurckoScaffold module Landrum et al. [2016]. We note a model can be trained for this task; however, given the efficiency of the algorithm it did not seem fruitful at this time.

2. $\texttt{Successor}_\Phi$ and `Predecessor`: the successors of a scaffold S_1 are the set of all scaffolds S which contains S_1 as a substructure (figure 7.1). The predecessors of a scaffold S_1 are all scaffolds S which S_1 is a superstructure. In general, there is no algorithm for successor given only a scaffold, as it requires sampling chemical space. However, predecessor has an efficient algorithm with a structure that can always be fragmented into smaller scaffolds without sampling other data. These operations are the atomic building blocks of navigating between scaffold classes (and induces a strict partial ordering $(\mathcal{S}, \prec)$). These operations are from $\mathcal{S}$ to $\mathcal{S}$. We also consider the standard graph structure induced by the relation $\texttt{Successor}_\Phi$ and `Predecessor`, and denote it $\mathcal{S}_\mathcal{G} = (\mathcal{S}, \texttt{Successor}_\Phi)$ where $\texttt{Successor}_\Phi$ can be used to determine the edge relation.

This graph can be directed or undirected, but for our case we consider the undirected graph mostly.

3. $\texttt{Union}_\Phi$ and `Intersection`: two scaffolds S_1 and S_2 can be combined to form a union. More formally, the union of S_1 and S_2 is the set of scaffolds that contain S where S has S_1, and S_2 has immediate predecessors. Similarly, the intersection of S_1 and S_2 is simply the maximum common substructure (MCS) of S_1 and S_2, for which an efficient algorithm exists for small drug-like molecules. Cao et al. [2008], Cone et al. [1977]. In general, MCS is NP-complete, but there are heuristics for drug-like molecules that provide a rather efficient algorithm Garey and Johnson [1979]. These operations are from $\mathcal{S}$ to $\mathcal{S}$.

These basic operations can be combined into more complex operations such as

$$\texttt{UpperCone}_\Phi(S) = \{A : S \prec A\} \tag{7.1}$$

$$\text{or} \quad \texttt{LowerCone}(S) = \{B : B \prec S\} \tag{7.2}$$

Upper cones of scaffold classes are actually a common object of interest for drug discovery. For instance, Penimocycline is in the upper cone of Penicillin-g's scaffold class (see figure 7.1). Successful exploration of upper cones is the theoretical cornerstone of fragment based drug design Schiebel et al. [2016], Murray and Blundell [2010]. Recently, fragment X-ray crystalgraphic screens have been performed on important drug targets such as SARS-CoV-2 proteases in search of an inhibitor Douangamath et al. [2020]. Given a set of fragment hits for a protein target in a binding region,$\{m_i\}_{i\in H}$, take the scaffold classes of those hit, $\{S_i^h\}_{i\in H}$. The principle of fragment based drug design can be expressed as there exists some index set I^* such that $I^* \subseteq H$ and

$$\hat{H} = \bigcap_{i\in I^*} \texttt{UpperCone}_\Phi(S_i^h) \tag{7.3}$$

where $\hat{H}$ is a set of scaffold classes, $\hat{H}$ is not empty, and some molecule in a scaffold in $\hat{H}$ is a likely candidate. In other words, a set of fragments can be grown to sets of larger drug-like molecules, and some intersection of those possible larger molecules will be a hit that is likely a drug lead for this protein target. In an embedding space such as $\mathbb{R}^n$, the same principal does not apply, and is dependent on the embedding context (for instance, based on a particular property Iovanac and Savoie [2019]). Furthermore, there no guarantees about molecules in an interval between two molecules, whereas the intersection of upper cones, for example, does have such guarantees (if it is not empty).

Given there is no algorithm for producing successor scaffolds without relying on sampling chemical space, we treat the problem as a learning problem. We note that we cannot rely on fragments as a vocabulary given this construction as other methods have (for instance, Jin et al. [2018b] utilized a finite vocabulary containing one member rings, linkers, and sidechains from the dataset). When using such a vocabulary, there are chains of scaffolds that cannot be represented as the $\texttt{Successor}_\Phi$ function can only sample scaffold classes S for which every one ring member in the $\texttt{LowerCone}(S)$ is in the finite vocabulary.

7.3 Modeling Chemical Space with Transformers

While the method outlined has no constraints on compounds' synthetic accessibility, it is a necessary and essential aspect of chemical space exploration for drug discovery. To focus on synthetic accessibility while paying attention to maximizing library size, we utilize a dataset from Synthetically Accessible Virtual Inventory (SAVI) Patel et al. [2020]. SAVI contains over 1.7 billion reaction products (along with rich reaction and metadata). We utilize only the SMILES of the products.

We build two datasets from SAVI. The first utilizes RDKit to determine the scaffold for each of the compounds listed Landrum et al. [2016]. We utilized a 200M sample from the entire dataset and extended the data by a factor of 5 by randomizing the SMILES

both for the target (scaffold) and source (molecule) Arús-Pous et al. [2019]. A set of 20M molecules with a unique scaffold class are held out as validation data. A second dataset is created by taking a subsample of the prior dataset, 20M, and utilizing the ScaffoldGraph package to decompose each scaffold into a network of scaffolds Scott and Chan [2020a]. We sample edges (representing the successor of two scaffold nodes), resulting in a dataset of five million successor pairs. This dataset is extended to 50M utilizing random smiles sampling. Predecessor data is flipping the columns (sources become targets, and targets become sources) for the successor datasets.

On the one hand, $\texttt{Successor}_\Phi$ and $\texttt{Expand}_\Phi$ are generative models—given a scaffold, those operators are required to sample the space of successor scaffolds or molecules that have that scaffold. On the other hand, they are seq2seq task, taking one sequence to a different sequence. This combination of wanting a dense sampling strategy combined with seq2seq modeling differs from applications we have found in the literature. Common approaches to generative models have been utilizing VAEs or GANs to train some encoder-decoder model on sample reconstruction error with some regularization Elton et al. [2019], Gupta et al. [2018], Grisoni et al. [2020]. Seq2seq approaches in this space have focused on solving problems with a relatively small optimal solution set such as reaction modeling Schwaller et al. [2019]. With the recent success of transformer models performing well on large datasets and seq2seq problems, we decided to follow the modeling as a seq2seq problem as Schwaller et al. have.

We utilize a transformer seq2seq model from the ONMT project Klein et al. [2017]. Other works have utilized RNNs, but we utilize a transformer for both the encoder and decoder of the model Vaswani et al. [2017b]. Given the goal of not simple generation but rather generalizing a very large hypergraph for which a pure algorithmic solution is intractable, transformer models are a good fit compared to simpler RNN models. Code is compiled into a GitHub repository with scripts for data gathering, data preparation, model training, and sampling. The interface is geared towards developing front-end functions for quick medicinal

chemistry questions regarding sampling molecular space.

7.4 Experiments

7.4.1 Computability of Scaffold Classes

We assess the structure of *scaffolding* chemical space, focusing on understanding the size of scaffold classes, how many scaffold groups there are in drug-like chemical space, and how they connect.

We impose a structure on $\mathcal{M}$ by creating scaffold classes $\mathcal{S} = \{S\}_{i \in I_s}$ such that every molecule m belongs to one and only one scaffold class, and all classes in $\{S\}_{i \in I_s}$ are disjoint. We also assign a hierarchy to scaffolds based on the number of rings. $\mathcal{H}_n$ is the set of all scaffold classes with ring size n.

$\mathcal{H}_0$ is the smallest hierarchy, which consists of only one scaffold class S_0, the set of all molecules with no rings (ring-less fragments, linkers, and side-chains). $\mathcal{H}_1$ is the set of all scaffold classes with one ring. The order of $\mathcal{H}_2$ is proportion to $|\mathcal{H}_1|$ choose 2 plus the combination of linkages and sidechain modifications from $\mathcal{H}_0$. We see growth similar to the partition function in theory. However, in practice, the distribution of molecules in real-world datasets typically follows a normal distribution with the mean around three rings (see figure 7.3).

Given this added structure of scaffolds, do scaffolds reduce the search space over molecules by many magnitude orders? If this is the case, we can search through a computable number of scaffolds, and once a few interesting classes are found, we can enumerate the molecules in that set. This strategy does not face the curse of 10^{68} drug-like molecules the current unstructured domain $\mathcal{M}$ faces. Given a 200M sample from SAVI, we found only 11.4M (5.7%) scaffold classes were needed to cover the entire dataset, and, in practice, there exists a large subset of molecules (165M) with only 685,000 (0.41%) scaffold classes. This reduction via scaffolds

Model	SMILES Validity	Type Accuracy	Correctness Accuracy
$\texttt{Successor}_\Phi$	98.9%	98.9%	97.9%
`Predecessor`	99.8%	99.8%	94.0%
$\texttt{Expand}_\Phi$	98.6%	-	96.9%

Table 7.1: Performance metrics from graph navigation models. Evaluations were performed with a holdout set from SAVI dataset. SMILES validity is the percent of samples that pass an RDKit parser. Type accuracy determines how many samples have the correct type (`Successor`, `Predecessor`, and `Union` models output type scaffold. In contrast, `Expansion` model outputs molecules (which can include a scaffold representative, and this metric is left out and computed as a part of correctness). Correctness accuracy is the percent of samples which are valid, typed correctly, and are equivalent to the algorithmic solution.

implies for a large subset of molecules, there is a reasonable 5 order of magnitude gain in search over scaffolds than pure molecules (from a database or chemical library perspective).

7.4.2 Hyerpgraph Navigation

We train three operations ($\texttt{Expand}_\Phi$, $\texttt{Successor}_\Phi$, `Predecessor`) utilizing three separate models. While there is an algorithm for `Predecessor`, we can compare it directly to the algorithm performance. Each model was trained for approximately two days on eight GPUs (NVIDIA Tesla V100). Each model was trained for 500,000 steps with a batch size of 8192. Further details of the training procedure can be found in (SI) and on GitHub.

To sample $\texttt{Expand}_\Phi$ and $\texttt{Successor}_\Phi$ we utilize beam search with a temperature of 1.5, beam size of 5, and randomizing the SMILES input. Samples are then validated utilizing RDKit. In table 7.1, we outline each model's accuracy. A uniform sample of scaffolds from the validation data was taken ($n = 1000$), and 100 samples were drawn for each scaffold class (figure 7.4).

Given the density of some scaffold classes in the data compared to others (figure 7.3), more advanced sampling methods required for $\texttt{Expand}_\Phi$ on these classes. For scaffold classes with over 10^6 members in the data (mostly 1-ring and 2-ring common scaffolds), resampling

Scaffold	Class Size (Data)	Unique Sampled	Overlap (Recall)
c1ccc(COc2ccccc2)cc1	373,939	168,261	4,146 (1.1%)
O=S(=O)(c1ccccc1)N1CCCCCC1	88,608	145,904	20,097 (22.7%)
O=S(=O)(NCCc1ccccc1)c1ccccc1	911,360	176,539	23,715 (2.6%)
c1ccncc1	818,230	183,838	23,999 (3.0%)
O=S(=O)(NS(=O)(=O)c1cccnc1)c1ccccc1	203,891	173,599	20,331 (10.0%)

Table 7.2: **Sampling dense classes with** $\texttt{Expand}_\Phi$. Five dense scaffold classes were taken from the validation data and sampled. We sampled 100,000 times for each scaffold, utilizing a temperature of 1.5 and a beam search of length five and capturing the top two best beams from the search. While we do not capture a large set of the data, we believe these classes' sheer size presents a combinatorics problem. The unique samples are all correct and valid.

validation data from the model is difficult (table 7.2). Given the uniqueness of sampling based on a category like scaffolds, rather than pure sampling points in a distribution or $\mathbb{R}^n$, comparisons to generative models' reconstruction accuracy are not reasonable.

Figure 7.5 is an example of a series of compounds which belong to a single scaffold class, but are sampled with different sidechains. The variety of sidechains while maintaining the single scaffold core is the basis of a QSAR series.

7.5 Sampling with a graph

7.5.1 Fast Docking with Random Walks

In order to sample from this graph to actively decide which molecules should be docked next, we need to carefully define how docking results will be applied to the sampling weights. There are various strategies for sampling nodes from a graph. The simplest strategy is random node sampling; however, this sampling strategy would not take advantage of our graph structure and would simply be the same as naive random screening molecules from a database. A more complicated strategy which incorporates the graph structure is random walks.

Random walks are a stochastic process which defines a technique for taking a series of steps over a set. For example, a random walk over a graph (with nodes and edges) implies

picking a starting node and randomly stepping in sequence from the starting node to one of its neighbors, then to one of the new node's neighbors, and so on and so forth. Random walks can be used as a sampling technique on graphs, where the samples are drawn by picking the nodes for which a random walker "walks" on.

The two essential pieces of information to define a random walk strategy is a starting node and a selection strategy to go from the current node to the next node. Given a graph $G = (V, E)$, where every $v \in V$ is a scaffold or molecule, we define a random walking strategy by setting the starting node to a random molecule and setting the walking strategy to be random half of the time and to select the neighbor with the best docking score the other half of time. In other words, half of the time the next molecule chosen will be a random neighbor of the current node, and the other times it will be a better scoring molecule. The use of both random walks and a particular objective function, in this case optimizing the docking score, is known as mixing. By utilizing mixing with the known docking scores we not only incorporate the graph structure we impose into the sample strategy, but we also incorporate structural information about the ligand into the search strategy. The goal is to avoid regions of chemical space which certainly would not contain hits (i.e. scaffolds which are merely the wrong shape for a particular binding pocket).

Let $\Phi_\theta : \mathcal{M} \to \mathbb{R}$ be a function mapping molecules to docking scores. Let $G_S = (V, E)$ be a molecular scaffold graph and let $H : V \to \mathcal{M}^<$ be a set function to produce the set of molecules that correspond to the scaffold from the data. The *lower cone* of a scaffold $s \in V$ is a set $L_s \subset \mathcal{M}^<$ and defined as

$$L_s = \{m : \text{t is a successor of s and } m \in H(t)\}. \tag{7.4}$$

Where a successor to a scaffold s is defined as a scaffold in a higher hierarchy level (where higher levels contain more rings) with molecules that contain the structure of scaffold s. Informally, a lower cone is the set of successor classes from any given scaffold. Shown in

Fig. 7.6 is a visualization of how we conceptualize these cones in the chemical space of scaffold classes. The lower cone L_s is a set essentially containing all molecules which share a common scaffold fragment, s. For each scaffold landed on during the random walk, we can sample either the molecules which strictly decorate that scaffold or we can consider all molecules which contain that scaffold (i.e. including larger scaffolds as well). In the result section, we show data utilizing the latter cone based conception given often times in drug discovery one fixes a particular scaffold component (such as an RNA-analog for proteases) and wants to find molecules which contain such structure.

Learning on Hypergraphs

Given a graph $G = (V, E)$ let F_V be node features so V and F_V have same length, which means each row of F_V corresponds to a feature vector. Let $L \subseteq V$ be the subset of labeled nodes and F_{VL} be the corresponding feature matrix. Let $U \subseteq V$ be the subset of unlabeled nodes and F_{VU} be the corresponding feature matrix. We focus on regression tasks, where our goal is to make real-valued predictions of a given property for the unlabeled nodes U. Our algorithm first makes a baseline prediction with a neural network – this step does not make any use of the graph G. Then we correct these predictions by propagating the residuals over the structure of G. In this step we assume graph smoothness and positive correlation of residuals along edges of the graph. This assumption of smoothness is by no means a guarantee for properties such as docking score over the structure of a scaffold graph. Instead, we hypothesize that the correctness of our final predictions for the nodes in U using this method can confirm the validity of the assumption for a given property. The algorithm is outlined in Alg. 1: (i) Train a simple multi-layer perceptron (MLP) neural network M on the subset L, using feature matrix F_{VL} and labels y_L. Make predictions on the whole graph $\hat{y} = M(F_V)$. (ii) Estimate the residuals for the unlabeled data r_U^{LP} using label propagation with the labeled residuals $r_L = y_L - \hat{y}_L$. Correct the predictions for the unlabeled nodes.

This algorithm is taken from Jia and Benson [2020], which demonstrates the effectiveness of label propagation with residuals on several regression tasks.

Algorithm 1: Label Propagation with Residuals

input : Subset L of labeled nodes, $\hat{y}$ predictions from model trained on L

output: $\hat{y}_U^{LP}$, smoothed predictions from LP on residuals

$r_L \leftarrow y_L - \hat{y}_L$;

$\hat{r}_U^0 \leftarrow \mathbf{0}$; `// set initial guess for unlabeled residual's`

$\mathcal{L} \leftarrow \boldsymbol{I} - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$; `//` A `and` D `are the adj. and deg. matrices`

$\hat{r}_U^{LP} \leftarrow$ `ConjugateGradient` $(\mathcal{L}_{UU}, \mathcal{L}_{UL} r_L, \hat{r}_U^0)$;

$\hat{y}_U^{LP} \leftarrow \hat{y}_U + \hat{r}_U^{LP}$

By decoupling feature-based learning and graph learning, we avoid the computational cost of training neural networks on large graphs while still benefiting from the inherent graph structure in our datasets using fast learning methods like label propagation. Our method for learning combines work from Huang et. al Huang et al. [2020], which uses a multi-step "correct and smooth" after a base prediction from an MLP on classification tasks; and from Jia et. alJia and Benson [2020], which corrects using residuals on predictions from a graph neural network. Both show that using an initial base prediction with a model, and then post-processing using label propagation on the graph can boost performance and decrease training time.

Results

7.5.2 Scaffold-space Partitions Virtual Screening Hits

We generate a scaffold network for Mcule's in-stock compound library utilizing the ScaffoldGraph package Kiss et al. [2012], Scott and Chan [2020b]. The Mcule library consists of a diverse set of purchasable compounds within known stock amounts. Overall, the network has 6,369,219 nodes and 8,808,841 edges. The graph networks induced by the standard, drug

screening libraries look as expected–high connectivity nodes include small one or two ring scaffolds with various pharmacophores, while terminal nodes are complex concatenations of the various predecessor graph nodes (Fig. 7.7 left hand panel).

Visualizing a large network with over 6 million nodes can be challenging – hence, we chose a small subset of molecules for analysis (Fig. 7.7 right hand panel). It is interesting to note that while many of the upper-level nodes in the graph (typically small functional groups such as benzene rings) have lower docking scores, successive transformations of these nodes towards the leaf edges have higher docking scores (favorable interactions). This means that traversing the graph from one radial end to the other can result in compounds that successively improve their overall ability to interact with the protein. Molecule nodes are only connected to their respective scaffold nodes, and scaffold nodes are connected based on substructure (i.e. scaffold node A is connected to scaffold node B if A is a substructure of B, and molecule node C is connected to node B if the scaffold of molecule node C is B). In a sense, this is simply a projection of a hypergraph to a regular graph. We assign molecule nodes a docking score based on a docking data set for SARS-CoV-2 main protease (Mpro) Clyde et al. [2021c].

Utilizing the notion of cones derived in the method section, Fig. 7.8A highlights the fact that molecules in the same cone are likely to have docking scores relatively close to each other. For the unstructured dataset, the mean docking score is 8.50 ± 1.05. The standard deviation of these cones is a proxy for the smoothness of the graph with respect to their docking scores. This is a good measure as it implies given another molecule with the same scaffold in that class, it is more likely that the molecule falls within the mean of the cone. The standard deviation of the cones is relatively small compared to the standard deviation of the overall data, and this difference is significant. This indicates that local scaffold neighborhoods are relatively uniform in their overall docking scores. We see that using scaffolds as a way of embedding molecules is natural and matches the intuition.

We explore scaffold-space's relation with docking scores. Using a simple random walk which we outlined in the methods section, we show the strategy is more efficient in finding high scoring hits than random sampling. In Fig.7.8C, the colored lines correspond to the best score seen so far by a random walker. The lowest gray line, for example, shows that after selecting 1000 nodes based on the random walk strategy, the lowest score found is -13.5. Conversely for the random strategy, the histogram in black in the zoomed-in pane shows that the lowest score found randomly was greater than -12.5. Therefore, random walking with mixing in docking scores is more efficient at sampling high scoring compounds in a small docking screen. We show that based on more activated scaffolds improves detection of the best docking scaffold classes over just randomly docking.

Scaffold-space is locally homogeneous with respect to experimental assay data. We illustrate JAK2 kinase data for approximately 3,000 compounds from Laufkötter et al. [2020] in Fig. 7.8B and D. The overall dataset reports p-values (higher indicates more likely inhibition) with a mean of 7.746 $\pm$ 1.03. We see that communities are much smoother than the entire dataset, with an average standard deviation half of the dataset as whole. In the chart of Fig. 7.8B, we see some high scoring communities (high p-values) have rather small standard deviations—this is illustrated in the graph depicted, where red and large nodes have larger p-values than blue and small nodes. We zoom in on a particular set of four communities in Fig. 7.8D, two of which are highly active based on the data and one which is not. We observe that (2), (3), and (5) are scaffolds, where less active molecule (1) has (2) as its scaffold, (3) is a super-scaffold of scaffold (5), and active molecule (4) has (5) as a scaffold. We observe that all molecules in the study which contain substructure (5) are active, while all molecules which have (2) has a scaffold are less active.

Table 7.3: 5-fold cross validation performance of the two step algorithm on test sets of labeled Mcule nodes and docking scores for 3CLPro

Algorithm step	r^2	MAE
MLP Model	0.58	0.55
Label propagation	0.57	0.58

Experimental setup and label propagation to scaffold nodes

We trained a DeepChem Ramsundar et al. [2019] Multitask Regressor network, a feed-forward multi-layer perceptron (MLP), with a 1,000 node hidden layer and ReLU activation function. Our feature matrix F_V consists of 2,048 bit vector circular fingerprints for each molecule in Mcule (F_{VL}) and each scaffold in the generated scaffold graph (F_{VU}). Docking scores for each molecule in Mcule are generated using OpenEye FRED docking tool. The model is trained on the labeled Mcule nodes and is run in inference mode for the unlabeled scaffold nodes. We performed 5-fold cross validation using 80% of the Mcule set as the training set for the MLP model and as the labeled nodes for label propagation using the conjugate gradient method. Results are shown in Table 7.3. We see that in this context label propagation of the residuals does not improve upon initial model performance. There may be several reasons for this, including the fact that our labeled dataset consists of only the outer "ring" of nodes in the scaffold graph (Fig. 7.7) and therefore is not uniformly sampled from the graph. Instead, we note that results in the previous section provide empirical evidence of the smoothness of docking score over communities of scaffolds. Further, we have shown that docking based on active cones can provide the enrichment needed to more effectively sample from databases.

7.5.3 Application to JAK2 Kinase Inhibitor Discovery

This conceptualization of the chemical space for drug discovery has many applications for drug design. Fig. 7.9 illustrates hits utility as a sampling method. For each descendent and

common scaffold, interactions with the protein remain relatively stable. On the contrary, if descending down a certain path in the graph does not have the desired interactions, it is unlikely continuing down the path will yield higher activity. We are currently working on quantifying this graph smoothness property of chemical space in terms of commute and hit times Chennubhotla and Bahar [2007]. This will provide bounds on how well the graph sampling notion outlined in this paper will accelerate giga-docking studies. Separately, by connecting chemical space topologically by scaffold, we are able to perform design as implied by network of residue contacts of docked scaffolds in Fig. 7.9.

7.5.4 Discussion

We outline first the need for a new computational paradigm for cheminformatics workflows. We show how the introduced network based on scaffolds solves the enumeration problem cheminformatics faces. Lastly, we outline the future direction of utilizing scaffolds and networks in drug design.

Computational Structures for Cheminformatics

Current computational approaches to virtual drug screening and design are generally unstructured Ruiz-Torres et al. [2017]. Computational scientists treat molecular space as a database listing. As a result of this simple but useful treatment of chemical space, interaction between the AI/ML community and medicinal chemists has taken off Elton et al. [2019]. This has been helped as well by the AI community's focus on keeping deep learning methods as hands off as possible Zhang et al. [2018a]. But treating chemical space as tabular listings of structures will ultimately fail. The sheer size of chemical spaces makes listing it intractable, let alone performing any type of inference no matter how simple. The database approach to computing on chemicals will not last long without radical changes in computing architectures (inference over 10^{68} is completely intractable). To solve this fundamental problem with the

current methodical theory, the community needs to move towards conceiving computational structures for chemical space which do not require listing and assume relationships between molecules.

Treating chemical space as a network solves the enumeration problem. By connecting molecule space and assuming some properties will be locally specific, nearly all molecules are only 10 hops away from each other. This paper proposes the start of this navigation paradigm, and there is certainly room to improve as we include the other aspects of a molecules' design as stated in Bemis and Murcko [1996] (such as sidechains/electrostatics). Furthermore, the network allows chemical space to be conceived as an inductive database Meo [2005]. An inductive network for chemical space would not require defining the graph at one time in memory, rather it would start from a particular set of molecules, and then utilize operations to grow and shrink the molecule space Clyde et al. [2021f], Li et al. [2019b], Lim et al. [2020]. Furthermore, the global graph representation of chemical space is unique to other sampling methods and molecular generation methods as SIMG focuses on utilizing known compound scaffolds and designs, rather than completely novel chemotypes, and connecting those designs to the data through a graph of molecules. Molecular generation methods generate molecular designs often with novel or synthetically unfeasible chemotypes and scaffolds structures Gao and Coley [2020]. Molecular generation techniques can be connected to the SIMG approach, by placing those novel compounds into a well ordered structured global graph. The SIMG technique does not preclude the use of novel structures, it provides a more ordered approach to integrating those chemical structures into the graph representation of chemical space.

Scaffold-space is smooth with respect to docking and protein-ligand experimental assays

We found docking scores to be locally smooth around scaffolds. Docking scoring functions include shape complementary as a large component of the score. Given that the scaffold of a molecule mostly defines the shape, it makes sense docking scores vary smoothly with respect to scaffolds. In Fig. 7.8D, we see three scaffolds (2), (3), and (4). The coloring is p-values from a binding assay with JAK2. All three scaffolds share a common pyrimidine ring. Scaffold (2) has a *3-pyrimidin-2-ylimidazo[4,5-c]pyridine* three ring system, which is different than the three ring system of (4) and (3). Further, we observe this difference alongside different local assay results. Molecules which have scaffolds (3) and (4) are active, while molecules which contain scaffold (2), such as molecule (1), are more likely inactive or less potent. This matches another study which focused on molecular generation constraining the scaffold class Li et al. [2019b]. They find "the predicted activity against DRD2 for generated samples is significantly higher compared to random sampled molecules from ChEMBL, showing the model is good at utilizing privileged structures to generate bioactive molecules." This corroborates our Mpro docking results, which show that sampling the best seen, or "privileged," scaffolds generates samples with higher activity as judged by docking score. As an observation, we also see that JAK2 kinase assay data forms local communities around scaffolds that are locally smooth. Other work has shown for yet another class of compounds that "the binding mode of the pyrazole carboxylic ester scaffold [to phosphodiesterase] does not change when substituents are added" Jhoti and Leach [2007], Card et al. [2005]. This scaffold-local behavior is well studied, and is the essence of scaffold based drug discovery (SBDD) or scaffold hopping Rabal et al. [2015], Dimova et al. [2017], Lai et al. [2020].

While scaffold hopping alone has been successful, very interesting work has been done with altering the core of a molecule directly to hop from low to high activity regions. One

such paper Gjorgjieva et al. [2016] switches the central core only of two molecular series, maintaining the electrostatics from sidechains, to obtain nM DNA gyrase B inhibitors. Ideally, our notion of chemical space would also include a specification for electrostatics to fully specify a molecule, rather than scaffolds alone. We believe this would more richly specify docking scores as it would account for variance seen by molecules with varying sidechains in the same scaffold class with different docking scores.

Scaffold graphs can have immediate impact on how medicinal chemists utilize computing systems for their work. The presentation of scaffold-space matches current practices in medicinal chemistry for molecular series design Lai et al. [2020]. Future work will aim to extend this concept to complete system for SBDD. For example, users will be able to provide molecular data, and ask for automatic generation of molecular series for experiments based on sampling likely scaffolds. Scaffold constrained generation is not novel, and has been successful in various campaigns such as scaffold based generative models Li et al. [2019b], Arús-Pous et al. [2020].

7.5.5 Conclusions

Our results suggest that a graph-based representation of chemical space libraries offers an automated and tractable approach to scaffold-based drug discovery. Our current results were illustrated on docking data for SARS-CoV main protease, and assay data for JAK2 kinase. For Mpro, we used docking data available to us through large-scale VS approaches to organize the chemical space and identify regions of the chemical space that may potentially lead to new molecules that can inhibit this protein. These approaches suggest the complementary role that graph-based approaches combined with knowledge about targets can be used as a means to capture complex biological (protein-target specific) information and ultimately define methods to model protein-ligand design.

From a computational perspective, we present a technique which solves the enumeration

problem of chemical space. Utilizing a network as an inductive dataset allows exploration of all of chemical space, and only a local neighborhood needs to be stored and queried at a given moment of the design process. We believe this is a step forward towards automated drug design which is integrated with medicinal chemists' workflows and knowledge. We aim to continue pursuing a knowledge based discovery system for scaffold based drug design.

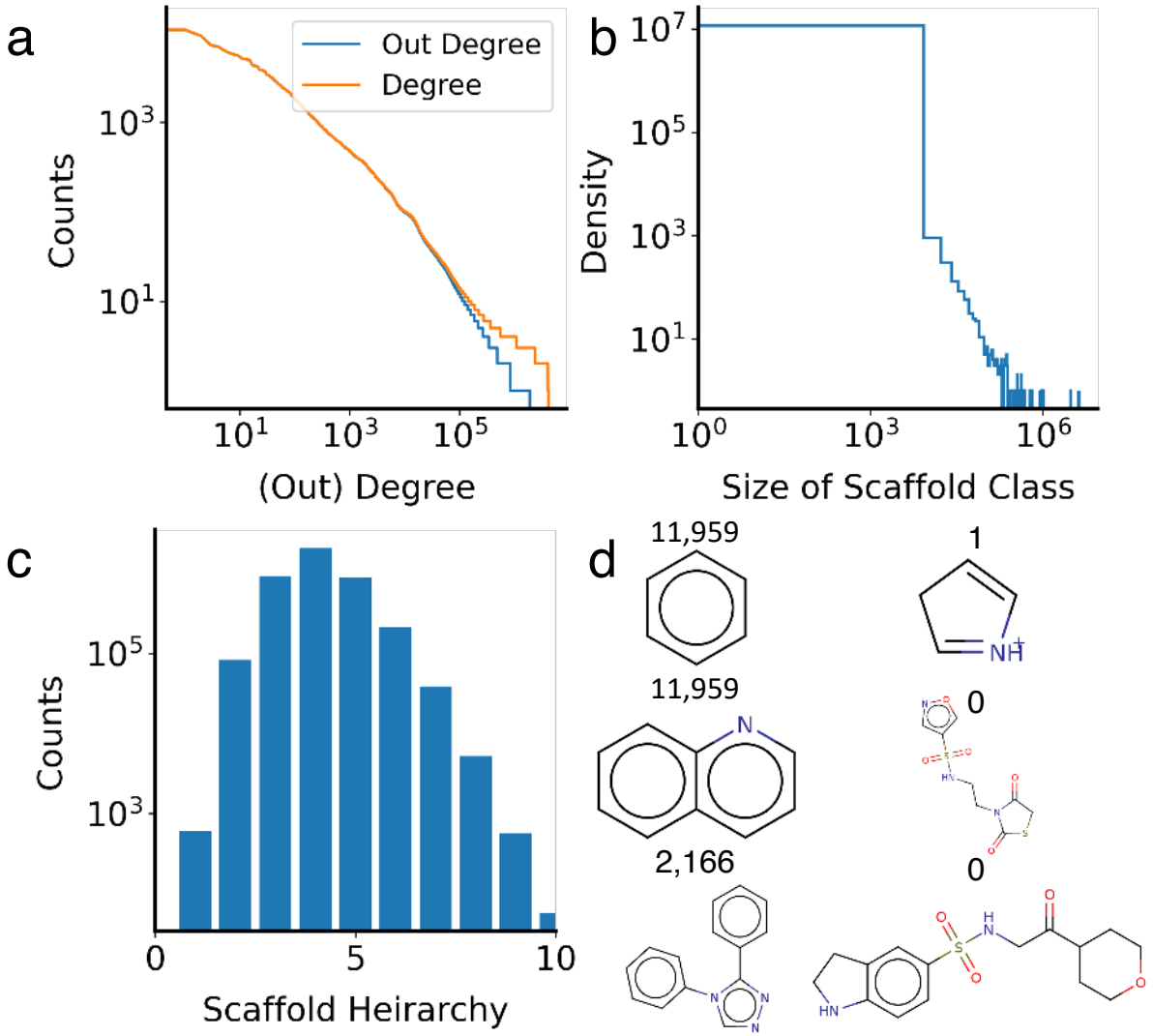


Figure 7.3: **Structure of scaffold classes** We constructed the scaffold classes (4M) for a random sample from SAVI (20M) molecules for (a)-(d). (a) We consider a random sample of 20M molecules from SAVI, and construct the scaffold classes and graph associated with the classes. Out degree indicates just `Successor` relations. (b) We show the distribution of the cardinality of (a)'s scaffold classes, which follows a power law for part of the distribution, and a uniform distribution for the other. (c) Scaffold classes are ordered into a hierarchy based on the number of rings its framework has. (d) The left column shows the scaffolds with the largest out degrees for hierarchies 1 to 3, and the right column shows random scaffolds of the least degree.

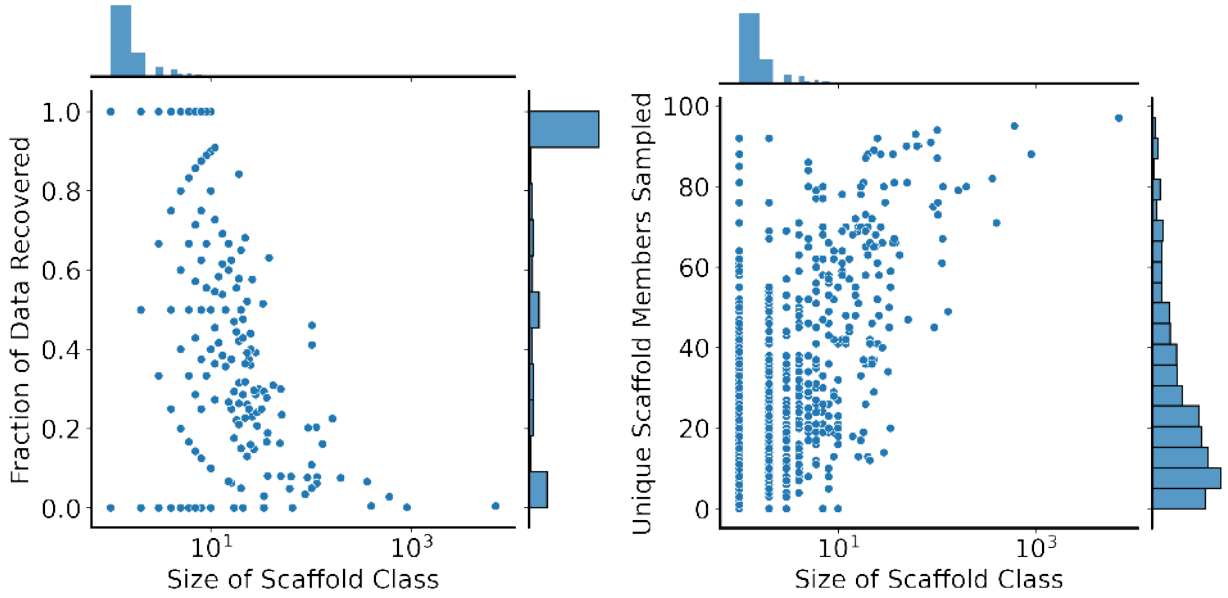


Figure 7.4: **$\texttt{Expand}_\Phi$ model reconstruction and sampling depth.** 1000 samples scaffold classes are drawn from the validation data, and $\texttt{Expand}_\Phi$ is sampled 100 times. Samples that are not valid smiles or passed verification are removed. (*left*) Samples for each scaffold are intersected with the known molecules in that scaffold class from the validation data, and the fraction found is plotted. Smaller scaffolds are often recovered while larger ones are not. (*right*) Even though the $\texttt{Expand}_\Phi$ model captures most of the dataset for smaller scaffolds, the model generates more valid molecules based on the natural distribution of the scaffold class sizes in the data.

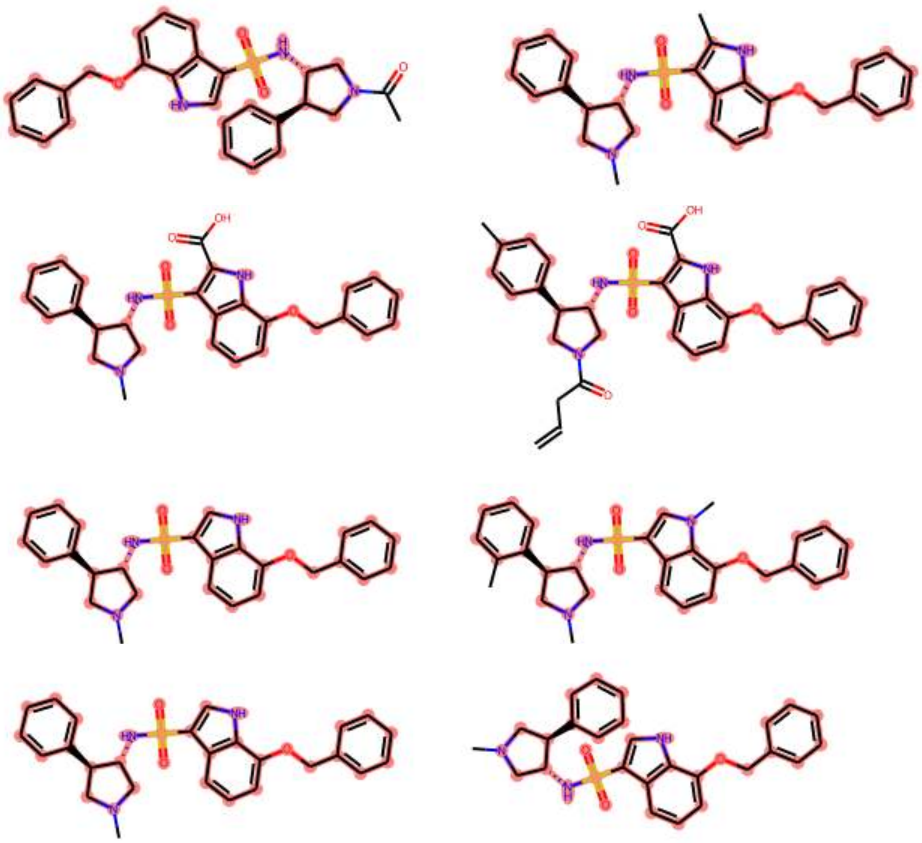


Figure 7.5: **Expansion of a scaffold.** The expansion of a scaffold class, highlighted in red, is expanded by sampling $\texttt{Expand}_\Phi$. Various side chains are added, but no sample is outside of the class.

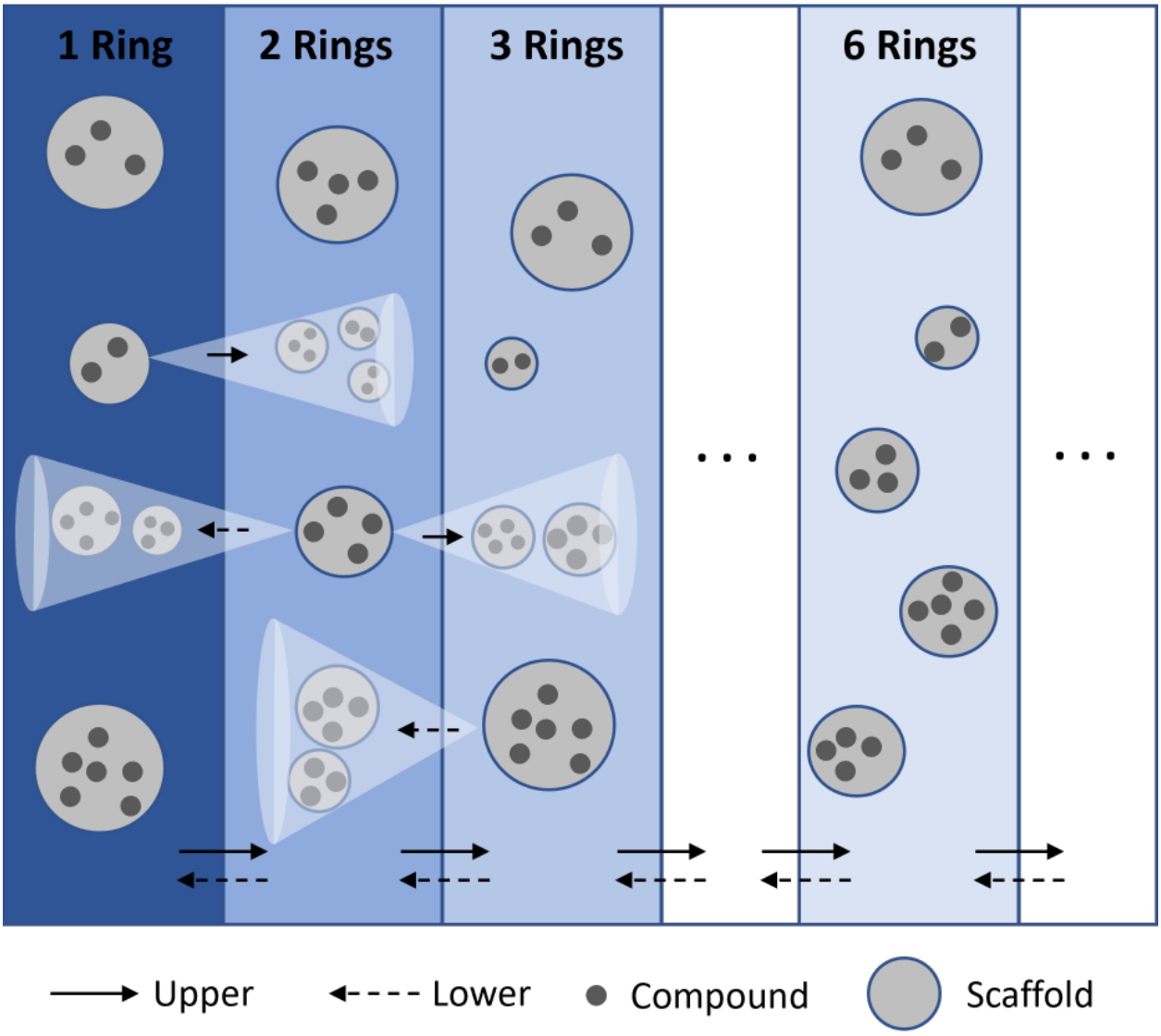


Figure 7.6: **Scaffolds as classes of chemical compounds in the chemical space.** Every chemical compound belongs to a single scaffold class. Scaffold classes are connected by common substructure thus forming a hierarchy. Upper and Lower are operations to traverse the scaffold classes at different levels of hierarchy.

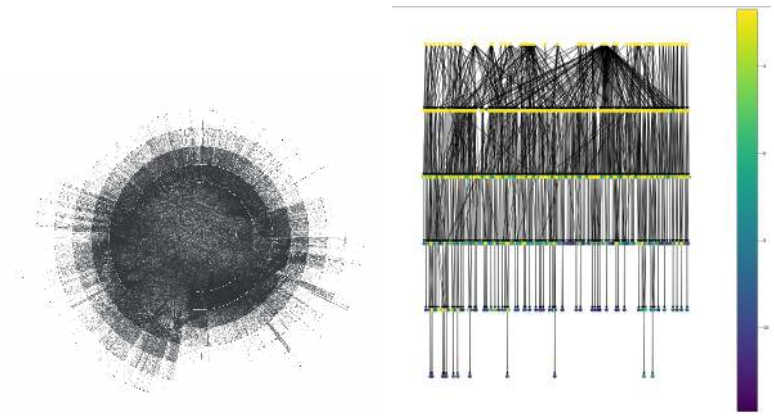


Figure 7.7: (left) Radial graph layout of the Mcule molecular database. Pulling from these compound libraries typically results in a large connected component with a tree-like structure (right) Subset of molecules structured in a tree graph used in COVID docking study Babuji et al. [2020b]. The enumerated compounds from libraries typically appeared as terminal nodes (in purple shades), while the other nodes (scaffolds) connect those molecules together and provide a further in-depth view of the chemical space (yellow).

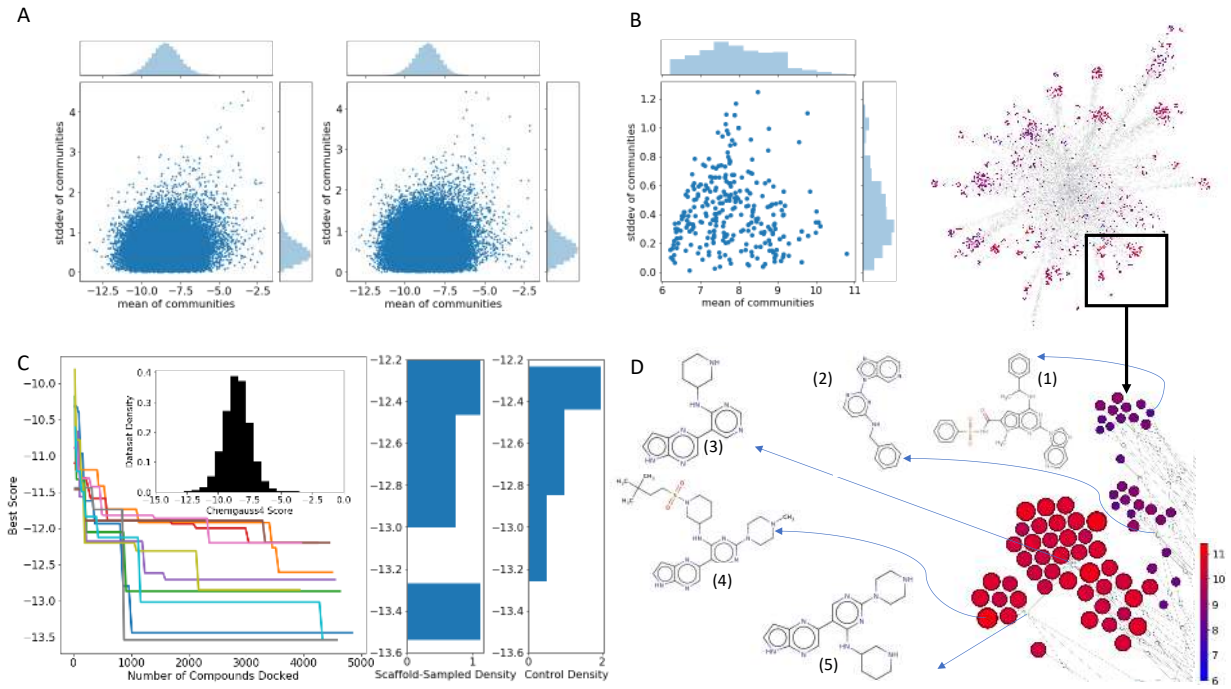


Figure 7.8: (A) Utilizing the scaffold graph based on the SARS-CoV-2 main protease (Mpro) dataset, communities contain all molecules which share a common substructure scaffolds. (left) this showcases the set of all communities based on three ring scaffolds, (right) the set of all communities based on four ring scaffolds. (B) The scaffold graph for JAK-2 kinase assay data from Laufkötter et al. [2020] (color bar in D). Red shaded nodes are near the top inhibitors in the data sample, and blue shaded nodes are near the bottom in the data sample. The grey shaded nodes are scaffold nodes which are not included in the data, but are used to organize the chemical space. (C) Highlights a docking simulation to show sampling based on the graph communities yields better performance than random sampling for capturing high performing compound classes with minimal docking (n=5000, 0.1% of total data). (D) Zoomed in pane of B, showing similar compounds sharing a few common ring structures, but diverge in terms of assay performance (1 is $p = 6.44$, 4 is $p = 10.76$.).

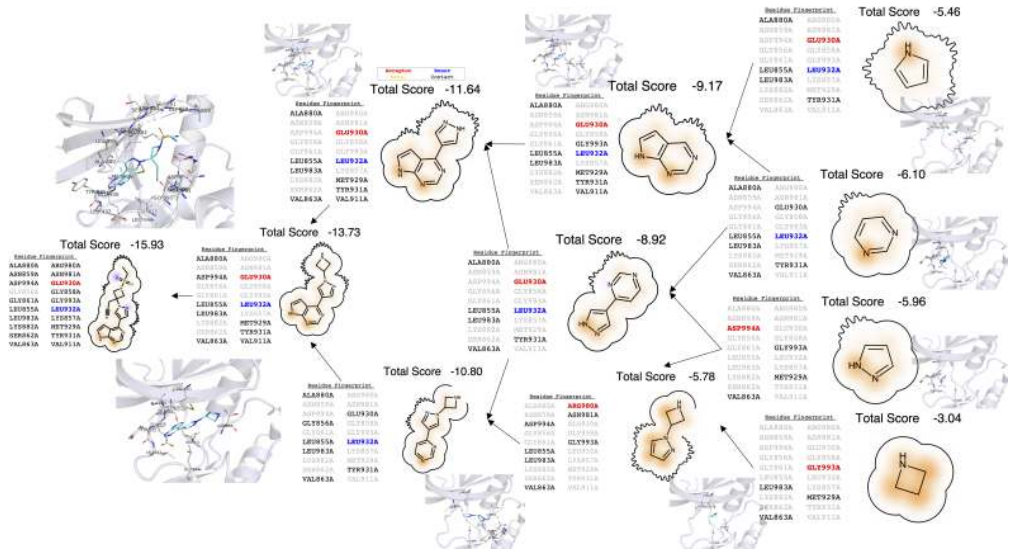


Figure 7.9: Baricitinib,a JAK2 inhibitor, (left) is decomposed by scaffolds into a graph (nodes being the molecules, edges showing how each decomposes to rudimentary single ring building blocks) McInnes et al. [2019]. Each scaffold was docked as an independent molecule to JAK2 kinase (3KRR) using the FRED pipeline from OpenEye Baffert et al. [2010], OEChem [2012a]. The residue fingerprints along with poses were also included. One can see how the scaffold based approach, on a microscopic level, shows promise as a meaningful conceptualization of chemical space—the merging of scaffolds with known properties leads to a compound with similar residue contacts.

CHAPTER 8

ANALYSIS OF VIRTUAL SCREENING MODELS

This chapter will overview how to analyze virtual screening models. The first section will be historical regarding the different sets of metrics used in the virtual screening community and their incongruence with those used in the deep learning community. The second section will introduce the idea of regression enrichment surfaces ([Clyde et al., 2020b]). *The third section will relate to this computational scaling* ([Lee et al., 2021, Ma et al., 2018]).

8.1 The problem of measures of central tendency

Ultimately, this computational effort requires selecting tiny sets of compounds from billion-scale compound libraries (akin to a needle in a haystack Raghuraman et al. [2006]). The top n selected compounds from a library discovered by virtual screening are referred to as *hits*, where n is often a mere 10 to 100. Unlike experimental high throughput screening in the lab, hits from uHTVS are not confirmed *in-vivo* and have no clear indication of "active" versus non-active Da et al. [2015], Cheong et al. [2009]. uHTVS protocols provide a score, and a distribution of them, along with some biological significance through the poses with associated protein binding regions. The hit selection process requires careful attention.

uHTVS distributions are highly skewed and vary heavily among targets and VS campaigns. uHTVS from docking are unlike normal distributions. In particular, one is interested in high scores rather than the mean or central tendency of data. Second, most compounds are not active against targets, and docking score distributions reflect this. If the goal is to locate 100 top scoring hits from a molecular database with one billion molecules, the model needs a predictive accuracy around 0.00001% (which should go without reference, is unheard of in machine learning literature). While various testing and model evaluation techniques have been established, few seem relevant to uHTVS, given that the entire distribution's

predictive success is not the overarching goal. In the literature, one often encounters predictive models being evaluated based on their mean absolute error (MAE), mean squared error (MSE), or correlation coefficients (such as r^2 or Pearson). In this section, we will argue these measures are uninformative at best and most often misleading and propose adopting a pragmatic model evaluation scheme based on the specifics of uHTVS.

A central tenant to training models, whether pursuing the re-scoring models or the docking surrogate models discussed earlier, is data splitting. Most are familiar with the typical train test split distinction (taking $x\%$ of the data for training and evaluating the model performance based on the $(1-x)\%$ remaining). Given the complexity and fine-tuning required with these models, authors have moved towards a train-test-eval partitioning system, utilizing only the train and testing data for model development. The evaluation or sometimes called holdout partition is not utilized until the end of the paper production process. Specifically, in molecular property prediction models, a close field of uHTVS, authors have been using more complex data splitting procedures. For example, scaffold splitting involves splitting molecular data into clusters based on their scaffold and holding out specific scaffolds from the training data to evaluate distribution model performance. In a similar light, time dimension splitting involves hiding compounds that were recently discovered pharmaceuticals to see if the model could have discovered them without seeing that class of molecular before [Feinberg et al., 2018].

While these are the currently accepted practices, we question if this is the correct approach for uHTVS. Most of these setups involve a dataset D sampled from a population distribution P, $D\tilde{P}$. The goal is when approached with a new sample, with possible shift or other sampling errors, $D'\tilde{P}$, how well can this model predict some outcome? For uHTVS, the problem should be stated differently. Given the population distribution of molecules P, we can only sample some known enumerated set, M. M consists of all known molecular species, even the finitely many ones we can list with a computer. M is the only set we

have. There is no notion of another sample different than M. Then, the goal of uHTVS is to score all molecules correctly in M and locate the best and most likely potential drug candidates. The difference between these two scenarios is that the domain for uHTVS is wholly known, computable, and finite. The domain for typical problems involves sampling from a population distribution. Here, given we know the finite larger domain exactly, we do not need to consider out of sample possibilities, and the implications of this ought to be further explored.

Given an increased historical attention towards experimental high-throughout screening, the developed metrics focused on the notion of *active* versus *inactive* (for VS, we have a continuous regression setting). For approaches to metrics in a uHTVS classification setting, see decoy detection ratio from a feasibility study [Irwin et al., 2009] or [Malo et al., 2006].

With the recent surge in deep learning research, metrics for model success have remained relatively stagnant. Data, models, and application domains seem endless, but the methods used to gauge success in the transfer of the model from research space to tools in real-world applications have rested on intrinsic value in mean squared error, classification accuracy, or assumption-based metrics such as r^2 score. In the classification side of the field, such as a subset of computer vision tasks, the metrics seem to represent the problem at hand truly. The accuracy of an image labeling system is representative of the real-world application, in classifying images for web searches or automatic keyword generation [Krizhevsky et al., 2012, Deng et al., 2009]. Domains such as natural language processing (NLP) have recognized this problem as they often employ specialty metrics such as the BLEU score [Papineni et al., 2002, Chen and Kuhn, 2011, Post, 2018]. In the new space of applying deep learning for drug discovery and virtual screening, the metrics used to score models have not moved towards the necessary specialization for exhibiting confidence and understanding their performance for the use in real lead discovery pipelines [Unterthiner et al., 2014, Zhang et al., 2017b, Pereira et al., 2016, Schneider, 2010].

A common problem in the drug discovery process is finding the few active drugs in the trove of the imaginable drug-like compounds, estimated to be around 10^{63} [Bohacek et al., 1996b, Fink et al., 2005]. Recent literature from medicinal chemistry has shown lingering inductive biases even in vast vendor libraries, indicating the idea of sub-sampling a library for testing as an ineffective means of screening, providing an impetus towards screening hundreds of millions to billions of diverse compounds rather than small curated libraries [Jia et al., 2019, Cleves and Jain, 2008]. Virtual screening is a computational technique for identifying possible subsets of hits suitable for downstream analysis with more expensive computational or lab experiments. Virtual screening is typically applied a costly or time-consuming process to select a subset of objects to either run those costly experiments on or pursue downstream work on. Virtual screens are applied to the biological sciences for drug screening, though recently there have been cross-over from material designs as both fields pursue machine learning screening workflows [Halls et al., 2013]. For example, virtual structural docking is a technique used to score compounds to understand how well, and if, a compound docks to a particular ligand [Lyne, 2002]. Producing an experimental value requires a time consuming experimental process, and is not possible at the scales of compounds that are screened virtually [Lyu et al., 2019]. With current high throughput laboratory and computational techniques, these sets can be large, even in the hundreds of millions in the case of *Lyu et al.*

In many ways, virtual screening is a ranking problem [Kontoyianni et al., 2005, Wilton et al., 2003, Kellenberger et al., 2008, Hawkins et al., 2007]. Given a set of drugs from a vendor library, researchers want an ordering of which compounds are likely to work where downstream experimentation or computation can begin. While classification metrics are used in the literature as a surrogate for the *what researchers are interested in* question, it is not uncommon in the lead discovery process to not know *a priori* what the cutoff or size of a desired hit set is—that is likely to be a function of budget, lab constraints, etc. By casting the problem in terms of rank ordering, it separates the development of the model and the

function of the model into research and development workflows. We want to use a model to rank the compounds in our domain, and we would like that ranking to align well with the actual rankings if finding those rankings were tractable. The field of information retrieval has created a variety of metrics based on document interest retrieval such as expected reciprocal rank (ERR) and discounted cumulative gain (DCG) [Chapelle et al., 2009]

$$ERR = \sum_{i=1}^{n} \frac{1}{r} P(\text{User is satisfied with r}) \qquad DCG = \sum_{i=1}^{k} \frac{2^{g_i} - 1}{\log(i+1)} \tag{8.1}$$

where $\psi(r)$ is the utility of a given rank (so $\psi(1) = 1$ and $\psi(x) \to 0$ as $x \to \infty$. In reality, the distributions encountered in this area are highly skewed, with a vast majority of compounds failing to work or being *interesting* in any sense to the experimentalist. As the rankings fail from the top two or three, the interest in their relative accuracy diminishes rapidly. Some progress has been made in ranking metrics though they generally fail to capture the heavy tail focus this problem exhibits and are ineffective in data with high regions of error propagating from experimental or random processes [Wang et al., 2014, Zou et al., 2016, Balakrishnan and Chopra, 2012, Cossock and Zhang, 2006]. The application of ranking measures to deep learning models is still in its infancy, leading to a disappearance of ranking from very recent literature with deep learning screening models.

Tradditional VLS metrics

Before moving on to discuss the difficulties associated with evaluating ML models in a virtual screening context, we outline a history of virtual screening metrics. In these settings, the targets were typically experimentally derived hits or known leads. In this setting, rankings or scores were typically not computed on the targets themselves, only on the predicted or computed values. New measures were created, such as the Boltzmann-Enhanced Discrimination of ROC (BEDROC) scores [Truchon and Bayly, 2007], enrichment factors, and ROC

Enrichment (ROCE) [Nicholls, 2008], robust initial enrichment (RIE) [Sheridan et al., 2008], and sum of the log of ranks test (SLR) [Zhao et al., 2009]. Borrowing notation from Truchon, let N be the study size and n be the number of active compounds. We will assume that there is some cutoff δ so that true hits are labeled as either a hit or not in the case that the true values are provided as real values. Let the cumlative density function $F_a(x)$ be the probability that an active compound is found prior to rank x, so if there are 1000 compounds in a screen, $F_a(10)$ is the probability that the top ten molecules from the screen provide a single hit. The probability density function is defined as $f_a(x)$ which is the probability the relative rank x is a hit. Using this CDF function, the area under the accumulation curve (AUAC) is simply equal to [Kairys et al., 2006, Truchon and Bayly, 2007]

$$\text{AUAC} = \int_0^1 F_a(x)\,dx = 1 - \langle x \rangle \tag{8.2}$$

where $\langle x \rangle$ is the average rank of a hit. ROC can be formulated form AUAC as [Truchon and Bayly, 2007]

$$\text{ROC} = \frac{\text{AUAC}}{R_i} - \frac{R_a}{2R_i} \quad \text{where} \quad R_a = \frac{n}{N}, R_i = 1 - R_a \tag{8.3}$$

As you can see in these formulations, the metrics cannot be tuned for early versus late recognition (see figure **??**). The insight came as data sizes grew and docking programs were screening larger than before libraries, where late recognition became costly. Eventually, moving the study towards early versus later recognition of hits, weighting schemes were created $w(x)$ such as $e^{-\alpha x}$

$$w\text{RIE} = \frac{\int_0^1 f_a(x)w(x)\,dx}{\int_0^1 w(x)\,dx} \qquad w\text{AUAC} = \frac{\int_0^1 F_a(x)w(x)\,dx}{\int_0^1 w(x)\,dx} \tag{8.4}$$

Notice in these formulations all measures are dependent on N, n and α such that [Truchon

and Bayly, 2007]

$$\mathrm{RIE}_{\max} = \frac{1 - e^{\alpha R_a}}{R_a(1 - e^{\alpha})} \qquad \mathrm{RIE}_{\min} = \frac{1 - e^{-\alpha R_a}}{R_a(1 - e^{-\alpha})}. \tag{8.5}$$

Relating these formulations by scaling and weighting

$$\mathrm{BEDROC} = \frac{\mathrm{RIE} - \mathrm{RIE}_{\min}}{\mathrm{RIE}_{\max} - \mathrm{RIE}_{\min}} = \frac{w\mathrm{AUAC} - w\mathrm{AUAC}_{\min}}{w\mathrm{AUAC}_{\max} - w\mathrm{AUAC}_{\min}}. \tag{8.6}$$

The most recent development for analysis is SLR [Zhao et al., 2009]

$$\mathrm{SLR} = \sum_{i=1}^{n} \log(r_i) \quad \text{where} \quad -\sum_{i=1}^{n} \log(\frac{r_i}{N}) \sim Gamma(n, 1). \tag{8.7}$$

The metric and its interpretation is dependent on n and N as well as the distribution of the ranking in the data.

All of these measures require at least one parameter setting a cutoff for marking a result a hit or not. AUROC represents the probability of active compounds ranking earlier than decoy or inert compounds. The BEDROC is related to robust initial enhancement (RIE) and both of them represent tuning the AUROC focus from late recognition to early recognition [Zhao et al., 2009]. In the framework from [Nicholls, 2008], these measures are satisfactory as they require parameter choices and change based on the data. Given the importance of cost structure analysis moving forward with deep leaning models in real world applications, we would argue they are also not interpretable in the desired sense.

Enrichment factor (EF) is most commonly used and is similar to a basic notion of hit rate (HR) [Krüger and Evers, 2010, Bender and Glen, 2005, Stahl, 2000, Shoichet, 2004, Pereira et al., 2016, Ericksen et al., 2017, Ain et al., 2015] where χ is used to represent a hit typically in terms of percentages so $\chi = 0.25$ means everything in the top 25% of the distribution is

labeled a hit

$$\mathrm{EF} = \frac{\mathrm{Hits}_{\mathrm{sampled}}/N_{\mathrm{sampled}}}{\mathrm{Hits}_{\mathrm{sampled}}/N_{\mathrm{total}}} = \frac{\int_0^{\chi} f_a(x)\,dx}{\chi}, \tag{8.8}$$

and gained popularity in the literature for comparing various docking methods. In this method, multiple cutoffs are used to translate docking scores, a continuous measure, to a hit classification. Yield hit rate (YHR) is another commonly seen metric which is effectively the same as EF in the literature [Harper et al., 2001, Ghosh et al., 2006]

Towards Deep Learning Surrogate Screening

In the relatively new area of virtual screening with statistical and machine learning (ML) techniques, machine learning models are used to replace docking programs or introduce early funnel stage lead discovery techniques [Chen et al., 2018]. Metrics for ML models are typically the only measures technicians have for answering questions regarding model convergence, parameter tuning, and learning success. In a sense, a metric for communicating the ML model utility must both satisfy the requirements for virtual screening as well as requirements for success in the ML community. While the analysis for most virtual screening metrics rely on a predetermined notion of hit, the sheer size and new domains ML models are creating has lead to a fuzzier notion of hit, where both the values being predicting on and the predictions themselves are continuous representations of a property (such as cell growth, docking scores, binding affinity δG values, etc.).

In the literature, most authors still report r^2 scores or area under the receiver operating characteristic curve (AUROC) with a cutoff [Feinberg et al., 2018, Unterthiner et al., 2014, Korotcov et al., 2017, Hamanaka et al., 2017, Wallach et al., 2015, Gonczarek et al., 2016, Menden et al., 2013b]. These metrics are standard to present in the ML community as a means of indicating a model matches a notion of being trained. Prior to the introduction of ML, metrics in the field were often used to compare docking methods, not to both evaluate the training a model as well as the usefulness of a model—two distinct notions, as structural

docking did not have a notion of model training or accuracy in the same way ML models do. At first, the choice of presenting AUROC for deep learning model performance was a sensible choice as virtual screening with structural docking has a large history of AUROC as a means of comparing scoring functions, programs, and efficacy [Triballeau et al., 2005, Nicholls, 2008]. ROC curves for virtual screening were preferred as they allowed for regression or uncertainty in a classification sense, though the targets or known hits had to be binned. The ROC curve plots sensitivity and specificity representing various cutoffs for marking a screened value a hit or not. Overtime, the curves would be omitted and represented solely by the AUROC, where the single value provides little information regarding early or late recognition problem and represents a balanced accuracy measure than the early ranking problem truly desired.

Of course, reporting these scores provides useful information regarding a general overview of the model performance (i.e. whether the model converged, is over-fitting, *working* in a usual sense); yet, r^2 or MSE does not provide actionable insight into how well this model will screen new compounds or treatments. If a drug company wanted to understand the cost-benefit analysis for lead development, a r^2 score does not provide insight into that computation. The choice of cutoff is also highly dependent on the downstream task, thus radically changing the actual scoring and recognition qualities of a model. Budget choices or downstream experiments are typically made by new progress in models and technology, but the current scoring techniques used for these models requires parameters based on the circular requirements.

There has been progress on converting these methods for regression models using some normality assumptions such as regression enrichment surface (REF) [Feinberg et al., 2018]

$$\mathrm{EF}_{\chi}^{(R)} = \frac{1}{\chi \cdot N} \sum_{i}^{\chi \cdot N} \frac{y_i - \overline{y}}{\sigma(y)} \tag{8.9}$$

where the experimental values y_i are ranked according to the model $\hat{y}_i$. This method is based on the distribution of the underlying data as it is normalized by $\sigma(y)$. This method should be preferred in regression settings; however the value itself is still dependent on some $\chi\%$ cutoff decreasing the interpretability across settings.

8.2 Regression enrichment surfaces

8.2.1 Instrumenting Virtual Screens

A metric for a VS model ought to guide decisions on whether or not to use the model in the first place. This means we must focus on the use case of the model in order to evaluate it. Luckily, for uHTVS surrogate models this is simple task: how often can we correctly identify the top n compounds?

A first attempt may look like an accuracy computation. We take the set of compounds, mark the underlying top ten as true and everything else false, and predict which ones are in the top ten. This simplicity, however, ignores a few rather essential insights. First, belonging in the set of the top ten is not the only data we have. We have a sense of magnitude on how far into the top ten it should go (for example, is 11th versus 10th place very different or somewhat similar?). Second, this accuracy computation, or enrichment factor, is not variable. When selecting the top n, it may be the case that the downstream task is rather complicated and n must be relatively small; however, what if a simulation task is introduced and one wants to understand how increasing n would change expected hit rates?

The solution we present, regression enrichment surfaces (RES), involves a continuous approach given the regression setting as well as a visual representation that escapes the single metric logic of model evaluation (see figure 8.2) [Clyde et al., 2020e]. To formalize the situation, we have y docking scores as output from the docking program, $\hat{y}$ docking scores as inferred by the model. Let us suppose the library size is L, and we want the top

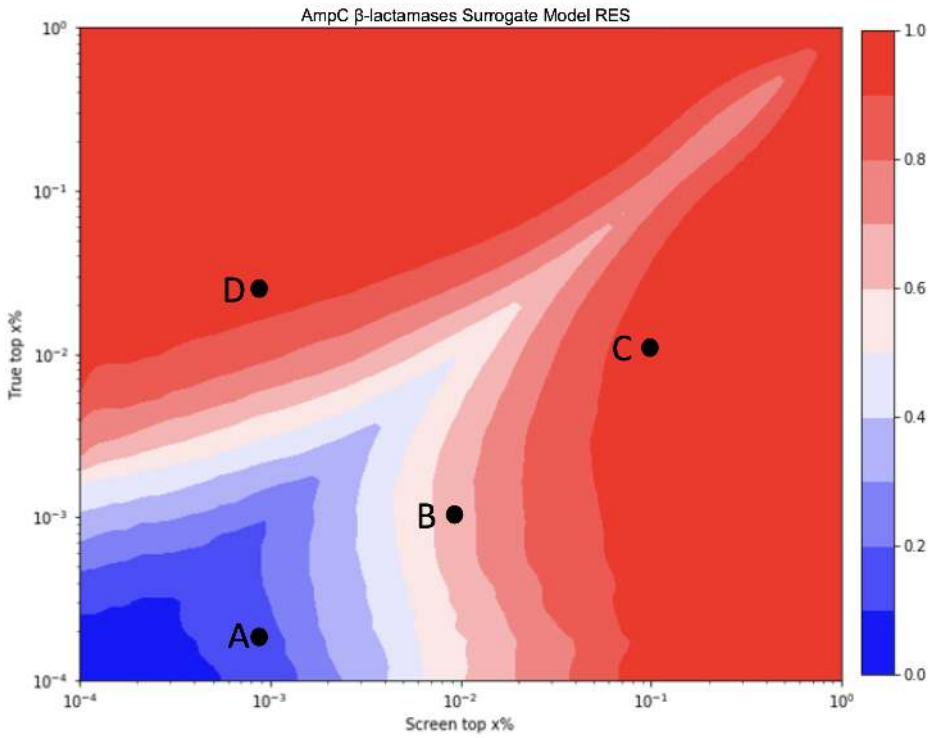


Figure 8.1: The RES plotted for validation data set of 50,000 molecules based on AmpC Beta-Lactamase docking scores. The model is a message-passing network. The model trained on 500K docking samples from data published by [Lyu et al., 2019]. Points indicate examples of plot interpretation. (A) shows that the predicted top 500 contain only about 10-20% of the true top 100 compounds, and just above that point, we see the predicted top 500 contain only 30% of the true top 500. (B) The predicted top 1% contains 50% of the true top 500, true top 0.1%. (C) The predicted top 10% capture all of the true top 1% (and further), thus this model at least allows one order of magnitude screen up where we capture most of the interesting true top distribution. (D) Points above the diagonal identify line are not insightful for this use case, however, (D) implies that the predicted top 500 contain 500 points which appear in the true top 8%.

α percentage of hits. Suppose we look at the top α predictions from $\hat{y}$, and we can ask the question, how many of the top α from y appear in $\hat{y}$? This produces the enrichment $\text{RES}(y, \hat{y}, \alpha, \alpha)$. Now we may want to impose a stricter notion of capturing really the very top, and consider $\text{RES}(y, \hat{y}, \alpha, \alpha/10)$. This is the flexibility of the questions we can ask under the RES paradigm.

We formalize this RES notion by considering the rankings induced by the true values, R, and the rankings induced by model inferences, $\hat{R}$. We call α the top percentile desired from

the true predictions, and $\hat{\alpha}$ the percentile that one will use from the inferences,

$$\mathrm{RES}(R, \hat{R}, \alpha, \hat{\alpha}) = \frac{|R_{\alpha} \cap \hat{R}_{\hat{\alpha}}|}{|R| \min(\alpha, \hat{\alpha})}. \tag{8.10}$$

From earlier, we considered the budget, or computational time required, T, with the library

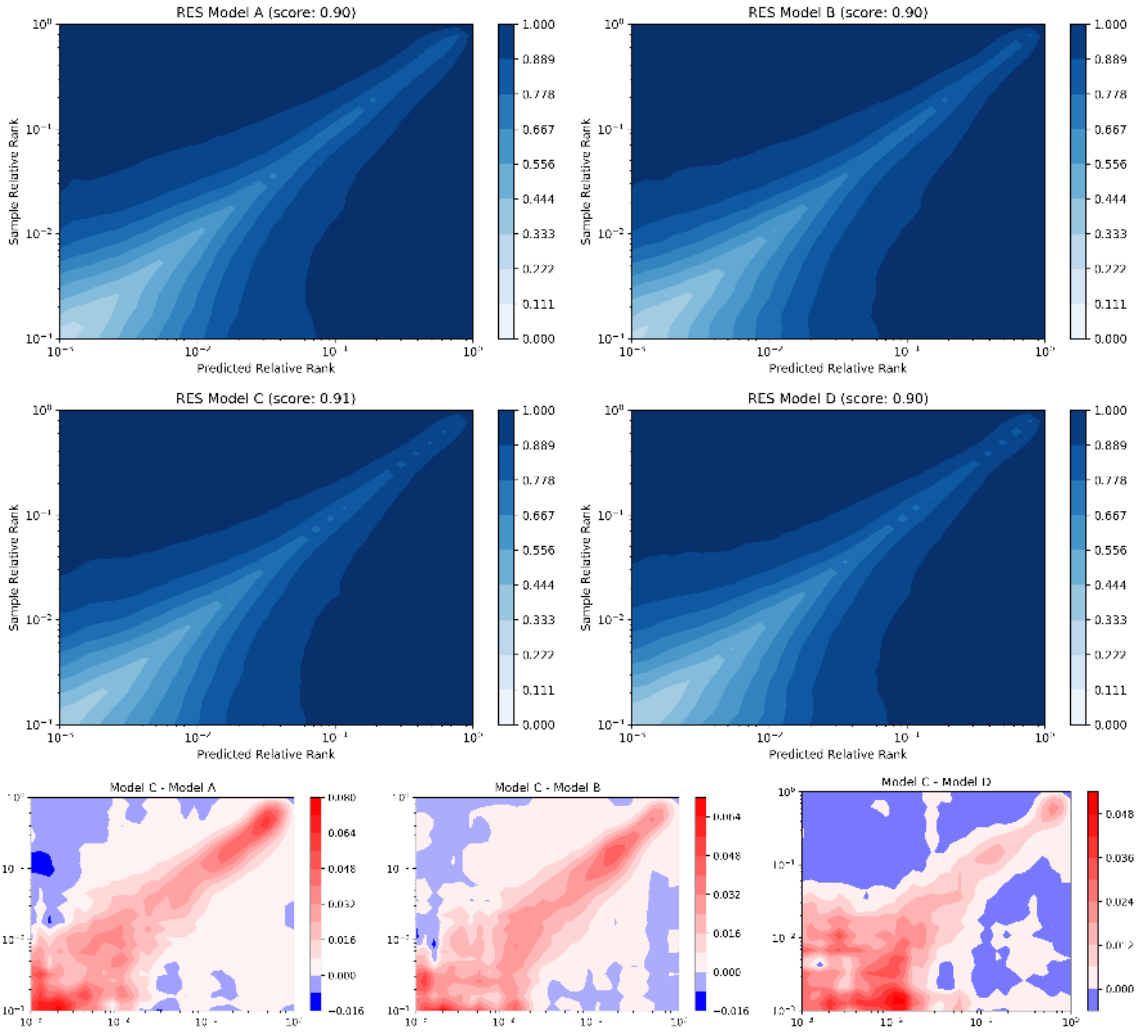


Figure 8.2: Regression enrichment surface (RES) plots for the associated model and predictions. The RES score noted in the title is an approximation of the integral where the bounds are alerted to be 0-1 for both x and y axis such that the best performance is 1 and the worsts performance is 0. It should be noted that the original bounds should be communicated so that the score can correctly be reported and reproduced.

size $L = |R|$, and α is the percentile of compounds one can dock. The relation with RES is

practical. Suppose one has a computational budget of T, which is fixed, and a specific library they want to deal with, fixing the size of L as well. Solving for α in $T = L(\phi + \alpha\psi)$, let us call that solution $\hat{\alpha}$ produces the top percentile of compounds one can solve the docking structure for. Now, within RES, one can check given the top $\hat{\alpha}$ of model inferences, how much of the original true top distribution would be correctly computed?

Once the hits are determined, uHTVS has completed a pass. Those hits can then be used for experimental screening, further simulation, or other property workflows. This funnel based strategy is quite common in drug discovery [Kitchen et al., 2004, Lee et al., 2020]

8.3 Accuracy and Scaling

Utilizing SPFD we report a 10x speedup to traditional docking with little to no loss of accuracy or methodical changes needed besides just utilizing ML-models as a prefilter. Our model detects 99.9% of the high scoring FRED compounds when filtering the dataset at 10%. We show that for a set of active 3CL-main protease compounds that SPFD would not miss any actives that the alternative FRED docking would identify. We see these results as conceptually tying together the model application (hit threshold and adversity to detection loss, choice of σ) with more traditional analysis such as performance characteristics and model performance evaluation. Furthermore, our data release consists of a square matrix of training data for further study on surrogate regressor models for accelerating docking studies..

To summarize the finding from the various comparisons in the results, we recommend utilizing an initial sample size of 100k as there was no observed accuracy increase using 1M initial samples. Using smaller initial sizes, if possible, decreases the overall training time and required molecular docking runs which increases the overall efficiency of the systems. Smaller initial sample sizes were not tested but will be in future studies. We further recommend uniform sampling the initial data to balance the respective docking scores to the best

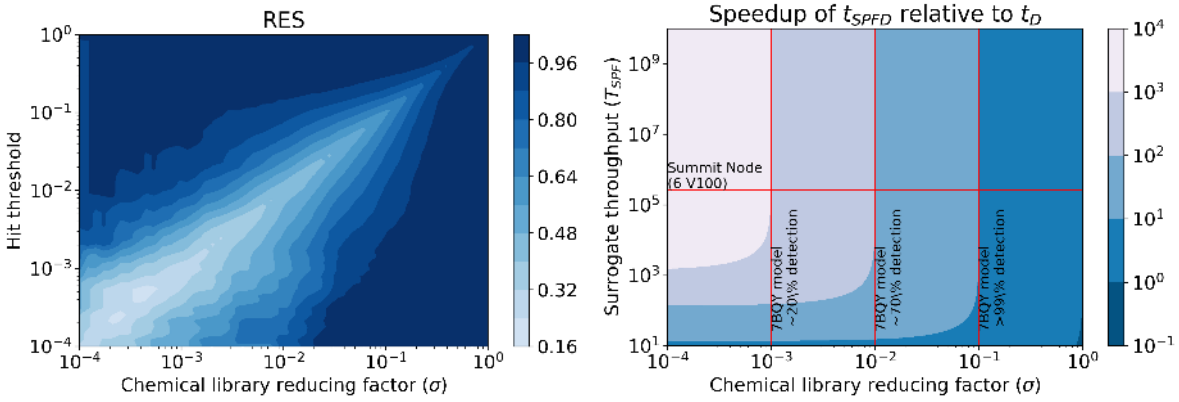


Figure 8.3: (left) Regression enrichment surface ($n = 200,000$) based on the surrogate model for 7BQY [Clyde et al., 2020b]. The x-axis represents σ which determines the level of filtering the model is used for (i.e., after predicting over the whole library, what percentage of compounds then used in the next stage docking). The y-axis is the threshold for determining if a compound is a hit or not. The point $(10^{-1}, 10^{-3})$ is shaded with 100% detection. This implies the model over a test set can filter out 90% of compounds without ever missing a compound with a score in the 10^{-3} percentile. In concrete numbers, we can screen 200,000 compounds with the model, take the top 20,000 based on those inference scores, and dock them. The result is running only 20,000 docking calculations, but those would contain near 100% of the top 200 compounds (as if one docked the entire dataset). (right) Based on equation (1) we compute the relative speedup using surrogate models over traditional workflows with fixed parameters library size (1 billion compounds) and $T_{\mathrm{D}} = 1.37 \frac{\text{samples}}{\text{seconds node}}$. The horizontal line indicates where current GPU, surrogate model, throughput is, T_{SPF}, and the vertical lines correspond to the RES plot values for hit threshold equal to 10^{-3}. The right-most vertical line implies a VLS campaign with surrogate models where the surrogate GPU-based model can with accuracy $> 99\%$ detect the top 10% from the bottom 90% implying a 10x speedup over traditional methods. By adding surrogate models as a pre-filter to docking, scientists can dock 10x more in the same amount of time with little detectable loss.

ability (see Fig. 3). This increased the r^2 score by nearly 15% in comparison to randomly sampling the initial training set (from 0.7 to 0.8). We observed significant improvement in the regression correlation (0.71 to 0.85) utilizing the larger feature set of molecular descriptors (1826 compared to 1613). The larger feature set includes 213 extra descriptors which pertain to 3D kernels (though they do not utilize any 3D structure). We recommend utilizing a quadratic weighting scheme as it decreased MAE the most towards the best docking scores, and shows insigificant difference on the least well scoring side of scores (which is less likely

to be an error as a bad docking score plus or minus a few points is still bad) (sec. 5.7).

This paper asks how can standard docking protocols be accelerated for large billion scale screening? Our timing analysis implies that to achieve a speedup beyond a single order computation does not need to be faster. Rather, the limiting factor to accelerating the workflow is a need for more accurate regressor models. Our analysis, outlined in Figure 8.3, highlights the choice of prefilter threshold as the limiting factor for seeing orders of magnitude speedup. In particular, focusing on speedups which show *no loss* of detection, model accuracy must be pressed forward as there is no path to accelerating traditional docking workflows without more accurate surrogate models. Given out of box modeling technique can speed up virtual screening 10x with no loss of detection power for a reasonable hit labeling strategy (top 0.1%), we believe the community is not far from 100x or even 1000x. The way to get there is to boost our model accuracies or develop techniques to recover hits in lossy SPFD regimes (such as not improving model performance but decreasing σ to 10^{-2} and applying another technique to recover the 30% loss of detection power). This benchmark is important, as successful early drug discovery efforts are essential to rapidly finding drugs to emerging novel targets. Improvements in this benchmark will lead to orders of magnitude improvement in drug discovery throughput.

We discuss the relative speedup of utilizing a pre-filter surrogate model for docking campaigns against a traditional docking campaign. We define two workflows for performing protein-ligand docking over a library of compounds: D (traditional docking, no surrogate-prefilter) and SPFD (surrogate-prefilter then dock). We construct timing models of both of these workflows to understand the relationship between computational accuracy, computational performance (time and throughput), and pre-filter hyperparameter (σ). We distinguish between the surrogate model's accuracy, which pertains to how well the surrogate model fits the data, and workflow accuracy, which pertains to how well the results of the whole SPFD workflow compares to the results of just traditional docking workflow.

As discussed in Section 5.1, the choice of pre-filter hyperparameter is a decision about workflow accuracy for detecting top leads. Model accuracy influences the workflow accuracy, but the workflow accuracy can be adjusted with respect to a fixed model accuracy (see figure 8.3 where the vertical lines each correspond to the same underlying model with fixed accuracy but differentiate the overall workflow accuracy with respect to traditional docking). Therefore we can interpret σ as a trade-off between the workflow throughput and workflow accuracy. For example, $\sigma = 1$ is always 100% workflow accurate since traditional docking is run on the whole library when $\sigma = 1$, but $\sigma = 1$ is even slower than traditional docking as it implies docking the whole library as well as utilizing the surrogate model. We can determine the overall workflow accuracy with the model by looking at the RES plot, which is of course has as a factor the performance characteristics of the surrogate model. Given a particular model's accuracy versus performance characteristics, different levels of pre-filtering (σ), correlate to different tolerances to detecting top-scoring compounds.

For the following analysis, we fix node types for simplicity. Let L be the number of compounds in a virtual library to screen. Assume the traditional protein-ligand docking software has a throughput T_{D} in units $\frac{\text{samples}}{(\text{seconds})(\text{node})}$, and the surrogate models have a throughput T_{SPF}. Let t_{D} and t_{SPFD} be the wall-clock time of the two workflows. The time of the traditional workflow, t_{D}, and the time of the surrogate prefilter then dock workflow, t_{SPFD}, are

$$t_{\mathrm{D}} = \frac{L}{T_{\mathrm{D}}} \quad \text{and} \quad t_{\mathrm{SPFD}} = \frac{\sigma L}{T_{\mathrm{D}}} + \frac{L}{T_{\mathrm{SPF}}}. \tag{8.11}$$

Notice, that t_{SPFD} is simply the sum of the time of running the surrogate model over the library, L/T_{SPF}, and the time of traditional docking the highest scoring σL compounds. The time to train the deep learning model is excluded as it is constant time (assuming 100k docking scores are used to bootstrap the model). Furthermore, the training time of our proposed neural network is roughly two to three hours on a single NVIDIA A100 GPU which is rather small compared to the run-time on hundreds of supercomputer nodes for

large-scale docking studies.

$$\text{Speedup} = \frac{T_{\text{SPF}}}{T_{\text{D}} + \sigma T_{\text{SPF}}} \tag{8.12}$$

This implies that the ideal speedup of our workflow is directly dependent on the throughput of both the docking calculation, surrogate model, and the parameter σ. σ is indirectly dependent on the model accuracy. If the surrogate model was completely inaccurate, even though $\sigma = 10^{-3}$ implies a 1000x speed up, no hits would be detected. If one wants to maximize workflow accuracy, that is not miss any high scoring compounds compared to traditional docking, then they must supply a threshold for hits (corresponding to the y-axis of RES). Suppose this threshold is $y_{\text{thres}} = 10^{-3}$. If they wanted to maximize not missing any compounds, they should set σ to 10^{-1} based on this model's RES plot since that is the smallest value of σ such that the detection accuracy of the surrogate model is near 100%. But, it is not always the case downstream tasks require 100% detection—hence σ is a true hyperparameter.

We infer T_{SPF} on a Summit (Oak Ridge Leadership Computing Facility) and on an A100 ThetaGPU node (Argonne Leader Computing Facility). Both tests were using 64 nodes, 6 GPUs per node, but the throughput was computed per GPU. We found the V100 summit node was capable of 258.0K$\frac{\text{samples}}{\text{(s)(node)}}$ while the A100 nodes were 713.4K $\frac{\text{samples}}{\text{(s)(node)}}$. We infer T_{D} as 1.37 $\frac{\text{samples}}{\text{(s)(node)}}$ based on a CPU docking run over 4,000 summit nodes with 90% CPU utilization from [Clyde et al., 2021d]. Thus, we can compute the speedup based on a Summit node head-to-head comparing setting T_{SPF} to 258.0K and T_{D} to 1.37 in Equation 1, resulting in a speedup of 10X for $\sigma = 0.1$. Based on the RES analysis in Figure 8.3, σ of 0.1 corresponds to a model accuracy of near 100% for filtering high scoring leads ($> 1\%$ of library). If one is willing to trade-off some loss of detection, say 70% detection of high scoring leads, then the speedup is 100X. The extreme case, a choice of $\sigma = 10^{-3}$, implies a speedup 1000X but means only roughly 20% of the top scoring leads may appear at the end.

Therefore, our analysis of SPFD implies that speedups are essentially determined by σ while T_{SPF} does not have as large of an effect (this is based on how fast ML inference currently is). As a hyperparameter, σ is dependent on the workflow's context and, in particular, what the researchers are after for that SBVS campaign. We can say, though, at least informally, model accuracy and σ are highly related. The more accurate the models are, the better the RES plot gets as one is willing to trust the ML model for filtering the best compounds from the rest. In figure 8.3, the x-axis of both plots are similar. The accuracy of a particular σ is found by setting one's level of desired detection, the y-axis of the RES plot, and then checking the (σ, y) point to see how accurate the model is there. The choice of σ is subjective based on how accurate one needs the model for their y-axis threshold for hits. We focus mainly on the case of no loss of detection, which means $\sigma = 0.1$ for our particular trained models. In order to focus on the theoretical model of relating computational accuracy, confidence (again, in a colloquial sense), and computational performance, we simplify over a richer model of performance analysis assuming uniformity of task timing and perfect scaling. Furthermore, we chose a head-to-head comparison of a particular node type's CPU performance to GPU performance. At the same time, we could have compared the best non-surrogate model workflow times to the best surrogate model workflow times.

8.4 Conclusion

In this chapter we outline the challenge with standard metrics in virtual ligand screening since ligand screening is detecting only a small fraction of hits under noisy conditions. We introduce the concept of regression enrichment surfaces as a model comparison tool. We then demonstrate that this can be related to computational efficiency of a workflow in order to characterize that future advances in virtual ligand screening must come from improved statistical models with decreased error rather than computing speedup.

CHAPTER 9

CONCLUSION AND OPPORTUNITIES

In this chapter we conclude with summarizing the developments in the previous chapters. I comment more broadly on different strategies that one can use beyond the HPC/AI workflows outlined in this chapter such as generative modeling and coupling it with simulations (RLMM). The third section will offer some forward looking ideas about how the next order of magnitude speeds up can be sought. The final section will conclude with ideas around automated discovery and autonomous laboratories.

In this dissertation, we have presented a particular model of computational drug discovery based on the idea of virtual screening chemical space $\mathcal{M}$. Chapters 4 through 6 demonstrate that naive screening is an extremely fruitful process capable of generating drug candidates [Clyde et al., 2021d]. In chapter 6, we introduce the complication that instead of there being a single scoring function f, that we can have scoring functions that have increased computational complexity which we stage molecules through depending on their overall score. In chapter 7, we introduced that each molecule may in fact not be independent—the score of one compound can inform calculations about scores of another compound. This insight can also be connected with a new database conception of using LLMs for queries. Finally, in chapter 8, we discuss the role of analysis. We argue that without breakthroughs in the calibration of models and the ability to measure extremes of a distribution with surrogate models that merely increasing computational speed will not scale drug discovery much further than the work we demonstrated here. Given this, in the follow three sections I outline three projects which we think can go beyond the model of drug discovery presented so far in this dissertation.

9.1 Reinforcement Learning for Molecular Modeling

Introduction

We present an end-to-end lead optimization system for discovery based on an AI-gym environment called “Reinforcement Learning for Molecular Modeling" (RLMM). RLMM automates running fully customizable molecular dynamic simulations inside of an agent-based molecular design protocol. RLMM is fully autonomous—from a single starting ligand, protein structure, and configuration file, RLMM cycles through designs for lead optimization informed by physics-based simulations. RLMM's design is flexible for integration with future autonomous laboratory protocols, with an aim of driving hypothesis generation. The flexibility of RLMM is driven by a graph-based representation of chemical space which is made tractable by a generative transformer model.

The multi-step process of drug discovery is incredibly resource intensive. The time it takes to go from initial compound search to a clinically tested, effective product can range between 10 to 15 years, and can cost over 1 billion dollars [Hughes et al., 2011]. Considering the universe of about 10^{60} possible compounds to traverse for effective drugs, we seek more efficient, higher throughput, and more meaningful frameworks for compound analysis [Bohacek et al., 1996a].

Computational and medicinal chemists have noted the small compound libraries used for HTS are biased towards certain properties and chemotypes [Jia et al., 2019]. One group found novel and unexpected nanomolar inhibitor for beta-lactamase by increasing the library size 100 million, going beyond the standard chemotypes in an ultra-large library screen [Lyu et al., 2019]. Accessing these large-scale libraries is an exciting area with unbounded potential for new pharmaceuticals and therapies. Machine learning for small molecule design is used as a lead discovery tool to filter a large dataset or to generate new leads based on experimental or computational data; however, physics based modeling is often required to

pursue downstream steps in lead development. Typically, leads from ML techniques are docked to the protein target with exhaustive rigid docking programs such as AutoDockVina or FRED—a highly noised indicator of possible activity with the target protein. Advanced physics based techniques are often used ranging from simple binding free energy estimation to highly accurate alchemical/relative free energy calculations.

In the pursuit of exploring the novel chemotypes in these libraries, the standard computational practice will not be sustainable beyond the current sizes of data available. With the current approach, every compound is typically run through inference property models followed by docking/physics modeling . While GPU computing has made ML inference very fast, it still would not be able to screen anywhere near the imaginable small molecule space. In the short term, high throughput docking/property prediction will be a useful technique for identifying active compounds in current libraries. Looking forward, a successful approach has to rethink the exhaustive search bottlenecking drug discovery.

In addition to novel high throughput screening methods, a growing body of work to improve lead generation and optimization with machine learning (ML) models and artificial intelligence (AI) frameworks has emerged Since Nichols et. al's concept of integrated MD "snapshots" was published in 2011. This includes a diverse array of applications, using recurrent, convolutional, and graph neural networks for various aspects of small molecule design [Elton et al., 2019]. Much of this work uses ML/AI as tools in lead discovery, filtering large datasets or generating new leads based on experimental or computational data. Similar to high throughput docking techniques, these models are used in a sequenced compound discovery process. ML inferences are often made, then validated with physics based models to screen and refine the molecules predicted, where they are then passed to other downstream steps in lead development. For example, many ML-based approaches will infer leads which are then screened with exhaustive rigid docking programs such as AutoDockVina or FRED [Gentile et al., 2020]. Advanced physics based techniques ranging from simple bind-

ing free energy estimations to highly accurate alchemical/relative free energy calculations are also sometimes leveraged to more rigorously explore the compounds coming out of ML/AI frameworks.

Reinforcement learning has grew in interest and been successful in molecular design projects [Zhou et al., 2019]. Reinforcement learning recasts many problems as a control problem, a Markov decision process. The hallmark of a Markov decision process is the framing of a action space, an agent who takes actions, and a world of states which actions move between. For drug discovery, this often means framing chemical space as the world, particular ligands as states, and an agent choosing between ligands. To the best of the authors knowledge, no work has yet to frame particular protein-ligand complexes as states to move between. Of course, molecular dynamics simulations can be thought of as having states (which are snapshots) and dynamics moves between the states. The difficulty of reinforcement learning has been the connecting all the pieces together. One popular solution has been the Gym environment [Brockman et al., 2016]. The Gym framework aims to abstract the various components of a Markov decision process to codeable modules.

The integration of MD simulations with screening has been suggested and sparsely implemented for VS methods. There is most notably a handful of work which addresses protein flexibility in docking through MD simulation. Recalling the “relaxed complex" scheme developed in the early 2000's[Lin et al., 2002], protein flexibility is one of the nuances that affects the true affinity of a molecule. The target protein itself is not a static object with a fixed binding site [Lin et al., 2003]. Instead, there are forces acting on the protein from the ligand, as well as other molecules and solvents in the environment, that determine whether the conformation of a molecule is more or less likely at discrete time-steps [Baron and McCammon, 2008]. All the atoms in this environment can be modeled by an equation and parameters, representing a force field [E Nichols et al., 2012].

Thus, the target's state is a factor influencing the ligand's ability to bind [Amaro et al.,

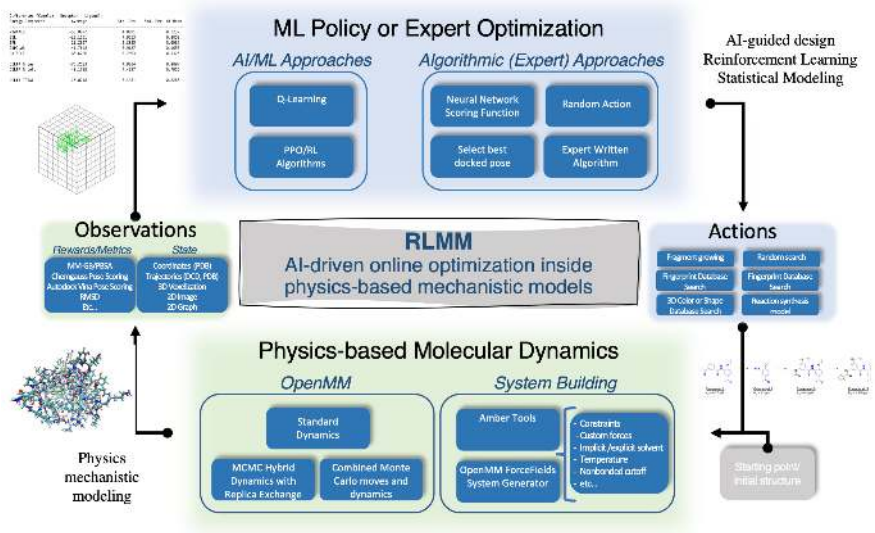


Figure 9.1: *RLMM workflow.* RLMM is an AI-driven lead optimization engine. There are four components of RLMM combined into an end-to-end loop. RLMM begins with a starting protein structure and docked (or bound) ligand.

2008]. MD simulations can model these state-changes of the target, and the interacting forces between the ligand and the target, as well as other forces present in the environment. This allows us to sample different states of the target's flexible binding sites, and identify ligands most likely to conform based on the forces present in the simulation [Nichols et al., 2011]. When a ligand binds, both it's shape and the shape of the target are altered, and the entropy and enthalpy involved in this reaction influence the binding free energy, as well as other components in the overall force field [Baron and McCammon, 2008]. While docking methods have incorporated the idea of the flexible ligand to some extent, their approximations of steric complimentarities to score different orientations of the ligand give far less insight into the question of affinity than the calculation of changes in binding free energy [Nichols et al., 2011].

We have developed a computational platform merging early stage drug discovery with late stage physics based molecular dynamics simulations. Reinforcement Learning Molecular Modeling (RLMM) is a platform for connecting state of the art neural networks to advanced

and trusted physics-based molecular dynamics software. Our approach combines concepts of high-throughput *in silico* computer-aided drug discovery methods with Molecular Dynamics (MD) simulations. MD simulations create high-accuracy representations of how compounds may react with a target in a rich, detailed environment. However, due to their computationally intensive nature, they have not been widely used in the high-throughput problem-space. The common approach is to combine them in a stepped sequence; identify leads through *in silico* methods, then preform the time-intensive simulations on the highest scoring compounds. Much of the literature follows this approach [Mirza et al., 2016, Okimoto et al., 2009, Sivanesan et al., 2005, Shukla et al., 2018, Brameld et al., 2008]. In this paper we aim to present RLMM as a novel workflow and approach to the lead discovery and optimization process.

9.1.1 Methods

RLMM connects various state-of-the-art molecular dynamics simulations with a agent-based policy environment. We outline the basis of the molecular dynamics simulations utilized in RLMM and describe the methods for navigating chemical space in an agent-based model.

Molecular Dynamics Simulations

Computational tools for assessing the binding affinity for a ligand generally rely on molecular dynamics simulations. Given the aim to connect machine learning with physics-based modeling, we focus on employing physics based simulations of protein-ligand complexes. Various software packages exist with an API for creating, running, and analysing molecular dynamics simulations Standard molecular dynamics models [Eastman et al., 2017b, **?**]. Molecular dynamics simulations are widely used to estimate the binding affinity of a proten-ligand complex compuationally [Verkhivker et al., 2001, Comitani and Gervasio, 2018, Mattedi et al., 2019, Woods et al., 2006, Rick, 2006]. Advanced sampling techniques are also used in molecular dy-

namics simulations such Markov Chain Monte Carlo (MCMC) or replica exchange [Chodera and Shirts, 2011, Wang et al., 2013]. Molecular mechanics generalized borne surface area (MMGBSA) is a technique for estimating the binding affinity of a protein-ligand complex. MMGBSA methods are less computationally expensive than free energy estimations. Free energy estimation software packages utilize more complicated achemical techniques [Rizzi et al.]. We utilize the MMGBSA.py script from Amber20 to estimate the MMGBSA scores for a series of molecular dynamics snapshots [Case et al., 2021b].

Drug Optimization as a Markov Decision Process

In terms of a Markov decision process, this module aims to define the space of available actions, A_s, possible dependent on the current state. In this formulation, the definition of state depends on the application of the RLMM workflow. For lead optimization, the state will contain the state of the simulation module as well as the ligand information. Available actions may depend on the current ligand. RLMM modifies the ligand at progressive steps in the modeling process. RLMM first generates a set of candidate "actions," or chemical modifications to the ligand, using one of two example action spaces.

9.1.2 Results

First, we discuss the result of training a transformer model to navigate the scaffold-based molecular space graph designed in section 7.2. Then we demonstrate the RLMM system as an end-to-end piece of software (figure 9.1). We perform two lead optimization tasks. The first task shows the detection and alterations of a known JAK2 Kinase decoy towards an analog of a known active inhibitor. We showcase the utility of RLMM for optimizing a known lead for the 3CL-main protease of the SARS-CoV-2 virus. We highlight the ability of RLMM to highlight potential novel chemotypes similar to known potent series against the main protease.

Reinforcement Learning for Molecular Modeling System

We demonstrate an end-to-end modular system for molecular optimization. RLMM code is available online at `github.com/aclyde/RLMM`.

Modules

The RLMM package is comprised of five general components that make up the backbone of the platform: system building, simulation setup, action space, observation space, and policy. Each of the five components consists of sub-modules with unique properties and behaviors. The connection between modules is provided by RLMM.

RLMM was designed with the intentions of being extremely modular and flexible in accommodating the build of a wide-breadth of configuration possibilities. The package is run by reading in a YAML configuration file, such that the lead optimization is fully initialized and optimized by parameters provided by the configuration file. Sub-modules are instantiated and parameters are parsed from the configuration file, which are passed through lead-optimization deployments via a configuration object containing all of the parameters detailed in the input configuration file.

With this configuration-based design, a user providing a valid configuration file is able to fully run RLMM with zero-to-minimal Python or coding experience, thus making RLMM relatively accessible within the likeness of other molecular dynamics programs. Additionally, most sub-modules contain default settings. Code-experienced users are afforded the ability to create and deploy their own policies within RLMM. User-defined policies and actions will unlock the great potential of RLMM's lead-optimization, accelerating research by allowing users the opportunity to exclusively focus on lead-optimization theory while leaving it to RLMM to automatically handle the environment and observations.

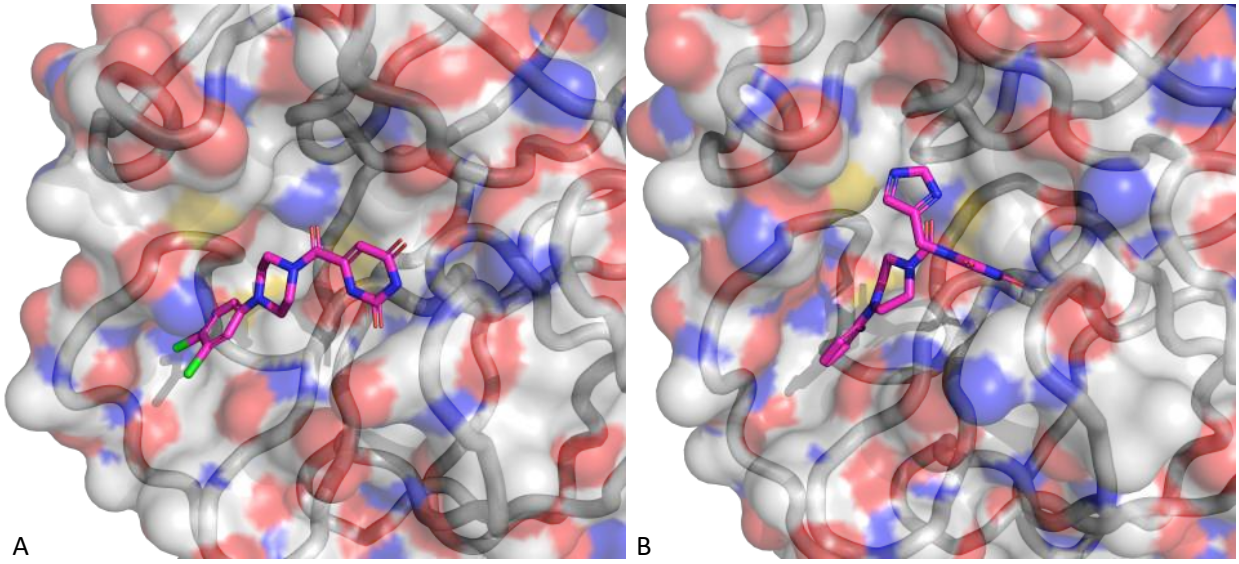


Figure 9.2: **An example of automatic system building.** RLMM automatically builds and places ligands in an aligned position as it replaces the ligand from (A) to (B) automatically. This allows near continuous molecular dynamics runs as the agent modifies the ligand. By maintaining a close position and automatically building the system, the stability of the simulated system is maintained.

Automatic system preparation and building

RLMM implements a system builder module designed with lead optimization in mind. The system building module abstracts the workflow required to take a protein structure and ligand (possibly already in complex or not) all the way to simulation ready (force field parameters, solvation, and built MD-ready system with forces specified). The steps can be broken down into two distinct steps, structure cleaning and parameterization, and OpenMM system building.

Several packages are built into RLMM to prepare an initial protein or complex structure for simulation. Initial structures can be provided in a PDB file containing a complex, separate protein/ligand structures, or just the protein with a proposed 2D molecule structure (in SMILES). User-provided protein structures are then prepared or "fixed" for simulation. Often, crystal structures pulled directly from the RCSB Protein Data Bank contain missing atoms/residues, clashes, and other artifacts that impact downstream preparation such

as parameterization with a force field [Suruzhon et al., 2020]. RLMM utilizes OpenMM's Modeller and pdbfixer, and Open Eye Scientific's SPRUCE toolkit to fix the initial structure [Eastman et al., 2017b, OEChem, 2012b, Eastman, 2013].

There are two parameterization protocols built into RLMM using Amber/tleap workflow or Open Force Field project's SystemBuilder [Chodera et al., 2021]. Both protocols perform similar operations, parameterizing the protein, ligand, and the complex to input into OpenMM. The AMBER-based protocol uses GAFF force fields for ligand parameterization and tleap for solvation with TIP3 [Case et al., 2021a, Maier et al., 2015b, Wang et al., 2004b]. The Open Force Field Initiative (OFFP) protocol utilizes Open Force Field Initiative's small molecule force field for ligand parameterization, and the Modeller package from OpenMM for solvation with TIP3 [Eastman et al., 2017b, Jorgensen et al., 1983].

If the initial ligand is not provided, or the proposed system only provides the ligand structure (via SMILES for example), proteinization and conformer generation is performed using Omega—tautomers and enantiomers are enumerated prior to conformer generation [OEChem, 2012b]. Ligand conformations are selected based on the structure building criteria. The criteria for both the selection of ligand conformation as well as the placement of the ligand in the protein is user selected. Currently three complex building methods are utilized and implemented: (1) the conformer which optimizes shape or pharamacmpore overlay with the previous ligand is selected, and the ligand is placed in that optimized overlay pose on the protein assuming no clashes. If the ligand clashes with the protein, this method attempts the next best overlaid conformation. (2) the conformer is selected which has the highest Chemgauss4 pose score from exhaustive rigid docking utilizing FRED [Mcgann et al., 2003a]. The ligand conformation is then placed in that docked position or ROCS can be utilized to place the successful conformation as closely as possible to the prior ligand position (figure 9.2). Beyond these techniques, users can extend the system building module to perform the placement and ligand preparation as their workflow requires.

Typically for lead optimization tasks, tautomers and enantiomers are enumerated for the incoming proposed analog or perturbation to the previous ligand. Conformer generation is performed on the ensemble of structures, generating 200-800 3D conformers for every enantiomer and reasonable tautomer generated. The conformer and placement of the ligand is selected based on the best shape overlay to the previous ligand. We utilize this system preparation method for lead optimization to mimic and interrupted simulation, where the start of the new simulation matches the end of the previous simulation as closely as possible.

Observations

Observation spaces in RLMM convert the state of the molecular dynamics simulations into a representation for downstream tasks. For example, the simplest observation space passes a PDB file with the coordinates and topology of molecular dynamics simulation. For different tasks utilizing deep neural networks, a voxelization module is available. Voxelization takes the unit cell and converts it into discrete 3D voxels. These voxels are contain feature information such as number of particles in the voxel, their type, and charge.

Observation spaces can also compute the reward or “goodness" of the current state. Currently, MMGBSA.py is utilized to return the MMGBSA score of the previous simulation [Miller III et al., 2012, Roe and Cheatham III, 2013]. More complex reward spaces are available such as utilizing an equilibration sampling method [Chodera, 2016].

Action Spaces

The action space abstraction in RLMM defines the space, or domain, of available actions available to the policy module. These actions define the transition from state to state in RLMM. To illustrate the strengths of this abstraction, we provide three implemented action spaces, with more robust formulations for synthetic chemistry restrictions coming. In principle, the action space formulation will allow for a robotic laboratory based action space,

calculating possible reactions given a set of known reactions and in-stock reagents [Lindsey, 1992]. During lead optimization, the goal is to modify a ligand to something similar with more desirable properties such as stronger binding or other properties. In order to transition the ligand, we implemented a similarity search, where the action space returns the n most similar molecules in terms of 3D shape overlay based on a user provided database. The action space for a given state is then defines as the set of molecules that are the top n most similar from a given database, such as PubChem. One benefit of this module is that enumerating the actions is exceptionally fast and all actions are synthetically reasonably, at least up to the quality of the database used. Candidates are run through the OpenEye BlockBuster filter to remove those that are not sufficiently drug-like [OEChem, 2012a]. RLMM then selects some of the candidates to simulate based on the policy. After simulation, RLMM selects a subset of the simulated ligands for further modification.

The first action space uses the FastRocs toolkit from OpenEye, which utilizes parallel GPUs to search a local database for known active compounds of similar shape to the given ligand, comparing millions of potential compounds per second [Sheridan et al., 2008]. It returns a configurable number of sufficiently similar compounds for further analysis as potential modifications to the ligand.

The second action space, "MoleculeGrow," uses the formulation of a molecular action space from the MolDQN library [Zhou et al., 2019]. It "grows" the ligand by adding a new atom to it (and a corresponding bond), then checks that it is chemically valid. A new set of all possible valid modifications is produced and returned at each step. Successful modifications are grown further at later steps in the modeling process.

The third action space is based on the derivation and models trained in this paper for a scaffold-based navigation model. In this formulation there are two set ups depending on the action space format. The action space can be defined through game-like controls (Predecessor, Expand, Scaffold, Successor). The action space can also be defined through

sampling neighboring nodes in the hypergraph view of the scaffold-based conception of chemical space. In order to match the formulation of the previous two action spaces, the code for ScaffoldBased action space follows the latter formulation.

Policies

In each episode of the simulation, the ligand structure will be perturbed to look for better binding and/or new ligand structures. The changed structure will then persist as the base structure in the next episode. RLMM supports various policies to allow for flexible choices in how the ligand will be modified in each episode and which modifications will persist. Currently, the library implements random and expert policies. In the random policy space, a stochastic perturbation is made to the ligand and the updated ligand persists to the next episode. In contrast, the expert policy computes docking scores, then chooses the modified ligand with the best docking score to persist to the next episode.

Transformation of a JAK2-Kinase Decoy to a Lead Analog

A known decoy for the JAK2-kinase was retrieved from a decoy dataset, DUD-E[Mysinger et al., 2012a]. The decoy was docked utilizing FRED to a JAK2-kinase structure (PDB: 5AEP) [Brasca et al., 2015]. RLMM ran starting with the inital decoy structure utilizing an expert policy based on docking scores. The docking score policy took into account the various conformations explored during the simulations by averaging the score over a series of snapshots. The ligand was modified X times. The series of modifications is available in the SI, along with a video of snapshots from the simulation including the alternating ligands. The ligand with the lowest MMGBSA score was pulled and compared to similarity to known active compounds.

Optimizing a lead for SARS-CoV-2 Main Protease

A lead for the main protease was located with a publicly available crystal structure (PDB: 7LTJ)[Clyde et al., 2021d]. The crystal structure was obtained, cleaned by removing waters and ions, and separated into protein and ligand. The system builder was set to model explicit waters. Each cycle of RLMM simulated the proposed ligand in complex with the main protease for 1 nanosecond. After each nanosecond of simulation, the trajectory was analyzed by computing the MMGBSA score utilizing MMGBSA.py. The local scaffold area of the ligand was enumerated, and each proposed new molecule was scored based on its fitness judged by the policy. The policy utilized selected the ligand to simulate next by computing docking scores against protein conformations seen so far. The system was run for 100 steps, thus simulating 100 different ligands (and running collectively 100ns of simulation).

9.1.3 Discussion

RLMM is a flexible end-to-end environment for lead optimization. RLMM combines information from molecular dynamics simulations, generative models for exploring chemical space, and a policy module to navigate drug design spaces with physics-based simulations. By its flexible gym-like design, RLMM can be utilized for lead optimization and constrained by the users for particular use cases.

One challenge of building an end-to-end drug design system formulated as a Markov decision process was the action space. Previous action space designs have forced non-synthetically accessible action spaces based on generative models or been unstructured[Zhou et al., 2019, Popova et al., 2018]. The scaffold-based approach allows for two formulations reinforcement learning, first based on a discrete action space of growing or contracting a scaffold, and second

RLMM is well positioned for deployment in future robotic laboratories. Robotic laboratories can provide rapid assay screening capabilities to RLMM[Gromski et al., 2020]. RLMM,

while searching through a particular chemical space, can send successful or interesting compounds to the lab for testing. The feedback from the test can be used to fine tune the scoring model in RLMM. Given the flexibility of the action space design (for ligand design), synthetic accessible libraries or even in-stock libraries can constrain the space so RLMM only searches through compounds that the robotic laboratory has in-stock or can produce through its reaction capabilities.

RLMM's integrated design has utility outside of molecular design. Peptides, protein, and antibody therapeutics have computational methods for simulation and binding affinity characterization [Vanhee et al., 2011]. In-silico peptide design can be made possible with RLMM. Peptides can be simulated, and modifications can be ranked and proposed based on the data from the simulations (such as mutating, adding, or deleting a residue). To modify RLMM for peptides, the simulation module would need to be modified to allow a new protocol for simulation which is more suitable for peptide design [Raut et al., 2005]. Future work will follow up with peptide design computational and experimental results.

We present Reinforcement Learning for Molecular Modeling (RLMM), an RL gym-like environment for lead optimization. RLMM brings together multiple efforts from molecular modeling to de-novo drug design into a single toolkit aimed to accelerate research in-silico drug design and autonomous discovery. We present a paradigmatic case of a lead optimisation from prior work for the 3CL-main protease from SARS-CoV-2 to illustrate the system moving molecules towards known protease inhibitor designs, as well as creating interesting analog series to assist the medicinal chemist in the lead-optimization process. Future work is aimed at extending the system towards peptide design and including detailed experimental characterization.

9.2 LLMs as Generative Databases

Recent success of large language models (LLMs) generative abilities has overtaken and excited many new areas. Much of the debate around their utility and lasting-potential as a driver of AI developed revolves around the contrast between data-regurgitation and novelty. The arguments may go: if LLMs are stochastic parrots, then what utility can they actually have? While this is an interesting research inquiry, this line of argumentation assumes that stochastic parrots are not interesting computational objects of study. In this article, I take on a different assumption and ask: in what ways might a parrot be useful?

I argue that LLMs conceptualized as a novel database tool are fruitful for three reasons. First, certain kinds of database are not amendable to extremely large data (such as graph databases) and LLMs may be able to provide a speed up both efficacy and memory. Second, through theorizing LLMs as databases, one can explore the idea of generative versus non-generative databases, provide deep connections for reinforcement-learning, and on-the-fly generation and exploration of databases without full enumeration. Finally, the evaluation of LLMs in situations where fact-grounding is necessary, such as in AI for Science and Security (AI4SS) or recent EU regulations for high-impact situations, taking a database first approach to LLMs unravels their scrutability. Rather than imagining LLMs are generative models, we theorize them as retrieval models with some unique properties. While such an approach is not ammenable to all database types (such as those with unclear generative functions, such as social networks), domain-specific knowledge graphs are in scope. Future work in hybrid approaches between databases and LLMs can be developed for these cases.

9.3 HPC and Drug Design

Future computing systems are being designed with AI for drug discovery in mind. Everything from national super computing infrastructure to hardware accelerators are focused on virtual

screening as outlined in this dissertation; however, as argued in chapter 8, it is unlikely these approaches will deliver new orders of magnitude acceleration in our ability to screen larger libraries and do so more accurately.

In large, the problem is calibration and accuracy of the underlying methods for understanding binding free energy. Current methods such as virtual screening do not take into account experimental data, AI models are typically bootstrapped from cheap scoring functions, and even simulations in staged workflows are based often on poor molecular force-fields. Each of these challenges can be overcome, but it will require expanding the basic notion of virtual ligand screening outlined in chapters 3 to 6 from a naive screening problem to a rich problem of interrelated samples as outlined in chapter 7.

REFERENCES

Enamine Hit Locator Library 2018. URL https://enamine.net/hit-finding/diversity-libraries/hit-locator-library-300.

URL http://pubs.acs.org/books/references.shtml.

Communication from the european commission to the european council and the european parliament: 20 20 by 2020: Europe's climate change opportunity. Technical report, Brussels, Belgium, 2008.

Angel Abarca, Pilar Gómez-Sal, Avelino Martín, Miguel Mena, Josep María Poblet, and Carlos Yélamos. Ammonolysis of mono(pentamethylcyclopentadienyl) titanium(IV) derivatives. *Inorg. Chem.*, 39(4):642–651, 2000. doi:10.1021/ic9907718.

Colin D. Abernethy, Gareth M. Codd, Mark D. Spicer, and Michelle K. Taylor. A highly stable N-heterocyclic carbene complex of trichloro-oxo-vanadium(V) displaying novel Cl—C(carbene) bonding interactions. *J. Am. Chem. Soc.*, 125(5):1128–1129, 2003. doi:10.1021/ja0276321.

Yasmin Abo-Zeid, Nasser SM Ismail, Gary R McLean, and Nadia M Hamdy. A molecular docking study repurposes FDA approved iron oxide nanoparticles to treat and control COVID-19 infection. *European Journal of Pharmaceutical Sciences*, 153:105465, 2020.

Atanu Acharya, Rupesh Agarwal, MB Baker, Jerome Baudry, Debsindhu Bhowmik, Swen Boehm, KG Byler, SY Chen, L Coates, Connor J Cooper, et al. Supercomputer-based ensemble docking drug discovery pipeline with application to COVID-19. *Journal of Chemical Information and Modeling*, 60(12):5832–5852, 2020.

Hagit Achdout, Anthony Aimon, Elad Bar-David, Haim Barr, Amir Ben-Shmuel, James Bennett, Melissa L Bobby, Juliane Brun, Sarma BVNBS, Mark Calmiano, et al. COVID Moonshot: Open science discovery of SARS-CoV-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning, 2020a. bioRxiv.

Hagit Achdout, Anthony Aimon, Elad Bar-David, Haim Barr, Amir Ben-Shmuel, James Bennett, Melissa L Bobby, Juliane Brun, BVNBS Sarma, Mark Calmiano, Anna Carbery, Emma Cattermole, John D. Chodera, Austin Clyde, Joseph E. Coffland, Galit Cohen, Jason Cole, Alessandro Contini, Lisa Cox, Milan Cvitkovic, Alex Dias, Alice Douangamath, Shirly Duberstein, Tim Dudgeon, Louise Dunnett, Peter K. Eastman, Noam Erez, Michael Fairhead, Daren Fearon, Oleg Fedorov, Matteo Ferla, Holly Foster, Richard Foster, Ronen Gabizon, Paul Gehrtz, Carina Gileadi, Charline Giroud, William G. Glass, Robert Glen, Itai Glinert, Marian Gorichko, Tyler Gorrie-Stone, Edward J Griffen, Jag Heer, Michelle Hill, Sam Horrell, Matthew F.D. Hurley, Tomer Israely, Andrew Jajack, Eric Jnoff, Tobias John, Anastassia L. Kantsadi, Peter W. Kenny, John L. Kiappes, Lizbe Koekemoer, Boris Kovar, Tobias Krojer, Alpha Albert Lee, Bruce A. Lefker, Haim Levy,

Nir London, Petra Lukacik, Hannah Bruce Macdonald, Beth MacLean, Tika R. Malla, Tatiana Matviiuk, Willam McCorkindale, Sharon Melamed, Oleg Michurin, Halina Mikolajek, Aaron Morris, Garrett M. Morris, Melody Jane Morwitzer, Demetri Moustakas, Jose Brandao Neto, Vladas Oleinikovas, Gijs J. Overheul, David Owen, Ruby Pai, Jin Pan, Nir Paran, Benjamin Perry, Maneesh Pingle, Jakir Pinjari, Boaz Politi, Ailsa Powell, Vladimir Psenak, Reut Puni, Victor L. Rangel, Rambabu N. Reddi, St Patrick Reid, Efrat Resnick, Matthew C. Robinson, Ralph P. Robinson, Dominic Rufa, Christopher Schofield, Aarif Shaikh, Jiye Shi, Khriesto Shurrush, Assa Sittner, Rachael Skyner, Adam Smalley, Mihaela D. Smilova, John Spencer, Claire Strain-Damerell, Vishwanath Swamy, Hadas Tamir, Rachael Tennant, Andrew Thompson, Warren Thompson, Susana Tomasio, Anthony Tumber, Ioannis Vakonakis, Ronald P. van Rij, Finny S. Varghese, Mariana Vaschetto, Einat B. Vitner, Vincent Voelz, Annette von Delft, Frank von Delft, Martin Walsh, Walter Ward, Charlie Weatherall, Shay Weiss, Conor Francis Wild, Matthew Wittmann, Nathan Wright, Yfat Yahalom-Ronen, Daniel Zaidmann, Hadeer Zidane, and Nicole Zitzmann. Covid moonshot: Open science discovery of sars-cov-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning. *bioRxiv*, 2020b. doi:10.1101/2020.10.29.339317. URL `https://www.biorxiv.org/content/early/2020/10/30/2020.10.29.339317`.

Hagit Achdout, Anthony Aimon, Elad Bar-David, Haim Barr, Amir Ben-Shmuel, James Bennett, Melissa L Bobby, Juliane Brun, BVNBS Sarma, Mark Calmiano, et al. Covid moonshot: open science discovery of sars-cov-2 main protease inhibitors by combining crowdsourcing, high-throughput experiments, computational simulations, and machine learning. *BioRxiv*, 2020c.

Paul D Adams, Pavel V Afonine, Gábor Bunkóczi, Vincent B Chen, Ian W Davis, Nathaniel Echols, Jeffrey J Headd, L-W Hung, Gary J Kapral, Ralf W Grosse-Kunstleve, et al. PHENIX: A comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.*, 66(2):213–221, 2010.

Steve Agajanian, Odeyemi Oluyemi, and Gennady M. Verkhivker. Integration of Random Forest Classifiers and Deep Convolutional Neural Networks for Classification and Biomolecular Modeling of Cancer Driver Mutations. *Frontiers in Molecular Biosciences*, 6:44, 1 2019. ISSN 2296-889X. doi:10.3389/fmolb.2019.00044.

Sajjad Ahmad, Hyder Wajid Abbasi, Sara Shahid, Sana Gul, and Sumra Wajid Abbasi. Molecular docking, simulation and MM-PBSA studies of *nigella sativa* compounds: A computational quest to identify potential natural antiviral for COVID-19 treatment. *Journal of Biomolecular Structure and Dynamics*, 0(0):1–9, 2020.

Qurrat Ul Ain, Antoniya Aleksandrova, Florian D Roessler, and Pedro J Ballester. Machine-learning scoring functions to improve structure-based binding affinity prediction and virtual screening. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(6): 405–424, 2015.

Matteo Aldeghi, Vytautas Gapsys, and Bert L. de Groot. Predicting kinase inhibitor resistance: Physics-based and data-driven approaches. *ACS Central Science*, 5(8):1468–1474, 2019. doi:10.1021/acscentsci.9b00590. URL `https://doi.org/10.1021/acscentsci.9b00590`.

Michael P Allen et al. Introduction to molecular dynamics simulation. *Computational soft matter: from synthetic polymers to proteins*, 23(1):1–28, 2004.

Raid Alzubi, Naeem Ramzan, Hadeel Alzoubi, and Stamos Katsigiannis. SNPs-based Hypertension Disease Detection via Machine Learning Techniques. *2018 24th International Conference on Automation and Computing (ICAC)*, pages 1–6, 2018. doi:10.23919/iconac.2018.8748972.

Rommie E Amaro, Riccardo Baron, and J Andrew McCammon. An improved relaxed complex scheme for receptor flexibility in computer-aided drug design. *Journal of computer-aided molecular design*, 22(9):693–705, 2008.

Amy C Anderson. The process of structure-based drug design. *Chemistry & biology*, 10(9): 787–797, 2003.

Chris Anderson. The end of theory: The data deluge makes the scientific method obsolete. *Wired magazine*, 16(7):16–07, 2008.

E. Anderson, G.D. Veith, D. Weininger, and Minn.) Environmental Research Laboratory (Duluth. *SMILES, a Line Notation and Computerized Interpreter for Chemical Structures*. Environmental research brief. U.S. Environmental Protection Agency, Environmental Research Laboratory, 1987. URL `https://books.google.com/books?id=DiFetwEACAAJ`.

Leah N. Appelhans, Daniele Zuccaccia, Anes Kovacevic, Anthony R. Chianese, John R. Miecznikowski, Aleco Macchioni, Eric Clot, Odile Eisenstein, and Robert H. Crabtree. An anion-dependent switch in selectivity results from a change of C—H activation mechanism in the reaction of an imidazolium salt with irh5(pph3)2. *J. Am. Chem. Soc.*, 127(46): 16299–16311, 2005. doi:10.1021/ja055317j.

Johan Åqvist, Carmen Medina, and Jan-Erik Samuelsson. A new method for predicting binding affinity in computer-aided drug design. *Protein Engineering, Design and Selection*, 7(3):385–391, 1994.

Anthony J. Arduengo, III, H. V. Rasika Dias, Richard L. Harlow, and Michael Kline. Electronic stabilization of nucleophilic carbenes. *J. Am. Chem. Soc.*, 114(14):5530–5534, 1992. doi:10.1021/ja00040a007.

Anthony J. Arduengo, III, Siegfried F. Gamper, Joseph C. Calabrese, and Fredric Davidson. Low-coordinate carbene complexes of nickel(0) and platinum(0). 116(10):4391–4394, 1994. doi:10.1021/ja00089a029.

Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of cheminformatics*, 11(1):1–13, 2019.

Josep Arús-Pous, Atanas Patronov, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Smiles-based deep generative scaffold decorator for de-novo drug design. *Journal of cheminformatics*, 12:1–18, 2020.

Jeremy Ash and Denis Fourches. Characterizing the chemical space of erk2 kinase inhibitors using descriptors computed from molecular dynamics trajectories. *J. Chem. Inf. Model.*, 57(6):1286–1299, 2017. doi:10.1021/acs.jcim.7b00048. PMID: 28471171.

Francisco Azuaje, Tony Kaoma, Céline Jeanty, Petr V. Nazarov, Arnaud Muller, Sang-Yoon Kim, Gunnar Dittmar, Anna Golebiewska, and Simone P. Niclou. Hub genes in a pan-cancer co-expression network show potential for predicting drug responses. *F1000Research*, 7:1906, 2018. doi:10.12688/f1000research.17149.1.

Open Babel. The open source chemistry toolbox, 2010.

Yadu Babuji, Ben Blaiszik, Tom Brettin, Kyle Chard, Ryan Chard, Austin Clyde, Ian Foster, Zhi Hong, Shantenu Jha, Zhuozhao Li, and others. Targeting SARS-CoV-2 with AI-and HPC-enabled Lead Generation: A First Data Release. *arXiv preprint arXiv:2006.02431*, 2020a.

Yadu Babuji, Ben Blaiszik, Tom Brettin, Kyle Chard, Ryan Chard, Austin Clyde, Ian Foster, Zhi Hong, Shantenu Jha, Zhuozhao Li, et al. Targeting sars-cov-2 with ai-and hpc-enabled lead generation: a first data release. *arXiv preprint arXiv:2006.02431*, 2020b.

Jonathan B. Baell and Georgina A. Holloway. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.*, 53(7):2719–2740, 2010. doi:10.1021/jm901137j.

Fabienne Baffert, Catherine H Régnier, Alain De Pover, Carole Pissot-Soldermann, Gisele A Tavares, Francesca Blasco, Josef Brueggen, Patrick Chène, Peter Drueckes, Dirk Erdmann, et al. Potent and selective inhibition of polycythemia by the quinoxaline jak2 inhibitor nvp-bsk805. *Molecular cancer therapeutics*, 9(7):1945–1955, 2010.

Suhrid Balakrishnan and Sumit Chopra. Collaborative ranking. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 143–152. ACM, 2012.

V. Balasubramanian, A. Treikalis, O. Weidner, and S. Jha. Ensemble toolkit: Scalable and flexible execution of ensembles of tasks. In *2016 45th International Conference on Parallel Processing (ICPP)*, volume 00, pages 458–463, Aug. 2016. doi:10.1109/ICPP.2016.59. URL `doi.ieeecomputersociety.org/10.1109/ICPP.2016.59`.

Vivek Balasubramanian, Matteo Turilli, Weiming Hu, Matthieu Lefebvre, Wenjie Lei, Ryan Modrak, Guido Cervone, Jeroen Tromp, and Shantenu Jha. Harnessing the power of many: Extensible toolkit for scalable ensemble applications. In *2018 IEEE international parallel and distributed processing symposium (IPDPS)*, pages 536–545. IEEE, 2018.

Vivek Balasubramanian, Shantenu Jha, Andre Merzky, and Matteo Turilli. Radical-cybertools: Middleware building blocks for scalable science. *arXiv preprint arXiv:1904.03085*, 2019.

Pedro J Ballester and John BO Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9): 1169–1175, 2010. Publisher: Oxford University Press.

Riccardo Baron and J Andrew McCammon. (thermo) dynamic role of receptor flexibility, entropy, and motional correlation in protein–ligand binding. *Chemphyschem: a European journal of chemical physics and physical chemistry*, 9(7):983–988, 2008.

Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A. Margolin, Sungjoon Kim, Christopher J. Wilson, Joseph Lehár, Gregory V. Kryukov, Dmitriy Sonkin, Anupama Reddy, Manway Liu, Lauren Murray, Michael F. Berger, John E. Monahan, Paula Morais, Jodi Meltzer, Adam Korejwa, Judit Jané-Valbuena, Felipa A. Mapa, Joseph Thibault, Eva Bric-Furlong, Pichai Raman, Aaron Shipway, Ingo H. Engels, Jill Cheng, Guoying K. Yu, Jianjun Yu, Peter Aspesi, Melanie de Silva, Kalpana Jagtap, Michael D. Jones, Li Wang, Charles Hatton, Palesc, Emanuele olo, Supriya Gupta, Scott Mahan, Carrie Sougnez, Robert C. Onofrio, Ted Liefeld, Laura MacConaill, Wendy Winckler, Michael Reich, Nanxin Li, Jill P. Mesirov, Stacey B. Gabriel, Gad Getz, Kristin Ardlie, Vivien Chan, Vic E. Myer, Barbara L. Weber, Jeff Porter, Markus Warmuth, Peter Finan, Jennifer L. Harris, Matthew Meyerson, Todd R. Golub, Michael P. Morrissey, William R. Sellers, Robert Schlegel, and Levi A. Garraway. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483:603—607, 2012. ISSN 0028-0836, 1476-4687. doi:10.1038/nature11003. URL `http://www.nature.com/articles/nature11003`.

Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816, 2017.

Rohit Batra, Henry Chan, Ganesh Kamath, Rampi Ramprasad, Mathew J Cherukara, and Subramanian KRS Sankaranarayanan. Screening of therapeutic agents for COVID-19 using machine learning and ensemble docking studies. *The Journal of Physical Chemistry Letters*, 11(17):7058–7065, 2020.

José E Belizário, Beatriz A Sangiuliano, Marcela Perez-Sosa, Jennifer M Neyra, and Dayson F Moreira. Using pharmacogenomic databases for discovering patient-target genes and small molecule candidates to cancer therapy. *Frontiers in pharmacology*, 7:312, 2016.

Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

Andreas Bender and Robert C Glen. A discussion of measures of enrichment in virtual screening: comparing the information content of descriptors with increasing levels of sophistication. *Journal of chemical information and modeling*, 45(5):1369–1375, 2005.

Noah E. Berlow, Rishi Rikhi, Mathew Geltzeiler, Jinu Abraham, Matthew N. Svalina, Lara E. Davis, Erin Wise, Maria Mancini, Jonathan Noujaim, Atiya Mansoor, Michael J. Quist, Kevin L. Matlock, Martin W. Goros, Brian S. Hernandez, Yee C. Doung, Khin Thway, Tomohide Tsukahara, Jun Nishio, Elaine T. Huang, Susan Airhart, Carol J. Bult, Regina Gandour-Edwards, Robert G. Maki, Robin L. Jones, Joel E. Michalek, Milan Milovancev, Souparno Ghosh, Ranadip Pal, and Charles Keller. Probabilistic modeling of personalized drug combinations from integrated chemical screen and molecular data in sarcoma. *BMC Cancer*, 19:593, 6 2019. doi:10.1186/s12885-019-5681-6.

Agastya P Bhati, Shunzhou Wan, Dario Alfè, Austin R Clyde, Mathis Bode, Li Tan, Mikhail Titov, Andre Merzky, Matteo Turilli, Shantenu Jha, et al. Pandemic drugs at pandemic speed: infrastructure for accelerating covid-19 drug discovery with hybrid machine learning-and physics-based simulations on high-performance computers. *Interface focus*, 11(6):20210018, 2021.

Mateusz K Bieniek, Agastya P Bhati, Shunzhou Wan, and Peter V Coveney. Ties 20: Relative binding free energy with a flexible superimposition algorithm and partial ring morphing. *Journal of chemical theory and computation*, 17(2):1250–1265, 2021.

Konrad H Bleicher, Hans-Joachim Böhm, Klaus Müller, and Alexander I Alanine. Hit and lead generation: beyond high-throughput screening. *Nature reviews Drug discovery*, 2(5): 369–378, 2003.

Margaret A Boden. 4 gofai. *The Cambridge handbook of artificial intelligence*, page 89, 2014.

Regine S Bohacek, Colin McMartin, and Wayne C Guida. The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal research reviews*, 16(1): 3–50, 1996a.

Regine S Bohacek, Colin McMartin, and Wayne C Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1): 3–50, 1996b.

Lydia Boike, Nathaniel J Henning, and Daniel K Nomura. Advances in covalent drug discovery. *Nature Reviews Drug Discovery*, pages 1–18, 2022.

Michael R. Boyd and Kenneth D. Paull. Some practical considerations and applications of the national cancer institute in vitro anticancer drug discovery screen. *Drug Development Research*, 34(2):91–109, February 1995. ISSN 0272-4391, 1098-2299. doi:10.1002/ddr.430340203. URL `http://doi.wiley.com/10.1002/ddr.430340203`.

Ken A Brameld, Bernd Kuhn, Deborah C Reuter, and Martin Stahl. Small molecule conformational preferences derived from crystal structure data. a medicinal chemistry focused analysis. *Journal of chemical information and modeling*, 48(1):1–24, 2008.

Maria Gabriella Brasca, Paola Gnocchi, Marcella Nesi, Nadia Amboldi, Nilla Avanzi, Jay Bertrand, Simona Bindi, Giulia Canevari, Daniele Casero, Marina Ciomei, et al. Novel pyrrole carboxamide inhibitors of jak2 as potential treatment of myeloproliferative disorders. *Bioorganic & Medicinal Chemistry*, 23(10):2387–2407, 2015.

Kyle R Brimacombe, Tongan Zhao, Richard T Eastman, Xin Hu, Ke Wang, Mark Backus, Bolormaa Baljinnyam, Catherine Z Chen, Lu Chen, Tara Eicher, et al. An opendata portal to share covid-19 drug repurposing data in real time. *BioRxiv*, 2020.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Michelle V Buchanan and Stephen Streiffer. Nvbl (national virtual biotechnology laboratory) overview. Technical report, USDOE Office of Science (SC)(United States), 2020.

Rosa Buonfiglio, Maurizio Recanatini, and Matteo Masetti. Protein flexibility in drug discovery: From theory to computation. *ChemMedChem*, 10(7):1141–1148, 2015.

Jonathon Byrd and Zachary C Lipton. What is the Effect of Importance Weighting in Deep Learning? 2018.

Gary W Caldwell, David M Ritchie, John A Masucci, William Hageman, and Zhengyin Yan. The new pre-preclinical paradigm: compound optimization in early and late phase drug discovery. *Current topics in medicinal chemistry*, 1(5):353–366, 2001.

Myrna Candelaria, Lucia Taja-Chayeb, Claudia Arce-Salinas, Silvia Vidal-Millan, Alberto Serrano-Olvera, and Alfonso Duenas-Gonzalez. Genetic determinants of cancer drug efficacy and toxicity: practical considerations and perspectives. *Anti-cancer drugs*, 16(9): 923–933, 2005.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2018.

Xuetao Cao. COVID-19: Immunopathology and its implications for therapy. *Nature Reviews Immunology*, 20(5):269–270, 2020. doi:10.1038/s41577-020-0308-3. URL `https://doi.org/10.1038/s41577-020-0308-3`.

Yiqun Cao, Tao Jiang, and Thomas Girke. A maximum common substructure-based algorithm for searching and predicting drug-like compounds. *Bioinformatics*, 24(13):i366–i374, 2008.

Graeme L Card, Landy Blasdel, Bruce P England, Chao Zhang, Yoshihisa Suzuki, Sam Gillette, Daniel Fong, Prabha N Ibrahim, Dean R Artis, Gideon Bollag, et al. A family of phosphodiesterase inhibitors discovered by cocrystallography and scaffold-based drug design. *Nature biotechnology*, 23(2):201–207, 2005.

Lorenzo Casalino, Abigail C Dommer, Zied Gaieb, Emilia P Barros, Terra Sztain, Surl-Hee Ahn, Anda Trifan, Alexander Brace, Anthony T Bogetti, Austin Clyde, et al. Ai-driven multiscale simulations illuminate mechanisms of sars-cov-2 spike dynamics. *The International Journal of High Performance Computing Applications*, 35(5):432–451, 2021.

David A Case, H Metin Aktulga, Kellon Belfon, Ido Ben-Shalom, Scott R Brozell, David S Cerutti, Thomas E Cheatham III, Vinícius Wilian D Cruzeiro, Tom A Darden, Robert E Duke, et al. *Amber 2021*. University of California, San Francisco, 2021a.

David A Case, H Metin Aktulga, Kellon Belfon, Ido Ben-Shalom, Scott R Brozell, David S Cerutti, Thomas E Cheatham III, Vinícius Wilian D Cruzeiro, Tom A Darden, Robert E Duke, et al. *Amber 2021*. University of California, San Francisco, 2021b.

Claudio N Cavasotto and Juan I Di Filippo. *In silico* drug repurposing for COVID-19: Targeting SARS-CoV-2 proteins through docking and consensus ranking. *Molecular Informatics*, 40(1):2000115, 2021.

E Cayley. Ueber die analytischen figuren, welche in der mathematik bäume genannt werden und ihre anwendung auf die theorie chemischer verbindungen. *Berichte der deutschen chemischen Gesellschaft*, 8(2):1056–1059, 1875.

Remzi Celebi, Oliver Bear Don't Walk, Rajiv Movva, Semih Alpsoy, and Michel Dumontier. In-silico Prediction of Synergistic Anti-Cancer Drug Combinations Using Multi-omics Data. *Scientific Reports*, 9:8949, 6 2019. doi:10.1038/s41598-019-45236-6.

Tim Cernak. A machine with chemical intuition. *Chem*, 4(3):401–403, 2018.

Tim Cernak, Kevin D Dykstra, Sriram Tyagarajan, Petr Vachal, and Shane W Krska. The medicinal chemist's toolbox for late stage functionalization of drug-like molecules. *Chemical Society Reviews*, 45(3):546–576, 2016.

Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

Ji-Wei Chang, Yuduan Ding, Muhammad Tahir ul Qamar, Yin Shen, Junxiang Gao, and Ling-Ling Chen. A deep learning model based on sparse auto-encoder for prioritizing cancer-related genes and drug target combinations. *Carcinogenesis*, 40:624–632, 7 2019. ISSN 0143-3334. doi:10.1093/carcin/bgz044.

Yoosup Chang, Hyejin Park, Hyun-Jin Yang, Seungju Lee, Kwee-Yum Lee, Tae Soon Kim, Jongsun Jung, and Jae-Min Shin. Cancer drug response profile scan (cdrscan): a deep learning model that predicts drug effectiveness from cancer genomic signature. *Scientific reports*, 8(1):8857, 2018.

Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630. ACM, 2009.

Boxing Chen and Roland Kuhn. Amber: A modified bleu, enhanced ranking metric. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 71–77, 2011.

Chao Chen, Kay Grennan, Judith Badner, Dandan Zhang, Elliot Gershon, Li Jin, and Chunyu Liu. Removing Batch Effects in Analysis of Expression Microarray Data: An Evaluation of Six Batch Adjustment Methods. *PLoS ONE*, 6:e17238, 2011. doi:10.1371/journal.pone.0017238.

Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel A Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.*, 66(1):12–21, 2010.

Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.

Yiran Chen, Yuan Xie, Linghao Song, Fan Chen, and Tianqi Tang. A survey of accelerator architectures for deep neural networks. *Engineering*, 6(3):264–274, 2020.

Chakra Chennubhotla and Ivet Bahar. Signal propagation in proteins and relation to equilibrium fluctuations. *PLoS Comput Biol*, 3(9):e172, 2007.

Raymond Cheong, Chiaochun Joanne Wang, and Andre Levchenko. High content cell screening in a microfluidic device. *Molecular & Cellular Proteomics*, 8(3):433–442, 2009.

John Chodera, Andrea Rizzi, Levi Naden, Kyle Beauchamp, Patrick Grinaway, Josh Fass, Alex Wade, Bas Rustenburg, Gregory A. Ross, Andreas Krämer, Hannah Bruce Macdonald, Jaime Rodríguez-Guerra, Dominicrufa, Andy Simmonett, David W.H. Swenson, Hb0402, Mike Henry, Sander Roet, and Ana Silveira. choderalab/openmmtools: 0.20.3 bugfix release, 2021. URL `https://zenodo.org/record/4639586`.

John D Chodera. A simple method for automated equilibration detection in molecular simulations. *Journal of chemical theory and computation*, 12(4):1799–1805, 2016.

John D Chodera and Michael R Shirts. Replica exchange and expanded ensemble simulations as gibbs sampling: Simple improvements for enhanced mixing. *The Journal of chemical physics*, 135(19):194110, 2011.

François Chollet et al. Keras. `https://keras.io`, 2015.

Alex M Clark, Paul Labute, and Martin Santavy. 2d structure depiction. *Journal of chemical information and modeling*, 46(3):1107–1123, 2006.

David E Clark. What has virtual screening ever done for drug discovery? *Expert opinion on drug discovery*, 3(8):841–851, 2008.

Robert D Clark, Alexander Strizhev, Joseph M Leonard, James F Blake, and James B Matthew. Consensus scoring for ligand/protein interactions. *Journal of Molecular Graphics and Modelling*, 20(4):281–295, 2002. Publisher: Elsevier.

Ann E Cleves and Ajay N Jain. Effects of inductive bias on computational evaluations of ligand-based modeling and on drug discovery. *Journal of computer-aided molecular design*, 22(3-4):147–159, 2008.

Austin Clyde. Ai for science and global citizens. *Patterns*, 3(2):100446, 2022a.

Austin Clyde. Ultrahigh throughput protein–ligand docking with deep learning. In *Artificial Intelligence in Drug Design*, pages 301–319. Springer, 2022b.

Austin Clyde, Dave Wright, and Shantenu Jha. Cafcw 120 integrating high-performance simulations and learning toward improved cancer therapy. 2019.

Austin Clyde, Tom Brettin, Alexander Partin, Maulik Shaulik, Hyunseung Yoo, Yvonne Evrard, Yitan Zhu, Fangfang Xia, and Rick Stevens. A systematic approach to featurization for cancer drug sensitivity predictions with deep learning. *arXiv preprint arXiv:2005.00095*, 2020a.

Austin Clyde, Xiaotian Duan, and Rick Stevens. Regression enrichment surfaces: a simple analysis technique for virtual drug screening models. *arXiv preprint arXiv:2006.01171*, 2020b.

Austin Clyde, Xiaotian Duan, and Rick Stevens. Regression Enrichment Surfaces: a Simple Analysis Technique for Virtual Drug Screening Models. *arXiv preprint arXiv:2006.01171*, 2020c.

Austin Clyde, Arvind Ramanathan, and Rick Stevens. Virtual screening with deep learning using cancer cell line dose-response data. *Clinical Cancer Research*, 26 (12_Supplement_1):36–36, 2020d.

Austin Clyde, Arvind Ramanathan, and Rick Stevens. *Virtual screening with deep learning using cancer cell line dose-response data*. American Association for Cancer Research, 2020e.

Austin Clyde, Thomas Brettin, Alexander Partin, Hyunseung Yoo, Yadu Babuji, Ben Blaiszik, Andre Merzky, Matteo Turilli, Shantenu Jha, Arvind Ramanathan, and Rick Stevens. Protein-ligand docking surrogate models: A sars-cov-2 benchmark for deep learning accelerated virtual screening, 2021a. URL `https://doi.org/10.26311/BFKY-EX6P`.

Austin Clyde, Thomas Brettin, Alexander Partin, Hyunseung Yoo, Yadu Babuji, Ben Blaiszik, Andre Merzky, Matteo Turilli, Shantenu Jha, Arvind Ramanathan, et al. Protein-ligand docking surrogate models: A sars-cov-2 benchmark for deep learning accelerated virtual screening. *arXiv preprint arXiv:2106.07036*, 2021b.

Austin Clyde, Stephanie Galanie, Daniel W. Kneller, Heng Ma, Yadu Babuji, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, Leighton Coates, Ian Foster, Darin Hauner, Vilmos Kertesz, Neeraj Kumar, Hyungro Lee, Zhuozhao Li, Andre Merzky, Jurgen G. Schmidt, Li Tan, Mikhail Titov, Anda Trifan, Matteo Turilli, Hubertus Van Dam, Srinivas C. Chennubhotla, Shantenu Jha, Andrey Kovalevsky, Arvind Ramanathan, Marti Head, and Rick Stevens. High throughput virtual screening and validation of a sars-cov-2 main protease non-covalent inhibitor. *bioRxiv*, 2021c. doi:10.1101/2021.03.27.437323. URL `https://www.biorxiv.org/content/early/2021/03/27/2021.03.27.437323`.

Austin Clyde, Stephanie Galanie, Daniel W Kneller, Heng Ma, Yadu Babuji, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, et al. High-throughput virtual screening and validation of a sars-cov-2 main protease noncovalent inhibitor. *Journal of chemical information and modeling*, 62(1):116–128, 2021d.

Austin Clyde, Stephanie Galanie, Daniel W. Kneller, Heng Ma, Vilmos Kertesz, Leighton Coates, Alexander Brace, Anda Trifan, Ben Blaiszik, Kyle Chard, Ryan Chard, Zhuozhou Li, Yadu Babuji, Ian Foster, Thomas Brettin, Hyungro Lee, Andre Merzky, Mikhail Titov, Matteo Turilli, Li Tan, Hubertus J J van Dam, Jurgen G. Schmidt, Shantenu Jha, Andrey Kovalevsky, Arvind Ramanathan, Marti Head, and Rick Stevens. High Throughput Virtual Screening and Validation of a SARS-CoV-2 Main Protease Non-Covalent Inhibitor. 3 2021e. doi:10.6084/m9.figshare.13641101.v1.

Austin Clyde, Bharat Kale, Maoyuan Sun, Michael Papka, Arvind Ramanathan, and Rick Stevens. Scaffold embeddings: Learning the structure spanned by chemical fragments, scaffolds and compounds. In *Workshop on Learning Meaningful Representation of Life*, 2021f.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

Anne M. Coghill and Lorrin R. Garson, editors. *The ACS Style Guide*. Oxford University Press, Inc. and The American Chemical Society, New York, 3 edition, 2006.

Jason C Cole, Christopher W Murray, J Willem M Nissink, Richard D Taylor, and Robin Taylor. Comparing protein–ligand docking programs is difficult. *Proteins: Structure, Function, and Bioinformatics*, 60(3):325–332, 2005.

Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

Federico Comitani and Francesco L Gervasio. Modeling ligand–target binding with enhanced sampling simulations. *Biomolecular Simulations in Structure-Based Drug Discovery*, pages 43–66, 2018.

Michael M Cone, Rengachari Venkataraghavan, and Fred W McLafferty. Computer-aided interpretation of mass spectra. 20. molecular structure comparison program for the identification of maximal common substructures. *Journal of the American Chemical Society*, 99(23):7668–7671, 1977.

Miles Congreve, Robin Carr, Chris Murray, and Harren Jhoti. A'rule of three'for fragment-based lead discovery? *Drug discovery today*, 8(19):876–877, 2003.

William S. Cooper, Fredric C. Gey, and Daniel P. Dabney. Probabilistic retrieval based on staged logistic regression. pages 198–210, 1992. doi:10.1145/133160.133199.

Isidro Cortés-Ciriano and Andreas Bender. KekuleScope: prediction of cancer cell line sensitivity and compound potency using convolutional neural networks trained on compound images. *Journal of Cheminformatics*, 11, 2019. doi:10.1186/s13321-019-0364-5.

David Cossock and Tong Zhang. Subset ranking using regression. In *International Conference on Computational Learning Theory*, pages 605–619. Springer, 2006.

James C Costello, Laura M Heiser, Elisabeth Georgii, Mehmet Gönen, Michael P Menden, Nicholas J Wang, Mukesh Bansal, Muhammad Ammad-ud din, Petteri Hintsanen, Suleiman A Khan, John-Patrick Mpindi, Olli Kallioniemi, Antti Honkela, Tero Aittokallio, Krister Wennerberg, NCI DREAM Community, Jean-Paul Abbuehl, Jeffrey Allen, Russ B Altman, Shawn Balcome, Alexis Battle, Andreas Bender, Bonnie Berger, Jonathan Bernard, Madhuchhanda Bhattacharjee, Krithika Bhuvaneshwar, Andrew A Bieberich, Fred Boehm, Andrea Califano, Christina Chan, Beibei Chen, Ting-Huei Chen, Jaejoon Choi, Luis Pedro Coelho, Thomas Cokelaer, James C Collins, Chad J Creighton, Jike Cui, Will Dampier, V Jo Davisson, Bernard De Baets, Raamesh Deshpande, Barbara

DiCamillo, Murat Dundar, Zhana Duren, Adam Ertel, Haoyang Fan, Hongbin Fang, Dan Gallahan, Robinder Gauba, Assaf Gottlieb, Michael Grau, Joe W Gray, Yuriy Gusev, Min Jin Ha, Leng Han, Michael Harris, Nicholas Henderson, Hussein A Hejase, Krisztian Homicsko, Jack P Hou, Woochang Hwang, Adriaan P IJzerman, Bilge Karacali, Samuel Kaski, Sunduz Keles, Christina Kendziorski, Junho Kim, Min Kim, Youngchul Kim, David A Knowles, Daphne Koller, Junehawk Lee, Jae K Lee, Eelke B Lenselink, Biao Li, Bin Li, Jun Li, Han Liang, Jian Ma, Subha Madhavan, Sean Mooney, Chad L Myers, Michael A Newton, John P Overington, Ranadip Pal, Jian Peng, Richard Pestell, Robert J Prill, Peng Qiu, Bartek Rajwa, Anguraj Sadanandam, Julio Saez-Rodriguez, Francesco Sambo, Hyunjin Shin, Dinah Singer, Jiuzhou Song, Lei Song, Arvind Sridhar, Michiel Stock, Gustavo Stolovitzky, Wei Sun, Tram Ta, Mahlet Tadesse, Ming Tan, Hao Tang, Dan Theodorescu, Gianna Maria Toffolo, Aydin Tozeren, William Trepicchio, Nelle Varoquaux, Jean-Philippe Vert, Willem Waegeman, Thomas Walter, Qian Wan, Difei Wang, Wen Wang, Yong Wang, Zhishi Wang, Joerg K Wegner, Tongtong Wu, Tian Xia, Guanghua Xiao, Yang Xie, Yanxun Xu, Jichen Yang, Yuan Yuan, Shihua Zhang, Xiang-Sun Zhang, Junfei Zhao, Chandler Zuo, Herman W T van Vlijmen, Gerard J P van Westen, and James J Collins. A community effort to assess and improve drug sensitivity prediction algorithms. *Nature Biotechnology*, 32:1202–1212, 6 2014. ISSN 1087-0156. doi:10.1038/nbt.2877.

Frank Albert Cotton, Geoffrey Wilkinson, Carlos A. Murillio, and Manfred Bochmann. *Advanced Inorganic Chemistry*. Wiley, Chichester, United Kingdom, 6 edition, 1999.

Patrick Cramer. Alphafold2 and the future of structural biology. *Nature Structural & Molecular Biology*, 28(9):704–705, 2021.

Jason B Cross, David C Thompson, Brajesh K Rai, J Christian Baber, Kristi Yi Fan, Yongbo Hu, and Christine Humblet. Comparison of several molecular docking programs: pose prediction and virtual screening accuracy. *Journal of chemical information and modeling*, 49(6):1455–1474, 2009.

Peter Csermely, Tamás Korcsmáros, Huba JM Kiss, Gabor London, and Ruth Nussinov. Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacology & therapeutics*, 138(3):333–408, 2013.

Song Cui, Qiang Wu, James West, and Jiangping Bai. Machine learning-based microarray analyses indicate low-expression genes might collectively influence PAH disease. *PLOS Computational Biology*, 15:e1007264, 8 2019. ISSN 1553-734X. doi:10.1371/journal.pcbi.1007264.

C Da, M Stashko, C Jayakody, X Wang, W Janzen, S Frye, and D Kireev. Discovery of mer kinase inhibitors by virtual screening using structural protein–ligand interaction fingerprints. *Bioorganic & medicinal chemistry*, 23(5):1096–1101, 2015.

Thomas G Davies and Ian J Tickle. Fragment screening using x-ray crystallography. *Fragment-Based Drug Discovery and X-Ray Crystallography*, pages 33–59, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Maya G. Deshmukh, Joseph A. Ippolito, Chun-Hui Zhang, Elizabeth A. Stone, Raquel A. Reilly, Scott J. Miller, William L. Jorgensen, and Karen S. Anderson. Structure-guided design of a perampanel-derived pharmacophore targeting the sars-cov-2 main protease. *Structure*, 29(8):823–833.e5, 2021. ISSN 0969-2126. doi:https://doi.org/10.1016/j.str.2021.06.002. URL `https://www.sciencedirect.com/science/article/pii/S0969212621002069`.

Dilyana Dimova, Dagmar Stumpfe, and Jürgen Bajorath. Computational design of new molecular scaffolds for medicinal chemistry, part ii: generalization of analog series-based scaffolds. *Future science OA*, 4(2):FSO267, 2017.

Abigail Dommer, Lorenzo Casalino, Fiona Kearns, Mia Rosenfeld, Nicholas Wauer, Surl-Hee Ahn, John Russo, Sofia Oliveira, Clare Morris, Anthony Bogetti, et al. # covidisairborne: Ai-enabled multiscale computational microscopy of delta sars-cov-2 in a respiratory aerosol. *bioRxiv*, 2021.

Alice Douangamath, Daren Fearon, Paul Gehrtz, Tobias Krojer, Petra Lukacik, C David Owen, Efrat Resnick, Claire Strain-Damerell, Anthony Aimon, Péter Ábrányi-Balogh, et al. Crystallographic and electrophilic fragment screening of the sars-cov-2 main protease. *Nature communications*, 11(1):1–11, 2020.

Kevin Duh and Katrin Kirchhoff. Learning to rank with partially-labeled data. page 251, 2008. doi:10.1145/1390334.1390379.

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

Francis Dutil, Joseph Paul Cohen, Martin Weiss, Georgy Derevyanko, and Yoshua Bengio. Towards gene expression convolutions using gene interaction graphs. *arXiv preprint arXiv:1806.06975*, 2018.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

Sara E Nichols, Robert V Swift, and Rommie E Amaro. Rational prediction with molecular dynamics for hit identification. *Current topics in medicinal chemistry*, 12(18):2002–2012, 2012.

P Eastman. Pdbfixer, 2013.

Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, Kyle A. Beauchamp, Lee-Ping Wang, Andrew C. Simmonett, Matthew P. Harrigan, Chaya D. Stern, Rafal P. Wiewiora, Bernard R. Brooks, and Vijay S. Pande. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):1–17, 07 2017a. doi:10.1371/journal.pcbi.1005659. URL `https://doi.org/10.1371/journal.pcbi.1005659`.

Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659, 2017b.

Jerry Osagie Ebalunode and Weifan Zheng. Unconventional 2d shape similarity method affords comparable enrichment as a 3d shape method in virtual screening experiments. *Journal of chemical information and modeling*, 49(6):1313–1320, 2009.

Daniel C Elton, Zois Boukouvalas, Mark D Fuge, and Peter W Chung. Deep learning for molecular design–a review of the state of the art. *Molecular Systems Design & Engineering*, 4(4):828–849, 2019.

Paul Emsley and Kevin Cowtan. Coot: Model-building tools for molecular graphics. *Acta Crystallogr. D Biol. Crystallogr.*, 60(12):2126–2132, 2004.

Andrew C English, Colin R Groom, and Roderick E Hubbard. Experimental and computational mapping of the binding surface of a crystalline protein. *Protein engineering*, 14(1): 47–59, 2001.

Spencer S Ericksen, Haozhen Wu, Huikun Zhang, Lauren A Michael, Michael A Newton, F Michael Hoffmann, and Scott A Wildman. Machine learning consensus scoring improves performance across targets in structure-based virtual screening. *Journal of chemical information and modeling*, 57(7):1579–1590, 2017.

Daniel A Erlanson, Stephen W Fesik, Roderick E Hubbard, Wolfgang Jahnke, and Harren Jhoti. Twenty years on: the impact of fragments on drug discovery. *Nature reviews Drug discovery*, 15(9):605, 2016.

Facts & Factors. With 39% cagr, ai in drug discovery market size surge to usd 485 million by 2026: Facts & factors, Mar 2021.

Ya Ju Fan, Jonathan E Allen, Sam Ade Jacobs, and Brian C Van Essen. Distinguishing between Normal and Cancer Cells Using Autoencoder Node Saliency. 2019.

Elizabeth Farrant. Automation of synthesis in medicinal chemistry: Progress and challenges. *ACS Medicinal Chemistry Letters*, 11(8):1506–1513, 2020.

Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. Potentialnet for molecular property prediction. *ACS central science*, 4(11):1520–1530, 2018.

Chunlai Feng, Hengwei Chen, Xianqin Yuan, Mengqiu Sun, Kexin Chu, Hanqin Liu, and Mengjie Rui. Gene Expression Data Based Deep Learning Model for Accurate Prediction of Drug-Induced Liver Injury in Advance. *Journal of Chemical Information and Modeling*, 59:3240–3250, 6 2019. ISSN 1549-9596. doi:10.1021/acs.jcim.9b00143.

Noelia Ferruz, Stefan Doerr, Michelle A Vanase-Frawley, Yaozhong Zou, Xiaomin Chen, Eric S Marr, Robin T Nelson, Bethany L Kormos, Travis T Wager, Xinjun Hou, et al. Dopamine d3 receptor antagonist reveals a cryptic pocket in aminergic gpcrs. *Scientific reports*, 8(1):1–10, 2018.

Tobias Fink, Heinz Bruggesser, and Jean-Louis Reymond. Virtual exploration of the small-molecule chemical universe below 160 daltons. *Angewandte Chemie International Edition*, 44(10):1504–1508, 2005.

Steve Fisher and Antony Wilson. SAGA Extension: Information Service Navigator API. OGF Proposed Recommendation, GFD.195, Open Grid Forum, 2012. URL `http://ogf.org/documents/GFD.195.pdf`.

Stefano Forli. Charting a path to success in virtual screening. *Molecules*, 20(10):18732–18758, 2015.

Marcela Franco, Ashwini Jeggari, Sylvain Peuget, Franziska Böttger, Galina Selivanova, and Andrey Alexeyenko. Prediction of response to anti-cancer drugs becomes robust via network integration of molecular data. *Scientific Reports*, 9:2379, 2 2019. doi:10.1038/s41598-019-39019-2.

E. Friedman-Hill. *Writing Rules in Jess*. Manning Publications Co., Greenwich, CT, USA, 1 edition, 2003.

M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, Montgomery, Jr., J. A., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople. *Gaussian 03*. Gaussian, Inc., Wallingford, CT, 2004.

Feng Gao, Wei Wang, Miaomiao Tan, Lina Zhu, Yuchen Zhang, Evelyn Fessler, Louis Vermeulen, and Xin Wang. DeepCC: a novel deep learning-based framework for cancer molecular subtype classification. *Oncogenesis*, 8:44, 8 2019. ISSN 2157-9024. doi:10.1038/s41389-019-0157-8.

Wenhao Gao and Connor W Coley. The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.

Michael R Garey and David S Johnson. Computers and intractability. *A Guide to the*, 1979.

Mathew J Garnett, Elena J Edelman, Sonja J Heidorn, Chris D Greenman, Anahita Dastur, King Wai Lau, Patricia Greninger, I Richard Thompson, Xi Luo, Jorge Soares, Qingsong Liu, Francesco Iorio, Didier Surdez, Li Chen, Randy J Milano, Graham R Bignell, Ah T Tam, Helen Davies, Jesse A Stevenson, Syd Barthorpe, Stephen R Lutz, Fiona Kogera, Karl Lawrence, Anne McLaren-Douglas, Xeni Mitropoulos, Tatiana Mironenko, Helen Thi, Laura Richardson, Wenjun Zhou, Frances Jewitt, Tinghu Zhang, Patrick O'Brien, Jessica L Boisvert, Stacey Price, Wooyoung Hur, Wanjuan Yang, Xianming Deng, Adam Butler, Hwan Geun Choi, Jae Won Chang, Jose Baselga, Ivan Stamenkovic, Jeffrey A Engelman, Sreenath V Sharma, Olivier Delattre, Julio Saez-Rodriguez, Nathanael S Gray, Jeffrey Settleman, P Andrew Futreal, Daniel A Haber, Michael R Stratton, Sridhar Ramaswamy, Ultan McDermott, and Cyril H Benes. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483:570–575, 3 2012. ISSN 0028-0836. doi:10.1038/nature11005.

Francesco Gentile, Vibudh Agrawal, Michael Hsing, Anh-Tien Ton, Fuqiang Ban, Ulf Norinder, Martin E Gleave, and Artem Cherkasov. Deep docking: a deep learning platform for augmentation of structure based drug discovery. *ACS central science*, 6(6):939–949, 2020.

Sutapa Ghosh, Aihua Nie, Jing An, and Ziwei Huang. Structure-based virtual screening of chemical libraries for drug discovery. *Current opinion in chemical biology*, 10(3):194–202, 2006.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural Message Passing for Quantum Chemistry. 2017a.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017b.

Coryandar M Gilvary, Jonathan R Dry, and Olivier Elemento. Multi-task learning predicts drug combination synergy in cells and in the clinic. *bioRxiv*, page 576017, 2019. doi:10.1101/576017.

Marina Gjorgjieva, Tihomir Tomasic, Michaela Barancokova, Sotirios Katsamakas, Janez Ilas, Päivi Tammela, Lucija Peterlin Masic, and Danijel Kikelj. Discovery of benzothiazole scaffold-based dna gyrase b inhibitors. *Journal of medicinal chemistry*, 59(19):8941–8954, 2016.

Jens Glaser, Josh V. Vermaas, David M. Rogers, et al. High-throughput virtual laboratory for drug discovery using massive datasets. *International Journal of High-Performance Computing Applications (to appear)*, 2020.

Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan O Hodas, and Nathan Baker. Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. 2017.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Adam Gonczarek, Jakub M Tomczak, Szymon Zaręba, Joanna Kaczmar, Piotr Dąbrowski, and Michał J Walczak. Learning deep architectures for interaction prediction in structure-based virtual screening. *arXiv preprint arXiv:1610.07187*, 2016.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Christoph Gorgulla, 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, et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature*, 580(7805):663–668, 2020.

Christoph Gorgulla, Krishna M Padmanabha Das, Kendra E Leigh, Marco Cespugli, Patrick D Fischer, Zi-Fu Wang, Guilhem Tesseyre, Shreya Pandita, Alec Shnapir, Anthony Calderaio, et al. A multi-pronged approach targeting sars-cov-2 proteins using ultra-large virtual screening. *Iscience*, 24(2):102021, 2021.

Andreas W. Götz, Mark J. Williamson, Dong Xu, Duncan Poole, Scott Le Grand, and Ross C. Walker. Routine microsecond molecular dynamics simulations with amber on gpus. 1. generalized born. *Journal of Chemical Theory and Computation*, 8(5):1542–1555, 05 2012. doi:10.1021/ct200909j. URL `https://doi.org/10.1021/ct200909j`.

M R Grever, S A Schepartz, and B A Chabner. The National Cancer Institute: cancer drug discovery and development program. *Seminars in oncology*, 19:622–38, 1992. ISSN 0093-7754.

Francesca Grisoni, Michael Moret, Robin Lingwood, and Gisbert Schneider. Bidirectional molecule generation with recurrent neural networks. *Journal of chemical information and modeling*, 60(3):1175–1183, 2020.

Piotr S Gromski, Jarosław M Granda, and Leroy Cronin. Universal chemical synthesis and discovery with 'the chemputer'. *Trends in Chemistry*, 2(1):4–12, 2020.

Robert L Grossman, Allison P Heath, Vincent Ferretti, Harold E Varmus, Douglas R Lowy, Warren A Kibbe, and Louis M Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112, 2016.

Helmut Grubmüller, Helmut Heller, Andreas Windemuth, and Klaus Schulten. Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6(1-3):121–142, 1991.

Isabella A Guedes, Felipe SS Pereira, and Laurent E Dardenne. Empirical scoring functions for structure-based virtual screening: applications, critical aspects, and challenges. *Frontiers in Pharmacology*, 9:1089, 2018. Publisher: Frontiers.

Rajarshi Guha. *The Ups and Downs of Structure–Activity Landscapes*, pages 101–117. Humana Press, Totowa, NJ, 2011. ISBN 978-1-60761-839-3. doi:10.1007/978-1-60761-839-3_3. URL `https://doi.org/10.1007/978-1-60761-839-3`$_3$.

Anvita Gupta, Alex T Müller, Berend JH Huisman, Jens A Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for de novo drug design. *Molecular informatics*, 37(1-2):1700111, 2018.

Neha Gupta. Introduction to hardware accelerator systems for artificial intelligence and machine learning. In *Advances in Computers*, volume 122, pages 1–21. Elsevier, 2021.

Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.

Lowell H Hall and Lemont B Kier. Issues in representation of molecular structure: the development of molecular connectivity. *Journal of Molecular Graphics and Modelling*, 20 (1):4–18, 2001.

Mathew D Halls, Peter J Djurovich, David J Giesen, Alexander Goldberg, Jonathan Sommer, Eric McAnally, and Mark E Thompson. Virtual screening of electron acceptor materials for organic photovoltaic applications. *New Journal of Physics*, 15(10):105029, 2013.

Masatoshi Hamanaka, Kei Taneishi, Hiroaki Iwata, Jun Ye, Jianguo Pei, Jinlong Hou, and Yasushi Okuno. Cgbvs-dnn: Prediction of compound-protein interactions based on deep learning. *Molecular informatics*, 36(1-2):1600045, 2017.

Yue Han, Chengyu Wang, Qi Dong, Tingting Chen, Fan Yang, Yaoyao Liu, Bo Chen, Zhangxiang Zhao, Lishuang Qi, Wenyuan Zhao, Haihai Liang, Zheng Guo, and Yunyan Gu. Genetic Interaction-Based Biomarkers Identification for Drug Resistance and Sensitivity in Cancer Cells. *Molecular therapy. Nucleic acids*, 17:688–700, 7 2019. ISSN 2162-2531. doi:10.1016/j.omtn.2019.07.003.

Corwin Hansch. Structure of medicinal chemistry. *Journal of Medicinal Chemistry*, 19(1): 1–6, 1976.

Gavin Harper, John Bradshaw, John C Gittins, Darren VS Green, and Andrew R Leach. Prediction of biological activity for high-throughput screening using binary kernel discrimination. *Journal of Chemical Information and Computer Sciences*, 41(5):1295–1300, 2001.

David I Harvey, Stephen J Leybourne, and Paul Newbold. Tests for Forecast Encompassing. *Journal of Business & Economic Statistics*, 16:254, 1998. ISSN 0735-0015. doi:10.2307/1392581.

Nafisa M Hassan, Amr A Alhossary, Yuguang Mu, and Chee-Keong Kwoh. Protein-ligand blind docking using quickvina-w with inter-process spatio-temporal integration. *Scientific reports*, 7(1):1–13, 2017.

Alexander S Hauser, Sreenivas Chavali, Ikuo Masuho, Leonie J Jahn, Kirill A Martemyanov, David E Gloriam, and M Madan Babu. Pharmacogenomics of gpcr drug targets. *Cell*, 172(1-2):41–54, 2018.

Paul C. D. Hawkins, A. Geoffrey Skillman, Gregory L. Warren, Benjamin A. Ellingson, and Matthew T. Stahl. Conformer generation with omega: Algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of Chemical Information and Modeling*, 50(4):572–584, 04 2010. doi:10.1021/ci100031x. URL `https://doi.org/10.1021/ci100031x`.

Paul CD Hawkins, A Geoffrey Skillman, and Anthony Nicholls. Comparison of shape-matching and docking as virtual screening tools. *Journal of medicinal chemistry*, 50(1): 74–82, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Annette Hegyi and John Ziebuhr. Conservation of substrate specificities among coronavirus main proteases. *Journal of General Virology*, 83(3):595–599, 2002. ISSN 0022-1317. doi:https://doi.org/10.1099/0022-1317-83-3-595. URL `https://www.microbiologyresearch.org/content/journal/jgv/10.1099/0022-1317-83-3-595`.

Moritz Hess, Stefan Lenz, Tamara J Blätte, Lars Bullinger, and Harald Binder. Partitioned learning of deep Boltzmann machines for SNP data. *Bioinformatics*, 33:3173–3180, 2017. ISSN 1367-4803. doi:10.1093/bioinformatics/btx408.

Alexander Hillisch, Luis Felipe Pineda, and Rolf Hilgenfeld. Utility of homology models in the drug discovery process. *Drug discovery today*, 9(15):659–669, 2004.

Maya Hirohara, Yutaka Saito, Yuki Koda, Kengo Sato, and Yasubumi Sakakibara. Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC Bioinformatics*, 19:526, 2018. doi:10.1186/s12859-018-2523-5.

Daniel Sik Wai Ho, William Schierding, Melissa Wake, Richard Saffery, and Justin O'Sullivan. Machine Learning SNP Based Prediction for Precision Medicine. *Frontiers in Genetics*, 10:267, 1 2019. ISSN 1664-8021. doi:10.3389/fgene.2019.00267.

Robert L Hoffman, Robert S Kania, Mary A Brothers, Jay F Davies, Rose A Ferre, Ketan S Gajiwala, Mingying He, Robert J Hogan, Kirk Kozminski, Lilian Y Li, Jonathan W. Lockner, Jihong Lou, Michelle T. Marra, Lennert J. Mitchell, Jr., Brion W. Murray, James A. Nieman, Stephen Noell, Simon P. Planken, Thomas Rowe, Kevin Ryan, George J. Smith, III, James E. Solowiej, Claire M. Steppan, and Barbara Taggart. Discovery of ketone-based covalent inhibitors of coronavirus 3CL proteases for the potential therapeutic treatment of COVID-19. *J. Med. Chem.*, 63(21):12725–12747, 2020.

Susan L Holbeck, Richard Camalier, James A Crowell, Jeevan Prasaad Govindharajulu, Melinda Hollingshead, Lawrence W Anderson, Eric Polley, Larry Rubinstein, Apurva Srivastava, Deborah Wilsker, et al. The national cancer institute almanac: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer research*, 77(13):3564–3576, 2017.

Qiwen Hu and Casey S Greene. Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics. *bioRxiv*, page 385534, 2018. doi:10.1101/385534.

Qiyue Hu, Zhengwei Peng, Jaroslav Kostrowicki, and Atsuo Kuki. Leap into the pfizer global virtual library (pgvl) space: creation of readily synthesizable design ideas automatically. In *Chemical Library Design*, pages 253–276. Springer, 2011.

Chen-Tsung Huang, Chiao-Hui Hsieh, Yun-Hsien Chung, Yen-Jen Oyang, Hsuan-Cheng Huang, and Hsueh-Fen Juan. Perturbational Gene-Expression Signatures for Combinatorial Drug Discovery. *iScience*, 15:291–306, 5 2019. ISSN 2589-0042. doi:10.1016/j.isci.2019.04.039.

Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R Benson. Combining label propagation and simple models out-performs graph neural networks. *arXiv preprint arXiv:2010.13993*, 2020.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35:73–101, 1964. ISSN 0003-4851. doi:10.1214/aoms/1177703732.

James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

Francesco Iorio, Theo A. Knijnenburg, Daniel J. Vis, Graham R. Bignell, Michael P. Menden, Michael Schubert, Nanne Aben, Emanuel Gonçalves, Syd Barthorpe, Howard Lightfoot, Thomas Cokelaer, Patricia Greninger, Ewald van Dyk, Han Chang, Heshani de Silva, Holger Heyn, Xianming Deng, Regina K. Egan, Qingsong Liu, Tatiana Mironenko, Xeni Mitropoulos, Laura Richardson, Jinhua Wang, Tinghu Zhang, Sebastian Moran, Sergi Sayols, Maryam Soleimani, David Tamborero, Nuria Lopez-Bigas, Petra Ross-Macdonald, Manel Esteller, Nathanael S. Gray, Daniel A. Haber, Michael R. Stratton, Cyril H. Benes, Lodewyk F.A. Wessels, Julio Saez-Rodriguez, Ultan McDermott, and Mathew J. Garnett. A Landscape of Pharmacogenomic Interactions in Cancer. *Cell*, 166:740–754, 2016. ISSN 0092-8674. doi:10.1016/j.cell.2016.06.017.

Nicolae C Iovanac and Brett M Savoie. Improved chemical prediction from scarce data sets via latent space enrichment. *The Journal of Physical Chemistry A*, 123(19):4295–4302, 2019.

John J Irwin and Brian K Shoichet. ZINC - A free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1): 177–182, 2005a.

John J. Irwin and Brian K. Shoichet. Zinc -a free database of commercially available compounds for virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182, 01 2005b. doi:10.1021/ci049714+. URL `https://doi.org/10.1021/ci049714+`.

John J Irwin, Brian K Shoichet, Michael M Mysinger, Niu Huang, Francesco Colizzi, Pascal Wassam, and Yiqun Cao. Automated docking screens: a feasibility study. *Journal of medicinal chemistry*, 52(18):5712–5720, 2009. Publisher: ACS Publications.

In Sock Jang, Elias Chaibub Neto, Justin Guinney, Stephen H Friend, and Adam A Margolin. Systematic assessment of analytical methods for drug sensitivity prediction from cancer cell line data. In *Biocomputing 2014*, pages 63–74. World Scientific, 2014.

Sheila Jasanoff. The idiom of co-production. In *States of knowledge*, pages 1–12. Routledge, 2004.

Harren Jhoti and Andrew R Leach. *Structure-based drug discovery*, volume 1. Springer, 2007.

Junteng Jia and Austion R Benson. Residual correlation in graph neural network regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 588–598, 2020.

Xiwen Jia, Allyson Lynch, Yuheng Huang, Matthew Danielson, Immaculate Lang'at, Alexander Milder, Aaron E Ruby, Hao Wang, Sorelle A Friedler, Alexander J Norquist, et al. Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis. *Nature*, 573(7773):251–255, 2019.

Jianchang Mao, K. Mohiuddin, and A. K. Jain. Parsimonious network design and feature selection through node pruning. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*, volume 2, pages 622–624 vol.2, Oct 1994. doi:10.1109/ICPR.1994.577060.

José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

Jose Jimenez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. Kdeep: Protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of Chemical Information and Modeling*, 58(2):287–296, 2018.

doi:10.1021/acs.jcim.7b00650. URL https://doi.org/10.1021/acs.jcim.7b00650. PMID: 29309725.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. 2018a.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018b. PMLR. URL http://proceedings.mlr.press/v80/jin18a.html.

Zhenming Jin, Xiaoyu Du, Yechun Xu, Yongqiang Deng, Meiqin Liu, Yao Zhao, Bing Zhang, Xiaofeng Li, Leike Zhang, Chao Peng, et al. Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. *Nature*, 582:289–293, 2020.

Clemens Jochum and Johann Gasteiger. Canonical numbering and constitutional symmetry. *Journal of Chemical Information and Computer Sciences*, 17(2):113–117, 1977.

Alexander Lawrence Johnson. 1-(alkylsubstituted phenyl)imidazoles useful in acth reverse assay, 1972.

J I Johnson, S Decker, D Zaharevitz, L V Rubinstein, J M Venditti, S Schepartz, S Kalyandrug, M Christian, S Arbuck, M Hollingshead, and E A Sausville. Relationships between drug activity in NCI preclinical in vitro and in vivo models and early clinical trials. *British Journal of Cancer*, 84(10):1424–1431, May 2001. ISSN 15321827. doi:10.1054/bjoc.2001.1796. URL http://www.nature.com/doifinder/10.1054/bjoc.2001.1796.

Mark A Johnson and Gerald M Maggiora. *Concepts and applications of molecular similarity*. Wiley, 1990.

W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8:118–127, 2007. ISSN 1465-4644. doi:10.1093/biostatistics/kxj037.

Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, WF Drew Bennett, Daniel Kirshner, Sergio E Wong, Felice C Lightstone, and Jonathan E Allen. Improved protein–ligand binding affinity prediction with structure-based deep fusion inference. *Journal of chemical information and modeling*, 61(4):1583–1592, 2021.

William L Jorgensen. Rusting of the lock and key model for protein-ligand binding. *Science*, 254(5034):954–956, 1991.

William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of chemical physics*, 79(2):926–935, 1983.

Visvaldas Kairys, Miguel X Fernandes, and Michael K Gilson. Screening drug-like compounds by docking to homology models: a systematic study. *Journal of chemical information and modeling*, 46(1):365–379, 2006.

Amit S Kalgutkar and Deepak K Dalvie. Drug discovery for a new generation of covalent drugs. *Expert opinion on drug discovery*, 7(7):561–581, 2012.

Anat Levit Kaplan, Danielle N Confair, Kuglae Kim, Ximena Barros-Álvarez, Ramona M Rodriguiz, Ying Yang, Oh Sang Kweon, Tao Che, John D McCorvy, David N Kamber, et al. Bespoke library docking for 5-ht2a receptor agonists with antidepressant activity. *Nature*, pages 1–10, 2022.

Taha A Kass-Hout, Zhiheng Xu, Matthew Mohebbi, Hans Nelsen, Adam Baker, Jonathan Levine, Elaine Johanson, and Roselie A Bright. Openfda: an innovative platform providing access to a wealth of fda's publicly available data. *Journal of the American Medical Informatics Association*, 23(3):596–600, 2016.

Daniel B Kassel. Applications of high-throughput adme in drug discovery. *Current opinion in chemical biology*, 8(3):339–345, 2004.

Sameer Kawatkar, Hongming Wang, Ryszard Czerminski, and Diane Joseph-McCarthy. Virtual fragment screening: an exploration of various docking and scoring protocols for fragments using glide. *Journal of computer-aided molecular design*, 23(8):527–539, 2009.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

Alexandra B Keenan, Sherry L Jenkins, Kathleen M Jagodnik, Simon Koplev, Edward He, Denis Torre, Zichen Wang, Anders B Dohlman, Moshe C Silverstein, Alexander Lachmann, et al. The library of integrated network-based cellular signatures nih program: system-level cataloging of human cells response to perturbations. *Cell systems*, 6(1):13–24, 2018.

Esther Kellenberger, Nicolas Foata, and Didier Rognan. Ranking targets in structure-based virtual screening of three-dimensional protein libraries: methods and problems. *Journal of chemical information and modeling*, 48(5):1014–1025, 2008.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019a.

Youngchul Kim, Daewon Kim, Biwei Cao, Rodrigo Carvajal, and Minjung Kim. PDXGEM: Patient-Derived Tumor Xenograft based Gene Expression Model for Predicting Clinical Response to Anticancer Therapy in Cancer Patients. *bioRxiv*, page 686667, 6 2019b. doi:10.1101/686667.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014.

Diederik P Kingma, Danilo J Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *arXiv preprint arXiv:1406.5298*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Peter Kirkpatrick and Clare Ellis. Chemical space. *Nature*, 432(7019):823–824, 2004.

Robert Kiss, Mark Sandor, and Ferenc A. Szalai. http://mcule.com: a public web service for drug discovery. *Journal of Cheminformatics*, 4(1):P17, 2012. doi:10.1186/1758-2946-4-S1-P17. URL `https://doi.org/10.1186/1758-2946-4-S1-P17`.

DB Kitchen, H Decornez, JR Furr, and J Bajorath. Docking and scoring in virtual screening for drug discovery: Methods and applications. *NATURE REVIEWS DRUG DISCOVERY*, 3(11):935–949, November 2004. ISSN 1474-1776. doi:10.1038/nrd1549.

Gerhard Klebe. Virtual ligand screening: strategies, perspectives and limitations. *DRUG DISCOVERY TODAY*, 11(13-14):580–594, July 2006. ISSN 1359-6446. doi:10.1016/j.drudis.2006.05.012.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P17-4012`.

Christiaan Klijn, Steffen Durinck, Eric W Stawiski, Peter M Haverty, Zhaoshi Jiang, Hanbin Liu, Jeremiah Degenhardt, Oleg Mayba, Florian Gnad, Jinfeng Liu, Gregoire Pau, Jens Reeder, Yi Cao, Kiran Mukhyala, Suresh K Selvaraj, Mamie Yu, Gregory J Zynda, Matthew J Brauer, Thomas D Wu, Robert C Gentleman, Gerard Manning, Robert L Yauch, Richard Bourgon, David Stokoe, Zora Modrusan, Richard M Neve, Frederic J de Sauvage, Jeffrey Settleman, Somasekar Seshagiri, and Zemin Zhang. A comprehensive transcriptional portrait of human cancer cell lines. *Nature Biotechnology*, 33:306–312, 2014. ISSN 1087-0156. doi:10.1038/nbt.3080.

Gilles Klopman. Artificial intelligence approach to structure-activity studies. computer automated structure evaluation of biological activity of organic molecules. *Journal of the American Chemical Society*, 106(24):7315–7321, 1984.

Daniel W Kneller, Stephanie Galanie, Gwyndalyn Phillips, Hugh M O'Neill, Leighton Coates, and Andrey Kovalevsky. Malleability of the SARS-CoV-2 3CL Mpro active-site cavity facilitates binding of clinical antivirals. *Structure*, 28:1313–1320.e3, 2020a.

Daniel W Kneller, Gwyndalyn Phillips, Andrey Kovalevsky, and Leighton Coates. Room-temperature neutron and X-ray data collection of 3CL Mpro from SARS-CoV-2. *Acta Crystallogr. F Struct. Biol. Commun.*, 76(10):483–487, 2020b.

Daniel W Kneller, Gwyndalyn Phillips, Hugh M O'Neill, Robert Jedrzejczak, Lucy Stols, Paul Langan, Andrzej Joachimiak, Leighton Coates, and Andrey Kovalevsky. Structural plasticity of the SARS-CoV-2 3CL Mpro active site cavity revealed by room temperature X-ray crystallography. *Nature Communications*, 11:3202, 2020c.

Daniel W Kneller, Gwyndalyn Phillips, Hugh M O'Neill, Kemin Tan, Andrzej Joachimiak, Leighton Coates, and Andrey Kovalevsky. Room-temperature X-ray crystallography reveals the oxidation and reactivity of cysteine residues in SARS-CoV-2 3CL Mpro: Insights into enzyme mechanism and drug design. *IUCrJ*, 7(6):1028–1035, 2020d.

Daniel W Kneller, Gwyndalyn Phillips, Kevin L Weiss, Swati Pant, Qiu Zhang, Hugh M O'Neill, Leighton Coates, and Andrey Kovalevsky. Unusual zwitterionic catalytic site of SARS-CoV-2 main protease revealed by neutron crystallography. *J. Biol. Chem.*, 295(50): 17365–17373, 2020e.

Daniel W Kneller, Hui Li, Stephanie Galanie, Gwyndalyn Phillips, Audrey Labbé, Kevin L Weiss, Qiu Zhang, Mark A Arnould, Austin Clyde, Heng Ma, et al. Structural, electronic, and electrostatic determinants for inhibitor binding to subsites s1 and s2 in sars-cov-2 main protease. *Journal of medicinal chemistry*, 64(23):17366–17383, 2021.

Maria Deloria Knoll and Chizoba Wonodi. Oxford–AstraZeneca COVID-19 vaccine efficacy. *The Lancet*, 397(10269):72–74, 2021. ISSN 0140-6736. doi:https://doi.org/10.1016/S0140-6736(20)32623-4. URL `http://www.sciencedirect.com/science/article/pii/S0140673620326234`.

David A. Knowles, Gina Bouchard, and Sylvia Plevritis. Sparse discriminative latent characteristics for predicting cancer drug sensitivity from genomic features. *PLOS Computational Biology*, 15:e1006743, 5 2019. ISSN 1553-734X. doi:10.1371/journal.pcbi.1006743.

Amar Koleti, Raymond Terryn, Vasileios Stathias, Caty Chung, Daniel J Cooper, John P Turner, Dušica Vidović, Michele Forlin, Tanya T Kelley, Alessandro D'Urso, et al. Data portal for the library of integrated network-based cellular signatures (lincs) program: integrated access to diverse large-scale cellular perturbation response data. *Nucleic acids research*, 46(D1):D558–D566, 2017.

Peter A Kollman, Irina Massova, Carolina Reyes, Bernd Kuhn, Shuanghong Huo, Lillian Chong, Matthew Lee, Taisung Lee, Yong Duan, Wei Wang, et al. Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models. *Accounts of chemical research*, 33(12):889–897, 2000.

Maria Kontoyianni, Glenn S Sokol, and Laura M MCclellan. Evaluation of library ranking efficacy in virtual screening. *Journal of computational chemistry*, 26(1):11–22, 2005.

Alexandru Korotcov, Valery Tkachenko, Daniel P Russo, and Sean Ekins. Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. *Molecular pharmaceutics*, 14(12):4462–4475, 2017.

George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014a.

George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014b.

Pawel M Kozlowski. Quantum chemical modeling of bond activation in b12-dependent enzymes. *Current Opinion in Chemical Biology*, 5(6):736–743, 2001. ISSN 1367-5931. doi:https://doi.org/10.1016/S1367-5931(01)00273-3. URL `https://www.sciencedirect.com/science/article/pii/S1367593101002733`.

Mario Krenn, Florian Häse, A Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Selfies: a robust representation of semantically constrained graphs with an example application in chemistry. *arXiv preprint arXiv:1905.13741*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Dennis M Krüger and Andreas Evers. Comparison of structure-and ligand-based virtual screening protocols considering hit list complementarity and enrichment factors. *ChemMedChem: Chemistry Enabling Drug Discovery*, 5(1):148–158, 2010.

Chih-Jung Kuo, Ya-Hui Chi, John T-A Hsu, and Po-Huang Liang. Characterization of SARS main protease and inhibitor assay using a fluorogenic substrate. *Biochem. Biophys. Res. Commun.*, 318(4):862–867, 2004.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

Sunyoung Kwon and Sungroh Yoon. DeepCCI: End-to-end Deep Learning for Chemical-Chemical Interaction Prediction. pages 203–212, 2017. doi:10.1145/3107411.3107451.

Junyong Lai, Jianxing Hu, Yanxing Wang, Xin Zhou, Yibo Li, Liangren Zhang, and Zhenming Liu. Privileged scaffold analysis of natural products with deep learning-based indication prediction model. *Molecular informatics*, 39(11):2000057, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Eric-Wubbo Lameijer, Thomas Bäck, Joost N Kok, and AD P Ijzerman. Evolutionary algorithms in drug design. *Natural Computing*, 4(3):177–243, 2005.

Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

Greg Landrum et al. Rdkit: Open-source cheminformatics software. *GitHub and SourceForge*, 10:3592822, 2016.

Oliver Laufkötter, Stefan Laufer, and Jürgen Bajorath. Kinase inhibitor data set for systematic analysis of representative kinases across the human kinome. *Data in Brief*, 32: 106189, 2020.

Maureen Lyndel C Lauron and Jaderick P Pabico. Improved Sampling Techniques for Learning an Imbalanced Data Set. 2016.

Tung Thanh Le, Jakob P Cramer, Robert Chen, and Stephen Mayhew. Evolution of the COVID-19 vaccine development landscape. *Nature Reviews Drug Discovery*, 19(10):667–8, 2020.

Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC bioinformatics*, 10(1):1–11, 2009.

Joshua Lederberg, Georgia L Sutherland, Bruce G Buchanan, Edward A Feigenbaum, Alexander V Robertson, Alan M Duffield, and Carl Djerassi. Applications of artificial intelligence for chemical inference. i. number of possible organic compounds. acyclic structures containing carbon, hydrogen, oxygen, and nitrogen. *Journal of the American Chemical Society*, 91(11):2973–2976, 1969.

Hyungro Lee, Andre Merzky, Li Tan, Mikhail Titov, Matteo Turilli, Dario Alfe, Agastya Bhati, Alex Brace, Austin Clyde, Peter Coveney, et al. Scalable hpc and ai infrastructure for covid-19 therapeutics. *arXiv preprint arXiv:2010.10517*, 2020.

Hyungro Lee, Andre Merzky, Li Tan, Mikhail Titov, Matteo Turilli, Dario Alfe, Agastya Bhati, Alex Brace, Austin Clyde, Peter Coveney, et al. Scalable hpc & ai infrastructure for covid-19 therapeutics. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–13, 2021.

Scott LeGrand, Aaron Scheinberg, Andreas F Tillack, Mathialakan Thavappiragasam, Josh V Vermaas, Rupesh Agarwal, Jeff Larkin, Duncan Poole, Diogo Santos-Martins, Leonardo Solis-Vasquez, et al. Gpu-accelerated drug discovery with docking on the summit supercomputer: Porting, optimization, and application to covid-19 research. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–10, 2020.

Min Li, Yake Wang, Ruiqing Zheng, Xinghua Shi, yaohang li, Fangxiang Wu, and Jianxin Wang. DeepDSC: A Deep Learning Method to Predict Drug Sensitivity of Cancer Cell Lines. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP:1–1, 5 2019a. ISSN 1545-5963. doi:10.1109/tcbb.2019.2919581.

Xuanyi Li, Yinqiu Xu, Hequan Yao, and Kejiang Lin. Chemical space exploration based on recurrent neural networks: applications in discovering kinase inhibitors. *Journal of Cheminformatics*, 12(1):1–13, 2020.

Yibo Li, Jianxing Hu, Yanxing Wang, Jielong Zhou, Liangren Zhang, and Zhenming Liu. Deepscaffold: A comprehensive tool for scaffold-based de novo drug discovery using deep learning. *Journal of chemical information and modeling*, 60(1):77–91, 2019b.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. 2015.

Jie Liang, Clare Woodward, and Herbert Edelsbrunner. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. *Protein science*, 7(9):1884–1897, 1998.

Zhirui Liao, Ronghui You, Xiaodi Huang, Xiaojun Yao, Tao Huang, and Shanfeng Zhu. Deepdock: enhancing ligand-protein interaction prediction by a combination of ligand and structure information. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 311–317. IEEE, 2019.

Jaechang Lim, Sang-Yeon Hwang, Seokhyun Moon, Seungsu Kim, and Woo Youn Kim. Scaffold-based molecular design with a graph generative model. *Chemical Science*, 11(4): 1153–1164, 2020.

Jung-Hsin Lin, Alexander L Perryman, Julie R Schames, and J Andrew McCammon. Computational drug design accommodating receptor flexibility: the relaxed complex scheme. *Journal of the American Chemical Society*, 124(20):5632–5633, 2002.

Jung-Hsin Lin, Alexander L Perryman, Julie R Schames, and J Andrew McCammon. The relaxed complex method: Accommodating receptor flexibility for drug design with an improved scoring scheme. *Biopolymers: Original Research on Biomolecules*, 68(1):47–62, 2003.

Alex P. Lind and Peter C. Anderson. Predicting drug activity against cancer cells by random forest models based on minimal genomic information and chemical properties. *PLOS ONE*, 14:e0219774, 1 2019. doi:10.1371/journal.pone.0219774.

Jonathan S Lindsey. A retrospective on the automation of laboratory synthetic chemistry. *Chemometrics and intelligent laboratory systems*, 17(1):15–45, 1992.

Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25, 1997.

Yuanhang Liu, Pritha Chanana, Jaime I. Davila, Xiaonan Hou, Valentina Zanfagnin, Cordelia D. McGehee, Ellen L. Goode, Eric C. Polley, Paul Haluska, S. John Weroha, and Chen Wang. Gene expression differences between matched pairs of ovarian cancer patient tumors and patient-derived xenografts. *Scientific Reports*, 9:6314, 4 2019. doi:10.1038/s41598-019-42680-2.

Siegfried N. Lodwig and Clifford J. Unkefer. Stereoselective synthesis of stable isotope-labeled L-α-amino acids: Electrophilic amination of Oppolzer's acyl sultams in the synthesis of L-[^{15}N]alanine, L-[^{15}N]valine, L-[^{15}N]leucine, L-[^{15}N]phenylalanine and L-[1-^{13}C, ^{15}N]valine. *J. Labelled Comp. Radiopharm.*, 38(3):239–248, 1996.

G Long. The biopharmaceutical pipeline: innovative therapies in clinical development. *Boston, MA: Analysis Group*, 2017.

Fabian López-Vallejo, Thomas Caulfield, Karina Martínez-Mayorga, Marc A Giulianotti, Adel Nefzi, Richard A Houghten, and Jose L Medina-Franco. Integrating virtual screening and combinatorial chemistry for accelerated drug discovery. *Combinatorial Chemistry & High Throughput Screening*, 14(6):475–487, 2011.

Shao-Yong Lu, Yong-Jun Jiang, Jing Lv, Tian-Xing Wu, Qing-Sen Yu, and Wei-Liang Zhu. Molecular docking and molecular dynamics simulation studies of gpr40 receptor–agonist interactions. *Journal of Molecular Graphics and Modelling*, 28(8):766–774, 2010.

Joseph H. Lubin, Christine Zardecki, Elliott M. Dolan, Changpeng Lu, Zhuofan Shen, Shuchismita Dutta, John D. Westbrook, Brian P. Hudson, David S. Goodsell, Jonathan K. Williams, Maria Voigt, Vidur Sarma, Lingjun Xie, Thejasvi Venkatachalam, Steven Arnold, Luz Helena Alfaro Alvarado, Kevin Catalfano, Aaliyah Khan, Erika McCarthy, Sophia Staggers, Brea Tinsley, Alan Trudeau, Jitendra Singh, Lindsey Whitmore, Helen Zheng, Matthew Benedek, Jenna Currier, Mark Dresel, Ashish Duvvuru, Britney Dyszel, Emily Fingar, Elizabeth M. Hennen, Michael Kirsch, Ali A. Khan, Charlotte Labrie-Cleary, Stephanie Laporte, Evan Lenkeit, Kailey Martin, Marilyn Orellana, Melanie Ortiz-Alvarez de la Campa, Isaac Paredes, Baleigh Wheeler, Allison Rupert, Andrew Sam, Katherine See, Santiago Soto Zapata, Paul A. Craig, Bonnie L. Hall, Jennifer Jiang, Julia R. Koeppe, Stephen A. Mills, Michael J. Pikaart, Rebecca Roberts, Yana Bromberg, J. Steen Hoyer, Siobain Duffy, Jay Tischfield, Francesc X. Ruiz, Eddy Arnold, Jean Baum, Jesse Sandberg, Grace Brannigan, Sagar D. Khare, and Stephen K. Burley. Evolution of the SARS-CoV-2 proteome in three dimensions (3D) during the first six months of the COVID-19 pandemic, 2020. URL `https://www.biorxiv.org/content/early/2020/12/07/2020.12.01.406637`. bioRxiv.

Joseph R Luft, Robert J Collins, Nancy A Fehrman, Angela M Lauricella, Christina K Veatch, and George T DeTitta. A deliberate approach to screening for initial crystallization conditions of biological macromolecules. *J. Struct. Biol.*, 142(1):170–179, 2003.

Paul D Lyne. Structure-based virtual screening: an overview. *Drug discovery today*, 7(20): 1047–1055, 2002.

Jiankun Lyu, Sheng Wang, Trent E Balius, Isha Singh, Anat Levit, Yurii S Moroz, Matthew J O'Meara, Tao Che, Enkhjargal Algaa, Kateryna Tolmachova, et al. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019.

Jiankun Lyu, John Irwin, and Brian Shoichet. Modeling the expansion of virtual screening libraries. 2022.

Heng Ma, Austin Clyde, Anda Trifan, Venkatram Vishwanath, Arvind Ramanathan, Debsindhu Bhowmik, and Shantenu Jha. Benchmarking machine learning workloads in structural bioinformatics applications. *interactions*, 27:32, 2018.

Jun Ma, Jenny Wang, Laleh Soltan Ghoraie, Xin Men, Rui Chen, and Penggao Dai. Comprehensive expression-based isoform biomarkers predictive of drug responses based on isoform co-expression networks and clinical data. *Genomics*, 4 2019. ISSN 0888-7543. doi:10.1016/j.ygeno.2019.04.017.

Gerald M. Maggiora and Veerabahu Shanmugasundaram. *Molecular Similarity Measures*, pages 39–100. Humana Press, Totowa, NJ, 2011. ISBN 978-1-60761-839-3. doi:10.1007/978-1-60761-839-3_2. URL `https://doi.org/10.1007/978-1-60761-839-3`$_2$.

James A. Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E. Hauser, and Carlos Simmerling. ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of Chemical Theory and Computation*, 11(8): 3696–3713, 08 2015a. doi:10.1021/acs.jctc.5b00255. URL `https://doi.org/10.1021/acs.jctc.5b00255`.

James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of chemical theory and computation*, 11(8): 3696–3713, 2015b.

Nathalie Malo, James A Hanley, Sonia Cerquozzi, Jerry Pelletier, and Robert Nadon. Statistical practice in high-throughput screening data analysis. *Nature biotechnology*, 24(2): 167–175, 2006.

Matteo Manica, Ali Oskooei, Jannis Born, Vigneshwari Subramanian, Julio Sáez-Rodríguez, and María Rodríguez Martínez. Towards explainable anticancer compound sensitivity prediction via multimodal attention-based convolutional encoders. *arXiv preprint arXiv:1904.11223*, 2019.

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):1–11, 2018.

Emanuelle Machado Marinho, João Batista de Andrade Neto, Jacilene Silva, Cecília Rocha da Silva, Bruno Coelho Cavalcanti, Emmanuel Silva Marinho, and Hélio Vitoriano Nobre Júnior. Virtual screening based on molecular docking of possible inhibitors of COVID-19 main protease. *Microbial Pathogenesis*, 148:104365, 2020.

Olivier Massot, Jean-Claude Rousselle, Marie-Paule Fillion, Dominique Januel, Mathieu Plantefol, and Gilles Fillion. 5-ht1b receptors: a novel target for lithium: possible involvement in mood disorders. *Neuropsychopharmacology*, 21(4):530–541, 1999.

Giulio Mattedi, Francesca Deflorian, Jonathan S Mason, Chris de Graaf, and Francesco L Gervasio. Understanding ligand binding selectivity in a prototypical gpcr family. *Journal of chemical information and modeling*, 59(6):2830–2836, 2019.

Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *Match*, 56(2):237–248, 2006.

Mark McGann. FRED and HYBRID docking performance on standardized datasets. *Journal of Computer-Aided Molecular Design*, 26(8):897–906, 2012.

Mark R Mcgann, Harold R Almond, Anthony Nicholls, J Andrew Grant, and Frank K Brown. Gaussian docking functions. *Biopolymers: Original Research on Biomolecules*, 68 (1):76–90, 2003a.

Mark R Mcgann, Harold R Almond, Anthony Nicholls, J Andrew Grant, and Frank K Brown. Gaussian docking functions. *Biopolymers: Original Research on Biomolecules*, 68 (1):76–90, 2003b. Publisher: Wiley Online Library.

Georgia B McGaughey, Robert P Sheridan, Christopher I Bayly, J Chris Culberson, Constantine Kreatsoulas, Stacey Lindsley, Vladimir Maiorov, Jean-Francois Truchon, and Wendy D Cornell. Comparison of topological, shape, and docking methods in virtual screening. *Journal of chemical information and modeling*, 47(4):1504–1519, 2007. Publisher: ACS Publications.

Iain B McInnes, Nicole L Byers, Richard E Higgs, Jonathan Lee, William L Macias, Songqing Na, Robert A Ortmann, Guilherme Rocha, Terence P Rooney, Thomas Wehrman, et al. Comparison of baricitinib, upadacitinib, and tofacitinib mediated regulation of cytokine signaling in human leukocyte subpopulations. *Arthritis research & therapy*, 21(1):183, 2019.

Jens Meiler and David Baker. ROSETTALIGAND: Protein–small molecule docking with full side-chain flexibility. *Proteins: Structure, Function, and Bioinformatics*, 65(3):538–548, 2006. Publisher: Wiley Online Library.

Michael P. Menden, Francesco Iorio, Mathew Garnett, Ultan McDermott, Cyril H. Benes, Pedro J. Ballester, and Julio Saez-Rodriguez. Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. *PLoS ONE*, 8:e61318, 2013a. doi:10.1371/journal.pone.0061318.

Michael P Menden, Francesco Iorio, Mathew Garnett, Ultan McDermott, Cyril H Benes, Pedro J Ballester, and Julio Saez-Rodriguez. Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties. *PLoS one*, 8(4):e61318, 2013b.

Michael P. Menden, Dennis Wang, Mike J. Mason, Bence Szalai, Krishna C. Bulusu, Yuanfang Guan, Thomas Yu, Jaewoo Kang, Minji Jeon, Russ Wolfinger, Tin Nguyen, Mikhail Zaslavskiy, In Sock Jang, Zara Ghazoui, Mehmet Eren Ahsen, Robert Vogel,

Elias Chaibub Neto, Thea Norman, Eric K. Y. Tang, Mathew J. Garnett, Giovanni Y. Di Veroli, Stephen Fawell, Gustavo Stolovitzky, Justin Guinney, Jonathan R. Dry, and Julio Saez-Rodriguez. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature Communications*, 10:2674, 6 2019. doi:10.1038/s41467-019-09799-2.

Rosa Meo. Inductive databases: Towards a new generation of databases for knowledge discovery. In *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pages 1003–1007. IEEE, 2005.

Arvind S. Mer, Wail Ba-Alawi, Petr Smirnov, Yi X. Wang, Ben Brew, Janosch Ortmann, Ming-Sound Tsao, David W. Cescon, Anna Goldenberg, and Benjamin Haibe-Kains. Integrative Pharmacogenomics Analysis of Patient-Derived Xenografts. *Cancer Research*, page canres.0349.2019, 2019. ISSN 0008-5472. doi:10.1158/0008-5472.can-19-0349.

Andre Merzky, Ole Weidner, and Shantenu Jha. SAGA: A standardized access layer to heterogeneous distributed computing infrastructure. *Software-X*, 2015. doi:10.1016/j.softx.2015.03.001. URL `http://dx.doi.org/10.1016/j.softx.2015.03.001`. DOI: 10.1016/j.softx.2015.03.001.

Andre Merzky, Matteo Turilli, Manuel Maldonado, Mark Santcroos, and Shantenu Jha. Using pilot systems to execute many task workloads on supercomputers. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 61–82. Springer, 2018.

Andre Merzky, Matteo Turilli, Mikhail Titov, Aymen Al-Saadi, and Shantenu Jha. Design and performance characterization of radical-pilot on leadership-class platforms, 2021.

AD Mesecar. Structure of COVID-19 main protease bound to potent broad-spectrum non-covalent inhibitor X77, 2020. RCSB Protein Data Bank.

Bill R Miller III, T Dwight McGee Jr, Jason M Swails, Nadine Homeyer, Holger Gohlke, and Adrian E Roitberg. Mmpbsa. py: an efficient program for end-state free energy calculations. *Journal of chemical theory and computation*, 8(9):3314–3321, 2012.

Amanda J Minnich, Kevin McLoughlin, Margaret Tse, Jason Deng, Andrew Weber, Neha Murad, Benjamin D Madej, Bharath Ramsundar, Tom Rush, Stacie Calad-Thomson, et al. AMPL: A data-driven modeling pipeline for drug discovery. *Journal of Chemical Information and Modeling*, 60(4):1955–1968, 2020.

Shaher Bano Mirza, Ramin Ekhteiari Salmas, M Qaiser Fatmi, and Serdar Durdagi. Virtual screening of eighteen million compounds against dengue virus: Combined molecular docking and molecular dynamics simulations study. *Journal of Molecular Graphics and Modelling*, 66:99–107, 2016.

N Moitessier, P Englebienne, D Lee, J Lawandi, Corbeil, and CR. Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. *British journal of pharmacology*, 153(S1):S7–S26, 2008. Publisher: Wiley Online Library.

Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10(1):1–14, 2018.

Aaron Morris, William McCorkindale, The COVID Moonshot Consortium, Nir Drayman, John D. Chodera, Savaş Tay, Nir London, and Alpha A. Lee. Discovery of sars-cov-2 main protease inhibitors using a synthesis-directed de novo design model. *Chem. Commun.*, 57: 5909–5912, 2021. doi:10.1039/D1CC00050K. URL `http://dx.doi.org/10.1039/D1CC00050K`.

Christopher JL Murray, Kevin Shunji Ikuta, Fablina Sharara, Lucien Swetschinski, Gisela Robles Aguilar, Authia Gray, Chieh Han, Catherine Bisignano, Puja Rao, Eve Wool, et al. Global burden of bacterial antimicrobial resistance in 2019: a systematic analysis. *The Lancet*, 399(10325):629–655, 2022.

Christopher W Murray and Tom L Blundell. Structural biology in fragment-based drug design. *Current opinion in structural biology*, 20(4):497–507, 2010.

Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012a.

Michael M Mysinger, Michael Carchia, John J Irwin, and Brian K Shoichet. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *Journal of medicinal chemistry*, 55(14):6582–6594, 2012b. Publisher: ACS Publications.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307, 2017.

Anthony Nicholls. What do we know and when do we know it? *Journal of computer-aided molecular design*, 22(3-4):239–255, 2008.

Sara E Nichols, Riccardo Baron, Anthony Ivetac, and J Andrew McCammon. Predictive power of molecular dynamics receptor structures in virtual screening. *Journal of chemical information and modeling*, 51(6):1439–1446, 2011.

Noel M O'Boyle. Towards a universal smiles representation-a standard method to generate canonical smiles based on the inchi. *Journal of cheminformatics*, 4(1):1–14, 2012.

TK OEChem. Openeye scientific software. *Inc., Santa Fe, NM, USA*, 2012a.

TK OEChem. Openeye scientific software. *Inc., Santa Fe, NM, USA*, 2012b.

Noriaki Okimoto, Noriyuki Futatsugi, Hideyoshi Fuji, Atsushi Suenaga, Gentaro Morimoto, Ryoko Yanai, Yousuke Ohno, Tetsu Narumi, and Makoto Taiji. High-performance drug discovery: computational screening by combining docking and molecular dynamics simulations. *PLoS computational biology*, 5(10), 2009.

Christopher J Oldfield and A Keith Dunker. Intrinsically disordered proteins and intrinsically disordered protein regions. *Annu Rev Biochem*, 83(1):553–584, 2014.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):1–14, 2017.

Tudor I Oprea. Current trends in lead discovery: are we looking for the appropriate properties? *Molecular diversity*, 5(4):199–208, 2000. Publisher: Springer.

Christine A Orengo, Annabel E Todd, and Janet M Thornton. From protein structure to function. *Current opinion in structural biology*, 9(3):374–382, 1999.

World Health Organization. Who covid-19 dashboard, 19.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Alexander Partin, Thomas Brettin, Yvonne A Evrard, Yitan Zhu, Hyunseung Yoo, Fangfang Xia, Songhao Jiang, Austin Clyde, Maulik Shukla, Michael Fonstein, et al. Learning curves for drug response prediction in cancer cell lines. *BMC bioinformatics*, 22(1):1–18, 2021.

Akash Parvatikar, Gabriel S. Vacaliuc, Arvind Ramanathan, and S. Chakra Chennubhotla. Anca: Anharmonic conformational analysis of biomolecular simulations. *Biophysical Journal*, 114(9):2040–2043, 2018. ISSN 0006-3495. doi:https://doi.org/10.1016/j.bpj.2018.03.021. URL `https://www.sciencedirect.com/science/article/pii/S0006349518303886`.

Manuel Pastor, Gabriele Cruciani, Iain McLay, Stephen Pickett, and Sergio Clementi. Grid-independent descriptors (grind): a novel class of alignment-independent three-dimensional molecular descriptors. *Journal of medicinal chemistry*, 43(17):3233–3243, 2000.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Hitesh Patel, Wolf-Dietrich Ihlenfeldt, Philip N Judson, Yurii S Moroz, Yuri Pevzner, Megan L Peach, Victorien Delannée, Nadya I Tarasova, and Marc C Nicklaus. Savi, in silico generation of billions of easily synthesizable compounds through expert-system type rules. *Scientific data*, 7(1):1–14, 2020.

Imre Pechan and Bela Feher. Molecular docking on fpga and gpu platforms. In *2011 21st International Conference on Field Programmable Logic and Applications*, pages 474–477. IEEE, 2011.

Janaina Cruz Pereira, Ernesto Raul Caffarena, and Cicero Nogueira dos Santos. Boosting docking-based virtual screening with deep learning. *Journal of chemical information and modeling*, 56(12):2495–2506, 2016.

Thanigaimalai Pillaiyar, Manoj Manickam, Vigneshwaran Namasivayam, Yoshio Hayashi, and Sang-Hun Jung. An overview of severe acute respiratory syndrome–coronavirus (sars-cov) 3cl protease inhibitors: Peptidomimetics and small molecule chemotherapy. *Journal of Medicinal Chemistry*, 59(14):6595–6628, 07 2016. doi:10.1021/acs.jmedchem.5b01461. URL `https://doi.org/10.1021/acs.jmedchem.5b01461`.

AR Plastino and A Plastino. Tsallis' entropy, ehrenfest theorem and information theory. *Physics Letters A*, 177(3):177–179, 1993.

Pavel G Polishchuk, Timur I Madzhidov, and Alexandre Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27(8):675–679, 2013.

Jay W Ponder and David A Case. Force fields for protein simulations. *Advances in protein chemistry*, 66:27–85, 2003.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

J.X. Qiao, R. Zeng, Y.F. Wang, Y.S. Li, R. Yao, Y.L. Zhou, P. Chen, J. Lei, and S.Y. Yang. Crystal structure of the SARS-CoV-2 main protease in complex with Telaprevir, 2020. RCSB Protein Data Bank.

Obdulia Rabal, Fares Ibrahim Amr, and Julen Oyarzabal. Novel scaffold fingerprint (sfp): applications in scaffold hopping and scaffold-based selection of diverse compounds. *Journal of chemical information and modeling*, 55(1):1–18, 2015.

Arjun Raghuraman, Philip D Mosier, and Umesh R Desai. Finding a needle in a haystack: development of a combinatorial virtual screening approach for identifying high specificity heparin/heparan sulfate sequence (s). *Journal of medicinal chemistry*, 49(12):3553–3562, 2006.

Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of chemical information and modeling*, 57(4):942–957, 2017.

Raziur Rahman and Ranadip Pal. Cancer Bioinformatics. *Methods in molecular biology (Clifton, N.J.)*, 1878:227–241, 2018. ISSN 1064-3745. doi:10.1007/978-1-4939-8868-6_14.

Eric M Rains and Neil JA Sloane. On cayley's enumeration of alkanes (or 4-valent trees). *Journal of Integer Sequences*, 2:Art–No, 1999.

Pichai Raman, Samuel Zimmerman, Komal S. Rathi, Laurence de Torrenté, Mahdi Sarmady, Chao Wu, Jeremy Leipzig, Deanne M. Taylor, Aydin Tozeren, and Jessica C. Mar. A comparison of survival analysis methods for cancer gene expression RNA-Sequencing data. *Cancer Genetics*, 235-236:1–12, 4 2019. ISSN 2210-7762. doi:10.1016/j.cancergen.2019.04.004.

Arvind Ramanathan, Andrej J Savol, Christopher J Langmead, Pratul K Agarwal, and Chakra S Chennubhotla. Discovering conformational sub-states relevant to protein function. *PLoS One*, 6(1):e15827, 2011.

Arvind Ramanathan, Akash Parvatikar, Srinivas C Chennubhotla, Yang Mei, and Sangita C Sinha. Transient unfolding and long-range interactions in viral BCL2 M11 enable binding to the BECN1 BH3 domain. *Biomolecules*, 10(9):1308, 2020.

Arvind Ramanathan, Heng Ma, Akash Parvatikar, and S Chakra Chennubhotla. Artificial intelligence techniques for integrative structural biology of intrinsically disordered proteins. *Current Opinion in Structural Biology*, 66:216–224, 2021.

Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. `https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837`.

Athri D Rathnayake, Jian Zheng, Yunjeong Kim, Krishani Dinali Perera, Samantha Mackin, David K Meyerholz, Maithri M Kashipathy, Kevin P Battaile, Scott Lovell, Stanley Perlman, et al. 3C-like protease inhibitors block coronavirus replication *in vitro* and improve survival in MERS-CoV-infected mice. *Sci. Transl. Med.*, 12(557):eabc5332, 2020.

Vivek P Raut, Madhuri A Agashe, Steven J Stuart, and Robert A Latour. Molecular dynamics simulations of peptide- surface interactions. *Langmuir*, 21(4):1629–1639, 2005.

Revanur Ravindra and Roland Winter. On the temperature–pressure free-energy landscape of proteins. *ChemPhysChem*, 4(4):359–365, 2003.

Nima Razzaghi-Asl, Saghi Sepehri, Ahmad Ebadi, Ramin Miri, and Sara Shahabipour. Effect of biomolecular conformation on docking simulation: A case study on a potent HIV-1 protease inhibitor. *Iranian journal of pharmaceutical research: IJPR*, 14(3):785, 2015. Publisher: Shahid Beheshti University of Medical Sciences.

Matthew G Rees, Brinton Seashore-Ludlow, Jaime H Cheah, Drew J Adams, Edmund V Price, Shubhroz Gill, Sarah Javaid, Matthew E Coletti, Victor L Jones, Nicole E Bodycombe, Christian K Soule, Benjamin Alexander, Ava Li, Philip Montgomery, Joanne D Kotz, C Suk-Yee Hon, Benito Munoz, Ted Liefeld, Vlado Dančík, Daniel A Haber, Clary B Clish, Joshua A Bittker, Michelle Palmer, Bridget K Wagner, Paul A Clemons, Alykhan F Shamji, and Stuart L Schreiber. Correlating chemical sensitivity and basal gene expression reveals mechanism of action. *Nature Chemical Biology*, 12:nchembio.1986, 2015. ISSN 1552-4469. doi:10.1038/nchembio.1986.

Matthew G Rees, Brinton Seashore-Ludlow, and Paul A Clemons. Computational Analyses Connect Small-Molecule Sensitivity to Cellular Features Using Large Panels of Cancer Cell Lines. *Methods in molecular biology (Clifton, N.J.)*, 1888:233–254, 1 2019. doi:10.1007/978-1-4939-8891-4_14.

Kelly E. Regan-Fendt, Jielin Xu, Mallory DiVincenzo, Megan C. Duggan, Reena Shakya, Ryejung Na, William E. Carson, Philip R. O. Payne, and Fuhai Li. Synergy from gene expression and network mining (SynGeNet) method predicts synergistic drug combinations for diverse melanoma genomic subtypes. *npj Systems Biology and Applications*, 5:6, 2 2019. doi:10.1038/s41540-019-0085-4.

William C Reinhold, Sudhir Varma, Fabricio Sousa, Margot Sunshine, Ogan D Abaan, Sean R Davis, Spencer W Reinhold, Kurt W Kohn, Joel Morris, Paul S Meltzer, et al. Nci-60 whole exome sequencing and pharmacological cellminer analyses. *PloS one*, 9(7): e101670, 2014.

Jean-Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3): 722–730, 2015.

Jean-Louis Reymond and Mahendra Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS chemical neuroscience*, 3(9):649–657, 2012.

Steven W Rick. Increasing the efficiency of free energy calculations using parallel tempering and histogram reweighting. *Journal of chemical theory and computation*, 2(4):939–946, 2006.

A Rizzi, PB Grinaway, DL Parton, MR Shirts, K Wang, P Eastman, M Friedrichs, VS Pande, K Branson, DL Mobley, et al. Yank: A gpu-accelerated platform for alchemical free energy calculations [internet].

Andrea Rizzi, Steven Murkli, John N. McNeill, Wei Yao, Matthew Sullivan, Michael K. Gilson, Michael W. Chiu, Bruce C. Gibb, David L. Mobley, and John D. Chodera. Overview of the sampl6 host–guest binding affinity prediction challenge. *Journal of Computer-Aided Molecular Design*, 32(10):937–963, Oct 2018. ISSN 1573-4951. doi:10.1007/s10822-018-0170-6. URL `https://doi.org/10.1007/s10822-018-0170-6`.

Bruno Rizzuti, Laura Ceballos-Laita, David Ortega-Alarcon, Ana Jimenez-Alesanco, Sonia Vega, Fedora Grande, Filomena Conforti, Olga Abian, and Adrian Velazquez-Campoy. Sub-micromolar inhibition of sars-cov-2 3clpro by natural compounds. *Pharmaceuticals*, 14(9), 2021. ISSN 1424-8247. doi:10.3390/ph14090892. URL `https://www.mdpi.com/1424-8247/14/9/892`.

Daniel R Roe and Thomas E Cheatham III. Ptraj and cpptraj: software for processing and analysis of molecular dynamics trajectory data. *Journal of chemical theory and computation*, 9(7):3084–3095, 2013.

David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50:742–754, 2010a. ISSN 1549-9596. doi:10.1021/ci100050t.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010b.

Mark Rogge and David R Taft. *Preclinical drug development*. CRC Press, 2016.

Monica Rosas-Lemus, George Minasov, Ludmilla Shuvalova, Nicole L. Inniss, Olga Kiryukhina, Joseph Brunzelle, and Karla J. F. Satchell. High-resolution structures of the SARS-CoV-2 2'-O-methyltransferase reveal strategies for structure-based inhibitor design. *Science Signaling*, 13(651), 2020. ISSN 1945-0877. doi:10.1126/scisignal.abe1202. URL `https://stke.sciencemag.org/content/13/651/eabe1202`.

Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Verónica Ruiz-Torres, Jose Antonio Encinar, María Herranz-López, Almudena Pérez-Sánchez, Vicente Galiano, Enrique Barrajón-Catalán, and Vicente Micol. An updated review on marine anticancer compounds: The use of virtual screening for the discovery of small-molecule cancer drugs. *Molecules*, 22(7):1037, 2017.

Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical Review Letters*, 108:058301, 2012. ISSN 0031-9007. doi:10.1103/physrevlett.108.058301.

Aymen Al Saadi, Dario Alfe, Yadu Babuji, Agastya Bhati, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, Austin Clyde, et al. Impeccable: integrated modeling pipeline for covid cure by assessing better leads. In *50th International Conference on Parallel Processing*, pages 1–12, 2021.

Michael Dominic Sacco, Chunlong Ma, Panagiotis Lagarias, Ang Gao, Julia Alma Townsend, Xiangzhi Meng, Peter Dube, Xiujun Zhang, Yanmei Hu, Naoya Kitamura, et al. Structure and inhibition of the SARS-CoV-2 main protease reveal strategy for developing dual inhibitors against Mpro and cathepsin L. *Sci. Adv.*, 6(50):eabe0751, 2020.

S. K. Sadiq, D. W. Wright, O. A. Kenway, and P. V. Coveney. Accurate ensemble molecular dynamics binding free energy ranking of multidrug-resistant HIV-1 proteases. *J. Chem. Inf. Model.*, 50(5):890–905, 2010. doi:10.1021/ci100007w.

Sebastian Salentin, Sven Schreiber, V Joachim Haupt, Melissa F Adasme, and Michael Schroeder. Plip: fully automated protein–ligand interaction profiler. *Nucleic acids research*, 43(W1):W443–W447, 2015.

Diogo Santos-Martins, Leonardo Solis-Vasquez, Andreas F Tillack, Michel F Sanner, Andreas Koch, and Stefano Forli. Accelerating autodock4 with gpus and gradient-based local search. *Journal of Chemical Theory and Computation*, 17(2):1060–1073, 02 2021. doi:10.1021/acs.jctc.0c01006. URL `https://doi.org/10.1021/acs.jctc.0c01006`.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20: 61–80, 2009. ISSN 1045-9227. doi:10.1109/tnn.2008.2005605.

Johannes Schiebel, Stefan G Krimmer, Karine Röwer, Anna Knörlein, Xiaojie Wang, Ah Young Park, Martin Stieler, Frederik R Ehrmann, Kan Fu, Nedyalka Radeva, et al. High-throughput crystallography: reliable and efficient identification of fragment hits. *Structure*, 24(8):1398–1409, 2016.

Gisbert Schneider. Virtual screening: an endless staircase? *Nature Reviews Drug Discovery*, 9(4):273, 2010.

Ansgar Schuffenhauer, Nathan Brown, Paul Selzer, Peter Ertl, and Edgar Jacoby. Relationships between molecular complexity, biological activity, and structural diversity. *Journal of chemical information and modeling*, 46(2):525–535, 2006.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

Oliver B Scott and A W Edith Chan. ScaffoldGraph: an open-source library for the generation and analysis of molecular scaffold networks and scaffold trees. *Bioinformatics*, 03 2020a. ISSN 1367-4803. doi:10.1093/bioinformatics/btaa219. URL `https://doi.org/10.1093/bioinformatics/btaa219`. btaa219.

Oliver B Scott and AW Chan. Scaffoldgraph: an open-source library for the generation and analysis of molecular scaffold networks and scaffold trees. *Bioinformatics*, 2020b.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

Nayim Sepay, Aishwarya Sekar, Umesh C Halder, Abdullah Alarifi, and Mohd Afzal. Anti-COVID-19 terpenoid from marine sources: A docking, admet and molecular dynamics study. *Journal of Molecular Structure*, 1228:129433, 2021.

Robert F Service. 'the game has changed.'ai triumphs at protein folding, 2020.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

Robert P Sheridan, Georgia B McGaughey, and Wendy D Cornell. Multiple protein structures and multiple ligands: effects on the apparent goodness of virtual screening results. *Journal of computer-aided molecular design*, 22(3-4):257–265, 2008.

Leming Shi, Gregory Campbell, Wendell D Jones, Fabien Campagne, Zhining Wen, Stephen J Walker, Zhenqiang Su, Tzu-Ming Chu, Federico M Goodsaid, Lajos Pusztai, et al. The microarray quality control (maqc)-ii study of common practices for the development and validation of microarray-based predictive models. *Nature biotechnology*, 28 (8):827, 2010.

Michael R Shirts and John D Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of chemical physics*, 129(12):124105, 2008.

AN Shivanyuk, SV Ryabukhin, A Tolmachev, AV Bogolyubsky, DM Mykytenko, AA Chupryna, W Heilman, and AN Kostyuk. Enamine real database: Making chemical diversity real. *Chemistry today*, 25(6):58–59, 2007.

Robert H Shoemaker. The nci60 human tumour cell line anticancer drug screen. *Nature Reviews Cancer*, 6(10):813, 2006.

Brian K Shoichet. Virtual screening of chemical libraries. *Nature*, 432(7019):862, 2004.

Rohit Shukla, Harish Shukla, Amit Sonkar, Tripti Pandey, and Timir Tripathi. Structure-based screening and molecular dynamics simulations offer novel natural compounds as potential inhibitors of mycobacterium tuberculosis isocitrate lyase. *Journal of Biomolecular Structure and Dynamics*, 36(8):2045–2057, 2018.

Zeenat A. Shyr, Kirill Gorshkov, Catherine Z. Chen, and Wei Zheng. Drug discovery strategies for SARS-CoV-2. *Journal of Pharmacology and Experimental Therapeutics*, 375(1):127–138, 2020. ISSN 0022-3565. doi:10.1124/jpet.120.000123. URL `https://jpet.aspetjournals.org/content/375/1/127`.

Pavel Sidorov, Stefan Naulaerts, Jeremy Ariey-Bonnet, Eddy Pasquier, and Pedro Ballester. Predicting synergism of cancer drug combinations using NCI-ALMANAC data. *bioRxiv*, page 504076, 2018. doi:10.1101/504076.

Pavel Sidorov, Stefan Naulaerts, Jeremy Ariey-Bonnet, Eddy Pasquier, and Pedro Ballester. Predicting synergism of cancer drug combinations using nci-almanac data. *Frontiers in chemistry*, 7:509, 2019a.

Pavel Sidorov, Stefan Naulaerts, Jérémy Ariey-Bonnet, Eddy Pasquier, and Pedro J. Ballester. Predicting Synergism of Cancer Drug Combinations Using NCI-ALMANAC Data. *Frontiers in Chemistry*, 7:509, 7 2019b. ISSN 2296-2646. doi:10.3389/fchem.2019.00509.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc

Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pages 243–246, 2015.

Dakshanamurthy Sivanesan, Rajendram V Rajnarayanan, Jason Doherty, and Nagarajan Pattabiraman. In-silico screening using flexible ligand binding pockets: a molecular dynamics-based approach. *Journal of computer-aided molecular design*, 19(4):213–228, 2005.

Miha Skalic, José Jiménez, Davide Sabbadin, and Gianni De Fabritiis. Shape-based generative modeling for de novo drug design. *Journal of chemical information and modeling*, 59 (3):1205–1214, 2019.

Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W Lowe. Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395, 2014.

David Slochower, Niel Henriksen, Lee-Ping Wang, John Chodera, David Mobley, and Michael Gilson. Binding thermodynamics of host-guest systems with SMIRNOFF99Frosst 1.0.5 from the Open Force Field Initiative. 7 2019. doi:10.26434/chemrxiv.9159872.v1.

Eric Smalley. Ai-powered drug discovery captures pharma interest, 2017.

Micholas Smith and Jeremy C Smith. Repurposing therapeutics for covid-19: supercomputer-based docking to the sars-cov-2 viral spike protein and viral spike protein-human ace2 interface. 2020. doi:https://doi.org/10.26434/chemrxiv.11871402.v4.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't Decay the Learning Rate, Increase the Batch Size. 2017.

Martin Stahl. Modifications of the scoring function in flexx for virtual screening applications. *Perspectives in Drug Discovery and Design*, 20(1):83–98, 2000.

J E Staunton, D K Slonim, H A Coller, P Tamayo, M J Angelo, J Park, U Scherf, J K Lee, W O Reinhold, J N Weinstein, J P Mesirov, E S Lander, and T R Golub. Chemosensitivity prediction by transcriptional profiling. *Proceedings of the National Academy of Sciences*, 98:10787–10792, 9 2001. ISSN 0027-8424. doi:10.1073/pnas.191368598.

Tomaž Stepišnik, Blaž Škrlj, Jörg Wicker, and Dragi Kocev. A comprehensive comparison of molecular feature representations for use in predictive modeling. *Computers in Biology and Medicine*, 130:104197, 2021.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Rick Stevens, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, and David Brown. Ai for science: Report on the department of energy (doe) town halls on artificial intelligence (ai) for science. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2020.

Thomas J Struble, Juan C Alvarez, Scott P Brown, Milan Chytil, Justin Cisar, Renee L DesJarlais, Ola Engkvist, Scott A Frank, Daniel R Greve, Daniel J Griffin, et al. Current and future roles of artificial intelligence in medicinal chemistry synthesis. *Journal of medicinal chemistry*, 63(16):8667–8682, 2020.

Ran Su, Xinyi Liu, Leyi Wei, and Quan Zou. Deep-Resp-Forest: A deep forest model to predict anti-cancer drug response. *Methods*, 2 2019. ISSN 1046-2023. doi:10.1016/j.ymeth.2019.02.009.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

Jocelyn Sunseri, Jonathan E. King, Paul G. Francoeur, and David Ryan Koes. Convolutional neural network scoring and minimization in the d3r 2017 community challenge. *Journal of Computer-Aided Molecular Design*, 33(1):19–34, Jan 2019. ISSN 1573-4951. doi:10.1007/s10822-018-0133-y. URL `https://doi.org/10.1007/s10822-018-0133-y`.

Miroslav Suruzhon, Tharindu Senapathi, Michael S Bodnarchuk, Russell Viner, Ian D Wall, Christopher B Barnett, Kevin J Naidoo, and Jonathan W Essex. Protocaller: Robust automation of binding free energy calculations. *Journal of Chemical Information and Modeling*, 2020.

K. Tan, N.I. Maltseva, L.F. Welk, R.P. Jedrzejczak, A. Joachimiak, and Center for Structural Genomics of Infectious Diseases (CSGID). doi:10.2210/pdb7JU7/pdb.

Niek Tax, Sander Bockting, and Djoerd Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management*, 51(6):757–772, 2015. ISSN 0306-4573. doi:10.1016/j.ipm.2015.07.002.

Christopher S Thom, Chintan D Jobaliya, Kimberly Lorenz, Jean Ann Maguire, Alyssa Gagne, Paul Gadue, Deborah L French, and Benjamin F Voight. Machine learning-based identification and cellular validation of Tropomyosin 1 as a genetic inhibitor of hematopoiesis. *bioRxiv*, page 631895, 2019. doi:10.1101/631895.

Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*, volume 11. John Wiley & Sons, 2008.

Richard E Trager, Paul Giblock, Sherwin Soltani, Amit A Upadhyay, Bhanu Rekapalli, and Yuri K Peterson. Docking optimization, variance and promiscuity for large-scale drug-like chemical space using high performance computing architectures. *Drug discovery today*, 21 (10):1672–1680, 2016.

Nicolas Triballeau, Francine Acher, Isabelle Brabet, Jean-Philippe Pin, and Hugues-Olivier Bertrand. Virtual screening workflow development guided by the "receiver operating characteristic" curve approach. application to high-throughput docking on metabotropic glutamate receptor subtype 4. *Journal of medicinal chemistry*, 48(7):2534–2547, 2005.

Ashutosh Tripathi and Vytas A Bankaitis. Molecular docking: From lock and key to combination lock. *Journal of molecular medicine and clinical applications*, 2(1), 2017.

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

Jean-François Truchon and Christopher I Bayly. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. *Journal of chemical information and modeling*, 47(2):488–508, 2007.

Matteo Turilli, Mark Santcroos, and Shantenu Jha. A comprehensive perspective on pilot-job systems. *ACM Comput. Surv.*, 51(2):43:1–43:32, April 2018. ISSN 0360-0300. doi:10.1145/3177851. URL `http://doi.acm.org/10.1145/3177851`.

Matteo Turilli, Vivek Balasubramanian, Andre Merzky, Ioannis Paraskevakos, and Shantenu Jha. Middleware building blocks for workflow systems. *Computing in Science & Engineering (CiSE) special issue on Incorporating Scientific Workflows in Computing Research Processes*, 2019a. URL `https://arxiv.org/abs/1903.10057`.

Matteo Turilli, Andre Merzky, Thomas Naughton, Wael Elwasif, and Shantenu Jha. Characterizing the performance of executing many-tasks on summit. In *2019 IEEE/ACM Third Annual Workshop on Emerging Parallel and Distributed Runtime Systems and Middleware (IPDRM)*, pages 18–25. IEEE, 2019b.

Thomas Unterthiner, Andreas Mayr, Günter Klambauer, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, and Sepp Hochreiter. Deep learning as an opportunity in virtual screening. In *Proceedings of the deep learning workshop at NIPS*, volume 27, pages 1–9, 2014.

Julie S Valastyan and Susan Lindquist. Mechanisms of protein-folding diseases at a glance. *Disease models & mechanisms*, 7(1):9–14, 2014.

Marat Valiev, Eric J Bylaska, Niranjan Govind, Karol Kowalski, Tjerk P Straatsma, Hubertus JJ Van Dam, Dunyou Wang, Jarek Nieplocha, Edoardo Apra, Theresa L Windus, et al. Nwchem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications*, 181(9):1477–1489, 2010.

Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18 (6):463–477, 2019.

Gary J. Van Berkel and Vilmos Kertesz. An open port sampling interface for liquid introduction atmospheric pressure ionization mass spectrometry. *Rapid Commun. Mass Spectrom.*, 29(19):1749–1756, 2015. doi:https://doi.org/10.1002/rcm.7274.

Niek van Hilten, Florent Chevillard, and Peter Kolb. Virtual compound libraries in computer-assisted drug discovery. *Journal of chemical information and modeling*, 59(2):644–651, 2019.

Peter Vanhee, Almer M van der Sloot, Erik Verschueren, Luis Serrano, Frederic Rousseau, and Joost Schymkowitz. Computational design of peptide ligands. *Trends in biotechnology*, 29(5):231–239, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. 2017a.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017b.

Marcel L Verdonk, Valerio Berdini, Michael J Hartshorn, Wijnand TM Mooij, Christopher W Murray, Richard D Taylor, and Paul Watson. Virtual screening using protein- ligand docking: avoiding artificial enrichment. *Journal of chemical information and computer sciences*, 44(3):793–806, 2004. Publisher: ACS Publications.

Gennady M Verkhivker, Paul A Rejto, Djamal Bouzida, Sandra Arthurs, Anthony B Colson, Stephan T Freer, Daniel K Gehlhaar, Veda Larson, Brock A Luty, Tami Marrone, et al. Parallel simulated tempering dynamics of ligand–protein binding with ensembles of protein conformations. *Chemical physics letters*, 337(1-3):181–189, 2001.

Josh Vincent Vermaas, Ada Sedova, Matthew B Baker, Swen Boehm, David M Rogers, Jeff Larkin, Jens Glaser, Micholas D Smith, Oscar Hernandez, and Jeremy C Smith. Supercomputing pipelines search for therapeutics against covid-19. *Computing in Science & Engineering*, 23(1):7–16, 2020.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order Matters: Sequence to sequence for sets. 2015.

Hans G Vogel. *Drug discovery and evaluation: pharmacological assays*. Springer Science & Business Media, 2002.

Konstantinos Vougas, Theodore Sakelaropoulos, Athanassios Kotsinas, George-Romanos P. Foukas, Andreas Ntargaras, Filippos Koinis, Alexander Polyzos, Vassilis Myrianthopoulos, Hua Zhou, Sonali Narang, Vassilis Georgoulias, Leonidas Alexopoulos, Iannis Aifantis, Paul A. Townsend, Petros Sfikakis, Rebecca Fitzgerald, Dimitris Thanos, Jiri Bartek, Russell Petty, Aristotelis Tsirigos, and Vassilis G. Gorgoulis. Machine learning and data mining frameworks for predicting drug response in cancer: An overview and a novel in silico screening process based on association rule mining. *Pharmacology & Therapeutics*, page 107395, 7 2019. ISSN 0163-7258. doi:10.1016/j.pharmthera.2019.107395.

Wayne Vuong, Muhammad Bashir Khan, Conrad Fischer, Elena Arutyunova, Tess Lamer, Justin Shields, Holly A Saffran, Ryan T McKay, Marco J van Belkum, Michael Joyce, et al. Feline coronavirus drug inhibits the main protease of SARS-CoV-2 and blocks virus replication. *Nat. Commun.*, 11:4282, 2020.

Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

W Patrick Walters. Going further than lipinski's rule in drug design. *Expert opinion on drug discovery*, 7(2):99–107, 2012.

W Patrick Walters and Regina Barzilay. Critical assessment of ai in drug discovery. *Expert opinion on drug discovery*, 16(9):937–947, 2021.

W Patrick Walters and Renxiao Wang. New trends in virtual screening, 2020.

Qian Wan and Ranadip Pal. An Ensemble Based Top Performing Approach for NCI-DREAM Drug Sensitivity Prediction Challenge. *PLoS ONE*, 9:e101183, 6 2014. doi:10.1371/journal.pone.0101183.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

Junmei Wang, Romain M. Wolf, James W. Caldwell, Peter A. Kollman, and David A. Case. Development and testing of a general amber force field. *Journal of Computational Chemistry*, 25(9):1157–1174, 2004a. doi:https://doi.org/10.1002/jcc.20035. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20035`.

Junmei Wang, Romain M Wolf, James W Caldwell, Peter A Kollman, and David A Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004b.

Kai Wang, John D Chodera, Yanzhi Yang, and Michael R Shirts. Identifying ligand binding sites and poses using gpu-accelerated hamiltonian replica exchange molecular dynamics. *Journal of computer-aided molecular design*, 27(12):989–1007, 2013.

Xuewei Wang, Zhifu Sun, Michael T. Zimmermann, Andrej Bugrim, and Jean-Pierre Kocher. Predict drug sensitivity of cancer cells with pathway activity inference. *BMC Medical Genomics*, 12:15, 1 2019. doi:10.1186/s12920-018-0449-4.

Timothy D Warner and Jane A Mitchell. Cyclooxygenase-3 (cox-3): filling in the gaps toward a cox continuum? *Proceedings of the National Academy of Sciences*, 99(21):13371–13373, 2002.

Gregory L Warren, C Webster Andrews, Anna-Maria Capelli, Brian Clarke, Judith LaLonde, Millard H Lambert, Mika Lindvall, Neysa Nevins, Simon F Semus, Stefan Senger, et al. A critical assessment of docking programs and scoring functions. *Journal of medicinal chemistry*, 49(20):5912–5931, 2006.

John A Wass. First steps in experimental design-the screening experiment. *Journal of validation technology*, 16(2):49, 2010.

Dietmar Weichert and Peter Gmeiner. Covalent molecular probes for class ag protein-coupled receptors: advances and applications. *ACS chemical biology*, 10(6):1376–1386, 2015.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988. doi:10.1021/ci00057a005.

Daniel S Wigh, Jonathan M Goodman, and Alexei A Lapkin. A review of molecular representation in the age of machine learning. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, page e1603, 2022.

Samuel A Williams, Wade C Anderson, Marianne T Santaguida, and Scott J Dylla. Patient-derived xenografts, the cancer stem cell paradigm, and cancer pathobiology in the 21st century. *Laboratory investigation*, 93(9):970, 2013.

David Wilton, Peter Willett, Kevin Lawson, and Graham Mullier. Comparison of ranking methods for virtual screening in lead-discovery programs. *Journal of chemical information and computer sciences*, 43(2):469–474, 2003.

Martyn D Winn, Charles C Ballard, Kevin D Cowtan, Eleanor J Dodson, Paul Emsley, Phil R Evans, Ronan M Keegan, Eugene B Krissinel, Andrew GW Leslie, Airlie McCoy, et al. Overview of the CCP4 suite and current developments. *Acta Crystallogr. D Biol. Crystallogr.*, 67(4):235–242, 2011.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

Chi C Wong, Inigo Martincorena, Alistair G Rust, Mamunur Rashid, Constantine Alifrangis, Ludmil B Alexandrov, Jessamy C Tiffen, Christina Kober, Chronic Myeloid Disorders Working Group of the International Cancer Genome Consortium, Anthony R Green, Charles E Massie, Jyoti Nangalia, Stella Lempidaki, Hartmut Döhner, Konstanze Döhner, Sarah J Bray, Ultan McDermott, Elli Papaemmanuil, Peter J Campbell, and David J Adams. Inactivating CUX1 mutations promote tumorigenesis. *Nature Genetics*, 46:ng.2846, 2013. ISSN 1546-1718. doi:10.1038/ng.2846.

Godwin Woo, Michael Fernandez, Michael Hsing, Nathan A Lack, Ayse Derya Cavga, and Artem Cherkasov. DeepCOP – Deep Learning–Based Approach to Predict Gene

Regulating Effects of Small Molecules. *Bioinformatics*, 2019. ISSN 1367-4803. doi:10.1093/bioinformatics/btz645.

Hyun-Myung Woo, Xiaoning Qian, Li Tan, Shantenu Jha, Francis J Alexander, Edward R Dougherty, and Byung-Jun Yoon. Optimal decision making in high-throughput virtual screening pipelines. *arXiv preprint arXiv:2109.11683*, 2021.

Christopher J Woods, Michael A King, and Jonathan W Essex. Replica-exchange-based free-energy methods. In *New Algorithms for Macromolecular Simulation*, pages 251–259. Springer, 2006.

Justin M. Wozniak, Rajeev Jain, Prasanna Balaprakash, Jonathan Ozik, Nicholson T. Collier, John Bauer, Fangfang Xia, Thomas Brettin, Rick Stevens, Jamaludin Mohd-Yusof, Cristina Garcia Cardona, Brian Van Essen, and Matthew Baughman. CANDLE/Supervisor: a workflow framework for machine learning applied to cancer research. *BMC Bioinformatics*, 19:491, 2018a. doi:10.1186/s12859-018-2508-4.

Justin M Wozniak, Rajeev Jain, Prasanna Balaprakash, Jonathan Ozik, Nicholson T Collier, John Bauer, Fangfang Xia, Thomas Brettin, Rick Stevens, Jamaludin Mohd-Yusof, et al. Candle/supervisor: A workflow framework for machine learning applied to cancer research. *BMC bioinformatics*, 19(18):59–69, 2018b.

DW Wright, A Devitt-Lee, A Clyde, K Palani, F Xia, M Turilli, J Karanicolas, S Jha, R Stevens, JD Chodera, et al. Combining molecular simulation and machine learning to inspire improved cancer therapy.

Renyi Wu, Lujing Wang, Hsiao-Chen Dina Kuo, Ahmad Shannar, Rebecca Peter, Pochung Jordan Chou, Shanyi Li, Rasika Hudlikar, Xia Liu, Zhigang Liu, George J. Poiani, Louis Amorosa, Luigi Brunetti, and Ah-Ng Kong. An update on current therapeutic drugs treating COVID-19. *Current Pharmacology Reports*, 6(3):56–70, 2020. doi:10.1007/s40495-020-00216-7. URL `https://doi.org/10.1007/s40495-020-00216-7`.

Yulun Wu, Nicholas Choma, Andrew Chen, Mikaela Cashman, Érica T Prates, Manesh Shah, Verónica G Melesse Vergara, Austin Clyde, Thomas S Brettin, Wibe A de Jong, et al. Spatial graph attention and curiosity-driven policy for antiviral drug discovery. *arXiv preprint arXiv:2106.02190*, 2021.

Fangfang Xia, Maulik Shukla, Thomas Brettin, Cristina Garcia-Cardona, Judith Cohn, Jonathan E. Allen, Sergei Maslov, Susan L. Holbeck, James H. Doroshow, Yvonne A. Evrard, Eric A. Stahlberg, and Rick L. Stevens. Predicting tumor cell line response to drug pairs with deep learning. *BMC Bioinformatics*, 19, 2018. doi:10.1186/s12859-018-2509-3.

Fangfang Xia, Jonathan Allen, Prasanna Balaprakash, Thomas Brettin, Cristina Garcia-Cardona, Austin Clyde, Judith Cohn, James Doroshow, Xiaotian Duan, Veronika Dubinkina, et al. A cross-study analysis of drug response prediction in cancer cell lines. *Briefings in bioinformatics*, 23(1):bbab356, 2022.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 2015.

Youjun Xu, Kangjie Lin, Shiwei Wang, Lei Wang, Chenjing Cai, Chen Song, Luhua Lai, and Jianfeng Pei. Deep learning for molecular generation. *Future medicinal chemistry*, 11(6): 567–597, 2019.

Ziqiao Xu, Orrette R Wauchope, and Aaron Terrence Frank. Navigating chemical space by interfacing generative artificial intelligence and molecular docking. *bioRxiv*, 2020.

Xiaoyu Xue, Hongwei Yu, Haitao Yang, Fei Xue, Zhixin Wu, Wei Shen, Jun Li, Zhe Zhou, Yi Ding, Qi Zhao, Xuejun C. Zhang, Ming Liao, Mark Bartlam, and Zihe Rao. Structures of two coronavirus main proteases: Implications for substrate binding and antiviral drug design. *Journal of Virology*, 82(5):2515–2527, 2008. ISSN 0022-538X. doi:10.1128/JVI.02114-07. URL `https://jvi.asm.org/content/82/5/2515`.

Bhagwan Yadav, Tea Pemovska, Agnieszka Szwajda, Evgeny Kulesskiy, Mika Kontro, Riikka Karjalainen, Muntasir Mamun Majumder, Disha Malani, Astrid Murumägi, Jonathan Knowles, Kimmo Porkka, Caroline Heckman, Olli Kallioniemi, Krister Wennerberg, and Tero Aittokallio. Quantitative scoring of differential drug sensitivity for individually optimized anticancer therapies. *Scientific Reports*, 4:srep05193, 2015. ISSN 2045-2322. doi:10.1038/srep05193.

Haitao Yang, Mark Bartlam, and Zihe Rao. Drug design targeting the main protease, the achilles' heel of coronaviruses. *Current Pharmaceutical Design*, 12(35):4573–4590, 2006. ISSN 1381-6128. doi:doi:10.2174/138161206779010369. URL `https://www.ingentaconnect.com/content/ben/cpd/2006/00000012/00000035/art00005`.

Mi Yang, Jaak Simm, Chi Chung Lam, Pooya Zakeri, Gerard J. P. van Westen, Yves Moreau, and Julio Saez-Rodriguez. Author Correction: Linking drug target and pathway activation for effective therapy using multi-task learning. *Scientific Reports*, 9:7106, 2019. doi:10.1038/s41598-019-43503-0.

Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J. Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A. Smith, I. Richard Thompson, Sridhar Ramaswamy, P. Andrew Futreal, Daniel A. Haber, Michael R. Stratton, Cyril Benes, Ultan McDermott, and Mathew J. Garnett. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Research*, 41:D955—D961, 2013. ISSN 0305-1048. doi:10.1093/nar/gks1111. URL `http://academic.oup.com/nar/article/41/D1/D955/1059448/Genomics-of-Drug-Sensitivity-in-Cancer-GDSC-a`.

Chun Wei Yap. Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, 32(7):1466–1474, 2011.

Jiaxuan You, Bowen Liu, Rex Ying, Vijay S. Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *CoRR*, abs/1806.02473, 2018. URL `http://arxiv.org/abs/1806.02473`.

Jiaying You, Robert D. McLeod, and Pingzhao Hu. Predicting Drug-Target Interaction Network Using Deep Learning Model. *Computational Biology and Chemistry*, 80:90–101, 3 2019. ISSN 1476-9271. doi:10.1016/j.compbiolchem.2019.03.016.

Ran Yu, Liang Chen, Rong Lan, Rong Shen, and Peng Li. Computational screening of antagonists against the SARS-CoV-2 (COVID-19) coronavirus by molecular docking. *International Journal of Antimicrobial Agents*, 56(2):106012, 2020.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

Chun-Hui Zhang, Krasimir A. Spasov, Raquel A. Reilly, Klarissa Hollander, Elizabeth A. Stone, Joseph A. Ippolito, Maria-Elena Liosi, Maya G. Deshmukh, Julian Tirado-Rives, Shuo Zhang, Zhuobin Liang, Scott J. Miller, Farren Isaacs, Brett D. Lindenbach, Karen S. Anderson, and William L. Jorgensen. Optimization of triarylpyridinone inhibitors of the main protease of sars-cov-2 to low-nanomolar antiviral potency. *ACS Medicinal Chemistry Letters*, 12(8):1325–1332, 08 2021a. doi:10.1021/acsmedchemlett.1c00326. URL `https://doi.org/10.1021/acsmedchemlett.1c00326`.

Chun-Hui Zhang, Elizabeth A. Stone, Maya Deshmukh, Joseph A. Ippolito, Mohammad M. Ghahremanpour, Julian Tirado-Rives, Krasimir A. Spasov, Shuo Zhang, Yuka Takeo, Shalley N. Kudalkar, Zhuobin Liang, Farren Isaacs, Brett Lindenbach, Scott J. Miller, Karen S. Anderson, and William L. Jorgensen. Potent noncovalent inhibitors of the main protease of sars-cov-2 from molecular sculpting of the drug perampanel guided by free energy perturbation calculations. *ACS Central Science*, 7(3):467–475, 03 2021b. doi:10.1021/acscentsci.1c00039. URL `https://doi.org/10.1021/acscentsci.1c00039`.

Haiping Zhang, Linbu Liao, Konda Mani Saravanan, Peng Yin, and Yanjie Wei. DeepBindRG: a deep learning based method for estimating effective protein–ligand affinity. *PeerJ*, 7:e7362, 1 2019. ISSN 2167-8359. doi:10.7717/peerj.7362.

Jianming Zhang, Priscilla L Yang, and Nathanael S Gray. Targeting cancer with small molecule kinase inhibitors. *Nature reviews cancer*, 9(1):28–39, 2009.

Li Zhang, Hai-Xin Ai, Shi-Meng Li, Meng-Yuan Qi, Jian Zhao, Qi Zhao, and Hong-Sheng Liu. Virtual screening approach to identifying influenza virus neuraminidase inhibitors using molecular docking combined with machine-learning-based scoring function. *Oncotarget*, 8 (47):83142, 2017a. Publisher: Impact Journals, LLC.

Linlin Zhang, Daizong Lin, Xinyuanyuan Sun, Ute Curth, Christian Drosten, Lucie Sauerhering, Stephan Becker, Katharina Rox, and Rolf Hilgenfeld. Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α-ketoamide inhibitors. *Science*, 368(6489):409–412, 2020.

Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today*, 22(11): 1680–1685, 2017b.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Shuxing Zhang, Kamal Kumar, Xiaohui Jiang, Anders Wallqvist, and Jaques Reifman. DOVIS: An implementation for high-throughput virtual screening using AutoDock. *BMC Bioinformatics*, 9(1):1–4, 2008.

Tianyu Zhang, Liwei Zhang, Philip R O Payne, and Fuhai Li. Synergistic Drug Combination Prediction by Integrating Multi-omics Data in Deep Learning Models. 2018b.

Wei Zhao, Kirk E Hevener, Stephen W White, Richard E Lee, and James M Boyett. A statistical framework to evaluate virtual screening. *BMC bioinformatics*, 10(1):225, 2009.

Hao Zheng, Mingming Chen, Wenju Liu, Zhanlei Yang, and Shan Liang. Improving deep neural networks by using sparse dropout strategy. In *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, pages 21–26. IEEE, 2014.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

Tao Zhou, Kim-Han Thung, Xiaofeng Zhu, and Dinggang Shen. Machine Learning in Medical Imaging, 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings. *Machine learning in medical imaging. MLMI (Workshop)*, 10541:132–140, 2017. ISSN 0302-9743. doi:10.1007/978-3-319-67389-9_16.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.

Maxwell I Zimmerman, Justin R Porter, Michael D Ward, Sukrit Singh, Neha Vithani, Artur Meller, Upasana L Mallimadugula, Catherine E Kuhn, Jonathan H Borowsky, Rafal P Wiewiora, et al. Sars-cov-2 simulations go exascale to predict dramatic spike opening and cryptic pockets across the proteome. *Nature chemistry*, 13(7):651–659, 2021.

Aleksandar Zlateski, Kisuk Lee, and H Sebastian Seung. Znni: maximizing the inference throughput of 3d convolutional networks on cpus and gpus. In *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 854–865. IEEE, 2016.

James Zou, Mikael Huss, Abubakar Abid, Pejman Mohammadi, Ali Torkamani, and Amalio Telenti. A primer on deep learning in genomics. *Nature genetics*, page 1, 2018.

Quan Zou, Jiancang Zeng, Liujuan Cao, and Rongrong Ji. A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing*, 173:346–354, 2016.

Robert W Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.