PROTON TO PROTEOME: A MULTI–SCALE INVESTIGATION OF DRUG DISCOVERY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jonathan A. Fine

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Gaurav Chopra, Chair
Department of Chemistry
Dr. Lyudmila Slipchenko
Department of Chemistry
Dr. Herman O. Sintim
Department of Chemistry
Dr. Hilkka I. Kentämaa
Department of Chemistry

Approved by:

Dr. Christine A. Hrycyna
Head of the School Graduate Program

This thesis is dedicated to my mom, Dr. Lisa Fine for inspiring me with the love of science.

ACKNOWLEDGMENTS

First and foremost I would like to acknowledge my Ph.D. advisor, Dr. Gaurav Chopra and my lab colleagues who have helped me throughout my years at Purdue. Priya Prakash, Erin Kiscuk, Matthew Muhoberac, Elizabeth Thayer, Travis Lantz, Joydeb Majumder, and Asarasin Adulnirath all of whom contributed to methods described in the chapter on cell scale drug discovery. Prageeth Rajitha helped to contribute to the protein scale drug discovery and Anand Rajasekar contributed to the machine learning method which predicts functional groups using FTIR and MS spectra. Hilkka Kentämaa and her student Judy Liu contributed to the chapter on predicting functional groups using molecule–ion reactions. Krupal Jethava and Armen Beck contributed to the methods described for proton scale interactions and Wei Zhang and Jordan McGraw were instrumental in the creation of the virtual reality chapter.

PREFACE

This dissertation is divided into chapters that describe drug discovery and design at different scales of development. Below is an overview of these topics that weave these chapters into an overall narrative that takes the reader from proteome–scale drug discovery in Chapter 1 to the proton in Chapter 6.

Drug discovery and design is a relatively new field when compared to that of other physical and life sciences. For its approximately 100 year existence, this field has been guided by a dogma which can be described as 'single-target' therapeutic drug design. In this approach, a single biological target (such as a misbehaving protein or RNA) is identified to cause a given indication and a compound is developed to selectively inhibit the action of this single target. Although this dogma has been applied to create a large portion of the drugs available on the market today, around 95% of the drugs created using this approach fail to pass the clinical trails created by the US FDA. There are several causes of these failures, many of which are a direct result of the 'single-target' approach. For example, multiple biological species may be the cause of a given disorder and if one does not inhibit all the critical species, then drug resistance and other issues may arise. Therefore, alternatives to this approach are currently being investigated and new techniques are being applied to improve upon single-target drug design.

The major alternative to single-target drug design is multi-target drug design which attempts to develop a single compound that interacts with multiple biological targets. The advantage of this approach is that it relaxes the assumption that only a single target is responsible for an indication. A perceived disadvantage to this approach is that inhibiting multiple targets leads to toxicity, but all drugs interact

with multiple targets. This concept can be extended to the design of compounds where interactions with an entire proteome are considered. Unfortunately, it is difficult to design new compounds to target the correct set of targets as one needs to tune structures to be selective to multiple biological species at the same time (e.g. proteins, RNA, etc). Modern drug discovery efforts around the development of kinase inhibitors and anti–depressants have already embraced this design principal. These efforts, however, are in their infancy and additional computational tools are required to address drug design at the proteome level. Such tools are introduced in Chapter 1 and are directly applied to the repurposing of psychoactive compounds for mental health indications. This chapter serves to be the chapter on proteome scale drug discovery.

Although this tool is useful for the repurposing of existing drugs, they give little guidance for the creation of novel drugs using a multi–target paradigm. To close this gap, one must consider the purpose of a drug at the biological level where its purpose is to alter the function of a cell (as opposed to a protein in the traditional paradigm of drug discovery). Here, one must understand biology as a series of pathways where multiple proteins contribute to an overall cellular function, such as cancer growth as a result of the androgen receptor signaling pathway. Since multiple protein pathways may lead to this function, one must inhibit all of the potential proteins in this pathway to achieve a desired outcome. Unfortunately, few tools exist for the mining of the proteome, so the creation of Lemon, a tool to mine data from the protein data bank, will be introduced in Chapter 2.1. These actions are described in detail in Chapter 2 where a new methodology for targeting a specific cell line is described in Chapter 2.4. This is the first chapter to introduce the concept of machine learning, which will be explored in depth in later chapters. Additionally, a formal introduction to machine learning is given in appendix A.

While the creation of novel drugs is a noble and important goal, it is not the only goal of drug discovery at the cell–scale. Another important goal at this scale is the identification of differing cell response using analytical chemistry techniques.

In this work, there are two examples of this, one which uses a combination of RNA sequencing and BioDynamic imaging to classify a biopsy as sensitive or resistant to chemotherapy (chapter 2.3), and a second which uses tandem mass spectrometry to determine deferentially expresses protein, lipids, and metabolites.

Computational chemistry plays several important roles in traditional drug design. Some examples include virtual screening, Quantitative Structure Activity Relationships (QSAR), and small-molecule docking. All of these techniques explore chemistry at the 'protein–level'. The models presented in Chapter 2 is dependant on the ability to calculate the interactions between a small molecule and a protein. This introduces drug discovery at the protein level where the implementation and benchmarking of a docking algorithm in Chapter 3 is discussed. At this scale, one analyzes the interactions a small–molecule has in the binding pocket of a protein, something that is paramount to such a algorithm. Therefore, docking is an essential tool for studying the interactions at the protein scale.

These interactions would not be possible unless the small molecule involved is known to be pure and posses the functional groups and overall structure which leads to the proper interactions with target proteins. The analysis of these functional groups using machine learning is introduced in Chapters 4 and 5 which serve to discuss drug discovery at the small–molecule scale. Here, small–molecules are discussed in terms of how they gain function through groupings of atoms and how these groupings can be studied through analytical chemistry.

Finally, the creation of small–molecules is performed through chemical reactions which involve chemistry at the proton level. The elucidation of one such reaction is given in Chapter 6 where a novel reaction between imines and carboxylic acids is discussed using a combination of quantum mechanics and machine learning. The next Chapter (Chapter 7) introduces a virtual reality platform for visualizing small molecules in protein binding pockets and the final chapter gives an outlook for drug discovery at all scales.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

SYMBOLS

S	Entropy
P	Probability
C	Context of a chemical environment
r	radius
ℓ	A bivariate loss function
∇	Derivative of a matrix operation
$\odot$	Element–wise matrix multiplication
σ	Sigmoidal activation function

ABBREVIATIONS

ADHD	Attention deficit with hyperactivity disorder
BDI	BioDynamic Imaging
L-BFGS	Limited–memory Broyden–Fletcher–Goldfarb–Shanno algorithm
CASF	Comprehensive Analysis of Scoring Functions
CRPC	Castration Resistant Prostate Cancer
CRSD	Circadian Rhythm Sleep Disorder
DLBCL	Diffuse large B-cell lymphoma
DP	Diagnostic Product
DT	Decision Tree
FDA	United States Food and Drug Administration
GPCR	G-Protein Coupled Receptor
HCP	Hexagonal Close Packed
JLS	Jet Lag Syndrome
MF1	Molecule F1 Score
MLP	Multi-Layer Perceptron Neural Network
MPR	Molecular Perfection Rate
NIST	United States National Institute of Science and Technology
NIH	United States National Institute of Health
NPIM	1-naphthyl(1-pentyl-1h-indol-3-yl)methanone
LR	Logistic Regression
PT	Proton Transfer
RF	Random Forest
RMSD	Root Mean Squared Deviation

RLS	Restless leg syndrome
ROC	Receiver Operator Characteristic
SAD	Seasonal Affective Disorder
SIMD	sleep initiation and maintenance disorder
SVM	Support Vector Machine
SWS	substance withdrawal syndrome
TSP	Top Seed Percent
QM	Quantum Mechanics

NOMENCLATURE

Amphetamine	Derivative of 1-phenylpropan-2-amine
Canabinoid	Derivative of 6,6,9-trimethyl-3-pentylbenzo[c]chromen-1-ol
Cathinone	Derivative of (2S)-2-amino-1-phenylpropan-1-one
Phenethylamine	Derivative of 2-phenylethanamine
Tryptamine	Derivative of 2-(1H-indol-3-yl)ethanamine
MOP	2-methoxy propylene

GLOSSARY

crystal-like pose	A pose within 2.0 Å of the crystal pose
diagnostic product	The adduct formed between an analyte and a neutral reagent
feature	The input to a model which is used to predict an outcome
loss function	A bivariate function taking true and predicted outcomes
observation	A sample with measured features and outcome(s)
outcome	A measured or predicted result which one wishes to model

ABSTRACT

Fine, Jonathan A. Ph.D., Purdue University, May 2020. Proton to Proteome: a Multi–scale Investigation of Drug Discovery. Major Professor: Gaurav Chopra.

Chemical science spans multiple scales, from protons to the proteins that make up a proteome. Throughout my graduate research career, I have developed statistical and machine learning (ML) models to better understand chemistry at these different scales, including predicting molecular properties of molecules in analytical and synthetic chemistry to integrating experiments with chemo-proteomic models for drug design. Starting with the proteome, I will discuss repurposing compounds for mental health indications and visualizing the relationships between indications. Moving to the cellular level, I will introduce Lemon, a data mining framework, and the use of ML to classify cancer resistance, use existing methods developed for the negative binomial distribution to develop a new bioinformatics methodology to find biomarkers of cellular response using data collected by mass spectrometry, and use ML to select potent, non-toxic, small molecules for the treatment of castration resistant prostate cancer. For the protein scale, I will introduce CANDOCK, a docking method to rapidly and accurately dock small molecules. Next, I will showcase a deep learning model to determine small-molecule functional groups using FTIR and MS spectra, followed by a similar approach used to identify if a small molecule will undergo a diagnostic reaction using mass spectrometry using a chemically interpretable graph-based ML method. Finally, I will examine chemistry at the proton level and how quantum mechanics combined with ML can be used to elucidate chemical reactions. In summary, ML models have the potential to accelerate several aspects of drug discovery including discovery, process, and analytical chemistry.

1. PROTEOME SCALE DRUG DISCOVERY

This chapter is available as

Fine, J., Lackner, R., Samudrala, R., Chopra G. Computational chemoproteomics to understand the role of selected psychoactives in treating mental health indications. Sci Rep **9**, 13155 (2019).
https://doi.org/10.1038/s41598-019-49515-0

1.1 Abstract

We have developed the Computational Analysis of Novel Drug Opportunities (CANDO) platform to infer homology of drug behavior at a proteomic level by constructing and analyzing structural compound-proteome interaction signatures of 3,733 compounds with 48,278 proteins in a shotgun manner. We applied the CANDO platform to predict putative therapeutic properties of 428 psychoactive compounds that belong to the phenylethylamine, tryptamine, and cannabinoid chemical classes for treating mental health indications. Our findings indicate that these 428 psychoactives are among the top-ranked predictions for a significant fraction of mental health indications, demonstrating a significant preference for treating such indications over non–mental health indications, relative to randomized controls. Also, we analyzed the use of specific tryptamines for the treatment of sleeping disorders, bupropion for substance abuse disorders, and cannabinoids for epilepsy. Our innovative use of the

CANDO platform may guide the identification and development of novel therapies for mental health indications and provide an understanding of their causal basis on a detailed mechanistic level. These predictions can be used to provide new leads for preclinical drug development for mental health and other neurological disorders.

1.2 Introduction

Drug discovery traditionally revolves around single biological targets and focuses on a limited set of relationships between a protein target and small molecules of interest. The goal of this approach is to change the biological function of a protein responsible for pathogenesis and subsequently determine the toxicity and side effect profile of a compound to make it a suitable clinical candidate. The expected result of this approach is a compound that modulates the single protein that it targets. Although this traditional approach has been successfully applied to develop the majority of approved drugs, it has been questioned in recent years as the number of newly approved drugs continues to decrease (currently down to 30 according to fda.gov). Additionally, many new drugs are analogs to already known drugs or reformulated to improve efficacy and filed as new patents. According to the Tufts Center for the Study of Drug Development (csdd.tufts.edu), the average cost to bring a new drug to market can be as large as $ 2.6 billion. Therefore, there exists a shortage of novel drug development because the current approach is both time and cost prohibitive [1–4].

One methodology to combat the rising cost and time commitment of novel drug development is to re–purpose already approved drugs that are known to have few deleterious side effects [3, 5–11]. Competitiveness in the pharmaceutical industry hinders the systematic exploration of potential repurposing opportunities, but computational approaches enable a workaround. Using computational multi–target docking with dynamics, we developed a drug repurposing approach for malaria and have since

validated our models numerous times experimentally [3, 8, 9, 12–17]. To expand the applicability of our work, we have developed a shotgun approach to evaluate all potential drug repurposing opportunities simultaneously by evaluating the relationships of compounds with entire proteomes (chemoproteome) in an indication–specific manner [8, 14]. Here, we describe the application of our platform to identify possible therapeutic uses of phenethylamines, tryptamines, and cannabinoids in treating mental health indications.

1.2.1 Leveraging computational chemoproteomics for drug discovery

Natural products have a profound impact on drug discovery. Many of these products come from plant sources, [18–20] where 60% of drugs approved by the FDA circa the 1990s came from these sources [21]. While this percentage has decreased to about 40% in recent years, it is clear that natural products have an important impact on drug discovery [22]. Since plants, animals, and other organisms have evolved together, we hypothesize that multiple modes of action are responsible for a small molecule to become a drug. We have thus developed a platform which relies on a "signature of interactions" (a row of binary or real numbers) to represent the interactions of compounds with a set of protein structures that are selected to represent the known structural universe. Our hypothesis requires that similar chemoproteome signatures indicate similar functional behaviors while non–similar signatures (or regions thereof) indicate off and anti–target (side) effects as these signatures infer proteomic homology of compound or drug behavior. We can use these chemoproteomic signatures to rank how well a compound can be repurposed for given indication and provide a set of protein interactions responsible for this ranking to obtain an understanding of drug mechanisms at the level of atomic interactions.

1.2.2 CANDO: A shotgun computational chemoproteomics platform for drug repurposing and discovery

Biologically active molecules, such as proteins and drugs, do not function in isolation. The absorption, dispersion, metabolism, and excretion (ADME) and effectiveness of a drug are dependent on the interactions of the drug with a system of proteins expressed at different sites in an organism. The Computational Analysis of Novel Drug Opportunities (CANDO) platform works at the proteomic level by leveraging the interaction signature of a compound to all proteins in a generic structural library. It compares the signatures of candidate compounds/drugs to those approved for particular indications to make drug repurposing predictions in a shotgun manner (here meaning an all versus all compound–proteome signature comparison).

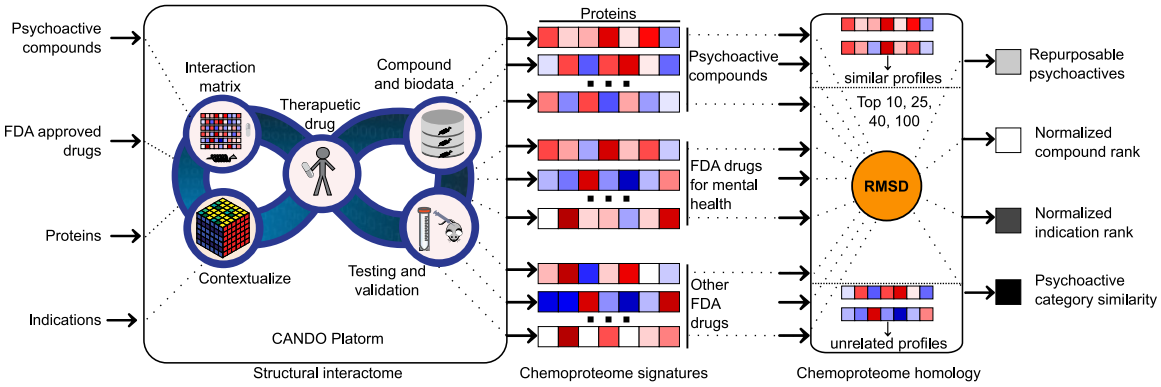


Figure 1.1. Schematic of computational chemoproteomics pipeline to identify psychoactives for mental–health indications using the CANDO platform.

The first version of the CANDO platform (CANDO v1) shown in Fig. 1.1 predicts interactions between 3,733 FDA approved drugs and a variety of other human ingestible compounds (including supplements and illegal substances) and 48,278 protein structures from multiple species (46,784 of which are used in this study and this pro-

tein list is provided in the GitHub data repository) either taken from the Protein Data Bank (PDB) [23] or representing high confidence homology models [24] constructed using protein structure prediction methods described previously [15, 25]. Specifically, the proteins structures include solved and modeled proteins obtained from eukaryotic, prokaryotic, archaea and viral organismal proteomes, including 14,595 human proteins (8,841 of these are high–confidence models), a set of 24,958 non–redundant solved protein structures in the PDB, in addition to the remaining solved and modeled structures from *Mycobacterium tuberculosis*, *Pseudomonas aeruginosa*, viral proteomes, and so on. We consider different conformations of protein structures by separately including multiple domains (chains) and isoforms of proteins for calculating all compound proteins interactions. As an example, for the experimental structures considered for the human proteome, we use a mapping between PDB chains and UniProtKB/SwissProt codes [26] in the human proteome. We also treat all such protein–compound interactions equally as proteins from different biological classes affect benchmarking accuracy results to predict putative repurposable drugs for diseases [15]. We employ our bioinformatic docking approach to construct a 3,733 by 46,784 compound–protein interaction matrix (see Compound–Proteome Interaction Signature section [15]) that is analyzed to determine similarity in drug behavior [15, 25]. No special methods were used for different protein classes (e.g., kinases and GPCRs) so that scores of two proteins from different classes could be compared directly. To generate a pose we used a hierarchical fragment–based docking with dynamics algorithm [27] using knowledge–based potentials [28] as done previously for the Ebola proteome [29]. We have previously shown that all–atom dynamics is necessary for accurate prediction of binding energies [30] and demonstrated all–atom knowledge–based force fields are more accurate than physics–based approaches for both protein structure prediction and docking [17, 30–35]. Furthermore, we have shown that multi–targeted docking

with dynamics leads to improved hit rates for finding inhibitors of pathogens relative to conventional approaches [7,8]. It should be noted that the interaction score stored in this matrix does not represent whether a given target will be inhibited or activated, only that the compound and target interact. As a result, the CANDO platform can be used for both inhibitors and agonists with the caveat that the predicted effect of a compound may be unknown until verified experimentally. For example, CANDO could predict cocaine for the treatment of cocaine–related disorders. Therefore, special care needs to be used when examining these predictions since dose selection is not part of the current model.

Once the interaction matrix is constructed, our methods compare the compound–proteome interaction signatures where the similarity of two signatures can be calculated using various metrics as simple as root mean squared deviations (RMSD) to sophisticated graph theory based comparisons that can take underlying protein–protein interactions (compiled from public sources [24, 36–38]) into account. Similarities between (regions of) interaction signatures indicate a relationship in functional behavior. However, the differences between two signatures are difficult to understand without further knowledge as it may indicate a more potent drug, a possible side effect, or no effect whatsoever. In addition to predicting a ranked list of putative drugs that are most likely to function similarly to other drugs approved for a particular indication, the signature comparison and ranking helps to analyze compound behavior in biologically relevant pathways [36–39]. Our CANDO platform is successful for prospectively validating putative leads for several indications [15, 25, 29].

1.2.3 Mental health indications and interventions

A large number of diseases and disorders have mental health implications as cataloged by the American Psychiatric Association (APA) [40].These indications affect

people in all age groups, social classes, and races [41–45]. The treatments for these indications mostly consist of small molecule therapeutics, varying individually for specific diseases, disorders, or conditions. According to a report published by the World Health Organization in 2011 [46], the number of United States (US) citizens taking medication to treat mental health has increased to over two million US citizens since 2001. Anxiety disorders make up the largest category of mental illness in the US affecting a total of 42 million people. The second largest category is major depression disorder affecting 14.8 million US citizens on any given day. Approximately 2.4 million US citizens have schizophrenia where no effective treatment or cure is currently available as schizophrenia medication typically results in metabolic issues leading to weight gain and type 2 diabetes [47]. Collectively mental health indications/disorders cost the US economy $192.3 billion each year and result in high morbidity, with suicide being the tenth largest cause of death [48, 49]. Unfortunately, adolescents are susceptible to depression and suicide, and the effectiveness of antidepressants for these individuals remains uncertain [50].

1.2.4 Human use of psychoactive substances

We define psychoactives as compounds that cross the blood–brain barrier, target proteins expressed in the brain as their primary modes of action, and thereby perturb human mental states. Although proteins expressed in the brain are paramount for the prediction of compounds as potential therapies for mental health disorders, synergistic effects may occur due to interactions in the periphery. For example, it has been shown that the gut microbiome plays an important role in the central nervous system [51] and multiple links between the peripheral mechanisms and depression have been found previously [52, 53]. We have also benchmarked CANDO to show that best drug repurposing accuracies are obtained when all protein structures are

used for interaction signature comparisons to determine compound similarity, suggesting the role of multiple networks working together in biology to achieve a certain phenotype/function, instead of specific proteins as used traditionally for drug discovery [15, 25]. This approach makes CANDO different than other methods that are focused towards single target inhibitor discovery vs drug discovery. Therefore, we believe that the study of proteome–wide interaction signature for repurposing psychoactive compounds is suitable in the context of mental health indications.

Since the time of the earliest records, humans have been ingesting psychoactive substances for religious and spiritual purposes (for example, dimethyltryptamine in Ayahuasca, mescaline in Peyote), for medicinal purposes (opium), and for recreation (caffeine, nicotine, alcohol) [54]. The vast majority of pyschoactives are considered taboo for a variety of reasons and, with few exceptions, are not investigated for potential medicinal properties. In this study, we focus on the phenylethylamine and tryptamine classes of psychoactives described by Alexander Shulgin [55, 56] as well as additional cannabinoids.

Due to recent changes in legislation, a few of these compounds are available as approved drugs in some jurisdictions (for example amphetamine for diet control and attention deficit hyperactive disorder, and tetrahydrocannabinol for anxiety). The action of these compounds is thought to affect human physiology by their structural similarity/mimicry to neurotransmitters (for example, psilocybin and lysergic acid diethylamide both mimic the compound serotonin). There is an increasing amount of evidence for cannabinoids having the ability to treat epilepsy and epilepsy–related indications [57, 58], but its legal status is still diffuse as cannabinoids remain classified as Schedule I by the United States Federal Government (a classification possessing no medicinal use). Similarly, psilocybin and ketamine have been shown to treat depression via a mechanism not targeted by current antidepressants [59, 60].

These examples are the tip of a proverbial iceberg, and recent reinvestigations into the clinical relevance of illicit psychoactive compounds suggest further investigation into the potential of these compounds in treating mental health indications [61]. This clear disconnect between current research and current legislation warrants a more comprehensive investigation for the use of these psychoactive compounds for medicinal purposes but in vitro and in vivo verification is currently difficult given their scheduling status. The CANDO shotgun drug discovery and repurposing platform is therefore uniquely suited to conduct such an investigation to make a case for experimental verification.

While other classifications of psychoactives could be utilized (for example, all compounds known to cross the blood–brain barrier), our goal in this study was to see if any of the selected psychoactive compounds, primarily known without any therapeutic utility, are predicted to treat mental health indications. Our work also demonstrates the more general utility of the CANDO platform in assessing the effect of drug classes on this specific class of indications.

1.2.5 Analyzing the role of psychoactives in mental health indications using CANDO

Most of our selected psychoactive compounds are illegal to synthesize and thus difficult to study in vitro (much less in vivo). Cannabinoids are in the process of being legalized for medicinal uses in some jurisdictions, and this serves as a justification for studying these drugs further. The cause of many mental health indications is not characterized by one protein, but by several proteins in several different categories [62–66]. Thus, the traditional high throughput screening methodology of testing one compound against one protein is not a suitable approach for mental health drug discovery. The CANDO platform allows for evaluation of all selected psychoac-

tives across a large library of protein structures, providing a logical and reasonable method to develop leads for medications that may be suitable for treating mental health indications. Our goal here is to study, analyze, and characterize these psychoactive compounds using the CANDO platform so that the potential medicinal properties of these compounds can be assessed and evaluated in further bench and clinical studies. The outcomes for this study are not necessarily to predict mental health therapies but rather to generate hypotheses if the predicted psychoactives serve as the most promising leads for different mental health indications based on similar chemoproteomics perspective.

1.3 Results

We describe our results based on two approaches of examining the relationships between the selected psychoactives and mental health indications using the top–ranked predictions by the CANDO platform. An example of these predictions is given in Table 1.1, where we are careful to list potential issues with the predicted psychoactive. At the outset, we examined the distributions of percentages of psychoactive compounds (relative to total compounds) in the top–ranked predictions for mental health indications. Conversely, we can compare the distributions of percentages of mental health indications selected by the psychoactive compounds in the top–ranked predictions. We further analyze the latter distributions broken down by psychoactive classes and the distributions of mental health indications. We conclude with three case studies illustrating the application and utility of the CANDO platform in discovering psychoactive therapeutics to treat mental health indications. We again caution that applying these predictions for the development of new therapeutics must be done judiciously.

Table 1.1.
Predictions which represent hypotheses of novel putative therapeutic leads for various indications.

Psychoactive	Prediction
3,4–dimethylmethcathinone	Anxiety Disorders (known for abuse)
3,4–dimethylmethcathinone	Depressive Disorder, Major (known for abuse)
dextromethorphan	ADHD (OTC antitussive)
dexfenfluramine	Autistic Disorder(known cardiac issues)
3–fluoroamphetamine	Bipolar Disorder
2–fluoroamphetamine	Cataplexy
metamfepramone	Delirium
bupropion	Depressive Disorder (depression treatment)
metamfepramone	Tourette Syndrome
ergoline	Erectile Dysfunction (migraine treatment)
α–pyrrolidinopentiophenone	Learning Disorders (Stimulant)
2–fluoroamphetamine	Narcolepsy
isopropylamphetamine	Obessive–compulsive disorder
3–fluoroamphetamine	Personality Disorders
pyrovalerone	Phobic Disorders (US Schedule V drug)
dextromethorphan	Psychotic Disorders (OTC antitussive)
3,4–dimethylmethcathinone	Restless Legs Syndrome
pyrovalerone	Schizophrenia (US Schedule V drug)
pyrovalerone	PTSD (US Schedule V drug)
α–pyrrolidinope	Tobacco Use Disorder (US Schedule V drug)
isopropylamphetamine	Panic Disorder
3,4–dimethylmethcathinone	Cocaine–Related Disorder (known for abuse)
3–fluoroamphetamine	Binge–Eating Disorder

1.3.1 Putative psychoactives for mental health indications

The results showing the distributions of percentages of psychoactives for mental health indications are given in Fig. 1.2. This figure shows that the difference in the random and non–random distributions. Since these distributions are statistically different, we conclude the selected psychoactive compounds are better at treating mental health indications on average than non–psychoactive compounds selected at random. As the number of compounds considered increases, the normal and randomized distributions become more alike. This result is expected as there are a larger number of

non–psychoactive compounds than the selected psychoactive ones and, therefore, the addition of a new compound is more likely to be non– psychoactive than psychoactive. Therefore, as the number of compounds in the result list increases the percentage of psychoactives predicted for any indication will decrease (see Fig. 1.2).

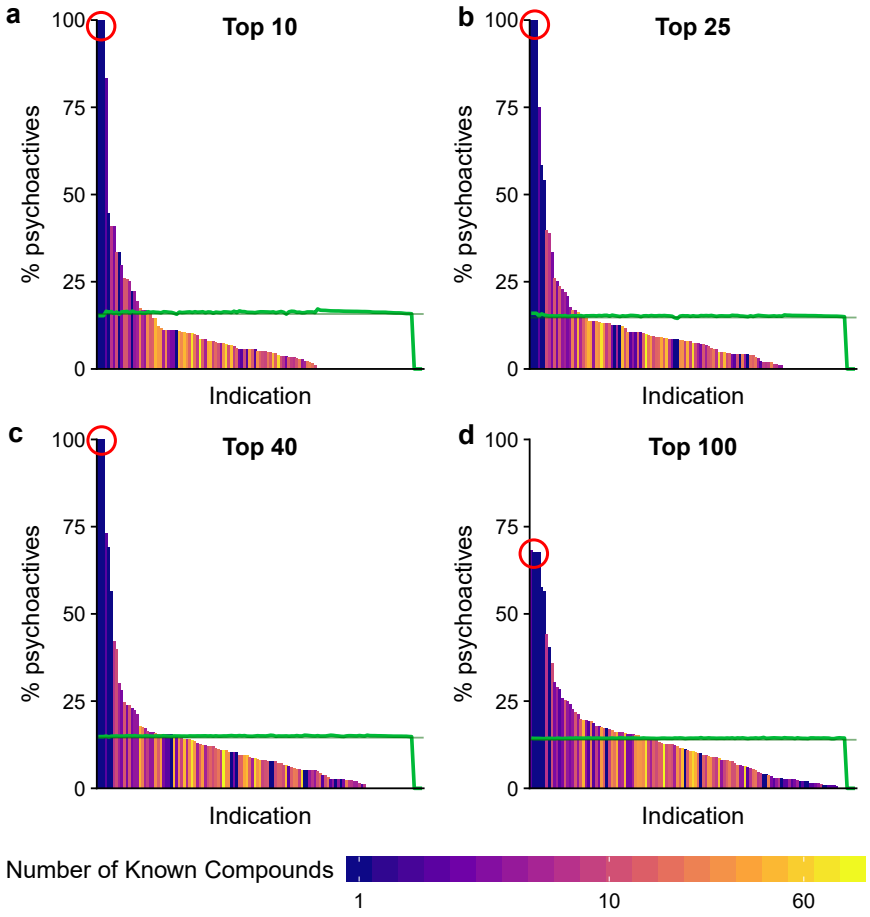


Figure 1.2. Normalized indication rank for all indications. The green line shows the results of randomizing the predicted compounds and the straight–line segment indicates the mean of the randomized distribution. The circled indications are SAD, CRSD, and JLS.

1.3.2 Selection of mental health indications by selected psychoactives

The distributions for the selection of mental health indications by selected psychoactives relative to all indications are shown in Fig. 1.3. The greater the percentage of mental health indications, the more selective the psychoactive. Furthermore, the indications selected by psychoactives using the CANDO platform yield a high percentage of mental health indications relative to random controls, illustrating that these psychoactives are more likely than non–psychoactives to be effective at treating mental health indications.

1.3.3 Comparison of randomized compound and indication distributions

The two randomized distributions in Fig. 1.3 (shown in green and blue) are distinct. The distribution representing randomized compounds is less uniform and has a larger average percentage (p–value less than 2×10^{-16} from a one–tailed student t–test for all four plots) than the randomized indication distribution. These data show that a single drug is more likely to treat multiple indications than a single indication is to be treated by multiple drugs. This has been shown previously by the ability to repurpose drugs [67–69] and is an important feature of the CANDO platform. The ability to repurpose previously approved compounds is increasingly important [70]. This result highlights the utility of the CANDO platform for drug repurposing.

1.3.4 Comparison of different psychoactive classes

Fig. 1.4 differentiates the psychoactives by compound class: amphetamines, cannabinoids, cathinones, phenethylamines, and tryptamines. Given the proteomic signature comparison approach used by CANDO to makes these predictions, this indicates that psychoactives from one category are predicted to bind to the same proteins

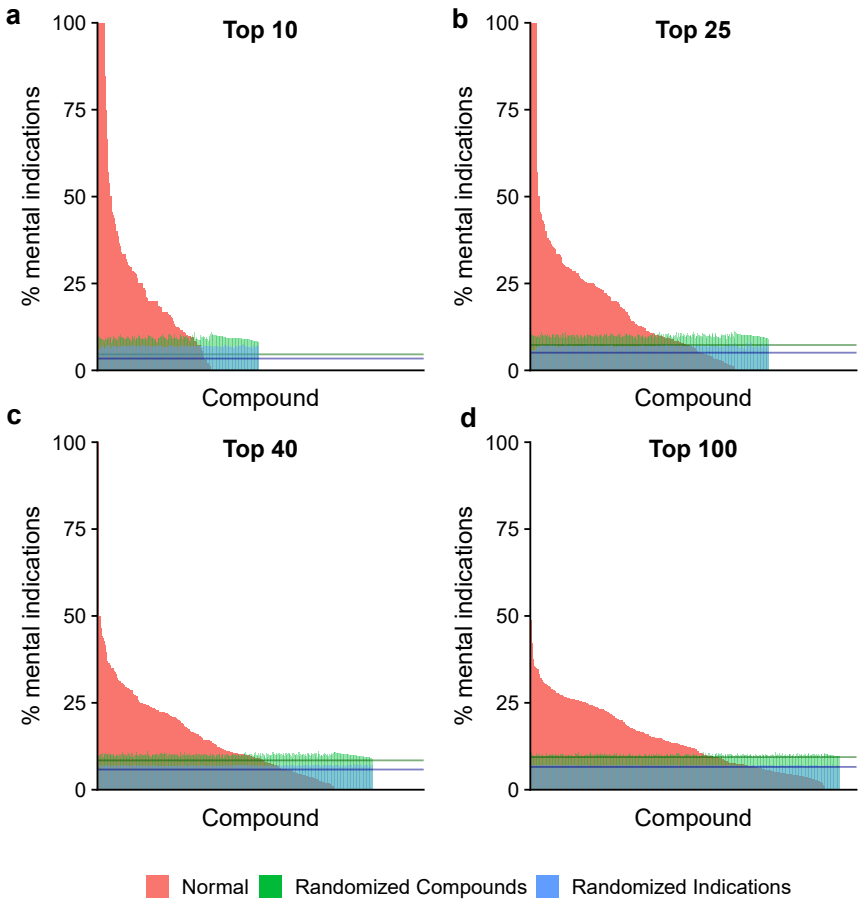


Figure 1.3. Normalized indication rank for all indications. The green line shows the results of randomizing the predicted compounds and the straight–line segment indicates the mean of the randomized distribution. The circled indications are SAD, CRSD, and JLS.

as psychoactives from a different category, resulting in a constant percent occurrence for all compounds predicted to treat an indication. Thus, the Top10 rankings provide the most specificity for analyzing the effect of a psychoactive class on selecting mental health indications. These figures and tables illustrate that the classification of a compound has an impact on which indications it is predicted to treat. Therefore, we will continue the discussion based on psychoactive compound–classes.

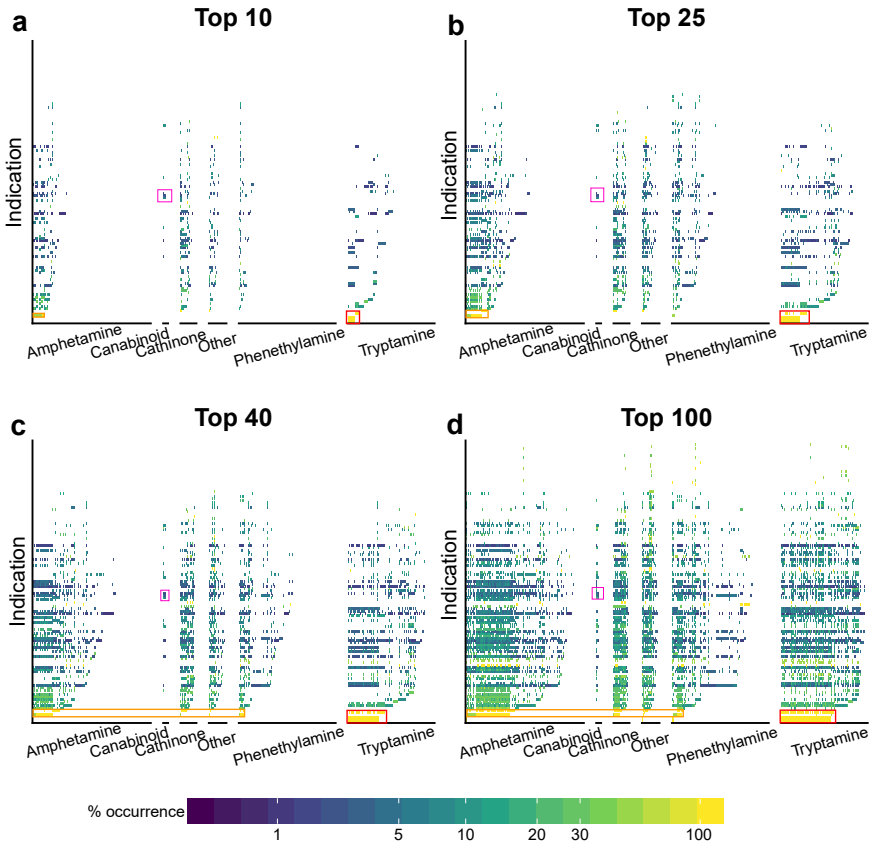


Figure 1.4. Distribution of mental health indications treated by different classes of psychoactives. As the number of predictions increases, the distribution of indications per class becomes increasingly similar. Indications of interest are shown with the following boxes: red for Seasonal Affective Disorder, Jet Lag Syndrome, sleep disorders and Broca Aphasia; orange is Binge–Eating Disorder, Narcolepsy, and Anorexia Nervosa; purple is Heroin Dependence, Substance–Related Disorders, and Epilepsy.

1.3.5 Relationships between mental health indications

Our predictions for indication–indication associations are shown in Fig. 1.5. Interestingly, some indication relationships have been verified clinically. These include: Epilepsy with Seizure, Cocaine–related disorders with depression, [71] Seizures with Substance Withdrawal Syndrome, [72] Depression with Anxiety, [73] and possibly relating binge–eating and personality disorder [74]. The ability of our repurposing

platform to reproduce known indication relationships suggests that our chemoproteomic signatures can capture key biological interactions. In addition, the number of overlapping psychoactive compound predictions strongly relate multiple mental health indications (width of the chords in Fig. 1.5). These psychoactives interact with multiple proteins (similar chemo–proteome signatures) suggesting common biochemical pathways. We are confident that our method may be useful to discover new disease pathways relating these indications. Identifying and validating these new pathways are beyond the scope of this work.

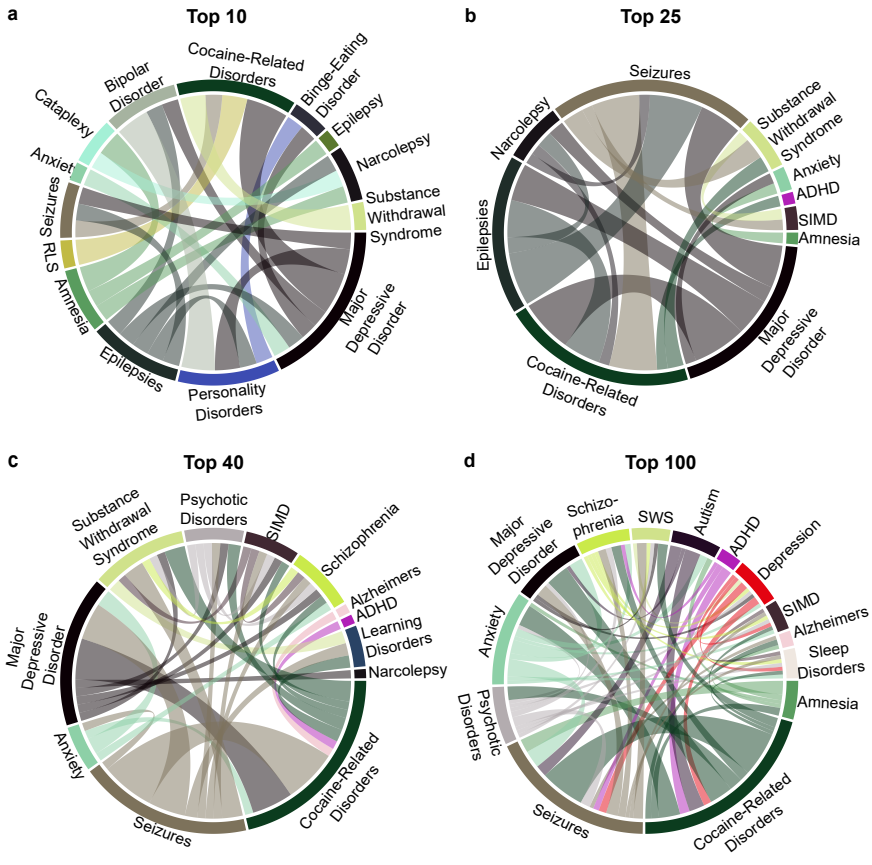


Figure 1.5. Indication–Indication association counts plotted as chord diagrams showing predicted relationships between the indications. The width of the chord is proportional to the number of predicted psychoactives relating two indications. Known relationships include Epilepsy with Seizure, Depression with Cocaine–related disorders.

1.4 Discussion

The Top10, Top25, and Top40 predictions in Fig. 1.2 for three mental health indications, Seasonal Affective Disorder, Circadian Rhythm Sleep Disorders, and Jet Lag Syndrome, consist only of psychoactives belonging to the tryptamine class (indication rank of 100%). The only compound known to treat all these indications is melatonin (also a tryptamine) [75–77], indicating that its proteomic interaction signature is most similar to the interaction signatures for these predicted psychoactives. This result demonstrates that the proteomic shotgun drug repurposing approach adopted by the CANDO platform makes sensible predictions of related compounds based on their similarly of interaction signature with all proteins, compared to traditional single target approaches. Studies by an Israeli pharmaceutical company give experimental evidence demonstrating that some of these tryptamine psychoactives are indeed likely to treat the aforementioned three indications [78]. These studies provide corroborative evidence for the efficacy of the CANDO platform and highlight its potential of finding new drugs for treating any indication that has at least one approved drug.

1.4.1 Comparison between predicted drugs and the literature

The remainder of this discussion will be used to highlight case studies which are verified in the literature. For a complete list of psychoactive predictions, please see appendix B.

The indication with the largest number of high ranking psychoactives in the top–ranked predictions is cocaine–related disorders belonging to the cathinone class of stimulants, a summary of which is given in Table 1.2. The similarity between the effects of cathinone and cocaine on behavior has been previously established as part of a similar pathway [79]. We are aware that some of these predictions are unlikely

to have any potential for the development of new therapeutics for cocaine–related disorders due to their associated toxicity [80, 81]. A cathinone of interest is the anti–depressant bupropion, which is well known for promoting smoking cessation and has also been proposed for the treatment of methamphetamine and cocaine substance abuse disorders [82, 83]. These findings and related uses further showcase the ability of CANDO platform to accurately associate compounds/drugs and indications. While this example is successful in showcasing CANDO's ability to find the relationship between compounds and mental health disorders, one needs to be cautious as these predictions may mimic cocaine and lead to adverse reactions depending on the dose. For example, Bupropion is perceived as a stimulant to those with a history of cocaine use [84, 85]. Further, the effects of dextromethorphan may be due to its stimulant properties [86]. However, in some cases we can obtain therapeutic benefit from potentially problematic compounds, example given methadone is an approved treatment for opioid abuse, but is known to have several opioid–related effects when given in high enough dosages [87].

Table 1.2.
The top predictions for these indications belong to the cathinone class.

Psychoactive	**Known effects and legal status**
flephedrone	Toxicity not well established
buphedrone	Illegal for human consumption
ethcathinone	Illegal due to similarities to mephedrone
mephedrone	High potential for abuse
methcathinone	Causes euphoria. Highly addictive
3,4–dimethylmethcathinone	Stimulant with a high potential for abuse
bupropion	Prescription anti–depressant
dextromethorphan	Over the counter antitussive
alpha–pyrrolidinopropiophenone	Stimulant
NPIM	Serious source of addiction
n,n–dibutyltryptamine	Hallucinogenic research chemical
isopropylamphetamine	Stimulant

The highest–ranking phenethylamine predicted to treat cocaine–related disorders is the antitussive drug, dextromethorphan. This compound, generally available over the counter, is known for its hallucinogenic side effects at high doses, which is reflected both in the predictions by CANDO and is reinforced in the literature [88–92]. The use of the CANDO platform for making predictions to treat specific mental health indications is strengthened by the accurate identification of bupropion and dextromethorphan (both selected psychoactives) in treating cocaine–related disorders.

The two psychoactive cannabinoids, tetrahydrocannabinol and cannabinol are predicted to treat Epilepsy and Absence Epilepsy by the CANDO platform, and cannabinol is also predicted to treat Status Epilepticus. While the cannabinoids are not the highest ranked compounds relative to other psychoactives for these indications, our findings are validated by recently published studies for the use of cannabinoids to treat epilepsy–related indications [57, 58]. The non–psychoactive cannabinoid (cannabidiol) is not predicted to treat any epilepsy–related indications, leading to an intriguing hypothesis concerning the likelihood of a cannabinoid treating epilepsy corresponding to its psychoactivity. However, given the limited data available, further study is warranted to verify this hypothesis. Our work illustrates the recovery of known corroborative associations between cannabinoids and epilepsy but also demonstrates how predictions made by the CANDO platform can be used to develop hypotheses on the biology of diseases for experimental investigation.

1.5 Methods

Here, we describe the approach used to analyze the data generated by this platform to characterize the role of the selected psychoactives in mental health indications.

1.5.1 Selection of phenethylamines, tryptamines, and cannabinoids

We collected a total of 428 compounds to be investigated using CANDO and categorized them into 291 phenethylamines and 109 tryptamines described by Alexander Shulgin [55,56], and 6 cannabinoids (cannabinol, cannabidiol, and tetrahydrocannabinol) using a subgraph based search methodology based on the structure of the parent molecule. An additional 22 compounds are not strictly classified as phenethylamines but have structural similarity to the phenethylamine class are included as unclassified. We further subdivided the 291 phenethylamine compounds into 149 amphetamines and 20 cathinones, the remaining 122 phenethylamines are simply referred to as phenethylamines. The CANDO v1 compound library includes these 428 psychoactives and their proteomic interactions signatures to repurpose psychoactives for indications/diseases [9]. Most of these psychoactives are classified as Schedule I substances by the United States Drug Enforcement Agency, indicating they have no known medicinal use, no accepted standards for safety, or have a high potential for abuse. Thus, when such a substance is discussed, the potential pitfalls are presented along with that substance. We selected this set of compounds as almost all of them are known to affect mental physiology upon ingestion [55,56,93]. A notable exception in the compounds evaluated is cannabidiol which is not strictly psychoactive [93] but is structurally similar to other cannabinoids and therefore warrants an investigation into its potential therapeutic value.

The CANDO v1 compound–proteome interaction signature (see Supporting Methods) includes all associations of treatment and side effects caused for each compound via the proteomic signature as this is composed of all target, anti–target, and off–targets proteins for each indication/disease. The compound proteomic interaction signature similarity yields therapeutic predictions by considering similarity to known drug signatures for each disease. It should be noted that this methodology can also

match a psychoactive to a compound known to worsen a given indication in addition to predicting a compound known to ameliorate the same indication. Therefore, the set of compounds that were used as therapy for a given indication did not include any of the aforementioned psychoactive compounds given that the nature of these compounds as treatments is still controversial. As a result, the ability of the platform to predict a psychoactive from another psychoactive compound–proteome signature is not investigated in this work. Most importantly, all predictions are made based on similarity to an approved non–psychoactive drug for a mental health indication, without any knowledge of therapeutic target associations for making predictions for psychoactive compounds. Therefore, no association between an indication and a protein target is used to weight the similarity between two compounds. For example, the interaction score of a psychoactive and the dopamine receptor is not given a special weight for Schizophrenia.

1.5.2 Selection of mental health indications

The Medical Subject Headings (MeSH) vocabulary is used to specify the diseases, disorders, and conditions that are classified as mental health indications. The United States National Laboratory of Medicine division of the National Institutes of Health (www.nlm.nih.gov) includes the latest version of the MeSH database. It should be noted that this database is compiled at the clinical level and does not consider the underlying biology leading to a specific indication. Therefore, some spurious and non–traditional indications may be included as mental health indications. Since a biological mechanism study is beyond the scope of this paper, we used all the indications suggested by MeSH.

The MeSH database is divided into tree structures with a specific tree (F03) denoted for Mental Disorders. The specific branches of the Mental Disorder Tree used in

this study are Anxiety Disorders (F03.080), Dissociative Disorders (F03.300), Feeding and Eating Disorders (F03.400), Neurocognitive Disorders (F03.615), Somatoform Disorders (F03.875), Conduct Disorders (F03.250), Neurodevelopmental Disorders (F03.625), Mood Disorders (F03.600), Neurotic Disorders (F03.650), Personality Disorders (F03.675), Schizophrenia Spectrum Disorders (F03.700), Sleep–Wake Disorders (F03.870), and Substance–Related Disorders (F03.900). All the indications listed in these branches were used along with Dyspareunia, Erectile Dysfunction, Paraphilias, Fetishism, and Paedophilia from the Sexual Dysfunctions (F03.835) branch yielding a total of 108 mental health indications that are analyzed in this work. A separate MeSH identification paradigm was done for epilepsy–related indications as these indications are placed in a separate MeSH tree because they are neurological disorders, not psychiatric disorders. The MeSH tree evaluated for epilepsy is C10.228.140.490 which includes Drug–Resistant Epilepsy, Myoclonic Epilepsies, Partial Epilepsies, Benign Neonatal Epilepsy, Generalized Epilepsy, Post–Traumatic Epilepsy, Reflex Epilepsy, Landau–Kleffner Syndrome, Lennox–Gastaut Syndrome, Seizures, and Febrile Seizures. A total of 29 additional epilepsy–related indications are presented in this work.

1.5.3 Calculation

Ranking the importance of predicted psychoactives for mental health indications

We generated Top10, Top25, Top40, and Top100 ranked compound lists for all indications (mental health related and otherwise) using the CANDO v1 platform for each indication and counted the number of times a compound prediction is present in

each of the ranked lists. A compound may be predicted to treat an indication several times if there are numerous known drugs for that indication.

For example, 65 known drugs are used clinically for schizophrenia and are included in the CANDO platform. Therefore, a set of 65 chemoproteomic signature similarities are used to predict an uncharacterized compound for schizophrenia. It is possible that the same uncharacterized compound may be predicted at most 65 times for schizophrenia. The number of times such a compound is predicted for a given indication is termed as the 'consensus count,' which is normalized as the percent occurrence. We compute percent occurrence as the ratio of consensus count to the maximum number of times a compound could be predicted for an indication using signature similarity (for example the number of known treatments for the indication).

We hypothesized that the higher the number of times a compound is predicted to treat a given indication, the greater the confidence in the prediction made because different drugs treat indications due to different biological pathways on the proteomic level. The combination of proteomic similarity implicitly includes a combination of all pathways to yield efficacy and is denoted by the frequency of compounds predicted for each indication. To investigate the general role of psychoactives for mental health, we took the frequency of psychoactive compounds predicted as putative drugs for each indication as a percentage of all compounds predicted for the given indication. The ratio of psychoactive to all compounds for a given indication is referred to as the normalized indication rank and quantifies the overall performance of psychoactive compounds versus non–psychoactive for a given indication.

A similar procedure is used to measure the propensity of a compound to be predicted for mental health indications and is used to ask the question: how many times an indication is (or percentage of mental health and non–mental health indications) listed either as a prediction for treatment by each psychoactive compound and is

referred to as the normalized compound rank. This measure allows us to express the preference of a given compound towards mental health indications. To illustrate metrics of normalized compound rank, and normalized indication rank, we consider a simple example where a psychoactive compound ergoline and a non–psychoactive compound aspirin are only predicted for the mental health indication Pica (an eating disorder) and Stomach pain (non–mental health indication). Now consider ergoline that is predicted seven times for Pica, and three times for Stomach pain based on the similarity of chemo–proteome analyzis while the non–psychoactive drug aspirin is predicted to treat Pica four times and Stomach pain six times. The normalized compound rank for this simple example will be 70% $[100 * 7/(7 + 3)]$ for ergoline and 40% $[100 * 4/(4 + 6)]$ for aspirin. Using the above example, we can also determine the normalized indication rank for Pica and Stomach pain. Since seven psychoactives and four non–psychoactive compounds were predicted for Pica, the normalized indication rank is 64% $[100 * 7/(4 + 7)]$ for Pica and similar calculation yields an indication rank of 33% for Stomach pain. Together these metrics suggest the importance of psychoactive compounds and their preference for mental health compared to non–mental health indications.

Computational randomized controls

To further ensure that our results were not arrived at by chance, the order of the predicted compounds is randomized, and the above procedures repeated. All compounds predicted to treat an indication are randomized regardless of whether the indication is categorized as a mental health indication. Thus, a compound not predicted initially to treat a mental health indication may, due to random chance, be predicted to treat a mental health indication in the randomized data set. If a random compound replaces a predicted compound multiple times for a single indication, the

random compound replaces the original (non–random) compound for every prediction. This randomization process is repeated 1000 times, and the compound and indication ranks for all the randomized searches are averaged.

A second random control is performed in addition to the one described above where the indications (mental health or otherwise) are randomly rearranged. Thus, a non–mental health indication may be classified as a mental health indication by chance (and vice versa). This procedure provides a second control that allows us to assess whether selected psychoactives are more likely to be predicted for mental health indications than non–mental health indications.

Determination of relationships between mental health indications

We relate two mental health indications when at least two different psychoactive compounds are predicted for both indications. The frequency of prediction for common psychoactive compound predictions is termed as **indication–indication association counts**. Next, to strongly relate the two indications, we also calculated the **consensus count** for all psychoactives predicted for each mental health indication. Note that a predicted psychoactive could have a different consensus count for each indication. For example, 1–naphthyl(1–pentyl–1H–indole–3–yl) methanone (NPIM) is predicted 10 times for seizures and 8 times for sleep initiation and maintenance disorders (SIMD). Therefore, the consensus count of NPIM for seizures is 10 and 8 for SIMD. Another compound, 2–(5–methoxy–2–methyl–1H–indole–3yl)–n,n–dimethyl ethanamine has consensus count of 3 for seizures and 2 for SIMD. Since two different compounds are common predictions for the two indications, the indication–indication association count for these two indications is 2. To strongly relate the indications in Top lists and limit a large number of associations, we selected indication pairs with predicted psychoactive compound consensus count as follows: ≥ 2 for

the Top10 set, ≥ 3 for the Top25 set, ≥ 4 for the Top40 set, and ≥ 6 for the Top100 set.

Tests for statistical significance

A one–tailed Kolmogorov–Smirnov test [94] was used to compare the distributions of psychoactives in the randomized and non–random distributions as this statistical test is typically used to show two distributions are dissimilar. For all statistical tests performed in this work, we formulated the null hypothesis to be that the distribution of psychoactives predicted to treat mental health disorders can be obtained by chance. Our alternative hypothesis is that the true distribution of psychoactives is greater than the randomized control (hence a one–tailed test). We also performed a one–tailed paired T–test to ensure that the mean of the differences between the test distribution and the randomized distribution is greater than zero.

1.6 Conclusions

Traditional drug discovery is limited by its narrow focus on one or a few targets. Drugs approved for one indication interact with multiple proteins and thereby work across multiple indications. The CANDO platform improves upon the traditional approach by examining all interactions between a compound and a universal proteome. This novel approach enables the study of drugs in a holistic chemoproteomic manner that is especially relevant for the development of compounds intended for treating mental health indications as these complex disorders are mediated by multiple proteins and pathways. In this study, we investigated the compounds previously described by Alexander Shulgin along with additional cannabinoids to identify potential therapies for mental health indications. The results of this study indicate the

selected psychoactive compounds perform better than compounds selected at random for mental health indications.

Conversely, the percentage of mental health indications selected by psychoactives is better than randomly selected compounds. This shows that psychoactives may represent promising leads for the development of therapeutics for the treatment of mental health indications. Specifically, the set shows promising results for sleep–related disorders, binge eating disorders, seasonal affective disorder, and cocaine substance abuse disorder. In addition, the other non–psychoactive compounds predicted by the CANDO platform present in the top–ranked predictions may also represent putative repurposable therapies for mental health indications, which will be explored in future studies. In a broader context, our work illustrates the advantages of using a computational chemoproteomics approach for drug discovery and repurposing by providing mechanistic information on which proteins are involved in the mediation of the therapeutic effect.

1.7 Future work

One of the major conclusions of this work is that relationships between mental health indications can be discovered using the repurposed drugs for a given indication. It remains a future work to generalize this concept to all indications in the general sense and potentially uncover interesting relationships at the protein network level. An attempt to do so is presented below, but work still needs to be done to formalize these concepts in a concrete and statistically significant manner.

The first step is to define a term to represent the number of times a given compound is predicted for the same indication and relate it statistically to the number of know treatments for a given indication. This term in the above work is *consensus count* of the prediction. If the number of known treatments is 1, then the *consen-*

sus count can be 0 or 1. Similarly, if the number of known treatments is 2, the the *consensus count* can be 0, 1, or 2. To normalize for the total number of known treatments, the *consensus count* is divided by the number of known indications to yield the *consensus percentage*; a quantity that has a natural relationship with the function $\frac{1}{x}$. Additionally, the *consensus percentage* increases as the top number of compounds increases. These relationships are shown in Fig 1.6.

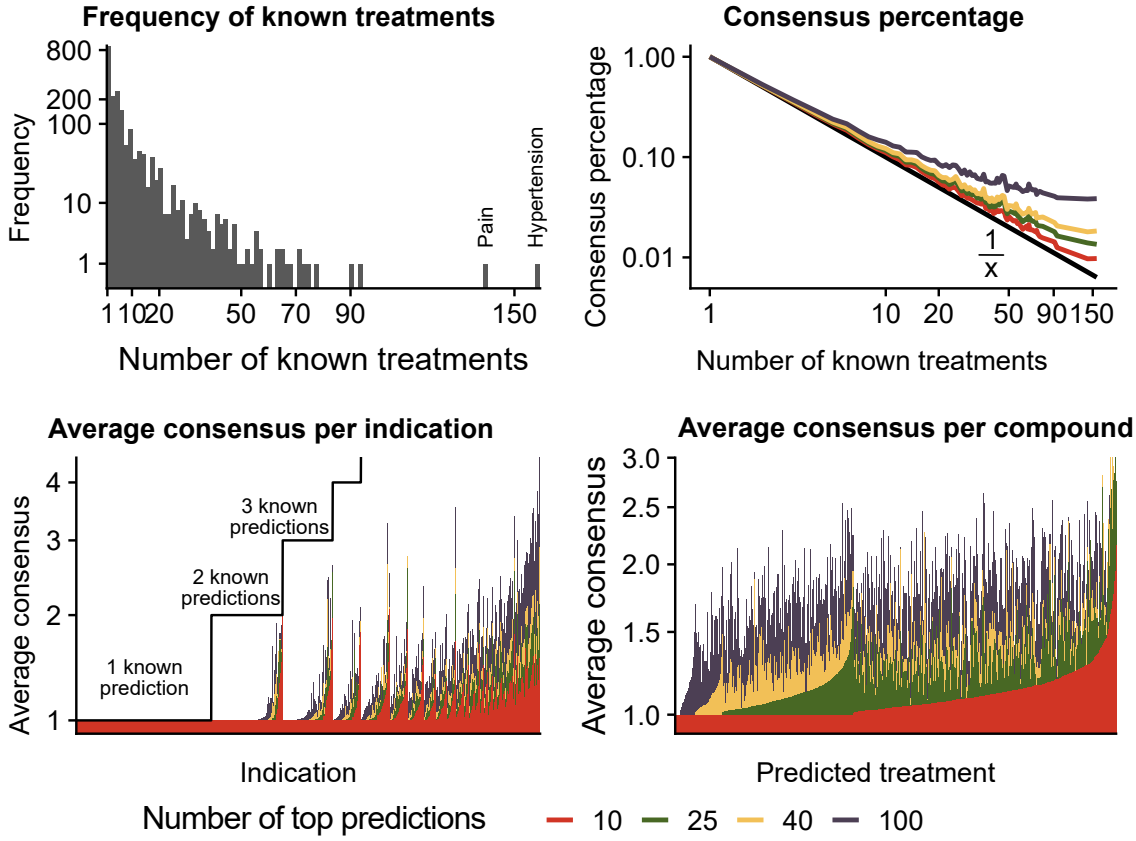


Figure 1.6. The distribution of known treatments is shown in the top left for all indications along with the relationship between the consensus percentage in the top right. The average consensus is given per indication and per compound in the bottom left and bottom right panels respectively.

With the consensus term defined and examined for all predictions for a given compound and/or indication, the spread of these consensus values can shown as Fig 1.7 below. This result shows that the *consensus count* for a given compound predicted for an indication is typically 1, even when multiple compounds are known to treat that indication. This again reinforces the idea that as one includes additional compounds, the *consensus count* will also increase, but also indicates that the number of new compound indication predictions increases as well. A good future work will take this result to help tease out the proper statistics to use for determining how well two indications are related.

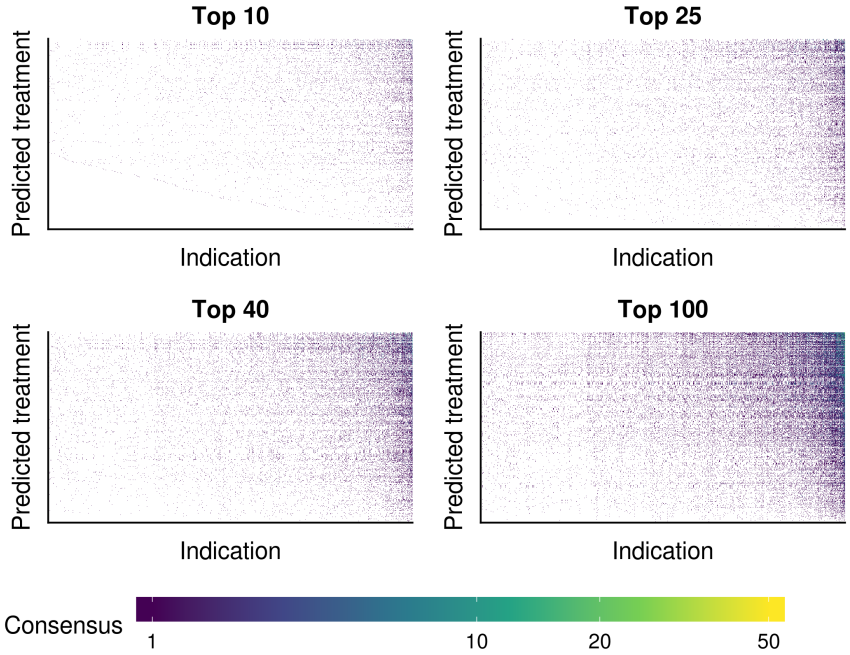


Figure 1.7. The consensus count for the Top 10, Top 25, Top 40, and Top 100 prediction counts.

With this result in mind, it is interesting to note how the number of unique compounds predicted for a given indication grows vs the total number of predicted compound for that indication. This relation, along with the inverse relation where

one examines the unique number of indications predicted for a compound vs the total number of predictions for that compound is given in Fig 1.8. Ideally, in a future work, one can create a statistical model to determine whether a prediction is true using these relationships and later use this result to determine the relationships between different indications.

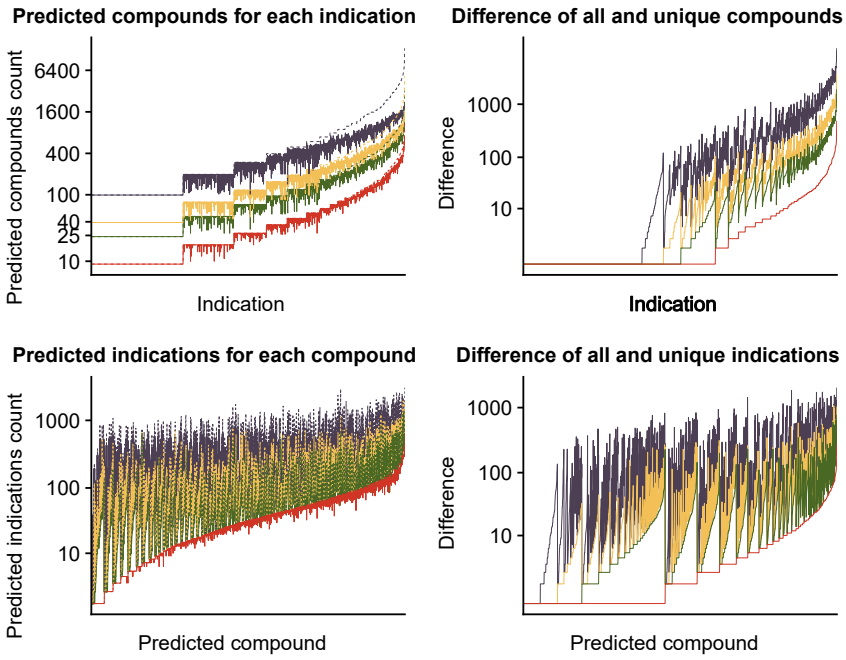


Figure 1.8. The total number of compounds predicted for a given indication and difference between this total and the unique number of compounds is given in the top half of the figure. The bottom half shows the reverse relation where the total number of indications predicted for a compound is given.

Now that the concept of *consensus count* has been explored for the pairing of a single compound and a single indication, we can begin to understand how two compounds can be used to relate two different indications to each other. This concept is codified as the *indication overlap* between the two indications. Initially, we can explore how known treatments can be used to define this overlap frequency and slowly increase the number of compounds predicted to see how the *indication overlap*

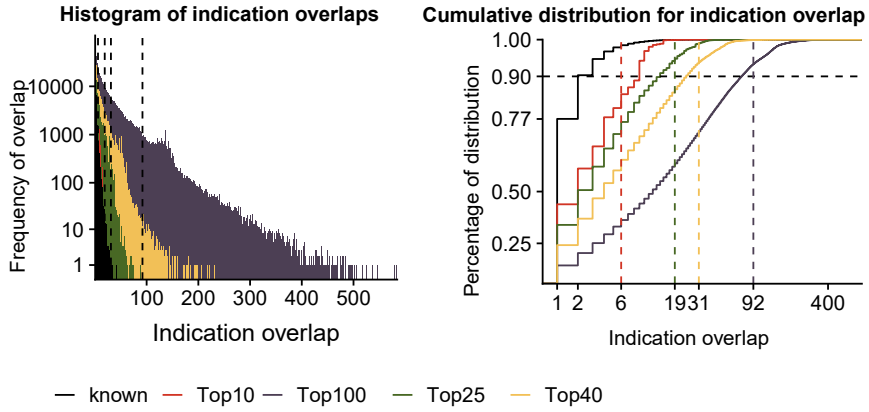


Figure 1.9. The indication overlap for known treatments, Top 10 predictions, Top 25 predictions, Top 40 predictions, and Top 100 predictions shown with both the histogram and cumulative distribution function.

changes as more compounds are added. In Fig 1.9, it can be seen that the *indication overlap* from known compounds is similar to those obtained from the Top predicted compounds.

2. CELL–BASED DRUG DESIGN

The portions of this chapter on the use of BioDynamic Imaging for the prediction of cancer resistance and the use of machine learning to find leads for castration–resistant prostate cancer are not published.

2.1 Mining structural information from the Protein Data Bank

This chapter section is available as

Fine, J., Chopra G. Lemon: a framework for rapidly mining structural information from the Protein Data Bank. *Bioinformatics*, Volume 35, Issue 20, 15 October 2019, Pages 4165–4167.

https://doi.org/10.1093/bioinformatics/btz178

2.1.1 Abstract

Motivation

The Protein Data Bank (PDB) currently holds over 140 000 biomolecular structures and continues to release new structures on a weekly basis. The PDB is an essential resource to the structural bioinformatics community to develop software that mine, use, categorize and analyze such data. New computational biology methods are

evaluated using custom benchmarking sets derived as subsets of 3D experimentally determined structures and structural features from the PDB. Currently, such benchmarking features are manually curated with custom scripts in a non-standardized manner that results in slow distribution and updates with new experimental structures. Finally, there is a scarcity of standardized tools to rapidly query 3D descriptors of the entire PDB.

Results

Our solution is the Lemon framework, a **C++11** library with Python bindings, which provides a consistent workflow methodology for selecting biomolecular interactions based on user criterion and computing desired 3D structural features. This framework can parse and characterize the entire PDB in < 10 min on modern, multi-threaded hardware. The speed in parsing is obtained by using the recently developed MacroMolecule Transmission Format to reduce the computational cost of reading text–based PDB files. The use of **C++** lambda functions and Python bindings provide extensive flexibility for analysis and categorization of the PDB by allowing the user to write custom functions to suite their objective. We think Lemon will become a one-stop-shop to quickly mine the entire PDB to generate desired structural biology features.

Availability and implementation

The Lemon software is available as a **C++** header library along with a PyPI package and example functions at https://github.com/chopralab/lemon.

2.1.2 Introduction

Experimental structures deposited in the Protein Data Bank (PDB) [95] have resulted in several advances for structural and computational biology scientific and education communities. Several software packages have been developed using and applying data available in the PDB. Computational structural biology methods are evaluated using several benchmarking datasets mined from the PDB. As one example, for protein-ligand docking, the Astex [96], PDBbind [97] and DUD–E [98] sets have been used to predict the 3D coordinates of ligands, rank target activity and discriminate binders from non-binders.

Additionally, the knowledge–based forcefields for protein structure refinement [99] and scoring functions used to evaluate ligand poses in a protein binding site [28] require extensive feature mining of the PDB. The process for developing these benchmarking sets, structural features for knowledge-based forcefields and scoring functions are non–standard, time–consuming and computationally challenging as it requires significant computational resources to mine different 3D descriptors in the PDB. Development of software for mining these 3D features and use of them for machine learning methods is challenging due to the increase in individual entry size as a significant computational cost is needed to parse large text–based formats.

The Macro Molecular Transmission Format (MMTF) [100] was recently introduced to significantly reduce the time required to parse text-based formats traditionally used to store crystallographic data. MMTF requires a fraction of the computation time to read multiple files into computer memory as it uses an encoding format tailored specifically to protein and nucleic acid coordinate data and topology. Specifically, MMTF stores connectivity and chemical grouping data not captured in the PDB and mmCIF formats that are leveraged by Lemon's data extraction framework. Lemon uses the entire PDB as Hadoop sequence files that are packaged as 578 independent

subsets for all MMTF entries and used for the development of highly parallel workflows (Fig 2.1). Lemon is the only C++11 software package to our knowledge to parse the Hadoop sequence files natively.

2.1.3 Materials and methods

The Lemon framework uses a paradigm similar to MapReduce developed by Google for mining 'Big Data' [101]. The user provides a 'worker' function that accepts two arguments: an object that represents the structure(s) of the chemical entities, and a string representing the four-letter PDBID. Lemon evaluates this function for all macromolecule entries in a multithreaded manner (Fig 2.1a), allowing one to perform any calculation on the structural information encoded by the MMTF file.

The MMTF object given to the user contains biomolecular data at the atomic, chemical group and molecular levels. This includes the position, name, element type and charge of the biomolecular atoms as well as the name, chain, biologic assembly, chemical links and composition type of chemical groups (e.g. protein residues). These features are examples that can be used to create workflows to select and extract desired 3D interactions.

Since a primary goal of the Lemon framework is to create standardized workflows, we have represented an example workflow pictorially (Fig 2.2). A workflow calculation is performed on the entire PDB database that is stored in its entirety on the user's local machine. However, users can also choose to pre-filter the database using a query generated on the RCSB website.

The workflow examples (Listings) are divided into 'simple,' 'distance-based,' and 'complex' categories based on the computational complexity of the workflow in appendix C. First, the user 'selects' chemical groups present in the PDB entry using functions in Lemon for selecting small-molecules, metals, nucleic acids, amino acids,

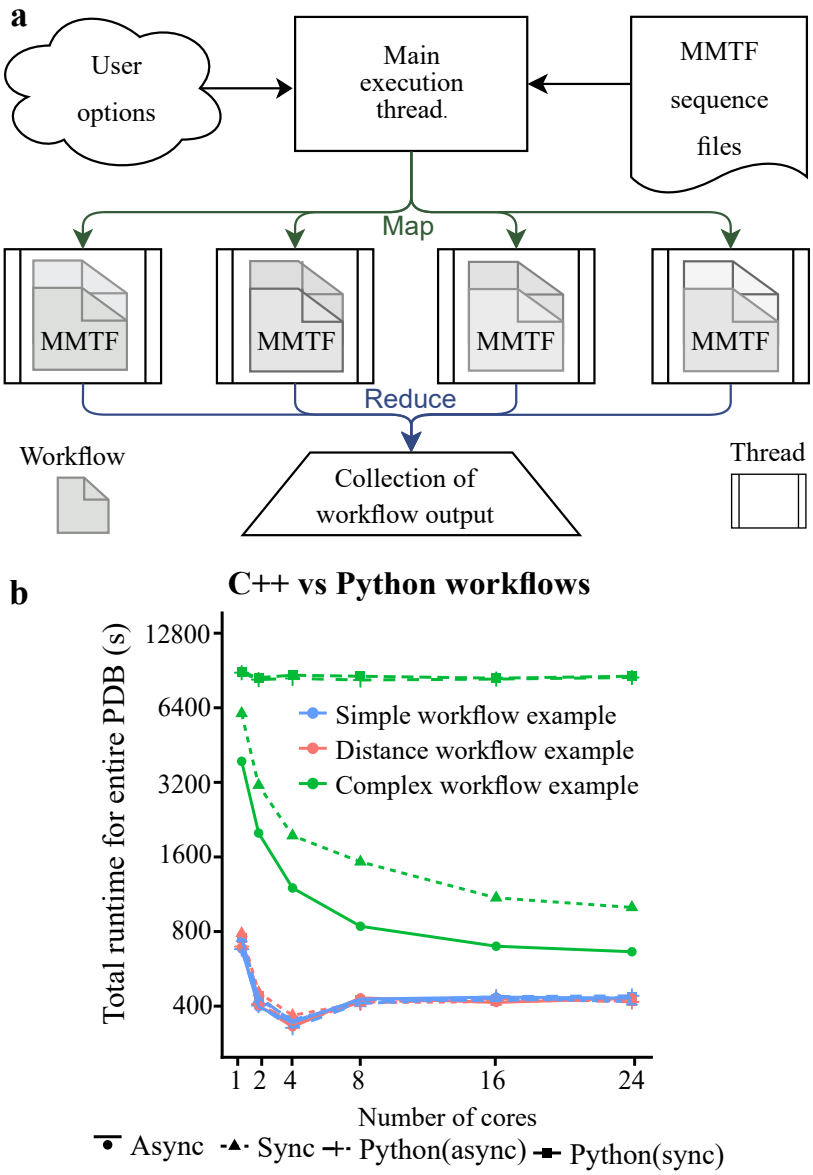


Figure 2.1. Workflow for Lemon. (a) The overall work follows for the Lemon framework is given. The user provides C++ or Python API Lambda functions which use pre-defined functions to query information about each complex to filter the PDB into a desired subset. (b) A comparison between the C++ and Python benchmarking sets, showing the effect of multiple cores on overall runtime for simple to complex workflows for GCC (asynchronous, 'Async' and traditional or synchronous, 'Sync' threading)

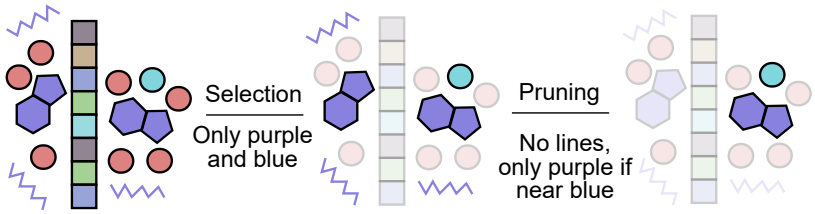


Figure 2.2. Diagram showing the recommended Lemon workflow. The workflow begins with selection when the user provides criterion on which chemical groups, they wish to perform calculations on. In this example, the purple groups represent small molecules, the red groups represent water, the cyan groups represent metals, and the boxed groups represent amino acids. Here, the user has selected small-molecules and metal ions. The next step is pruning of the selected residues. Here, the user has decided to remove the small molecules which do not contain rings and remove small molecules which are not within proximity of a metal ion. Finally, the user can perform a calculation on their selected pairs.

etc. These functions work on the group level by querying the group's size and composition type. Additionally, it can also include the selection of topological information. Examples for these selectors are given in Listings 1–6 in appendix C.

After obtaining a list of groups, the user can further divide ('prune') these groups based on 3D environment, biologic relevance or frequency in the PDB. Lemon provides functions to find biologically identical groups, common groups and interacting groups via spatial relationships in 3D. Example lambda functions for 'pruning' groups are given in Listings 7–12 in appendix C.

Finally, a workflow will calculate a feature of interest. For example, a user may perform structural alignment to a reference protein (Listing 13), calculate a docking score (Listing 14) or output statistics on geometries of bonded entities (Listings 15–18 in appendix C). To show case the Python version of Lemon, three example workflows were ported to Python (Listings 19–21 in appendix C). The information obtained

from these workflows can then be directly used in machine learning approaches and the development of new structural biology methods.

Lemon also implements two different threading models based on the specifications of the C++ standard library. The first is a traditional (synchronous, 'sync') threading approach which divides the PDB into 578 subsets and launches a user-defined number of threads to handle an equal portion of these 578 subsets (e.g. if the user selects two threads each thread will handle 289 subsets). The second is an asynchronous ('async') model that schedules 578 threads and executes a given number of them in parallel. Specifically, for async, the next queued thread executes when a thread completes, compared to the 'sync' model that requires all threads to complete.

Benchmarking Lemon

All Lemon benchmarks were run on the Brown high performance computing cluster. The nodes that comprise this cluster have 96 Gb of memory and two Sky Lake CPUs clocked at 2.60GHz, resulting in a total of 24 cores. Further details about this compute cluster can be found at https://www.rcac.purdue.edu/compute/brown.

The calculation of the timings for individual workflows versus the overall runtime was done by using the LEMON_BENCHMARK flag during compilation. These timings include decompression of the MMTF file using the Gzip algorithm but include neither the time required to read the compressed MMTF from the Hadoop sequence file into memory, nor the time required to output the results. These timings are printed to STDERR after the workflow completes for a single entry.

This procedure was performed three times using different compilation settings to understand the difference between these settings and overall workflow runtime. The three settings are (1) using the Intel **C**++ Compiler 17.0.1.132 with traditional (synchronous) threading enabled, (2) GNU **C**++ Compiler 6.3.0 with traditional

threading, and (3) GNU C++ Compiler 6.3.0 with asynchronous (async) threading enabled.

Lemon jobs were submitted to the cluster's using the following script when different processor counts where supplied during submission. All benchmarking was performed on a single node unless otherwise specified.

```
#!/usr/bin/env bash
#PBS -d .
#PBS -l walltime=04:00:00

if [[ -z $LEMON_PROG ]]
then
    echo "You must specify the LEMON_PROG variable"
    exit 1
fi

PPN=$(wc -l $PBS_NODEFILE | cut -f1 -d' ')

# /dev/shm is the location of shared memory
# on RedHat systems.
# you may need to change this location!
tar -xf full.tar -C /dev/shm/

SECONDS=0
time lemon/build_${compiler}/bin/lemon/$LEMON_PROG \
     -w /dev/shm/full -n $PPN > ${LEMON_PROG}.log

echo "$LEMON_PROG $PPN $compiler $SECONDS"

rm -fr /dev/shm/full
```

Obtaining Lemon

The Python version of Lemon is available on the PyPI package repository and can be installed on Windows, MacOS and Linux using the following command. Note that

this only installs the Python version of Lemon and does provide access to the C++ API.

python3 -m pip install —-user candiy–lemon

The lemon benchmarking framework is written in C++11. Its only dependencies are a C++11 compiler, the CMake tools, and the Chemfiles library. Note that the Chemfiles library will be automatically installed by the lemon install scripts if it is not already installed your system. Note that Lemon itself is a 'header-only' library, but users will also need a copy of the Chemfiles library to include it in their projects.

There are two threading methodologies that can be used by Lemon. The default is based on std::thread provided by the C++11 standard and should compile with any C++11 compiler. A second methodology uses a thread pool which uses std::async and requires C++14 features.

Lemon has been freely released on GitHub. To obtain the software, complete the following steps:

```
git clone https://github.com/chopralab/lemon.git
cd lemon/
mkdir build/
cd build
cmake .. –DCMAKE_BUILD_TYPE=Release
make
```

The example binaries will be created in the 'progs' subdirectory of 'build'. It is recommended that you supply the 'make' command with an additional argument '-j NUM' where 'NUM' is the number of physical cores on your machine. To use the std::async version of Lemon, add -DLEMON_TEST_ASYNC=ON to the cmake configuration line. To build documentation, add -DLEMON_BUILD_DOCs=ON to the configuration line. An online version of this documentation is available at https://chopralab.github.io/lemon/latest/.

2.1.4 Results and discussion

Querying the PDB takes minutes

To measure Lemon's execution time, we ran all example listings provided for different levels of multithreading and compiler architectures. The calculations were performed on a community cluster with each node consisting of two 12-core Intel Xeon Gold 'Sky Lake' processors. There are differences in computational time for a 'simple,' 'distance-based,' and 'complex' workflow (Listings 6, 10 and 18) including the time to decompress and parse the MMTF files (Fig. 2.3). The average runtime for all workflows with 'async' threading on eight cores (commodity hardware) takes ∼8 min to complete. The Lemon outputs for these queries are shown in appendix C.

Workflow runtime influences threading efficiency

Asynchronous threading is more efficient for 'complex' workflows compared to sync threading (Fig 2.1b). Theoretically, the sync threading time should be more than async because it needs to wait for other threads to complete. However, the async and sync runtimes are similar for 'simple' and 'distance-based' workflows (Listings 6 and 10) but differ for complex workflow (Listing 18) for 2 and 4 cores (Fig 2.4). The runtime reduces with an increase in the number of cores (see 1, 2 and 4 cores in Fig 2.5). However, for some simple and distance-based workflows runtime increased from 4 to 8 cores (Fig 2.5). This result may be due to increased performance penalty for atomic (thread locking) operations after completion of each thread. This hypothesis is supported by the continued increase in performance for 'complex' operations as they are less likely to become bound.

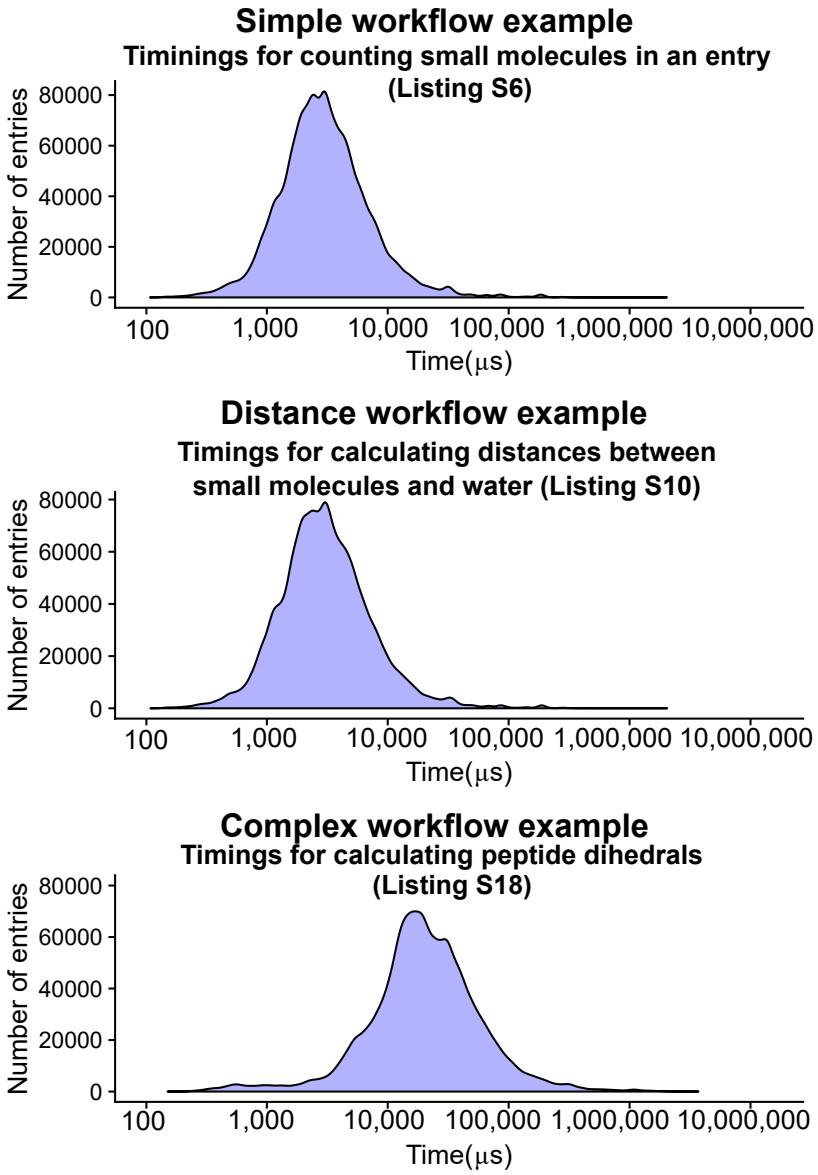


Figure 2.3. Timings for individual workflows are given as examples from the three different types of workflows. These times were taken from a single core launch to ensure that each timing was as independent of other calculations. These results indicate there is little difference between the 'simple' and 'distance-based' calculations, a potential result of the reduced computational cost due to carefully 'selecting' and 'pruning' chemical groups before performing the distance calculation.

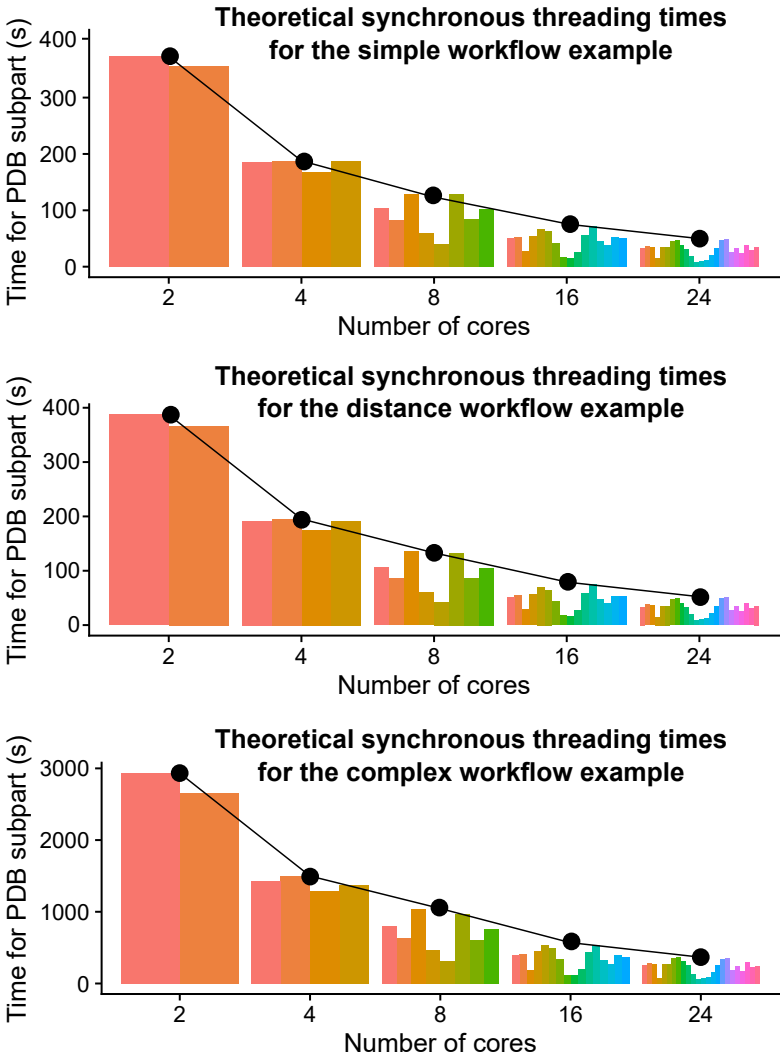


Figure 2.4. Theoretical minimum performance of each traditional thread as computed for the three examples shown in Fig. 2.3. These are calculated by grouping the individual entries by their subgroup and summing the total time. The result is the colored subpart, which is dependent on the number of cores executed by the user as this number is used to calculate the total number of subparts. This plot shows that the maximum runtime of a subpart decreases as the number of cores increase for all three example operations (black line). It also indicates that the time taken by each sub part is the same for 2 and 4 cores but diverges for higher core counts.

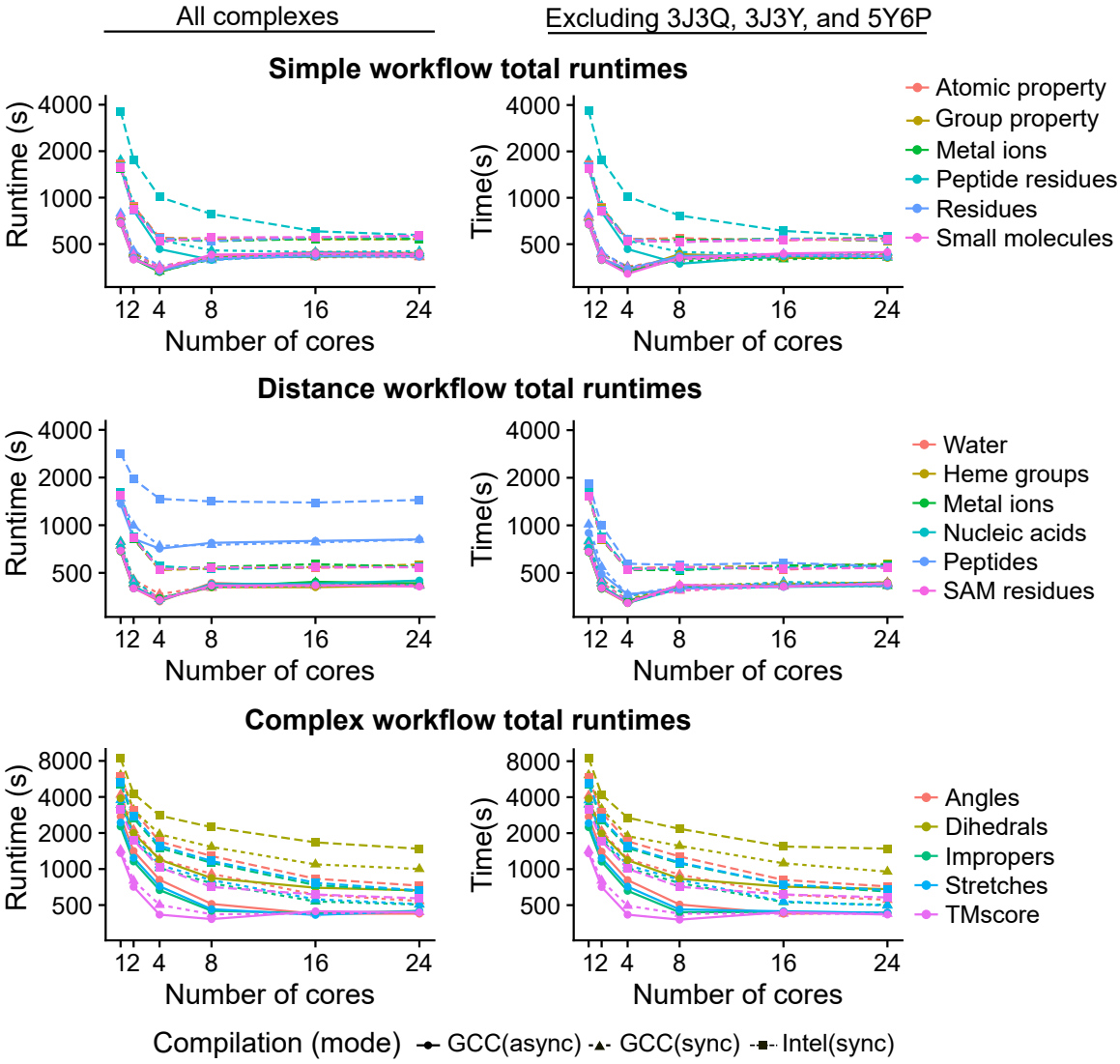


Figure 2.5. Benchmarking results for the Lemon workflows listed previously in this document. Here, we have divided these workflows by their relative complexity. We ran the benchmarking set for the entire PDB with (left column) and without the three largest size PDB entries, 3J3Q, 3J3Y, and 5Y6P (right column). These entries have a processing time at least 3 times greater than the remaining entries. Note that runtimes given in the Y-axis are plotted logarithmically. These plots show that 4 cores provide the optimal run time for 'simple' and 'distance-based' operations. Additional cores do improve runtime for 'complex' operations, however, indicating the possibility of an Input-Output bottleneck on fast calculations.

Large biomolecules do not affect runtime

Fig 2.5 shows that removal of the largest size PDBs (3J3Q, 3J3Y, 5Y6P) does not significantly reduce the overall runtime for most workflows when compared to the entire PDB (left column in the figure). An exception is the calculation of small-molecule/peptide interactions that requires distance calculations between millions of atoms for large complexes (see Peptides in Fig 2.5). Hence, Lemon workflows scale with the size PDB entries. This is a significant result given the increase in the amount of large structures in the PDB (RCSB stats page).

Compiler choice significantly impacts runtime

The selection of the C++ compiler dramatically affects the performance of Lemon (Fig 2.6). However, the timings shown in Fig 2.6 indicate that there is only a marginal difference between the 'sync' and 'async' models averaged over all workflows. The GNU Compiler Collection (GCC) version 6.3.0 with 'sync' threading compilation outperforms the Intel compiler version 17.0.1.132 with sync threading (Fig 2.6, green and blue bars). This discrepancy could be a result of GCC's use of a modern version of the C++ standard library or the specific optimizations performed by this compiler are better for Lemon. Further profiling is beyond the scope of this work and may be addressed in future publications.

Python is slower than C++ for complex workflows

The data shown in Fig 2.1b indicates that the Python bindings are just as fast as the C++ version for 'simple' and 'distance-based' workflows. Complex calculations scale poorly with the number of cores, a result due to the Python global interpreter

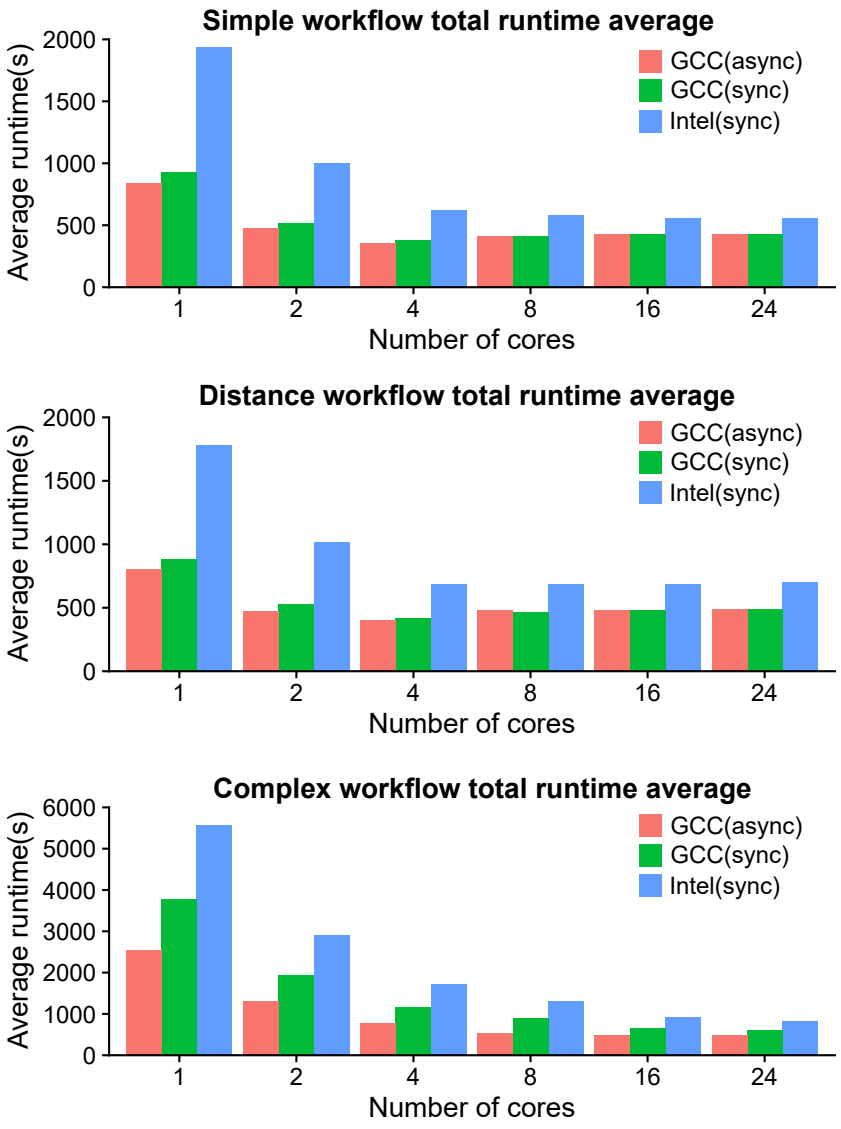


Figure 2.6. The average runtimes for the three different compiled versions of Lemon. These data show that overall the GCC compiler out performed the Intel compiler int all test cases. Further, they show the 'async' threading model is only marginally faster for the 'simple' and 'distance-based' workflows but holds improvements for the 'complex' calculations.

lock. This underlines the importance of development in the C++ language, potentially after prototyping a complex workflow in Python.

Code availability

Lemon is hosted on GitHub (see 'Obtaining Lemon') along with C++ and Python API documentation on the GitHub page repository. File input and output are provided by the Chemfiles [102] library. A link to the Lemon GitHub repository has been added to the official MMTF webpage on mmtf.rcsb.org.

2.2 Identification of differing cell populations through the measurement of biological species

While the ability to identify to model cells using protein coordinates is important, so is the ability to identify the state of the cell through its expressed lipids, proteins, and metabolites (a biological specie). A technique utilizing MS/MS has been developed to measure the expression of these species. However, the measurement of these species is noisy and it can be difficult to determine statistical significance given the amount of noise. The analytical technique produces an ion count for each specie, but this number can be measured for blank and control samples. Fortunately, these ion counts follow a negative–bioomial distribution which allows it to be modeled using techniques similar to those used for RNA–seq. Using these details, the following statistical technique has been developed and applied to several biological samples.

All statistics determined for the comparisons between cells treated with a condition versus a control were calculated using the edgeR package [103]. Here, the ion count for a given biomarker (i.e. protein, lipid, or metabolite) will be referred to using the subscript s for the sample (cell replicate for a class of analyte) and b for the specific biomarker (i.e. a single lipid). An additional 'intercept' sample is added to model the experimental blank performed using just the injection media to ensure that all comparisons are significant with respect to this control. The edgeR package

fits a generalized linear model to the following log–linear relationship for the mean–variance: $\log \mu_{bs} = X_b^T \beta_g + \log N_s$ for each biomarker b in sample s where the sum of all ion intensity for sample s sums to N_s. This allows for the calculation of the coefficient of variation (CV) for the ion count for a biomarker in a sample (y_{bs}) using the following relationship $CV^2(y_{bs}) = 1/\mu_{bs} + \Phi_b$ where Φ_b is the dispersion of the biomarker. This dispersion term is estimated using the common dispersion method [104]. These values are used to calculate the associated log–fold change between treated and non-treated cells along with the p-values are obtained using the likelihood ratio test. These p–values are then adjusted for multiple testing using the BH method to obtain false discovery rates [105].

An example project with Priya Prakash using primary microglia is shown below for cells treated with amyloid–beta compared to a control group. As can be seen from the principal component analysis plots in Fig 2.7, this technique can be used to identify significantly different metabolites in these samples.

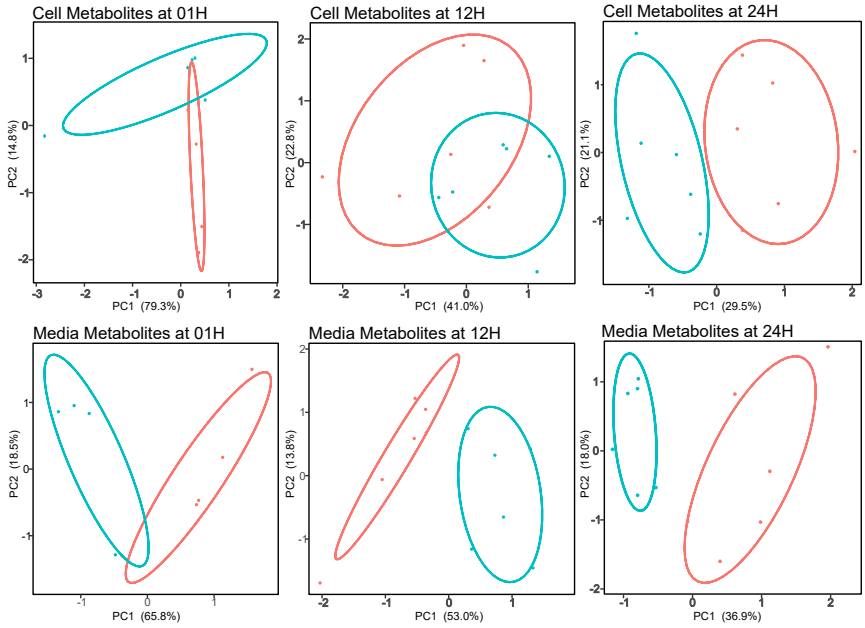


Figure 2.7. Red is Abeta treated, blue is control.

2.3 Combining Biodynamic Imaging and RNA-sequencing yields an improved machine-learning model for predicting resistance to chemotherapy in canine lymphoma

This section is currently under preparation for submission to a peer–reviewed journal as a brief application note. It contains contributions from Deepika Dhawan, Sagar Utturkar, Phillip San Miguel, Gaurav Chopra, John Turek, David Nolte, Michael Childress, and Nadia Lanman.

2.3.1 Abstract

Diffuse large B-cell lymphoma (DLBCL) is a common, aggressive cancer diagnosed in approximately 25,000 patients each year, 1/3 of whom will die from the disease. Challenges in predicting those patients whose cancers will respond well to a given therapy is a major reason for this lack of success. A novel method to predict the effectiveness of therapy for individual patients is desperately needed. Recently, dogs with naturally-occurring DLBCL have been proposed as a valuable model in which to develop novel personalized medicine strategies for humans with this cancer. In this study, Biodynamic Imaging and RNA-seq data were collected on tumor samples from pet dogs with spontaneous DLBCL, before and after chemotherapy treatment. Dogs were classified as sensitive or resistant to chemotherapy and data were integrated to build a machine learning model which is a perfect classifier for predicting sensitivity versus resistance to chemotherapy from pre-chemotherapy data alone. Together, these data show that BDI and RNA-Seq data, with careful feature selection can be generalizable predictors of chemotherapy response in a disease that is notoriously difficult to treat in part due to the heterogeneity observed in response to standard of care therapies.

2.3.2 Introduction

Diffuse large B-cell lymphoma (DLBCL) is characterized by marked molecular and biochemical heterogeneity that have confounded the use of targeted drugs to improve cure rates from conventional chemoimmunotherapy [106]. Genetic analysis alone is insufficient to predict the response of individual cases of DLCBL to drug therapy [107]. Here, we close this predictive gap with a novel technique termed biodynamic imaging (BDI) [108–113], an optical imaging technology that records phenotypic responses of fresh, three-dimensional tumor tissues to chemotherapeutic drugs in the ex vivo setting. These responses are identified via Doppler spectroscopy, the data from which has been statistically associated with clinical outcomes such as objective tumor response or survival time. Although preliminary results show that BDI predicts chemosensitivity of naturally-occurring DLBCL in dogs [112], the relationship of BDI data to molecular processes underlying a tumor's phenotypic drug response has yet to be defined. We show that combined with gene expression and BDI data create a perfect classifier of clinical chemotherapy response in canine DLBCL. Due to the success of machine learning applications across chemistry and biology [114–116], we explore multiple machine learning methodologies for this classifier and discuss the implications of this model's predictions for the biology of these tumors.

2.3.3 Methods

In this study, biodynamic Imaging (BDI) was performed on nineteen dog tumors (see Table 2.3.3 for full information). Details for the collection, culturing, and imaging of these dogs are given in the supporting information under BioDynamic Imaging (BDI) and Analysis. This analysis yielded a total of 81 features from 5 chemotherapy treatments tested on these ex vivo tumor samples. Of these tumors, 6 patients were

later categorized as resistant to chemotherapeutics (progression-free survival time < 100 days) and the remaining 13 patients were considered sensitive to chemotherapy (progression-free survival time > 200 days). In addition to BDI, gene expression was measured using RNA-seq and analyzed using a standard RNA-seq pipeline. Details for the quality control, alignment, and differential expression analyses of the RNA-Seq data are given in the supporting information under Details for RNA-Seq analysis. RNA-seq was performed both before and after chemotherapy, yielding a pre and post-treatment set of genes for analysis. A total of 37 statistically significant differentially expressed genes identified by two independent differential expression analysis packages in sensitive vs resistant samples were combined to ensure complete coverage of the relevant transcriptomic landscape [103, 117, 118]. RNA-seq and BDI features were combined to yield a total input feature space of 119 features which can be used to predict the sensitivity of the lymphomas. Given that the split between the number of resistant and sensitive lymphomas is uneven, we decided to use the Cohen Kappa Statistic [119] to determine the success of the models where the resistant tumor samples were taken to be the positive case and the sensitive tumor samples were taken to be the negative case. The Caret machine learning software package [120] was used to train all models where the train control was set to "Leave one out cross–validation" (LOOCV) for hyperparameter tuning. This technique trains the model on all but one patient tumor (training set) and evaluates the resulting model on the tumor not used for training (validation set) to find the methodology and parameters which generalize best for the input features. Typically, these parameters are used to train a final model that is tested against patients not used for LOOCV (a test set), but due to the lack of data available for training, each patient samples were iteratively left out for LOOCV to enable testing of the final model on the left out patient. This process is repeated in a similar manner to that of LOOCV. The full description for

this training, validation, and testing paradigm is given in Appendix A and referred to as 'Leave one out Testing' (LOOT). The code required to perform this validation technique can be found at github.com/pccr/bdi_rna_seq_for_lymphoma.

2.3.4 Results and Discussion

Regularized logistic regression is an established machine learning technique that is known to generalize well to data not seen in its training or validation sets [121,122]. A regularized logistic regression model trained solely on all 81 BDI variables yielded a Kappa statistic of 0.11 when validated using LOOCV, likely due to overfitting resulting from too many input variables. To address this, we ranked all BDI variables based on their Area Under the Precision Recall Curve (AUPRC, Tables 2.3) and their Area Under the Receiver Operator Curve (AUROC, Table 2.4) and empirically found that the following 3 BDI variables (SDIP1dox, LOF0chop, and ALLF1pred) yielded the best model with a Kappa statistic of 0.42. A second model trained on only the 37 RNA-seq variables identified as statistically significant either in the pre- or the post-treatment tumor samples yield a validation Kappa value of 0.46. It was hypothesized that a combination of these two variable types would yield a more generalizable model than one trained on only one type of variable. Hyper parameter details are given in Appendix D.

Simply combining the All 81 BDI and 37 RNA variables resulted in a Kappa statistic of 0.20 but using the best 3 BDI variables and 37 RNA variables yielded a Kappa of 0.46. Therefore, we decided to reduce the number of RNA variables as well and simply used all the significant variables in the pre-treatment (ENSCAFG0000011225 and ENSCAFG00000004237 or KIAA1217) and the top three RNA variables as per their AUPRC (Table 2.3.4, ENSCAFG00000004237 KIAA1217, ENSCAFG00000005330 or SHD4A, ENSCAFG00000016518 or FGFR4) for a total of 4 new variables as

Table 2.1.
Details on the nineteen canines evaluated in this study. Each patient is assigned a unique sample ID for pre and post chemotherapy and a tumor sample barcode to represent the dog both before and after treatment. Since the post treatment RNA-seq data is only used to identify additional genes and not used for the creation of any models, the Dog Identifier is used in all future figures and tables.

Sample	RNA Condition	Dog Identifier	Clinical Outcome
12BDI	Pre	LY05	Sensitive
31BDI	Post	LY04	Sensitive
100BDI	Pre	LY04	Sensitive
80BDI	Pre	Ly43BD	Sensitive
RL84BD	Post	Ly58BD	Resistant
RL60BD	Pre	Ly58BD	Resistant
95BDI	Pre	Ly51CL	Sensitive
21BDI	Post	LY03	Sensitive
RL52CC	Pre	LY03	Sensitive
LY12CW	Post	case782-104	Sensitive
75BDI	Pre	case782-104	Sensitive
16BDI	Post	LY02	Resistant
42BDI	Pre	LY02	Resistant
RL06CJ	Post	Ly42GJ	Sensitive
LY54CJ	Pre	Ly42GJ	Sensitive
RL46JD	Post	LY08	Sensitive
LY11JD	Pre	LY08	Sensitive
LY29JW	Post	LY01	Resistant
LY16JW	Pre	LY01	Resistant
96BDI	Post	LY06	Resistant
RL10KM	Pre	LY06	Resistant
RL82KG	Post	case785-528	Sensitive
RL69KG	Pre	case785-528	Sensitive
RL89LB	Post	case786-844	Resistant
LY99LB	Pre	case786-844	Resistant
LY72MS	Post	Ly83MS	Sensitive
43BDI	Pre	Ly83MS	Sensitive
LY79SM	Post	LY10	Sensitive
LY54SM	Pre	LY10	Sensitive
LY49SP	Post	LY09	Resistant
37BDI	Pre	LY09	Resistant
LY17TT	Post	LY07	Sensitive
94BDI	Pre	LY07	Sensitive
RL02YB	Pre	Ly01YB	Sensitive

Table 2.2.
The AUPRC for each of the differentially expressed genes in both pre- and post-condition.

ENSEMBL id	AUPRC	external gene name
ENSCAFG00000004237	0.978016	KIAA1217
ENSCAFG00000005330	0.959124	SH2D4A
ENSCAFG00000016518	0.954301	FGFR4
ENSCAFG00000023923	0.911471	VSIG10L
ENSCAFG00000001316	0.89246	DAPK1
ENSCAFG00000028669	0.886733	FAM171B
ENSCAFG00000011158	0.88414	CFAP46
ENSCAFG00000019071	0.840163	TRARG1
ENSCAFG00000001672	0.832966	LEP
ENSCAFG00000005023	0.748777	HRH1
ENSCAFG00000030137	0.687407	RGS13
ENSCAFG00000010274	0.648399	CHI3L1
ENSCAFG00000006735	0.642685	GZMA
ENSCAFG00000000408	0.63348	IFNG
ENSCAFG00000023928	0.63312	CCL8
ENSCAFG00000018470	0.628268	PLD6
ENSCAFG00000003665	0.626614	SLC17A7
ENSCAFG00000025287	0.625136	GZMB
ENSCAFG00000001835	0.618981	ABCB1
ENSCAFG00000000257	0.612058	IL20RA
ENSCAFG00000013940	0.611059	CLEC4E
ENSCAFG00000025299	0.596955	AOX2
ENSCAFG00000006240	0.593768	MCF2L
ENSCAFG00000012762	0.592697	COCH
ENSCAFG00000011225	0.58571	
ENSCAFG00000001635	0.572127	MLLT3
ENSCAFG00000015883	0.571482	NRG3
ENSCAFG00000008523	0.571264	P2RY14
ENSCAFG00000009569	0.570568	SPP1
ENSCAFG00000005056	0.568922	SCN10A
ENSCAFG00000011168	0.565751	NRGN
ENSCAFG00000007621	0.555166	C1H19orf12
ENSCAFG00000014860	0.55239	CASP4
ENSCAFG00000005835	0.54928	ZIC2
ENSCAFG00000002835	0.526799	CTNNAL1
ENSCAFG00000004855	0.523287	EPB41L5
ENSCAFG00000017656	0.52102	PSTPIP2

Table 2.3.
The AUPRC for each of the BDI biomarkers. This value is calculated where the 'sensitive' class is taken to be the control value and the 'resistant' class is taken to be the comparison value. These values are calculated independently of each other and represent how well each biomarker can be used to predict the clinical outcome of a patient.

BDI Variable	AUPRC	BDI Variable	AUPRC	BDI Variable	AUPRC
ALLF1_pred	0.93056	LOF0_chop	0.92794	SDIP1_dox	0.89674
HI1_cyclop	0.89043	ALLF0_chop	0.88747	LOF0_pred	0.87307
LOF1_cyclop	0.87300	MID0_chop	0.87291	SDIP2_chop	0.87247
HI1_pred	0.87191	ALLF1_cyclop	0.87103	LOF0_dox	0.86012
DSF_cyclop	0.85813	CDIP1_dox	0.84696	SDIP2_vinc	0.84020
LOF1_pred	0.83219	MID2_cyclop	0.83093	ALLF2_cyclop	0.82473
CDIP2_cyclop	0.82369	HI2_dox	0.82110	MID1_cyclop	0.81612
MID2_dox	0.81430	DNSD_cyclop	0.81130	LOF1_vinc	0.80992
DNY_cyclop	0.80770	ALLF2_dox	0.80303	CDIP2_vinc	0.80138
DBSB_dox	0.80081	DNY_pred	0.77775	LOF0_cyclop	0.76925
DDR_cyclop	0.76669	SDIP2_pred	0.76622	DSF_pred	0.76269
MID1_pred	0.74951	LOF2_pred	0.74792	ALLF1_vinc	0.74631
ALLF1_dox	0.74140	HI1_chop	0.74085	CDIP0_dox	0.73929
DSF_dox	0.73407	DDR_dox	0.73114	ALLF0_dox	0.73004
DDR_pred	0.72820	DBSB_vinc	0.72749	LOF0_vinc	0.71543
DHW_dox	0.71345	DDR_chop	0.71088	ALLF2_chop	0.70642
HI1_vinc	0.70440	DDR_vinc	0.68787	CDIP2_chop	0.68486
CDIP0_chop	0.67647	DNY_vinc	0.67431	DHW_cyclop	0.66812
DSF_vinc	0.66756	ALLF2_vinc	0.66078	DHW_pred	0.65882
MID0_dox	0.65515	CDIP1_pred	0.65308	DSF_chop	0.64619
DNY_dox	0.61599	SDIP1_pred	0.60346	CDIP0_cyclop	0.59756
CDIP2_pred	0.58986	CDIP0_vinc	0.58612	LOF2_vinc	0.58452
LOF2_chop	0.58039	CDIP0_pred	0.57298	SDIP0_vinc	0.56466
SDIP0_dox	0.55826	LOF1_dox	0.55717	HI0_vinc	0.55512
SDIP1_cyclop	0.55475	SDIP0_chop	0.54384	HI0_dox	0.54278
DNSD_dox	0.54139	DHW_vinc	0.53620	SDIP1_vinc	0.53178
CDIP1_vinc	0.53056	SDIP0_pred	0.52287	CDIP1_cyclop	0.49202

Table 2.4.
The AUROC for each of the BDI biomarkers. The AUROC is calculated where the 'sensitive' class is taken to be the control value and the 'resistant' class is taken to be the comparison value. These values are calculated independently of each other and represent how well each biomarker can be used to predict the clinical outcome of a dog.

BDI Variable	AUROC	BDI Variable	AUROC	BDI Variabl	AUROC
LOF0_chop	0.83333	ALLF1_pred	0.82692	LOF0_pred	0.80128
LOF1_cyclop	0.79487	SDIP1_dox	0.75	LOF1_pred	0.74359
LOF1_vinc	0.73077	DSF_cyclop	0.71795	ALLF1_cyclop	0.71795
ALLF0_chop	0.71795	HI1_cyclop	0.71795	DBSB_dox	0.70513
MID0_chop	0.69872	HI1_pred	0.69872	SDIP2_chop	0.69231
MID1_pred	0.69231	SDIP2_vinc	0.68590	DNY_cyclop	0.66667
CDIP1_dox	0.66026	DDR_pred	0.66026	LOF0_dox	0.64744
DNSD_cyclop	0.64103	CDIP2_vinc	0.64103	MID2_dox	0.62180
MID1_cyclop	0.61538	CDIP2_cyclop	0.61538	HI2_dox	0.61538
DBSB_vinc	0.60897	MID2_cyclop	0.60897	DNY_pred	0.60256
LOF0_vinc	0.58974	DDR_dox	0.58333	ALLF1_vinc	0.58333
DSF_pred	0.57692	DDR_cyclop	0.57692	ALLF2_cyclop	0.56410
CDIP0_chop	0.55769	DDR_vinc	0.55128	ALLF2_dox	0.55128
CDIP1_pred	0.55128	DHW_pred	0.55128	ALLF2_vinc	0.55128
DSF_dox	0.53846	LOF0_cyclop	0.53205	CDIP0_dox	0.53205
DSF_chop	0.52564	DHW_dox	0.52564	CDIP2_chop	0.51282
DDR_chop	0.51282	ALLF0_dox	0.51282	SDIP2_pred	0.51282
MID0_dox	0.5	DSF_vinc	0.5	HI1_vinc	0.5
HI1_chop	0.48077	DNY_vinc	0.47436	ALLF1_dox	0.47436
LOF2_pred	0.46154	DHW_cyclop	0.45513	CDIP2_pred	0.41026
ALLF2_chop	0.40385	SDIP1_pred	0.39744	CDIP0_vinc	0.38462
LOF2_vinc	0.36538	CDIP0_pred	0.35897	CDIP0_cyclop	0.35256
DNY_dox	0.33974	LOF2_chop	0.33974	SDIP0_vinc	0.30769
HI0_vinc	0.30769	SDIP1_cyclop	0.30128	LOF1_dox	0.28205
DNSD_dox	0.26923	HI0_dox	0.25641	SDIP1_vinc	0.25641
CDIP1_vinc	0.24359	DHW_vinc	0.24359	SDIP0_chop	0.20513
SDIP0_pred	0.20513	SDIP0_dox	0.15384	CDIP1_cyclop	0.11538

KIAA1217 is repeated. We checked the correlation between these RNA values and the BDI variables to ensure that there was no significant correlation (Fig 2.8a) and that these 7 variables could be used to separate the 19 patients (see the Principal

Component Analysis in Fig 2.8b) where additional correlation and PCA plots are given in the supporting information (Fig. 2.9). A logistic regression model obtained for a model trained on these 7 variables yielded a validation kappa of 0.88. These results show that the selected variables are apt for predicting the resistance of a given patient lymphoma. Details for the models are in Appendix D.

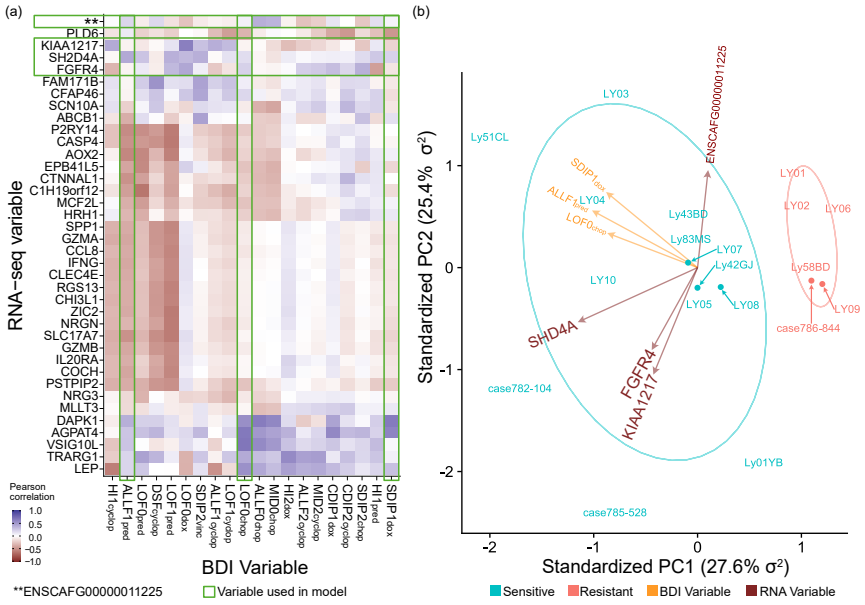


Figure 2.8. Correlations between the top 20 BDI variables and statistically significant RNA-seq variables are given in (a) where green boxes show the decision tree selected variables used to build the logistic regression model. A principal component analysis plot is given in (b) to show how the selected variables separate the resistant versus sensitive lymphoma tumors. Additional correlation plots for the Spearman and Kendall correlations are provided in the supporting information.

To rigorously ensure that our model performs better than the other possible models, we perform Leave One Out Testing (LOOT) on the All BDI, Best 3 BDI, All RNA, and combined BDI and RNA models (Table 2.5). These results show that the combined model with full variable normalization (row 5) outperforms the models trained on only one type of variable (rows 1 and 2). However, we were concerned about the

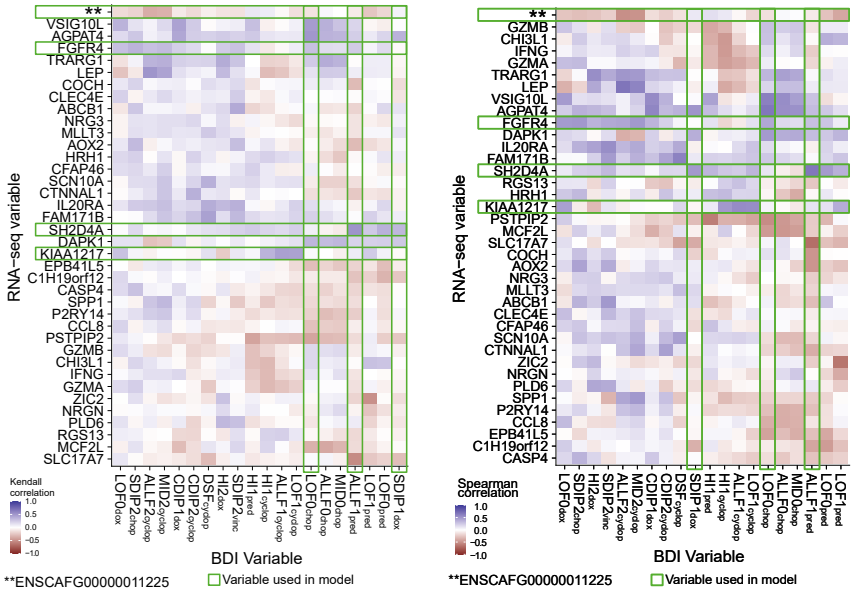


Figure 2.9. Correlation plot for BDI variables and RNA-seq variables. For these correlations, the Kendall tau (a) and Spearman rho (b) correlation is used to calculate the ordinal relationships between the variables. All nineteen dogs are used to calculate these correlations. The variables used to create the final model are highlighted with green. The Principal component analysis (PCA) plot for the RNA and BDI variables are shown to represent the difficulty in separating the resistant from sensitive dogs using only RNA (c) or only BDI (d) variables.

normalization of the BDI variables as this leads to a dependence on the original mean and standard deviation of the 19 patients. To address this, we wished to remove this normalization from the model and we observed that the test set Kappa improved (final row of Table 2.5). We believe this is a result contributes to the stability of the BDI variables in their use of predicting canine lymphomas.

Table 2.5.
Summary of Leave One Out Testing. Details are given in Appendix D

Model	Accuracy	Precision	Recall	F1	Test Set Kappa	Mean Validation Kappa
All BDI variables	0.6316	0.4000	0.3333	0.3636	0.1074	0.1894
Only RNA variables	0.7368	0.6667	0.3333	0.4444	0.2963	0.4001
Best 3 BDI variables	0.7368	0.5714	0.6667	0.6154	0.4172	0.4272
BDI and RNA variables (no normalization)	0.7895	0.7500	0.5000	0.6000	0.4648	0.6521
BDI and RNA variables (all normalized)	0.8947	0.7500	1.0000	0.8571	0.7765	0.8664
BDI and RNA variables (RNA normalized)	0.9474	0.8571	1.0000	0.9231	0.8834	0.8739

2.4 Protein–target identification from computationally designed small–molecles for castration resistant prostate cancer treatment

2.4.1 Abstract

Drug resistance is a widespread problem in cancer therapy due to the heterogenetic nature of cancer signaling pathways and networks. Traditional cancer therapies are developed through a single target approach. . This single-target therapy can get rid of the major mass of the tumor by targeting a particular receptor dominant clone. However, it will let the minor populations continue to grow and become resistant to the treatment. The ideal resolution to this problem is to develop a multi-target drug which can inhibit all survival pathways of cancer. Unfortunately, the development of multitarget inhibitors is difficult because, to our knowledge, no method exists to predict the efficacy of a small molecule using multiple targets in a given protein network and identify the interactions responsible for its efficacy. In this work, we have presented an iterative machine learning method to predict the efficacy of small molecules against castration-resistant prostate cancer (CRPC) and identify their mechanisms of action. After a set of *in vitro* biological testing, our approach has yielded a novel drug candidate, GCL.2, for CRPC treatment. GCL.2 has shown to successfully inhibit multiple interactions of a protein network complex of RORG, AKR1C4, CYP17A1, SHBG, and AR, which are crucial for CRPC clones. Furthermore, *in vivo* patient-derived models has shown that GCL.2 significantly inhibited tumor growth. With these results, we believe that our machine learning method can become essentially helpful on designing multi-target drug candidates for complex diseases as cancers.

2.4.2 Introduction

Multitarget drug design

The major alternative to single target drug design championed by our group is that of multitarget drug design [9,15,25,29,123,124]. The central hypothesis behind these works is that molecules which become drugs have multiple modes of action and do not function in isolation, a typical assumption in the traditional single-target approach. Using this hypothesis, we developed the Computational Analytics for Novel Drug Opportunities (CANDO) platform for repurposing drugs to utilize all the interactions between a drug and 46,784 proteins. We calculated proteome-wide interactions for 3733 human ingestible compounds and compared the signatures of these compounds to those of known treatments to postulate relationships in functional behavior. Compounds with similar signatures are presumed to be surrogates for each another in a disease-specific context. While this approach has seen wide success in the repurposing of known therapeutic [7,8], it is not applicable to the design of novel compounds. To fill this gap, we propose a new pipeline which combines traditional Computer Aided Drug Design (CADD) techniques with *in vitro* validation and machine learning to rationally design novel, multitarget, and nontoxic compounds for a specific disease. To do so, we have selected Castration-Resistant Prostate Cancer as a model disease for a multitargeted approach as it is traditionally associated with multiple, single-target therapies [125–129].

CRPC is a multitarget disease

Prostate cancer is the most common solid cancer and the second leading cause of death from cancer in men. In 2019, 174,650 new cases and 31,620 deaths are estimated in the US [130]. Although, 5-year survival rate of overall patients is 98%, the rate

dramatically declines when the progression stage is higher. Metastatic castration-resistant prostate cancer (mCRPC) is the main cause of death for prostate cancer patients. Only one-third of the metastasized prostate cancer patient will survive after 5-year of diagnosis. The median survival from CRPC diagnosis is only 14 months [131]. CRPC is defined as a stage that patients have a rising Prostate-Specific Antigen (PSA) in spite of medical or surgical castration [132]. For non-metastatic CRPC patients, the standard treatment is Apalutamide or Enzalutamide with continued androgen deprivation [132]. Both drugs specifically target Androgen Receptor (AR) and inhibit AR signaling, consequently, they can prevent the transcription of tumor genes. As the gold standard, both drugs have shown a significant increase of metastasis-free survival and time to PSA progression compared to a placebo group. However, half of the patients who received Enzalutamide developed resistance to the treatment within 37.2 months for PSA progression and progressed to mCRPC within 39.6 months after the treatment [133]. Likewise, the median metastasis-free period was 40.5 months for the patients who received Apalutamide.18 Furthermore, 24% and 10% of the patients did not respond to Enzalutamide or Apalutamide, respectively [132,133]. These data suggest that the single inhibition of AR activity is not sufficient to stop CRPC progression. There must be other players or mechanisms significantly involving in the resistance mechanisms aside from AR. For example, retinoic acid receptor-related orphan receptor gamma (ROR-γ), Sex hormone-binding globulin (SHBG), Cytochrome P17a1 (CYP17a1) and AR-V7 are widely known as the activators of AR expression and signaling in CRPC [125,129,134]. Therefore, CRPC is an apt disease to approach from a multitargeting perspective, especially given current opinion has suggested a combination therapy of several single target drugs to receive the maximum additive effects [135]. However, dose adjustment and side effects are the biggest concern for this approach. Therefore, developing a multi–target drug to interrupt the major

network proteins of CRPC progression seems to be the better strategy to cure this complex disease.

Traditional CADD

Modern drug design strategies often employ the use of computers to aid in the prediction and characterization of interactions a small molecule has on a biological target [136]. Of these techniques, docking is of particular interest as it can reproduce the 3D conformation of a small molecule in a binding pocket, predict the binding affinity of a small molecule, and serve as a method to virtually screen for active compounds against a single target [134, 137–143]. Given the success of docking for obtaining the interactions in the single target paradigm, we have employed it in the use of calculating the interactions a small molecule has with multiple proteins, yielding multiple interaction scores, referred to as an interaction signature.

Machine learning in chemistry

The integration of popular machine learning architectures into the drug design pipeline has seen wide spread adoption [144–146]. Chemists have applied Support Vector Machine [147–149], Random Forest [150–152], Multiple Layer Perception [153–155], Generalized Adversarial Networks [156–160], Recurrent Neural Networks [97, 161, 162], and one-shot learning techniques [163] to predict molecular properties and determine the characteristic features responsible for these classifications. Since the interactions between a small-molecule and protein targets are paramount in our approach, we choose the input feature space to reflect such interaction. Herein, we use docking scores as a surrogate for small-molecule protein interactions and train machine learning models to predict experimental results. Finally, analysis of these

models provides us with the targets responsible for compound activity which are verified *in vitro* to validate our hypothesis that the development of CRPC compounds is best tackled through a multitargeted approach. An overview of this work is shown in Fig 2.10.

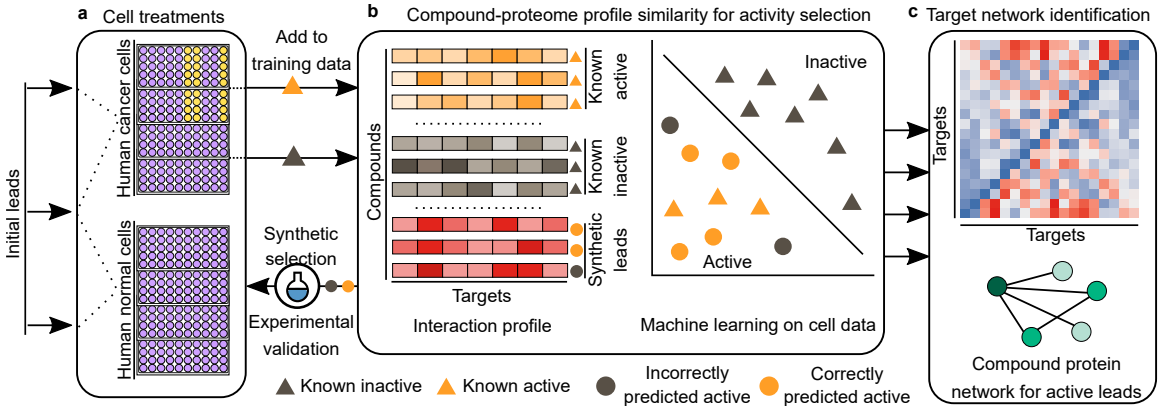


Figure 2.10. (a) Initial leads obtained from the CANDO platform are tested in vitro for both potency and toxicity on numerous human cell lines. Results of these tests label the compounds as either active (orange) or inactive (grey). (b) The interaction profiles of all compounds (both active and inactive) and additional "synthetic" compounds are calculated with targets of interest using CANDOCK (see Chapter3). These profiles are used as features in a SVM model to predict the activity of the untested molecules (shown as circles). Molecules predicted to be active are synthesized and test in vitro, resulting in additional active and inactive compounds to be used as training data in future SVM models. (c) The models generated from iterative machine learning (the combination of a and c) are analyzed to create compound-disease specific networks that identify which protein targets each compound interacts with to achieve its potency.

2.4.3 Results and discussion

CANDO yields initial compounds for prospective testing

We utilized the CANDO platform [9,15,25,29,123,124] to bootstrap our investigation of novel CRPC therapies as this platform provides a multitargeted approach to repurposing human ingestible compounds for any given disease. In the CANDO platform, there are 72 known treatments for Prostatic Neoplasms which are matched to 3733 other compounds present in the platform. To select potential new therapies for CRPC, we counted the number of times each of these 'other' compounds was predicted in the Top 100 compounds to treat Prostatic Neoplasms. The underlying theory behind this methodology is to select potential compounds which interact with multiple prostate cancer pathways because inhibiting more pathways yields a better potential for treating CRPC, a more aggressive form of the disease. Of all the predicted compounds for this disease, cinnarizine was predicted 31 times for prostatic neoplasms, the most out of any compound in the CANDO platform. The remaining compounds were filtered so that only compounds predicted more than 5 times remained, filtering the predictions down to 482 compounds and the compounds tibolone, norethisterone, levonorgestrel, cinnarizine, buspirone, talampicillin, azaperone, didanosine, pipamperone, and cetraxate were selected for further study. Of these compounds, tibolone, norethisterone, and levonorgestrel are steroids with minor structural differences as only the location of a double bond and the location of a methyl group in the steroid change between these compounds. These 10 drugs are our initial leads for developing new CRPC therapeutics using a multi–targeted approach.

Experimental *in vitro* testing of 10 compounds yields three active compounds

We tested all ten drugs discussed in the previous section using *in vitro* in both LNCaP and C4–2 cells to see their growth inhibition effect to identify which of these drugs were potential leads for CRPC. LNCaP and C4–2 cell lines were chosen to represent androgen-sensitive and androgen-independent prostate cancer cells, respectively. . In this experiment, both the LNCaP and C4–2 cells were treated for 6 days, and cellular proliferation was measured using Cell Titer-Blue Cell Viability Assay (Promega, Madison, WI). Among these ten predicted drugs, tibolone, norethisterone and levonorgestrel displayed promising growth inhibition with an IC_{50} of 24.86, 32.52 and 181.0 nM respectively in LNCaP cells and 3.12, 7.04 and 41.78 nM in C4–2 cells while for the other drugs this IC_{50} value was more than 5.0 μM (Fig 2.11a, Table E.1). Given the significant inhibition of C4–2 cell proliferation by these three drugs, we wished to ensure that these compounds were non-toxic using the standard MTT assay of RWPE-1 cells. The cytotoxicity IC_{50} of TIB, NOR and LEV were found to be 23.29, 86.30 and 59.70 μM respectively (Fig 2.11b). We investigated whether these initial leads reduce the amount of AR translated into the nucleus for LNCaP and C4–2 cells as translocation of this nuclear hormone receptor is known to cause proliferation in these cells. To measure whole cell expression of AR, we performed western blot analysis of LNCaP and C4–2 cells' lysates after the cells were treated with the three compounds. Fig 2.11c-d reveals that these compounds reduce the amount of AR in LNCaP cells by 25–45 % and by 15–25 % in C4–2 cells as compared to vehicle treatment. Given how the mechanism of AR proliferation is tied to the concentration of AR in the nucleus, we measured the amount of nuclear AR expression. To understand this, we performed anti-AR immunofluorescent staining of the LNCaP and C4–2 cells after fixation and permeabilization with Triton X-100,

followed by probing with monoclonal antibodies to AR(Fig 2.11e-f). Both norethisterone and levonorgestrel treatments lead to a respective 60 % and 30 % decrease in nuclear AR level in LNCaP and C4–2 cells. Using the growth inhibition profile of LNCaP and C4–2 cells, cytotoxicity profile of normal human RWPE-1 cells, western blot and immunofluorescence findings, we have identified TIB, NOR and LEV as the initial active leads for CRPC. Therefore, the CANDO approach has a prospective accuracy of 30% (3 active compounds out of 10) and we decided that this approach is not suitable for the development of new compounds.

Rational design and docking used to predict 6 additional compounds for testing prospectively

All three active compounds (tibolone, norethisterone, and levonorgestrel) are steroidal with similar scaffolds as all three compounds have a carbonyl group at the 3 position of the steroid ring and a methyl or ethyl groups at the 13 position, therefore we decided to use an approach which is specific to the chemical space of the active compounds. Using this approach, we created 50 novel designs (Table E.5) using the following medicinal chemistry principals: (i) hydroxyl group bio–isostere replacement (**1, 3, 5-8, 28** and **35**), (ii) carbonyl group bio isostere replacement (**12-20, 30-32, 40** and **42**), (iii) ethynyl group isostere replacement (**21-27**), (iv) carbon-carbon double bond isomerization (**2, 11, 29, 33, 34** and **50**), and (v) methyl and ethyl group isomerization (**4, 9, 10, 36-39, 41** and **43-49**). Since the number of drugs developed using this strategy is too large to test individually, we wished to use the activity results obtained from the 10 CANDO compounds to predict which designs would be active in a multitarget fashion. To do so, we decided to investigate the interactions these compounds have with multiple proteins instead of a single compound as is done in traditional drug design. We selected 18 targets which are known to play a role in

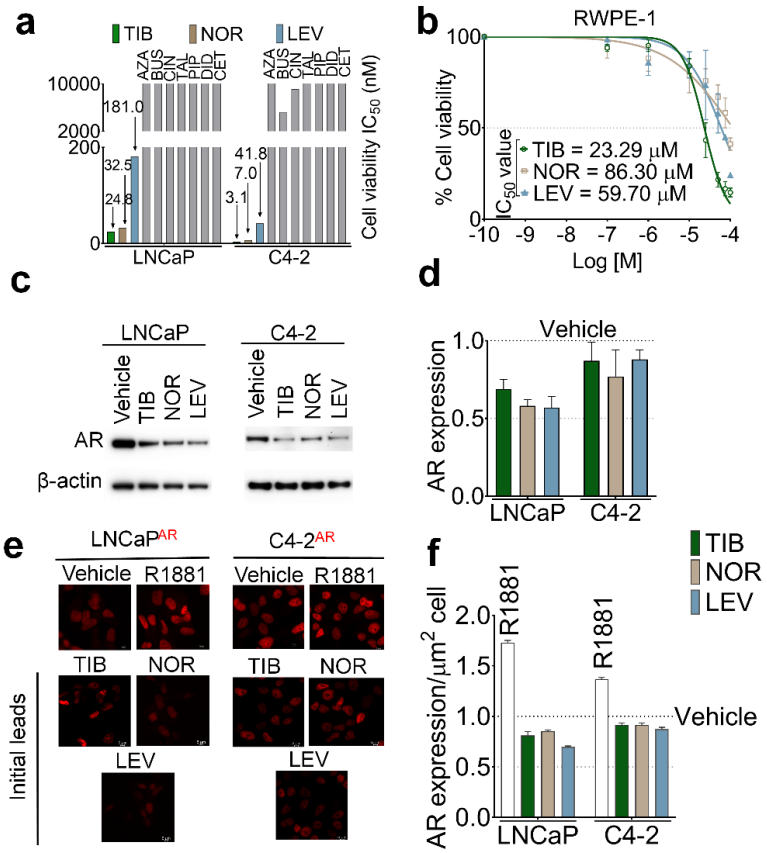


Figure 2.11. (a) Cell viability IC_{50} plots for the predicted drugs azaperone, buspirone, cinnarizine, talampicillin, pipamperone, cetraxate, didanosine, tibolone, norethisterone and levonorgestrel in human prostate cancer LNCaP and CRPC C4-2 cells. (b) IC_{50} graphs of the active drugs tibolone, norethisterone and levonorgestrel in normal human prostate epithelial RWPE-1 cell line. (c) Effect of the initial leads on the reduction of AR expression in LNCaP and C4-2 cells. Cells were treated with 1 μM concentrations of the indicated drugs/compounds or DMSO-growth media as vehicle control for 24 h and the expression of AR protein was analyzed from their lysates by western blot. Protein expression was normalized to β-actin (loading control) and densitometry was calculated using ImageJ Software. (d) AR expression in both LNCaP and C4-2 cells quantified from the western blots of (c). (e) Immunofluorescent staining of LNCaP and C4-2 cells for AR target after 24 h treatment with 1 μM of the indicated compounds. (f) Nuclear AR expression in both LNCaP and C4-2 cells quantified from the images of (e). Tibolone, norethisterone and levonorgestrel were newly identified as active and non – toxic initial leads for CRPC.

the proliferation of prostatic neoplasms (Table E.2, Fig 2.12a) and verified their role in the androgen pathway. Then, we docked all 60 compounds (10 CANDO compounds plus 50 designs) using our in–house docking software, CANDOCK v0.2.0 [143]. We have previously shown that our software provides accurate interaction scores for multiple small molecules which interact with multiple proteins [17]. The docked pose of the three active compounds with AR is shown along with corresponding CANDOCK binding scores in Fig 2.12b (all scores available in Tables E.3 and E.4). Using CANDOCK, we obtained the interaction scores for the 10 CANDO compounds (active shown in orange and inactive shown in grey in Fig 2.12c) and the 50 novel designs (Fig 2.12d) with the 18 protein targets, yielding an 18 protein target interaction signature for these compounds. Given the wide success of Support Vector Machine (SVM) classification in drug discovery and chemistry [149, 164], we applied this technique to predict which designs would be active given an interaction signature and the known activity of the 10 CANDO compounds. The number of predicted active compounds is quite large when using traditional SVM cut off values because all the compounds are steroidal and therefore have similar signatures as compared to non-steroidal compounds. This initial model performs well for separating the active and inactive CANDO compounds (Fig E.3, Table E.5), with the interesting exception of didanosine, which is typically predicted as active. To address this limitation and reduce the number of compounds required for synthesis and testing, we calculated a more restrictive cut off based on the decision values obtained from SVM learning (see Figure E.2) using the 90^{th} quantile of the decision values. Our methodology designated six designs, **2, 4, 11, 29, 40,** and **42**, as potentially active against CRPC cell lines (signatures displayed in Fig 2.12e). These predicted compounds and their relationship to the parent scaffold are given in Fig 2.12f.

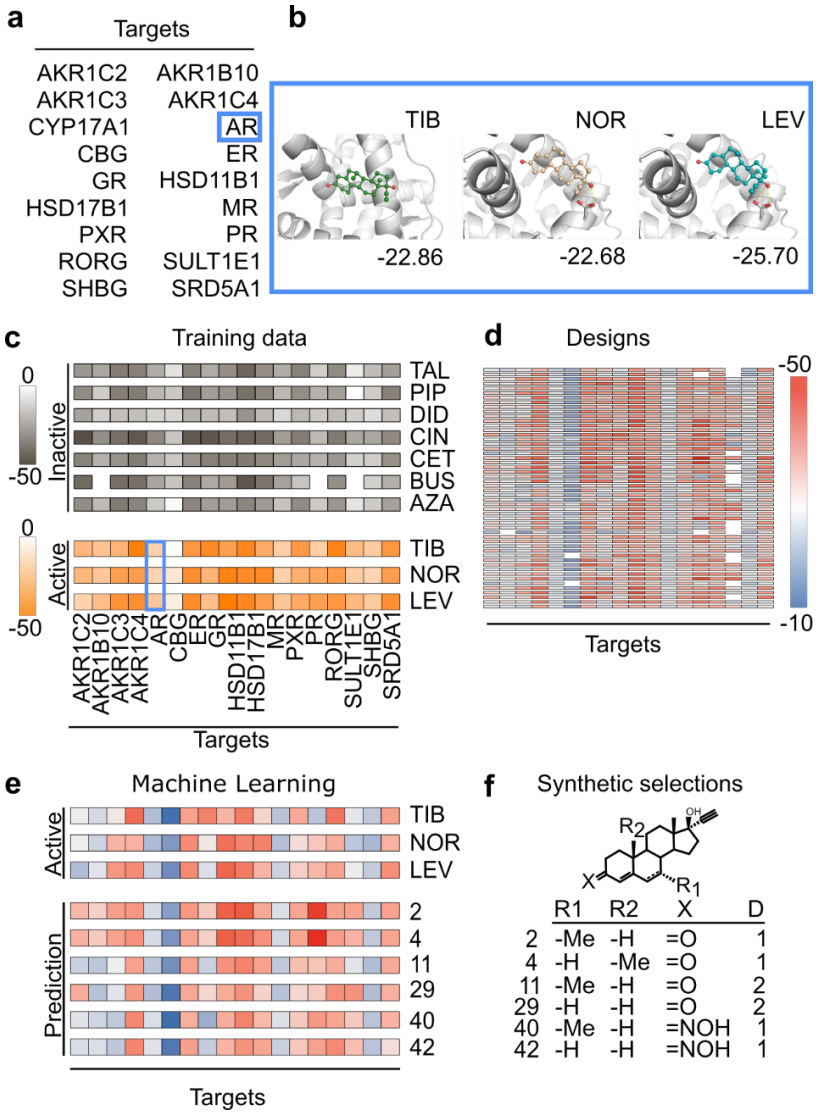


Figure 2.12. (a) List of CRPC targets used to create docking profiles for all compounds presented in this paper. (b) Docking pose of the initial leads in AR with their respective docking scores. (c) Docking profiles for all initial predictions used for training data (providing both positive and negative data in the form of active and inactive compounds respectively) in the prediction of new leads from isomeric designs. The docking scores for AR are highlighted in blue to demonstrate that this value alone is unable to produce a model capable of predicting activity against CRPC. (d) Docking profiles for novel designs. (e) Machine learning selection of profiles that match the profiles of the initial leads, leading to a new set of predictions for experimental verification. (f) Predicted actives after the first round of machine learning represented as modification of the original scaffold.

Prospective validation shows 4 out of 6 compounds are active We synthesized five putative active designs: **2, 11, 29, 40,** and **42** using the scheme shown in Fig 2.13a. Design **4** is ethisterone, a commercially available drug used to gynecological disorders. We measured the growth of LNCaP and C4–2 cells after treatment with all six of these compounds with CellTiter-Blue Cell Viability Assay (Promega, Madison, WI) or MTT (3-(4,5- dimethylthiazol-2-yl)-2,5-diphenyltetrazolium bromide) assay (Promega, Madison, WI). . Designs 2, 4, 40 and 42 all inhibited the proliferation of LNCaP cells with an IC_{50} of 5.6, 21.05, 62.08 and 62.83 nM, respectively, and C4–2 cells with an IC_{50} of 0.72, 11.01, 52.83 and 79.52 nM, respectively. However, the IC_{50} values of **11** and **29** are greater than 5.0 μM in both cell lines (Fig 2.13b). Given these results, we deemed designs 2, 4, 40, and 42 as active and designs 11 and 29 as inactive. The cytotoxicity IC_{50} of these molecules was found to be 54.67, 7.31, 7.05, 5.59, 13.88 and 21.36 μM in RWPE-1 cells for **2, 4, 11, 29, 40** and **42**, respectively (Fig 2.13c and S12c). Consequently, our initial machine learning model has a prospective precision of 0.67, an impressive result given the limited amount of training data available to the model. The receiver operator characteristic (ROC) and precision recall values plots are given in Fig E.3 and indicate that the model performs well as the area under the ROC curve is greater than 0.9 and the F1 score is greater than 0.8 (assuming a recall of 1.0). One explanation for this precision value is that the chemical space is limited to a that of a steroid with a limited number of modified functionalities. We plan to address the limitations of this approach in a future publication to allow for more complex molecules which allow for increased chemical diversity.

We were also interested to see the potency of our most potent lead 2 against the current CRPC drugs abiraterone (potent CYP17A1 inhibitor) and enzalutamide (potent AR antagonist). Since these drugs target two different proteins, we also tested

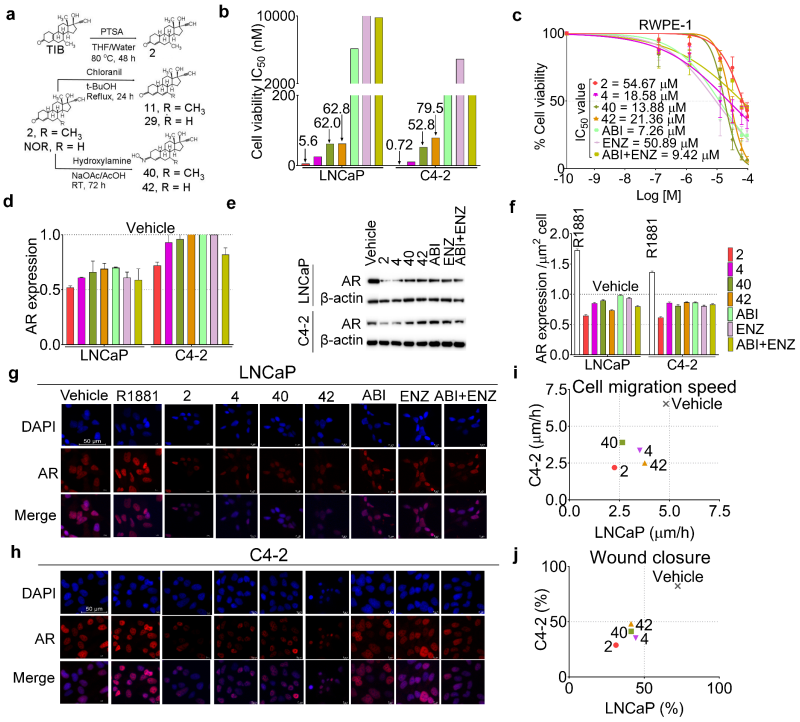


Figure 2.13. (a) Synthetic scheme for 2, 4, 40 and 42 from the initial leads. (b) IC_{50} plots of 2, 4, 40, 42, ABI, ENZ and ABI+ENZ against LNCaP and C4-2 cancer cell lines. (c) IC_{50} graph of 2, 4, 40, 42, ABI, ENZ and ABI+ENZ against RWPE-1 normal cell line. (d) and (e) Western blot analysis for AR and β-actin (loading control) in LNCaP and C4-2 cells treated with Vehicle and 1 μM concentration of the indicated compounds for 24h. Protein expression was normalized to β-actin and densitometry was calculated using ImageJ Softwar. (f) Nuclear AR expression in LNCaP and C4-2 cells after treated with the indicated drugs/compounds. (g) and (h) are respective Immunofluorescent staining of of nucleus (DAPI and AR in LNCaP and C4-2 cells treated with Vehicle and 1 μM concentration of the indicated compounds for 24h. (i) and (j) are respective migration speed and wound closure rate in both LNCaP and C4-2 cells in presence of the **2, 4, 40** and **42** designs. Synthetic lead 2 was found to be more potent in inhibiting the viability of LNCaP and C4-2 cells, less toxic in normal human epithelial RWPE-1 cells than the other leads as well as known CRPC drugs.

a combination of both to see if a single compound can be potent than a combination. A proliferation assay of LNCaP and C4–2 cells was performed in presence of design 2, abiraterone, enzalutamide, and a 1:1 combination of abiraterone and enzalutamide. The proliferation IC_{50} of design 2 (5.65 nM) was found to be significantly less than that of the individual CRPC drugs abiraterone (6166.0 nM) and enzalutamide (> 10000.0 nM) as well as their 1:1 combination treatment (> 7566.0 nM) in LNCaP cells. Additionally, the proliferation IC_{50} of design 2 (0.72 nM) was found to be less than that of the individual CRPC drugs abiraterone (291.0 nM) and enzalutamide (4922.0 nM) as well as their 1:1 combination treatment (782.1 nM) in C4–2 cells. The cytotoxicity IC_{50} of design 2 (Please add IC_{50} here) was found to be higher than abiraterone (3.80 μM) and enzalutamide (50.89 μM) and 1:1 combination of abiraterone and enzalutamide (9.42 μM) in RWPE-1 cells. These results clearly indicated not only the enhanced potency of 2 as compared to the known CRPC drugs but also its differential targeting action (CYP17A1 inhibition and AR antagonistic effect).

With the new active and inactive designs, we ensured that none of the 50 designed compounds would be predicted as active after including these findings in a new machine learning model. A retrained machine learning model with 16 observations instead of the original 10 produced similar decision values for the 34 untested compounds (Fig E.4a). Using the 90th quantile indicated that none of the remaining compounds need to be tested (Fig E.4b) and we therefore concluded that no additional compounds needed to be tested for activity in LNCaP and C4–2 cell lines.

Machine learning used to identify targets

After identifying 2 as most potent and non-toxic synthetic lead for CRPC, we investigated its mechanism of action to ensure that it is active against multiple CRPC

targets. Since the AR signaling pathway is known to be paramount for CRPC progression, we treated androgen independent human prostate cancer PC–3 cells and other human cancer cells such as non-small cell lung cancer H460 cells, neuroblastoma SHSY–5Y cells and bladder cancer HTB-9 cells with the four active designs **2, 4, 40**, and **42**. None of these desings displayed activity ($IC_{50} > \mu M$) on the androgen independent cell lines, indicating that these designs are specific to cell lines which are androgen dependant, specifically LNCap and C4–2. Since the designs are more potent than a known AR inhibitor(ENZ), a known CYP17A1 inhibitor (ABI) and a combination of the two (AR+CYP17A1), we wished to identify any additional targets inhibited by our designs. In order to identify other proteins, we performed computational studies of these synthetic leads against the 18 targets used to build the SVM model. Due to the success of SVM classification for resolving active vs inactive relationships, we began this analysis by creating a new SVM model using active and inactive steroidal compounds previously identified in this work. We hypothesized that the feature independence in the steroidal SVM model is paramount for determining the importance of a given feature in the disease network. To test this, we calculated the correlation between all 18 features used to create the SVM model and used these correlations to determine to most independent features of our model. The ranked order of all features is shown in Fig 2.14a (detailed calculations are given in Fig E.5).

Now that we have obtained a ranked list of features, we created SVM models that only consider a subset of the original 18 features while producing the same result as the complete model. The predictive capability of these smaller models, as compared to the larger 18 feature model, is maintained because they only contain features important for activity against CRPC. Although AR activity is generally considered paramount for treat CRPC, it is not the most important feature for classifying a compound as active or inactive. Instead, RORG, AKR1C4, and CYP17A1 are the

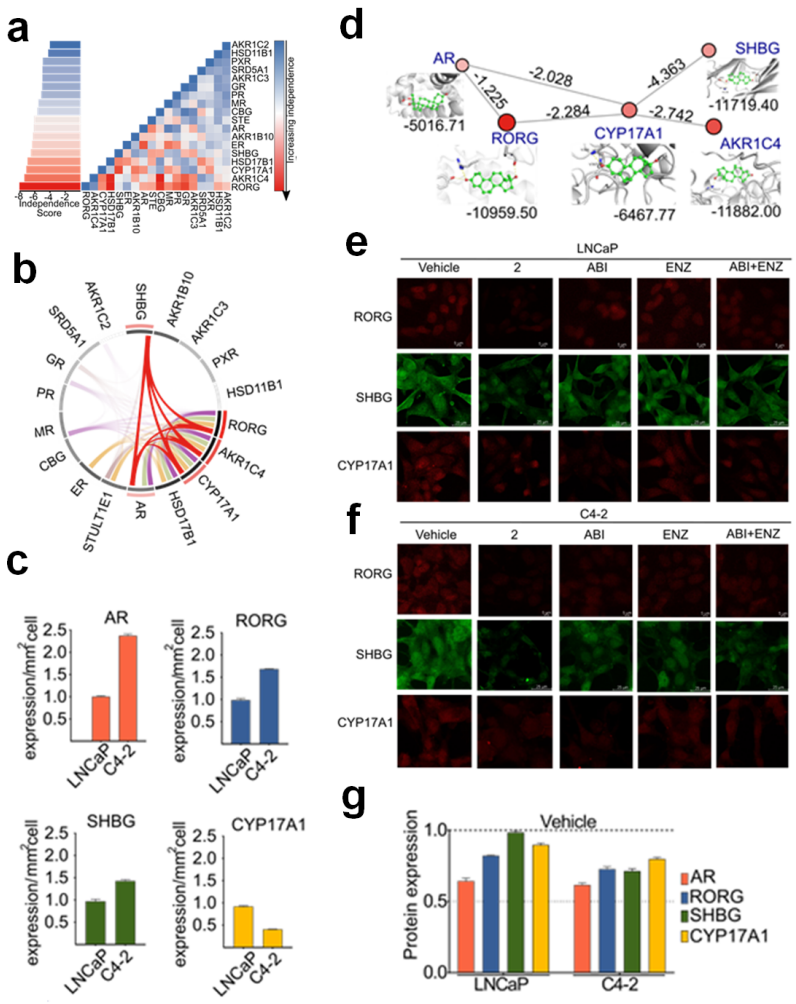


Figure 2.14. (a) Features ranked in the order of increasing independence (top to bottom) as calculated from the correlation matrix from the second round of machine learning. (b) Compound specific networks created from using the most independent features and keeping the prediction value of its compound paramount during the remodeling process. (c) Relative expression of AR, RORG, SHBG and CYP17A1 in both LNCaP and C4-2 cells. (d). Computational data showing differential targeting (RORG, SHBG, CYP17A1, AKR1C4 and AR) network proteins for the potent lead 2. (e) and (f) Respective immunofluorescent staining of LNCaP and C4-2 cells for RORG, SHBG and CYP17A1 proteome targets after 24 h treatment with 1 μM of the indicated compounds. (g) Expression of AR, RORG, SHBG and CYP17A1 in both LNCaP and C4-2 cells after treated with potent lead **2**.

most important features for determining the activity of our compounds because these features are both the most independent features from machine learning and are all required in our smaller networks to properly predict activity (Figs E.6, E, E, E, E, E, and E,). Therefore, we hypothesised that activity against these targets is important for activity against CRPC.

Validation of predicted targets

After identifying a compound specific network for AR signaling pathway (Fig 2.14b) for compound 2, we investigated RORG, SHBG and CYP17A1 targets for network validation by immunofluorescence assay (Fig 2.14e-d). To investigate the effect of other synthetic leads on the expression of RORG, SHBG and CYP17A1, we performed immunofluorescence staining of the LNCaP and C4–2 cells after treated with the indicated compounds/drugs (Fig 2.15). All the treatments led to a decrease in nuclear RORG level in both the LNCaP and C4–2 cells to an extent of 10-40% and 2 was found to be most potent (approximate 40% reduction of RORG expression) than the other treatment and known drugs. We also observed that, all these treatments displayed 10-25% reduction of whole cell SHBG level in both the cells and 2 was most potent among them as displayed by approximate 25% decrease. Similarly, all these treatments reduced 10-45% level of CYP17A1 in both the cells and 2 was found to moderately degrade CYP17A1. Thus, our immunofluorescence assays not only validated the four targets (AR, RORG, SHBG and CYP17A1) out of the five-target network identified for these leads, but also revealed 2 as the most potent lead among the synthetic compounds for CRPC. Also, lead 2 was found to be more potent as compared to that of the parent leads TIB, NOR and LEV which showed only 10-20% reduction of the identified network proteins (RORG, SHBG, CYP17A1 and AR). To investigate the enhanced potency of our synthetic lead 2 in CRPC C4–2 cells (0.72

nM) as compared to that of the normal prostate cancer LNCaP cells (5.65 nM), we analyzed relative expression of the network proteins in C4–2 and LNCaP cells by immunofluorescence assay. We observed 1.5 to 2.5 fold more expression of the network proteins (RORG, SHBG and AR) in C4–2 as compared to that of LNCaP cells. Thus, the enhanced potency of our synthetic lead 2 in C4–2 over LNCaP cells (5.65 nM) could be attributed due to the more expression of the network targeted proteins in former than the latter cells. Thus, we demonstrated an exciting application of our well-developed computational methods to identify a potent disease specific network for our leads and validated that identified network by experimental findings.

Tumor growth inhibitory effect of Candidate 2 on patient – derived xenograft mouse model

After investigating the effects of all candidates to human cell lines, we found that Candidate **2** was the most potent molecule to reduce the expression of all network proteins related to CRPC. Then, we studied tumor growth inhibitory effect of Candidate **2** (10 mg/kg) on an animal model. To mimic human prostate cancer progression, LuCaP35 xenograft mouse model was used. The drug treatment group showed significant inhibitory effect on the tumor growth compared to vehicle control group (Fig 2.16a). At the end of the treatment period, Candidate **2** suppressed tumor growth by 50.22%. In the meantime, Candidate **2** did not cause significant change on body weight profile (Fig 2.16b). The isolated tumors of Candidate **2** group were smaller in size (Fig 2.16c). and had 57.79% lower mass (Fig 2.16d) compared to the control group. In addition, Candidate **2** treatment did not cause any changes in vital organ mass (Fig 2.16e). The data suggested that 10 mg/kg of Candidate **2** significantly inhibited tumor growth on LuCaP35 xenograft mouse model without any signs of toxicity. The efficiency of this candidate was more than 50% suppression of both

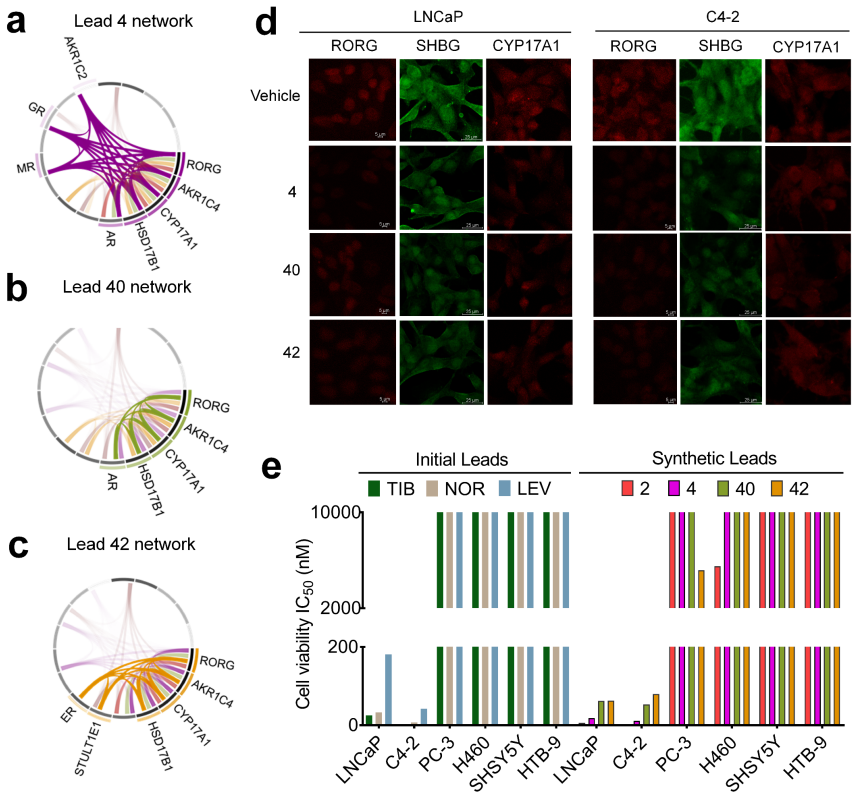


Figure 2.15. (a) and (b) are respective Immunofluorescent staining of LNCaP and C4-2 cells for RORG, SHBG and CYP17A1 proteome targets after 24 h treatment with 1 μM of the indicated compounds. (c) and (d) are respective expression of RORG, SHBG and CYP17A1 in both LNCaP and C4-2 cells. Synthetic lead 2 was most effective in degrading the network proteins in both LNCaP and C4-2 cells.

tumor volume and mass. This confirmed our hypothesis on the prediction by our machine learning model as well as the above *in vitro* experiments.

2.4.4 Conclusion

In this work, we have demonstrated that the multi–targeted hypothesis can be applied to design novel compounds against CRPC. Using the measured activity of compounds against this disease and the docking scores of these molecules with 18

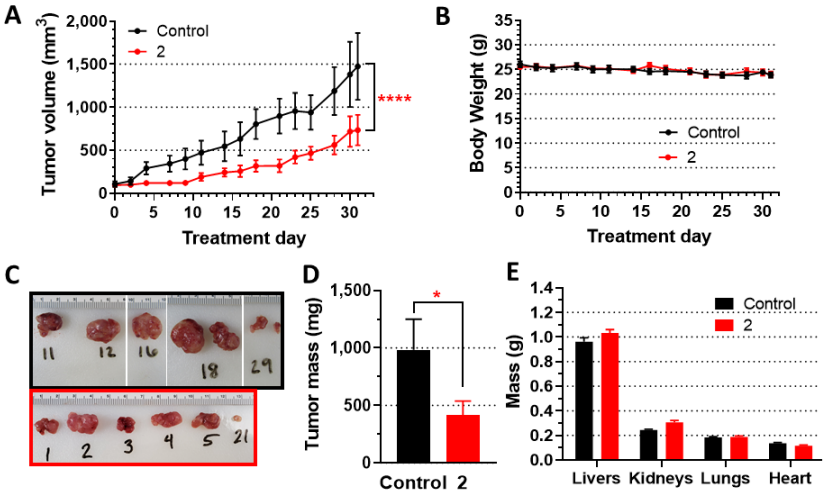


Figure 2.16. Tumor inhibitory effect of Candidate **2** compared to vehicle control on in vivo LuCaP xenograft model. (A) Tumor growth profile and (B) body weight profile of daily (M-F) oral administration of vehicle control (black) or of 10 mg/kg of Candidate **2** (red). After that, the mice were sacrificed. (C) Isolated tumors of vehicle control group (black frame) and Candidate **2** group (red frame). Mass of (D) the isolated tumors and (E) isolated major internal organs of the vehicle control group (black) and the control group (red). The data were shown as mean ± SEM. The statistical significance was indicated as *: $p < 0.05$ and ***: $p < 0.001$ between two groups. Credit: Asarasin Adulnirath

known CRPC targets, we have developed a machine learning model to select similar steroidal designs for potency against CRPC with a prospective accuracy of 66%. The compounds predicted by this model are more potent than a combination of known prostate cancer therapeutics which target proteins in the AR signaling pathway, indicating that these compounds target additional proteins in this pathway. We used our machine learning model to identify which proteins contribute to whether a compound will be active against CRPC and verified these proteins experimentally. The methods described in this work indicate that drug discovery can be performed in a

multi–targeted manner, opening a way for this paradigm to be further embraced in future studies.

2.5 Application to cells with unknown pathways

While the above methodology will work for cells with known pathways, there is no methodology known to target cellular function without target information. An example of such a cell type is myeloid-derived suppressor cells with no lineage defining transcription factor to target. Details of this approach can be found in the Ph.D. dissertation of Dr. Erin Kischuk, but they will be repeated in brief here. An overview of this approach can be found in Fig 2.17.

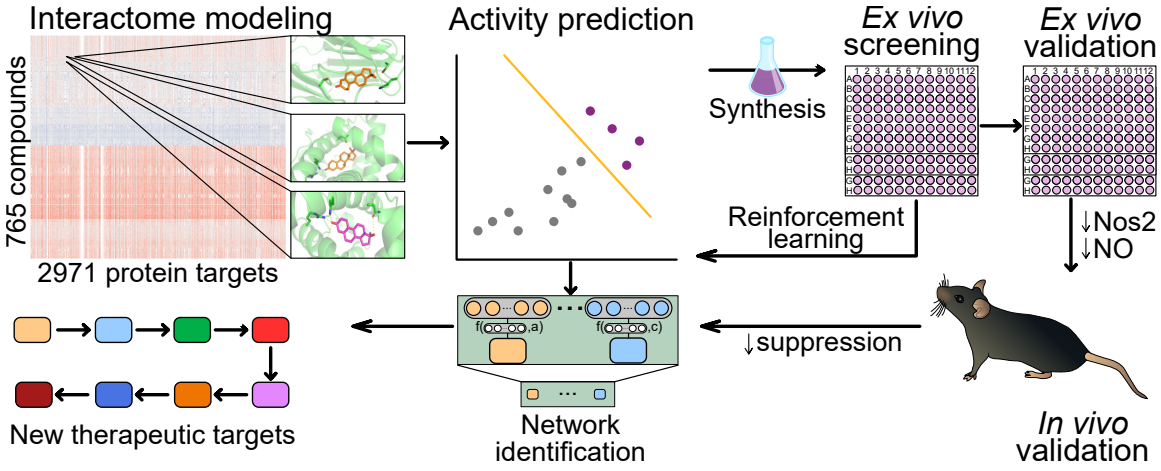


Figure 2.17. Overview of the approach used to develop new compounds for changing the function of cells where the pathways are unknown.

This process involves the iterative creation of models using novel compounds which are predicted by the previous model. This is shown in Fig 2.18. Each time the model was trained with 2987 proteins and their interactions with the training molecules. The molecules predicted to be active are then tested *in vivo* and the model is retrained.

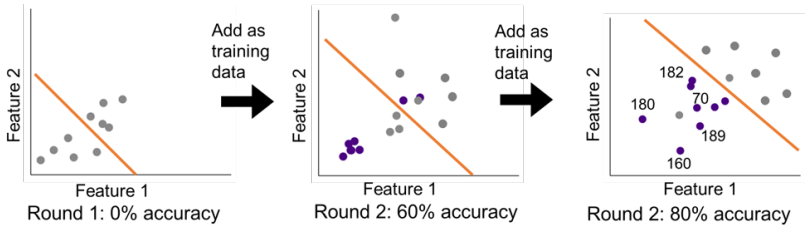


Figure 2.18. Iterative training and validation of cell specific models.

After each retraining, the accuracy of the model improves as more data is given for training.

3. SMALL-MOLECULE INTERACTIONS WITH A PROTEIN

Reprinted with permission from Jonathan Fine, Janez Konc, Ram Samudrala, and Gaurav Chopra. CANDOCK: Chemical Atomic Network-Based Hierarchical Flexible Docking Algorithm Using Generalized Statistical Potentials. *Journal of Chemical Information and Modeling* **2020** 60 (3), 1509-1527. Copyright 2020 American Chemical Society.
DOI: 10.1021/acs.jcim.9b00686

Note that the text and some figures have been modified to suit the formatting of this document and the future work section is not part of the *JCIM* publication.

3.1 Abstract

Small–molecule docking has proven to be invaluable for drug design and discovery. However, existing docking methods have several limitations such as improper treatment of the interactions of essential components in the chemical environment of the binding pocket (e.g., cofactors, metal ions, etc.), incomplete sampling of chemically relevant ligand conformational space, and the inability to consistently correlate docking scores of the best binding pose with experimental binding affinities. We present CANDOCK, a novel docking algorithm, that utilizes a hierarchical approach to reconstruct ligands from an atomic grid using graph theory and generalized statistical potential functions to sample biologically relevant ligand conformations. Our algorithm accounts for protein flexibility, solvent, metal ions, and cofactor interactions in the binding pocket that are traditionally ignored by current methods. We evaluate the algorithm on the PDBbind, Astex, and PINC proteins to show its ability to reproduce the binding mode of the ligands that is independent of the initial ligand conformation

in these benchmarks. Finally, we identify the best selector and ranker potential functions such that the statistical score of the best selected docked pose correlates with the experimental binding affinities of the ligands for any given protein target. Our results indicate that CANDOCK is a generalized flexible docking method that addresses several limitations of current docking methods by considering all interactions in the chemical environment of a binding pocket for correlating the best-docked pose with biological activity. CANDOCK along with all structures and scripts used for benchmarking is available at https://github.com/chopralab/candock_benchmark.

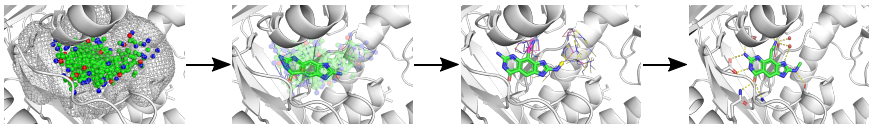


Figure 3.1. Table of contents figure for the online publication

3.2 Introduction

Computational docking provides a means to predict and assess interactions between ligands and proteins with relatively little investment. Docking refers to physical three-dimensional (3D) structural interactions between a receptor (typically proteins, DNA, RNA, etc.) and a ligand (small molecules, proteins, peptides, etc.) [3, 7, 8, 14, 136, 165–174]. Docking methods are evaluated by predicting the correct pose/binding mode (evaluated using root-mean-square deviation (RMSD) or TM-Score of the coordinates of the atoms) or by measuring predicted binding affinities [8, 166, 170, 171, 175]. Application to protein targets involved in disease holds the promise of discovering new therapeutics using traditional single target approaches or by virtually measuring the interactions of a compound with the proteins from multi–organism proteomes [9, 15, 25, 29]. The resulting chemoproteome interactions can be

interrogated to study polypharmacology [25] and investigate the effect that drugs and agents have on protein classes in a disease-specific context [25, 123]. In previous works, we have used the algorithm presented herein to combat Ebola [29], determine the toxicity of potential diabetes therapeutics [17], and rank the affinity of kinase inhibitors for the treatment of acute myeloid leukemia [16].

More than 20 molecular docking software tools, such as Autodock Vina [176], Gold [177], MedusaDock [178–180], and Glide [165], are currently in use for pharmaceutical research. However, after decades of method development and application, the promise to computationally determine new therapeutics has not been fully realized and computational methods for drug discovery are still in its infancy [181, 182]. The CANDOCK algorithm confronts several outstanding technical and practical problems in computational docking. For example, one significant problem is assessing goodness-of-fit or the likelihood that the given pose is the most physically realistic (native-like) pose among many unrealistic binding poses. Another significant limitation is the lack of full protein flexibility in the docking methods used today. The induced fit is a widely recognized challenge in computational drug screening [141, 179, 180], where the protein and the ligand undergo conformational changes upon ligand binding. Therefore, the traditional treatment of proteins as rigid structures may be insufficient and often misleading for structure-guided drug screening and design, as shown by us and others previously [30]. Docking ligands to their protein targets is particularly challenging when attempting to reproduce the binding mode of small molecules to ligand- free or alternative ligand-bound protein structures, which invariably occurs for practical application of any docking method. Specifically, docking with ligand-bound (holo) protein structures typically leads to an accuracy of 60–80%, whereas ligand–free (apo) structures yield a docking accuracy of merely 20–40% [138, 173, 183–185].

Several methods have been implemented to account for protein and ligand flexibilities, including multiple experimentally derived structures from X-ray crystallography [186], nuclear magnetic resonance [186], rotamer libraries [179, 187], Monte Carlo [176, 188], and molecular mechanics [99, 189–193]. The same principle limits the use of multiple experimentally derived protein structures or side-chain rotamer libraries: binding a ligand to a protein can cause conformational changes in either molecule that are not captured by these methods [194]. The sampling problem is compounded by the fact that the protein main–chain torsion angles are also frequently altered from their ligand-free conformations, which these methods fail to capture. Molecular mechanics is well suited for capturing fine detail side–chain and main–chain motions and rearrangements through energy minimization. However, molecular mechanics is limited in that adequate sampling of all degrees of freedom between the protein and ligand–rotation, translation, and torsion angle – is frequently computationally intractable. Further, the use of unrestrained molecular dynamics has been shown to disrupt the ligand from its native pose [139].

Modern docking methods address these issues by employing algorithms such as the genetic algorithm [141, 177, 195, 196], to flexibly sample the conformational space. However, it has been shown that these methods do not consistently produce poses that rank the biological activity of the ligand well [196, 197], and that the ability of these methods to produce a correct pose is dependent on the starting conformation of the ligand [198, 199]. Some methodologies use a fragment-based approach to docking to sample the conformational space for a given ligand efficiently [200]. These fragment-based methods have reported a greater ability to rank activity between the given ligands [201, 202]. Therefore, we believe that further innovation in fragment-based methods is an appropriate way to improve docking methods.

We have developed the CANDOCK algorithm around new protocol for hierarchical (atoms to fragments to molecules) docking with iterative dynamics during molecule reconstruction to "grow" the ligand in the binding pocket. The docking protocol is based on two guiding principles: (i) binding sites possess regions of both very high and very low structural stabilities [203] and (ii) a tandem sequence of small protein motions is generally sufficient to predict the correct binding mode of protein–ligand interactions [194]. The hierarchical nature of this method is derived from an "atoms-to-fragments", "fragments-to-ligands" approach that generates chemically relevant poses given the ligand and surrounding any chemical environment (e.g., protein, RNA, DNA binding sites, or interfaces). For any flexible ligand, the expectation is that at least one or a few fragment conformations assembled using ligand–receptor atomic interactions in the binding pocket will bind to a structurally stable region of the receptor. Following identification of such a binding mode, subtle conformational changes of the receptor are necessary for reconstructing the ligand using these fragments as "seeds" to generate accurate receptor–ligand binding modes (poses). We show that CANDOCK can accurately reproduce the binding mode of ligands and rank the activity of these ligands in such poses using a generalized statistically derived force field, demonstrating the potential to overcome traditional challenges with induced-fit docking methods.

3.3 Materials and methods

We first introduce our generalized statistical scoring function and then provide details of the CANDOCK algorithm and selection of benchmarking data sets for evaluating pose election and receptor–ligand affinity ranking.

3.3.1 Generalized Statistical Scoring Function

A generalized statistical scoring potential is used to account for varying chemical environments, such as metal ions, cofactors, and water molecules, and have shown great promise for selecting correct poses in both small-molecule and protein–protein docking [204]. The scoring function employed by the CANDOCK algorithm is a pairwise atomic scoring function that is based on our previous work [28]. Here, we reproduce the fundamental equations to clarify the terminology used in our manuscript. The scoring function calculates the potential between two atoms based on the distance between atoms i and j with atom types a and b and takes four input terms that determine the method by which the score is calculated. The possible terms are "functional", "reference", "composition", and "cutoff", which define the probability function P given in equation 3.1.

$$S\left(r_{ab}^{ij}\right) = -\sum_{ij} ln \frac{P(r_{ab}^{ij} \vee c)}{P(R^{ij})} \tag{3.1}$$

The "functional" term determines the numerator of equation 3.1 and can be defined as a "normalized frequency" function $f(r)$ in equation 3.2 where N_s is the number of observed atoms found at a given distance. Alternatively, it can be described as a "radial" distribution function $g(r)$ where N_s is divided by the volume of the sphere $V_s(r)$ which is described in equation 3.3. To distinguish between these two functions, "radial" scoring functions start with "R", while "normalized frequency" functions start with "F".

$$P(r_{ab}^{ij} \vee c) = f(r_{ab}) = \frac{N_s(r_{ab})}{\sum_r N_s(r_{ab})} \tag{3.2}$$

$$P(r_{ab}^{ij} \vee c) = g(r_{ab}) = \frac{\frac{N_s(r_{ab})}{V_s(r)}}{\sum_r N_s(r_{ab})} \tag{3.3}$$

The "reference" term determines the denominator of the scoring function. It can be defined either as "mean", in which case it is calculated as a sum of all atom-type pairs divided by the number of atom types, or as the "cumulative" sum of all atom-type pairs. The mean term can be used with either "normalized frequency" equation 3.4 or "radial" equation 3.5. The "cumulative" option can be used together with "normalized frequency" to equation 3.6 and "radial" to equation 3.7.

$$P(r) = f(r) = \frac{\sum_{ab} f(r_{ab})}{n} \tag{3.4}$$

$$P(r) = g(r) = \frac{\sum_{ab} g(r_{ab})}{n} \tag{3.5}$$

$$P(r) = f(r) = \frac{\sum_{ab} N_s(r_{ab})}{\sum_r \sum_{ab} N_s(r_{ab})} \tag{3.6}$$

$$P(r) = g(r) = \frac{\sum_{ab} \frac{N_s(r_{ab})}{V_s(r)}}{\sum_r \sum_{ab} \frac{N_s(r_{ab})}{V_s(r)}} \tag{3.7}$$

Scoring functions compiled with the "mean" option are denoted as "M", while those compiled with the "cumulative" are denoted as "C". The third term defines the composition of the scoring function. This term controls the number of unique atom pairs used for compiling the scoring function. The "complete" option will result in the scoring function compiled from all possible atom-type pairs, while the "reduced" option will only use the atom types present in either the protein (including cofactors, waters, and post-translational modifications) and ligand. The letter "C" is used to denote a *complete* scoring function, while "R" is used to denote a scoring function that is compiled with the "reduced" option. A total of eight scoring function families can be created with these three options (RMR, RMC, RCR, RCC, FMR,

FMC, FCR, FCC). The fourth and final term used to compile the scoring function is the "cutoff", which controls the maximum distance at which the interactions will be calculated, with possible values ranging from 4 to 15 Å. With all four options, there are a total of 96 possible scoring functions (8×12) to account for generalized parameters for identifying native poses and activity across a diverse set of biomolecular interactions in varying chemical environments (proteins, nucleic acids, interfaces, cofactors, etc.). Example scoring functions are "radial-mean-reduced-6" (RMR6), "normalized frequency-cumulative-complete-8" (FCC8), and so on, as denoted in the manuscript. It should be noted that not all 96 scoring functions are intended to be used for all docking simulation, and the selection of the appropriate scoring function for a given goal will be discussed in later sections.

3.3.2 Phase I: Structure Preparation.

The CANDOCK algorithm's input is a set of compounds to be docked, a query protein structure, and a set of binding sites on the query protein structure. In a three-phase protocol (Fig 3.2), it performs semi– or fully flexible docking of compounds to the protein and outputs docked and minimized protein–compound complex structures together with their predicted scores.

Parse Receptor and Compounds.

The inputs to the algorithm are the 3D coordinates and topology of a query receptor (e.g., protein structure) consisting of single or multiple chains, which may also contain cofactors and post-translation modifications in the protein data bank (PDB) format and compounds in the MOL2 format. Compounds are processed in

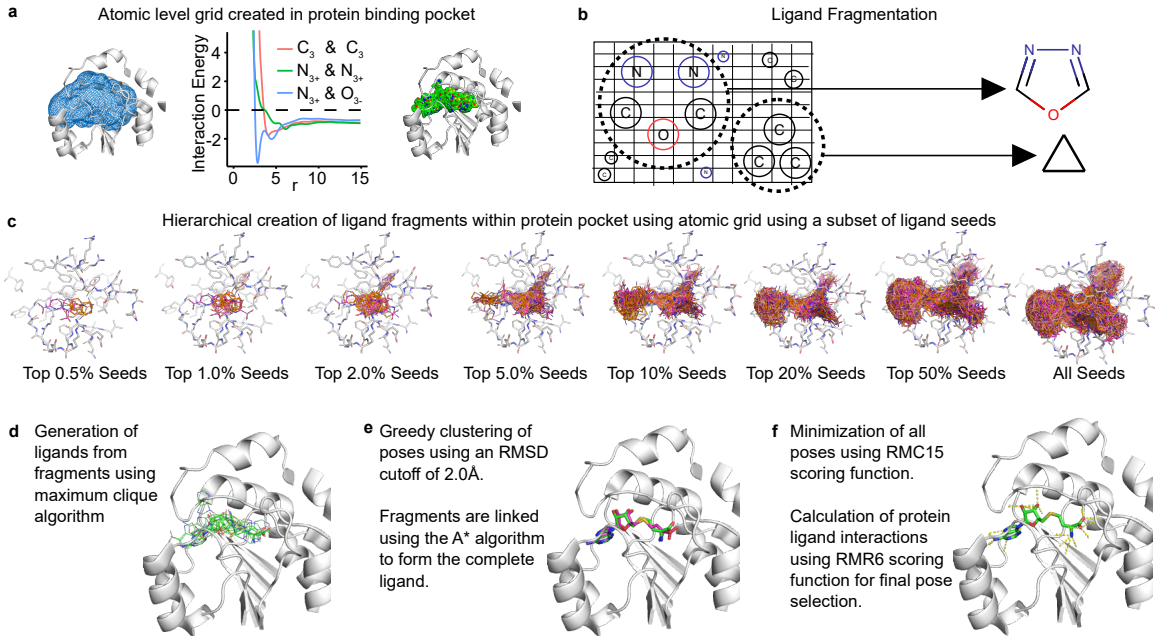


Figure 3.2. Overview of the CANDOCK docking algorithm. Phase I consists of processing the input protein (a) and the ligand (b). During Phase I, an atomic grid is created in the protein binding site, with scores of all possible atom types at each point in the binding site grid. Simultaneously, the input ligand(s) are fragmented along the rotatable bonds present in the ligand. The grid is used to recreate the rigid fragments in the binding pocket. Phase II constructs the rigid ligand fragments in the binding site grid producing "seeds" that can be grown into the full ligand (c). Phase III identifies potential ligand poses using maximum clique algorithm (d), clusters and links these poses using A* algorithm (e), and minimizes the poses into the binding site (f).

batches of size 10 to enable reading of large molecular files that do not fit in computer memory. An example of a ligand is given in Fig 3.3a.

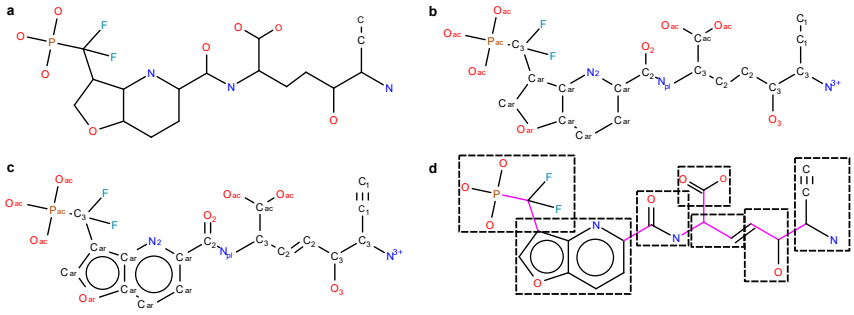


Figure 3.3. Atom–type assignment and fragmentation procedure in CANDOCK. The procedure begins with the topology and 3D coordinates of the ligand (a). Using these data, the IDATM type is assigned to each atom in the ligand using a previously described algorithm [205] (b). This yields the hybridization state of all atoms, allowing for the assignment of bond orders for all atoms (c). The bond orders and topologies are used to assign a rotatable flag for each bond in the ligand using rules derived from the DOCK 6 program [206]. The rigid fragments identified using this method are boxed (d).

Compute Atom Types.

To compute atom types for proteins, cofactors, and compounds, we implemented the IDATM algorithm [205] (results given in Fig 3.3b). We also implemented an algorithm [207, 208], to assign AMBER general force field (GAFF) atom types to cofactors, ligands, and post–translational modifications, while GAFF types for proteins are obtained from the AMBER10 topology file available as part of the OpenMM package [209].

Assignment of Bond Orders.

Using the hybridization information provided by the newly assigned IDATM atom types, several potential bond order states can be generated as to fit with the expected number of bonds (valence) for each ligand atom. These potential bond order assignments are evaluated in a trial-and-error fashion to determine whether they form a valid molecule using valence state rules derived for all atom types. The bond order set that satisfies the set of valence states with the lowest sum of atomic penalty scores over all atoms (see Fig 3.3c) is used to assign GAFF bond orders of the ligand.

Fragment Compounds.

Rotatable bonds are first identified in each compound using the extended list of rotatable bonds adapted from the UCSF DOCK 6 software [206]. Next, structurally rigid fragments consisting of atoms between the rotatable bonds are identified. Bond vectors for rotatable bonds are retained for each rigid fragment to be used during the reconstruction of docked fragments. Fragments consisting of more than four atoms, in which at least two atoms are rigid (connected by a nonrotatable bond), are considered as seed fragments. These are subsequently rigidly docked into the protein binding site. All other nonseed fragments are considered as linking fragments during the compound reconstruction process. This result is shown in Fig 3.3d.

Assignment of Force Field Atom Types.

Using the computed GAFF atom types, the bonded forces of the AMBER force field are generated for the protein and the docked compounds. Protein–compound interactions are scored using the knowledge-based radial-mean-reduced (RMR) discriminatory function defined previously [28] with a 6 Å cutoff. This function calculates

a fitness score for each compound's or fragment's atom in a protein by considering all protein atoms within a 6 Å radius of that atom. It is an atomic level radial distribution function with mean reference state that averages over all pairwise atom types from a reduced atom-type composition (protein's and compound's atom types), using experimentally determined intermolecular complexes in the Cambridge Structural Database (CSD) [210] and in the Protein Data Bank (PDB) [95] as the information sources. The objective function that is used for the minimization of the protein–compound interactions is computed using the RMC scoring function with a 15 Å cutoff as follows: for each possible pair of atom types present in the protein–ligand complex, the RMC function is sampled at discrete 0.1 Å intervals and is smoothed using B-spline interpolation. Potential energy values and their first derivatives are calculated at 0.01 Å intervals over the [0, 15] Å interval for the smoothed function. The objective function is implemented as a custom knowledge-based force object in OpenMM [209], which is used as a library from the CANDOCK source code.

Prepare Protein for Molecular Mechanics.

The N- and C-terminal residues are renamed according to the AMBER topology specification, e.g., ALA to NALA or CALA; disulfide bonds are added to the protein by connection of SG atoms that are closer than 2.5 Å, and inter-residue bonds are also added by connection of main-chain C and N atoms that are closer than 1.4 Å.

3.3.3 Phase II: Rigid Fragment Docking.

Compute Rotations of Seeds.

For each seed fragment, we compute its rotational transformations about the geometric center, which is fixed at the coordinate origin. Accordingly, we first compute

uniformly distributed unit vectors around the coordinate origin. Then, the seed fragment is rotated by 10° increments around the axis formed by each unit vector. To speed up the subsequent step of rigid fragment docking, the rotated fragment atoms' coordinates are mapped on a hexagonal close-packed (HCP) grid of 0.375 Å resolution. This mapping enables efficient docking of fragments to a protein binding site since their rotational transformations need to be computed only once. The fragment's clashes with the protein and the fragment's RMR6 scores are determined by translations of the rotational fragment grid over the compatible HCP binding site grid using fast integer arithmetic.

Generate Binding Site Grid.

A binding site location for docking is specified using one or more centroids, each consisting of the Cartesian coordinate of its center and its radius. We generate a grid that covers the space of all centroids that represent the binding site (Fig 3.4a). We use an HCP grid that provides maximal packing efficiency, covering the same volumetric space of a simple cubic grid with approximately 40% fewer grid points to achieve the same maximal interstitial spacing. The grid points are in a distance range of $0.8 < d < 8$ Å from any protein atom. We use a grid spacing of 0.375 Å with a maximal interstitial spacing of 0.22 Å to densely represent the protein binding sites (Fig 3.4b).

Dock and Cluster Rigid Fragments.

Intermolecular geometric and chemical complementarity between a protein and a ligand is essential for binding. Energetically preferred positions of ligand atom types can be captured using a discriminatory function (Fig 3.4c). Docking of seed frag-

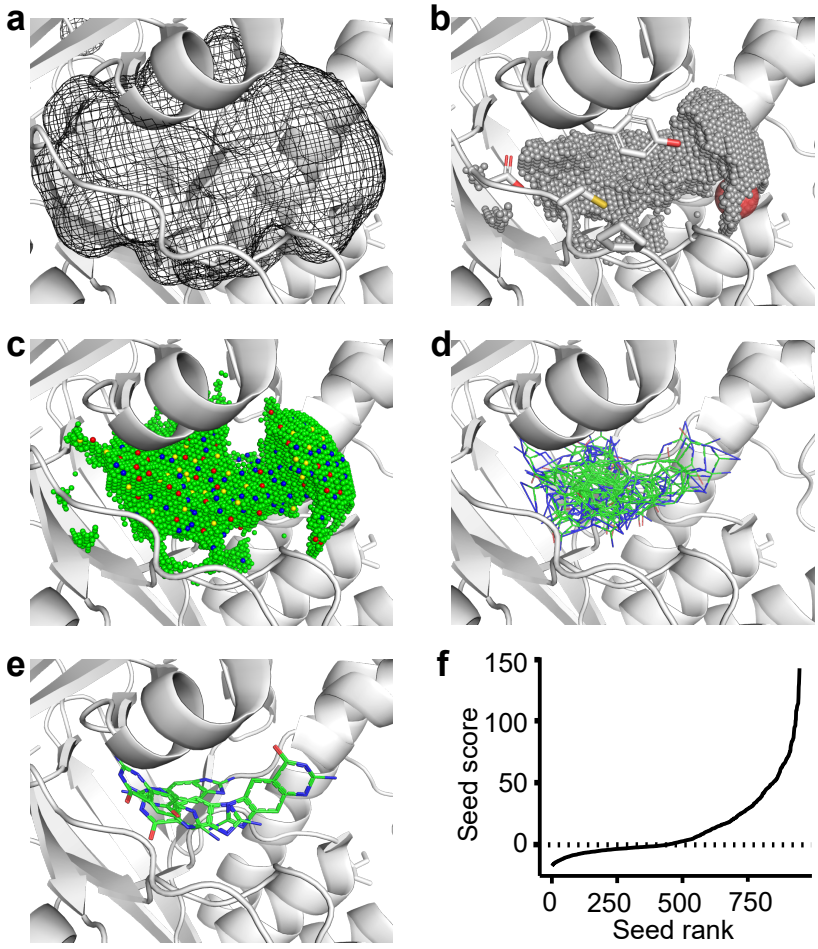


Figure 3.4. Detailed overview of the hierarchical relationship between the atomic grid and ligand fragments. The protein binding site is supplied as a series of centroids to form the binding pocket (a). Regions of this volume that do not clash with receptor atoms are filled with an HCP grid (b). The RMR6 score of all atom types present in the ligand is calculated. (c). Ligand fragments from the previous step are translated and rotated within this grid (d). This collection of ligand fragments is clustered using a greedy clustering algorithm using RMSD fragment similarity. If two fragments are within 2.0 Å RMSD of each other, the fragment with a higher RMR6 score is deleted and remaining docked fragments are kept as seeds (e). The exponential score distribution of a typical seed is given in (f).

ments to the binding site grid is performed by moving the seed's rotational grid over the binding site grid points. Docked fragment poses that are in a steric clash with

the protein are rejected (Fig 3.4d). A steric clash is considered if any interatomic distance between the fragment and the protein falls within nine-tenths of the atoms' respective van der Waals sum. Each fragment translation and rotation that passes this initial filter is then evaluated with the RMR6 discriminatory function [28]. Finally, greedy clustering of docked and scored fragment poses in the root-mean-square deviation (RMSD) space computed based on their heavy atoms at 2 Å cluster cutoff is performed, resulting in a uniform distribution of locally best-scoring docked seed fragments covering the entire protein binding site (Fig 3.4e).

3.3.4 Phase III: Flexible Docking with Iterative Minimization

Generate Partial Compound Conformations

For each compound to be docked, a user-specified percentage of each of its best–scoring rigidly docked seed fragment poses is considered. Among these, we search for such compatible pairs of docked seeds that are at appropriate distances, that is, the distance between them is less than the maximum of their known bond distance. The maximum possible distance between a pair of seeds is calculated by traversing the path between the fragments in the original compound and summing up the distances between the end points of each rigid fragment on the path. We construct an undirected graph in which vertices represent seed fragments, and edges indicate that the corresponding pair of seed fragments is linkable. Using the MaxCliqueDyn algorithm [211], we then find all fully connected subgraphs consisting of k vertices (k-cliques) in this graph, where the default value of k is set to three or to the number of seed fragments, whichever value is less. Each k-clique corresponds to a possible partial conformation of the docked seed fragments, in which these fragments are appropriately distanced so that they may be linked into the original compound. The

maximal clique algorithm of Bron and Kerbosch [212], which was previously used for pose matching [213], differs significantly from our maximum clique algorithm [211]. While a maximal search covers all cliques that are not subgraphs of another clique, maximum clique algorithms only search for the clique with the maximum number of vertices. Consequently, although both address an NP-hard problem, finding a maximum clique requires an order of magnitude less computing time. The possible partial conformations are then clustered using a greedy clustering algorithm at an RMSD cutoff of 2 Å, where the best-scored cluster representatives are retained. The partial conformations sorted by their RMR6 scores from the best- to the worst-scored are used as an input to the next step of compound reconstruction.

Reconstruct Compound with Protein Flexibility.

Each identified partial conformation of the docked seed fragments is gradually grown into the original ligand by the addition of nonseed fragments using the A* search algorithm. This can be done at different levels of protein flexibility. Protein minimization may be performed at each step of the linking process or only at the end when the compound has been reconstructed. Each seed fragment is linked to adjoining fragments according to the connectivity of the original compound. Each added nonseed fragment is rotated 360° about the bond vector at 60° increments. If the user has specified full protein flexibility, the resulting conformation of the partial compound and the protein is subjected to knowledge-based energy minimization using the RMC15 scoring function as for intermolecular forces. Simultaneously, bonds, angles, and torsions of the partial compound and the protein are minimized using the standard AMBER molecular mechanics energy minimization. This procedure uses the popular OpenMM software package, specifically its implementation of the L-BFGS minimization algorithm [214]. With each round of minimization, the RMR6 score

is calculated for the protein–compound interactions and the scored conformation is added to the priority queue, which consists of the growing compound conformations in the order from the best-scored to the worst-scored.

At each subsequent step of reconstruction, the A* search algorithm chooses the best-scored conformation from this priority queue and attempts to extend it. This conformation must meet an additional condition, which is that its attachment atoms that are to be connected by rotatable bonds to fragments not yet added need to be at appropriate distances from the attachment atoms on the remaining seed fragments. The algorithm iterates until the priority queue is empty, in which case the compound has been completely reconstructed and is in a local minimum energy state. Alternatively, if the specified maximum number of steps was exceeded (1000 by default), then the reconstruction failed. The A* search is repeated for each partial conformation of docked seed fragments until all have been considered for reconstruction into a differently docked conformation of the original compound. A final energy minimization procedure is performed on the protein–ligand complex treating the protein as fully flexible (side chain and backbone) to remove steric clashes in the process of growing the ligand into the binding site. In addition to knowledge-based and molecular mechanics energy minimization, the fragment reconstruction process intrinsically accounts for ligand flexibility in the docking process. The described protocol results in a ranked list of docked and minimized protein–compound complexes. These steps are summarized in the flow chart shown in Fig 3.5.

3.3.5 Benchmarking the CANDOCK Algorithm.

Throughout the paper, we evaluated different scoring functions for their ability to "select" the crystal-like ligand pose (i.e., a pose within 2.0 Å of the crystal ligand pose) as the most negatively scored pose (best-ranked pose) and termed them as

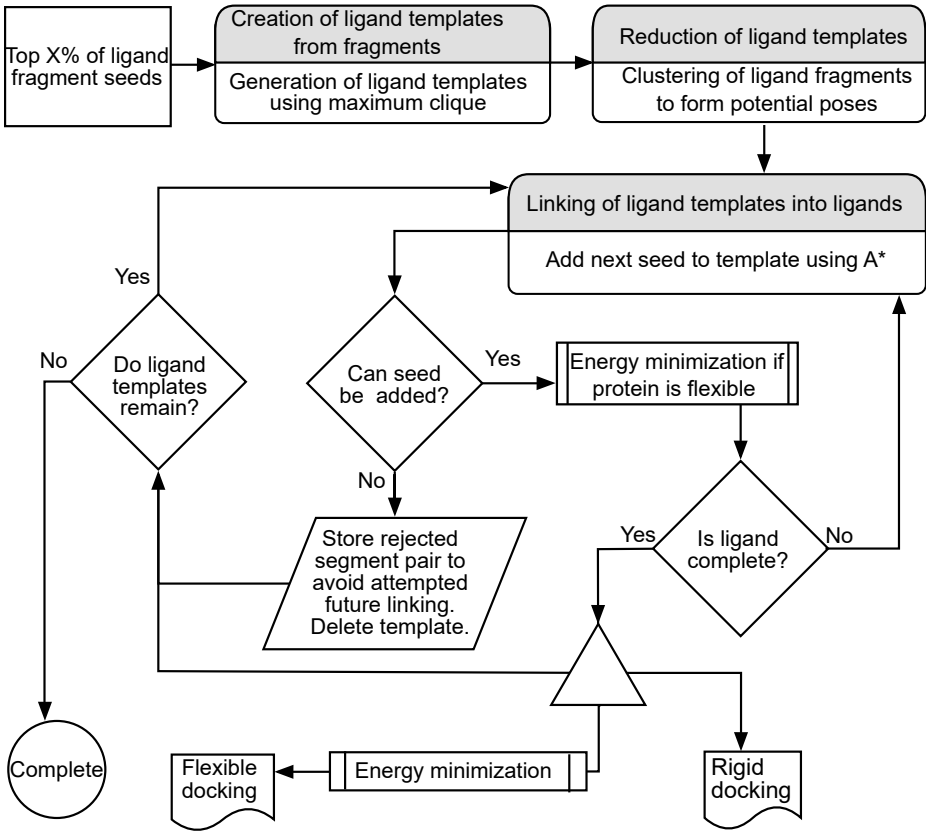


Figure 3.5. Workflow of the fragment linking procedure. The algorithm begins with a set of ligand fragments docked into the binding site of the protein (termed as seeds), which are selected based on their RMR6 score. The number of seeds is determined by the Top Seed Percent parameter. These fragments are joined together into ligand templates using the maximum clique algorithm, and the potential ligand templates are clustered using a greedy clustering algorithm, which remove ligand fragments within an RMSD of 2.0 Å from each other. The remaining ligand templates are joined using the A* algorithm, which determines whether a seed can be added to the growing ligand template. If the seed cannot be added, the template is rejected, and the pair is added to a list of failed pairs. If the seed can be added, then it is added to the ligand template. Once all seeds have been added to the ligand template, the template is accepted and energy-minimized in the binding pocket. The algorithm ends once all templates have been added or rejected.

"selectors" henceforth. Here, we define the selection rate as the fraction of the best-ranked poses (most negatively scored) within 2.0 Å of the crystal ligand pose. We calculated this selection rate for each scoring function at different radius cutoff values (4–15 Å) to identify the best selectors. This metric should not be confused with the success rate, which is simply the algorithm's ability to produce a crystal-like pose.

Benchmarking Sets of Choice.

There are a wide variety of benchmarking sets to evaluate docking programs to evaluate docking methods, most of which are derived from the Protein Data Bank [215]. We evaluated the CANDOCK hierarchical docking algorithm using a benchmarking set (1) to determine whether the algorithm can reproduce the crystal binding pose of the ligand in the binding site of the protein and (2) to correlate the scores of the three-dimensional (3D) docked poses of the ligand to the measured Kd/Ki values of the ligand binding with the protein. The PDBbind benchmark [97, 216] is very well suited for this analysis because, for each protein in this set, it provides 3D coordinates and corresponding activity values for five protein–ligand complexes. In the CASF-2016 benchmarking set (also referred to as the PDBBind Core set v2016), there are a total of 285 such complexes for 57 proteins of interest to the medicinal chemistry community. This benchmarking set includes decoy poses, which are used to validate our scoring functions independently of the CANDOCK algorithm. The number of fragments present in a given ligand range from a single fragment to ligands consisting of 13 fragments, enabling an evaluation of our method on both rigid and flexible ligands.

In addition to CASF–2016, we have also benchmarked our method against the Astex Diverse set [96] as several protein–ligand complexes in this set include metal ions and other cofactors, allowing us to showcase these examples and assess how our

algorithm handles these particular cases. We obtained each structure from the Astex set from the Protein Data Bank directly and only considered the biological assembly used to create the original benchmark. Additionally, to ensure that CANDOCK can generate native-like poses when not given the crystallographic coordinates of a ligand as input, we generated the 3D structure of each ligand from its SMILES string using Molconverter [217] and compared these results to those obtained when the original crystallographic coordinates were used.

To evaluate the performance of CANDOCK against noncognate protein structures, we have included benchmarking examples for the PINC is Not Cognate (PINC) benchmarking set [218]. From this set, we have chosen six target cases to evaluate CANDOCK: β-secretase1, carbonic anhydrase II, cyclin-dependent kinase 2, map kinase 14, PTP1b, and PPARγ. For this benchmarking set, multiple ligands with known crystallographic poses are supplied for a given target along with five example proteins crystallized with different ligands. The goal of this benchmark is to obtain the crystal pose of the supplied ligands in these noncognate protein crystal structures.

Input Preparation.

The binding site for both benchmarking sets is defined by spheres with a radius of 4.5 Å centered around each atom of crystal ligand. We did not remove any cofactors, solvent molecules, ions, or glycans when preparing our docking runs. The provided reference ligand was used to generate fragments and seeds for docking. The Astex benchmark was run again using input ligand coordinates generated using only the SMILES representation of the molecule and the Molconverter package from Chemaxon [217].

Parameters Chosen for Benchmarking.

The most important parameter present in CANDOCK for linking seeds into ligands is the Top Seed Percent parameter as it is crucial to select the number of seeds used to generate potential conformations via the maximum clique algorithm [211]. If this number is too small, then there will not be enough potential conformations generated to sample the conformational space of the ligand properly. In fact, there is a possibility that no conformations are generated during the linking step, causing CANDOCK to fail to produce any conformations. If the Top Seed Percent is too large, then the conformational search space is too large, and CANDOCK will become computationally inefficient (especially in the case of fully flexible protein docking). Therefore, we wanted to sample the potential Top Seed Percent values to determine how well our method does at various levels of conformational space sampling. The values chosen for this parameter are 0.5, 1.0, 2.0, 5.0, 10, 20, 50, and 100%. Similar to the conformational space sampled, we also investigated the effect of protein flexibility on the ability of the CANDOCK algorithm to reproduce the binding pose of a ligand. Accordingly, we used the algorithm in three modes: no protein flexibility (no energy minimization performed, maximum final iterations set to zero), with semiflexible protein (final energy minimization only, default options), and with a fully flexible protein (iterative energy minimization performed, iterative flag turned on). The RMSDs for all poses generated from all Top Seed Percent values and all flexibility modes are calculated with respect to the experimental crystal pose using a symmetry-independent method.

Finally, we determined the best-scoring function to select the pose from all generated poses that best reproduces the crystal ligand pose (the "selector" scoring function) and potentially differentiate it from another scoring function used to rank the activity of a given ligand to the protein target of interest (the "ranker" scoring func-

tion). To do this, we calculated the score of all poses generated for CASF-2016 using all scoring functions described previously. We then evaluated the ability of each scoring function to select the crystal pose of a ligand from all poses, as well as the correlation between the score assigned to the selected pose and the experimental binding affinity. As there are 96 scoring functions, there are 9216 (96 ways to select by 96 ways to rank) different methods to rank the affinity of the ligands in CASF–2016. An overview of this benchmarking process for activity prediction is given in Fig 3.6.

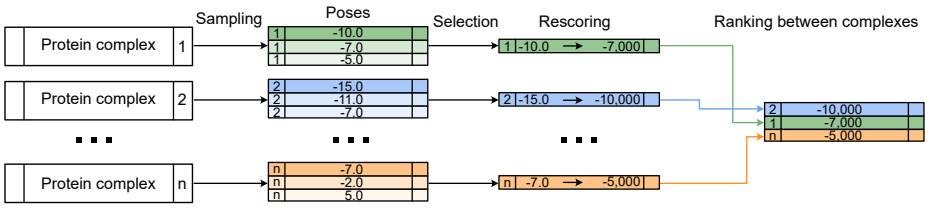


Figure 3.6. CANDOCK activity evaluation pipeline. Sampling is performed using the RMR6 scoring function to generate thousands of ligand poses. The best pose is selected with a selector scoring function to represent the protein–ligand complex. Only this selected pose is rescored using the ranker scoring function, which is used to assign a new score to the complex. The best ranker score on the selected pose is used to rank the protein–ligand complex based on correlation with pK_d/pK_i data.

3.4 Results and discussion

We discuss the performance of the CANDOCK algorithm in reproducing the crystal pose of a ligand via sampling the conformational space of the ligand in the binding pocket (including the entire chemical environment with cofactors, metal ions, crystal waters, and so on) modeled with different levels of protein flexibility for two benchmarking sets. In addition, we evaluate the ability of the algorithm to discriminate the crystal pose from all poses generated by the algorithm and the ability to rank the activity of the ligands against the protein targets of interest.

3.4.1 Knowledge-Based Scoring Functions Perform Well on the Decoys present in the CASF-2016 Benchmark

Before evaluating the ability of the CANDOCK algorithm to reproduce the crystal pose of a ligand in the binding pocket of a protein as measured by success rate, we first show that the scoring functions perform well at selecting a crystal-like pose from the decoy poses provided by the CASF–2016 benchmark set [216]. First, we evaluated our 96 scoring functions on the "docking power" test provided by the CASF–2016 benchmark. Docking power is the ability for a scoring function to select a pose within 2.0 Å of the crystal pose and is synonymous with selection rate with the exception that docking power is measured on poses not generated by CANDOCK. Our results show that the RMR5 and the RMR6 scoring functions outperform all of the others with success rates of 87 and 86%, respectively, when the crystal pose is not included with the decoys. When the crystal pose is included, the docking powers increase to 95 and 94%, respectively. These values outperform all other scoring functions in the original CASF–2016 paper [216]. Moreover, our best–performing scoring functions (RMR5 and RMR6) also outperform a machine-learning-based scoring function, recently introduced to improve its performance [219]. It should be noted that the performance of the scoring functions is within the statistical error of both RMR5 and RMR6 (compare the first three columns of Tables 1 and S4—S9 published for the CASF–2016 benchmark [216]), suggesting that our scoring functions perform at least as good as the best-scoring functions benchmarked in the original work.

Using the selector/ranker methodology described in Fig 3.6, we used both RMR5 and RMR6 as selectors and 12 other scoring functions (RMC10, RMC11, RMC12, RMC13, RMC14, RMC15, FMC10, FMC11, FMC12, FMC13, FMC14, and FMC15) as rankers for the scoring power and ranking power tests for binding affinity, as described in the original CASF-2016 paper [216]. Additionally, the RMSD of the

provided decoy pose is used as a selector to test whether knowledge of the crystal pose is needed for adequate ranking and scoring. The corresponding Pearson and Spearman correlation coefficients are given in Table 3.1. The best selector ranker pair for the proved decoys is RMR5/ RMC13 with a Pearson correlation of 0.626 (confidence interval of [0.566–0.6779]). This result places this correlation within statistical error of the best published nonmachine learning scoring functions [216,219]. For the ranker test, the best combination is RMR5/RMC14 with a Spearman correlation of 0.5964 (confidence interval of [0.49–0.675]), a result which places our scoring functions within the top 10 nonmachine learning scoring functions and within statistical error of the best-scoring function. It should be noted that all of the selectors chosen for this analysis (see Table 3.1) perform within the statistical error of each other, indicating that the family of scoring function with large cutoffs, using mean reference state, and complete reference for the protein–ligand complex is well suited for ranking ligand affinities.

3.4.2 Ligand Conformational Sampling Is Enhanced by Fragment Docking and Protein Flexibility.

An important feature of any receptor–ligand docking methodology is its ability to generate docked crystal-like ligand poses within 2.0 Å RMSD of the experimentally determined pose of the native ligand [220, 221]. Using the CASF-2016 benchmarking set, we validated the ability of CANDOCK to generate crystal-like poses among the docked poses. We plotted the cumulative frequencies of all docked poses with the RMSDs from their corresponding crystal ligand's poses for all Top Seed Percent values and for varying degrees of protein flexibility using the RMR6 scoring function (Fig 3.7, left-hand-side panels). Expectedly, these plots indicate that the use of larger ($> 20\%$) Top Seed Percent values generated significantly more poses within 2.0 Å

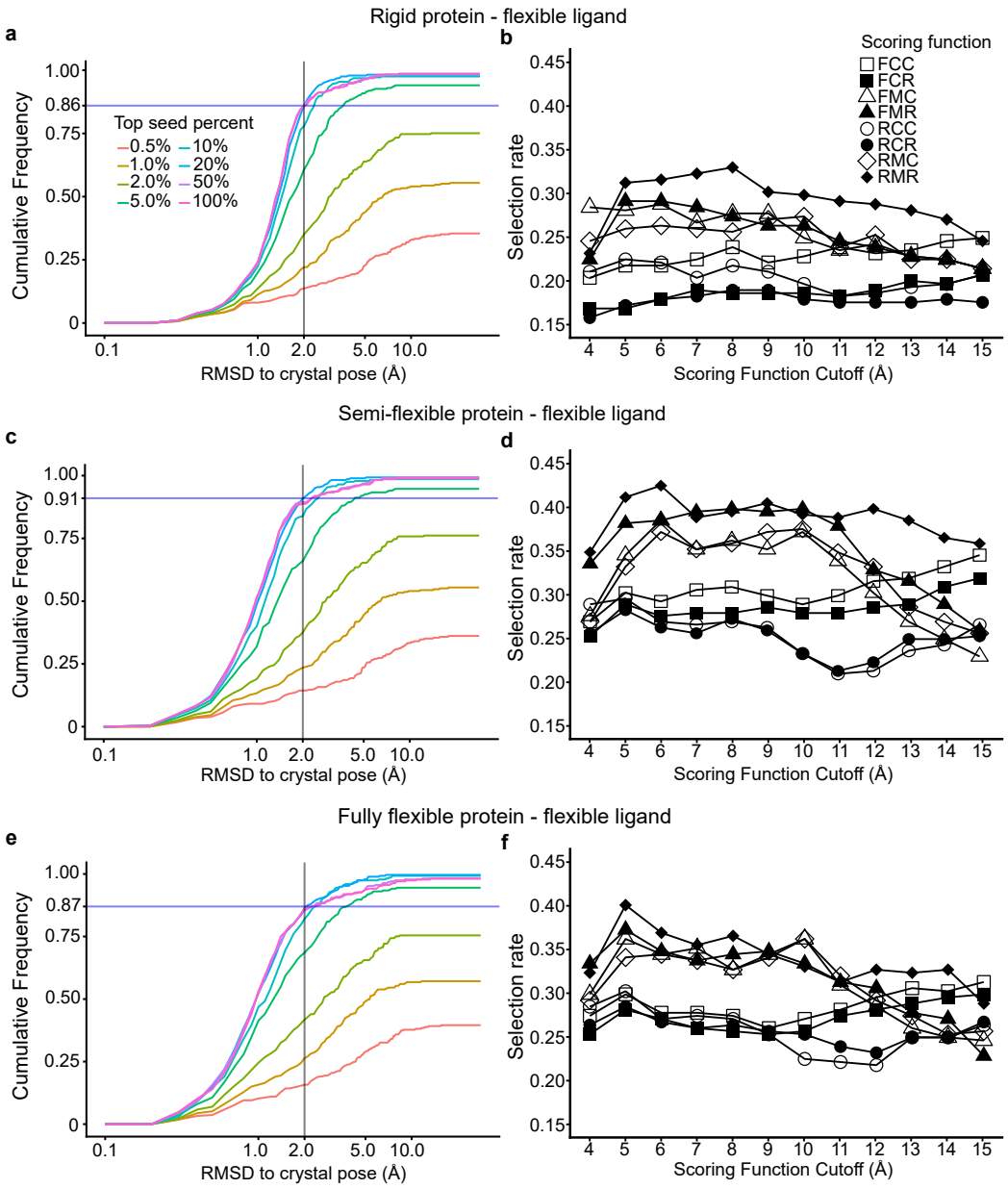


Figure 3.7. Cumulative frequencies of the best RMSD pose generated for rigid (flexible ligand only with no energy minimization of protein–ligand complex), semiflexible (energy minimization of protein–ligand complex at the end), and fully flexible (iterative energy minimization during the linking procedure) CANDOCK docking results for the 285 proteins in CASF-2016 using the RMR6 scoring function are given in (a), (c), and (e) respectively. The selection rate, i.e., the portion of the best-scored docked poses within 2.0 Å of the crystal pose, is given for different scoring functions employed in (b), (d), and (f).

Table 3.1.
Statistics Shown for the Docking Power (Selector Only), *Scoring Power* (Pearson Correlation between the Ranker and Binding Affinity), and *Ranking Power* (Spearman Correlation between the Ranker and Binding Affinity) Tests. The RMSD of the decoy is an additional selector to show that the RMSD is not required to achieve the best correlation.

Selector	**Native**	**Docking**	**Ranker**	**Scoring**	**Ranking**
RMR5	no	84.0–90.0	RMC13	0.5660–0.6779	0.4804–0.6625
	yes	92.0–96.0	RMC14	0.5642–0.6755	0.4900–0.6750
			RMC15	0.5577–0.6712	0.4696–0.6661
			FMC13	0.5637–0.6764	0.4875–0.6679
			FMC14	0.5624–0.6742	0.4857–0.6714
			FMC15	0.5565–0.6692	0.4696–0.6661
RMR6	no	83.0–90.0	RMC13	0.5637–0.6745	0.4643–0.6446
	yes	92.0–96.0	RMC14	0.5618–0.6721	0.4732–0.6589
			RMC15	0.5553–0.6678	0.4500–0.6411
			FMC13	0.5590–0.6716	0.4696–0.6500
			FMC14	0.5600–0.6713	0.4696–0.6518
			FMC15	0.5533–0.6661	0.4500–0.6429
RMSD			RMC13	0.5634–0.6680	0.3405–0.5214
			RMC14	0.5560–0.6624	0.3429–0.5179
			RMC15	0.5502–0.6569	0.3357–0.5125
			FMC13	0.5622–0.6668	0.3482–0.5232
			FMC14	0.5558–0.6613	0.3393–0.5161
			FMC15	0.5496–0.6557	0.3339–0.5143

than lower ($< 10\%$) Top Seed Percent values. For the semiflexible (Fig 3.7c) method, the Top Seed Percent value of 20% yielded the highest number of poses within 2.0 Å of the crystal pose, with the corresponding cumulative frequency of 91%, compared to independent benchmark of the best-performing methods resulting in an 80% success rate to generate the pose [185]. The semiflexible method thus outperformed the rigid protein (Fig 3.7a) and the fully flexible (Fig 3.7e) methods for the larger Top Seed Percent values that correlate with a higher sampling of the ligand conformational space during fragment docking. However, the fully flexible protein method

outperformed the semiflexible (Fig 3.7c) and the rigid protein (Fig 3.7a) methods for smaller Top Seed Percent values such as 5 and 10%. In addition, the Boltzmann-like distributions in the RMSD plots (Fig 3.8) indicate that the CANDOCK algorithm adequately sampled the ligand conformations both far and close to the crystal ligand pose in CASF–2016. This suggests that the prediction of energetically favorable ligand con- formations is dependent on near-native protein flexibility during the linking of docked fragments. There are only 17 cocrystal structures (out of 285), where the semiflexible algorithm failed to find a single crystal-like pose for the native ligand (1H22, 1H23, 1NVQ, 1U1B, 1YDT, 2P15, 2QNQ, 3AG9, 3BV9, 3KWA, 3O9I, 3PRS, 3UEU, 3URI, 3ZSO, 4EA2, 5C2H) for any Top Seed Percent value. An additional nine complexes (2C3I, 2CET, 2W66, 2WCA, 3ARU, 3BGZ, 3OZT, 3RR4, 3UEX) failed to find a crystal-like pose when the semiflexible algorithm was used with a Top Seed Percent value of 20%. Two of these complexes (3BV9, 3URI) contain a peptide ligand with a protein, a situation generally treated differently in other docking studies. 37 When fully flexible docking is considered, CANDOCK fails on a total of 10 complexes, out of 285, resulting in an overall success rate of 96% to generate crystal-like poses. Specifically, CANDOCK generates successful (crystal-like) poses for 7 complexes out of 17 failures from semiflexible docking (3O9I, 2QNQ, 1YDT, 3ZSO, 5C2H, 3UEU, and 4EA2), and 2P15 becomes a near-hit with an RMSD of 2.04 Å. These results indicate that the hierarchical generation of the ligand poses with the protein flexibility considered after fragment docking and ligand reconstruction is a successful strategy for enhanced sampling of the conformational space of ligands in protein–ligand complexes.

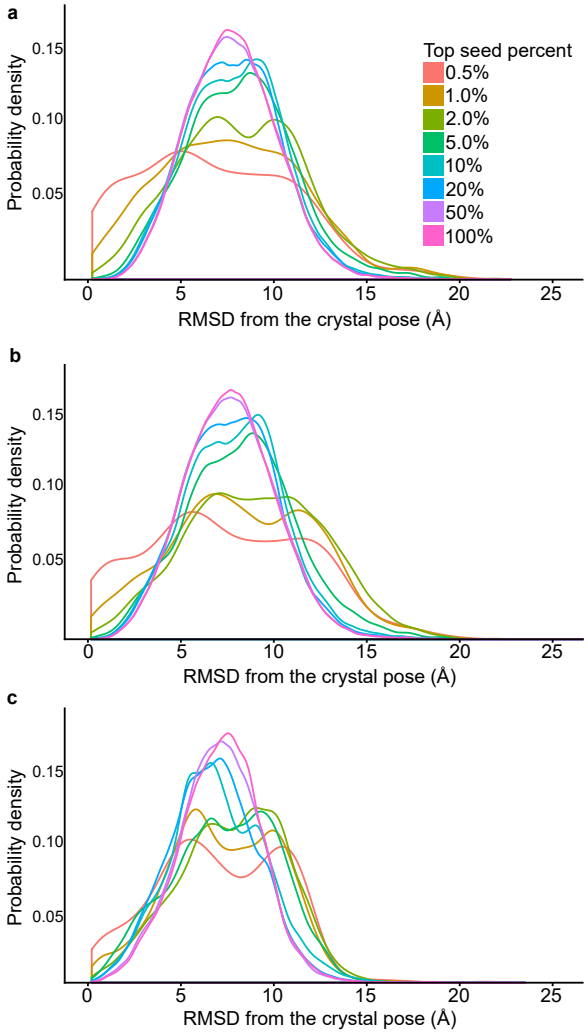


Figure 3.8. Distribution of RMSD values (Å) for all ligand poses generated by CANDOCK for docked poses in the CASF-2016 benchmark for (a) rigid-protein docking, (b) semi-flexible protein, and (c) fully-flexible protein docking.

3.4.3 Radial-Mean-Reduced (RMR) Scoring Function Family Generates Best-Docked Ligand Poses

The RMR family of scoring function at a cutoff radius value of 6 Å from each atom of the ligand (RMR6) performed best for the semiflexible protein method (Fig 3.7, right-hand-side panels). The best selector scoring functions for the rigid protein

method were RMR8 and RMR5 for the fully flexible protein method. This shows that the RMR scoring function family is the best selector among eight other generalized families of scoring functions. Conversely, the radial cumulative complete (RCC) scoring function family performed the worst in selecting the crystal pose from the generated poses with the RCC11 scoring function being the overall worst selector.

To elucidate the rationale behind the good performance of RMR6 in selecting a crystal-like pose, we plotted the RMR6 score of the docked ligands with the lowest RMSD from the crystal pose against the RMR6 score of the crystal pose (Fig 3.9). For Top Seed Percent values $> 10\%$, there is a clear separation between the successful poses within 2.0 Å (blue points) and the failed poses far from the crystal ligand pose (red points). Moreover, these failed poses cluster above the diagonal line, indicating that RMR scores of failed complexes have higher energy value (as expected) than the crystal pose during sampling for Top Seed Percent values $> 10\%$ (Fig 3.9). The number of failed poses decrease to lower numbers with increasing "Top Seed Percent," from 244 for 0.5%, 218 for 1.0%, 178 for 2.0%, 97 for 5.0%, 46 for 10%, 26 for 20%, 30 for 50%, and 32 for 100%. These data suggest that a Top Seed Percent of 20% yields the highest number of poses within 2.0 Å of the crystal pose (previous section; Fig 3.7, left-hand-side panels) and the number of failed cases are rare and clearly discriminated from both the crystal pose and the successful near-native docked poses (blue points) by using the RMR6 scores. Therefore, RMR6 can discriminate native and near- native interactions from a set of incorrect conformations generated by our docking method. Furthermore, RMR6 scoring function is a decent selector as the top pose selection rate of 41% for semiflexible docking at a Top Seed Percent of 20% (Fig 3.7, right-hand center panels) and is comparable to the state-of-the-art independent benchmarks [185]. Clearly, for these successful cases, the best (most negative) RMR6 score corresponds to a pose within 2.0 Å RMSD of the crystal pose

(Fig 3.10). However, RMR6 has a bias toward incorrectly scoring a noncrystal-like pose better than the experimental crystal pose for both successful and failed cases (see scoring function correlation to pose deviations in appendix F).

If we include predicted poses other than the best-scored pose, then we get a much higher selection success rate of 55% when top 2 poses are selected, 69% when top 5 poses are selected, and 76% when top 10 poses are selected. While the RMR6 scoring function is a decent selector, more work is needed to enhance the selection success rate, perhaps in combination with other scoring functions at different cutoffs along using machine learning methods [151, 222]. However, it is good to note that, without any machine learning, our generalized RMR6 scoring function is comparable to successfully selecting a pose to a recently published neural-network-based scoring selection [223] with a selection rate of 50% for the top pose and 65% for the top 5 poses. This suggests that a reduced composition over all pairwise protein's and compound's specific atom types with mean reference state improves discriminatory accuracy by giving "context" to the specific pose by solely including atom-type interactions that are possible between the receptor and the ligand.

3.4.4 Docking Long Aliphatic Chains Needs Enhanced Sampling.

We identified six complexes (1H22, 1H23, 3AG9, 3KWA, 3UEU, and 4EA2) out of 17 failed cases with CANDOCK semiflexible algorithm with ligands that contain long aliphatic carbon chains (greater than 4 atoms). The remaining 11 complexes that fail are 3URI (8-mer peptide), 3O9I, 1U1B, 2QNQ, 3BV9 (6-mer peptide), 3PRS (14 fragments), 1YDT, 1NVQ, 2P15, 5C2H, and 3ZSO. If fully flexible protein docking is considered, we get 4 complexes out of 10 failed cases that contain long aliphatic carbon chains (1H22, 1H23, 3AG9, 3KWA). CANDOCK does not consider an aliphatic chain consisting of three carbon atoms (sp3-hybridized carbon; C3) as fragments for

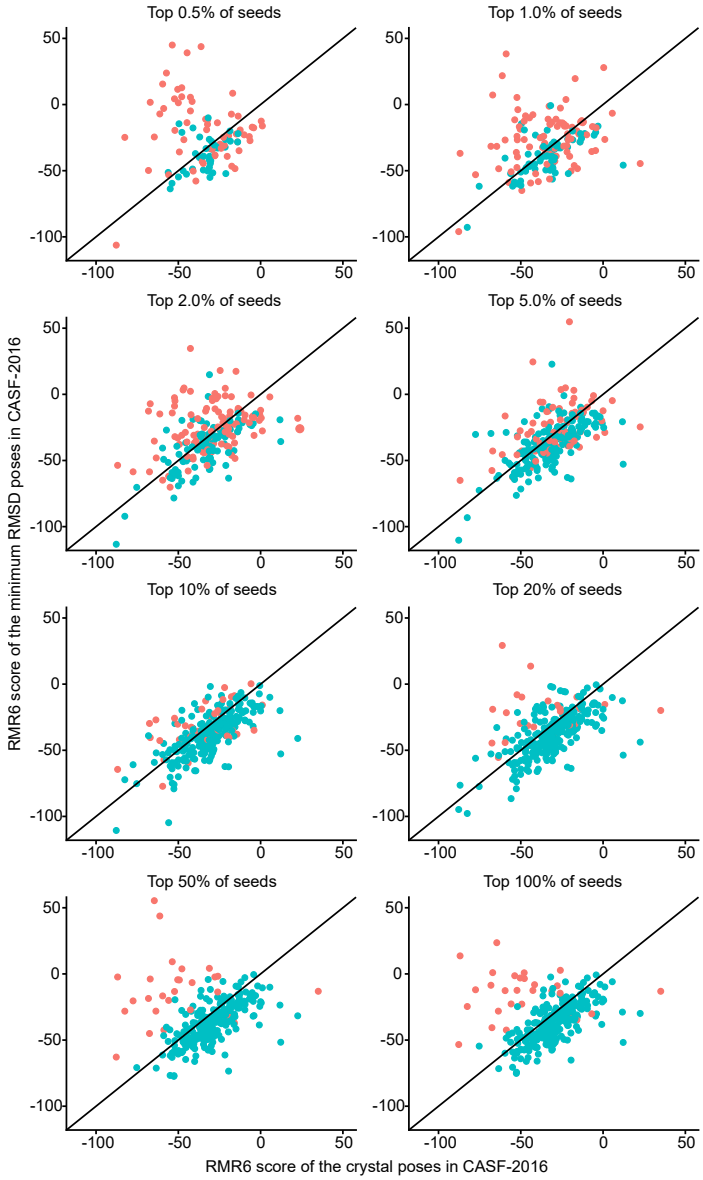


Figure 3.9. Correlations between the RMR6 scores of the crystal poses and the pose with the lowest RMSD are shown for all eight top percent values for complexes in CASF–2016. Poses within 2.0 Å of the crystal pose are shown in blue (success) while poses with RMSD $>$ 2.0 Å (failures) are shown in red. For top percent values greater than 20%, the complexes that failed cluster above the y=x line. Therefore, in these cases, the CANDOCK algorithm did not sample the conformation space close to the binding pocket.

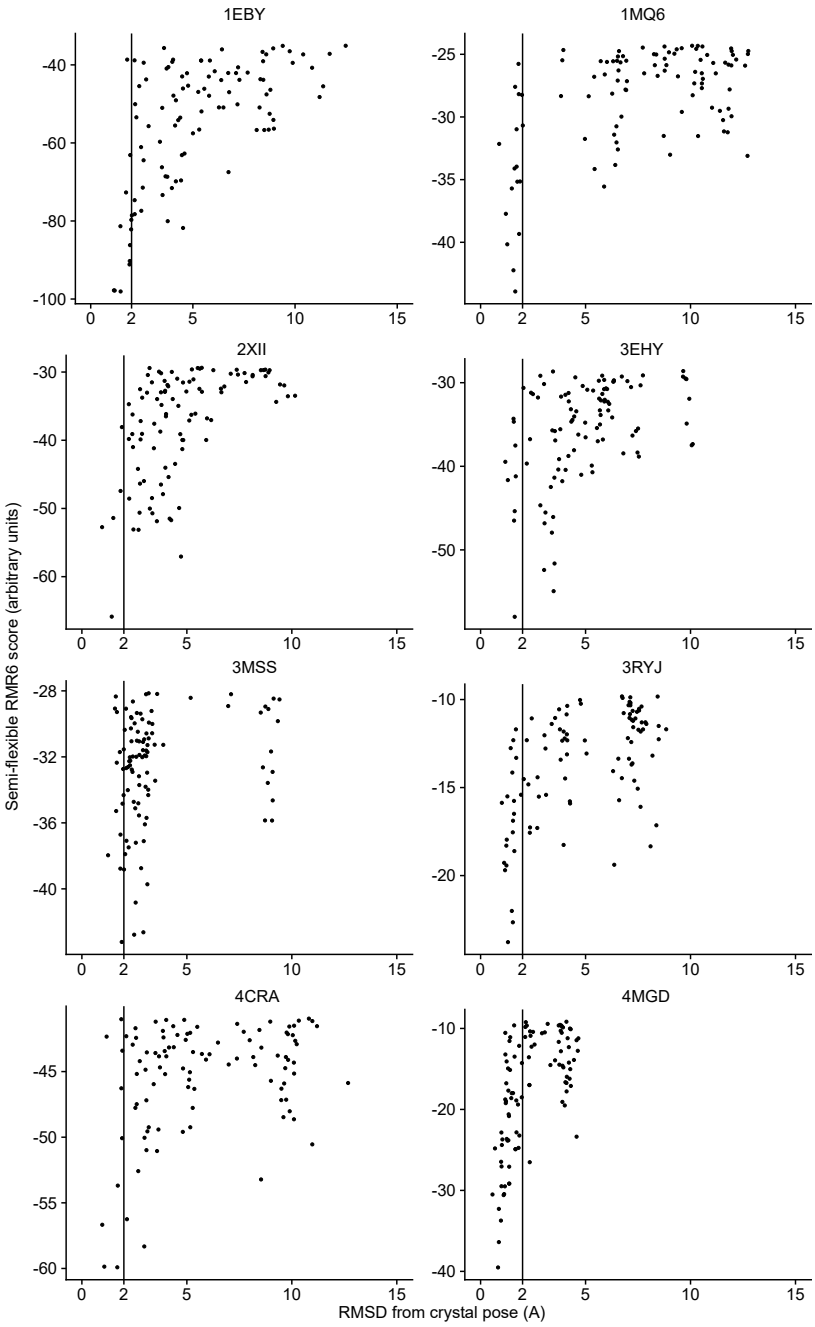


Figure 3.10. Plots of the RMR6 score of all poses produced by CANDOCK for selected proteins in CASF-2016 versus the RMSD of the pose. In all plots, the RMSD ranges from 1Å to 15Å. The poses were obtained using the semi flexible method at a Top Seed Percent value equal to 20%. These of these plots show a tunnel–like affect around as one approaches an RMSD of zero, showing the scoring functions ability to select the crystal pose in these cases.

docking. Instead, the A* search algorithm determines the docked positions by rotating them around the bond vectors of the growing chain at 60° increments. We hypothesize that this discrete sampling of conformational space, and not the potential functions in CANDOCK, is the cause for the poor performance of the algorithm on these compounds with many rotatable bonds. To test our hypothesis for the six failed long aliphatic carbon chain complexes (1H22, 1H23, 3AG9, 3KWA, 3UEU, and 4EA2), we scored the decoys provided by the CASF benchmarking set [216] that included at least one pose within 2.0 Å RMSD. In all six cases, the RMR6 scoring function selected a pose within 2.0 Å RMSD of the crystal ligand, indicating that our generalized scoring function does not account for failure to identify crystal-like conformations (see the sheep plot in scoring function correlation to pose deviations of appendix F). We plan to address this issue in detail in future versions of the algorithm by implementing a new sampling method or a ligand-class-specific scoring function, similar to what was done for the support of carbohydrates in Autodock Vina separately [224].

3.4.5 Protein Flexibility Improves Docking Ligands with Many Rotatable Bonds.

The number of rotatable bonds in a ligand significantly influences the ability of docking algorithms to generate docked crystal-like ligand poses [185]. To study the effect of rotatable bonds on the performance of the algorithm, we compute the selection rate of the RMR6 scoring function against the number of fragments in a ligand (Fig 3.11). Due to the hierarchical fragment-based nature of the CANDOCK algorithm, the number of ligand fragments is used instead of the number of rotatable bonds to measure CANDOCK's performance. By comparing the fully flexible protein method (Fig 3.11c) to the rigid protein method (Fig 3.11a) and to the semiflexible

method (Fig 3.11b), we show that the selection rate for flexible ligands increases with including protein flexibility during docking. Here, we define a flexible ligand with greater than 4 total fragments as the average number of fragments is 3.8 and the median is 3 fragments in the CASF-2016 data set. Specifically, for the 216 ligands with 4 or fewer fragments, the semiflexible (Fig 3.11b) and the fully flexible (Fig 3.11c) methods performed equally well. The rigid, semiflexible, and fully flexible methods have respective selection rates of 50 ± 3.5, 66 ± 3.2, and $65 \pm 3.2\%$ for the top pose; 65 ± 3.3, 76 ± 2.9, and $77 \pm 2.9\%$ when top 2 poses are selected; 74 ± 3.0, 83 ± 2.6, and $86 \pm 2.4\%$ when top 5 poses are selected; and $79 \pm 2.8\%$, $88 \pm 2.2\%$, and $91 \pm 2.0\%$ when top 10 poses are selected. Thus, full protein flexibility is not essential for ligands with less than 5 fragments as there is little difference in the selection rate between semiflexible and fully flexible docking (Fig 3.11b,c). In contrast, for 69 ligands with greater than 4 fragments, the rigid, semiflexible, and fully flexible methods have respective mean selection rates of 29 ± 5.6, 54 ± 5.9, and $54 \pm 6.0\%$ for the top pose; 35 ± 5.8, 64 ± 5.7, and $68 \pm 5.7\%$ when top 2 poses are selected; 47 ± 6.0, 75 ± 5.2, and $79 \pm 4.9\%$ when top 5 poses are selected; and 53 ± 6.1, 77 ± 5.1, and $87 \pm 4.1\%$ when top 10 poses are selected. Better performance of flexible methods versus the rigid method for larger ligands is most likely caused by the plateauing and even slight decline in the number of poses generated for ligands with > 5 fragments for Top Seed Percent values $> 10\%$ (see the number of poses generated in timing section of appendix F). This suggests that there is an upper limit to the sampling space possible for a given binding site and for a given ligand, and once this limit is reached, the algorithm is no longer able to produce more docked ligand poses. From the values given, it is clear that the semiflexible and fully flexible methods are superior to the rigid method. However, while it is difficult to determine a direct superiority of the fully flexible method over the semiflexible method for the top pose through

the top 5 poses, the fully flexible method outperforms the semiflexible method when considering the top 10 poses. Therefore, we conclude that protein flexibility is an important feature of the CANDOCK algorithm.

3.4.6 Inclusion of Chemical Environment and Cofactor Interaction in Binding Sites Lead to Accurate Crystal-like Ligand Pose Generation.

The Astex diverse set [96] is a widely used benchmarking set for measuring a docking program's ability to predict the native pose of a ligand. One important feature of this set, compared to CASF–2016 [216], is the inclusion of several cofactors and metal ions such as zinc ions and heme groups in the binding sites. Traditionally, with docking methods, the cofactors in the binding pockets have been ignored or treated as nonphysical models with improper representations that affected performance [216]. As an example, for heme groups, we used a previously published extension to the GAFF force field to ensure proper representation of this cofactor during the minimization procedure [225], compared to other methods treating it as a hydrogen bond donor [176]. We hypothesize that, to perform well on this benchmarking set, the docking algorithm must properly sample ligand conformations interacting with metal ions and doing so requires an adequate representation of metal–ligand interaction potentials at the atomic scale. A generalized potential function can include all relevant cofactors, metal ions, etc. in the binding pocket as separate interactions (Fig 3.12) compared to one metal-ion type used by others [176, 216]. To highlight the ability of our scoring function to characterize such interactions in a pairwise fashion, we plotted various atom pair interactions of interest to medicinal chemists (Fig 3.12).

The number of complexes in this benchmarking set, where the CANDOCK algorithm produces a ligand pose within 2.0 Å RMSD of the crystal pose, is given in

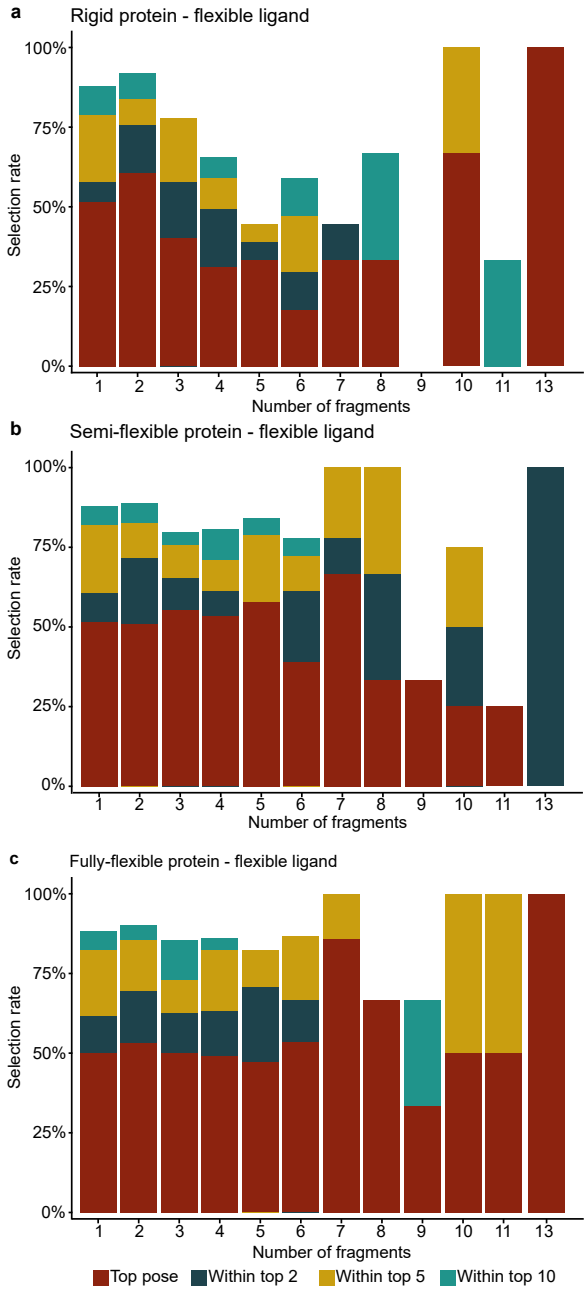


Figure 3.11. Selection rates for the RMR6 scoring function with rigid (a), semiflexible (b), and fully flexible (c) CANDOCK docking arranged by the number of ligand fragments in CASF–2016. For fragment counts greater than 13, no poses within 2.0 of the crystal pose was generated.

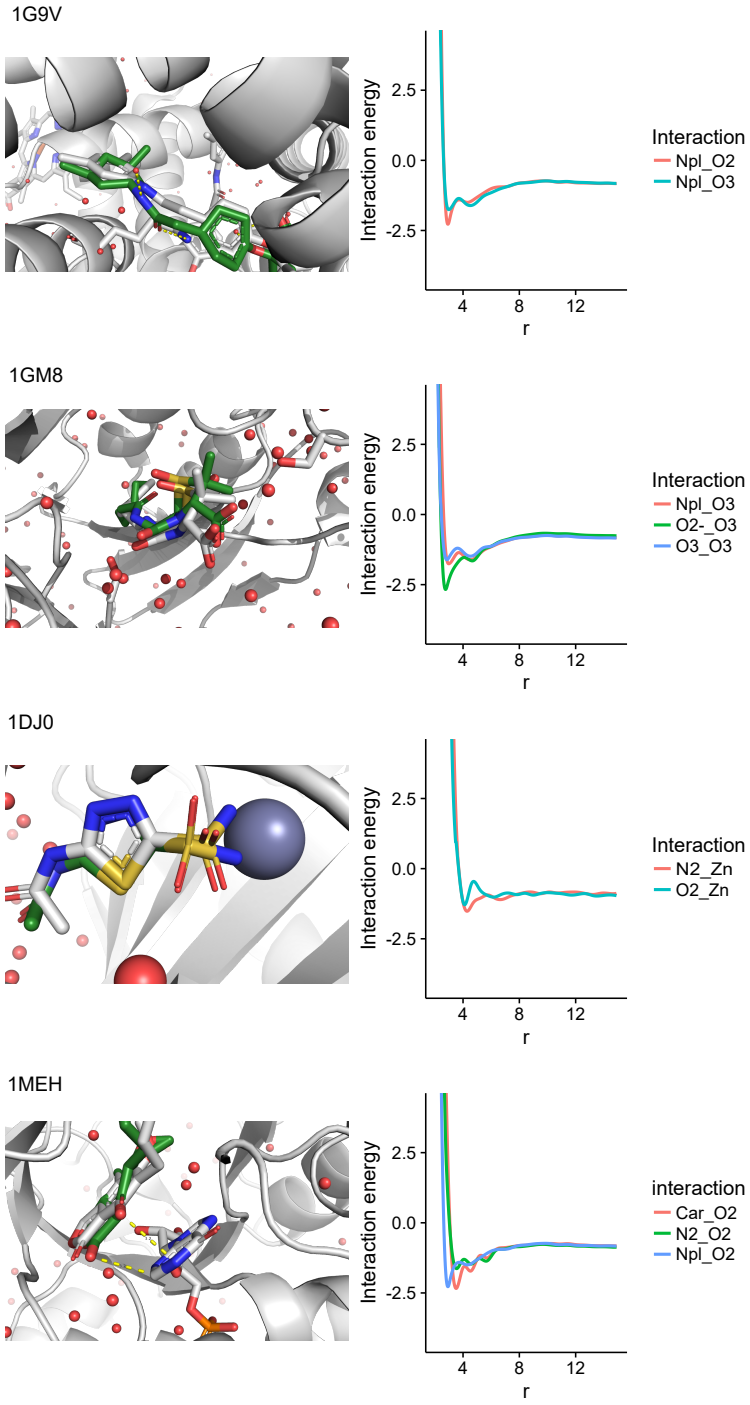


Figure 3.12. Examples where CANDOCK is able to produce a good docking pose where other methods are not able. The best CANDOCK pose is given on the lefthand side of the figure and important interactions between the ligand protein are given on the right.

Table 3.2. CANDOCK successfully generates a crystal pose for 97.6% of the Astex benchmarking set (83 of the 85 complexes). We attribute this success to the ability of our algorithm to properly sample the conformational space of a ligand in the binding pocket while considering all interactions of the ligand within the binding pocket, including cofactors, metal ions, etc. In a recent comparison using Astex data set [141], the success rates for FlexAID [141], Autodock Vina [176], FlexX [226], and rDock [195] are 66.7, 81.8, 78.8, and 89.4%, respectively, when all 85 complexes are considered. When 16 complexes containing a metal ion were removed (1GKC, 1HP0, 1HQ2, 1HWW, 1JD0, 1JJE, 1LRH, 1MZC, 1OQ5, 1R1H, 1R55, 1R58, 1UML, 1XM6, 1XOQ, 1YQY), the success rates of these methods increased to 72.1, 83.6, 79.7, and 91.3%, respectively [141]. CANDOCK outperforms these methods without removing metal-ion complexes from the benchmarking set, supporting the hypothesis of adequate sampling and included proper representation of interactions within the binding site. The two complexes where CANDOCK nearly missed to generate a crystal pose using the semiflexible method are 1HP0 (lowest RMSD of 2.08) and 1W1P (lowest RMSD of 2.734). Additionally, when the protein is considered as a rigid body (rigid docking), CANDOCK failed to find crystal poses for 1Y6B and 1MZC as well (81 of 85 complexes in Table 3.2). The algorithm also performs well on complexes that failed by using other popular docking methodologies for the Astex diverse set. According to a previous study, [141] there are four complexes (1G9V, 1GM8, 1JD0, and 1MEH) where Autodock Vina [176], rDock [195], FlexX [226], and FlexAID [141] all have difficulty reproducing the crystal-like pose of the ligand but CANDOCK successfully generated a crystal-like pose. CANDOCK is able to select a crystal-like pose 52% of the time for the top-scored pose, 60% of the time for the top 2 poses, 66% of the time in the top 3 poses, 75% of the time in the top 5 poses, and 79% of the time in the top 10 poses.

Table 3.2.
Number of Successes in the Astex Diverse Set for all TSP Values. OpenBabel [227] was used to change ligand conformation of the crystal pose for AutoDock Vina.

CANDOCK		
TSP	Rigid	Semiflexible
0.5%	7	7
1.0%	14	15
2.0%	28	33
5.0%	57	60
10%	67	74
20%	77	79
50%	79	82
100%	78	81
All%	81	83
AutoDock Vina		
	Native input	Non-native input
	79	68

When CANDOCK was given starting coordinates generated from the SMILES string of the Astex ligands using Molconverter [228], it produced a crystal-like pose for 77 of the 85 complexes. As compared to running CANDOCK with 20% of the docked seeds and the crystallographic coordinates as input, there are three additional failures: 1M2Z, 1XM6, and 1XOZ. Conversely, 1MCZ was docked successfully when using coordinates generated from a SMILES string; however, the best RMSD score when using crystallographic input ligand was a near-hit with a value of 2.15 Å. These three complexes all have large ring structures, which cause fewer than 100 seeds to be created after fragment docking. Decreasing the clustering radius for the clustering step of the linking phase resulted in crystal-like poses for all three complexes, and a similar strategy yielded a crystal-like pose when applied to 1HP0. Therefore, we conclude that the CANDOCK algorithm performs equally well when given noncrystallographic coordinates provided that large rings are accommodated in the clustering

step of the linking phase. CANDOCK's performance with non-native ligand inputs is in contrast to that of Vina, where the use of non-native coordinates yields only a crystal pose for 68 of the 85 poses as compared to 79/85 when crystallographic coordinates are used (Table 3.2).

The interactions of the ligand with cofactors in the binding pocket for these complexes are shown in Fig 3.12. Specifically, 1G9V has cation–π interaction and 1GM8 has $\pi - \pi$ interactions between an aromatic ring and the surrounding protein environment. Similarly, 1MEH contains a $\pi - \pi$ stacking interaction between the ligand and a cofactor. 1JD0 has an interaction between the zinc ion and a sulfonyl group. These complexes showcase the success of our hierarchical docking method over previously published works.

We also consider specific cases where CANDOCK successfully reproduced the crystal pose of ligands, which interact with a cofactor (Fig 3.13). Specifically, in Fig 3.13a,b, for oxygen–zinc interactions in 1HWW and 1R55 during docking, the energy minimization procedure moved the location of the Zn^{2+} ion in the binding pocket (2.4 and 1.5 Å, respectively) as there are no constraints to restrict its movement within the binding pocket. This movement does not prevent the algorithm from generating a ligand pose within 2.0 Å RMSD of the native structure. For 1OQ5 and 1JD0, the docked poses of ligands interact with a zinc ion through a sulfonyl amide group (Fig 3.13c,d), and it is interesting to note that the zinc ion moved much less in these cases (0.5 and 0.6 Å). For the ligand in 1OQ5 (Fig 3.13c), the orientation of the sulfonyl amide aligns perfectly with the reference crystal pose, suggesting that the interactions with the sulfonyl amide group caused the zinc ion to stay in place. For the ligand in 1JD0 (Fig 3.13d), the docked pose of the same group does not align with its reference; however, the overall pose still is within 2.0 Å of this reference. Therefore, the ability

of the algorithm to produce a pose within 2.0 Å of the reference is not dependent on correctly predicting the orientation of all functional groups in a given molecule.

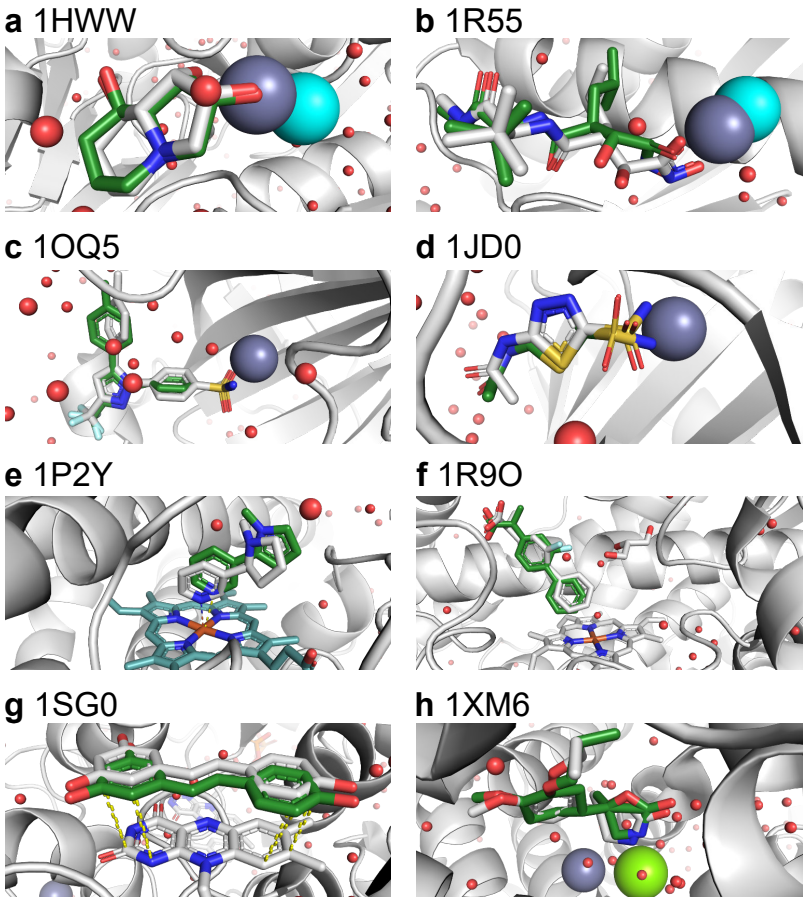


Figure 3.13. The reference pose is given in white and the lowest RMSD pose predicted by CANDOCK with a Top Seed Percent value of 20% using the semiflexible method is given in green. Panels (a) and (b) were selected due to the presence of oxygen–zinc interactions. The zinc ions before and after energy minimization are given in gray and cyan, respectively. The complexes in (c) and (d) show the interactions between sulfonylamide groups and a zinc ion. The interaction of a compound with a heme group via a nitrogen lone pair is shown in (e), and the interaction of an aromatic carbon with a heme group is given in (f). Finally, panels (g) and (h) show the interactions of compounds with other cofactors, such as a $\pi - \pi$ interaction of a compound with flavin–adenine dinucleotide and interaction of a compound with zinc and magnesium in a binding pocket.

We selected a larger organic cofactor (heme group) in the binding site of the protein–ligand complexes, 1P2Y and 1R9O (Fig 3.13e–h). The heme group is present in several liver enzymes [229–231]; therefore, predicting the location of a ligand relative to this group is important for medicinal chemistry. For 1P2Y, CANDOCK predicts the pose of a compound relative to the heme group when the nitrogen of the compound is interacting with the iron atom of this group (Fig 3.13e). Similarly, for 1R9O, a successful pose is generated including the interaction between an aromatic carbon and the iron atom (Fig 3.13f) indicating that proper representation of the heme group is essential to capture such interactions to generate the binding pose. We also demonstrate that generating a crystal- like docked ligand pose in the presence of a large cofactor is independent of the size of the cofactor itself. This is shown for the 1SG0 complex containing the flavin–adenine dinucleotide cofactor (Fig 3.13g), where the dominant interaction between the ligand and the cofactor is $\pi - \pi$ stacking. A crystal-like pose was also reproduced when the type of interaction changed dramatically, as shown in 1XM6 for the binuclear metal center formed by zinc and magnesium ions (Fig 3.13h). These interactions are important for developing phosphodiesterase inhibitors [232]; therefore, it is encouraging to observe CANDOCK's ability to reproduce a crystal pose in these cases. We conclude that the algorithm is able to generate a crystal-like docking pose by including interactions with diverse cofactors in the binding pocket.

3.4.7 Radial Mean Complete (RMC) Scoring Function at 15 Å Cutoff Is Best for Energy Minimization

A potential or scoring function, used for energy minimization of a protein and a ligand, should correlate quantitatively with the RMSD between the docked ligand and the crystal ligand so that a decrease in score corresponds to a decrease in RMSD.

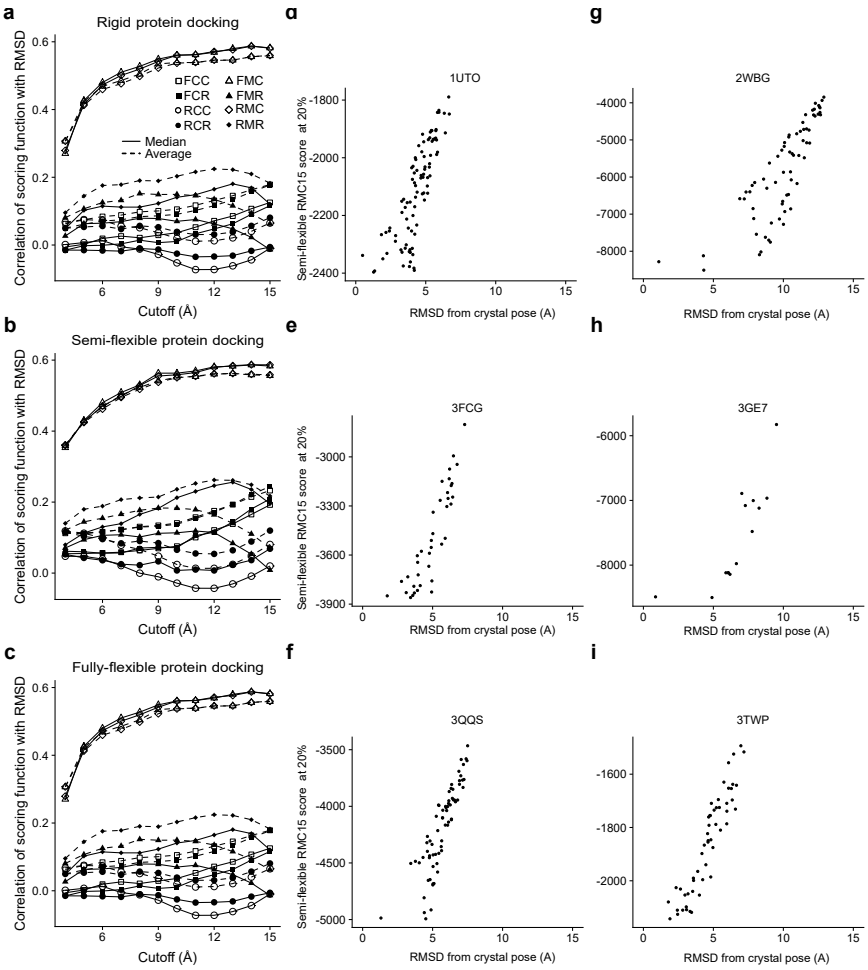


Figure 3.14. Correlations between score and the RMSD of a pose from the crystal pose for rigid protein (a), semi-flexible protein (b), and fully flexible proteins (c). The remaining plots (d-i) are of the RMC15 score of all poses produced by CANDOCK for selected proteins in CASF-2016 versus the RMSD of the pose. In these plots, the RMSD ranges from 1 Å to 15 Å The poses were obtained using the semi flexible method at a Top Seed Percent value equal to 20%.

Therefore, to determine the best minimization function, we calculated these correlations expressed as the average and the median Pearson correlation coefficients for all of the scoring functions evaluated over CASF-2016. Fig 3.14a-c shows that the RMC and FMC scoring function families have the largest correlation with RMSD (average across all cutoffs is 0.30 units greater than averages for other scoring functions).

Moreover, with an increase in the cutoff value for RMC and FMC scoring functions, the correlation also increased from an average of 0.36 at 4 Å to an average of 0.56 at 15 Å, suggesting that including long-range interactions is essential. We also show that the median and the average of these correlation values for the RMC and FMC scoring function families are relatively similar, indicating that the distribution of correlation values is not biased toward high or low correlations for any given protein in the CASF-2016 set. In addition, the RMC15 score of the experimental crystal pose has a strong correlation with the RMC15 score of the lowest RMSD pose (Fig 3.15, $r^2 > 0.99$). Finally, the pose with the lowest RMC15 score correlates well with the RMC15 score of the crystal pose ($r^2 > 0.95$). Taken together, we conclude that, using the RMC15 scoring function in the CANDOCK algorithm to calculate intermolecular forces and energies during crystal, the energy minimization of the docked protein–ligand complexes correlates well with the RMSD from ligand pose (few example cases of RMSD vs RMC15 score plots are shown in Fig 3.14d-i).

3.4.8 CANDOCK Can Reproduce the Binding Pose of a Ligand in a Noncognate Crystal Form.

To assess CANDOCK's ability to reproduce the crystal pose of a small molecule in a holoprotein bound to a different ligand (a noncognate protein form), we benchmarked CANDOCK against the PINC Is Not Cognate benchmarking set. This benchmark is divided into 12 protein targets, each having 5 crystal structures bound to a ligand and an additional set of ligands with known crystal poses in the target protein. The goal of the benchmark is to reproduce the crystal pose of the provided ligands using the five noncognate protein structures. From the 12 protein targets, we focused on the following 6 targets as they were previously identified as being difficult to dock [218]: β-secretase 1, carbonic anhydrase II, CDK2, MAPK14, PTP1b,

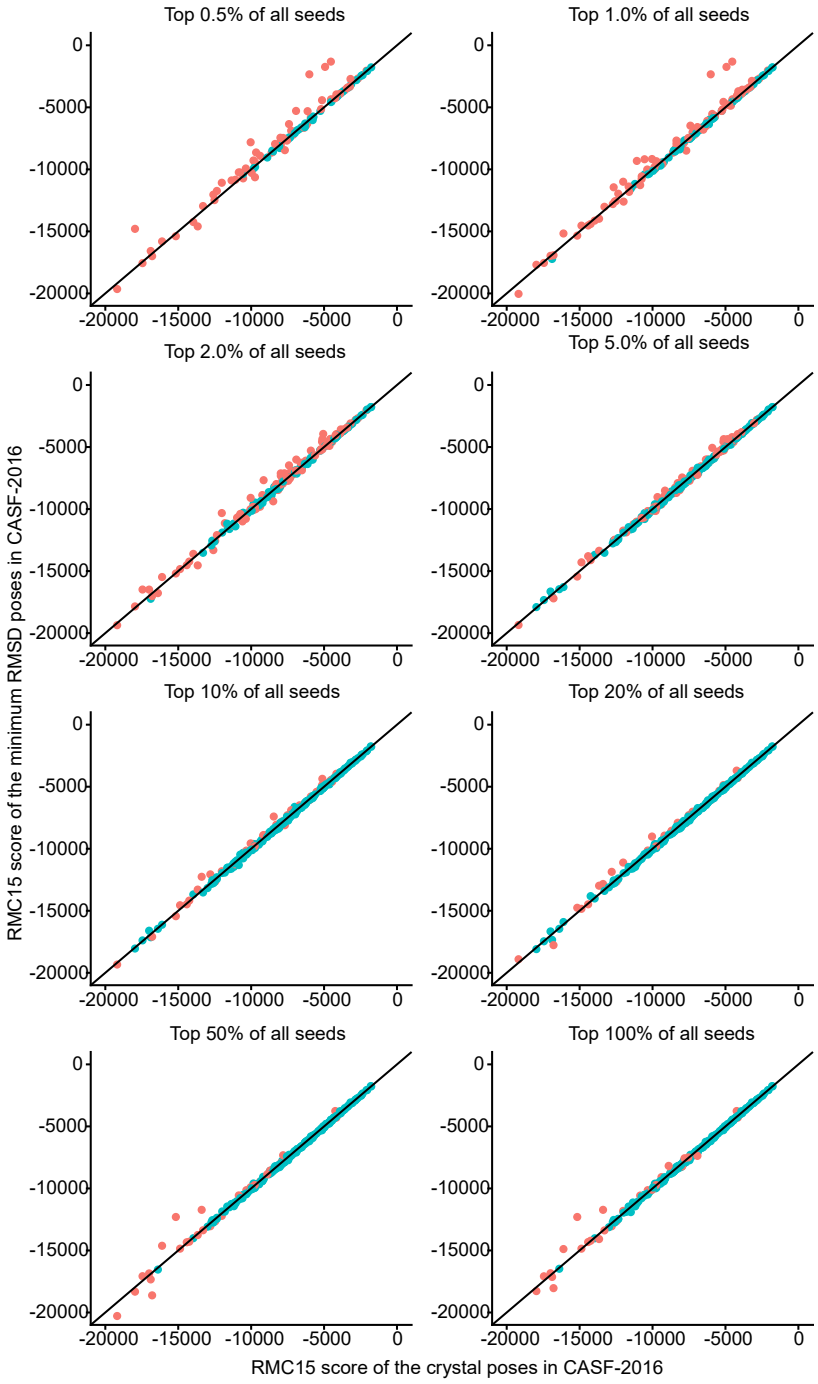


Figure 3.15. Correlations between the RMC15 scores of the crystal pose and the pose with the RMC15 score of the lowest RMSD are shown for all eight Top Seed Percent values. Poses within 2.0 Å of the crystal pose are shown in blue (successful runs) while poses greater than 2.0 Å are shown in red.

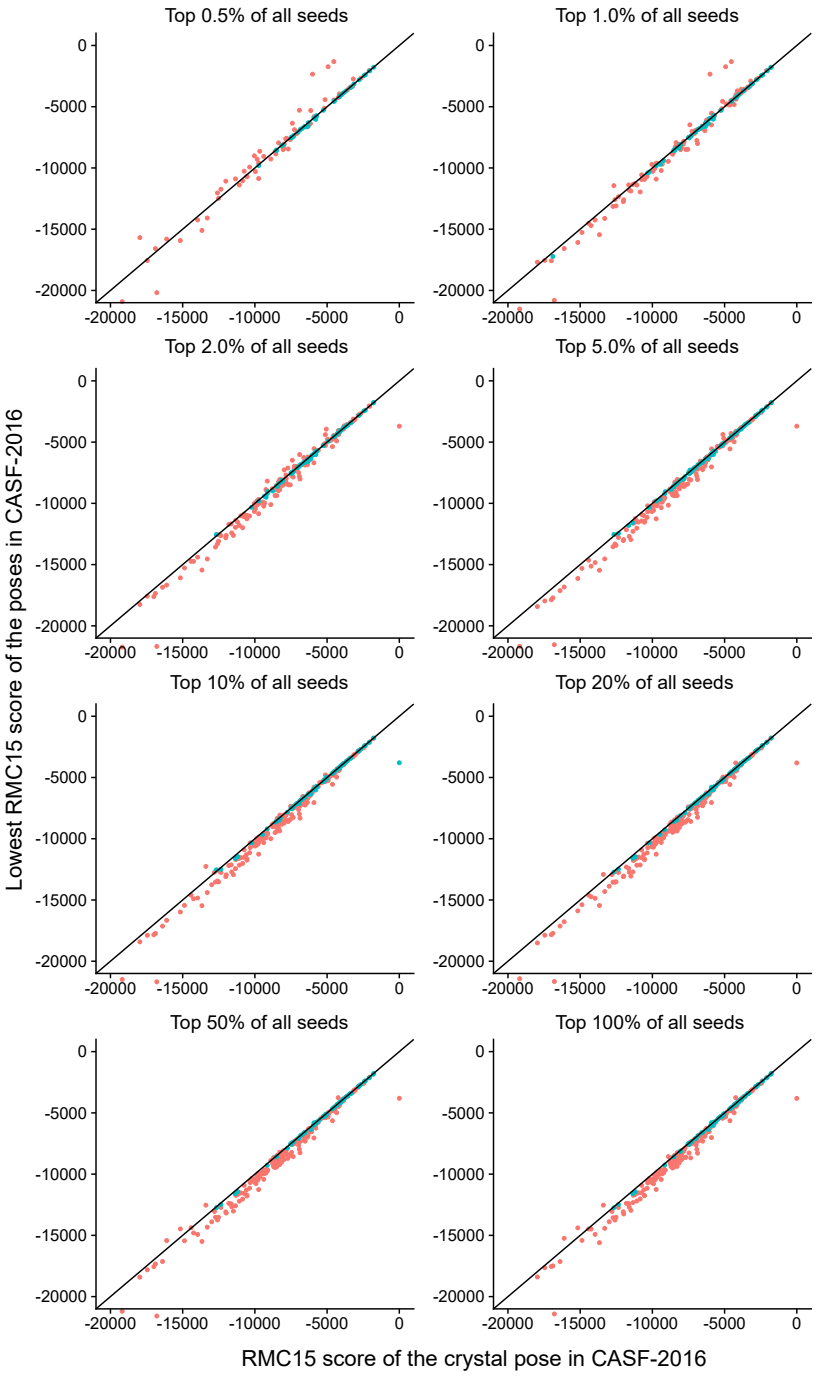


Figure 3.16. Poses within 2.0 Å of the crystal pose are shown in blue (successful runs) while poses greater than 2.0 Å are shown in red. Here it is shown that the successful poses occur only on the y=x line, while the unsuccessful poses cluster above this line. This indicates that further minimization with RMC15 may improve the RMR6 selection rate.

and PPARγ. The cumulative distributions of the best pose produced by CANDOCK are provided for each of these targets in Fig 3.17. To compare with an established docking procedure, we have also produced quantification of these results for both CANDOCK and AutoDOCK Vina [176] (cumulative distribution given in appendix F) in Table 3.3. For each protein in all targets, CANDOCK is able to produce more crystal-like proteins than Vina with the exception of proteins 1 and 2 for β-secretase 1 and protein 1 for carbonic anhydrase II. In each of the exceptions, Vina only produces a crystal-like pose for a single noncognate ligand more than CANDOCK. When CANDOCK outperforms Vina, it typically produces twice as many crystal-like poses as compared to Vina, and in one case, it produces 5 times as many poses as Vina (see Protein 2 of PTB1b in Table 3.3). When considering all five proteins for each target, CANDOCK reproduces the crystal pose for all proteins more frequently than AutoDock Vina, with the notable exception of MAPK14 where Vina is only able to produce two more crystal–like poses than CANDOCK.

A possible explanation for CANDOCK's ability to outperform Vina on the PINC benchmark is that the poses generated by CANDOCK do not depend on the input conformation of the ligand. The input ligand is fragmented and reassembled in the binding pocket, thereby removing any input conformational bias from the ligand. This allows CANDOCK to create a wide variety of ligand poses (see Fig 3.8). Conversely, Vina is dependent on the starting conformation of the ligand. For example, when we did the Astex benchmark, Vina produced a crystal pose in 93% of the target ligands when it was provided the binding pose, but only 80% when the ligand is minimized before being used as input to Vina. Therefore, we can conclude that CANDOCK is superior to Vina for generating poses in the binding site.

Table 3.3.
Number of Successes for Six Targets in the PINC Benchmarking Obtained for Both CANDOCK and AutoDOCK Vina. With the exception of MAPK14, CANDOCK is able to find a pose within 2.0 Å of the noncognate ligand with greater frequency than Vina when considering all proteins for each target.

CANDOCK

Target	**Protein 1**	**Protein 2**	**Protein 3**	**Protein 4**	**Protein 5**	**All**	**Total**
BACE1	42	33	32	31	27	76	103
CAII	73	103	107	103	105	119	128
CDK2	94	35	108	114	96	124	127
MAPK14	43	41	32	34	47	78	92
PPARγ	37	27	32	34	42	53	62
PTP1b	14	15	20	23	12	33	52

AutoDock Vina

Target	**Protein 1**	**Protein 2**	**Protein 3**	**Protein 4**	**Protein 5**	**All**	**Total**
BACE1	43	34	31	10	17	66	103
CAII	74	73	83	71	31	108	128
CDK2	42	12	45	43	35	78	127
MAPK14	43	31	32	25	35	80	92
PPARγ	10	9	13	13	10	29	62
PTP1b	11	6	4	12	9	21	52

3.4.9 Correlation between Docking Score and Binding Affinity Is Not Influenced by the Deviation of the Scored Pose from the Native Pose.

Another critical aspect of the scoring function is the ability to accurately rank the relative binding affinities of known binders to the same protein target. A stringent

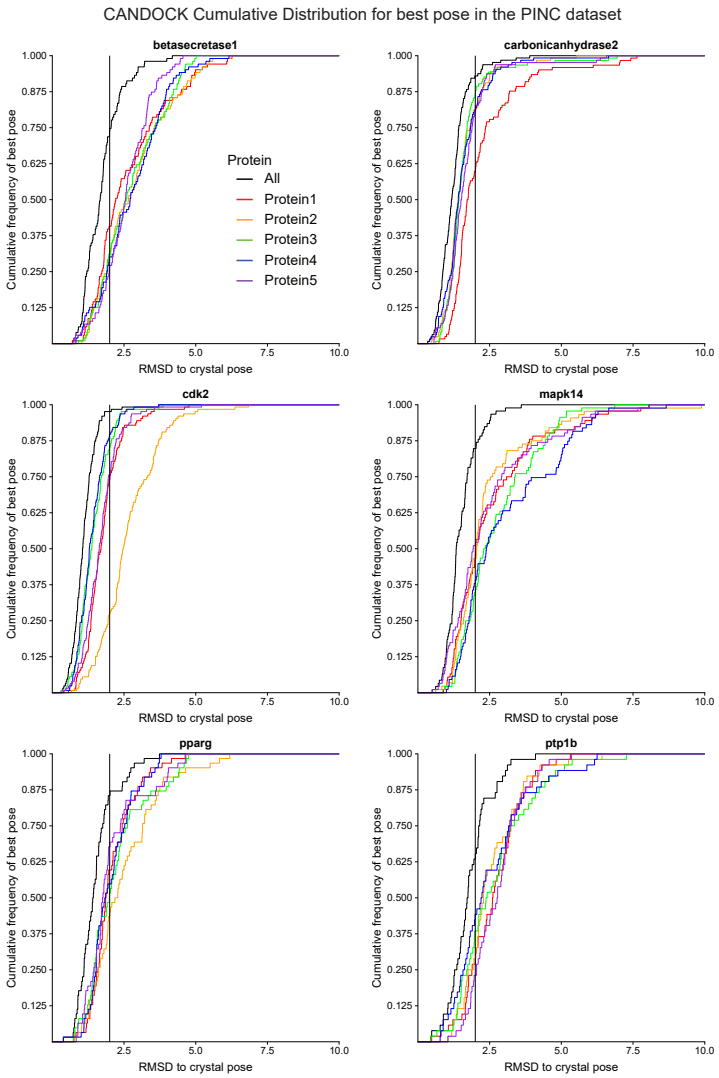


Figure 3.17. Cumulative distributions for the best pose produced by CANDOCK on the PINC benchmarking set using the top 20% of all seeds.

criterion for testing the ranking ability of a scoring function is by docking the compounds to the targets and comparing them to experimental binding affinities, i.e., without knowing the crystal pose of the ligand. CASF-2016 provides experimental binding affinities (pK_i/pK_d) and three- dimensional coordinates of 57 protein targets with 5 compounds each for a total of 285 pK_i/pK_d values for protein–ligand complexes. We determined the overall correlation between the 285 experimental binding affinities (pK_i/pK_d) with docking scores for 285 docked poses selected using each of the generalized scoring functions (docking with 20% Top Seed Percent value using CANDOCK). We found that RMR6, our best selector scoring function for selecting the crystal-like pose, does not correlate with the pK_i/pK_d values supplied by CASF-2016 with an overall Pearson correlation of -0.275 and a Spearman correlation of -0.349. When these correlations are calculated separately over 57 protein targets (each with 5 compounds) and then averaged, we get an average Pearson correlation of -0.38 and an average Spearman correlation of -0.431. This suggests a need for a different scoring function for scoring the crystal-like selected pose. Therefore, we developed a procedure (Fig 3.6) to first select the representative docked pose of a complex using a scoring function (selector) and then rank using another scoring function (ranker) to correlate with the pK_i/pK_d values. The best ranker scoring functions are RMC15 and FMC15 (Fig 3.18a,b) that were selected based on both Pearson and Spearman correlations between all 96×96 selector and ranker scoring function combinations with the experimental pK_i/pK_d data in CASF–2016. The overall Pearson and Spearman correlations for RMR6 as selector and RMC15 as ranker are -0.343 and -0.464 (correlations are -0.43 and -0.418, respectively, when averaged over 57 protein targets). It is important to note that the RMC15 score of weak binders in CASF-2016 ($pK_i/pK_d < 2.5$) does not correlate similar to other binders (Fig 3.18c,d) as removal of these weak binders improved the correlation between the RMC15 score

and binding affinity to an overall Pearson and Spearman correlation of -0.584 and -0.593, respectively.

Next, we show that there was little difference between the worst crystal pose selector (RCC11 that selects top pose 22% of the time, Fig 3.18c) and the best selector (RMR6 that selects top pose 43% of the time, Fig 3.18d) to correlate with binding affinity. The difference in the Pearson correlation for the worst (RCC11) and the best (RMR6) selectors in combination with the best ranker (RMC15) score is 0.024. Furthermore, the correlation between the RMC15 score (best ranker) and the pK_i/pK_d data for all 96 possible selectors (shown in Fig 3.18e) has a small deviation (standard deviation of 0.0829 for the average Pearson correlation). This suggests that the selection of the pose has a minor impact on ranking the activity of the ligand. This result is further supported by the section Correlation between score and binding affinity for each protein in CASF-2016 in appendix F. The results in Fig 13.19b shows that, on a class-wise basis, there is little difference between the correlations for poses selected by the lowest RMSD and the pose selected by RMR6. We find that either of these selectors does not improve the ability of the best ranker (RMC15) scoring function to rank the pK_i/pK_d data of compounds binding to the same protein. Additionally, there is little difference in the overall Pearson and Spearman correlations (0.001 and 0.004, respectively) for the lowest RMSD pose vs the best-scored RMR6 pose (selectors) that is rescored with RMC15 (ranker). While these findings are encouraging as they suggest removing the burden of finding the crystal pose of the ligand, a more detailed study with an additional benchmarking set, such as the Directory of Useful Decoys (DUD-E) [98], is required to determine the proper choice of scoring function or combinations to rank protein–ligand complexes.

To further illustrate that other docked poses in addition to the crystal-like pose contribute toward binding affinity, we calculated the correlation between the RMC15

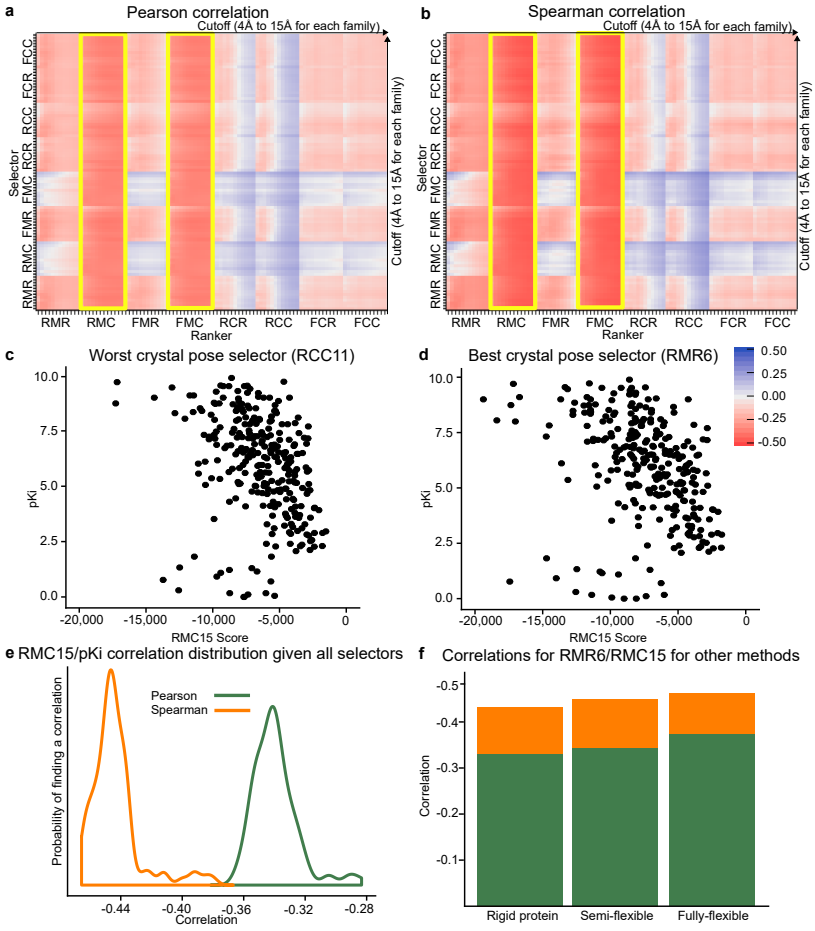


Figure 3.18. Pearson (a) and Spearman (b) correlation coefficients between all pairs of selector and ranker scoring functions (arranged by family) and the experimental pK_i of any complexes in CASF–2016. Note that a negative correlation between score and pK_i/pK_d is expected as the "p" operator introduces a negative sign to the affinity (the smaller the K_i, the larger the pK_i). The RMC and FMC (highlighted in yellow) families perform best, and there is a general trend where an increase in cutoff (from left to right) results in improved performance in ranking complexes in order of their measured pK_i. Plots of pK_i vs RMC15 score are given in (c) and (d) for the worst crystal pose selector (RCC11) and the best crystal pose selector (RMR6), respectively. The lack of major differences between these two selectors with the same ranker indicates the lack of importance in selecting the correct binding pose for ranking the pK_i of a protein–ligand complex. (e) Distribution of all correlations, regardless of selector, for the RMC15 scoring function. (f) Correlations for other docking methods with RMR6 as the selector and RMC15 as the ranker.

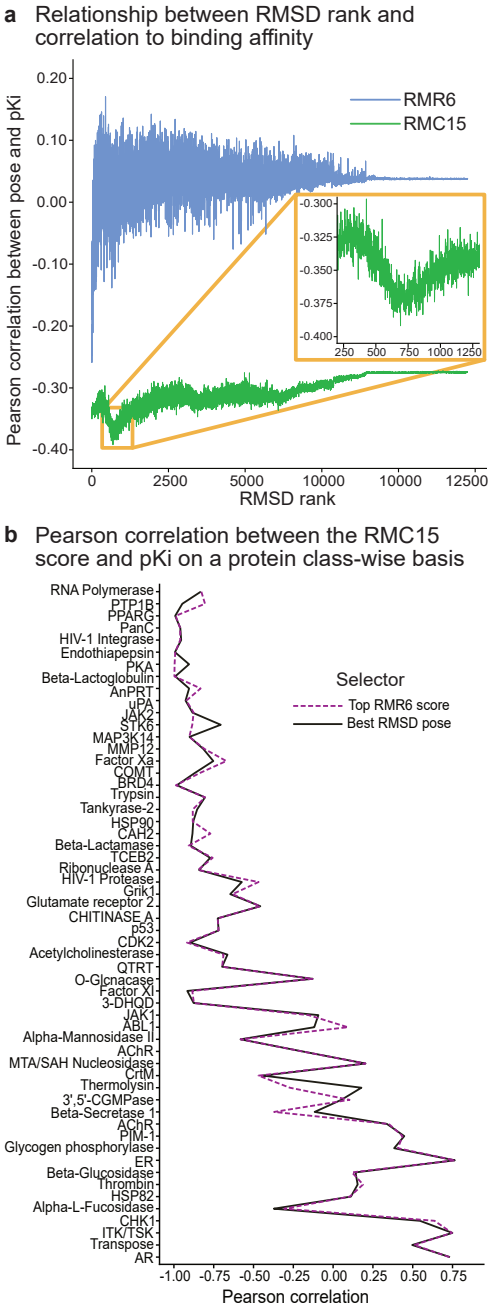


Figure 3.19. Relationship between the RMSD rank of docked poses and the overall Pearson correlation between the RMR6 (blue) and RMC15 (green) scores for CASF–2016 binding affinity of 285 protein–ligand complexes is shown in (a). An inset is used to highlight the correlation between RMC15 and binding affinity around the 750^{th} pose as ranked by the RMSD between the pose and the native pose. The class-wise correlation between the RMC15 score of a pose selected by the best RMR6 score and the lowest RMSD is shown in (b).

score and binding affinity while varying the RMSD rank used to select the pose for scoring. First, only the best RMSD pose for each of the 285 protein targets is scored using RMC15, and the correlation between this score and the binding affinity is measured. This is repeated for the second-best RMSD pose of each complex and then continued similarly for all docked poses ranked in the ascending order of RMSD from the crystal ligand. If fewer docked poses are available for any protein target than the RMSD rank, the worst RMSD pose is used. Results of this procedure for the RMR6 and RMC15 scoring functions are given in Fig 3.19a and indicate that the lowest RMSD rank does not always yield the best correlation with binding affinity for the RMC15 scoring function. In fact, the best correlation is achieved around the 750^{th} pose as ranked by RMSD (Fig 3.19a, green line and yellow inset) and other RMSD ranks also produce a similar correlation. In contrast, the RMR6 scoring function is dependent on the RMSD of the pose (Fig 3.19a, blue line) but does not correlate with binding affinity. Finally, as mentioned previously, there is no difference in correlation between the RMC15 score and the binding affinity for different protein classes using both the best RMR6 scored pose and the lowest RMSD selected pose (Fig 3.19b), suggesting that the knowledge of the crystal pose is not necessary for predicting binding affinity. We would like to stress that further investigation into these patterns is required and will be addressed in future works.

Similar to the selector used, the flexibility mode (rigid, semiflexible, fully flexible) used to generate ligand poses does not have a significant impact on the correlation between score and binding affinity (see Fig 3.18f). While the fully flexible methodology has a significant advantage for the kinases such as ABL1, JAK2, and CHK1, there are many other examples of protein–ligand complexes where the semiflexible method provides a clear advantage over the fully flexible and rigid methodologies. This is significant because the semiflexible method is less computationally demanding than

the fully flexible method and can be used efficiently in a virtual screening pipeline. Moreover, there is a large variation in Pearson and Spearman correlations between the scores, and pK_i/pK_d data have variability based on the type of protein varying from -1.0 (best) to $+1.0$ (worst), as shown in Fig 3.19b. For example, the nuclear hormone receptors ER and AR have positive correlation values instead of the expected negative ones; the best selector/ranker pair for HIV proteases in CASF-2016 is RMC15/RMR6, which is the opposite of what was found for other test cases of CASF–2016, in general. Therefore, the use of different scoring functions for different protein classes may be advantageous in ranking the relative binding affinity of the ligands to the protein targets but extensive benchmarking is needed to obtain class–specific biases.

3.5 Conclusions

We present the CANDOCK algorithm, our hierarchical atomic network-based docking algorithm that accounts for protein flexibility and ligand interactions with all cofactors, metal ions, etc. in the binding pocket using generalized statistical scoring functions. We demonstrated that these scoring functions worked very well to generate a crystal-like pose for 94% of the CASF-2016 data set consisting of 285 protein–ligand complexes. There were 17 (of 285) failures in total with semiflexible docking, which were reduced to 10 failures with fully flexible, including 4 (of 10) failures that contain long aliphatic chains. We found that the RMR6 scoring function was the best at selecting a crystal-like ligand pose and RMC15 scoring function scored the selected poses to rank ligands according to their measured binding affinities. Our algorithm only requires a final energy minimization of the protein and the ligand (semiflexible) to generate crystal-like ligand poses for ligands consisting of less than 6 fragments, compared to fully flexible methods needed for larger ligands. CANDOCK was devel-

oped to provide proper representations of ligand, receptor, and all cofactors in the binding pocket. It performs well by including ligand and cofactor interactions in the binding pocket using the generalized statistical potential and without the need for parameterization. CANDOCK successfully generates a crystal pose for 97.6% of the Astex benchmarking set (83 of the 85 complexes) that includes generating crystal-like poses for cases that failed with all popular docking methods (e.g., containing metal–organic interactions). We show that the RMR6 scoring function using a short distance cutoff and reduced atom-type set is adequate for selecting the crystal pose of the ligand. However, a longer distance cutoff and complete atom-type set used in the RMC15 scoring function are essential to achieve a reasonable correlation between the docking score and the RMSD of a docked ligand from the crystal ligand, which justifies the use of RMC15 as the minimization function. The RMC15 scoring function was also the best at reproducing reasonable correlations between scores and ligand binding affinities. We believe that the release of the CANDOCK algorithm will give the community a valuable freely available tool for generating chemically relevant ligand poses for use in drug discovery efforts. The hierarchical nature of our method presents a powerful and flexible tool to perform proteome-wide docking studies efficiently, yielding an improved drug discovery and design pipelines. We have placed all of the scripts and input protein and ligand structures required to reproduce our results at *github.com/chopralab/candock*$_b$*enchmark*.

3.6 Future work

A major future work of this project is improving the success rate and correlation with binding affinity of the scoring function. A potential strategy for achieving these improvements is to partition the scoring of different related proteins together. To do this, one can partition the PDBBind benchmark into 27 different classes and using a

machine learning model trained to select crystal poses for each of these proteins. An overview of this procedure is given in Fig 3.20. The selection of a machine learning model should be done carefully as to take advantage of properties of docking as the majority of machine learning based scoring functions used generic ML methods and do not offer substantial improvements in this field. An introduction to machine learning methods is provided in appendix A.

Upon doing so, one can notice that different protein classes have different performances when using a different scoring functions (Fig 3.21. Additionally, ligands consisting of 4–7 fragments produce the most number of poses and Factor XA produces the highest number of conformations while GPCRs produce a significantly fewer number of poses. The selection rate, is dependant on both the conformation search space and the class of protein. For example, our method performs very well on oligopeptide binders, but poorly on carbonic anhydrase. Finally, if the search space is too small, then too few poses are present for ranking. After a search space of 5%, there is no significant increase in the success rate. These results show that different protein classes have significant differences in how they should be scored.

Next, we decided to investigate which scoring functions would work best for a given protein class. Previously (and in this chapter), we determined that the RMR6 scoring function is best at selecting a pose overall. As can be expected from the previous result, the ability of scoring function to select a pose is dependant on the protein class (Fig 3.22b).

Given these findings, we suspected that the properties of alike conformations can be used to improve the selection rate of a scoring function. For simplicity, we simply decided to cluster the docking results of completed CANDOCK jobs, but a future work should integrate these steps into the CANDOCK algorithm itself. This clustering yields the number of alike conformations for a given pose and has been labeled

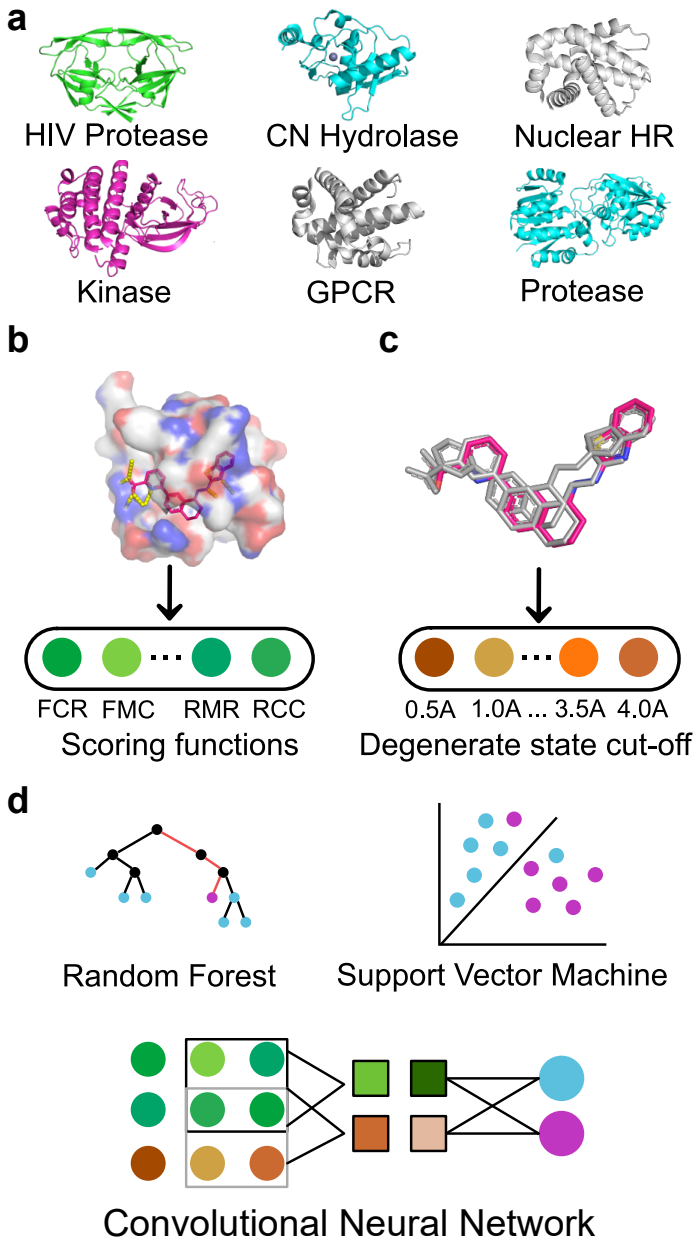


Figure 3.20. Overview of the docking methodology presented in this work. (a) Protein classification is performed on the target protein to assign it to a single class out of 27 possibilities using the Enzyme Classification (EC) and the Gene Ontology (GO). (b) Clustering to identify conformationally degenerate poses (c) as to calculate the conformational entropy for all poses. (d) The knowledge-based score and the conformational entropy are used as features in a machine learning based selection procedure.

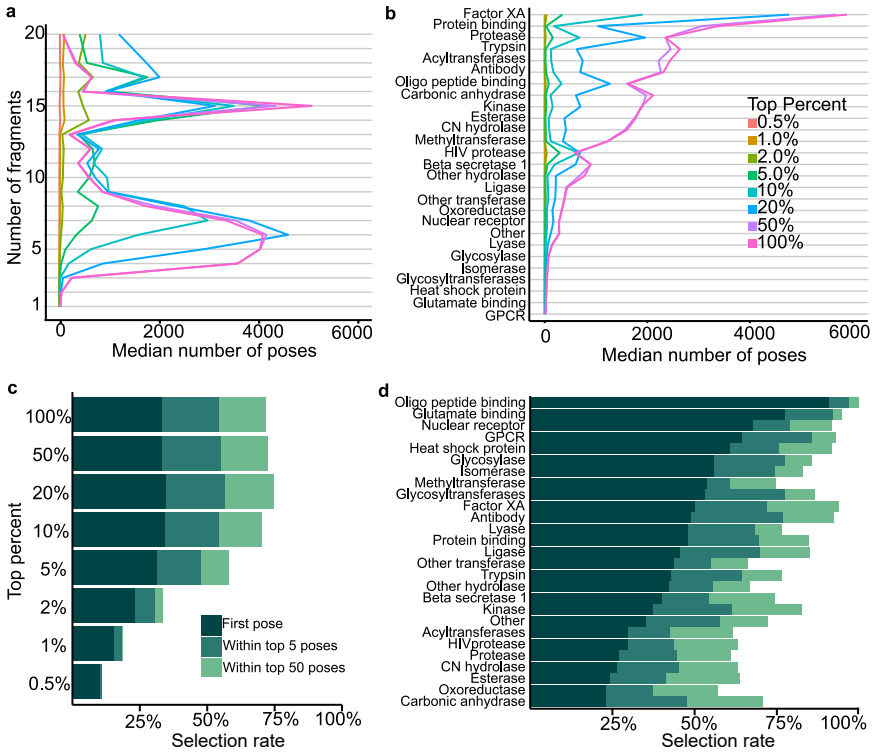


Figure 3.21. Importance of class dependent docking. (a) Number of poses generated for each protein class (b) Effect of protein class on the number of conformations generated. (c) Effect of Top percent on success rate. (d) Effect of protein class on selection rate.

conformational entropy. The average conforational energy is in fact dependant on the protein class as shown in Fig 3.22d. Since the crystal pose appears to be somewhat related to the conformational entropy of the pose, we decided to include it as a feature in future ML models.

Table 3.4.
Success rates resulting from the test–set benchmarks for various methods of selecting the crystal pose.

	No classification	Classification
Scoring Function	34.3 %	42.8 %
Neural-Network Methods	41.1 %	53.6 %
Random Forest and SVM Methods	41.1 %	52.7 %
Machine Learning Overall	41.1 %	57.0 %

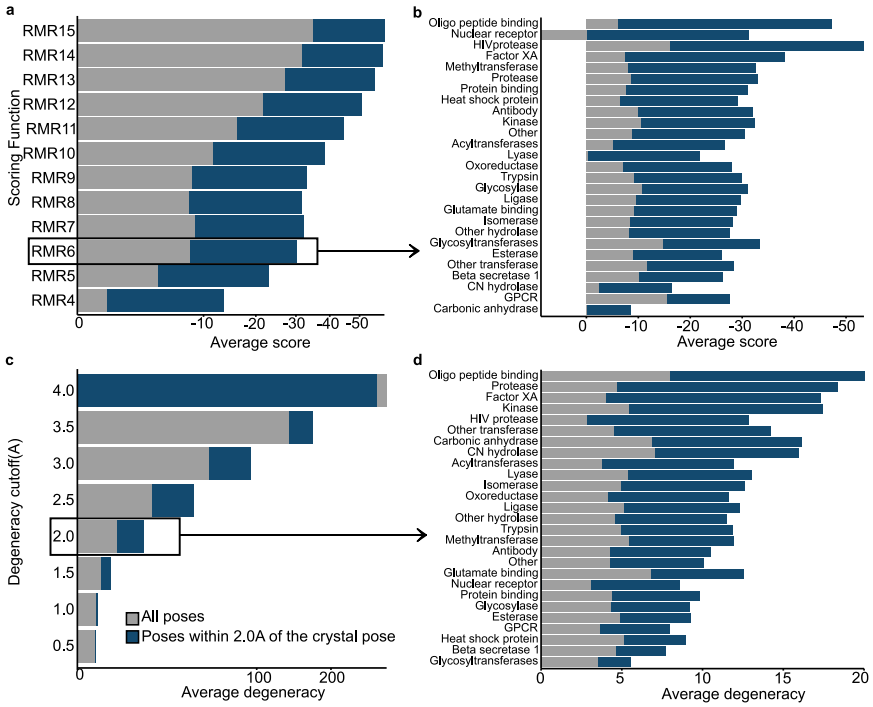


Figure 3.22. (a) The average value for the given scoring functions is shown for all poses and poses with 2.0Åof the crystal pose. (b) Ligand-protein scores calculated using the RMR6 scoring function, averaged in a class specific manner. The plot is arranged so the difference in the average for all poses versus poses near the crystal pose is decreases from the top of the figure to the bottom. (c) The average degeneracy for all poses and poses near the crystal pose for all RMSD cutoff values. For the 2.0 Åcutoff, the class-specific degeneracy averages are provided in a similar manner to (b).

The results of training various machine learning models are shown in Fig 3.23. In Fig 3.23a, these results are shown for various different methodologies as shown with various colors. The training of these methods was done using four different methodologies: (blue) using all poses from all Top Percent docking runs with individual models created for each class, (green) using poses only generated with the best Top Percent value select for the given class with individual models created for each class, (red) all poses used similar to (blue), but a single model is created for all classes, (yellow) poses are selected in a manner similar to (green), but a single model

is created for all classes. However, each of these methods have different success rate for different protein classes, which should be expected given the conclusions of the previous figures. This allows method to select the proper machine learning model for a given class, an yield a selection rate of 57% (Table 3.4). Unfortunately, this number is still comparable to the best docking scoring functions and further work is needed to boost it further.

Representative docking poses for oligopeptide binders is shown in Fig 3.24(a)-(c) with the top five single scoring function selected poses in blue and the crystal pose given in green. These poses indicate that the binding site for this class of proteins in quite small and fits around the ligand in manner that few poses other than the crystal pose are possible. Therefore, if a pose is generated then it is highly likely to be the pose in question. Thus, it is reasonable that a single scoring function is able to perfectly select the correct pose. What is interesting, however, is that the machine learning based pose selection method performs worse than the scoring function alone, which can attributed to training error due to the fact that the new selection model will not be perfectly able to reproduce the results from the single scoring function. Fig 3.24(d)-(f) The presence of a Zinc ion commonly present in the acetyl transferase class causes the single scoring function method to not place a ligand in the proper location with respect the ion. This causes out models to not predict the proper location of the ligand and shifts the ligand away from its proper binding location. The combination of other scoring functions allows the machine learning based pose selection method to select the pose where the distance between the ion and ligand is correct. Fig 3.24(g)-(i) The ligand poses selected by the single scoring function method for the CN hydrolase class are flipped from the crystal ligand pose. It is interesting to note that ligands near the crystal pose for the CN Hydrolase class have higher degeneracy values than those far from the crystal pose. This indicates that

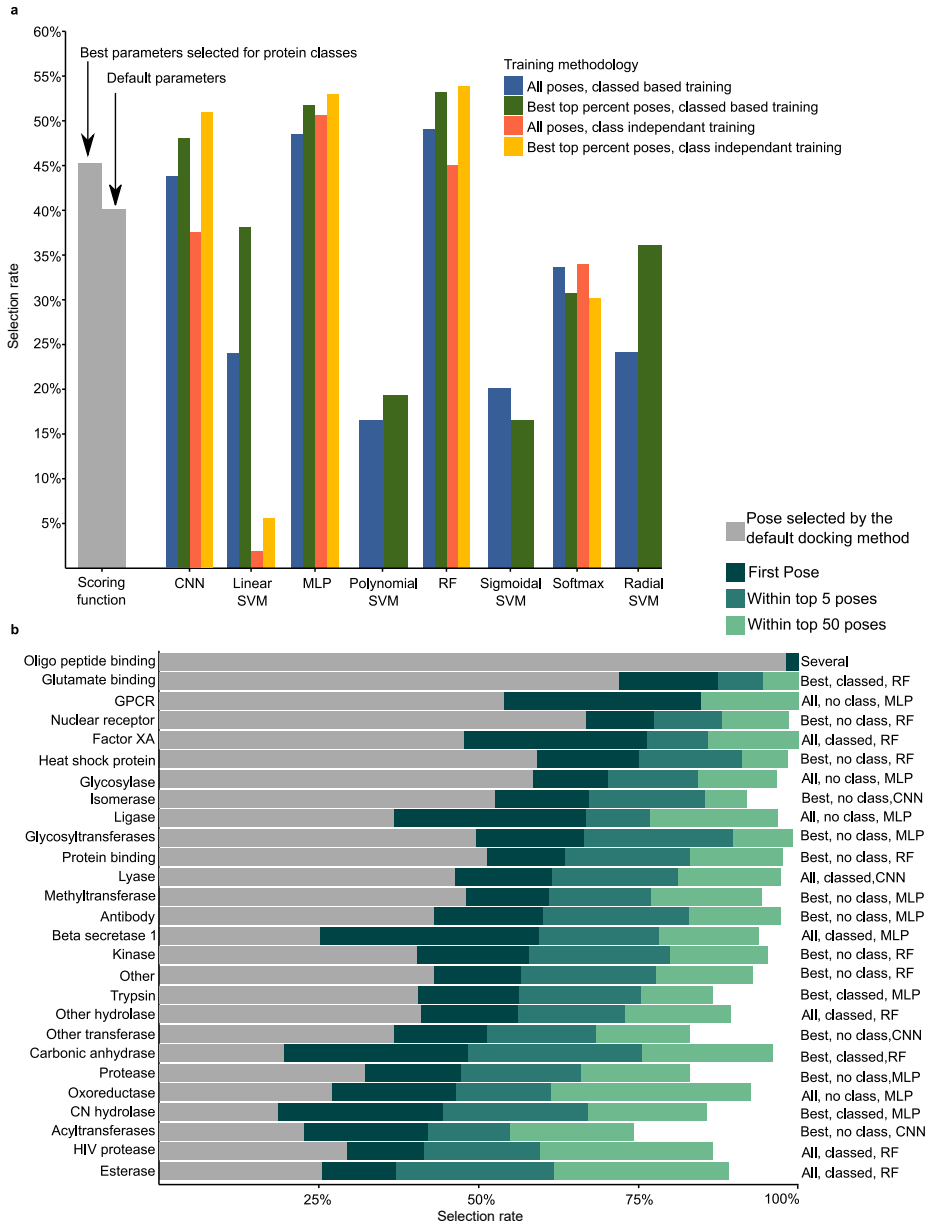


Figure 3.23. Advantage of class specific machine learning (a) Success rate for the various machine learning methods employed in this work. The success rates for a single scoring function are given in grey for reference. (b) Success rates are shown for the methods that perform best on a given class.

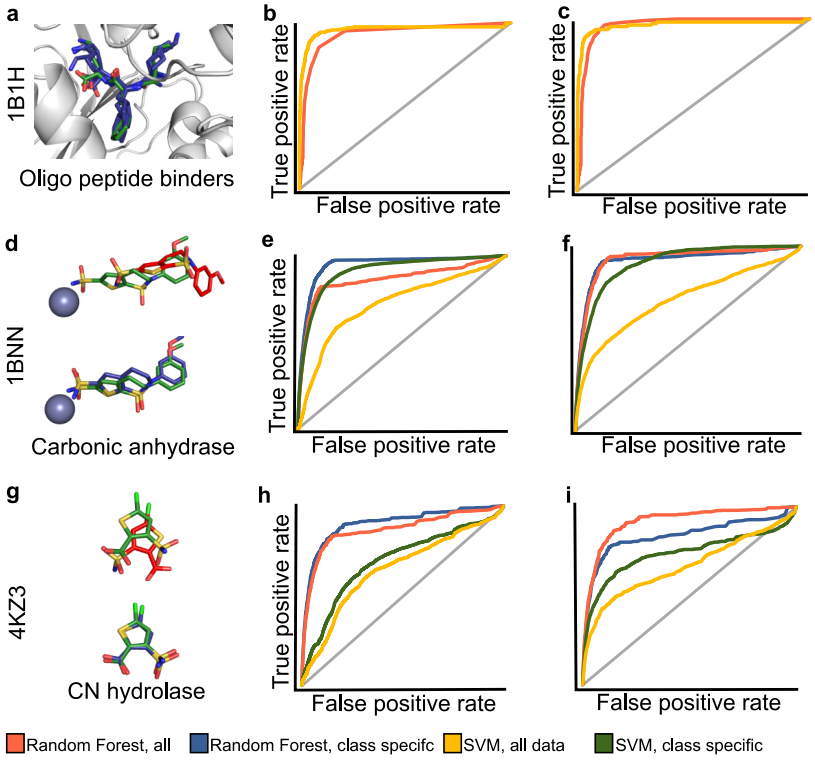


Figure 3.24. Representative docking poses are shown for oligopeptides, Carbonic anhydrases, and CN hydrolases.

the binding of ligand for this class are dependent on multiple conformations being present for binding to occur.

4. SMALL-MOLECULE DESIGN: DETERMINATION OF FUNCTIONAL GROUPS

This chapter is available as

Fine, J., Rasjashekar, A., Jetheva, K,. Chopra G. Spectral Deep Learning for Prediction and Prospective Validation of Functional Groups. *Chemical Science*, **2020**, Advance Article. DOI: 10.1039/C9SC06240H

4.1 Abstract

State–of–the–art identification of the functional groups present in an unknown chemical entity requires the expertise of a skilled spectroscopist to analyze and interpret Fourier Transform Infra-Red (FTIR), Mass Spectroscopy (MS) and/or Nuclear Magnetic Resonance (NMR) data. This process can be time-consuming and error-prone, especially for complex chemical entities that poorly characterized in the literature, or inefficient to use with synthetic robots producing molecules at an accelerated rate. Herein, we introduce a fast, multi-label deep neural network for accurately identifying all the functional groups of unknown compounds using a combination of FTIR and MS spectra. We do not use any database, pre–established rules, procedures, or peak-matching methods. Our trained neural network reveals patterns typically used by human chemists to identify standard groups. Finally, we experimentally validated our neural network, trained on single compounds, to predict functional groups in

compound mixtures. Our methodology showcases practical utility for future use in autonomous analytical detection.

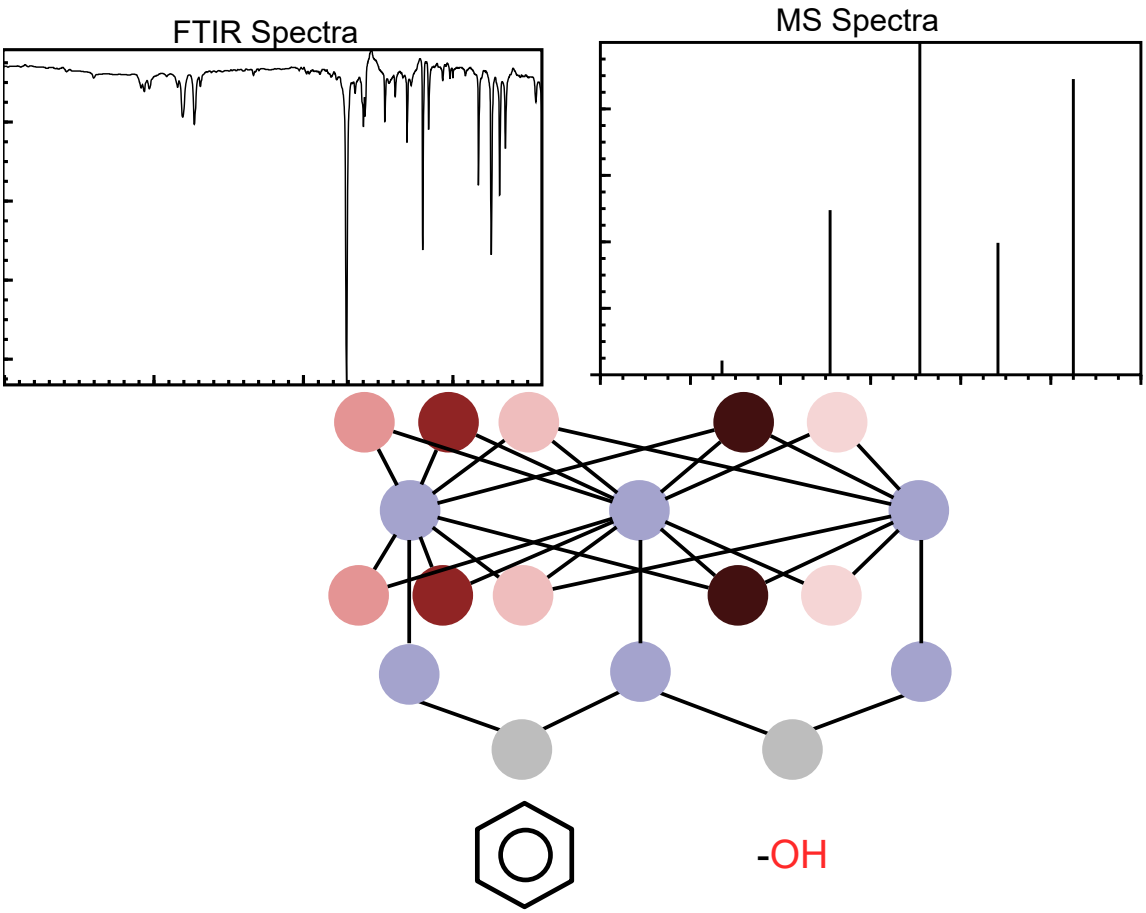


Figure 4.1. Table of contents figure for the online publication

4.2 Introduction

The arrangement of atoms within a molecule dictates its physical, chemical, and spectral properties. Small discrete, or large repeating arrangements of atoms which give rise to measurable changes in a molecule's reactivity [233–235], boiling point [236, 237], melting point [238, 239], and other characteristics are called functional groups.

Given the structural formula of a molecule, a chemist can identify functional groups present (e.g. aldehyde, carboxylic acid, alcohol, etc.) and can postulate characteristic reactivity and physical properties for a given molecule based on the presence of these groups. Therefore, the identification of functional groups present within an unknown compound is a key step in qualitative organic synthesis and structure elucidation; it is routinely practiced by chemists to validate the synthesis of novel small molecules or identify unknown structures in complex mixtures. Techniques for assigning functional groups based on "rules of thumb" or by matching profiles from known databases are commonly applied in organic chemistry [240], metabolomics [241, 242], and forensic sciences [243–245]. Furthermore, monitoring of functional group changes can be used to determine the progress of a reaction [246], and can even be used to identify the components of complex mixtures for a reaction coordinate.

Chemists often rely on spectroscopic techniques like Fourier Transform Infrared (FTIR) spectroscopy, Mass Spectroscopy (MS), and Nuclear Magnetic Resonance (NMR) spectroscopy for the assignment of functional groups. FTIR utilizes the frequencies associated with the bonds in a molecule, which typically vibrate around $4000cm^{-1}$ to $400cm^{-1}$, known as the Infrared region of the electromagnetic spectrum [240]. This region is associated with specific frequencies that change the oscillating patterns of chemical bonds in the analyte, resulting in an FTIR spectrum [247]. Typically, a spectroscopist manually analyzes this spectrum to identify patterns corresponding to a given functional group using previously established rules and principals [240], a time-consuming process subject to human bias and interpretation. Alternatively, if the compound has previously been characterized, the spectroscopist can use software to match the peaks of the analyte to a database of known compounds for identification [248].

Mass spectroscopy (MS) is another technique commonly used by chemists for the identification of unknown compounds [240]. One of the first, and still a popular MS ionization technique is electron ionization (EI–MS) [249], a method performed by bombarding the analyte in the gas phase with high energy electrons (70 eV) for molecular ionization. The resulting cationic radicals are energetically unstable and break apart, resulting in smaller charged particle fragments that are specific to the analyte. Such fragmentation patterns are dependent on molecular functional groups and their arrangements with other functional groups and motifs. The abundance of fragments with a given mass to charge ratio (m/z) is recorded and reported as the mass spectrum. These spectra are used to search through a database of MS peaks of known compounds, but large-scale automated identification of unknown molecules is still a major challenge [241, 250–252]. In addition, a popular tandem mass spectrometry (MS/MS) method, namely collision-activated dissociation (CAD) has been extensively used for the characterization of complex mixtures [253, 254]. For CAD, the analyte ions are accelerated and allowed to collide with an inert gas for fragmentation and subsequent MS/MS analysis. Furthermore, in addition to EI–MS and CAD based fragmentation soft ionization techniques such as electrospray ionization mass spectrometry (ESI–MS) have been developed. For ESI–MS, the analyte is sprayed through a spray needle into a carrier gas chamber where an electric field is applied to charge the analyte. This is then passed into a heated capillary which desolvates the analyte, forcing it into the gas phase. Since ESI–MS is a soft ionization method, it is possible to perform repeated charging of the analyte with no fragmentation due to ionization. With repeated charging, the (m/z) values of the resulting ions become lower and detectable. This has been used to determine biomolecular structures, atomic interactions, post-translation modifications, protein sequence information, and has been extended to inorganic, organic, and metal–organic complexes [255]. However,

high-performance liquid chromatography is typically used for molecular fractionation prior to mass–spectrometric analysis to identify the structure of unknown constituents in complex sample mixtures [255].

Human intervention to analyze FTIR or MS spectrum is useful but achieving the next generation of autonomous instrumentation for reaction screening requires a completely automated method for determining whether a reaction occurred. The current approaches to automating functional group identification are similar to those applied by humans, using a set of rules and pattern (peak) matching to map spectra to a functional group [251, 256]. Such methods typically utilize only selected spectral regions to identify functional groups, and often afford relatively low confidence predictions owing to a limited database of known compounds18. Furthermore, to our knowledge, these methods can only incorporate data from a single spectral technique (i.e., either FTIR or MS) and ignore relationships between different spectral data for identification. Hence, there is a need for automated and accurate methods capable of multiple-spectra integration without the use of pre-established patterns on known databases. Such methods will need minimal-to-no human intervention, progressing chemistry towards the realization of automated synthetic robots that screen functional groups and combine spectral data to validate each step during reaction screening and multi-step automated synthesis [257]. The state-of-the-art robot for automated reaction detection currently employs different techniques to determine the presence of a reaction [258], but only predefined compounds can be identified. It is a major challenge to develop fully automated robots to discover new reactions that produce unexpected products. Our goal is to extend the capabilities of these automated synthetic robots by developing a fast, automated methodology for functional group determination that can be used in real-time, thereby enabling reaction

screening through the identification of functional group changes in a database-free manner.

Machine Learning (ML) is a set of techniques used by computers to perform a specific task without an explicit set of instructions provided by the user. ML techniques have been successfully applied to multiple chemical problems in recent years, and still show promise for the advancement of several areas of chemistry. Popular machine learning architectures, such as Random Forest [150–152], Multiple Layer Perception [153–155], Generalized Adversarial Networks [156–160], and Recurrent Neural Networks [161, 162, 259] have been used on chemical data for small-molecule design [144, 145], metabolism [260, 261], toxicology [163, 260], photo-electric properties, solubility, and retrosynthesis [162, 259]. It has been shown that direct molecule as a subgraph of groups of atoms (i.e., functional groups) has distinct advantages over fingerprinting methods [262, 263]. The representation of a molecule or dataset can be reduced to a lower-dimensional latent space by using an autoencoder [145]. Here, we also used an encoder to create a corresponding latent space based on spectra to predict functional groups which may also be useful to design molecules for specific spectral properties. A few ML techniques to analyze spectra has been used previously [264–268] but such attempts for function group prediction used only one type of spectral data, the training data was specific to the application, and classified groups separately as a multiple binary classification problem [267, 268]. Binary classifiers are not optimal for a large number of classes and are sensitive to class imbalances during training resulting in problems identifying all functional groups in a molecule or mixtures [261, 269]. In this work, we present the first ML method, to our knowledge, that integrates FTIR and MS data to obtain a combined set of features as a multi-class, multi-label classification methodology. Our method predicts multiple functional groups for a given molecule in a database-free manner, as compared to

identifying a molecule through peak matching or only identifying the major functional group in the molecule (Fig 4.2). In this work, we have also outlined a framework to measure the success of such a multi-label neural network by introducing molecular F1 score and molecular perfection rate metrics. We hope that others will build-upon our suggested framework and methodology to catalyze further development of functional group identification methods for accurate and autonomous molecular structure elucidation.

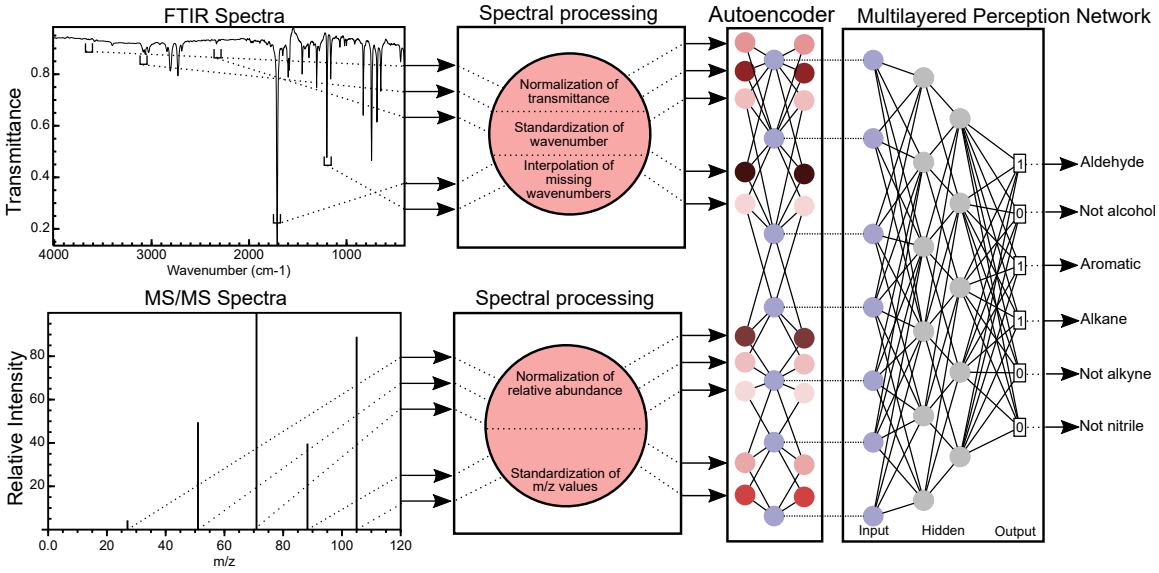


Figure 4.2. Overview of the MLP methodology for the classification of functional groups using FTIR and MS data. FTIR spectra are processed as to normalize the transmittance of the spectra and discretize the wavenumber numbers (creating wavenumber bins), thereby standardizing the wavenumbers for all FTIR spectra. Missing wavenumber bins in each spectrum are interpolated using B–Splines. A similar process is used for mass spectra data with the exception that no interpolation is performed. The normalized transmittance in all bins is encoded into a latent space by an autoencoder network and This latent space this then used to predict the functional group of a molecule.

4.3 Methods

4.3.1 Collection of training data

We obtained both FTIR and MS spectra from standard reference spectra published by the United States National Institute for Science and Technology [270] for 7,393 compounds and standardized these spectra using the procedure described in the supporting information under Standardization of FTIR spectra and Standardization of MS spectra.

4.3.2 Training of Neural Networks

We used a 3 layered Multi-layered Perceptron (MLP) network using binary cross entropy as the loss function to allow for multi-label prediction of functional groups. The ReLU activation function was used to introduce non-linearity between layers of the network along with dropout regularization and batch normalization to combat overfitting. To train the weights of the model, we applied the Adam optimizer. We applied Five-fold cross validation was used to ensure a model without overfitting and with minimal bias to training data. All reported validation metrics are averaged over 5 folds and the best hyperparameters were chosen based on these validation metrics. For the autoencoder, a linear autoencoder with an embedding layer of 256 dimensions was used to encode the spectra. Learned encodings were then given as input to the neural network. Autoencoder helps in removing redundant information and noise from data. Additional details on training and optimization of the neural networks presented in this work are mentioned in supporting information section titled Training and testing of neural networks.

4.3.3 Assignment of functional groups

We obtained IUPAC InChI strings for all compounds of interest by resolving the CAS number associated with the molecule using the PubChem API [271]. Then, RDKit [272] performed substructure matching on each string via SMARTS strings to identify the presence of a predefined molecular topology. If a match for a functional group's SMARTS was found, then the compound was deemed a member of the given functional group, and each SMARTS string was tested independently. Therefore, multiple functional groups could be assigned to a single molecule. Initially, we picked functional groups common between those discussed in the previous works [264, 267, 268]. These functional groups were chosen to mirror those typically identified using FTIR such that the machine learning model can be analyzed to gain insights from learned chemical patterns, as traditionally done by human chemists. However, it should be noted that more abstract definitions of functional groups can be used in future works. After training our initial model and analyzing the results (see Compounds that fail to be appropriately predicted show chemical patterns), we decided to add more functional groups to our model to attempt to improve our results. The SMARTS strings used for both models discussed in this work are shown in Table 4.3.3 and the distribution of functional groups are given in Fig 4.3.

4.3.4 Calculation of a Molecular F1 metric

Since correct assignment of all functional groups in a single molecule is paramount to the analysis of organic reactions, we have devised a single metric to quantify the predictive capability of our models versus the performance on individual functional groups. Therefore, the focus of our optimization methodology is to create a model that maximizes this overall accuracy measure as opposed to the accuracies of individual

Table 4.1.
SMARTS strings used to identify the presence of a functional group given the 2D topology of a molecule.

Functional group	**Smarts String**
Alkane[a]	[CX4]
Alkene	[$([CX2]=[X2])]
Alkyne	[$([CX2]#C)]
Arene	[c]
Ketone	[#6][CX3](=O)[#6]
Ester	[#6][CX3](=O)[OX2H0][#6]
Amide	[NX3][CX3](=[OX1])[#6]
Carboxylic acid	[CX3](=O)[OX2H1]
Alcohol	[CHX4][OX2H]
Amine	[NX3;H2,H1;!$(NC=O)]
Nitrile	[NX1]#[CX2]
Akyl halide	[CX4][F,Cl,Br,I]
Acyl halide	[CX3](=[OX1])[F,Cl,Br,I]
Ether[b]	[OD2]([#6])[#6]
Nitro[b]	[$([NX3](=O)=O),$([NX3+](=O)[O-])][!#8]
Methyl[b]	[CH3X4]
Alkane[b]	[CX4;H0,H1,H2]

[a] The definition of alkane changed between functional group sets due to the introduction of the methyl FG.
[b] not present in the original set of functional groups.

functional groups. Similar to the concept of an F1 measure, this metric normalizes the performance when the classes (functional groups) are unbalanced. Hence, we have termed this metric as the 'Molecular F1 score' as it describes the success of the model on the whole molecule. This number is calculated for each molecule in the validation set by calculating a 'Molecular Precision' and 'Molecular Recall' value for the functional groups predicted for a given molecule. Precision is the number of functional groups predicted correctly (true positives) divided by the total number of functional groups predicted to be present (the sum true positives and false positives). Molecular recall is the number of functional groups predicted correctly divided by

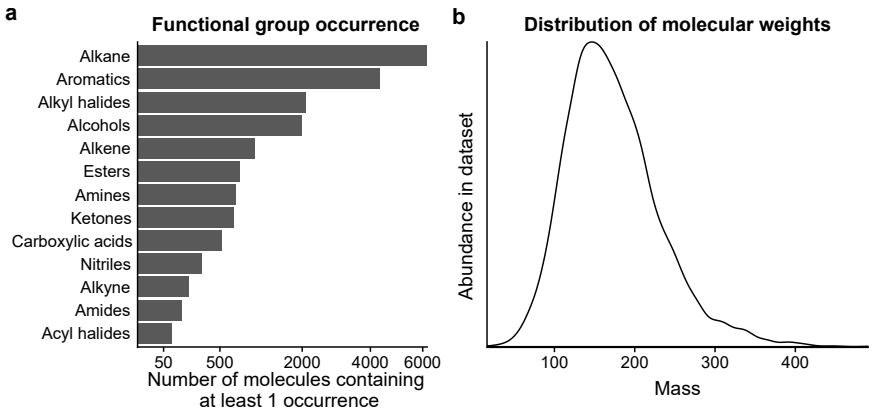


Figure 4.3. (a) The distribution of various functional groups in the NIST database. (b) The distribution of molecular masses present in the NIST database.

the total number of actual functional groups present in the molecule (the sum of true positives and false negatives). Similar to the calculation of an F1 score for given functional groups, the Molecular F1 is the harmonic mean of the Molecular Precision and Molecular Recall. The overall Molecular F1 score for a given validation set is the arithmetic mean of all Molecular F1 scores. The difference between the Molecular F1 and Functional Group F1 is illustrated in Fig 4.4.

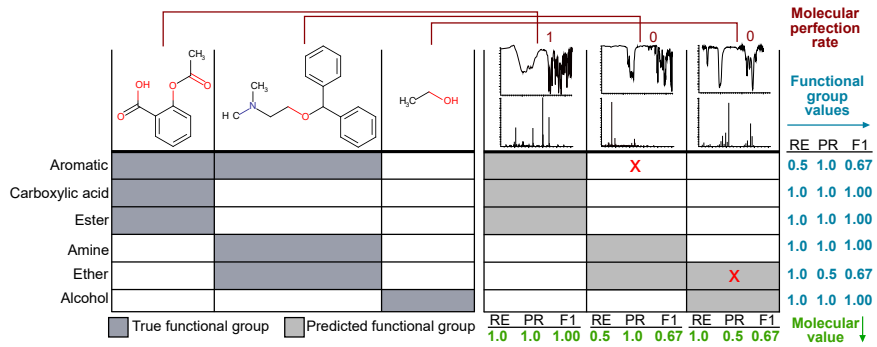


Figure 4.4. The left-hand side of the figure depicts the ground truth functional groups present in the example molecules, and the right-hand side are example predictions of the predicted functional groups given only their FTIR and MS spectra. Sample calculations for functional group F1, MF1, and MPR score are given in the figure.

4.3.5 Calculation of a Molecular Perfection Rate metric

While the knowledge of overall Molecular F1 score is useful for comparing models to one another, it does not represent the more stringent criterion of whether a given method produces all functional groups within a given molecule without error. Therefore, we have devised a second metric termed 'Molecular Perfection Rate' to rigorously measure the accuracy of our model on a per molecule basis. To calculate this metric, we compare the known functional groups to the predicted functional groups. If the predicted functional groups perfectly match the defined functional groups of the target molecule, then the molecule prediction pair is assigned a Molecular Perfection of 1; otherwise, it is assigned a Molecular Perfection of 0. The 'Molecular Perfection Rate' for each validation set is calculated as the sum of all individual 'Perfections' values divided by the total number of molecules. This metric can also represent the percentage of all molecules with a Molecular F1 score of 1.0, as shown in Fig 4.4.

4.3.6 Creation of synthetic models

A control was developed for the addition of new functional groups and termed as 'synthetic models'. They are created using a predefined accuracy to assign functional groups. To generate a synthetic model, one takes the original functional group matrix (where columns are functional groups and rows are molecules) and predicts each functional group for every molecule individually based on random numbers. The accuracy of each synthetic model is fixed, and the predictions are randomly assigned as correct or incorrect to obtain the specified accuracy. Unlike a truly random model, the synthetic model has access to the original functional group assignment matrix and the predictions of the matrix are not randomly assigned but are instead 'purposefully' correct or incorrect based on a uniform random distribution. For example, consider a

synthetic model that has an accuracy of 50% and is being generated for 4 functional groups. It is given a molecule where only the first 2 functional groups are present ([1,1,0,0]). Four random numbers are generated using a uniform distribution, e.g.: 0.25, 0.75, 0.85, and 0.10. Since the second and third random numbers are greater than the assigned accuracy (0.50), they are deemed incorrect and the model will predict ([1,**0**,**1**,0]). This example has a molecular recall of 0.5, a molecular precision of 0.5 and molecular perfection of 0.

Synthetic models with accuracies of 99, 95, and 90% are given below showing a decrease in MPR with an increase in the number of functional group predictions.

4.4 Results and Discussion

4.4.1 Multi-layer perceptron neural networks outperform Random Forest classifiers.

We performed an initial computational experiment to determine the choice of a machine learning method with the best performance to identify functional groups without doing extensive model optimization. We selected Random Forest (RF) and Multi-layered Perceptron (MLP) to test on FTIR spectra to determine if there was a need for using neural networks (MLP) as compared to ensemble methods (RF). An unoptimized MLP consistently outperformed RF models (Fig 4.5) with an average functional group F1-score of 0.771 for the MLP model compared to 0.650 for RF. We trained the MLP to predict all functional groups simultaneously as one multi-label classifier. In order to evaluate the effect of transfer learning that has been previously done for MLP [260,261,269], we also evaluated 13 binary classifiers in addition to the 1 multi-label network. The binary classifier approach did not improve the performance of the MLP model significantly as these models only produced an improvement in

functional group F1 score of 0.006 over the multi-label model, suggesting that transfer learning is not a significant factor in the multi-label network.

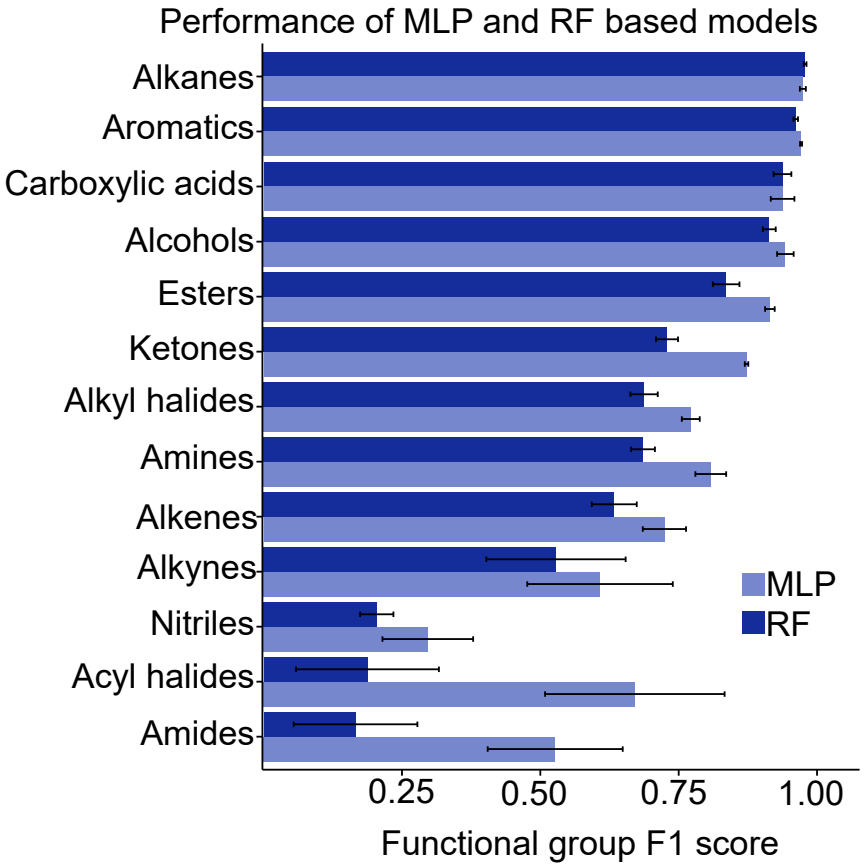


Figure 4.5. The comparison of Random Forest and Multi-Layered Perception validation set performance for the selected functional groups indicates that the MLP methodology outperforms RF for the majority of functional groups. Both methods were trained on the FTIR spectra only and no hyperparameters were used to optimize the model. Each bar represents the mean of a 5-fold cross-validation, and the error bars indicate the standard deviation over the 5-folds. Here, the MLP model outperforms random forest and this is apparent for amides, acyl halides, amines, alkyl halides, ketones, and esters.

4.4.2 Multiple functional groups prediction in a single compound present a second optimization problem.

Analysis of the receiver operator characteristic (ROC) plots (Fig 4.6) shows that at 1% of the false-positive rate, the model identifies over 80% of the true positive functional groups. Therefore, we used a dynamic threshold for each functional group to determine the presence of a functional group in the molecule. This threshold is calculated to maximize the functional group F1 score for the training set after training is complete. While the ability of the model to predict the presence of a particular functional group is important for evaluating the performance of the model, a metric better suited for the study of chemistry and essential for autonomous instrumentation will be to measure the performance to prediction all functional groups in a given molecule. Therefore, we have introduced new metrics, such as the 'Molecular F1 score (MF1)' and the 'Molecular Perfection Rate (MPR)' (see Fig 4.4 and the methods section for more details) and optimized our models for the FTIR and FTIR+MS data. After optimization, the FTIR+MS model was able to perform on par or better than the optimized combined IR for the majority of functional groups (Fig 4.7). The resulting models have comparable average MPRs (72.5 vs 74.9%) and MF1s (0.923 vs 0.931) for FTIR and FTIR+MS respectively. The hyperparameters for these models are given in the supporting information under Details of the neural networks.

4.4.3 MS data addition improves the prediction of specific functional groups.

Our optimized MLP model trained on FTIR data performs well on alkanes, ketones, arenes, carboxylic acids, and esters (average validation F1-score of 0.926) but it did not perform at par to predict nitriles, amines, amides, and acyl halides with an

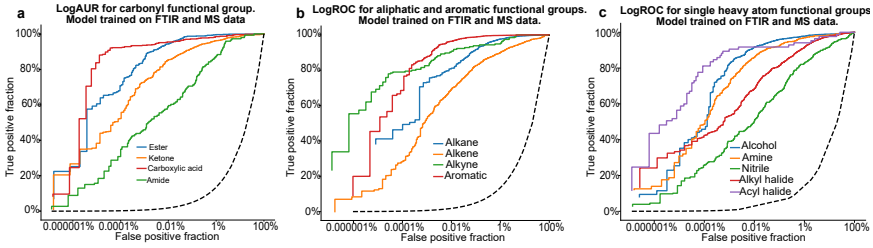


Figure 4.6. ROC plots for the model trained on both FTIR and MS spectra. (a) performance for carbonyl functional groups, (b) groups consisting of only carbon and hydrogen, and (c)the remaining functional groups. The underperformance of amides and nitriles can be discerned from these plots. These plots also allow us to select the best threshold value for each functional group which maximizes the F1 score for that functional group.

average validation F1-score of 0.663 (Fig 4.73c). We included the chemical features captured by mass spectrometry (MS) to augment the MLP-FTIR model (Fig 4.7d) to address these problematic functional groups. First, we trained an MLP model only on MS data to investigate its predictive capacity for functional groups (Fig 4.8a). The difference between the F1 scores of the training set compared to the validation set indicates that MS data needs other models to generalize for consistent performance compared to FTIR data using an MLP architecture. Similar to the MLP-FTIR model, the MLP-MS model performed well with more data for a given functional group (e.g. alkanes, arenes, alkyl halides), and poorly when fewer data were available (e.g. acyl halides, amides, and amines). An additional concern is the low resolution of the MS data with 1 (m/z) resolution that was used for training the model since it this resolution may not adequate in distinguishing some structures from each other.

Next, we investigated if combining FTIR and MS data could improve de novo prediction of functional groups by concatenating spectral data features into an FTIR+MS model (see experimental section). The improvement of the FTIR+MS model over the FTIR model is presented as Fig 4.8b, and the direct F1 scores are shown in Fig 4.7d

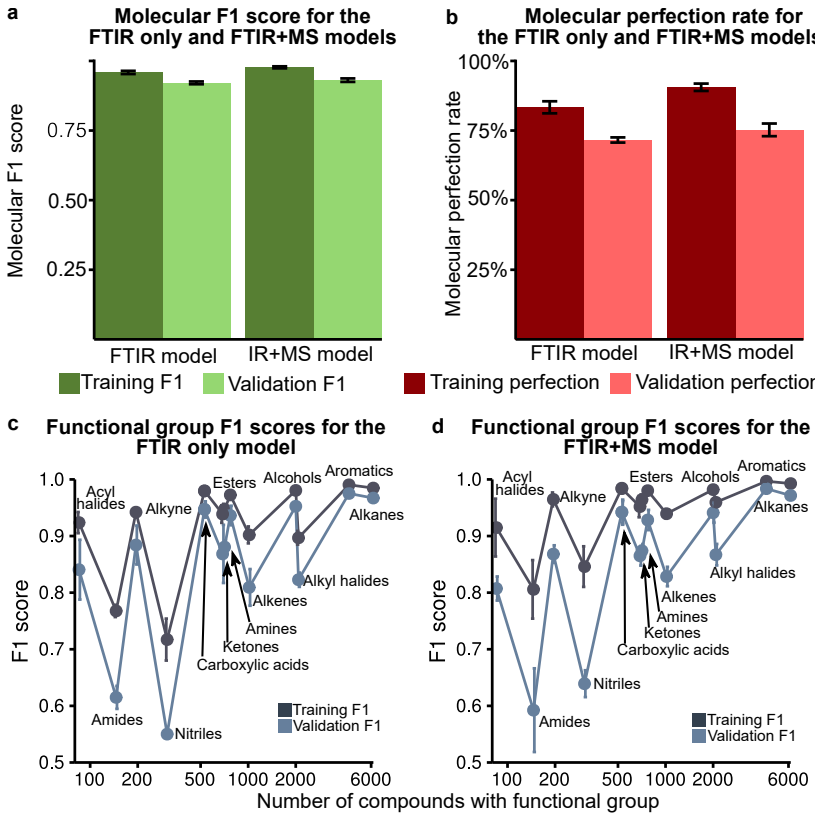


Figure 4.7. (a) The molecular F1 score for training and validation over the 5 folds is shown for both the optimized IR only and IR+MS models. The error bars indicate the standard deviation over the folds. (b) The molecular perfection for training and validation over 5 folds is shown for both the optimized IR only and IR+MS models. (c) The F1 score of the optimized IR only model plotted against the number of occurrences of that functional group. (d) The F1 score of the optimized IR+MS model plotted against the number of occurrences of that functional group.

with an average improvement of 0.024 overall functional groups. However, combining FTIR and MS data results in a substantial increase in validation F1 scores for the nitrile, alkene, and alkyl halide functional groups with improvements of 0.124, 0.048, and 0.061 respectively. The amide functional group remains unchanged as the F1 score of 0.563 is the as the MLP-FTIR model. The improvement of alkyl halides (Fig 4.84b) may appear to match chemical intuition given the distinct pattern of

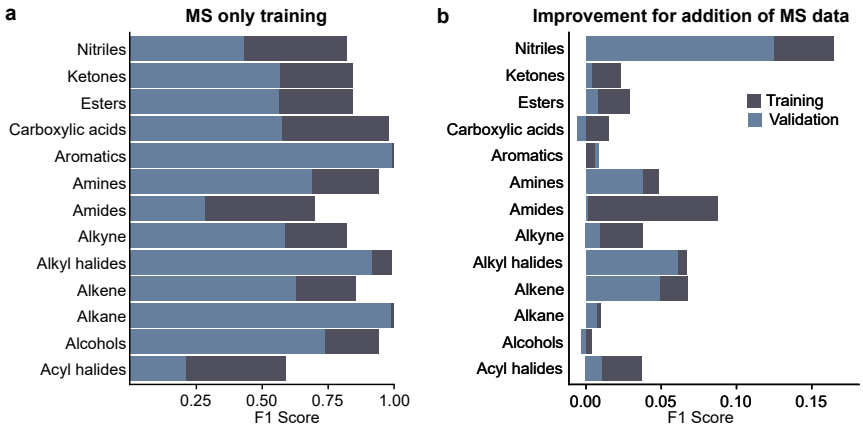


Figure 4.8. (a) Per functional group performance for an MLP model trained only on MS data shows that the model trained only FTIR data outperforms the model trained only on MS data during K-Fold validation. Also, the MS only model tends to become overtrained in comparison to the FTIR model potentially due to a greater degree of generalization for FTIR data. (b) The improvement in performance for each functional group when MS spectra are introduced in addition to FTIR data.

halogen isotopes observed with MS. However, this conclusion is not supported by the architecture of an MLP model as each input neuron is independent. Future work incorporating the differences in abundance peaks instead of raw values may improve the performance of the MS only model.

4.4.4 Guided backpropagation of the MLP model shows known FTIR and chemical patterns.

We performed guided backpropagation on the optimized MLP-FTIR model for molecules that were both predicted with an MPR of 1 and has the greatest activation in the neuron corresponding to the respective functional group (Fig 4.9). Several backpropagation plots reveal a known chemical association between peaks in FTIR spectroscopy and functional group assignment. This is encouraging as the model was

trained without any 'expert' or chemical information about the location of the peaks corresponding to each functional group. Specifically, we discuss several functional group cases for our selected set of molecules. The alkane functional group backpropagation shows the use of peaks near $3000cm^{-1}$, matching the known location of alkane CH peaks tabulated in the literature. The remaining peaks, however, do not provide any additional chemical intuition with regards to the alkane functional group. Aromatic compounds are identified by a peak between 1400-1600 cm-1, and the model selected peaks within this region. In addition, the model was able to identify the alkene bending motion around $900cm^{-1}$. A C-O stretch is typically observed around $1150cm^{-1}$, and the backpropagation plots for carboxylic acids, alcohols, and esters indicate a peak in this region is used by our model for each of these functional groups. Additionally, a strong C=O peak is typically observed for carbonyl compounds near $1600cm^{-1}$, but the model only placed importance on this peak for the amide functional group. The example alcohol compound contained both an alcohol group and a carboxylic acid, and the model ignored the C=O in the prediction of the alcohol, instead placing importance on peaks corresponding to the O-H stretch near $3500cm^{-1}$. These results show that the model reproduces the 'known chemistry' of functional group features without explicit input of peak to functional group relationships.

However, from our chosen set of molecules with MPR of 1, none of the backpropagation plots revealed any chemically significant characteristics for alkynes, amines, ketones, alkyl halides, and acyl halides. Instead, it appears that these functional groups are identified by the lack of sharp peaks in various regions of the spectra. This observation is interesting as the functional group F1 for these groups are relatively high. While nitrile groups have the lowest performance, the model was able to identify the 2210-2260 cm-1 band that is characteristic of this functional group. For the amine functional group, the model places high importance on a peak around

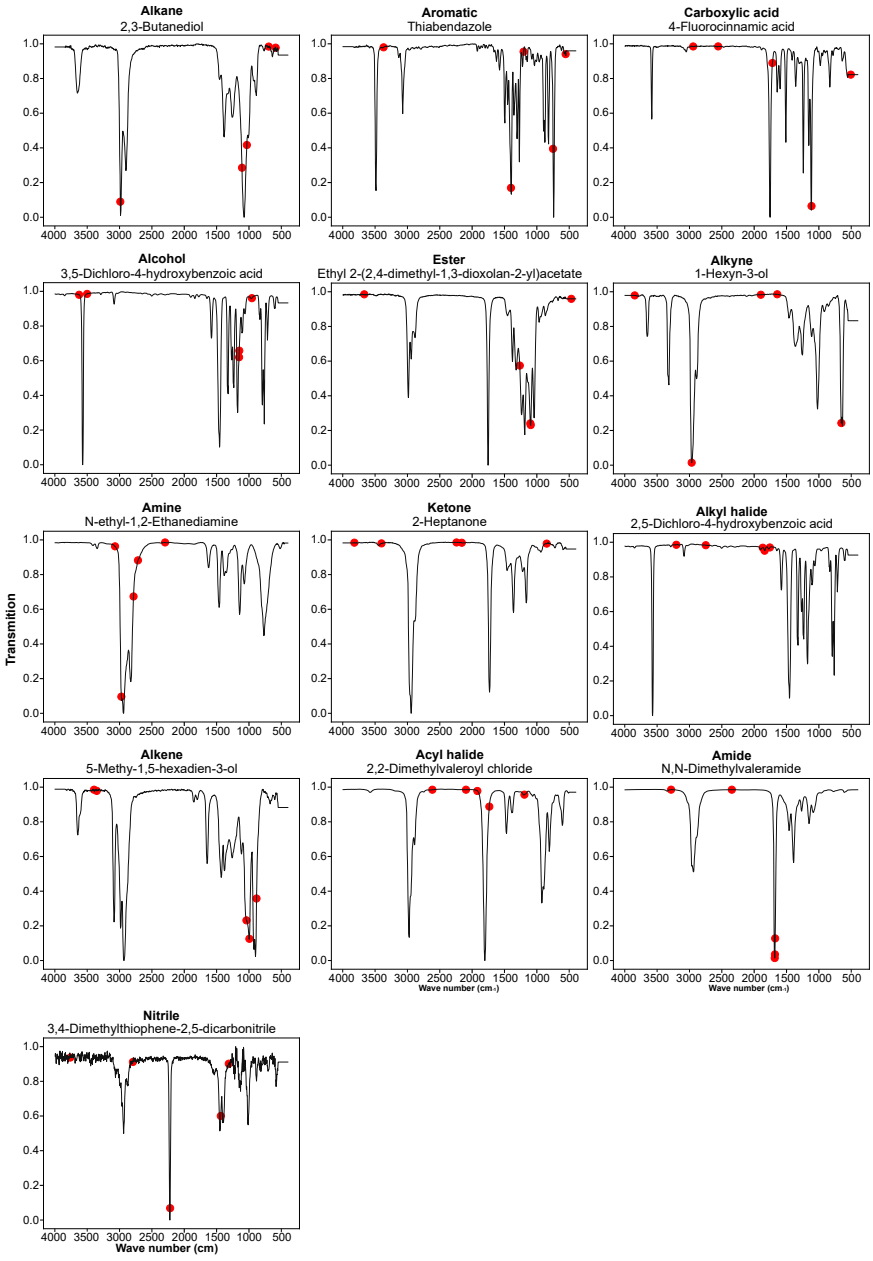


Figure 4.9. Backpropagation analysis for all 13 functional groups was performed to identify the regions of the spectra responsible for the result given. These plots are listed above in order of decreasing F1 score for the optimized FTIR+MS model.

1550-1640 cm-1. Although this may appear to indicate learned chemistry since the known N-H bending in this region, it also conflicts with the N-O bend of a nitro group. This observation may explain the reason our model misclassifies many nitro compounds as amides. Fortunately, there is a second N-O bend present which may rectify this issue if we include nitro groups to the model separately.

Next, we investigated the compounds with at least one incorrect functional group prediction (MPR = 0) provided in Listing S1. There are noticeable patterns of functional group types present in the set of failures. One example is nitro groups, which appear over 20 times in the failed compounds. This group is of interest as it is characterized by two strong bands which overlap with bending modes in alkane and amides functional groups. Many of these nitro compounds are misclassified as amides or alkanes and this observation partially explains the poor performance of amide functional groups shown in Fig 4.7a-b. Although it is discouraging to note that the model was unable to 'ignore' these peaks, the low count of amides present in the dataset may attribute to this poor performance.

4.4.5 Additional functional groups classification does not affect model performance of the original definitions.

In the previous section, we show that some functional groups explicitly trained in the MLP model were incorrectly classified due to overlapping peaks belonging to functional groups that were not included in our original set of functional group types. We hypothesized that the separate classification of the "overlapping" functional groups could affect the performance of our model. To test this hypothesis, we introduced the 'nitro,' 'ether,' and 'aldehyde' groups to the model. The 'nitro' group has significant overlap with the nitrile group (see the previous section), while the 'ether' group did not have peak values which overlapped with other functional groups in our previous

definition. Another limitation of our model is the inability to distinguish methyl groups from other alkane functional groups. We propose that this is possible due to the lack of a C-C stretch in methyl groups and methyl groups contain characteristic peaks not present in other alkane groups (i.e. the CH3 bend). In the NIST dataset many alkyl halides are present which do not contain any C-H bonds as all hydrogens in the molecule have been halogenated. Due to the large size of the alkane functional group in the training set, we hypothesize that splitting the alkane group into methyl and 'other' alkanes will not result in a large decrease in performance. Therefore, we decided to subdivide the 'alkane' group into 'methyl,' and 'other' alkanes as these groups performed the best out of all other groups in the original model.

Fig 4.10a-c show the results of these two hypotheses. The relatively high F1 scores for the 'methyl' (0.932) and 'other' alkane (0.936) groups support our hypothesis that sub–division of the original alkane definition does not decrease performance. Fig 4.10a-b also suggest that our hypothesis to improve low performance of functional groups by the introduction of new functional groups for both FTIR and FTIR+MS MLP model is incorrect. Although the nitrile and amide groups do not show improvement after the introduction of the nitro and ether groups as the F1 score for nitriles decreased by 0.019 and amides increased by 0.032, the new groups perform well as compared to the original problematic groups (0.932 for nitro groups and 0.923 for ethers). This suggests that the addition of new functional groups does not cause a significant loss in F1 score for other groups. Therefore, we speculate that more complex groups could be added to the model to provide detailed structural information, such as a model to identify heterocyclic aromatic rings from rings comprised of only carbon. While further subdivision of functional groups is beyond the scope of this work, they present a potential extension of this work towards realization of autonomous instrumentation that results in minimal manual intervention.

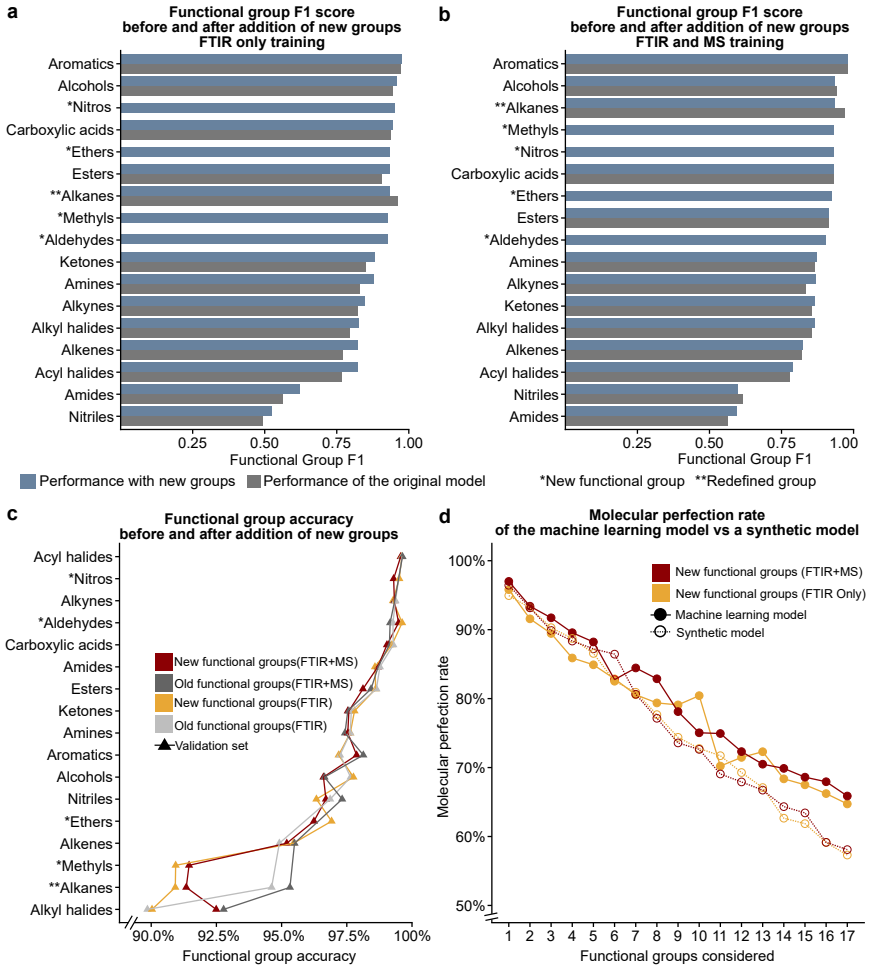


Figure 4.10. The bar plots given in (a) – (b) compare the functional group F1 scores for the original definitions of functional groups to the new definitions (see Table 4.3.3) showing that the addition of new additional functional groups does not have a significant impact on the previous functional groups. The line plot in (c) shows that the accuracy only decreases for the redefined functional group. The plot of molecular perfection rate in (d) compares the performance of the machine learning model to a synthetic model to show that the decrease in molecular perfection rate is expected as the number of functional groups increases.

4.4.6 Number of functional group predictions affects molecular perfection rate.

We hypothesized that our stringent metric of MPR was affected by the increase in the number of functional group predictions for a given model. To test this hypothesis,

we have created synthetic models based on the accuracies of each functional group from the trained FTIR+MS model (see Synthetic Models in the methods section). The machine learning model outperforms these synthetic models (Fig 4.10d and S5d), indicating that increasing the number of functional groups does not decrease this metric more than what would be expected from the inclusion of additional functional groups alone. The overall conclusion of this section is encouraging as it suggests that more functional groups can be added to our model without hurting the model's ability to predict other functional groups. Values for the MPR and MF1 scores for the new functional group definitions are 64.0335% and 0.909212 for the model trained on only FTIR data, and 65.2510% and 0.912017 for the model trained on both FTIR and MS data.

We were also interested in the performance of our model on molecules with a differing number of functional groups. To do so, we calculated the molecular perfection rate for compounds with one through six functional groups, for the original set of functional groups and the new set of functional groups (results shown in Fig 4.11). Unfortunately, no definite conclusions can be made from this data as the original versus new functional group definitions follow very different patterns. However, the original set of functional groups outperforms the new set of definitions. This observation is likely due to the reduced accuracy of the new alkane due to the split into methyl and non-methyl groups as both have accuracies of 91% where the previous model had an accuracy of 95% (Fig 4.10c).

4.4.7 Encoding spectra data in latent space retains functional group prediction performance.

Given the success of our MLP model in predicting functional groups using complete standardized spectra, we wished to investigate the ability of an autoencoder to

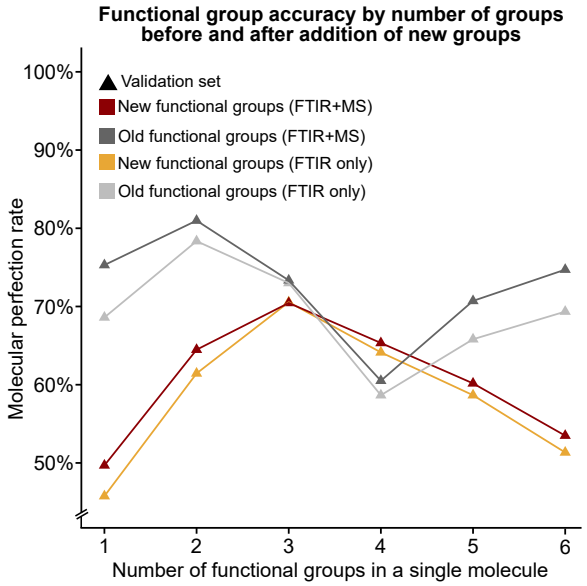


Figure 4.11. The molecular perfection rate calculated on molecules with a specific number of functional groups for both the original and new set of functional groups.

reduce the spectra into a latent space. This approach is different than that employed to create the SPLASH keys [273] for mass spectra. Unlike SPLASH hashed keys, a latent space of spectral data can be uniquely 'decoded' back to the original spectra without the use of any external database or additional information. We trained a simple linear model for encoding the FTIR and MS spectra into a 256-length vector and decoding this vector back to the original spectra used to create the vector (see Fig 4.2). The 256-length vector was used to train a second network for multi-task functional group prediction. For individual functional groups, the autoencoder model performs similar to that of the original MLP model. The molecular performance of the autoencoder model is similar to that of the original MLP model (Fig 4.12) as the MPR for the autoencoder model is 62.6% and the MF1 score is 0.905 as compared to 65.2% and 0.912 for the original model. This reveals that the original spectra contain redundant features that relate FTIR and mass spectra. We plan to explore the use

of this latent space for inverse design of molecules with combined spectral properties in future works.

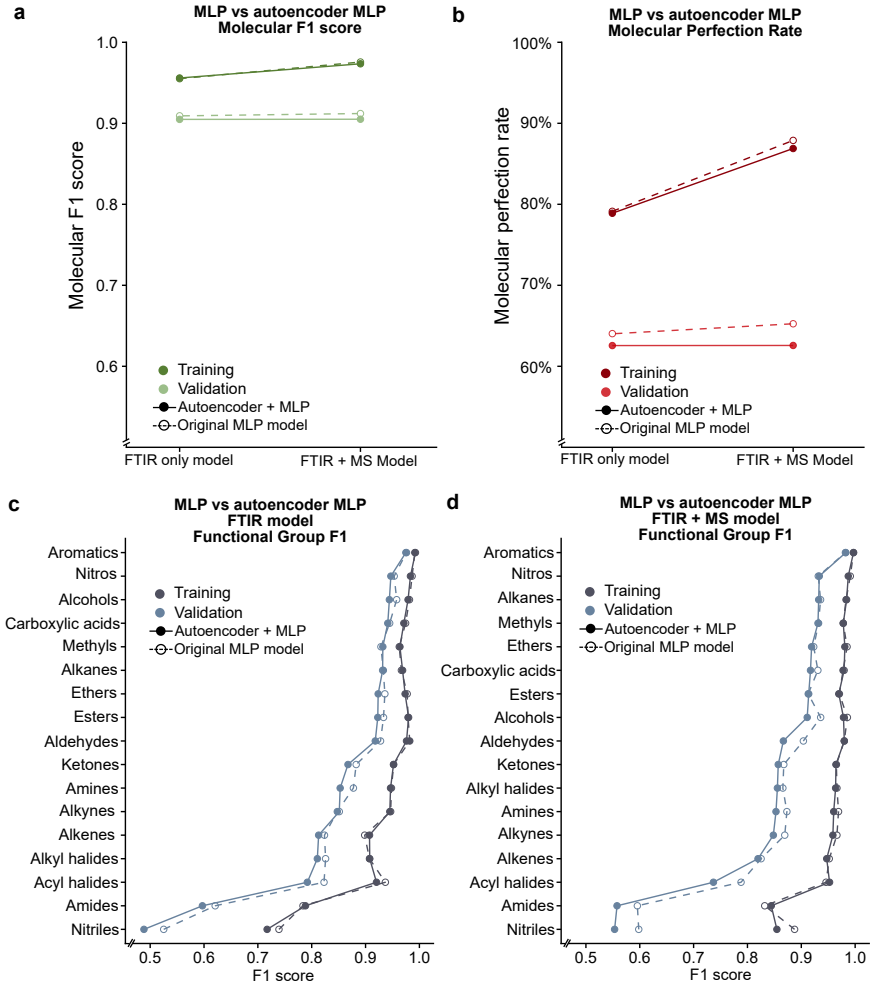


Figure 4.12. Comparison between the original MLP model and the autoencoder based model using the (a) molecular F1 metric and (b) molecular perfection rate are shown. Individual functional group F1 scores are provided for the FTIR only (c) and FTIR+MS (d) latent spaces.

4.4.8 Deep learning model trained on single compounds predicts functional groups in mixtures.

The ability to identify all the functional groups in a mixture of compounds expands the applicability of our methodology. To our knowledge, we are the first group to report the ability of machine learning methods to classify mixtures of compounds using a model trained on single compounds. To validate our method on compound mixtures, we obtained the FTIR spectra of three different mixtures of molecules (raw spectra given in Appendix G) and predicted all the functional groups of the compounds in the mixture using our MLP-FTIR machine learning model (see Table 4.2). For this test set, we have not included MS data since only a minor improvement was gained from addition of MS spectra based on training. In future works, we plan on improving the performance of functional group prediction by addition of MS data using more advanced machine learning architectures and molecular features. We stress the point that these spectra are obtained in our lab, are not part of the NIST dataset, and are obtained from instruments different than those used by the NIST as it is essential to validate a machine learning method for practical use in different laboratories. Since these spectra are external to the NIST webbook data, they constitute a 'test set' for our model. The compound mixtures were prepared by mixing two solid compounds and each mixture contained a different set of functional groups. Performance metrics, such as molecular F1 score etc., described previously for single molecules are applied to a mixture of molecules by considering the set of all functional groups (a union of all functional groups present in the mixture). For mixture 1, our FTIR-only method correctly predicted 2 out of the 4 functional groups present in the mixture, and predicted an additional functional group not present in the mixture, yielding an MF1 score of 0.65 (Table 2). Given the resolution of spectral data, the lack of an O-H peak above $3500 cm^{-1}$ could also lead a human chemist to

conclude that no carboxylic acid is present in the mixture. Additionally, the presence of a peak near $2940cm^{-1}$ may lead a human to conclude that a methyl group is present in the mixture. For mixture 2, we obtain an MF1 score for the mixture of 0.80 as we correctly predict 2 out of the three functional groups present in the mixture and do not predict any additional functional groups. The only missed functional group is the amide group, which is known to be problematic in our model (functional group F1 score < 0.60) and the lack of a strong peak near $1650cm^{-1}$ may contribute to a human's inability to identify this functional group. For mixture 3, our method correctly predicts 3 out of the 6 functional groups in the mixture and does not predict any additional groups in the mixture, yielding a molecular F1 score of 0.67. The model was not able to identify a methyl group and a human may make the same mistake given the lack a peak near $2940cm^{-1}$. The model also failed to predict the presence of a nitro group and the presence of an ether, potentially due to the peaks corresponding to these groups overlapping with other peaks in the aromatic region of the spectra. Our results show that the deep learning model trained on single compound spectra can give reasonable performance to predict functional groups for mixtures of compounds. Future work entails training on compound mixture spectral data along with using other deep learning architectures, such as Generative Adversarial Networks. This is essential for correctly estimating the limitations of machine learning models for adoption in industry for autonomous instrumentation.

4.4.9 Reaction networks allow one to verify that a reaction has occurred in an automated fashion.

One of the most important applications of machine learning for functional group prediction is the ability to automate the analysis for determining whether a given reaction has occurred by predicting the functional groups of the reactants and prod-

Table 4.2.
SMARTS strings used to identify the presence of a functional group given the 2D topology of a molecule. The functional groups marked with a * were not present in the original set of functional groups. The definition of alkane changed between functional group sets due to the introduction of methyl.

	Mixure 1	**Mixture 2**	**Mixure 3**
Component 1	6-Chloro purine	2-iodo acetamide	4-methoxy-2-nitro aniline
Component 2	Hippuric acid sulfonic acid	4-hydrazino benzene acid anhydride	4-bromo-1,8-naphthalic
Component 1 FGs	Halide, aromatic	Halide, amide	Ether, methyl, nitro, amine
Component 2 FGs	Aromatic, amide,	carboxylic acid	Aromatic Halide, aromatic
Mixture FGs	Aromatic, carboxylic acid, halide, amide, alkane	Aromatic, halide, amide	Aromatic, halide, nitro, ether, methyl, amine
Predicted FGs	Aromatic, halide, methyl	Aromatic, halide	Aromatic, halide, amine

ucts [274]. As the final case study for this work, we present the prediction of a 'reaction network' using the combined IR+MS model (Fig 4.13). We selected the synthetic scheme for small-molecule inhibitor for Programmed Cell Death-1/Programmed Death-Ligand 1 (PD–1/PD–L1) Interaction[ref]. The commercially available starting material 3–Bromo–2–methyl benzoic acid was used to synthesize a final compound (designated as kpgc01s94) in the multi-step process. Firstly, 3–Bromo–2–methyl benzoic acid was reduced to aryl methanol (kpgc01s02) using Borane tetrahydrofuran complex solution, and then C–C bond formation was achieved using Suzuki-coupling reaction between kpgc01s02 and phenyl boronic acid to get kpgc01s05. Next, kpgc01s05 and 4–hydroxy–2,6–dimethoxybenzaldehyde were reacted under Mitsunobu reaction condition to prepare ether linkage in kpgc01s37. Further, reductive amination yielded kpgc01s94. However, based on IR+MS spectra, a chemist needs to identify the correct functional group to predict the desired product formation, but such a task is difficult due to functional group region overlaps of the FTIR spectrum. For example, the frequency of the carbonyl group is dependent on which type of functional group is present (ketone, aldehyde, ester etc.) as well as the presence of extended conjugation (aliphatic or aromatic compound). Thus, FTIR+MS model would be beneficial to predict functional groups in the compound. Considering these aspects, for each compound shown in the reaction scheme of Fig 4.13a, the FTIR+MS model was used to predict the functional groups present in the compound using both its IR and MS spectra (Fig 4.13b). The change in these functional groups throughout this reaction scheme are shown in Fig 4.13d while Fig 4.7c gives the true changes in functional groups obtained from the known compounds in the reaction scheme. The model predicted 3–Bromo–2–methyl benzoic acid as both a halide and an alkene (incorrect), but did not predict any aromatic or carboxylic acid functional groups. While the inability of the model to predict any carboxylic acid has been investigated

in previously in this work (see the previous section), the inability of the model to predict aromatic functionality is concerning given that this functional group has the greatest F1 score of all functional groups presented in this work. The incorrect prediction of an alkene can be explained by the presence of peaks in the alkene region of the IR spectra (1640–1680 cm^{-1}). For the next compound in the scheme, kpgc01s02, the model incorrectly predicted the lack of a halide group while correctly predicting the presence of an alcohol and aromatic groups. As a result, the reaction network generated by our model suggests that the alkene and halide groups transform into an aromatic group and an alcohol group instead of a simple reduction from a carboxylic acid to an alcohol. The model predicts the same functional groups for kpgc01s05 as it does for kpgc01s02, therefore there is no change in the reaction network for this step to correspond with the disappearance of the halide group in the the actual reaction network (orange arrow in Fig 4.13c). This failure to predict a change in functional groups represents the greatest failure of our model as it would predict no reaction has occurred while, in reality, a reaction did occur. For kpgc01s37 and kpgc01s94, the model incorrectly predicts that both compounds contain a halide group, resulting in the incorrect addition of this halide group in the predicted reaction network. The model also failed to predict the presence of an ether functional group for both kpgc01s37 and kpgc01s94, therefore it is not present in the predicted reaction network. A ketone functional group is predicted instead of an aldehyde group for kpgc01s37, a mistake which a human can make given that these two functional groups share the same region of the IR spectrum. The model correctly predicts the presence of an amine group in kpgc01s94 and thus obtains the correct edge connecting a carbonyl group to an amine in the final step of the reaction. However, the model fails to predict the presence of a carbonyl group in the final compound of the scheme, but this is expected given the model's prior failures in predicting this functional group. Overall,

these results indicate that the model is capable of determining whether a reaction has occurred by identifying changes in functional groups for the given molecule, but is unable to determine the exact functional group changes which are present in these reactions. We plan on addressing these issues in future works by developing novel ANN models for detecting the presence of a reaction in a given reaction network, possibly by the inclusion of NMR data.

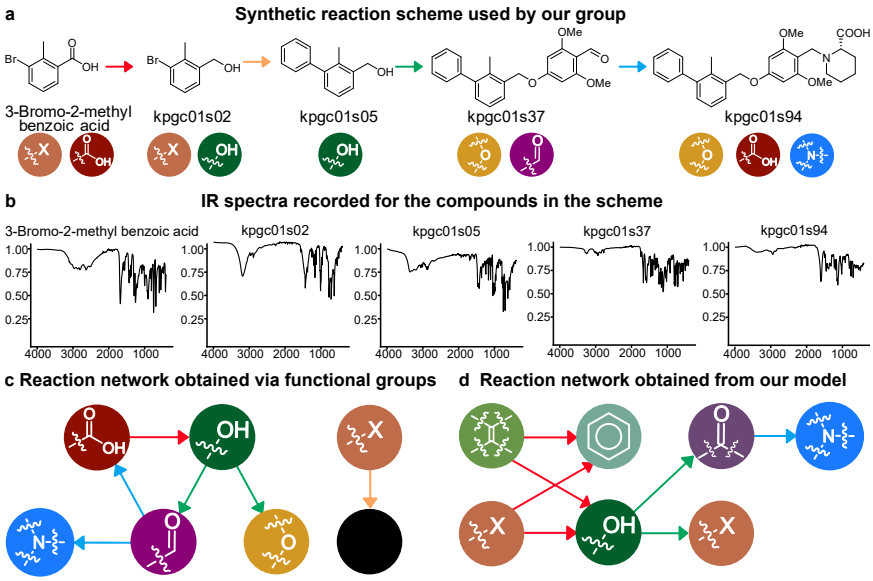


Figure 4.13. A synthetic scheme proposed in our lab is presented along with the functional groups which change in the given reactions (a). The colors of the arrows indicate which reaction has occurred. The IR spectra of each member of the reaction scheme is given in (b). The reaction network for the actual compounds is represented as the changing of functional groups in (c) and the predicted reaction network obtained from our model is given in (d).

4.5 Conclusion

We present a machine learning method for de novo prediction of functional groups using a combination of FTIR and MS data. We introduce two new metrics apart from

functional group F1 score, namely, molecular F1-score and molecular perfection rate for practical use of our models. Our results show that, in general, the FTIR data is more consistent for predicting functional groups than MS data, a conclusion backed by chemical intuition. However, several functional group predictions benefit from the inclusion of MS data. Additionally, our model architecture is more optimal for analysis of FTIR data due to the continuous nature of these spectra, and the mathematical structure of an MLP model. Our model's performance is not affected by the number of functional groups present in the training data and it predicted all the functional groups consistently across all metrics. Moreover, several known chemical patterns in the spectra were identified as features for the model to identify common functional groups without any expert training of the system. We conclude that a multi–class, multi–label perspective is apt for further studies which may combine differing spectroscopic data types that may reveal unknown features useful for the identification of compounds. We show that our approach for functional group predictions is flexible as it can be extended to introduce new or sub-divide existing functional groups without affecting the performance of original functional group definitions. Furthermore, reducing chemical spectral data in a latent space does affect model performance to predict functional groups but can be used for inverse design of molecules based on a combination of spectral properties. Finally, we have verified that our model also produces reasonable results for a mixture of compounds containing multiple, different functional groups. Therefore, our machine learning model can be used for database-free identification of functional groups in pure and complex mixtures of compounds. We believe that these accomplishments are significant advancements in the development of algorithms and methods for the autonomous identification of functional groups. We hope that the continued development of future spectral learning methods builds upon our work and will adopt or improve upon the molecular F1 score and

molecular perfection rate metrics to assess their models to predict multiple functional groups for molecular structure elucidation.

4.6 Future work

This work is only preliminary and much remains to be done. The first and immediate next step is to remove the manual assignment of functional groups by using a graph–based method to determine the environments of atoms. An example would be using the Morgan fingerprining algorithm to assign environments to atoms and then predicting the presence of atoms using their FTIR spectra. A major drawback to this approach is that FTIR spectra are a function of bonds in a molecule, not necessarily the atoms. Therefore, the success of such a method is highly dependant on the ability of this algorithm to capture these bond details. An alternative approach would be to develop a specialized fingerprinting algorithm that determines the presence of bonds instead of atoms. This method, however, would be inferior to a method developed to encode edge features in graph (similar to a graph neural network, see appendix A). Unfortunately, This maybe the limit of the limit of this technique for structure determination. However, it is interesting to note that one can use it to analyze spectra over time, as shown in Fig 4.14.

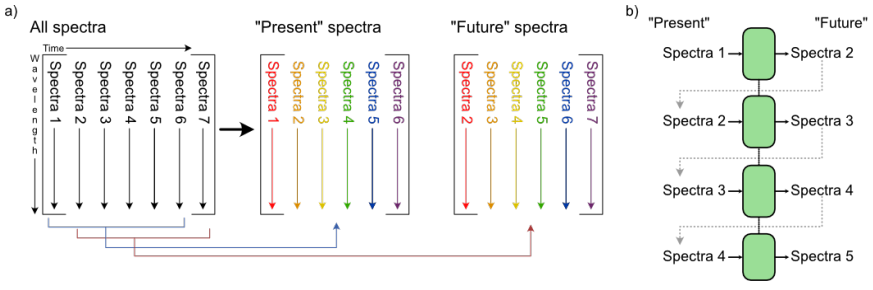


Figure 4.14. A potential model for predicting FTIR spectra over time. This model can be integrated into the work shown in this chapter in the near future.

Conversely, there is a lot more work that can be done for the prediction of functional groups using MS data. However, this can only be done using a novel ML model more suited for the analysis of MS data. For example, a model which takes advantage of mass differences and a unique architecture could be used to take this problem. For example, one could use a graph convolutional neural network [275] as a canvas for predicting functional groups and re-weight this canvas using the predicted atoms and functional groups. Another approach could make use of an autoencoder to encode graph properties along with spectral properties and decode this latent space into functional groups in the molecule. This second model is detailed in Fig 4.15. These works mark the next steps in this emerging field.

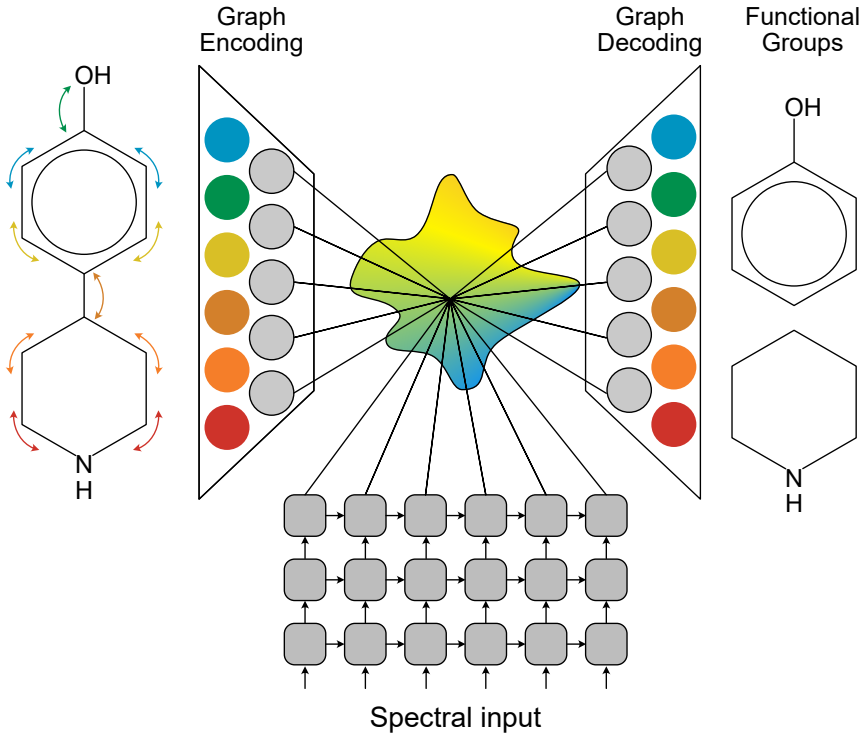


Figure 4.15. A new model for incorporating MS data into functional group prediction.

5. IDENTIFYING THE FUNCTIONAL GROUPS OF SMALL MOLECULES USING ION–MOLECULE REACTIONS

This chapter is available as

Fine, J., Liu J., Beck A., Alzarieni K., Ma X., Boulos V., Kenttämaa., Chopra G. Graph Based Machine Learning Interprets Diagnostic Isomer-Selective Ion-Molecule Reactions in Tandem Mass Spectrometry. *ChemRxiv* (2019). https://doi.org/10.26434/chemrxiv.11466183.v1

5.1 Abstract

Diagnostic ion-molecule reactions using tandem mass spectrometry can differentiate between isomeric compounds unlike a popular collision-activated dissociation methodology for the identification of previously unknown mixtures. Selected neutral reagents, such as 2–methoxypropene (MOP) are introduced into an ion trap mass spectrometer and react with protonated analytes to produce product ions diagnostic of the functional groups present in the analyte. However, the interpretation and understanding of specific reactions are challenging and time-consuming for chemical characterization. Here, we introduce a first bootstrapped decision tree model trained on 36 known ion-molecule reactions with MOP using graph-based connectivity of an analyte's functional groups as input. A Cohen Kappa statistic of 0.72 was achieved, suggesting substantial intermodel reliability on limited training data. Prospective diagnostic product predictions were made and validated for 14 previously unpublished

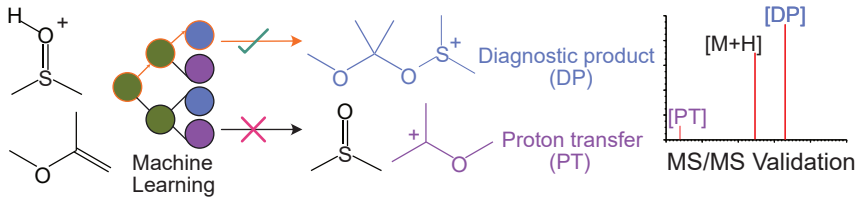


Figure 5.1. Table of contents figure for the online publication

analytes . Chemical reactivity flowcharts were introduced to understand the decisions made by the machine learning method that will be useful for chemists.

5.2 Introduction

Tandem mass spectrometry (MS/MS) is a powerful analytical tool that is extensively used for the characterization of complex mixtures in many fields, such as proteomics, petroleomics, and drug discovery [276–279]. Currently, the most commonly used MS/MS technique to obtain structural information for ionized and isolated mixture components is collision-activated dissociation (CAD) [253, 254]. In these experiments, the analyte ions are accelerated and allowed to collide with an inert gas, such as helium. Upon the collisions, part of the kinetic energy of the ions is converted into their internal energy, resulting in fragmentation. This approach is limited by the fact that isomeric ions often generate identical fragmentation patterns, making identification of compounds via CAD mass spectra unreliable [279, 280]. To address this issue, a MS/MS approach based on diagnostic, reliable and predictable gas-phase ion-molecule reactions has been developed [280–284]. This approach can be used to identify specific functional groups or their combinations in ionized and isolated mixture components to thereby facilitate the differentiation of isomeric ions, often without the need for reference compounds.

One of the neutral reagents used previously to differentiate two isomeric drug metabolites is 2-methoxypropene (MOP) [280]. In these experiments, protonation of the analytes was achieved through atmospheric pressure chemical ionization (APCI) in a linear quadrupole ion trap (LQIT) mass spectrometer. The protonated analytes were transferred into the ion trap, isolated and allowed to react with MOP that was continuously introduced into the ion trap (Fig 5.2). Formation of a diagnostic, stable addition product, proton transfer to MOP, or no reaction with MOP was monitored. The diagnostic addition product ions were only observed for the protonated sulfoxide drug metabolite and not for its keto–isomer (Fig 5.3). This was verified via studies of several protonated model compounds [281].

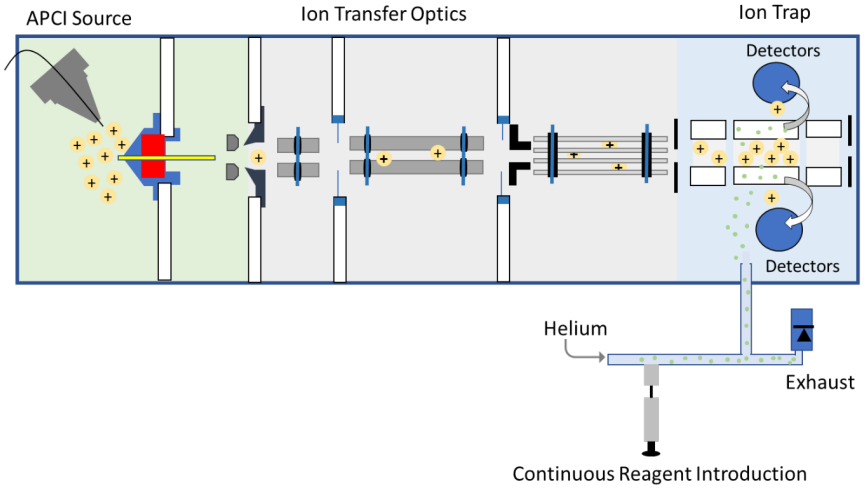


Figure 5.2. Schematic diagram of a linear quadrupole ion trap mass spectrometer equipped with an APCI source and an external reagent mixing manifold (bottom) [283, 285]. This instrument can be used to detect diagnostic ions formed between analytes protonated upon APCI and a neutral reagent (introduced using the reagent mixing manifold) in MS/MS experiments occurring in the ion trap.

Interpretation of the data obtained for complex mixtures in the above experiments is challenging and time-consuming due to the large amount of data. In order to facilitate this process, we decided to develop a chemical graph based interpretable

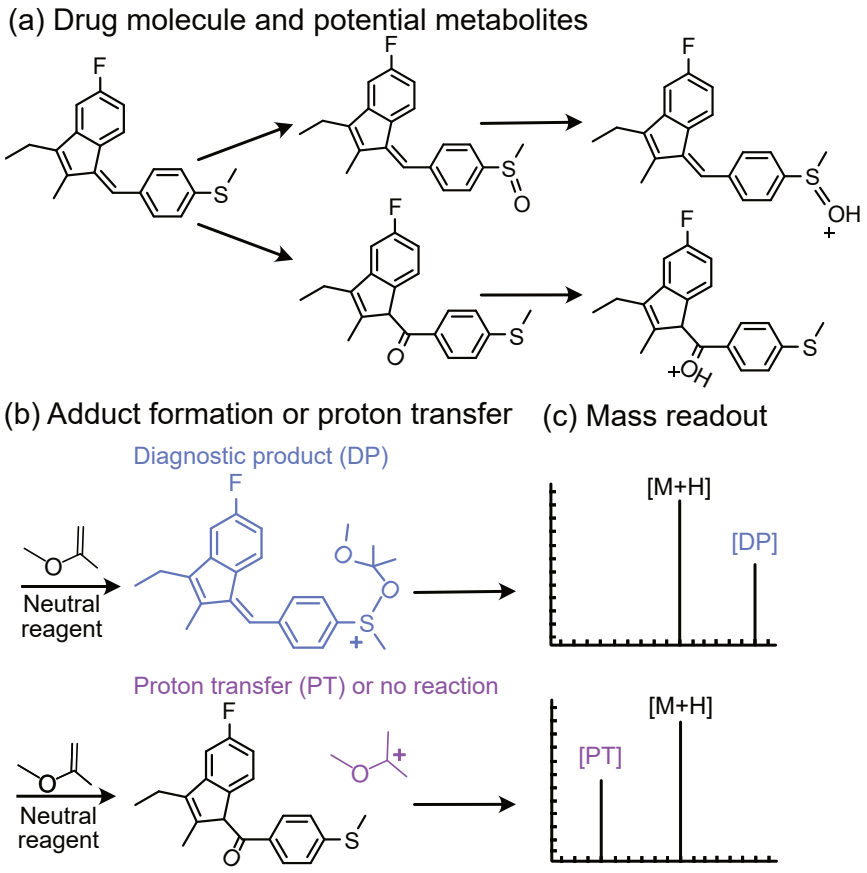


Figure 5.3. The diagnostic utility of employing neutral reagents, such as MOP, to identify functional groups in protonated metabolites of a drug. After the metabolites were (a) protonated and isolated, (b) they were allowed to react with MOP and (c) the formation of a diagnostic addition product (DP) as opposed to proton transfer (PT) no reaction was monitored. Only the protonated sulfoxide metabolite generated the diagnostic addition product ion (DP) with MOP.

machine learning methodology to facilitate data interpretation and prediction of whether a given protonated analyte will form a diagnostic product ion upon reactions with MOP. Multi-Layer Perceptron (MLP) [145, 155, 286], Long-Short Term Memory (LSTM) [97,287], and Graph Convolution Networks (GCN) [262,288–290] approaches have been demonstrated to be suitable for predicting reaction outcomes when a large number of known reactions are available. Unfortunately, due to the specificity of the

diagnostic ion-molecule reactions of interest here, only a relatively small set of known reactions exist. Additionally, these models are difficult to understand and yield no additional chemical insight. Although one-shot and few-shot learning has proven useful in the literature for systems with a small number of observations [31,163,291,292], these models are typically difficult to interpret and only limited information can be obtained about the reactions. Therefore, a ma-chine learning methodology that can be interpreted by humans is developed in this work.

Previously, the proton affinity (PA) of an analyte was used to predict whether a protonated analyte would undergo diagnostic product formation, proton transfer or no reactions with MOP [281]. If the PA of the analyte is lower than that of MOP, proton transfer usually dominates. On the other hand, if the PA of the analyte is greater than that of MOP, a diagnostic adduct may be formed. However, accurate predictions between formation of the diagnostic adduct and no reactions were not possible. Nevertheless, PA values may be used as a baseline for benchmarking potential machine learning methods or as a source for additional input features.

5.3 Results and discussion

5.3.1 Choice of the Machine Learning Model.

Given the sparsity of data available for training a machine learning model, traditional architectures known to perform well with small amounts of data were evaluated. These machine learning architectures include regularized logistic regression [149,293], decision tree models [294, 295], partial least squares [296], generalized linear models [297], and k-Nearest Neighbor [298]. Each of these models solves classification problems in a very different manner. For example, logistic regression attempts to assign numeric weights to an input vector. This vector is then used to linearly trans-

form the input into two probabilities for assignment of the input as a given class. On the other hand, decision trees (when trained for classification) attempt to reduce the Shannon Entropy of the predicted class by splitting the data using a set of Boolean operations. This yields a flowchart of logical decisions that one can use to evaluate the decisions made by the model (see the methods section for details of this procedure). The major advantage of decision tree models, with analytes represented as an input bit vector of functional groups, is that the resulting flow chart diagram can be interpreted by chemists to gain a deeper understanding of the chemistry resulting in a reaction taking place. This procedure is widely used in both biology [295] and chemistry [164, 299] to identify and interpret how input features (in this case the collection of functional groups) correlate with a property of interest (reactivity toward MOP in this case). Recently, similar techniques have been applied to reaction chemistry [150] to understand how various chemical moieties are related to the reactivity of a molecule. Here, we used bootstrapping of several decision tree models to ensure robustness of our model for prospective experimental validations. Moreover, a comparison of the performance of decision trees to other machine learning models was also performed to ensure that efficacy was not compromised for the sole sake of interpretability.

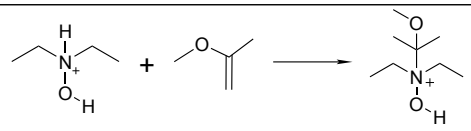
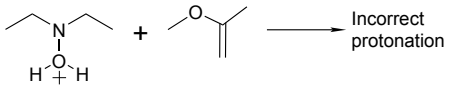
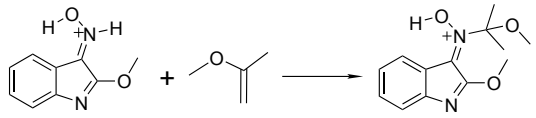
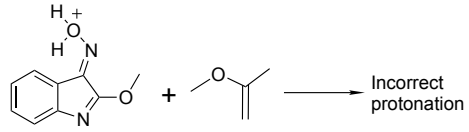
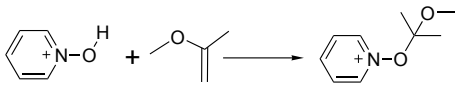
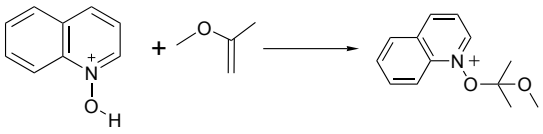
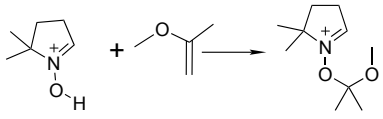
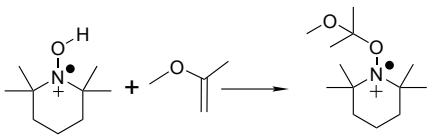
To develop a chemically interpretable machine learning model, the presence or lack of a topology of a collection of atoms (referred to as functional groups) was related to predicted reactivity. The Morgan Fingerprint algorithm [300, 301] was used to represent such functional groups. It avoids the use of manually created functional groups subject to human bias and interpretation. Additionally, previous work indicates that the use of Morgan Fingerprints in machine learning is an effective approach across chemical disciplines [302–304]. Briefly, this algorithm functions by finding all subgraphs of a molecular graph (i.e. the connectivity of the atoms in the molecule)

and assigns a number to these subgraphs calculated via a set of hashing functions applied to each atom and its respective neighborhood. This yields an integer which can be used as a surrogate for the functional group. The size of these subgraphs was determined by a radius parameter that is supplied by the user a priori. Application of a small radius in machine learning has been shown to avoid the potential for the same integer to represent the same functional group, a phenomenon known as a bit collision [305]. In this work, the ability of models trained on different radii were also com-pared to ensure that the selection of fingerprint radius is optimal for the task at hand.

5.3.2 Cutoff Assignments for the Machine Learning Model.

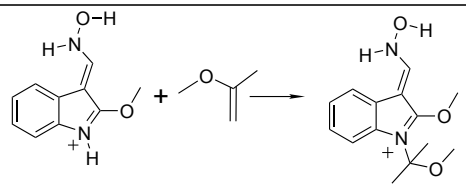
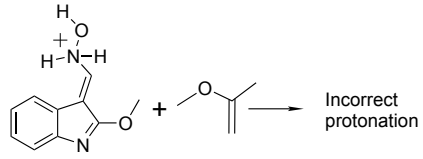
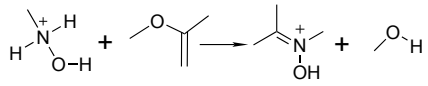
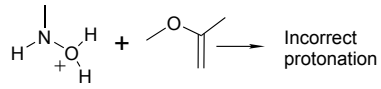
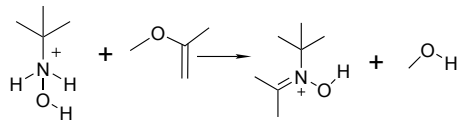
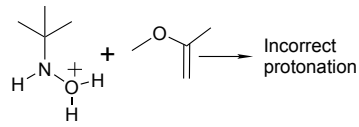
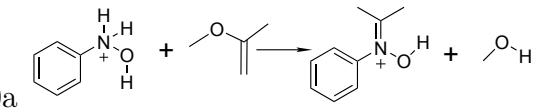
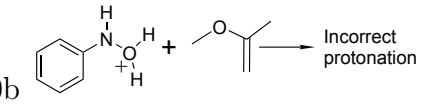
Since the experimental outcome of a given analysis was either proton transfer/no reaction, or the formation of a diagnostic addition product ion (see Fig 5.3), and a limited amount of data was available for training, a binary classifier is preferable to other supervised machine learning models. The training set for this classifier included a set of 36 protonated analytes whose reactions with MOP have been studied along with their product branching ratios [281, 306, 307] (see Table 5.1 for all MOP reactions). The distribution of product branching ratios measured for the diagnostic addition reaction (see Fig 5.4a) shows a large gap between 65 to 83% as no compounds have a diagnostic product branching ratio between this percentage gap. This gap indicates that a cutoff of 70% or greater for the branching ratio should be used in this binary classifier to determine whether a given analyte will undergo the diagnostic addition reaction with MOP.

Table 5.1.: The 36 known reactions used for training the machine learning models.

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
1a 	85%	220.2 [d] Ammonia
1b 		188.2 Methanol
2a 	11%	206.7 Ammonia
2b 		177.0 Methanol
3 	99%	221.4 [d] Methanol
4 	86%	226.2 [d] Methanol
5 	66% [c]	224.7 [d] Methanol
6 	50% [c]	212.6 Methanol

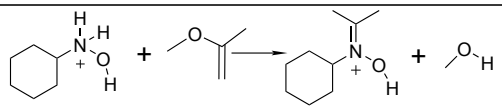
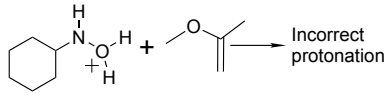
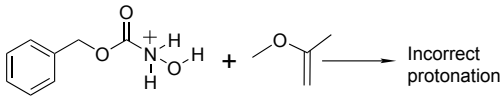
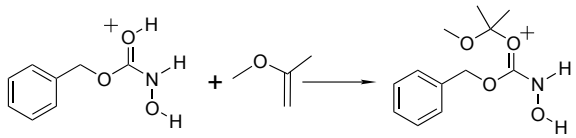
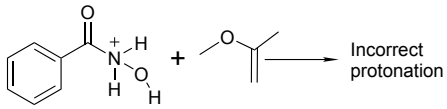
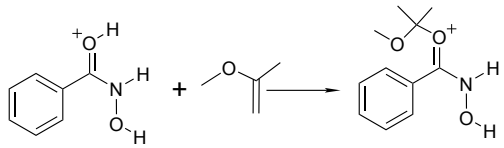
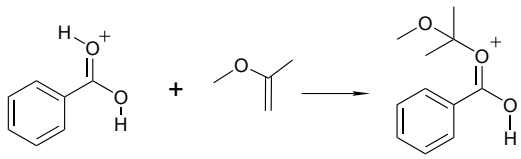
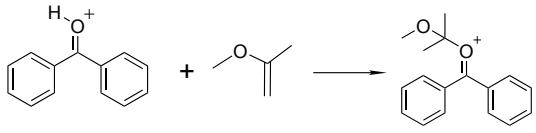
continued on next page

Table 5.1.: *continued*

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
7a 	84%	236.0 [d] Ammonia
7b 		194.0 Ammonia
8a 	3%	204.2 Ammonia
8b 		175.8 Methanol
9a 	27%	213.9 Ammonia
9b 		182.1 Methanol
10a 	13%	205.8 Ammonia
10b 		204.9 Methanol

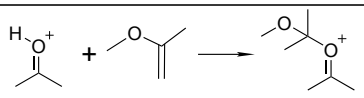
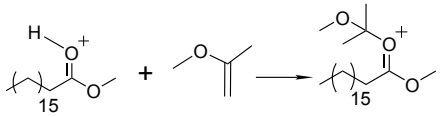
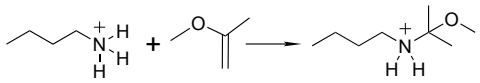
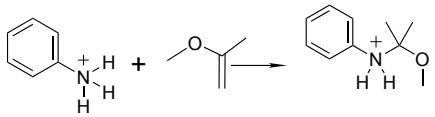
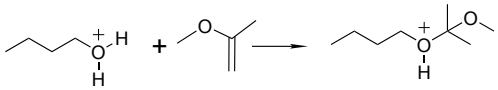
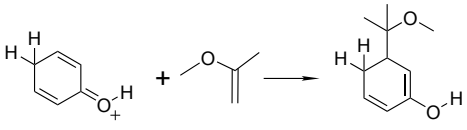
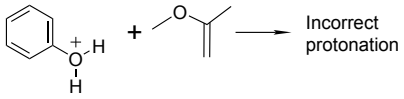
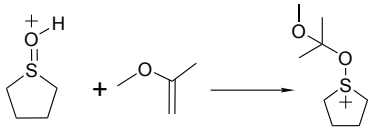
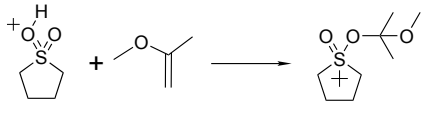
continued on next page

Table 5.1.: *continued*

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
11a 	25%	214.3 Ammonia
11b 		184.7 Ammonia
12a 		189.2 Ammonia
12b 	23%	205.2 Methanol
13a 		195.8 Ammonia
13b 	12%	209.7 Methanol
14 	0%	196.2 196.2 Methanol
15 	2%	211.3 210.9 Methanol

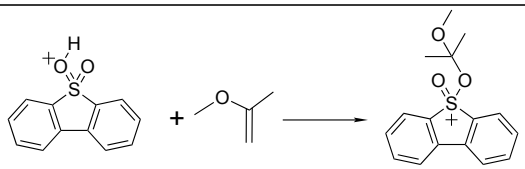
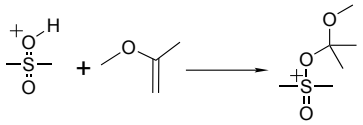
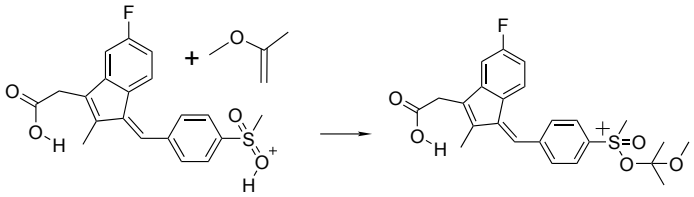
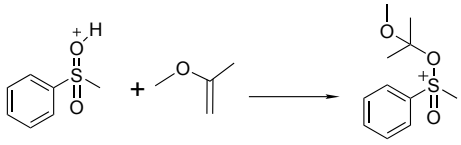
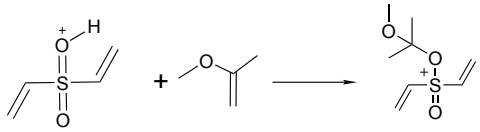
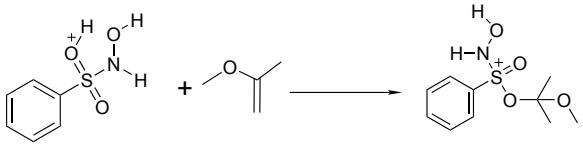
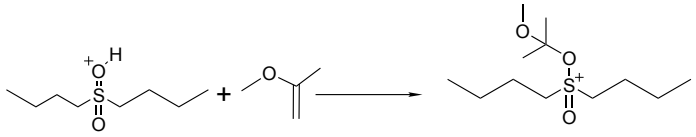
continued on next page

Table 5.1.: *continued*

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
16 	0%	194.4 194.0 Methanol
17 	0%	207.9 Methanol
18 	2%	220.0 [d] 220.2 Ammonia
19 	0%	209.6 210.9 Ammonia
20 	0%	188.0 188.6 Methanol
21a 	2%	194.0 195.3 Benzene
21b 		178.0 Methanol
22 	37%	220.9 [d] Methanol
23 	0%	199.7 Methanol

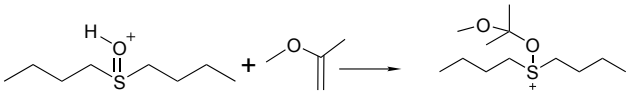
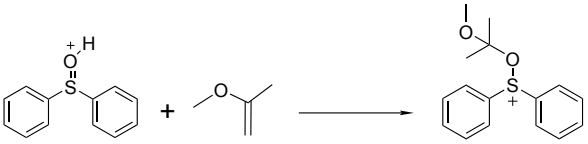
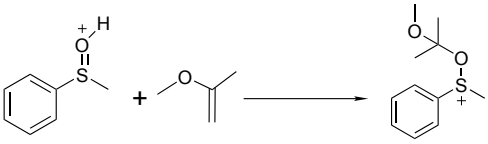
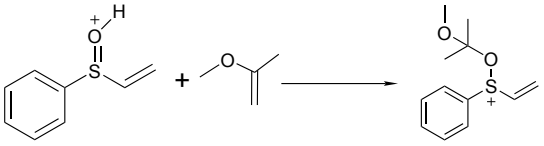
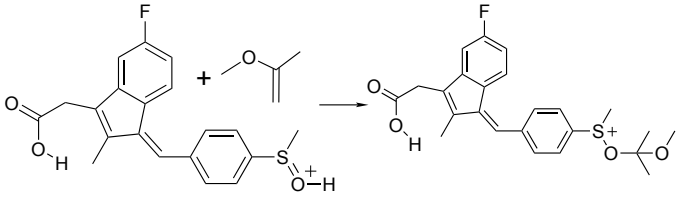
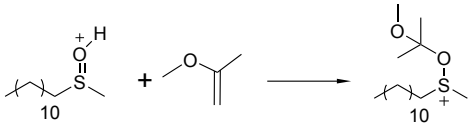
continued on next page

Table 5.1.: *continued*

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
24 	0%	205.2 Methanol
25 	0%	195.5 Methanol
26 	15%	200.8 Methanol
27 	1%	202.0 Methanol
28 	1%	196.9 Methanol
29 	2%	195.2 Methanol
30 	1%	200.8 Methanol

continued on next page

Table 5.1.: *continued*

Protonated analyte for training	DPBR [a]	PA (DFT) Reference PA experiment[b]
31 	99%	223.9 [d] Methanol
32 	99%	223.2 [d] Methanol
33 	54 %	221.1 [d] Methanol
34 	50 %	222.9 [d] Methanol
35 	97%	222.0 [d] Methanol
36 	98%	222.8 [d] Methanol

[a] Diagnostic product branching ratio [281, 282, 285, 306–308].

[b] Experimental value for the proton affinity [309].

[c] Reactivity change from the 67^{th} to 78^{th} quantile. See Cutoff Assignments for Machine Learning Model and Fig 5.4 for details.

[d] Value greater than the proton affinity of MOP (214.42 kcal/mol) as calculated using Density Functional Theory, see section *Calculation of proton affinities* in methods for details.

The selection of the above cutoff resulted in 8 protonated analytes being classified as forming a diagnostic addition product ion with MOP and 28 protonated analytes being considered as non-diagnostic. Since this split was unbalanced (i.e. more non-diagnostic reactions than diagnostic), the Cohen Kappa Statistic [119] (see appendix A) was used to compare the success of different models. A Kappa statistic of zero indicates that the model performs at random and a value of positive 1 indicates a perfect classifier (see Methods section for details). To further investigate the effects of this cutoff value, models created with a 70% cutoff were compared to those created with 10, 20, 30, 40, 50, 60, and 90% cutoffs to ensure that this choice was logical with respect to how the models performed for reactions not used to train the model. Note that a cutoff of 80% was not considered as it produced the same set of analytes that underwent the diagnostic reaction as the 70% cutoff.

A potential alternative to the 70% cutoff is 40% as this represents the second-largest gap in the distribution of diagnostic product branching ratios (see Fig 5.4a). This value is approx-imately at the 67^{th} quantile of the data and resulted in a split of 13 analytes that underwent the diagnostic addition reaction, compared to 23 analytes that did not. When considering the result of the binary classifier with different cutoffs, 70% and 40%, the model classified four analytes, TEMPO (an N-oxide radical), 5,5–dimethyl–1–pyrroline N-oxide, methyl phenyl sulfoxide, and (ethenesulfinyl)benzene (a sulfoxide) (see Fig 5.4b) differently. Conversely, with both cutoffs, the model classified all sulfones, alcohols, and amines to undergo proton transfer or no reaction instead of forming a diagnostic addition product ion. The similarities and differences between the 70 and 40% cut-offs could be used to further understand how the model per-forms and assigns classifications.

To ensure that a decision tree model will perform well prospectively, 14 compounds that were not present in the training set (i.e., test set) were evaluated using a boot-

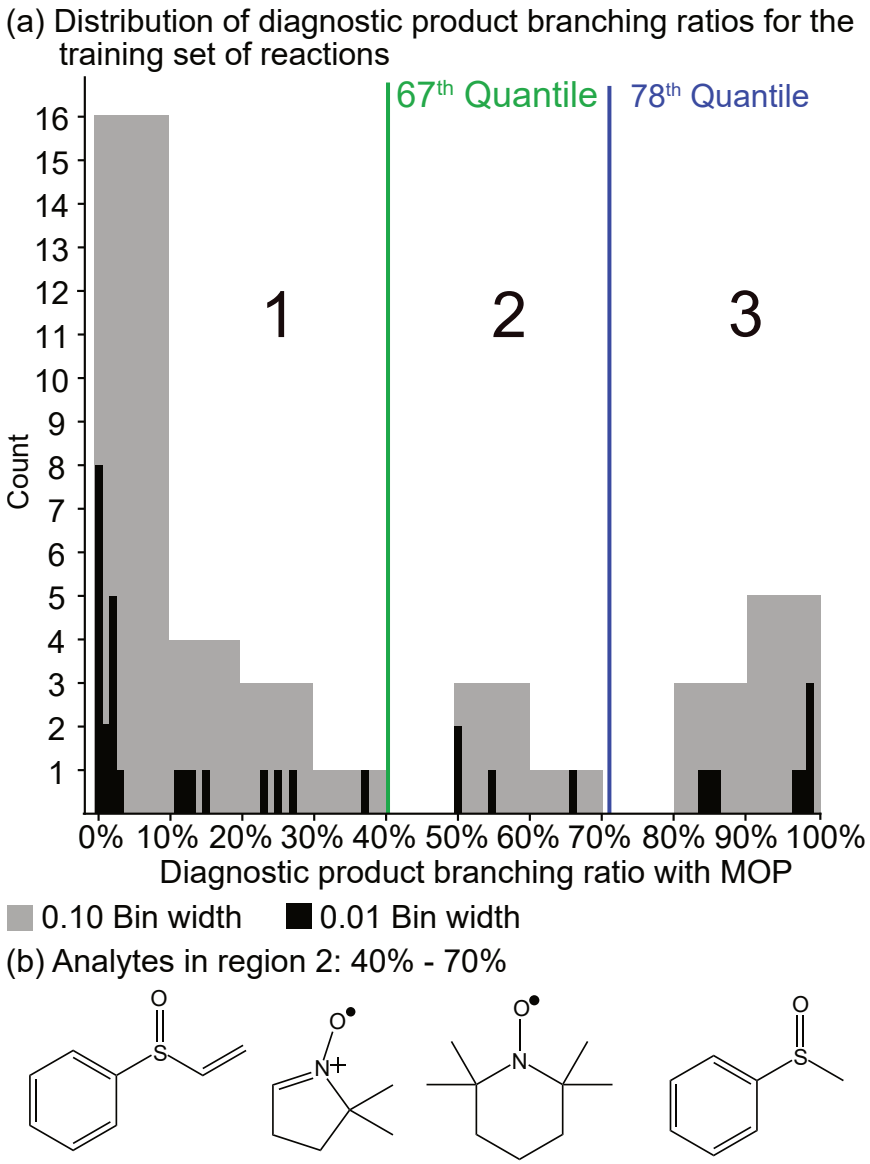


Figure 5.4. (a) The distribution of diagnostic product branching ratios for the initial training set of 36 reactions. (b) Structures for representative analytes with diagnostic product branching ratios between 40 and 70%.

strapped set of models trained with different diagnostic branching ratio cutoffs. In addition, models were trained using different fingerprint radii to ensure that a radius of 1 is appropriate (see Introduction for details). These 14 compounds (Table 5.3.2) were selected from an in-house library of available compounds and the model was

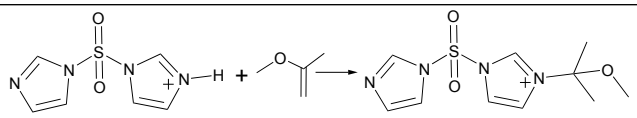
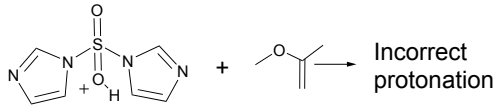
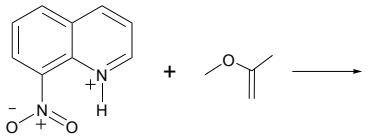
prospectively tested using ion-molecule reactions with MOP. These 14 compounds were selected based on a criterion that either their functional groups were not present in the compounds of the training set or all bootstrapped decision tree models resulted in the prediction of formation of a diagnostic addition product with MOP. The results are shown in Table 5.3 and appendix H. The probabilities of the analytes to form a diagnostic product as assigned by the radius 1 decision tree models are given in Table 5.3.2 and for other radii in appendix H. These tables show that the 60 and 70% cutoffs produced the models best suited for the external test set with a kappa value (0.72) that is greater than for the other cutoff values. The prediction prob-abilities for the analytes that underwent no diagnostic reaction (#3, #6, #7, and #9) were zero in the 70% cutoff model but above 30% in the 60% cutoff model. Therefore, the 70% cutoff was superior to 60% as it produced lower probabilities of diagnostic addition product formation for analytes that predominantly reacted via proton transfer or not at all. Additionally, other machine learning methods, including regularized logistic regression, k-Nearest Neighbor, and partial least squares classification (Tables S4-S7), were evaluated. None of these methods outperformed the 70% decision tree model trained with a fingerprint radius of 1. Finally, the proton affinity model achieved a kappa value of 0.44, indicating that the decision tree model significantly outperformed the manual approach of identifying reactions based on proton affinities. One should note that the proton affinities relevant to test reactions #1 and #9 and the calculated proton affinity of MOP are all within 0.1 kcal/mol of each other. Therefore, the correct ordering of these proton affinity values may not have real significance. Moreover, using the experimental value for the MOP proton affinity instead of the calculated value results in a kappa value of 0.31, further demonstrating the superiority of the decision tree model (kappa = 0.72) over that of proton affinity calculations.

Table 5.2.: The probability for assignment of a correct reaction for all decision tree models.

#	DP[a]	20%	30%	40%	50%	60%	70%	PA
1	Yes	51%	54%	50%	47%	100%	100%	214.43[b]
2	No	0%	8%	0%	0%	0%	0%	225.23[b]
3	No	0%	8%	0%	0%	33%	0%	229.51[b]
4	Yes	100%	100%	100%	100%	100%	100%	206.80
5	No	0%	0%	0%	0%	0%	0%	188.57
6	No	59%	58%	50%	44%	4%	0%	222.71[b]
7	No	0%	0%	0%	0%	33%	0%	195.01
8	Yes	100%	100%	100%	94%	100%	100%	224.15[b]
9	No	0%	0%	0%	0%	33%	0%	214.36
10	No	100%	100%	100%	100%	100%	100%	213.07
11	Yes	100%	100%	100%	100%	100%	100%	222.83[b]
12	No	100%	100%	100%	94%	100%	100%	205.64
13	Yes	100%	100%	100%	88%	100%	100%	226.38[b]
14	Yes	100%	100%	100%	100%	100%	100%	232.58[b]
κ		0.59	0.59	0.59	0.57	0.72	0.72	0.44

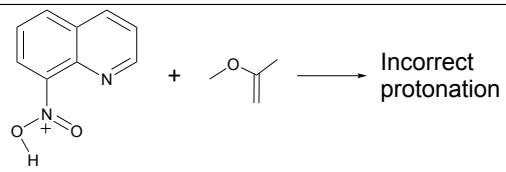
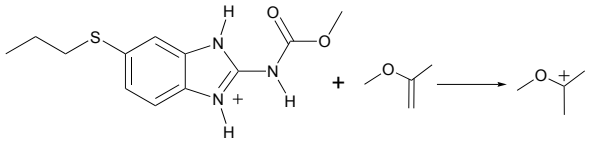
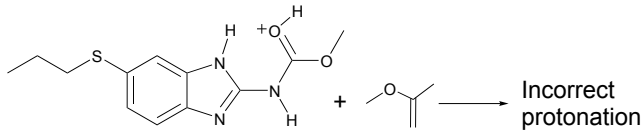
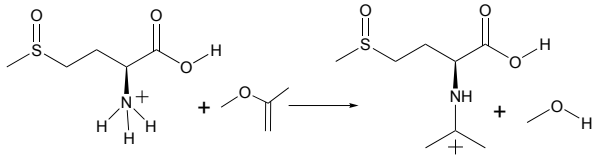
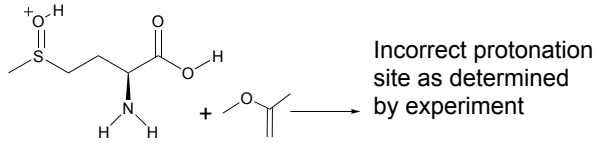
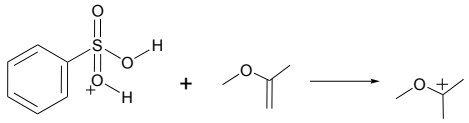
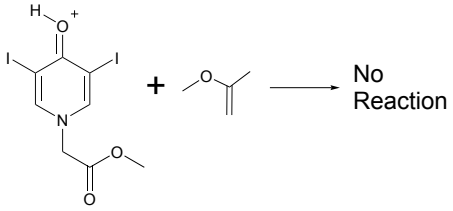
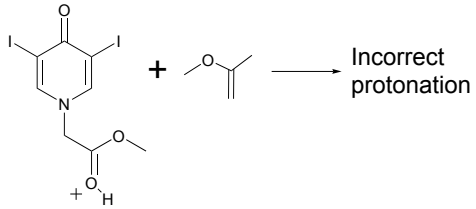
[a] See appendix H for assignment of diagnostic production formation.
[b] Value greater than the proton affinity of MOP (214.42 kcal/mol) as calculated using Density Functional Theory, see section *Calculation of proton affinities* in methods for details

Table 5.3.: Additional details for the calculation of PA for the test set reactions.

Test Set Reaction #	Proton affinity kcal/mol	Reference
1a 	214.43 [a]	Ammonia
1b 	170.12	Methanol
2a 	225.23 [a]	Ammonia

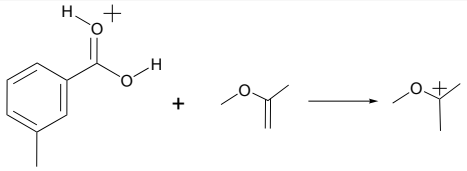
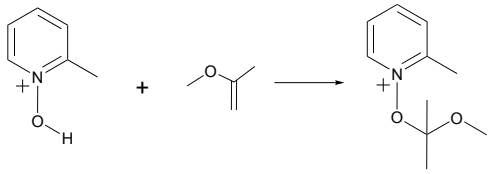
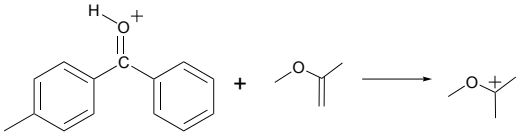
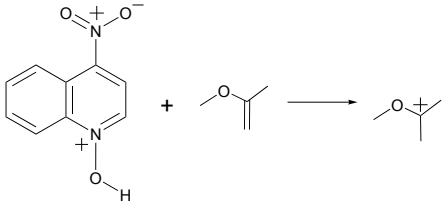
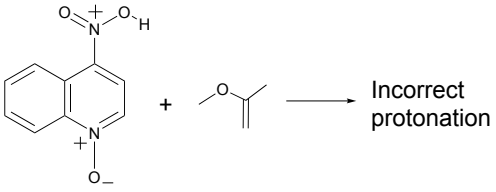
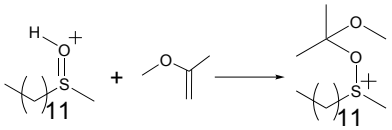
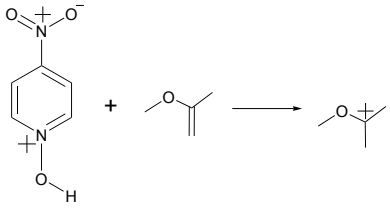
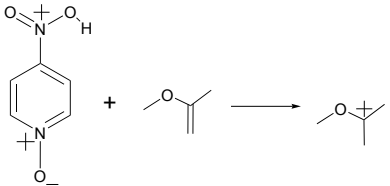
continued on next page

Table 5.3.: *continued*

Test Set Reaction #	Proton affinity kcal/mol	Reference
2b 	199.46	Methanol
3a 	229.51 [a]	Ammonia
3b 	190.26	Methanol
4a 	206.80	Ammonia
4b 	223.68 [b]	Methanol
5 	188.57	Methanol
6a 	222.71 [a]	Methanol
6b 	165.34	Methanol

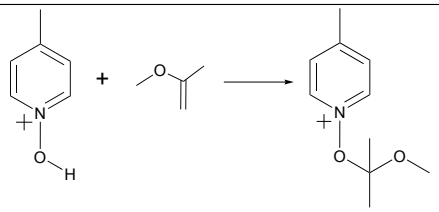
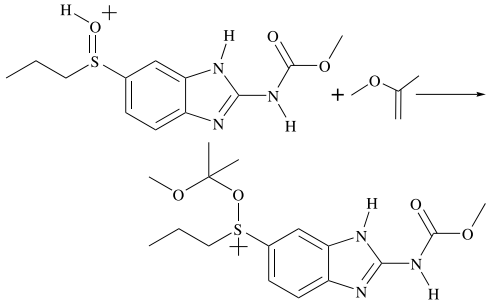
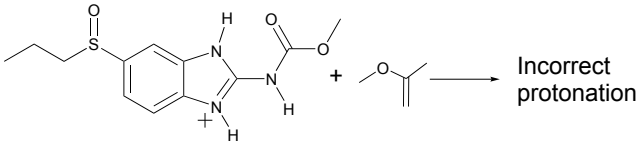
continued on next page

Table 5.3.: *continued*

Test Set Reaction #	Proton affinity kcal/mol	Reference
7 	195.01	Methanol
8 	224.15 [a]	Methanol
9 	214.36	Methanol
10a 	213.07	Methanol
10b 	186.45	Methanol
11 	222.83 [a]	Methanol
12a 	205.64	Methanol
12b 	179.97	Methanol

continued on next page

Table 5.3.: *continued*

Test Set Reaction #	Proton affinity kcal/mol	Reference
13 	226.38 [a]	Methanol
14a 	232.58 [a]	Methanol
14b 	223.44 [a]	Ammonia

[a] Value greater than the proton affinity of MOP (214.42 kcal/mol) as calculated using Density Functional Theory, see section *Calculation of proton affinities* in methods for details

[b] Although the proton affinity of the sulfoxide is greater than that of the amine, the MS/MS validation for this reaction shown in appendix H indicates that the amine is protonated as the diagnostic product formed from this reaction has a mass of M+H+MOP-MeOH instead of M+H+MOP. The mechanism for this loss of methanol requires the amine to be protonated and is not observed when a sulfoxide is protonated and the result of this experiment requires further study which is beyond the scope of this work. We have assigned this as an incorrect prediction for the PA model as a result.

Given the straightforward interpretability of decision tree models, we introduce a chemical reactivity flowchart to rationalize the logic behind the 70% model used here to make predictions. The decision tree flow chart for the 70% cutoff and the fingerprint radius of 1 atom is given in Fig 5.5 and a chart for the 40% cutoff is provided in Fig 5.6. The logic begins by checking for the presence of a sulfoxide functionality with at least one aliphatic carbon atom bound to it in the analyte and if found, the analyte

is assigned as "reactive" (see Fig 5.5d). Then, the model checks for the presence of a nitrogen atom with three substituents in a heteroaromatic ring (note that dashed lines indicate an aromatic bond) and assigns the analyte as "reactive" if such an atom is present. If neither functional group is present, the model checks for a junction between sp^2 hybridized atoms and assigns analyte containing this group as "reactive". If this group is not present, the model checks for a sulfoxide group located next to one or more aromatic rings and assigns the analyte as "reactive" if the sulfoxide group is between two aromatic rings. After this, the model checks for a terminal carbon bound to any atom and as-signs all analytes lacking this functionality as "unreactive". Those analytes that contain this functionality are checked for terminal oxygens or carbonyl groups and compounds lacking these functionalities are checked for a hydroxylamino group for final "reactivity" assignment. It should be noted that these features are identified by the trained decision tree model and that they make chemical sense in several cases, such as that com-pounds containing sulfoxide group with at least one aliphatic carbon atom bound to it (feature) generating the diagnostic product with MOP (Fig 5.5).

All cutoff models correctly predicted the three test sulfoxide compounds (#4, #11, and #14 in Table 5.3.2) to be "reactive" to-wards MOP with 100% certainty; a result which can be ex-plained by the fact that all protonated sulfoxides in the training set, except for one, had a reaction efficiency greater than 40%. Therefore, this result reflects the true experimental conclusion regarding sulfoxide compounds. This concept was reflected by the presence of a sulfoxide group as the paramount feature in the model (at the top of Fig 5.5a-c). Similarly, all the models predicted that analytes containing an N-oxide functionality are "reactive" (#8, #10, #12, and #13). However, experiment results show that compounds containing nitro groups (#12 and #10) are "unreactive" (do not undergo a diagnostic addition reaction).

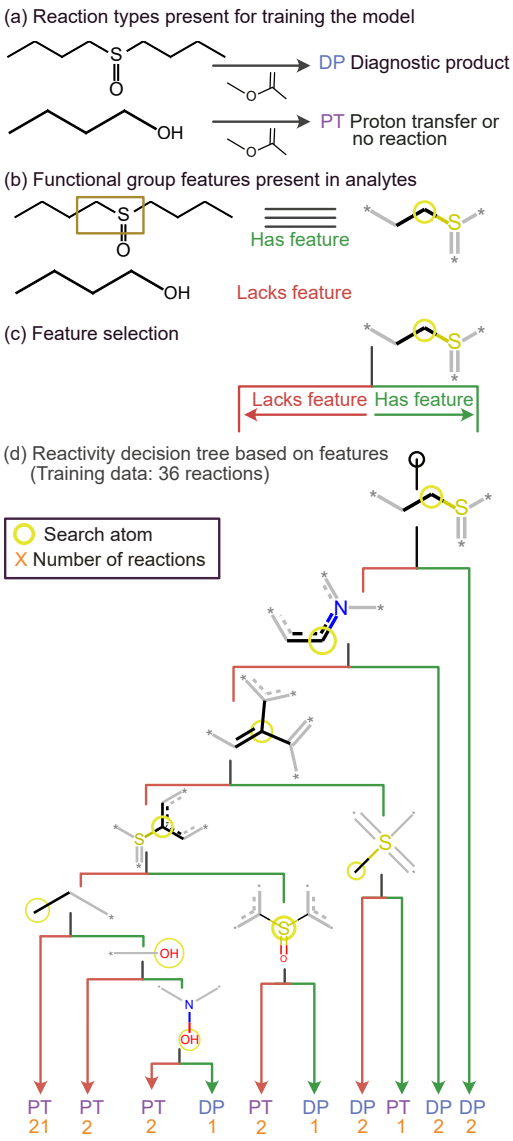


Figure 5.5. (a) Analytes that form the diagnostic product (DP) or undergo proton transfer or no reaction (PT). (b) Compounds identified as having a specific functional group feature (left), such as a sulfoxide with at least one aliphatic carbon atom bound to it (right). No structure is shown when the feature (sulfoxide) is absent in the molecule that does not form a DP. (c) Flowchart for decision making based on the presence or absence of the feature (sulfoxide). (d) The decision tree model trained on a diagnostic product branching ratio cutoff of 70%. The model classifies analytes as reactive or unreactive towards MOP based on their functional groups determined by the Morgan algorithm with a radius of 1 atom.

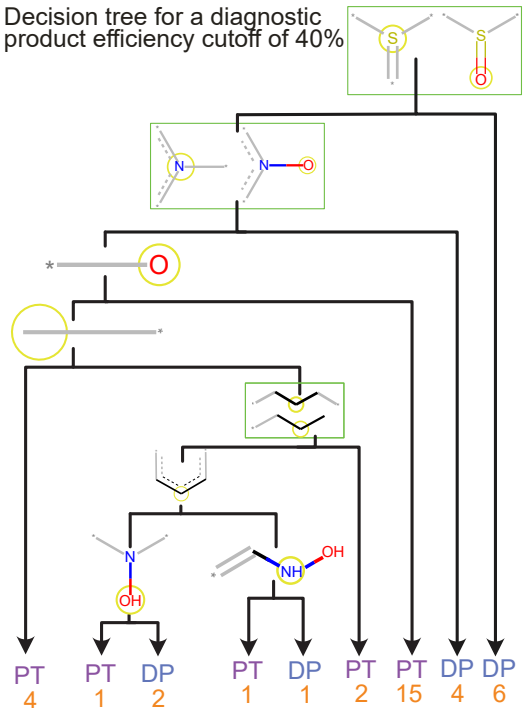


Figure 5.6. Decision tree for the 40% cutoff model. This model shares some similarities to the 70% cutoff model presented in the main text in that it uses the presence of a sulfoxide group and a N-oxide group as the primary features for the prediction of whether a compound forms a diagnostic addition product (DP) over proton transfer or no reaction (PT).

These two compounds represented the only two errors made by the 70% cutoff model and these failures may be due to a nitro group not being present in compounds in the training set. The proton affinity model, however, correctly predicted these two nitro compounds as "unreactive" towards MOP, suggesting that when new functional groups are added into the model, a proton affinity verification step could be used to ensure that the new reaction predictions are correct. Since proton affinity model in-correctly predicted that compounds #2, #3, and #6 will form diagnostic addition products and that compound #1 will not, and none of these compounds contain functional groups present in the training set, it is best to apply this verification only

if the compound contains functional groups not present in compounds in the original training set.

5.3.3 Retraining the decision tree model on new reactions.

To en-sure that the introduction of new data does not cause extensive changes to the decision tree model, a new model was trained with the addition of 14 analytes to the initial 36 analytes in the training set. The new model obtained by training with all these 50 analytes is shown in Fig 5.7. The minimal changes in the chemical features seen in this model indicate that the new model does not have many logical changes as compared to the previous model shown in Fig 5.5. The first three comparisons were the same between both the original 36-analyte model and the new 50-analyte model and the new model only introduced four additional functional groups. Three of these new functional groups were related to the nitro group present in the compounds in the new training set: 4–nitropyridine N–oxide and 4–nitro–quinoline N-oxide. Therefore, one can deduce that the model has added an additional comparison to prevent these com-pounds from being predicted as "reactive". As more protonated analytes with known reactivities towards MOP are identified, this model can be retrained to incorporate these new analytes, yielding improved predictions in the future while retaining baseline performance and simplicity.

5.4 Conclusions

The work presented here demonstrated that a combination of machine learning and tandem mass spectrometry experiments based on diagnostic ion-molecule reactions can be used to identify analytes in a semiautomated fashion while generating results in a manner readily understandable to chemists. This ma-chine learning methodology

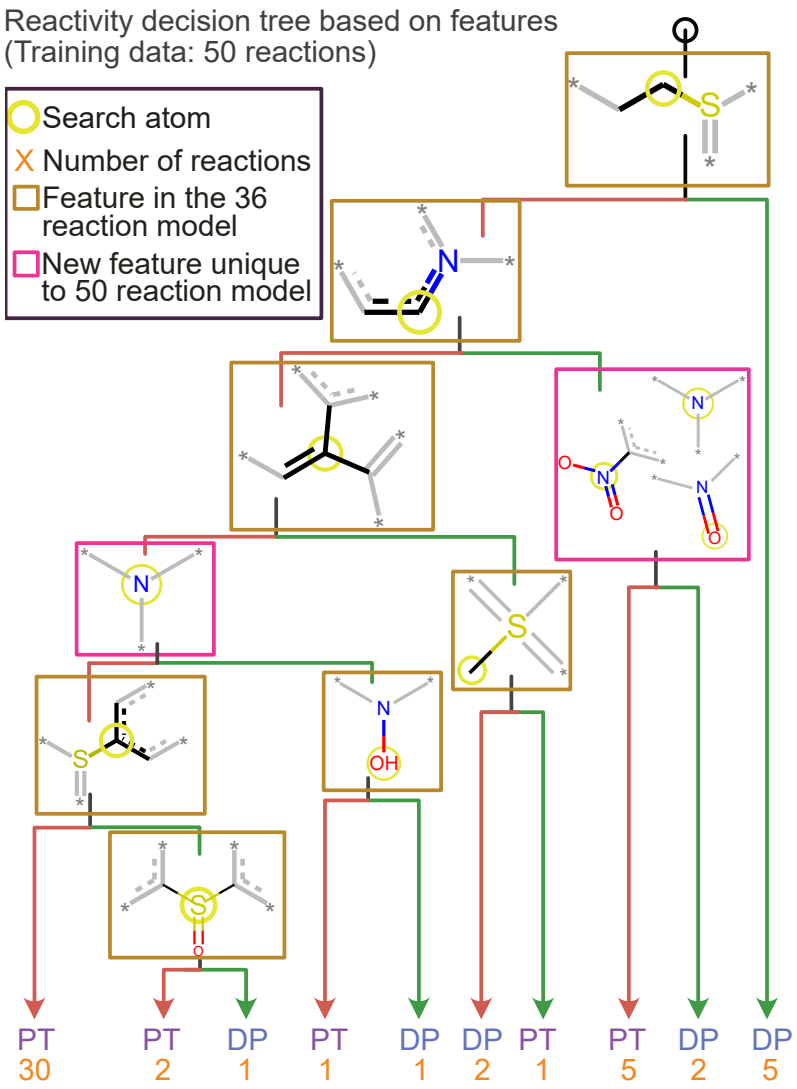


Figure 5.7. The decision tree model obtained by retraining the first model by using the 70% cutoff and all 50 reactions (original 36 and new 14 test reactions). This model is similar to the one obtained via a training set of 36 reactions but has an additional check for a nitro group which was not included in the original model. The lack of any major changes from the model shown in Fig 5.5 indicates that the final model is robust and is able to incorporate new functional groups.

combined an automated functional group identification method (Morgan Fingerprinting) with a decision tree model trained on only 36 analytes and was prospectively validated using 14 external analytes of unknown experimental outcomes. The model correctly predicted reactivity for 12 of the 14 analytes present in the test set without any additional proton affinity-based QM calculations, and 14 of 14 analytes when an additional QM filter based on the relevant proton affinities was applied. In addition to outperforming other traditional machine learning models, the decision tree model is easily interpretable by humans using the chemical reactivity flowcharts shown in this work. Additionally, the inclusion of new data resulted in only minor changes to the model as op-posed to the creation of an entirely new model, which suggests a robust selection of chemical features.

The methodologies presented herein will pave the way for expanding the above MS/MS method to include new diagnostic reactions for the identification of many different functionalities in, for example, drug metabolites in an easy, accurate, and automated manner. The ultimate goal of this research is to develop methodology for the fast determination of unknown isomeric metabolites of medicinal compounds via the identification of diagnostic product ions formed with selected neutral reagents. In the future, a fully automated pipeline for mixture component identification incorporating multiple models similar to the one presented here will be showcased along with how this methodology can be used to aid in the development of new therapeutics. The detailed output of all machine learning models is given in the Supporting Information along with the MS/MS spectra measured for all MOP reactions not previously reported in the literature. Additionally, all computer code, machine learning inputs, and other relevant scripts are provided on our GitHub page: $https://www.github.com/chopralab/mop\_reactivity\_analysis$.

5.5 Methods

5.5.1 Mass Spectrometry

All experiments were performed using a Thermo Scientific linear quadrupole ion trap mass spectrometer (LQIT) equipped with an atmospheric pressure chemical ionization (APCI) source and operated in positive ion mode. Sample solutions were prepared at concentrations ranging from 0.01 to 1 mg/mL with methanol as the solvent. The solutions were injected into the APCI source through a syringe pump at a rate of 15 $\mu L/min$ by using a 500μL Hamilton syringe. In the APCI source, typical flow rates for sheath and auxiliary gases (N_2) were 30 and 10 (arbitrary units), respectively. The vaporizer and capillary temperatures were 300 and 275 $^{\circ}C$, respectively. The ions generated upon APCI were transferred into the ion trap. The voltages applied to the ion optics were optimized for each protonated analyte via the tune feature of the LTQ Tune Plus interface. The neutral reagent, MOP, was introduced into helium buffer gas line of an external reagent mixing manifold via a syringe pump operating at a rate of $5\mu L/h$ [285, 310]. The surrounding areas of the syringe port were heated to about 120 $^{\circ}C$ to ensure that MOP evaporated completely. MOP was then diluted and directed into the ion trap by a constant flow of helium gas, controlled by a leak valve. Protonated analytes were isolated using an isolation width of 2 m/z units and a q value of 0.25, and then allowed to react with MOP in the ion trap for up to 10,000 ms. After this, all ions were detected using external electron multipliers. The MS/MS results for the test sets used in this paper are given in appendix H.

5.5.2 Creation and evaluation of the Decision Tree models.

The prediction of adduct formation of a protonated analyte with MOP is possible through a combination of fingerprinting techniques and corresponding machine

learning techniques. For each reaction, the protonated analyte and adduct were written as a stoichiometrically-balanced reaction-SMILES string. The field for the *name* of the reaction is annotated with the diagnostic product ratio as shown in Table 5.1. This reaction was then converted to a Morgan fingerprint [300] using the RDkit software package [272] with a radius of one, two, and three atoms and a bit length of 2048 bits. For the sample case presented herein, 36 reactions (training set) of known protonated analytes with the MOP reagent were examined [281, 306, 307] in the decision tree model and each reaction was assigned a binary response of "no-hit" or "hit" based on the branching ratios of the products. The deci-sion tree models were created using, the Julia implementation of Decision Tree, DecisionTree.jl ($https://github.com/bensadeghi/DecisionTree.jl$) using a minimum leaf size of 2 to reduce over-fitting to a single analyte. Details of this methodology can be found in appendix A.

A bootstrapping technique was used to address the fact that the creation of an individual decision tree model relies on the selection of random input features to be used as splits. Through this technique, 10,000 decision tree models were created for each radius and cutoff value and the frequency of each functional group used by the models was measured along with the number of times a given test analyte was predicted to be "reactive" toward MOP. The frequencies of the functional groups were used to create the chemical reactivity flowcharts shown in Fig 5.5 and Fig 5.7 and Fig 5.6.

For the logistic regression [293], partial least squares [296], generalized linear models [297], and k-Nearest Neighbor predictions [298], the Caret software package [311] was utilized to create and evaluate the models. A simple grid search was performed to obtain a set of optimal hyperparameters. The input features were the Morgan Finger-

print bit-vectors and the output was the binary out-come of whether the protonated analyte would be "reactive" to-ward MOP.

5.5.3 Calculation of proton affinities

All quantum chemical calculations were performed using Gaussian16 revision B.01 [312] and the M06–2x density functional [313]. The 6-311++G(d,p) basis set was employed for all compounds except for 3,5–diiodo–4-pyridone–1–acetic acid that was calculated using the D-Gauss Double Zeta Valence Polarized basis-set (DGDZVP) to account for the iodine atoms [314]. The three-dimensional structures for all analytes were constructed using the *Clean Structure in 3D* feature as implemented in MarvinSketch [228]. Then, GaussView [315] was used to add protons to generate protonated molecules (see Table 5.3 for the location of the additional proton). The resulting structures were optimized and the difference between the electronic energies for the neutral and the protonated molecules was determined and compared to the known proton affinity of a simple reference compound used in an isodesmic reaction. Here, methanol was used when the proton affinity was calculated for an oxygen atom, ammonia was used when the proton affinity was calculated for a nitrogen atom, and 2–methyl propene is used for MOP. See Table 5.3 for individual proton affinity values and the associated content on *https* : *//www.github.com/chopralab/mop_reactivity_analysis* for the Gaussian16 input and output files respective to the aforementioned calculations.

5.6 Future work

5.6.1 Inclusion of additional functional groups

The next immediate steps of this research is to add additional neutral reagents to the prediction pipeline. While the methodology presented in this chapter can be used on a per neutral reagent level, it is not recommended as this approach cannot take advantage of cross–learning. Ideally, the next step steps will we a more sophisticated approach than a simple binary classifier and not rely on fingerprinting to accomplish its tasks. The next method may need to incorporate additionally QM calculations as inputs to the model. Once a model (or models) is created, additional algorithms can be used to systematically determine the functional groups in a given analyte. Unfortunately, it is difficult to gain full structural elucidation from this method, but achieving the goal of determining a metabolite can easily be achieved within the next few years. In the final chapter, a potential solution to this issue is suggested using more advanced machine learning architectures.

5.6.2 Development of a novel method for storing and analyzing molecular data

In this chapter, molecules are treated as graph structures and machine learning is used to associate sub-graphs of these molecules with their reactivity towards MOP. While this treatment is useful in this context, it does not capture the 3–D information of the molecule. These features can lead to different reactivities and other phenomenon. Therefore, an intriguing improvement to these methods should add these additional features. Unfortunately, there is no standard method to store this type of data. To address this, the Chemical Index for Properties based on Hierarchical Extendable Representation (CIPHER) is introduced below:

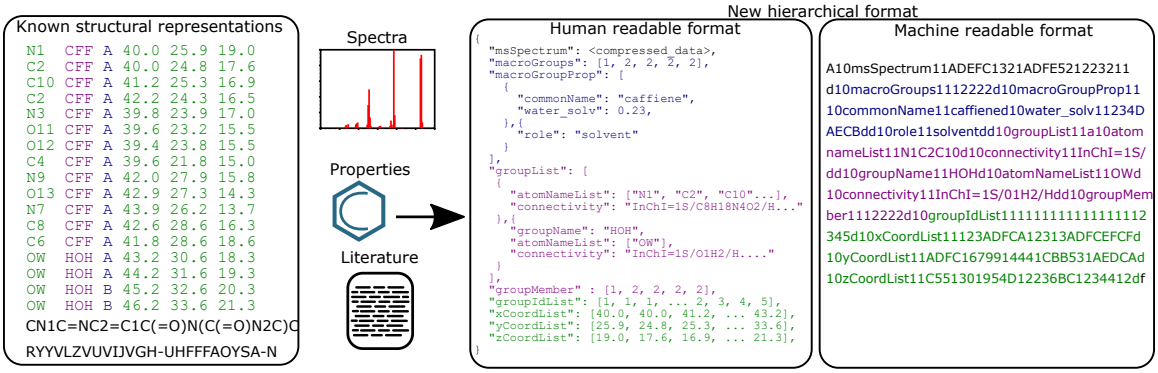


Figure 5.8. A graphical description of the cipher format

The purpose of this format is to include multiple different types of information into a single format. This information can come from patents, PubChem, the literature, etc. Relationships between these sources are then represented as a knowledge graph (see Fig 5.9). Machine learning architectures can then be used to reduce this knowledge graphs into a chemical latent space which relates various chemical properties into a single compressed representation.

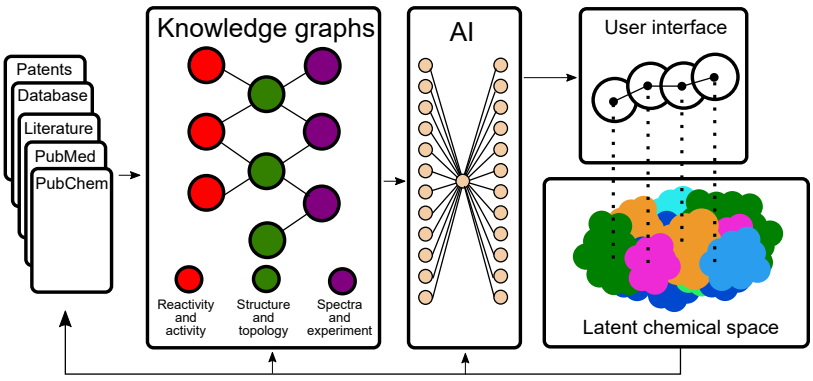


Figure 5.9. Create of knowledge graphs and the create of a latent space

After the knowledge graph for a molecule has been created, it can be associated with the graph structure of the molecule in a hierarchy. This representation can be matched with information obtained from chemical simulations obtained at various

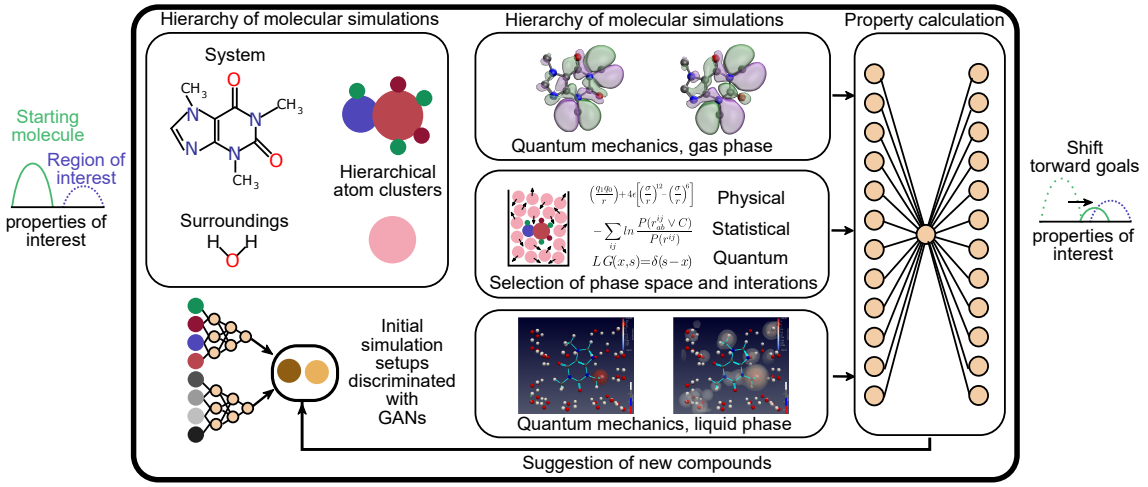


Figure 5.10. Use of knowledge graphs to optimize a molecular input towards a desired set of molecular properties

scales (quantum mechanics, statistical simulations, physical simulations, etc) and the relation ships between these simulations and the encoded property can be used to optimize a given molecular input and shift it towards a desired set of properties. There are many machine learning architectures that can be used to accomplish this task, such a Generative Adversarial Network (GAN) or policy based network.

6. DRUG DISCOVERY AT THE PROTON LEVEL – UNDERSTANDING REACTIONS AND REACTIVITY

This chapter is available as

Jethava, K., Fine, J., Chen, Y., Hossain, A., Chopra, G. Accelerated Reactivity Mechanism and Interpretable Machine Learning Model of N-Sulfonylimines Towards Fast Multicomponent Reactions. *ChemRxiv* (2020).
https://doi.org/10.26434/chemrxiv.12116163.v1

6.1 Abstract

Predicting the outcome of chemical reactions using machine learning models has emerged as a promising research area in chemical science. However, the use of such models to prospectively test new reactions by interpreting chemical reactivity is limited. We have developed a new fast and one–pot multi–component reaction of N–sulfonylimines with heterogeneous reactivity. Fast reaction times ($<$ 5 min) for both acyclic and cyclic sulfonylimine encouraged us to investigate plausible reaction mechanisms using quantum mechanics to identify intermediates and transition states. The heterogeneous reactivity of N–sulfonylimine lead us to develop a human-interpretable machine learning model using positive and negative reaction profiles. We introduce chemical reactivity flowcharts to help chemists interpret the decisions made by the machine learning model for understanding heterogeneous reactivity of N–sulfonylimines. The model learns chemical patterns to accurately predict the re-

activity of N–sulfonylimine with different carboxylic acids and can be used to suggest new reactions to elucidate the substrate scope of the reaction. We believe our human-interpretable machine learning approach is a general strategy that is useful to understand chemical reactivity of components for any multi–component reaction to enhance the synthesis of drug–like libraries.

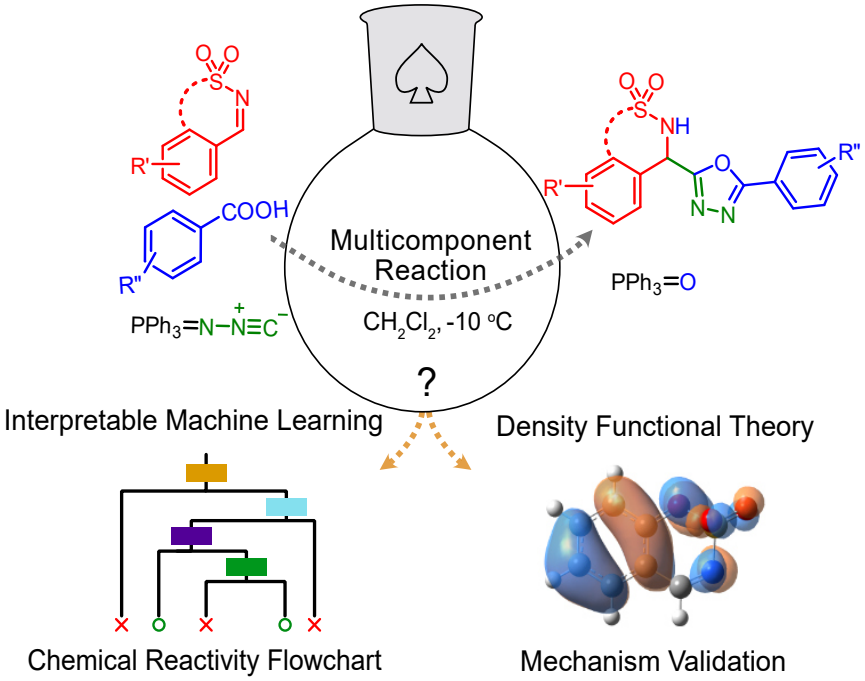


Figure 6.1. Table of contents figure for the online publication

6.2 Introduction

Computer-assisted organic chemistry has a huge potential for predicting chemical reaction conditions and for automating synthetic chemistry [258, 304, 316]. In recent years, machine learning (ML) based approaches have been successfully applied to screen libraries of drug–like molecules [317, 318], for quantitative structure–activity relationships (QSAR) [153], for retrosynthetic planning [319], and for reaction condi-

tion prediction. Reactivity prediction is a hard problem that often requires specific experimental datasets to train ML models [289, 320]. Traditionally, creating such experimental databases requires a large number of manual experiments to check the feasibility of available starting materials to react together. However, with careful training of ML models using both positive and negative reaction data, it is possible to train on smaller datasets to test specific synthetic objectives. The results from ML models are helpful in building a chemical library that is otherwise tedious to explore by screening each reaction to check substrate feasibility under certain reaction conditions. To date, there is limited literature precedence for prospective prediction of desired chemical reactions and interpreting its reactivity using machine learning methods [150, 321]. We provide a first report, to the best of our knowledge, of a fast and one-pot multi–component reaction to explore heterogeneous reactivity of N–sulfonylimines by training a human–interpretable machine learning model that identifies chemical patterns of reactivity to predict and test new reactions prospectively.

We selected N–sulfonylimines as our model substrate because N–sulfonylimines are one of the important synthons in organic chemistry that are being used for a variety of chemical transformations. N-sulfonylimine is a good source of an electrophilic carbon for radical12 [322] and nucleophilic addition [323] reactions. There are several reports available for N–sulfonylimines reactions where a carbon–nitrogen double bond is exploited [324]. Notably, the use of sulfamidate [325], a cyclic N-sulfonylimine, has been used to prepare interesting heterocyclic scaffolds. Sulfamidate is transformed into a fused heterocycle using Michael addition [326], cycloaddition [327–332], arylation [333–335], alkenylation [336–338], or alkynylation [336] strategy by leveraging electrophilicity of cyclic N–sulfonylimines (Scheme 6.2).

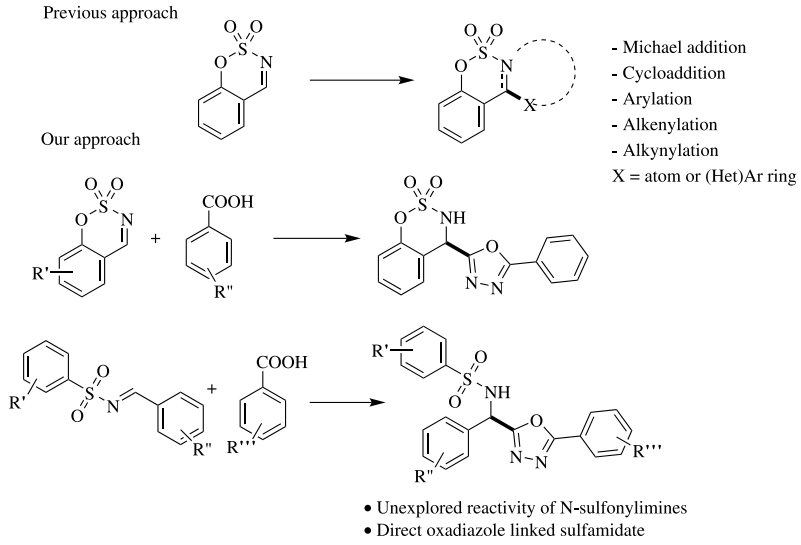


Figure 6.2. Strategy to explore N–sulfonylimine reactivity towards multi–component reaction

However, among reported synthetic strategies, direct C-C bond connection between the imine carbon and the (het)aromatic partner is underrepresented in the literature. Specifically, a synthetic strategy for the direct C-C bond linkage between sulfamidate and oxadiazole has not been explored till date. The oxadiazole scaffold finds a unique presence in many biologically active compounds [339,340], pharmaceutical agents and considered as a privileged scaffold in material science [341]. Among different types of five–membered heterocycles, 1,3,4–oxadiazole plays important in organic synthesis and medicinal chemistry representing broad spectrum bioactivities including anticancer, antimicrobial, antiviral, and antifungal pharmacological activities [342, 343] (Fig 6.3). For example, the recently discovered CA-170 contains a 1,3,4-oxadiazole moiety and is a promising immune checkpoint inhibitor in the tumor microenvironment as a dual antagonist of Programmable death ligand-1 and V-domain Ig suppressor of T-cell activation. Although the structure of CA-170 is not disclosed, a speculated structure is shown in Fig 6.3 [344]. Conventional approaches

to synthesize 1,3,4–oxadiazole is a multistep procedure that includes transformation of carboxylic acid into acyl chloride. Then a nucleophilic substitution reaction with hydrazide to produce an amide bond followed by cyclization step to get a 1,3,4–oxadiazole [345].

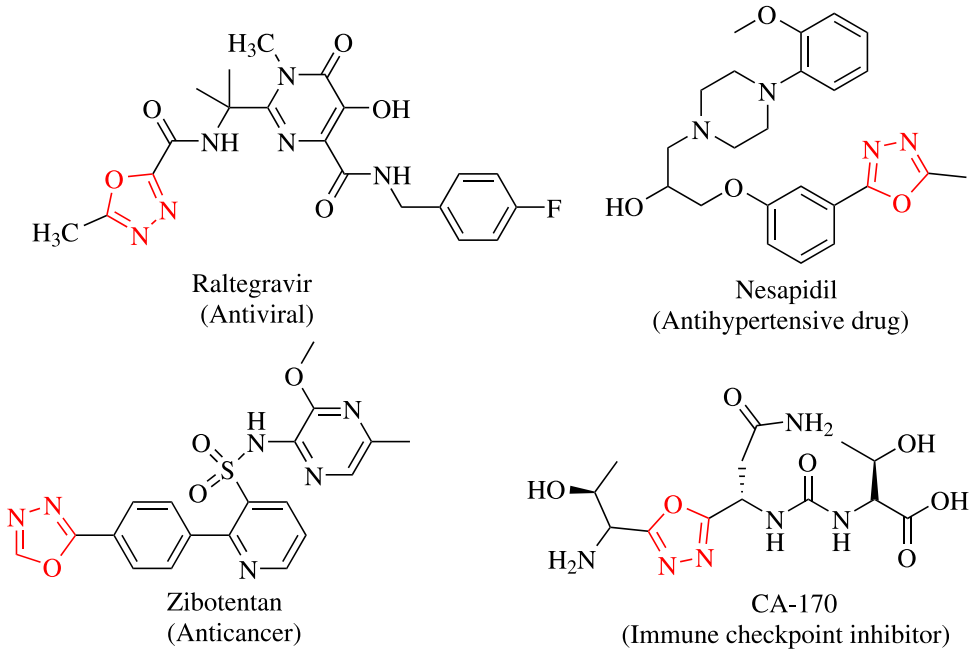


Figure 6.3. Showing compounds with presence of 1,3,4–oxadiazole in medicinal chemistry.

Multi–component reactions (MCRs) that reduce the number of synthetic steps have been attractive as they combine two or more building blocks to generate diverse chemical libraries including new heterocyclic chemical structures that are useful in medicinal chemistry [346–349]. Ramazani et al [350] reported a four-component reaction yielding 1,3,4–oxadiazole scaffold using aromatic aldehyde, benzoic acid, N–isocyano triphenylphosphorane (Pinc), and secondary amine as reaction partners. The formation of 1,3,4–oxadiazole involves an essential reactant, Pinc which is the nucleophilic partner that reacts with the imine. This species is generated *in situ* from

the amine and aldehyde and reacts with a carboxylic acid followed by cyclization to yield 1,3,4-oxadiazole. A similar strategy was extended by Yudin et al. [351, 352] to perform an intramolecular reaction for the synthesis of oxadiazole containing cyclic peptide or macrocycle where two end terminals are stapled to form oxadiazole ring. This strategy also relies upon *in situ* imine formation from an aldehyde, a secondary amine, and an additional amine group. It is noteworthy that *in situ* formations of imines are not always favorable as it is highly dependent upon its starting materials – an aldehyde and an amine, potentially limiting the use of these approaches. To address this issue, we provide the first report to use N–sulfonylimine as a substrate for a fast and single–step approach to synthesize sulfamidate embedded 1,3,4–oxadiazole using an MCR.

6.3 Results and discussion

We started our investigation with the idea that several types of cyclic N–sulfonylimines (aldimines or ketimines), acyclic N–sulfonylimines, and aromatic imines can be synthesized. To determine the reactivity pattern of various imines with carboxylic acids, we used Fukui reaction parameters calculated using Density Functional Theory (DFT) [353] and identified the most suitable imines using the electrophilicity of the carbon atom (Fig 6.5). Both cyclic and acyclic N–sulfonylimines are highly susceptible

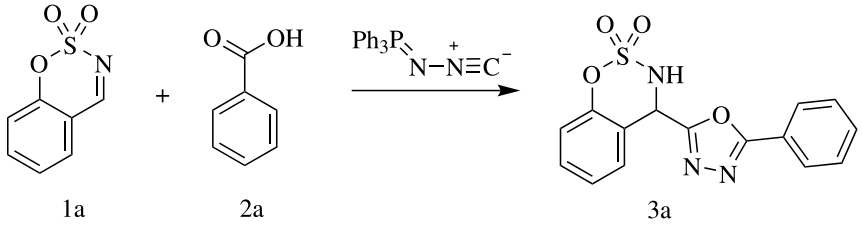


Figure 6.4. Synthesis of 1,3,4–oxadiazole using cyclic imine with benzoic acid under optimized reaction conditions.

toward nucleophilic attack of carboxylic acids. Therefore, we started using the model substrate cyclic N–sulfonylimine (sulfamidate) **1a**, which can be easily synthesized from substituted salicylaldehydes. We initially selected benzoic acid as the reaction partner because of its moderate nucleophilic tendency (Fig 6.5) and the selection of optimized conditions for future use with a chemically diverse range of carboxylic acids. Further, the synthesis of other derivatives with the optimized condition would serve as a training dataset to develop a machine learning model.

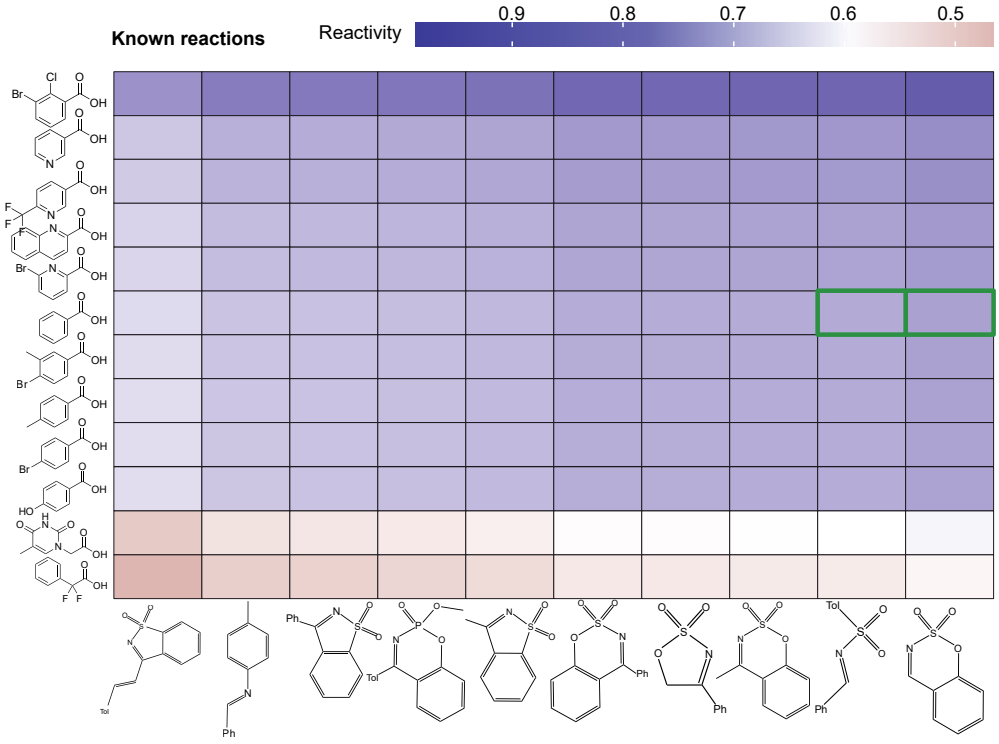


Figure 6.5. Heatmap of Fukui reaction parameters calculated for imines and carboxylic acids.

Having a synthetic and computational strategy in mind, we performed an optimization study using sulfamidate (**1a**) and benzoic acid (**2a**) to form the desired product **3a**. Reaction conditions from the literature for similar MCRs resulted in a messy TLC and trace product formation as identified using HPLC–MS (entry 1

in Table 6.1). The replacement of a mixture of solvents with only dichloroethane (DCE) and room temperature conditions gave trace amounts of product as detected by HPLC–MS (entry 2). Next, replacing dichloroethane with dichloromethane (DCM) afforded a detectable quantity of desired product **3a** (entry 3). While doing a time-point study with a 30 minutes interval, we observed that the desired product was formed within 30 minutes (entry 4). However, TLC analysis shows multiple products, so we decreased the reaction temperature. At $0°C$ the desired product formed within 5 minutes (entry 5) as determined by a 5 minute time-point study. In all the above attempts, benzoic acid was added slowly. At $-10°C$, an additional experiment where DCM is added at the end increased the yield significantly (entry 6 vs 7) – suggesting that sulfamidate has high reactivity.

Table 6.1.: Optimization of the synthesis of sulfamidate–oxadiazole[a](Scheme 6.4).

#	Solvent	Temp. (°C)	Time (min)	Yield (%)[e]
1[b]	DCE:MeCN	50	120	Messy TLC
2	DCE	25	120	Trace
3	DCM	25	120	<5
4	DCM	25	30 to 120	<5
5	DCM	Ice-bath	5 to 30	25
6[c]	DCM	0	5	∼40
7[d]	DCM	-10	5	67

[a] Reactions are at 0.1 mmol scale
[b] Reaction condition followed as per literature [351]
[c] benzoic acid added at the end
[d] solid components taken together with solvent added last
[e] isolated yield

6.3.1 Mechanism of the cyclic and acyclic N–sulfonylimines

Next, we applied the optimized reaction condition to the acyclic imine selected using DFT calculations (Fig 6.5)) as it was the second most reactive imine. Inter-

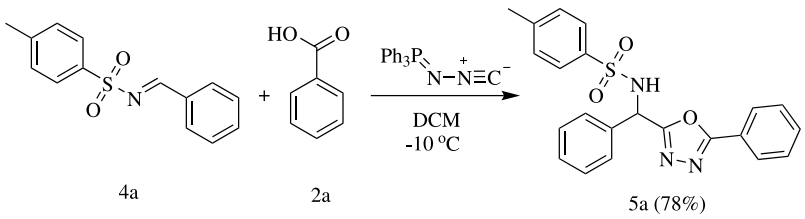


Figure 6.6. Synthesis of 1,3,4–oxadiazole using acyclic imine with benzoic acid under optimized reaction conditions.

estingly, the reaction afforded the desired product with good yield, but with longer reaction time (10 mins) for the complete conversion as compared to sulfamidate (< 5 min). This led us to investigate the mechanism and the energy profile of various plausible intermediates formed in this reaction.

To gain mechanistic insights of the chemical reactions, we conduct-ed DFT calculations using a polarized continuum model for DCM solvation at $-10°C$ to identify transition states and intermediates for acyclic and cyclic N–sulfonylimines (Fig 6.7, 6.8). The nucleophilic attack by negatively charged carbon atom of Pinc on the electrophilic center of N–sulfonylimine yields Intermediate-1. The subsequent Intermediate–2 is formed by a nucleophilic attack of benzoic acid. Next, intramolecular cyclization at the carbonyl carbon and subsequent removal of triphenylphosphine oxide yields the desired 1,3,4–oxadiazole containing the product. Both imines have the same rate-limiting step where the Pinc reagent attacks the carbonyl carbon and both steps have small activation energies (12.6 kcal/mol and 16.3 kcal/mol for the cyclic and acyclic imines respectively), suggesting both reactions will occur quickly.

The mechanisms of multi–component reaction shown in the main text using an acyclic N-sulfonylimine (Fig 6.7) and a cyclic N–sulfonylimine (Fig 6.8) show many similarities and differences that match chemical intuition to determine the relative reactivity of the two imines. After the formation of Intermediate–2, the mechanisms for both cyclic and acyclic N-sulfonylimine reactions become very similar. The differ-

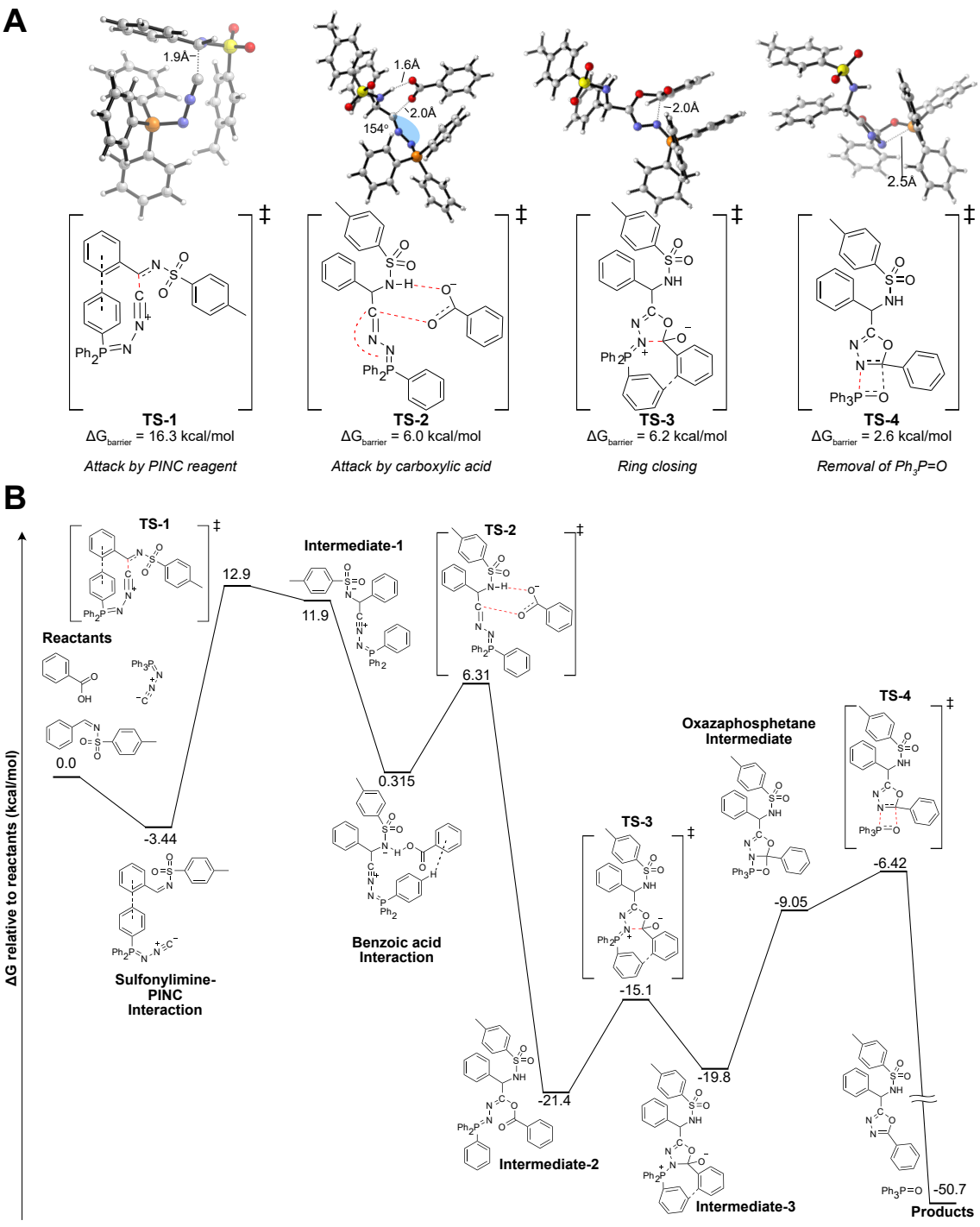


Figure 6.7. A. 3D and 2D structure of each transition state for the acyclic reaction with their respective geometries (show in red in 2D). The barrier energy for each transition state is also given. B Full mechanism with their energies shown with respect to the reactants.

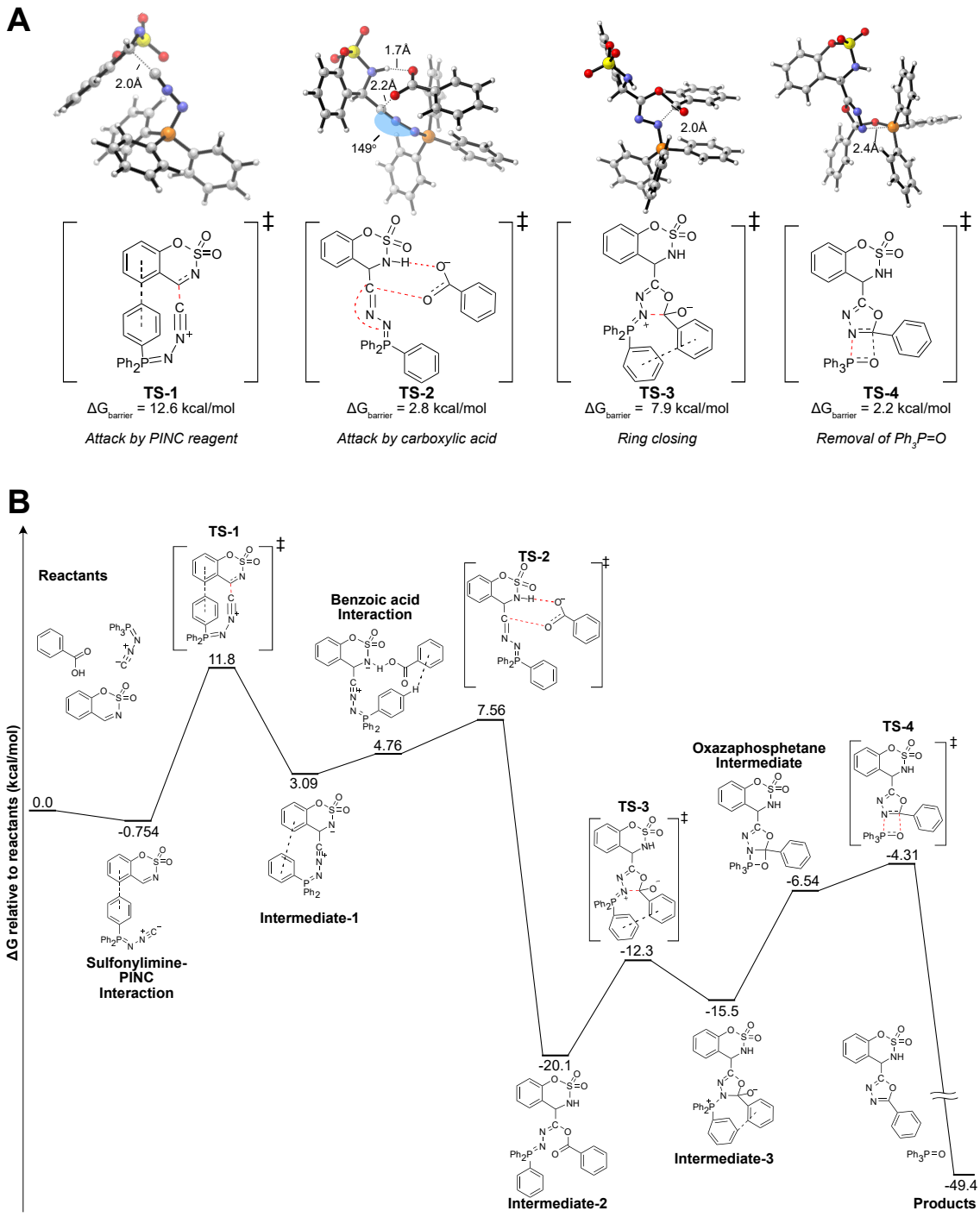


Figure 6.8. A. 3D and 2D structure of each transition state for the cyclic reaction with their respective geometries (show in red in 2D). The barrier energy for each transition state is also given. B Full mechanism with their energies shown with respect to the reactants.

ence between the changes in energy between the two mechanisms for the same step are all within 2 kcal/mol of each other as shown in Table 6.2.

Table 6.2.
Difference between the changes in energy between the two mechanisms

Event	Acyclic	Cyclic
Formation of Intermediate–2	27.7 kcal/mol	27.8 kcal/mol
$\Delta G_{barrier}$ of TS-3	6.2 kcal/mol	7.9 kcal/mol
Formation of Intermediate–3	4.7 kcal/mol	3.3 kcal/mol
Oxazaphosphetane Intermediate formation	10.8 kcal/mol	9.0 kcal/mol
$\Delta G_{barrier}$ of TS–4	2.6 kcal/mol	2.2 kcal/mol
Formation of the products	44.3 kcal/mol	45.1 kcal/mol

This similarity is expected given the proposed mechanisms as this portion of the mechanisms corresponds to the formation of the oxadiazole ring and the atoms of this ring come from only the Pinc reagent and carboxylic acid. None of the oxadiazole atoms originate from the N–sulfonylimine and therefore it is expected that this half of the mechanism would not be highly dependent on the imine.

The first half of the reaction, however, does involve the chemistry of the imine and is more dependent on whether it is cyclic or acyclic. In both cases, the rate limiting step is the attack of the PINC regent onto the imine with a $\Delta G_{barrier}$ of 16.3 and 12.6 for the acyclic and cyclic imines respectively. Therefore, it is expected that the cyclic imine would be faster than the acyclic imine, which has been shown experimentally. The majority of this difference is due to the stabilization of the PINC-imine interaction which is much greater for the acyclic imine versus the cyclic imine. This is likely explained by the higher degree of freedom in the acyclic compound and its ability to form better interactions with Pinc before it attacks. These interactions are shown in Fig 6.9.

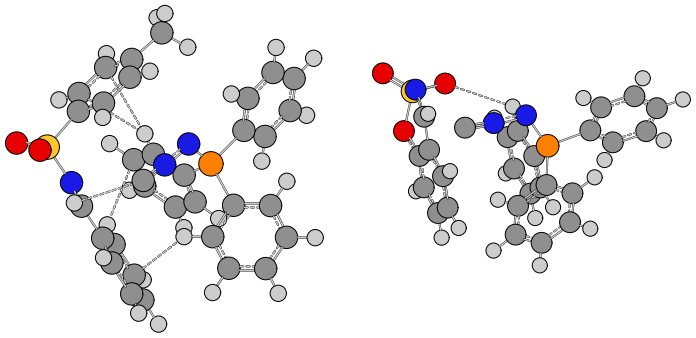


Figure 6.9. The minimized structure for the acyclic (left) and cyclic (right) imines. The increased number of interactions that the acyclic imine has with the Pinc reagent causes is a probable reason that the energy difference between this step and the following transition state is larger than for the cyclic imine.

However, these steps do not explain the difference in reactivity for the carboxylic acids as this reagent has not yet been introduced into the mechanism. The difference in energy between intermediate 1 and its interaction with the benzoic acid is negative for the acyclic imine reaction and positive for the cyclic imine reaction. This is again explained by the increased degree of freedom present in the acyclic imine and the interactions formed between the two aromatic rings present in this imine (see following Fig 6.10). The cyclic imine cannot form these interactions and therefore the interaction complex between intermediate 1 and benzoic acid is not as energetically favorable. Differences in this step may explain the increased reactivity of acyclic imines, but further work is needed to elucidate how changes in the carboxylic acid will change this reactivity.

After this step, the two mechanisms converge as discussed previously. The intrinsic reaction coordinates for the acyclic and cyclic reaction are given on the following pages, Fig 6.11, 6.12. The plots match the relative energy differences of the full mechanisms shown in the main text and they confirm that the transition states are correct for the preceding and following intermediates. All the transi-

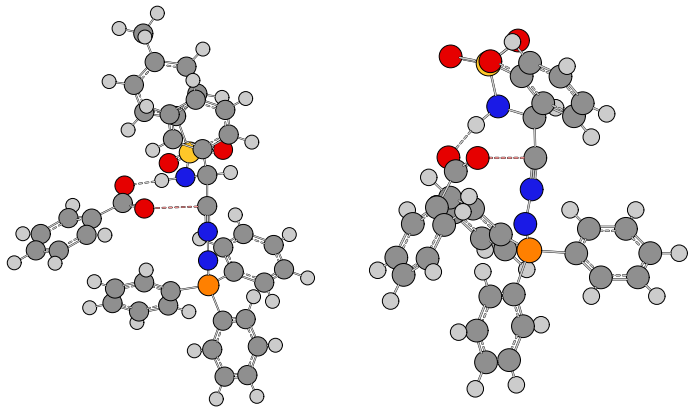


Figure 6.10. Interaction between intermediate 1 and benzoic acid for the acyclic mechanism (left) and the cyclic mechanism (right).

tion states shown have a single negative frequency and all intermediates and interaction structures have zero negative frequencies. These can all be visualized at $https://chopralab.github.io/n\_sulfonylimine\_reactions$.

6.3.2 Investigation of substrate scope

Using the optimized conditions, we started investigating various sulfamidates and carboxylic acid derivatives. The reaction of the diethylamine containing sulfamidate (**1b**) with benzoic acid afforded the desired product **3b** in 46% yield. The reaction of sulfamidate **1b** with p–toluic acid (**2b**) also formed product **3c** but in low yield (17%). Further, reaction of methoxy substituted sulfamidate **1c** with benzoic acid (**2a**) formed expected product 3c in moderate yield (52%). However, naphthyl sulfamidate (**1d**) did not react effectively giving 1,3,4–oxadiazole **3e** in poor yield. Notably, bromo derivatives of sulfamidate 1e with benzoic acid (**2a**) did not afford the desired product (**3f**). Nonetheless, when sulfamidate **1c** was reacted with pyridine carboxylic acid **2c**, it formed the expected product with inseparable isomer in poor yield. Further, 4–hydroxybenzoic acid (**2d**) did not react with sulfamidate **1c** to form

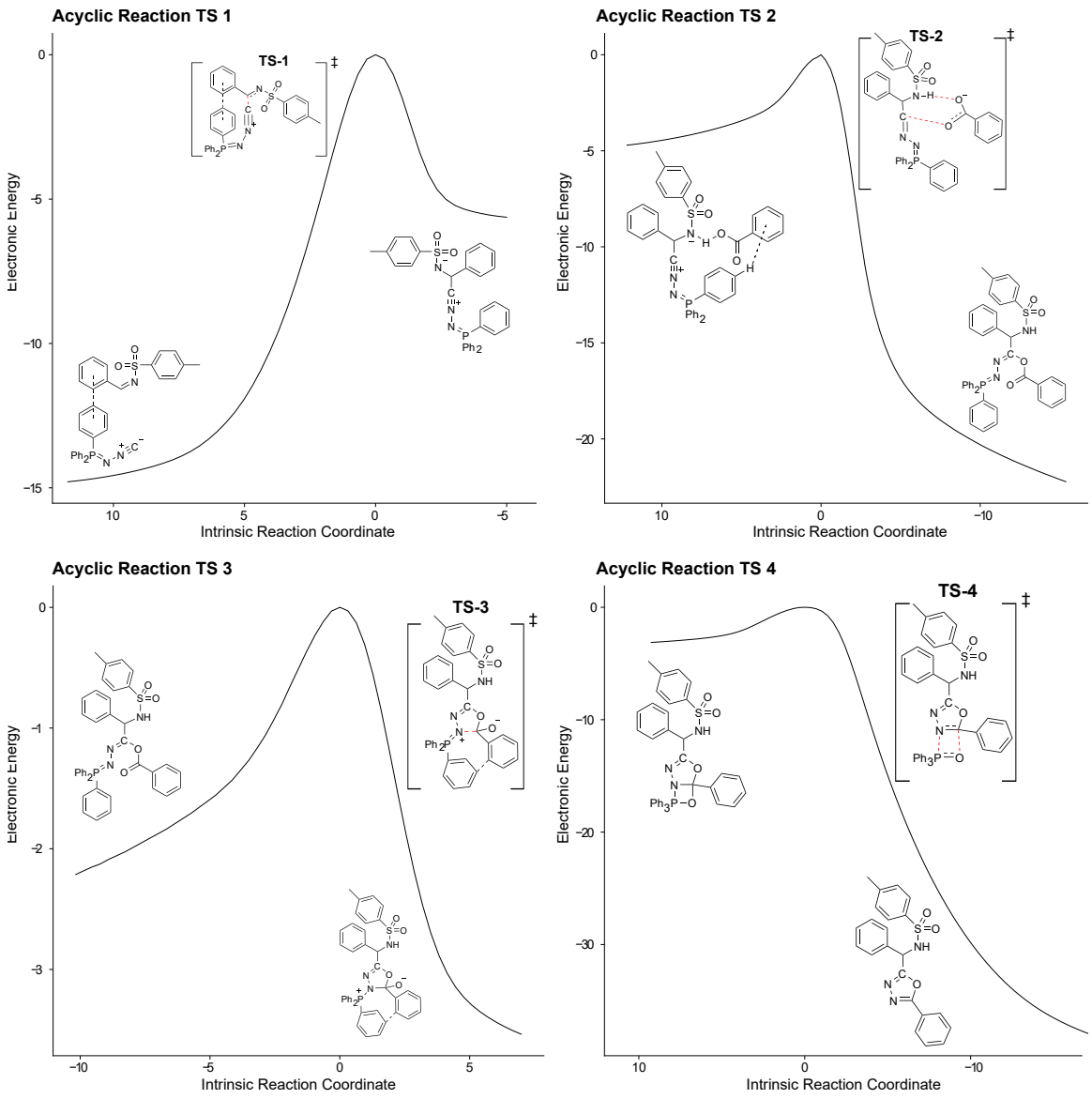


Figure 6.11. Intrinsic reaction coordinate for all steps of the acyclic reaction. Some of the steps have their coordinate flipped so that the direction of the graph matches the forward direction of the mechanism. Note that energy values may differ as they do not contain corrections for entropy or the zero-point energy correction.

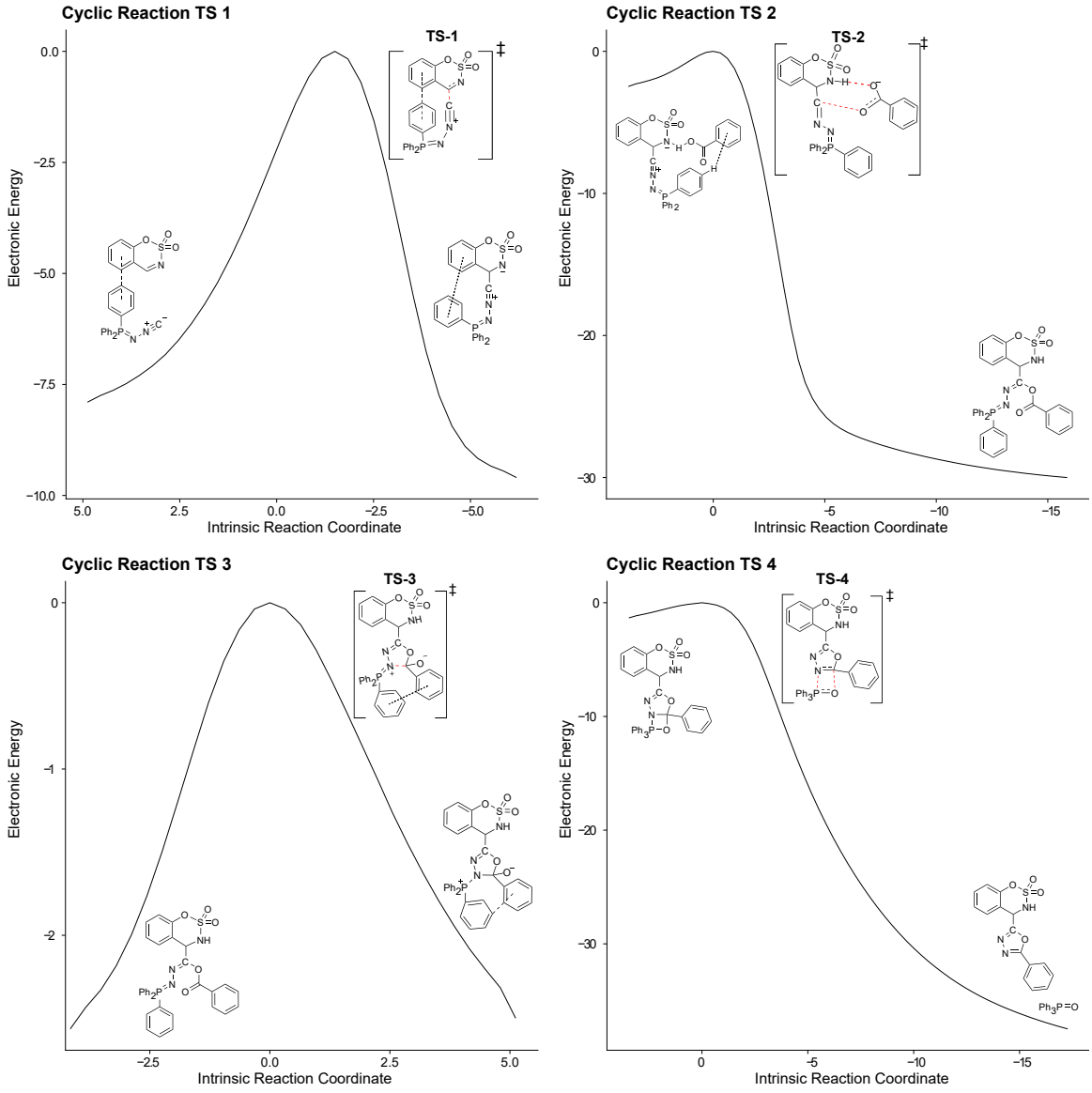


Figure 6.12. Intrinsic reaction coordinate for all steps of the cyclic reaction. Some of the steps have their coordinate flipped so that the direction of the graph matches the forward direction of the mechanism. Note that energy values may differ as they do not contain corrections for entropy or the zero-point energy correction.

desire product **3h**. Next, we also sought to study the reactivity of other carboxylic acids with sulfamidates. So, apart from the products shown in Scheme 6.13, we also attempted other reactions to study reactivity of sulfamidate with other carboxylic acids (Scheme 6.14). For example, difluoro arylacetic acid, pyrimidine–2–carboxylic acid, terephthalic acid etc. – did not react well with sulfamidates. This observation intrigued us to study the reactivity of acyclic N–sulfonylimines with carboxylic acids after successful model reaction shown in Scheme 6.6.

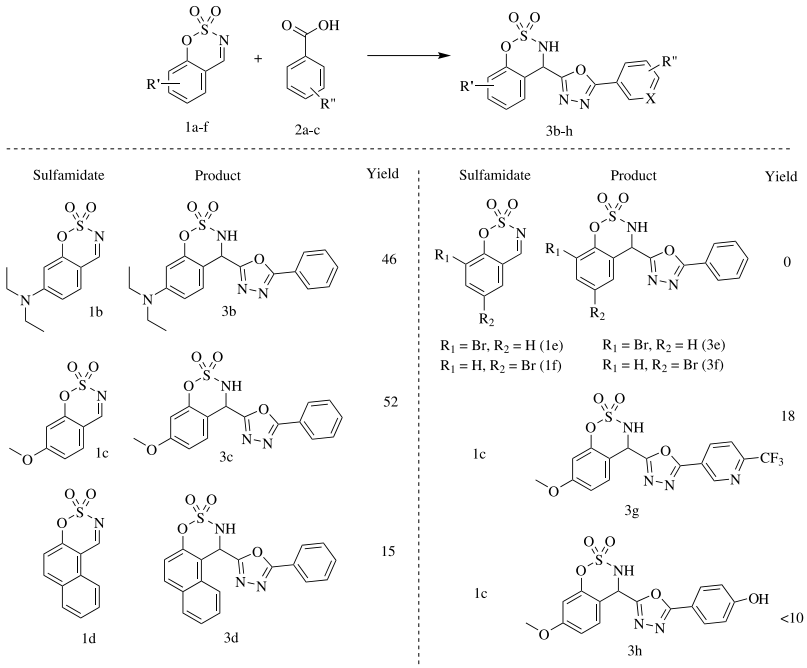


Figure 6.13. Substrate scope for representative cyclic N–sulfonylimine with various carboxylic acids.

As shown in Scheme 6.15, acyclic N–sulfonylimine substrates were reacted with benzoic acids. Unlike halogenated sulfamidates, the reaction of halogenated acyclic N–sulfonylimine **4b** reacted well with benzoic acid (**2a**) and 4–bromo–2–methyl benzoic acid (**2b**), giving desired products **5b** and **5c** in 53% and 37% yields, respectively. Further, the synthesis of 5d and 5e were achieved success-fully using trimethoxy sub-

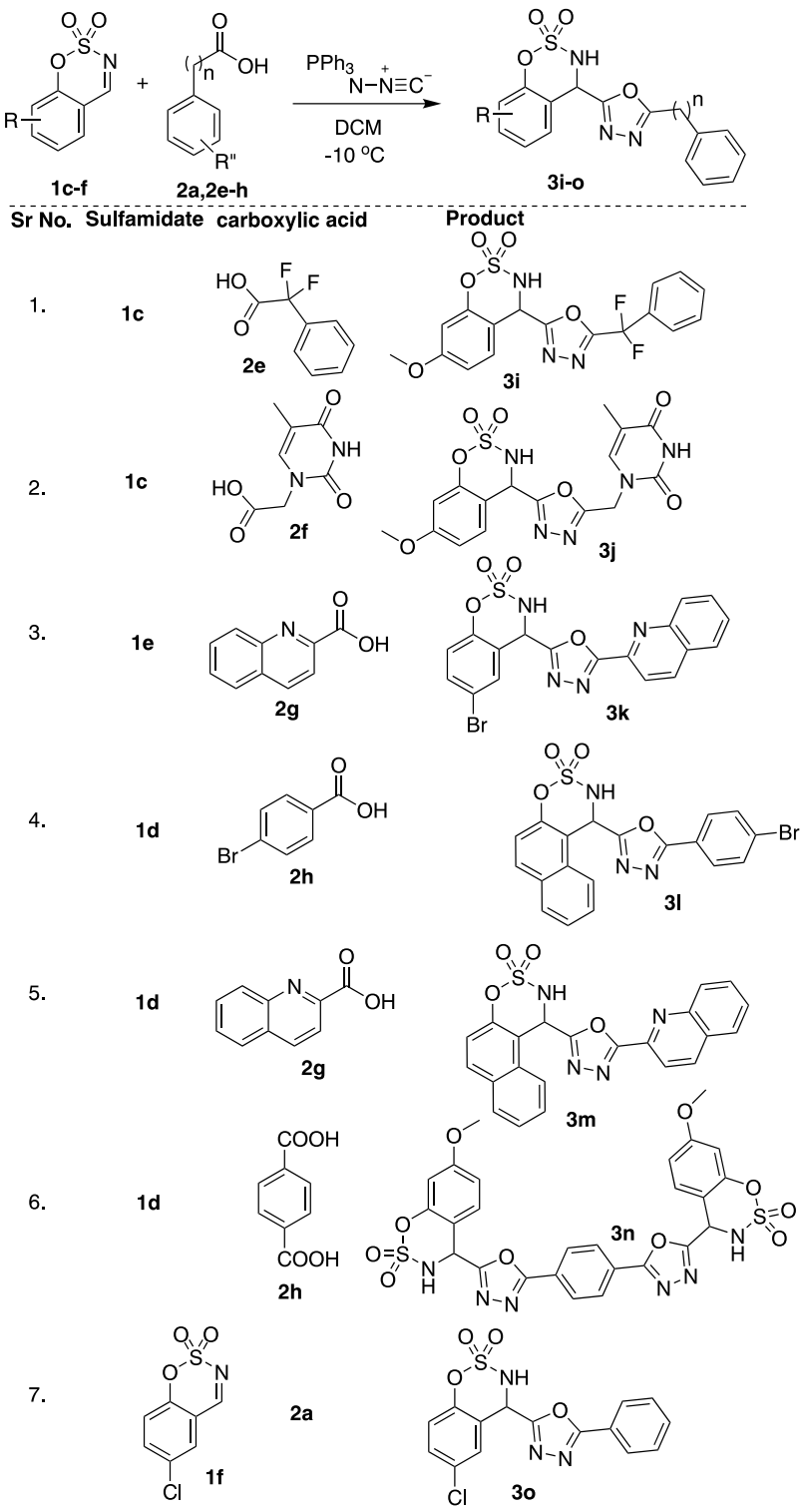


Figure 6.14. Attempted synthesis of 1,3,4–oxadiazole using sulfamidates and other carboxylic acids.

stituted N-sulfonylimine (**4c**), and 4–hydroxy 3–nitro substituted N–sulfonylimine (**4e**), and they were well tolerated to afford desired products **5d** and **5e** (70% and 64% yields, respectively).

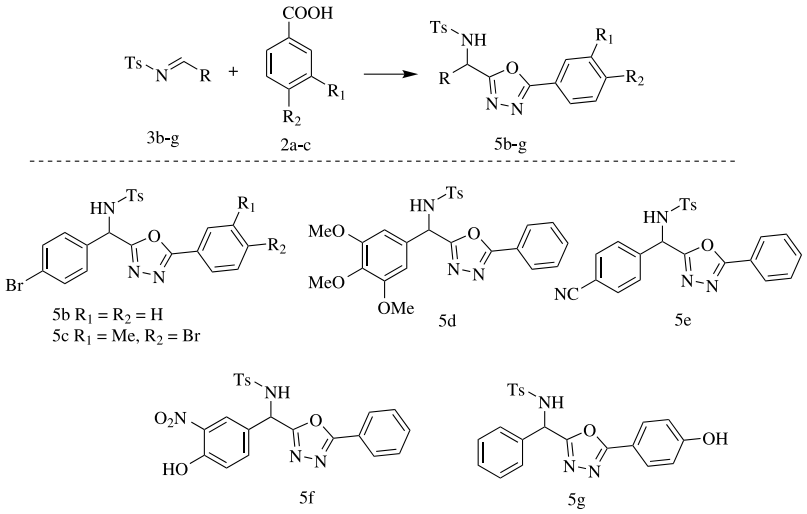
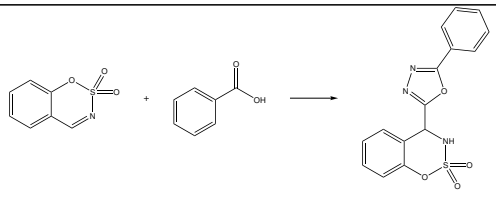
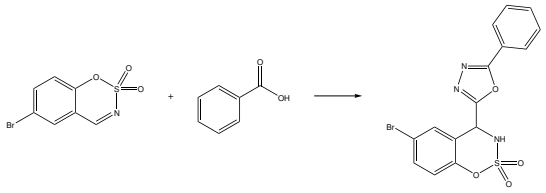
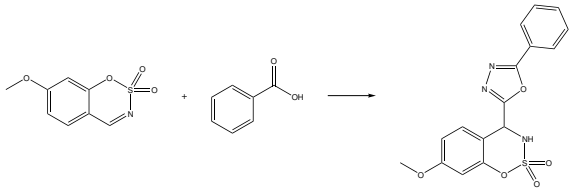
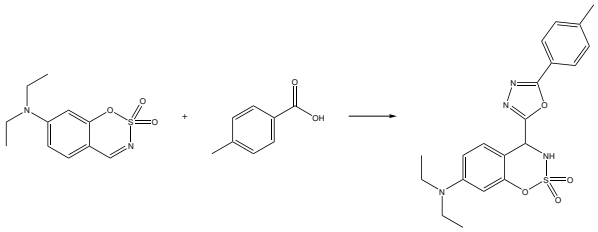
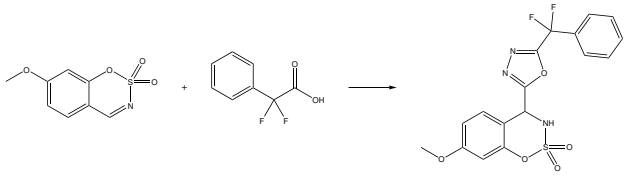
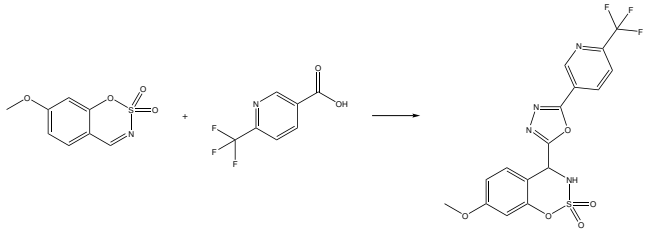


Figure 6.15. Substrate scope for representative acyclic N–sulfonylimine with various carboxylic acids.

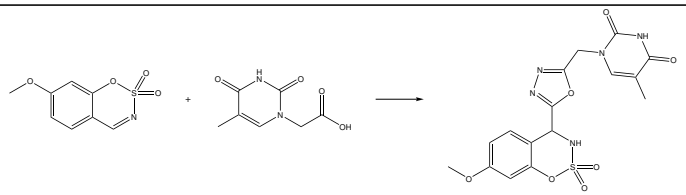
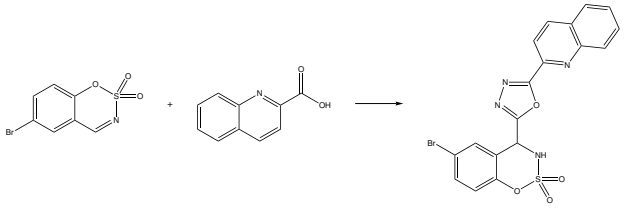
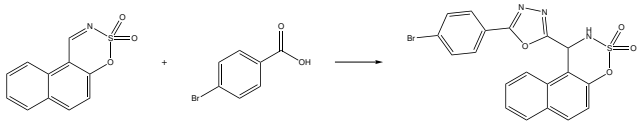
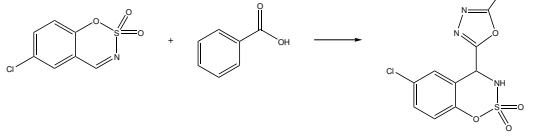
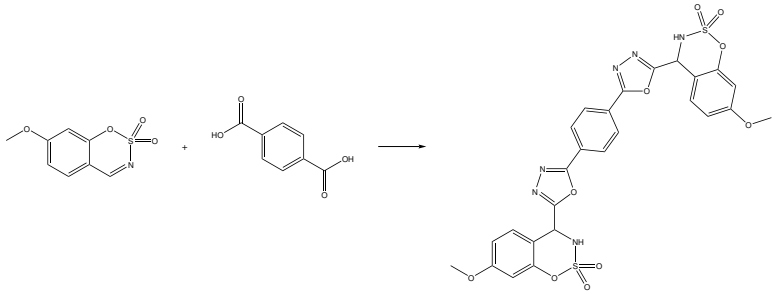
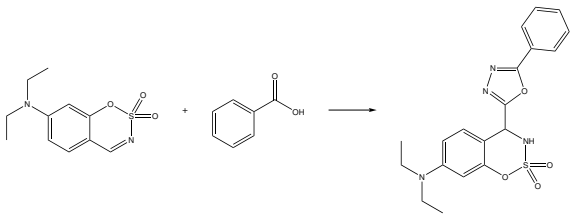
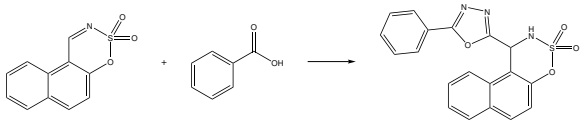
Considering heterogeneous reactivity of cyclic and acyclic sulfonylimines, motivated us to develop a machine learning model using the successful and unsuccessful reactions. We trained decision tree [294] using the Extended Connectivity Fingerprints (ECFPs) [301] of both the carboxylic acid and imine (Fig 6.18A-D) as separate 16384 bit fingerprints with a atom radius of 3 atoms. We used bootstrapping of several decision tree models to ensure the robustness of our model for predicting prospective experimental outcomes.

Table 6.3.: Reactions used to train the machine learning model

Reaction ID	Reaction	Works
kpgc02s195		yes
kpgc02s197		no
kpgc02s198		yes
kpgc02s199		yes
kpgc02s202		no
kpgc02s203		yes

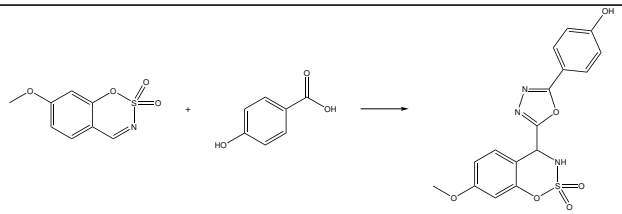
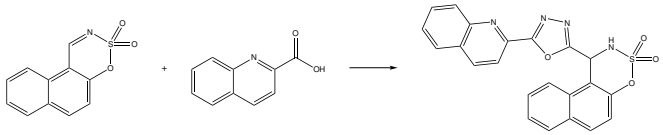
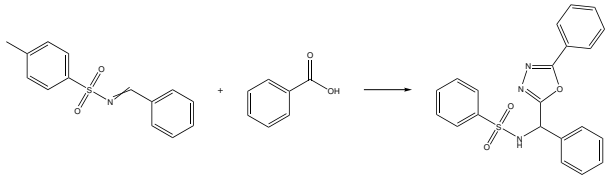
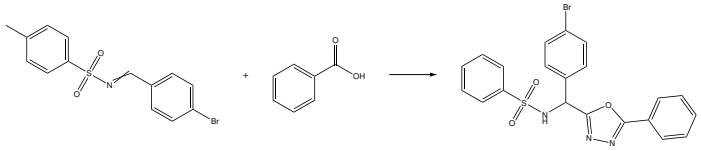
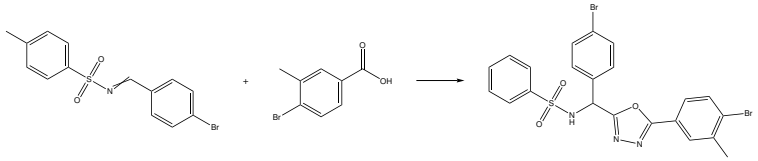
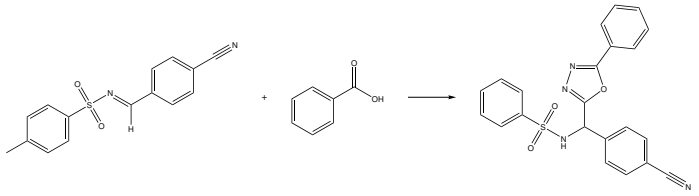
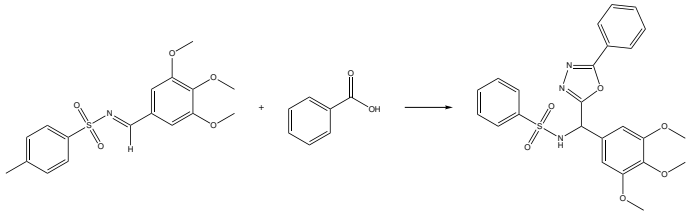
continued on next page

Table 6.3.: *continued*

Reaction ID	Reaction	Works
kpgc02s206		no
kpgc02s212		no
kpgc02s213		no
kpgc02s228		no
kpgc02s229		no
kpgc02s230		yes
kpgc02s232		yes

continued on next page

Table 6.3.: *continued*

Reaction ID	Reaction	Works
kpgc02s201		no
kpgc02s208		no
kpgc02s233		yes
kpgc02s244		yes
kpgc02s246		yes
kpgc02s247		yes
kpgc02s248		yes

Machine Learning models were trained using the individual ECFP fingerprints [301] of the imine and acid with a bit length of 16384 using RDkit. For a fingerprint

radius of 0 through 3, no bit collisions were observed. Each reaction is assigned a reaction ID and a binary condition ('Worked' in the above Table) to represent whether a reaction occurs between the N-sulfonylimine and carboxylic acid. The goal of the decision tree models is to predict the 'Worked' response using the fingerprints. Due to the limited amount of reactions available for training (20 reactions), multiple fingerprint features may represent the same split in the decision. To address this issue, the validation of the decision trees was performed 1000 times to sample the different possible models that can be created for a given fingerprint radius (shown in Figs 6.16 and 6.17).

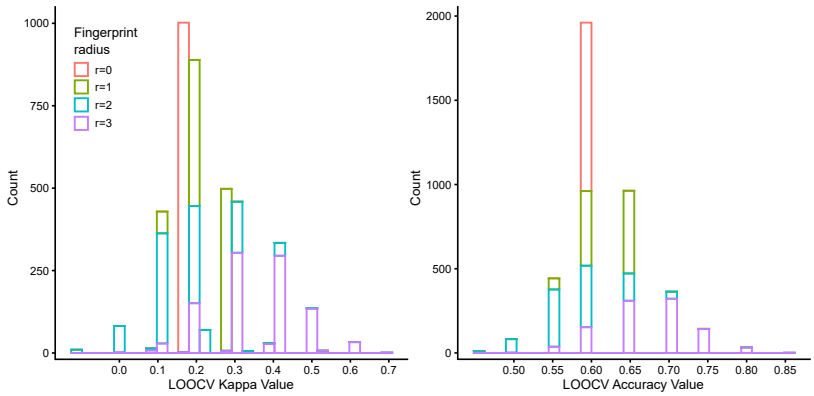


Figure 6.16. The Cohen Kappa (left-hand side) and accuracy (right-side side) value obtained from bootstrapping the decision tree model using different fingerprinting radii. These results show that a fingerprint radius of 3 yields the best decision tree models.

The bootstrapping results show that a fingerprint radius of 3 yields the best decision tree models. The maximum kappa value for fingerprint radii of 0, 1, 2, and 3 are 0.158, 0.286, 0.510, and 0.706 respectively, and the maximum accuracies are 0.60, 0.65, 0.75, and 0.85 respectively (Fig 6.16). However, the models generated with a radius of 3 also have the largest spread of kappa values, indicating that a proper selection of features is required to ensure that the model performs well. To do so, the incorrect predictions obtained from the validation scheme were also examined

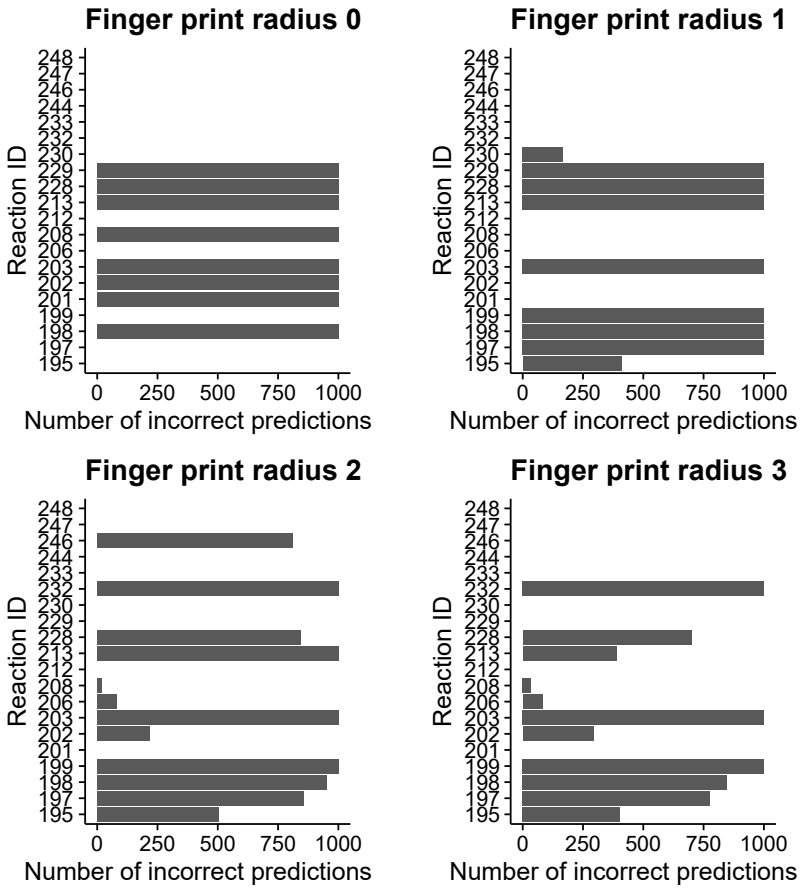


Figure 6.17. The distribution of reactions which are incorrectly predicted during bootstrapping. The y-axis shows the number xyz in the reaction ID KPGC02Sxyz.

(Fig 6.17). The models generated using a radius of 0 made consistent mispredictions for reactions 198, 201, 202, 203, 208, 213, 228, and 229. This indicates that this model does not have a large enough input space to make accurate predictions. A similar argument can be made for a radius of 1 as it consistently mispredicts 197, 198, 199, 213, 228, and 229. The results for radii of 2 and 3 are less consistent but showcase that the selection of features is paramount for obtaining a performant final model. To do so, a model trained on all reactions was created 1000 times and the features which appeared multiple times were measured. The depth of the feature is

used to assign importance to each feature in the tree and these importance values are summed for all 1000 trees. This methodology yields Fig 6.18 and can be used to identify additional reactions that need to be investigated to ensure the model is predictive while maintaining interpretability. These reactions are shown in Scheme 6.19 are used to elucidate that final decision of the model as this rule is supported by the least important features available to the model and the fact that two different features could be used to differentiate reactions.

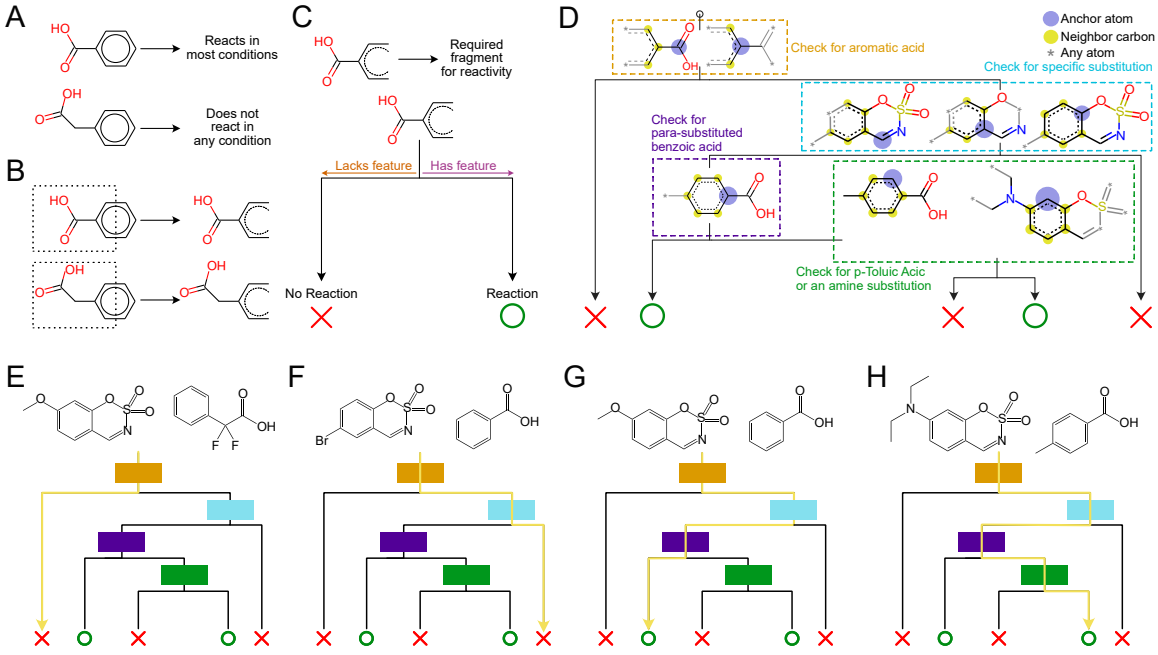


Figure 6.18. Chemical reactivity flowchart. Decision tree based chemical model for the substrate scope of the reaction between the imine and acid. A-C. Showing a pictorial explanation of how the model assigns rules for predicting reactivity. D. Showing the final bootstrapped model trained on all data with details for each rule shown in colored boxes. E-H. Examples of each of these rules using the training data. Box colors represent features shown in D and yellow line of the flowchart shows the outcome of the reaction based on chemical features.

We have high confidence in this model as all, but one of the final decisions are supported by multiple reactions (Fig 6.18E-H). The decision made with a single reaction is Fig 6.18H, and we wished to elucidate whether the use of p–toluic acid or the amine substitution is responsible for the positive reaction condition indicated with a green box in Fig 6.18D.

All decisions made by the ML model were highly confident except for the final decision (green box in Figure 6.18). This decision is only supported by a single reaction and that reaction is identified by either para–toluic acid or an amine substitution. Therefore, the model is unable to distinguish between specific features that resulted in a successful reaction. To elucidate the chemistry at this step, we tested the reaction between **1c** (imine without an amine substitution) and **2b** (para–toluic acid) and noted that the reaction occurred. Conversely, the reaction between **1b** (imine with an amine substitution) and **2d** (4–hydroxy benzoic acid) did not occur. These results show that the final decision should check for para–toluic acid and not an amine substation. Finally, we tested **2d** with the acyclic imine **4a** to see if this rule applied to acyclic amines and noted that the reaction does occur. These reactions are shown in Scheme 6.19 and show how our ML strategy can be used to better understand and expand the substrate scope of an MCR. The additional reactions clarify this rule as they show that a para–methyl substitution is responsible for reactivity and that a para hydroxy substitution leads to decreased reactivity for cyclic amines. It also clarifies that acyclic imines are more reactive as they also react with para–hydroxy benzoic acid.

6.4 Conclusion

In summary, we have developed a fast MCR of acyclic or cyclic N–sulfonylimines that was used as a representative reaction type to develop ML models for predicting

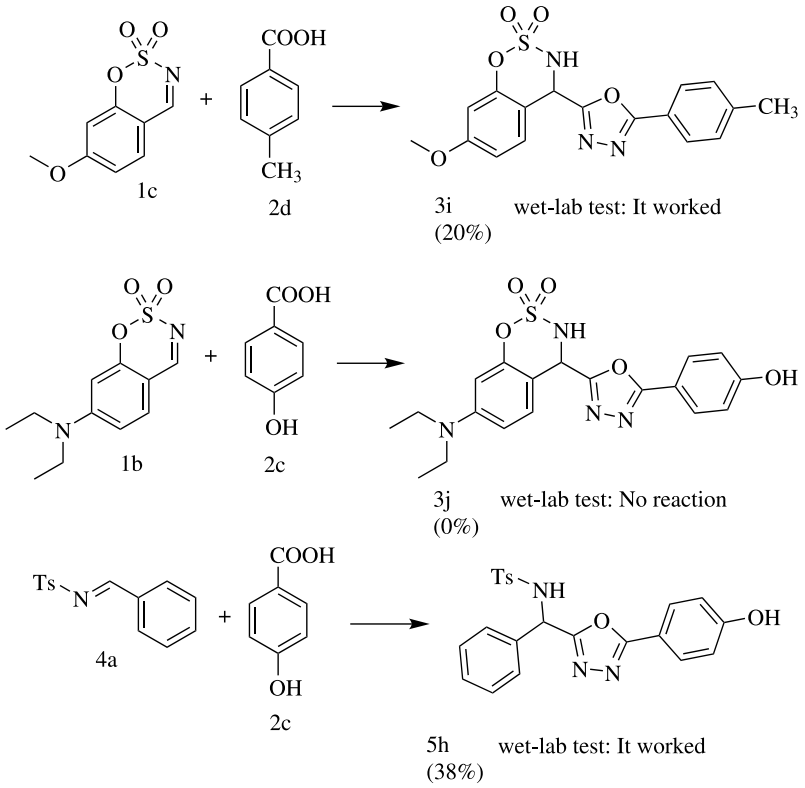


Figure 6.19. Reactions performed to test the ML model.

reaction outcomes in a blind prospective manner. The fast and peculiar reactivity mechanism of N–sulfonylimines was explained using DFT calculation to understand the critical role of transition states and intermediates. Boot–strapped decision tree–based ML models resulted in a chemical reactivity flowchart that explained the choices made by the model to predict reaction outcomes. The human interpretable ML approach can be extended to explore any MCR or any chemical reaction used to synthesize a library of compounds in a quick and efficient manner. This work provides a framework for developing fast MCRs, understanding the underlying reaction mechanism and identifying chemical features for predicting the reactivity of components that results in successful reactions to save valuable time for chemists to not chase dead-end leads.

6.5 Future Work

While the work presented in this chapter details how to validate a machine learning model based limited to the substrate scope of a single reaction, it does not address the development of a machine learning model to find new reaction conditions for an existing reaction. Work performed by Coley et al. has attempted to solve this problem through the use of a multitask neural network [354] and a similar fingerprinting methodology. Similar work has been done to guide solvent choice in organic chemistry reactions (manuscript currently in progress). These predictions can be verified both experimentally and using QM methods. An example of such a validation is given in Fig 6.20. A lot remains to be done in the area of reaction optimization and these are just first steps.

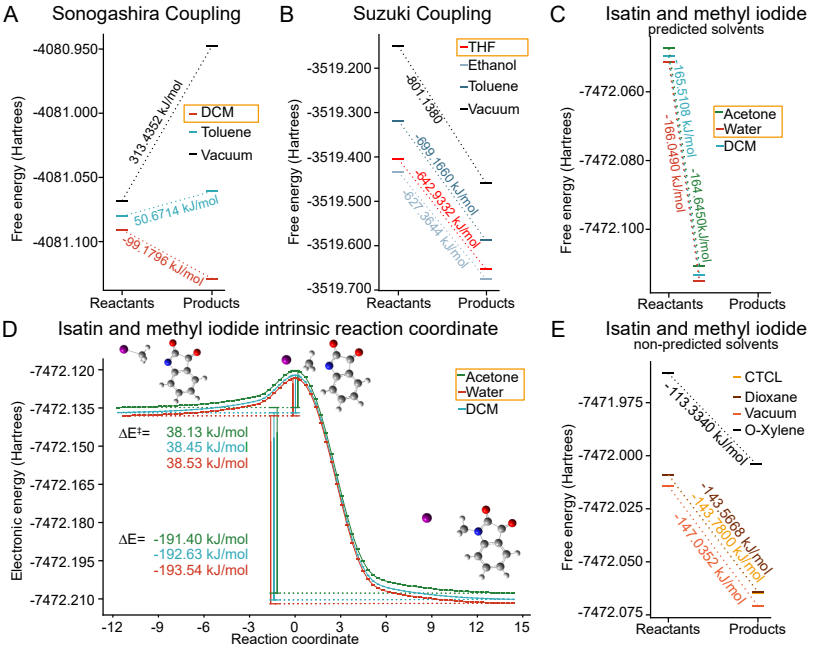


Figure 6.20. Example of QM validation for an ML prediction. The boxed solvents have been predicted by an ML model.

7. VISUALIZING PROTEIN–SMALL MOLECULE INTERACTIONS

This chapter is available as

Zhang W., Fine, J., Sculley C., McGraw J., Chopra G.Molecular Interactions Using New Technology: A Virtual Reality Gaming Platform to Visualize and Manipulate Molecules. *ChemRxiv* (2019). https://doi.org/10.26434/chemrxiv.9889994.v1

7.1 Abstract

The representation of complex biomolecular structures and interactions is a difficult challenge across life sciences. Researchers and students use unintuitive 2D representations to gain an intuitive understanding of 3D space and molecular interactions. Since this is cumbersome for complex structures, such as protein-ligand interactions, several solutions have been proposed to help elucidate the 3D space. However, these representations are often static or do not fully leverage the interactivity that modern computing systems can provide. Our solution, Molecular Interaction using New Technology (MINT), is the first *gaming* platform to effectively represent and manipulate structures in 3D space using virtual reality while simultaneously scoring biomolecular interactions in real-time. Utilizing this combination of manipulation and real-time feedback, MINT provides scientists with an intuitive and effective method for drug discovery. We hope the combination of an intuitive interface with a powerful chem-

istry backend will expand molecular understanding and drug discovery for scientists and non-scientists.

7.2 Introduction

In 1966, Cyrus Levinthal published *Molecular Model-building by Computer* along with the first interactive display for visualization and manipulation of molecular structures [355] that revolutionized the field of molecular visualization. Before the introduction of more advanced virtual systems, physical ball and stick models were developed and used by several scientists, such as Watson and Crick to investigate the structure of DNA and John Kendrew to solve the first crystal structure of protein [356]. As the capabilities for creating virtual 3D environments advanced, the use of physical models decreased, and the physical ball and stick models have been replaced by a mouse, keyboard, and computer monitor. The current way of chemical structure modification along with 3D position and orientation involves the scripting of computer programs or the use of complex graphical user interfaces where inputs are given by a mouse and keyboard [357–362]. Manipulation of molecules through such methods is complicated as it often requires extensive knowledge of programming or a deep understanding of the user interface. Such systems lag behind the state–of–the–art tools developed for human-computer interaction. Relatively simple actions such as positioning molecules are only mastered after a steep learning curve as the many nuances required are difficult to understand for both professional researchers and students [363].

Although the current software packages afford a great deal of flexibility in representation and visualization styles, they lack intuitive manipulations because of their reliance on a mouse and keyboard. The proper representation of biomolecules in a 3D space is crucial to the understanding of various intermolecular interactions [364].

However, 2D displays can misrepresent the understanding of such interactions in 3D and the steep learning curve to manipulate structures is a bottleneck for widespread use of tools beyond scientists [365]. Conversely, physical models avoid these pitfalls by offering an environment where users can manipulate objects intuitively [366] whereby anyone can manipulate physical model by bending angles, breaking bonds, adding new atoms and functional groups, and changing positions of multiple atoms by rotation around one or more bonds. However, the physical models for larger complexes are expensive to make, hard to maintain, and lack real-time feedback to understand molecular interactions. We believe that the intuitive nature of physical models needs to be incorporated into *in silico* 3D modeling software; a feature that can be accomplished with the application of Virtual Reality (VR) hardware.

Currently, there are numerous platforms offering visualization and manipulation of molecular structures [360–362], and several more that have the capability to visualize molecules within a VR environment [367–370]. Noteworthy examples include Molecule Viewer [371] which allows for protein visualization and UnityMol [372] which provides an immersive environment for exploring molecules. Neither platform allows the user to edit and manipulate the chemical environment, a shortcoming addressed by Nanome, a collaborative VR environment implemented with a wide variety of molecular manipulation functionalities, and ChimeraX VR [373], an application utilizing UCSF Chimera and the SteamVR toolkit to enable molecular data analysis and manipulation through simple controller–based input commonly seen in VR applications. While these new tools offer visualization and manipulation capabilities, they are unable to provide insight into the underlying chemical significance of these interactions. Simply porting this functionality from conventional 2D molecular visualization systems, such as PyMol or Jmol, into VR does not exploit the full potential of the new technology for learning and adoption by scientists and layman alike. Ad-

ditionally, these tools lack the collaborative nature and scalability to be effectively applied in the classroom. The motivation for utilizing virtual reality for molecule visualization over conventional 2D displays lies in the inherent intuitiveness and 3D nature of virtual reality, which in turn promotes interaction with the molecule. We believe that interactions between the user and the molecular structure via a feedback-driven system is a key aspect of molecular visualization because the synergy of these features empowers a viewer's natural curiosity to further explore, study, and research chemistry and biochemistry in a unique and rewarding manner.

To address the lack of chemical insight provided by current VR implementations, we have developed the Molecular Interactions using New Technology (MINT), a virtual reality biomolecular visualization platform. Our implementation serves to lift biomolecular visualization to the forefront of the technological frontier and foster a mainstream understanding of the biomolecular research that accompanies drug discovery. We seek to provide an easy-to-use, intuitive, and powerful platform to simultaneously visualize and manipulate molecular structures, allowing any user, regardless of scientific training, to optimize molecular structures and receive real-time visual feedback through MINT's comprehensive virtual toolkit. MINT introduces additional new features not present in the visualization platforms mentioned previously:

1. it is integrated with a backend computational chemistry platform our lab to efficiently compute scoring functions to monitor how manipulation behaviors change the chemical environment

2. it *gamifies* the process of molecular optimization, fostering a playful relationship between user and molecule as well as competition between users for the creation of optimal structures

3. it is scalable across multiple devices from smartphones to workstations. Herein, we present MINT's features that allow for intuitive manipulation and visualization of molecules, followed by a discussion on how these features lead to gamification and the creation of a platform intended for the instruction of chemistry.

7.3 Results and Discussion

7.3.1 MINT provides an intuitive interface to chemistry

MINT utilizes intuitive controls and an immersive environment to allow for a unique visualization and manipulation environment. MINT's workflow is a 4-step procedure consisting of input, visualization, manipulation and output (Fig 7.1). MINT starts by interpreting binary files that contain molecule structure information (Fig 7.1A). With the molecular structures obtained from the input files, MINT generates and displays a 3D model and presents this model through a VR headset such as the HTC Vive. The 3D models are fully interactable, allowing users to reposition and manipulate the entire molecule by working with a menu interface consisting of three different panels: Manipulation, Visualization, and Utilities. This interface groups the essential elements found on many conventional molecule editors into one simplistic and VR centric format (Fig 7.2A).

The Chemical Algorithms for Network–based Decisions on Interactions for modeling reactivitY (CANDIY) software suite is integrated into MINT to bridge the manipulations performed by the user and the underlying chemistry of the VR representation. CANDIY aids the scientific community in their efforts to model how molecules interact with their environment by providing a platform for the development of algorithms and procedures tailored for specific purposes. The role of MINT in this software suite is to allow for user manipulation in an intuitive manner, opening

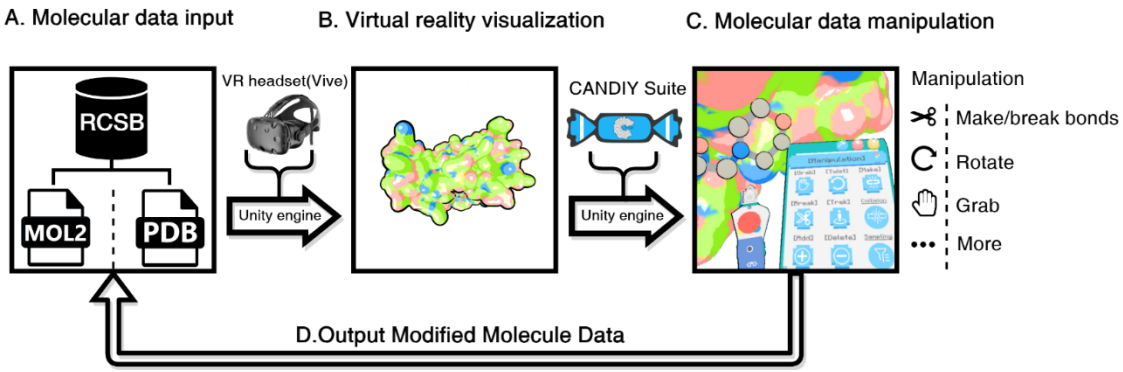


Figure 7.1. Overview of MINT's workflow cycle. PDB and Mol2 files, containing molecule data, are interpreted in MINT and transformed into visualization in a virtual reality environment. User can manipulate molecule structures using MINT's manipulation interface and output new molecule data files.

the use of this software to the layman. After each manipulation performed by the user, CANDIY validates the user input and provides feedback through a combination of haptic and visual interfaces. This creates a relationship which ensures the chemical legitimacy of each operation without the need to instruct the user in advanced scientific concepts. Currently, the CANDOCK [143] package is the most integrated into MINT, but we plan on integrating other packages and machine learning methods that we are developing, such as our biomolecular structure searching software, Lemon, in the near future [215].

CANDIY provides the ability to interpret 3D coordinates and molecular topology obtained from molecular file formats [102] (Fig 7.1A). The molecular information is passed onto MINT for visualization, where it is processed and rendered. CANDIY calculates the interaction between biomolecules, such as a ligand and a protein, in real-time by using a generalized statistical potential function [28]. When a user manipulates the ligand or protein in the VR environment, the changes in the 3D conformation of the molecule and protein are communicated to CANDIY which in

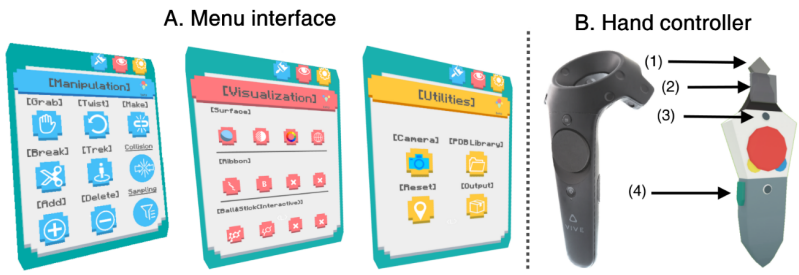


Figure 7.2. (A) An overview of MINT's menu interface (pre-release) consisting of three different panels: Manipulation panel for changing interaction types between user and molecule, Visualization panel for changing visualization types and Utilities panel for functionalities like inputting/outputting molecular data. (B) A side by side comparison between the physical product model of HTC Vive's hand controller (Left) and the virtual model of MINT's hand controller (right) in VR. MINT's controller is a custom-made virtual representation of HTC Vive's handheld controller that is meant for replacing hand presence in the virtual environment. This virtual controller copies the button layout of Vive's physical model and defines these components as: (1) The pointer tip part of the controller. The user uses this tip to touch and interact with the visualization and user interface. (2) A small display panel to indicate the manipulation type that is currently being used. (3) A button to open and close the menu interface. (4) A button on the side of each controller to help the user navigate in the virtual environment through transforming camera position and scaling viewport size.

turn returns a numeric score to the user. This process is key to the gamification concepts presented later in this work.

7.3.2 MINT provides multiple visualization and manipulation modes

The visualization of molecular structures in 3D is a necessary component of the MINT workflow. MINT's VR interface embraces a range of visualization techniques to improve the understanding of a 3D environment, e.g., the binding of a drug to a protein, in a versatile and robust way (Fig 7.3). Protein structures can be rendered

via a surface model with dynamic lighting and shadow effects (Fig 7.3A) where different atom types are represented by different colors on the surface. In MINT, this is the default rendering mode for large molecules due to its intuitive demonstration of a molecule's size and spatial information. Fig 7.3B, 7.3C and 7.3D show a ligand structure rendered in various forms and forge the basis of most molecular manipulations performed by the user. Fig 7.3E and 7.3F depict protein structure in ribbon form and a specialized rendering of the backbone, respectively. Both visualization options allow the user to develop a more holistic comprehension of the biomolecular structure. The user can dynamically tailor the virtualization using visualization panel, giving them the freedom to mix and match different options to create unique and complex visualizations.

Fig 7.4 shows a comparison between a visualization provided by PyMol, a standard molecule visualization program, and a visualization provided by MINT. While both programs offer specular textured surfaces and ball and stick representations, MINT does not require complex scripting like PyMol to represent the binding site tunnel. Instead, MINT helps the user achieve these actions via the VR interface. MINT allows its users to perform several different manipulations using the toolkit depicted in Fig 7.5. By linking these simple manipulations together, users can quickly perform complex maneuvers in a short time as compared to traditional methods of interaction, such as scripting or 2D graphical user interfaces.

- Hand tool (Fig 7.5A), maneuver the ligand as if it was a rigid object.
- Bond rotate tool (Fig 7.5B), rotate a portion of the molecule via an axis of rotation. To define this axis, the user selects two atoms to create a vector pointing from the first atom to the second). This movement is directly inspired by physical models which allowed for different configurations to be created by quick twists and turns.

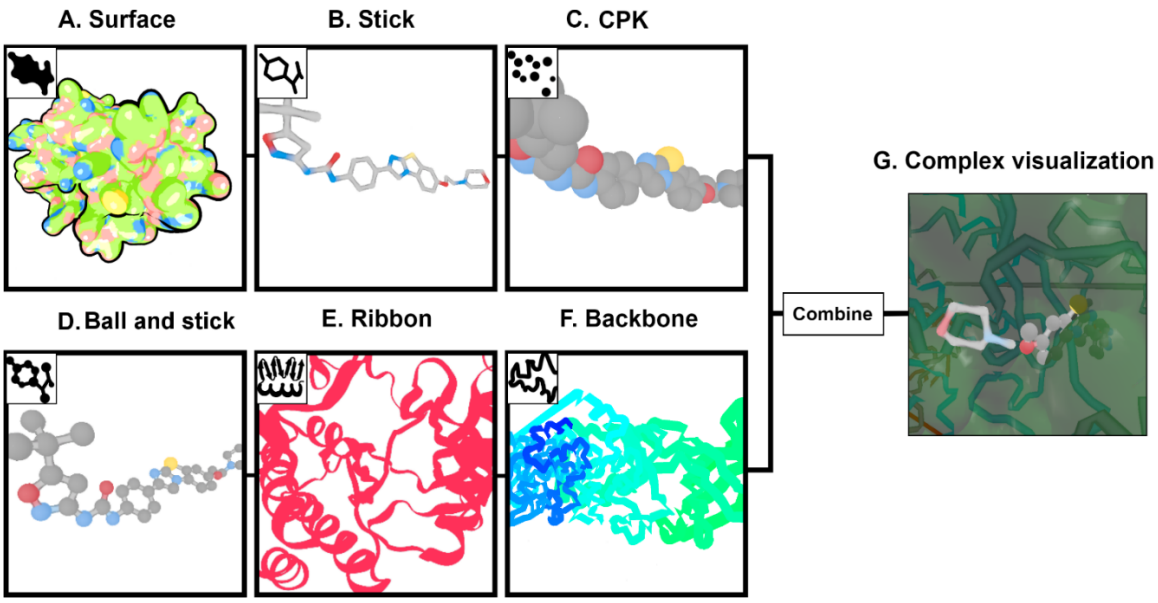


Figure 7.3. Molecule visualization options using MINT and the combination of these options to make complex and interactive rendering of 3D molecule models. (A) Surface model of a molecule structure; (B), (C) and (D) Molecule structures rendered as the stick, CPK, and ball-and-stick models; (E) and (F) Protein structure rendered in ribbon diagram and its backbone representation. (G) A combination of the options above, in which the surface model is rendered in transparency.

- Bond tool (Fig 7.5C), make and break bonds by clicking on two different atoms simultaneously. This action is monitored by the backend program CANDIY that prevents the creation of invalid molecules.

- Selection tool (Fig 7.5D), select specific atoms to manipulate instead of working with the whole entity.

- Surface trekking (Fig 7.5E), a quick way to navigate in the environment by walking on a surface model.

- By linking these simple manipulations together, users can quickly perform complex maneuvers in an exponentially shorter time than traditional methods of interaction like scripting or 2D graphical user interfaces.

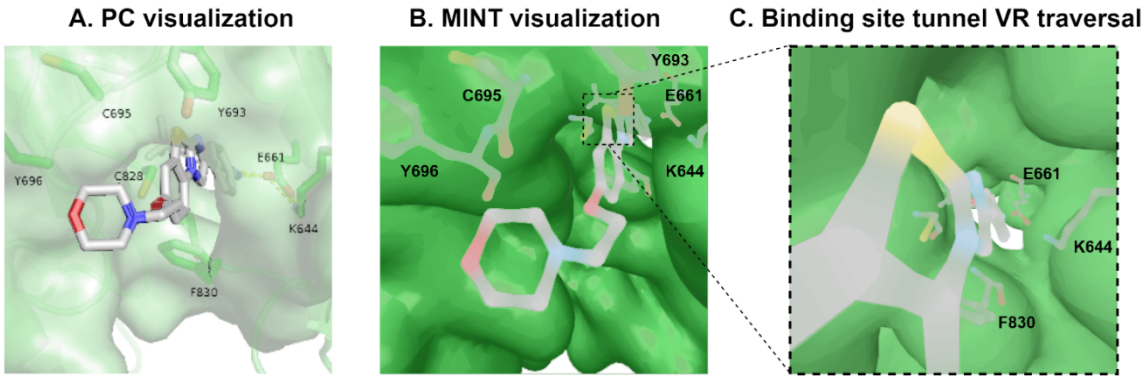


Figure 7.4. Side by side visualization comparison between (A) PyMol and (B) MINT. (C) Zoom-in view of the binding site, showing MINT's ability to perform binding site tunnel traversal.

To illustrate the MINT workflow, we have detailed out each step of the workflow for PDBID 4XUF in Fig 7.6. MINT begins by interpreting the PDB input file (Fig 7.7A) and converts the textual atomic records into atom data arrays that form a complete representation of the molecule in working memory (Fig 7.6B). MINT produces an intuitive VR visualization using these coordinates which the user can interact with to optimize the docking score of the ligand (Fig 7.6C). Once a user has performed a manipulation on the ligand such as rotation or translation, the change is reflected in working memory (Fig 7.6B). Finally, MINT outputs the modified data as a PDB file that can be used in other applications or reopened in MINT for further analysis (Fig 7.6E).

The driving force of such workflow is MINT's ability to consolidate every manipulation made by the user into a numeric 'score' which represents the chemical validity of these actions. A detailed description of this score is given in the section entitled

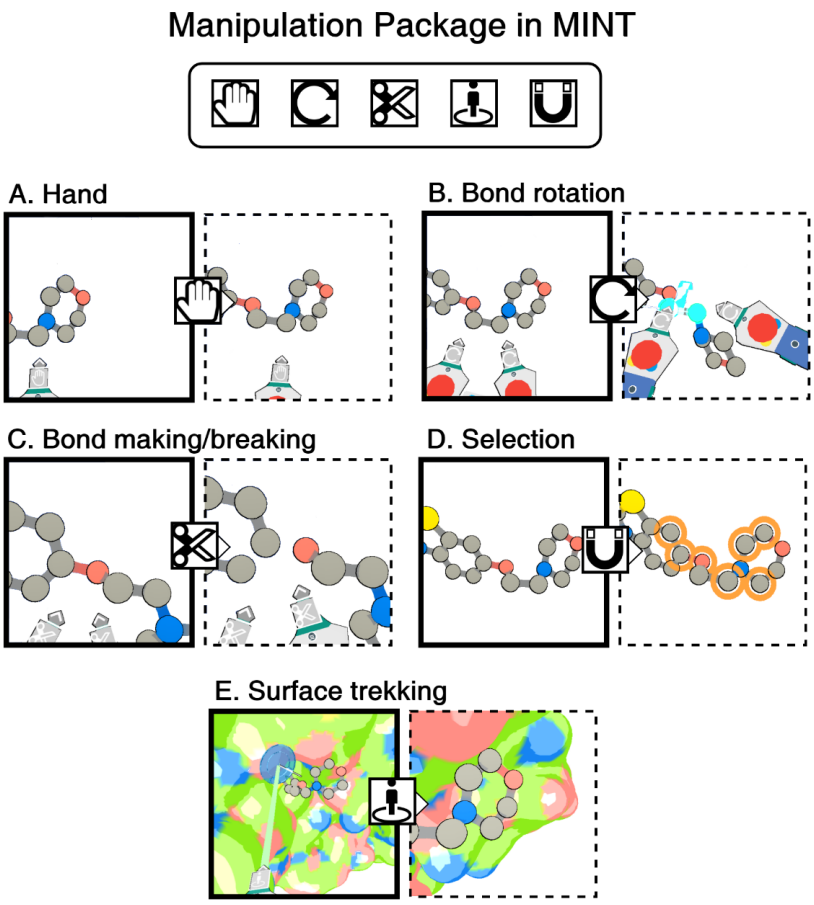


Figure 7.5. (A-E) Five basic types of molecule manipulation using MINT interface. For example, for Hand tool (A), snapshot on the left of (A) shows the state of molecule structure before Hand tool manipulation is operated, and snapshot on the right of (A) shows the state after Hand tool manipulation is operated. The hand tool is used for moving molecular clusters in the VR environment.

Scoring of Player's ligand conformations. In Fig 7.7, PDB 4XUF, a protein-ligand complex, has a score of 331 in its initial state (Fig 7.7A). The user then performs a bond rotation on the ligand structure through the VR controllers and interface (Fig 7.7B). This action results in the score increasing to 333 in real-time, and indication that the user has improved the potential effectiveness of the ligand towards the target protein.

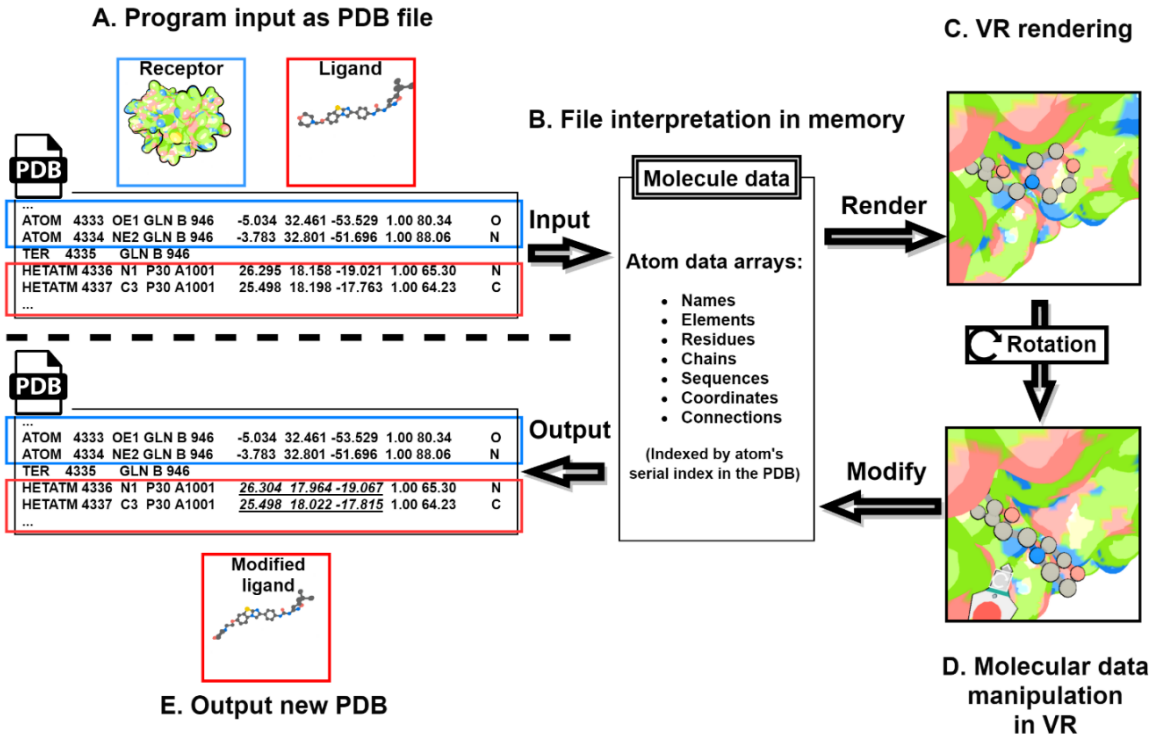


Figure 7.6. A detailed look at the input and output processing pipeline of MINT. (A) MINT interprets the PDB file's textual atomic records line by line and (B) transfers the information into data arrays in Molecule data classes) which Unity Engine can understand and further passes down to Unity's rendering pipeline. (C) MINT renders receptor atoms in surface form and ligand atoms in colored ball and stick form. (D) A rotation operation is performed on the ligand atoms, altering its angular conformation This action modifies the atom data in the memory. (E) All of the atom data arrays are written out as a new PDB file with the modified atomic records reflecting the rotation operation that is performed in (D).

7.3.3 Gamification of molecular interactions is an integral component in MINT

The influence of video games on contemporary culture is immeasurable and the practice of utilizing factors that involve game mechanics like challenges, tasks, and levels into the design of non-game consumer software has surged in recent years [374,375].

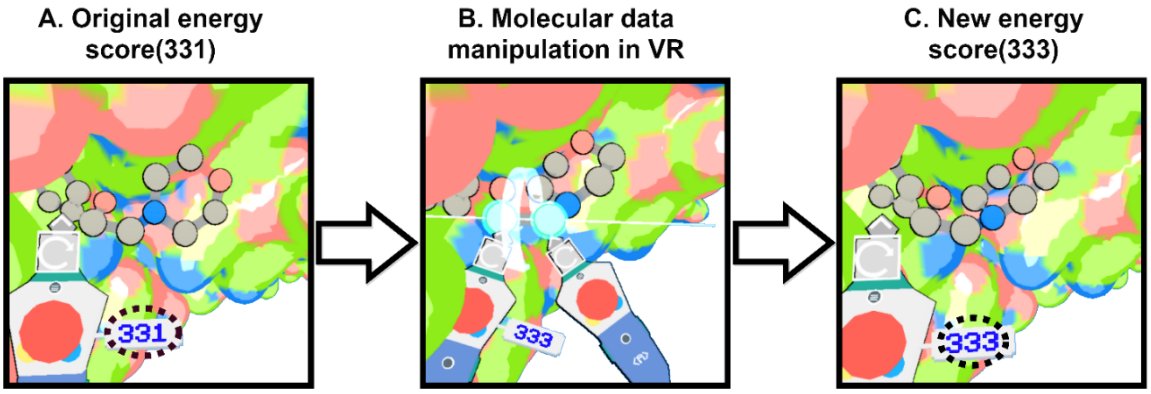


Figure 7.7. Demonstration of the score feedback feature in MINT. (A) shows the visualization of the structure 4XUF that contains both a receptor protein and a ligand molecule. 331 is the original score this structure possesses, which relatively indicates its energy level between the receptor target and the ligand. After going through the manipulation in (B), the score updates to 333. These two scores are calculated through CANDIY's scoring functions in real-time.

The notion of a *serious game* [376], for example, is a practice parallel to gamification and is often categorized by its emphasis on training the player for a specific real-world task or completion of non-entertainment objectives through specially oriented gameplay. The incentive of incorporating *gamification* into non-game software amplifies the user's engagement with the experience and stimulates motivation and curiosity to further facilitate accomplishing an objective regardless of whether it is learning, training or simulation. The benefits of gamification have been explored in many studies [377–380].

One excellent example that combines biochemistry, protein folding, and gamification is Foldit [381], a platform which presents a multiplayer puzzle game to help solve protein folding questions. This 'game' takes each protein structure as a challenge or a level for the player to conquer by using the intuitive folding tools provided by the application and leverages the crowdsourcing nature of gameplay to unite all players

and further facilitate biochemistry experimentation and research. The Foldit player base has achieved remarkable accomplishments including helping decipher the crystal structure of a monomeric retroviral protease linked to HIV/AIDS [382]. In a similar manner, we aim to incorporate gamification elements like player collaboration/competition, challenges, scores, and a playful user interface into the design of MINT and to excel at being intuitive and engaging with the help of VR.

The scoring feature in MINT (Fig 7.7) functions as an indicator of the interaction energy between two structures such as a protein receptor and a small molecule ligand, or as a method for players to self-validate their in-game actions. Given the importance of score for the *gamification* of a given objective, we present this interaction score as the core mechanism for gamification in MINT. A receptor and ligand complex obtained from the protein databank (PDB) [23] is imported into MINT to produce a level or a quest, where players can compete against each other to find the optimal score (provided by CANDIY). To do so, the players must manipulate the conformation and topology of the ligand, yielding a drug discovery platform which is naturally crowdsourced. One can further extend this pipeline to rank scores obtained from different players on the same complex on a leaderboard in order to encourage competition between players. Such practice can be achieved through a backend server that collects players' gameplay data, providing an implementation for crowdsourced drug design.

A series of playful aesthetics are utilized by the MINT user interface to instill a game–like theme throughout the gameplay experience. For example, the coloration of each element in the program, including the menu interface (Fig 7.2A), the controllers (Fig 7.2B) and the 3D models rendered in MINT (Fig 7.3), tend to fall on the brighter sides of the color spectrum, and are selected to have a high contrast with one another. The menu interface takes the form of a virtual clipboard that the user can hold using

the controller and the textual elements such as tooltips on the interface are pixelated and 2D image icons are used for representing each functional item on the menu. Furthermore, the toon shading technique (cel shading) developed by Luque is used in the program for rendering a surface model of macromolecules to give them an outline on the edges and produce simple lighting visual effects, yielding an environment which imitates a comic book drawing.

Haptic feedback [383] is the sensorial mechanism used to simulate a sense of touch and is used to convey the application of motion or forces, the difference between the weight of virtual objects, or the textural feeling of geometry or surface. HTC Vive's hand controller (Fig 7.2B) has a built–in haptic feedback mechanism which vibrates to simulate a sense of weight and friction. MINT exploits this feature to make the overall user interface responsive and lively. The variation of vibration depends on both its duration and its strength and adjusting these two factors opens different dialogs with user: For example, clicking a button on the menu interface generates short and mild vibration that imitates the sense of pressing a mechanical button, while clashing a protein and a ligand by dragging them together produces long and strong vibration to indicates the physical collision of such a clash. Similarly, rotating angular bonds between atoms returns a consecutive and blunt vibration in short intervals on to resembles the sense of turning a crank.

Another important feature for immersive gameplay is the use of gesture–based and motion–based interaction, therefore a major component of MINT's manipulation system is performed using intuitive gestures and motions. For example, breaking a bond is performed by pulling two atoms apart from each other instead of having users simply click on two connected atoms with a computer mouse. This motion can be augmented with the gradation of vibration on Vive's hand controllers to express the energy cost associated with the operation. We have implemented these features in

MINT to add complexity and a sense of skill to gameplay, ultimately contributing to the *gamification* of molecular manipulation.

7.3.4 Scalability for collaboration and education

While gamification is a defining difference between MINT and conventional molecular visualization and manipulation software, it is only one of the several factors that improve the scalability of MINT as a collaborative project over competitors. Education, research, and entertainment are the three pillars guiding the developmental roadmap of MINT and applying the molecular visualization and manipulation capability of MINT in education and research spans the gambit from classroom teaching to drug design prototyping. Since the number of active VR users worldwide is increasingly rapidly [384], a large potential user base is anticipated to become participants in this project, benefiting drug design and discovery. Therefore, we want to catalyze the popularity of MINT by introducing mobile versions and multiplayer gameplay features. Due to our use of the Steam VR toolkit and the Unity3D engine, our visualization platform can run on multiple hardware platforms. Although we targeted the HTC Vive due to its superior support for human-computer interaction, the Oculus Rift is also a potential target for our platform as others have attempted to use this platform to target drug design [385]. However, we believe that our program is both more intuitive and scalable than these approaches due to the better human-computer interface offered by the Vive.

Molecular data representation in memory space is a crucial component of the processing workflow for scalability to create vivid and intuitive graphics in VR. To manage different sections of the workflow, MINT has a hierarchy of data classes and helper classes that are dedicated to representing and managing molecular data receiving from CANDIY. All the entries in the PDB file that describe atoms are used

to generate atom data arrays, which are stored within the molecule data class. From a single molecule data object, various in-game representations can be generated (Fig 7.8). First, MINT's algorithm generates a molecule representation base from the atom data arrays, which connects the molecular data with its in-game representation, because a single molecule often possesses many different types of molecule representation forms. Next, MINT uses the marching cube algorithm [386] to generate a mesh that simulates the protein's surface and uses native features in the Unity game engine such as game object instantiation and line renderers to simulate atoms and bonds for ball-and-stick representation. The molecule representation base forms the basis for atom manipulation and interaction with the user by enabling collision detection with the user's VR controller. Physical collision provides the player with useful feedback about the position and orientation of the molecule. The changes that the user makes on the molecular structure, such as transformation, making/breaking bonds and angular bond rotation, update the molecule and atom data which then go through CANDOCK for error checking, automatic optimization, and most importantly, validation of these VR operations to ensure the scientific accuracy. Finally, CANDIY updates the molecule data, which is then returned to the user through visualization via the molecule representation in Unity.

We have released a version of MINT on the Google Play Store that targets the Android Platform (MINT Mobile). This version is compatible with Google Cardboard, a low-cost head-mounted VR platform developed by Google for smartphones. Currently, the mobile version only supports molecular visualization and is equipped with a user interface that is tailored towards smartphones, taking into consideration of smartphone's limited computational power and the lack of physical controllers when compared to PC. In this version, the user can load molecule structures as different visualization on the fly and study the visualization using tools like surface trekking

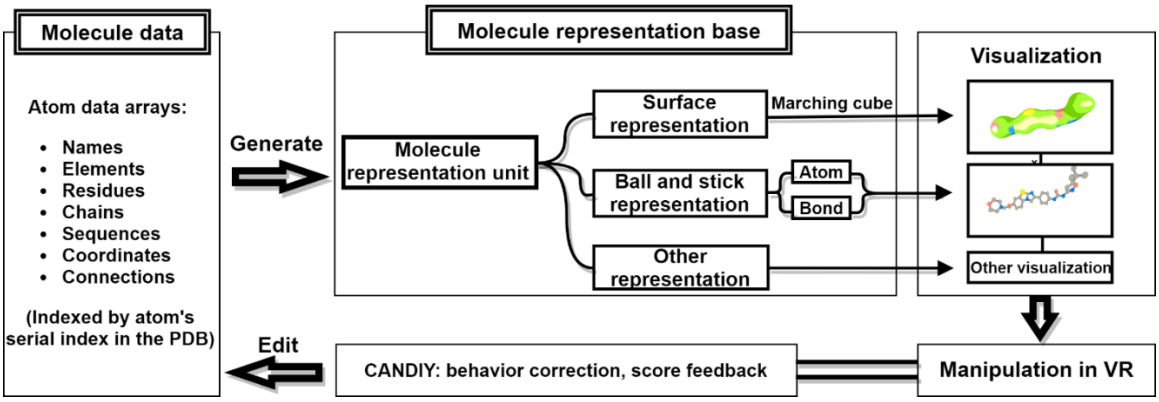


Figure 7.8. A more in-depth look at the B, C and D sections from Fig 7.6. Molecule data class, containing a list of atom data arrays, generates Molecule representation base class, in which visualization representation of molecules are diverged into different forms. The manipulation input on these 3D visualizations from the user is sent to CANDIY to be furthered processed. Finally, CANDIY returns the modification upon Molecule data class.

and camera orbiting. Additionally, MINT mobile includes the environment grid guide (Fig 7.9A), which is intended to help offer the reference of camera orientation and position in the virtual environment.

To enable collaboration of molecular exploration at real-time, we have developed a multiplayer version of MINT which allows for multiple users to cooperate in the same virtual space (Fig 7.9B). In the multiplayer mode, one user hosts a virtual environment using the HTC Vive headset and controllers, allowing them to manipulate and modify molecular structures. Other users can enter the hosted environment as guests through the use of the mobile version of MINT and spectate the host user's actions in real-time. Guest users can walk around in the virtual environment, observe the structures from different perspectives, and suggest manipulations to the host.

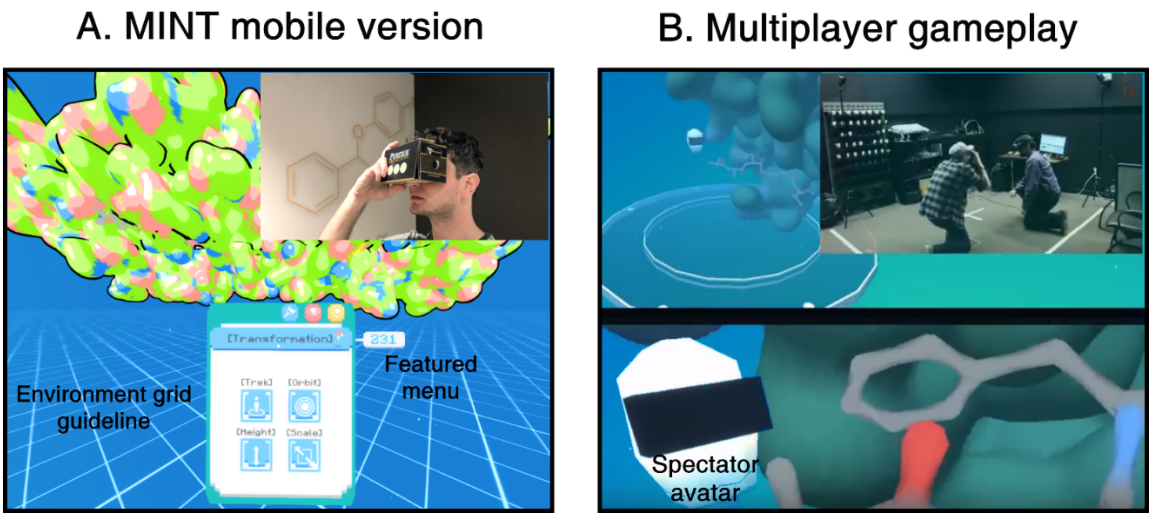


Figure 7.9. (A) shows a mobile version of MINT that runs on the Android platform using Google Cardboard. (B) shows the multiplayer gameplay in MINT, in which one user is the operator of molecular manipulation and the others as spectators in VR.

7.3.5 Conclusion

MINT is a VR platform that challenges the conventional molecular visualization and manipulation tools used in a 3D environment. Equipped with an intuitive interface and a variety of features in visualization, MINT brings ease of use and better comprehension to biochemistry research and study. Due to its use of CANDIY, our program pipeline is user-friendly because it allows for input and output compatibility with conventional chemical file formats and is responsive towards user actions while keeping a chemically accurate simulation. Users do not need to possess specialized programming or scripting knowledge to perform complex manipulations in MINT. Instead, a few quick movements with a VR controller can surpass what many lines of codes can do in other molecular visualization software and in an exponentially shorter time.

The concept of gamification is ingrained into every component of MINT, from the design of interaction between molecule and user, to the aesthetics of interface and the feedback of each action that happened in VR. A fun and enjoyable user experience is born through the utilization of such elements and ultimately yields a more scalable software that can reach a broader audience through the implementation of an intuitive interface for molecular manipulations. These features along with tight integration of our platform with the CANDIY suite for evaluating the molecular interactions of small molecules, provide a unique functionality equivalent to traditional molecular visualization packages like PyMol and Jmol while offering a unique experience that is immersive and interactive due to the power of VR. MINT allows users to develop visual comprehension of molecular structures while making it easier to manipulate the structures in a short period of time. Finally, MINT will be released as an open-source project which welcomes collaborative efforts from all members of the VR and chemistry community. Decades have passed since Levinthal's system was first introduced, revolutionizing the way we perceive the microscopic world and we hope MINT can be part of the next revolution by incorporating modern technology and other advancements previously unavailable.

7.4 Methods

7.4.1 Surface Generation

Surface generation utilizes a modified version of the marching cubes algorithm [386] specifically tailored towards Unity, which results in a continuous, dough-like surface. While this algorithm is especially useful for generating protein surfaces, it can be exceedingly costly as well. Limits upon the number of faces a procedural mesh can contain in Unity requires certain models to be generated and pieced together

in smaller parts. Generation speed and stability of the surface depend heavily upon thresholds set by the player and the size of the datasets. To optimize performance, the existing serial density field code was modified to run in parallel on multiple threads. To accomplish this, a collection of work threads is generated and assigned a group of cells. This code is structured analogous to SIMD systems, as each cell's final density is independent of the surrounding cells.

7.4.2 Molecule Input/output

To generate and further manipulate a 3D molecular model, MINT requires preset data input that defines the molecule's structural formation. MINT's visualization and manipulation pipeline is focused on the atomic records and bonding records that exist in the PDB file. Each atomic record is a line of text that starts with the label "ATOM" or "HETATM", followed by the atom's index number, element type and other information along with its 3D coordinates. Some PDB files contain both receptor atomic records and ligand atomic records. MINT's interpretation of the input PDB file starts by delegating file reading to CANDIY. The various column-aligned parameters are read and interpreted by CANDIY according to standard PDB file protocol, which then transforms textural data into memory space and sends them back to MINT. After receiving the data returned by CANDIY, MINT puts them into data arrays in Molecule Data class which the Unity Engine can understand and further passes them down to the graphics rendering pipeline. The structural changes associated with this operation update the arrays in the memory space. Finally, the atom data arrays existing in the memory space are written out as a new PDB file with the modified atomic records reflecting the various manipulations that the user enacted. The new PDB file generated from MINT can be passed down to other visualization platforms to create display renderings of the new 3D molecule model that reflect the

structural modifications from MINT or enter another round of MINT's input/output cycle to be further studied on in VR. In addition, MINT has a PDB-fetching VR panel that searches and fetches PDBs when users input the name of the PDB to the interface. MINT communicates with RCSB cloud backend to retrieve PDBs in-app so there is no need for the user to take off the VR headset and download resources manually.

7.4.3 VR Interaction

The hand tool is used to grab the ligand. In this mode, the ligand will follow the orientation and position of the player's hand in 3D space when the user activates the trigger button. Haptic feedback on the controllers is provided to give the player a feeling of weight and resistance, as well as a signal when ligand molecules are brushing against the protein's surface model. Players are also given the ability to freely rotate atoms along with their bonds by using the rotate tool, a method of interaction is directly inspired by molecular modeling kits. To perform this action, players must first grab an atom that they wish to act as an "anchor" with their offhand. Next, players use their primary hand to select an atom that determines their axis of rotation. This atom is referred to as the directional atom. An axis of rotation is defined from the vector pointing from the center of the anchor atom to the center of the directional atom. Several steps are required to calculate the angle of rotation. First, a perpendicular plane is formed using the vector between the two key atoms. Next, the position of the player's primary hand is then projected onto the plane. This point in space always lies upon the plane that is perpendicular to the axis of rotation. This new local space can be conceptualized as a 2D plane, where the (x,y) position of the two atoms are located precisely at (0,0). Every frame, an angle in degrees is calculated between the center of this space and the projected

coordinate of the primary controller. The current angle is compared with that of the previous frame and a delta is calculated. The anchor atom is then rotated by this delta angle to produce a rotation that mimics the rotation of the hand around the axis of rotation.Also, bonds between two atoms may be created or destroyed with the 'Make 'n Break' tool. This works very similarly to the rotate tool, by creating or destroying a bond once the user has selected two atoms.

Players are also allowed to scale themselves around the atom they are interacting with, enabling the player to resize themselves in the atomic world. Currently, the method of scaling requires the player to perform a *pinch–zoom* gesture with both controllers to allow for fine control of scale when the player's hands are spaced farther apart. In addition, the world is also simultaneously translated about the vector between the two controllers. This is done to make the scale tool feel much more natural to use. MINT also supports a trekking feature which allows players to navigate the surface of the protein molecule as if they were walking inside a cavern. Trekking is a special form of teleportation that allows the player to move to points of the surface while keeping the relative local space unaltered. When the trekking tool is enabled, the player points their offhand at any point along the surface of the receptor. A transparent disk is shown at the point of contact, along with a perpendicular pole that shows where the player's up vector will point after the move. Upon pressing the designated button on their controller, the player is quickly warped to the new location. Also, the scale of the world is increased dramatically, inspiring the impression that they are standing on a surface of titanic proportions. To reduce simulator sickness, the screen is blurred slightly, and their position is quickly linearly interpolated between their origin and destination. Any additional change in rotational orientation is performed instantly to avoid nausea.

7.4.4 Scoring of Player ligand positions

The CANDIY uses a generalized statistical potential function derived from the Cambridge Structural Database (CSD) [210]. This scoring function is applicable to a variety of chemical environments including small molecules, proteins, RNA complexes, metal ions, cofactors, water molecules, etc. The score of a given molecular pose is calculated as the sum of all pair-wise interactions occurring between the small molecule and the biomolecule of interest within a cutoff of 15 angstroms. The interaction between two atoms with distance r is defined by the ratio of a functional term by a reference term, given below.

$$S\left(r_{ab}^{ij}\right) = -\sum_{ij} ln \frac{g\left(r_{ab}^{ij}\right)}{g\left(r^{ij}\right)}$$

Here, r_{ab}^{ij} is the distance between atom i of type a and atom j of type b. The numerator (functional term) is defined below, which is derived from the radial distribution function:

$$g\left(r_{ab}\right) = \frac{\frac{N_s(r_{ab})}{V_s(r)}}{\sum_r \frac{N_s(r_{ab})}{V_s(r)}}$$

Here, N_s is the number of times an atom of type b is found within a given distance from an atom of type a and V_s is the volume of a sphere with radius r. The denominator (reference term) is defined as follows:

$$g\left(r\right) = \sum_{ab} g\left(r_{ab}\right) = \frac{\sum_{ab} \frac{N_s(r_{ab})}{V_s(r)}}{\sum_r \sum_{ab} \frac{N_s(r_{ab})}{V_s(r)}}$$

7.4.5 Automatic optimization of user-created molecules

One of MINT's important features is the relevance of chemistry presented to players during the VR experience. Since we cannot expect all players to be experts in chemistry, we have included an energy minimization functionality to help players correct potential mistakes such as bad bond lengths, stretches, angles, and contacts. This functionality is optional but highly recommended for ensuring proper chemistry is incorporated into the game. CANDIY provides this functionality via the use of the OpenMM toolkit [209]. Bonded forces are calculated using the Amber Forcefield [387] and non–bonded forces are calculated using the aforementioned scoring function.

7.4.6 Compatibility on mobile devices

Molecular visualization and manipulation can be costly computational wise, especially for proteins, as it oftentimes deals with a large quantity of atomic data. We have found that for mobile platform surface model generation and rendering of a protein takes approximately 5 to 6 times more duration to complete than for the desktop version. Plus, virtual reality display, compared to conventional 2D display, is innately more expensive due to its requirement from hardware to render an image twice, one for the left eye and the other for the right eye. To combat the above limitation and to improve the overall user experience, we have been looking into optimizing the program via multi-threading processing and GPU processing. Also, as MINT is built using Unity3D game engine, Unity's recent engine update in 2018, which improves its support for developers to build a high-performance application, will contribute to the optimization of MINT as well.

7.5 Future work - continued development of the Spear library

Note that the current version of **Spear** is in an alpha stage and therefore many names and features mentioned in this document could change or be reworked. The overall design of **Spear** is not likely to change, however.

Spear is a library for the creation of software packages that require comprehensive graph theory and 3D coordinate support. Similar to RDKit, it is backed by the Boost Graph Library, which is currently the fastest graph available for C++, and the OpenMM software package for molecular dynamics. Using the **Spear** interface, users can perform an operation on the topology of a molecule, use the topological descriptors generated from their method to perform an MD simulation, and pass the trajectory of the simulation to a Neural Network for analysis. Currently, **Spear** provides abstract classes for the creation of custom atomtypes, forcefields, charge schemes, and fingerprinting algorithms which can be extended in C++. Eventually, support for extending these classes in Python is planned. Finally, integration with Psi4 for quantum chemistry calculations and libTorch or TensorFlow for machine learning is also planned.

Using **Spear**, one can create packages to score ligand poses, simulate advanced environments using charges calculated from a QM backend, and fingerprint a molecule. Unlike other toolkits which attempt to implement all features 'in-house', **Spear** relies on other packages for core functionality. All packages used by **Spear** are available under the BSD license, allowing **Spear** to also be published under this license. The only exception is PSI4 (under the LGPL), therefore these integrations are built as an extension to **Spear** instead of being included in the core library. These integrations set **Spear** apart from other packages because they greatly expand the feature set available to **Spear** without the direct need to implement thousands of features into the core library itself.

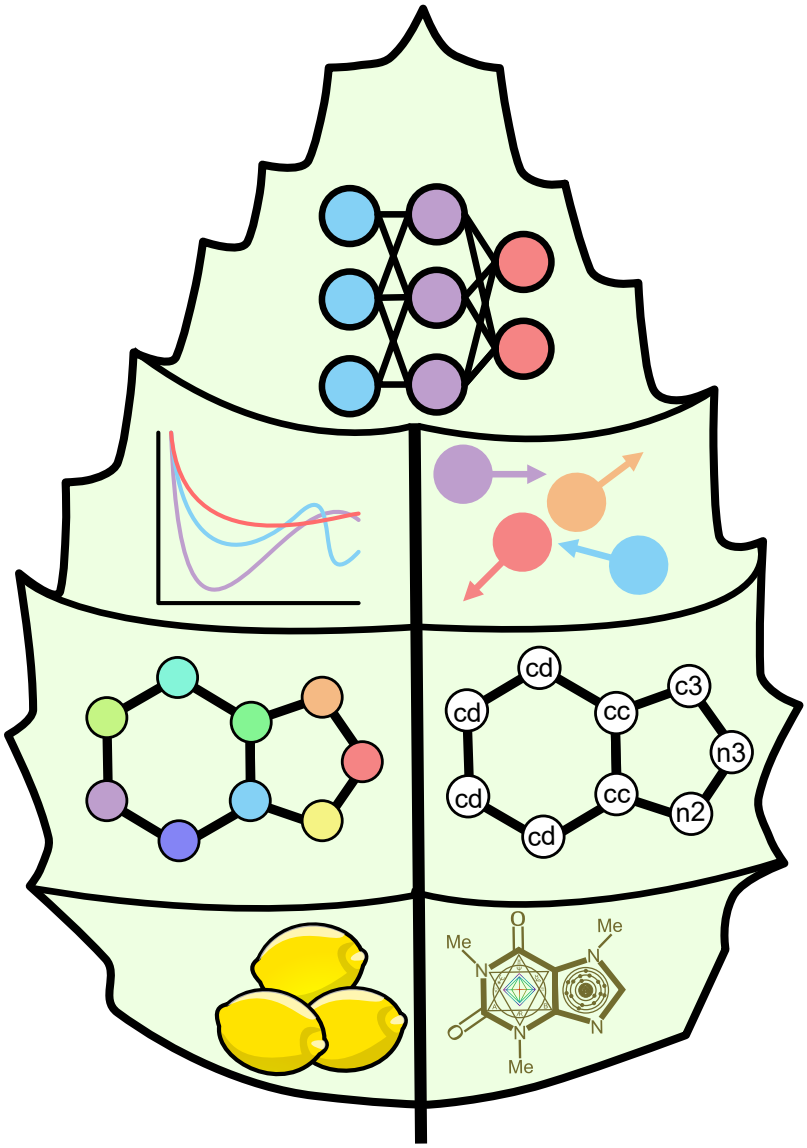


Figure 7.10. Graphical description of Spear

The first rung of **Spear** is dedicated to its ability to integrate with other softwares for processing large amounts of data (for example, an interfacing with Lemon in 2.1) and advanced storage formats such as CIPHER (see future work of Chapter 5. The next rung represents the graph algorithm capabilities of **Spear** and through these features such as atom typing and partial charge assignment. The penultimate rung

represents 3D coordinate abilities such as the implementation of scoring functions and molecular dynamics (through OpenMM). The final rung represents the ability to integrate **Spear** into machine learning libraries.

7.5.1 Reasons to create Spear instead of existing software packages

1. OpenEye: This software package is proprietary and an OSS license is preferable as it allows others to integrate **Spear** into their packages with little issue.

2. Indigo: similar to OpenEye, but under the GPL3 license. Additionally, it has not been developed for a few years and many of its classes are not actually implemented, so they do nothing.

3. OpenBabel: Although this software is both popular and available under an OSS license, the GPL license prevents easy integration of this software into other packages. Additionally, the major focus of this library is on converting chemical file formats - making it an odd choice for developing software packages.

4. RDKit: While this package is under a free license (BSD) and is quite popular in the machine learning community, it is not written in modern C++ and is difficult to contribute to as a result. Additionally, it does not support molecular dynamics outside of UFF and MMFF and does not support file-formats designed for these applications. Therefore, RDKit is not used for many applications outside of fingerprinting methods. However, its graph–based capabilities are numerous and powerful and the BSD license allows one to take portions of their code and rework them into a new and better library that does not have over a decade's worth of crust. Several different packages and features have been added to RDKit over the years, but these features are not well supported (IE you can build a neural network directly in RDKit, but not a well supported

one). In contrast, **Spear** is designed to work with other libraries and not have all solutions built into the library itself.

5. chemkit: This package is the closest to the goals outlined by **Spear**, but appears to no longer be developed. From a software design perspective, however, chemkit is superbly thorough and its overarching design principles can be taken from this package, such as the use of abstract classes to provide a data-driven architecture. Unfortunately, it is dependant on Qt for some of its core features, which is not ideal for the creation of other libraries as this library is large and should be avoided when creating command-line tools.

7.5.2 Class design

Molecule class

The center class of **Spear** is the *Molecule* class and its design reflects the overall design of the **Spear** library. It contains two major components: (1) an *std::vector* constraining the 3D location of all atoms in the molecule and (2) an undirected graph structure which will be described in greater details in the following paragraph. The *Molecule* implements methods for addition, removal, and swapping of atoms and bonds. Random access iterators are available for going through all atoms and all bonds and are implemented through the underlying graph structure. All-atom and bond properties are stored in a 'data-driven' manner (IE in a vector or matrix).

The most complex component of the *Molecule* class is the graph structure used to store the topology of the molecule. It is an undirected unweighted adjacency list graph which uses a vector to store node components and a set to store the edge components. These container types are chosen so that vertex index can be one-to-one with the *std::vector* used to contain the atom positions and other properties

stored in the *Molecule* (for example the charge on an atom). Each node/vertex is given a name, which is an unsigned integer corresponding to the atomic number of the element. Similarly, each bond is given a bond name, corresponding to the order of the bond (single, double, triple, quadruple, amide, or aromatic). Unlike other packages, **Spear** relies on its data-driven structures to store atom and bond properties instead of storing points to class objects as the node/vertex, allowing for the use of default graph matching algorithms (see *Functional Group*).

In order to simplify the data-driven paradigm used by the *Molecule* class, convenience classes called *AtomVertex* and *BondEdge* are available and are returned by the members of the *Molecule* class. *BondEdge* provides access to the bond properties *source* and *target*, *order*, and *index* of a bond where both *source* and *target* return *AtomVertex* objects. The *AtomVertex* allows one to query information about the atom, and iterate over the atoms neighbors. Internally, *AtomVertex* is implicitly convertible to a vector index which allows it to quickly look up data in its parent *Molecule*. Some *AtomVertex* properties require the use of an *AtomType*, which can be implemented by sub-classing the *AtomType* class. Querying the number of implicit hydrogens, formal charge, aromaticity, and planarity of an atom is done through this class, which is described in a later section.

Functional groups

The *PartialCharge* class is an abstract class that allows one to implement partial charge schemes to implemented and stored in the *Molecule* class. It is privatetly derived from an *std::vector<double>*, which allows charges to be accessed easily in the 'data-driven' model of the *Molecule* class. This internal vector is kept congruent with the parent molecule's size.

Atomtype class

Since many features in **Spear** are built upon the assignment of hybridization and aromaticity, the user is given the ability to provide custom definitions of these concepts. This is done through sub-classing the *AtomType* class; a process which requires the implementation of a method to retrieve the hybridization, aromaticity, and planarity of each atom. Additionally, this class must be iterated over so that algorithms that depend on atomtype can access the data in a modern C++ fashion. Internally, each atomtype class must store an unsigned integer vector which is congruent with the parent molecule's size. This vector represents the atomtypes and can be converted to strings (and back) using template methods.

By *Default*, this class is added to a *Molecule* during construction. The internal atom type is simply the atomic number of the element and hybridization/planarity are determined through querying the bond orders of the atom. The aromaticity of the atom is defined using the RDKit aromaticity model.

Fingerprint class

Fingerprinting algorithms can be implemented by sub-classing the *Fingerprint* class. These implementations are graph–based algorithms which take the topology of a *Molecule* and produce a vector of counts, which are typically reduced to a vector of bits.

Scoring Functions

Given the ability to define atomtypes in **Spear** and the ability to perform matrix calculations with the Eigen library, it is easy to implement various scoring functions in **Spear**. This is done by sub-classing the *ScoringFunction* class and implementing the

score virtual function. One can implement knowledge–based, empirical, or physical scoring functions in this manner.

Force-field classes

Spear interfaces with OpenMM, a package from SimTK designed to perform molecular dynamics and minimization. The C++ interface to this library is fairly low-level and it requires the user to input all bonded, non-bonded, and external forces for each atom in the system. The **Spear** MD interface is built off of the one used by OpenMM and many of the implementation details which follow reflect this relationship.

The heart of the **Spear** MD interface is the *Simulation* class. Internally, it handles operates on the OpenMM classes *System*, *Context*, and *Platform*. When a *Molecule* is added to the *Simulation* class, the masses of the *Molecule* atoms are passed to the OpenMM. A *BondedForceField* class must be added in addition to the molecule, which in turn adds any bonded forces to the *System* and provide the masses of each atom added. After all molecules have been added to the system, a user may add an unlimited number of *NonBondedForceField* classes, which operate on all added molecules simultaneously (unlike the *BondedForceField* class). Once all forces have been added to the *Simulation*, the user may create the appropriate contexts and call the corresponding *minimize* and *step* member functions. The *System* is immutable at this point and must be reinitialized if any underlying changes are to be made.

BondedForceField is an abstract class with two pure virtual methods, *masses(Molecule)* and *add_forces(Molecule mol, OpenMM::System)*. The first returns the mass of each atom in the *Molecule* as per the force-field definitions and the second adds all bonded forces in the *Molecule* to the *OpenMM::System*. Since these classes are virtual, all underlying functionality must be implemented on a per force field

basis. Currently, classes are available to read the *FFMXL* format and create a *BondedForceField* from the definitions in the file. Only the AMBER and GAFF forcefields have been tested thoroughly, however.

NonBondedForceField is an abstract class with one pure virtual method, *add_forces (MoleculeVector vector, OpenMM::System system).* This method must be called after the addition of all *Molecule* objects, as per the requirements of OpenMM. The first argument contains all *Molecule* objects added to a *Simulation* which are used to populate the second argument with forces that act on the entire system.

Since *BondedForceField* and *NonBondedForceField* are both pure virtual classes, they can be combined in a multi-inheritance manner. This is done for force-field definitions which contain both bonded and non-bonded forces.

7.5.3 Integration with other languages and internal projects

Spearmint

Spear is used as the backend of the Molecular Interaction using New Technology MINT. This virtual reality 'game'. This interface is built as a small C++ library (which links to **Spear** for all the heavy-lifting) that exports a C Application Binary Interface (ABI). Since this library is developed to support the development of MINT, its feature-set is geared towards the scoring of ligand poses, adding/removing atoms/bonds, and performing energy minimization/dynamics. The ABI supported by Spearmint is data-driven to match both **Spear** and the rendering pipeline developed by the MINT team.

StarMix

Integration with the remainder of the CANDIY-suite is provided through the StarMix project where it integrates with Lemon (see Chapter 2.1). Currently, one can create Lemon workflows which incorporate **Spear** features (such as scoring).

Python (planned)

A Python interface created through the use of PyBind11 is planned. This will resemble the one provided by Lemon. It will likely be developed outside of the main **Spear** development branch (as is done by chemfiles). This should prevent the creation of Python centric wrappers being built into the C++ library (as is done by RDKit and presumably OpenEye). An interface to Julia could also be created in a similar manner (it is required by the Julia community for this to be a separate project). Other language interfaces can be built using the C interface.

8. OUTLOOK

While each chapter has given a future works section to describe what immediate steps should be taken at each of these scales, I have included my outlooks for these different scales in this final chapter. These are developments that I believe will occur in the next 10-50 years and are not realizable today, but I hope that the work shown in this dissertation will help pave the way for these developments.

8.1 The future of proteome scale drug design

8.1.1 Addressing the issue of reverse design

In the upcoming years, proteome scale drug design will only grow in popularity. Recent works show that the consideration of proteome wide effects can be used to measure the toxicity of a compound [17]. Unfortunately, these works only apply these techniques to known molecules closer to the end of the drug design pipeline. A major road block to the adoption of these techniques is likely due to a lack of high quality, fast, accurate, and widely available tools. While the CANDO platform is useful for tackling this issue, it is difficult to use it to design new drugs as it is unable to give topological or structural insights for a given repurposing prediction. Recent work for incorporating small molecule fingerprints into the CANDO matrix may yield some insights, but lack of 3D information cripples its ability to be used for design principals. Although it may be tempting to use a technique such as docking to add this vital information, a notion that helped to inspire the development of CANDOCK, these techniques offer few design opportunities that can be applied across 10,000s of

proteins. However, the rise of GAN methodologies offer a glimpse into the future of what can be done to optimize a potential molecule to fit a chemeoproteomic profile. Such techniques have already been applied at the protein scale (see the respective section below) and the ever increasing computational power and parallelism will allow these techniques to be applied to a greater number of proteins. Additionally, newer techniques such as conditional generation (see Fig 8.1) can be used to generate a new molecule directly from a chemeoproteomic signature. Therefore, we can conclude that the issue of generating drugs using proteome scale interactions will be solved within the decade.

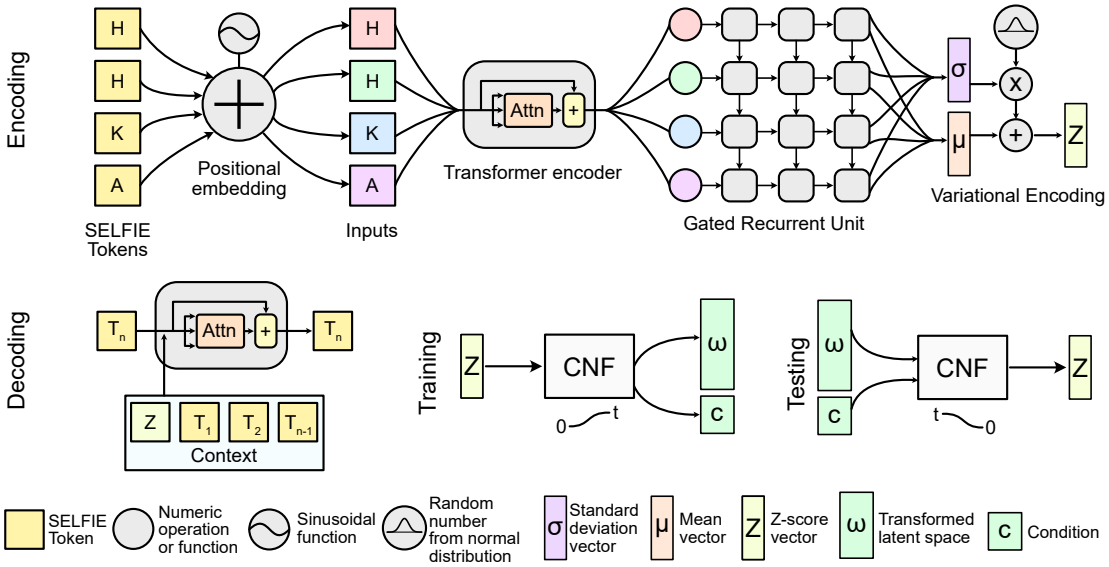


Figure 8.1. Compound generation via a conditional neural network.

8.1.2 Improvements to the accuracy of chemeoproteomic signatures

The generation of a chemeoproteomic signature is computationally demanding and therefore many approximations are used to determine the small–molecule protein interactions that make up this signature. Using current technology, Lawrence

Livermore National Lab has calculated these interactions using molecular dynamics [388]. While this is a noble effort, molecular dynamics is an approach that is too slow to be applied for drug design. Current and future work on the calculation of these interactions using solely the topology of a molecule and the sequence of a protein may prove to address this issue, especially if the affinities between more compounds and non-target proteins are measured. Therefore, it is the lack of known proteome wide interactions that limit the development of more accurate methods for this task. These issues will be discussed more in the section on the future of small–molecule scale drug design, but it is worth noting this as one of the most important issues in proteome scale design.

8.2 The future of cell scale drug design

8.2.1 Prediction of cell response to a compound

The majority of issues facing proteome scale design are also problems for cell scale design. However, many pharmaceutical companies have begun to adopt a paradigm for kinase inhibitors that takes into consideration a specific set of kinases that are related to a specific cancer. A major influence in the increased interest of industry in this area is the rise of resistance to cancer treatments. Additionally, it is currently feasible for modern technology to screen a single molecule against many known targets, leaving the analysis of these results as the only major concern. As shown in Chapter 2, the use of traditional machine learning techniques can be used to create models from this data. Therefore, there will be a rise in the coupling between automated testing on multiple proteins and machine learning models used to predict cellular response to the compound. The pharmaceutical industry is especially poised to take advantage of these developments and will likely incorporate them into their medicinal chemistry

pipelines. Similar generational techniques can be used as the ones mentioned for proteome scale, the major difference being context used to generate the molecules will be derived from data collected from high–throughput instrumentation instead of theoretical scores obtained from docking studies. The reasons for this decline of docking will be detailed in the respective section on small molecule design.

Since a large amount of training data is not available for training cell specific models, especially when given a constrained chemical space, one shot learning approaches will be deployed more in this field. One example of such a model is the contrastive loss model shown below. This model allows for multiple single measures for a given compound to be combined and used to compare the molecule against a molecule with known activity. The result of applying this method is similar to dynamic 'clustering' method that 'learns' how to reduce the distance between similar compounds.

8.2.2 The future of cell differentiation detection

The ability to distinguish differing cell types through analytical techniques will likely not be a huge issue in the upcoming years. Current advances in tandem MS and the associated statistical tools will continue to be honed in upcoming years. The types of cells being identified will, however, change as the focus in biology slowly shifts toward the immune system and related cells. Instead, new techniques will be developed to measure the fate of a cell and its response to certain stimuli such as treatment with a given compound, activation of a signaling pathway, or another related event. This prediction is based on the belief that biology is not static and, although the current state of a cell is important, being able to predict how it will change over time using its current state will prove to be paramount. The current use of Multiple Reaction Monitoring (MRMs) for the identification of lipids, proteins, and (in the near future) metabolites will continue to grow. These techniques will not be

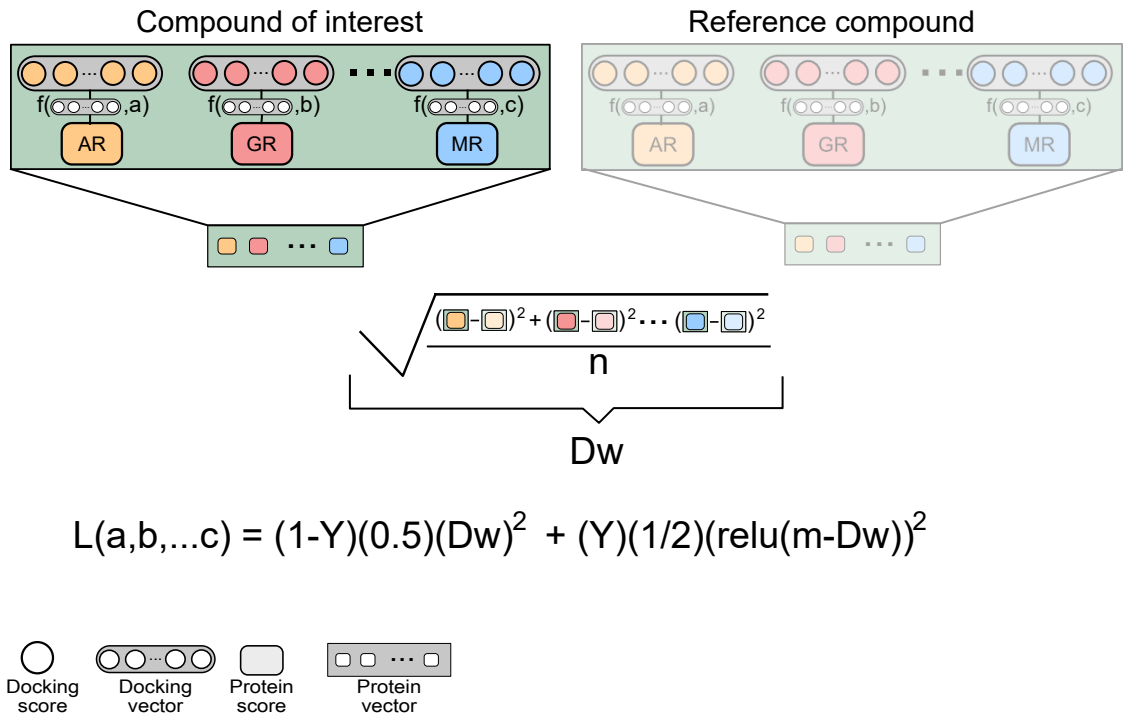


Figure 8.2. The contrastive loss model. Here multiple measurements for the activity of a single compound against a single protein are combined into a single score. Multiple compound–protein scores are then used to calculate the distance.

used in isolation but will be instead combined with other techniques such as machine learning.

8.3 The future of protein scale drug design

8.3.1 Combination of docking with machine learning

Machine learning based scoring functions are growing in popularity [216,389,390], but still do not offer the vast improvements in the ability to predict relative binding affinities for a single protein target. In the near term, and even now, docking methodologies are being combined with machine learning architectures. One exam-

ple is the fusion of CANDOCK (see chapter 3) and graph neural networks (GNNs) to target PD-1/PD-L1. This work is the beginning of the future of docking, but once pure graph methodologies are developed for these targets, docking will begin to decline. This is not to say that molecular topologies contain all the information needed for predicting these interactions, but newer methods will be able to incorporate environmental factors in ways that compensate for the information supplied by docking.

8.3.2 The decline of docking

There are currently dozens of docking methods and counting and very few of these methods are used outside of academia to design new molecules. The few that are used provide scores that do not correlate well with binding affinity. As the number of docking methodologies increase, so do the number of machine learning methods that can provide insight into how well a compound will bind to a given receptor. These methods will become both faster and more accurate than what mainstream docking methodologies provide. Currently, they cannot be used to model the interactions a molecule has in a 3D environment, but new techniques such as normalizing flow may be able to address this issue. A recent work suggested that compounds can be generated using only binding site interactions [391], so it is reasonable to predict a similar method can be used to position a graph topology in a binding site. Once these methods are fully developed, docking in its current form will decrease in popularity as it is replaced with faster and more accurate methods. This will be partially aided by improvements in automated synthesis and testing.

8.3.3 The rise of autonomous instrumentation

Combinatorial synthesis has proven as an apt methodology to create large libraries of compounds quickly and effectively and continued advancements in this field will have important impacts on protein scale synthesis. It is currently assumed in many docking benchmarks that a single compound will always interact with a single target, but hopefully the increased testing of small molecules with additional targets will prove to the community that this notion is false. This will further the decline of docking methods and show that machine learning methods mentioned in the last section are superior to docking. Therefore, the rise of increased automation will lessen the importance of docking in the drug design community and a harmony between increased automation for high–throughput synthesis and machine learning based design will be the final nail in the docking coffin.

8.4 The future of small–molecule scale drug design

The rise of autonomous instrumentation has not been paralleled with advances in the ability to access how well a given synthesis has been carried out. The chapters on machine learning applications in analytical chemistry shows some progress in this area, but there is still a lot of work to be done in elucidating full structure from spectra. Within the next few decades, these issues will be solved using a combination of spectra (IR, MS, NMR) and knowledge of the starting material of a reaction. At this time, the need for additional analytical techniques is probably not needed to realize this goal, instead an increased amount of public data for existing techniques is required. These advancements will be made on top of the advancements made for the identification of functional groups and will likely incorporate expert–based methods in addition to those derived from machine learning methods. The development of algorithms which

can stitch together desperate functional groups into a complete molecule will likely be a fundamental improvement to functional groups derived from spectra using machine learning. These algorithms may (and probably will) incorporate aspects of ML to make these connections and it can be envisioned that they resemble the automated synthesis algorithms of today. These advancements will be mostly applied to the identification of natural products as improvements in functional group prediction will be substantial enough on their own to revolutionize automated synthesis in their own right.

The goal of full structure elucidation in an automated manner will require additional advances in the field of machine learning applied to analytical chemistry. One solution may be to use compression graphs to generate large portions of a molecule (multiple moieties, see Fig 8.3). Such a methodology, in combination with storing chemical data in a hierarchy, will yield large advances in the field. This work has already been seen in the creation of junction tree variational autoencoders [291], but future developments are needed to generate new moieties not present at training.

The identification of these 'compressed' graphs can be done using a one–shot learning approach. These approaches will be necessary until enough training data is available to create more generalized models. An example of such an approach is shown in the following figure.

8.5 The future of proton scale drug design

It is clear that ML methods have had a large impact on the development of new reactions and these advancements will continue over the next decades. What will be more important and powerful is the combination of improvements in automated synthesis along with the ability to accurately predict functional groups *de novo* will enable completely automated reaction screening. This will allow autonomous robots

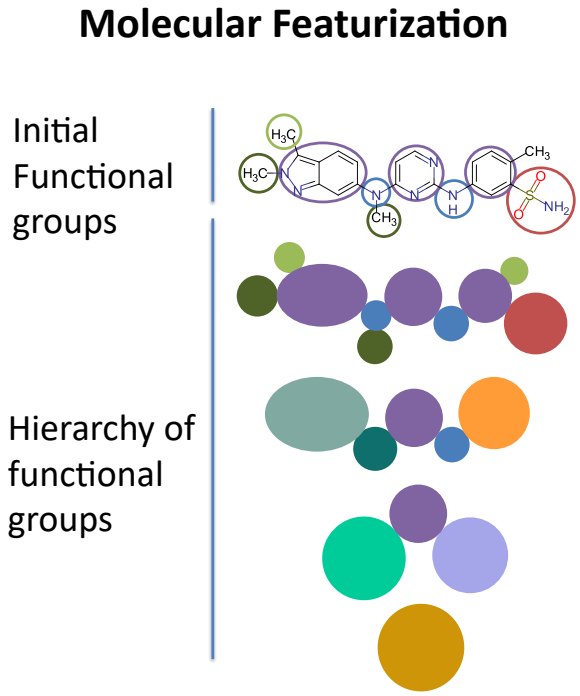


Figure 8.3. Compression graph of a molecule to combine multiple moieties together. This is an example where the hierarchies are determined by the location of rotatable bonds.

to find new reactions and associated conditions, revolutionizing how new reactions are found and verified. These reactions will still be verified using techniques such as DFT, but this field will be changed greatly in its own right by techniques that allow for quick energy calculations. These techniques are already developed, but at this time are not accurate enough to be used in transition state identification. Once this bridge is crossed, the direct use of QM will decline and these new methods will be used in increasing popularity. This will happen naturally over time, especially since QM properties can be calculated *en masse* and these calculations have already been shown to yield massive improvements throughout the world of molecular simulations [392, 393].

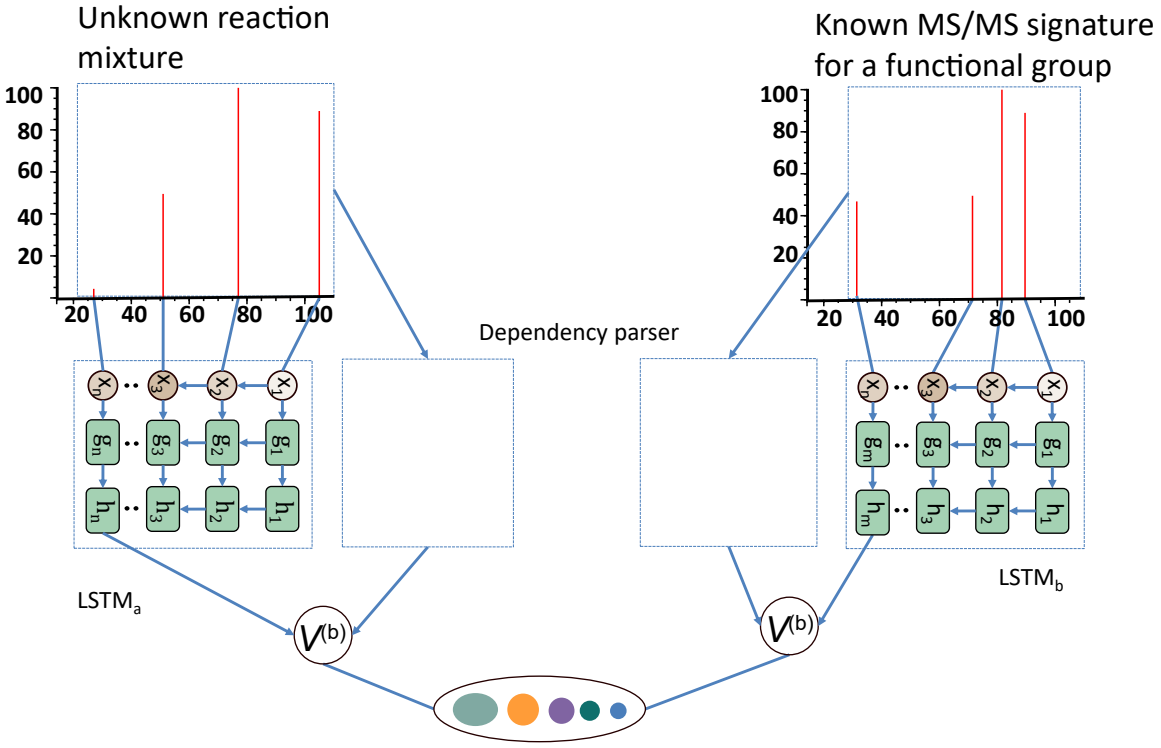


Figure 8.4. Compression graph of a molecule to combine multiple moieties together. This is an example where the hierarchies are determined by the location of rotatable bonds.

REFERENCES

[1] Nicklas Bonander and Roslyn M Bill. Relieving the first bottleneck in the drug discovery pipeline: using array technologies to rationalize membrane protein production. *Expert Review of Proteomics*, 6(5):501–505, 10 2009.

[2] Stephen H Gillespie and Kasha Singh. XDR-TB, what is it; how is it treated; and why is therapeutic failure so high? *Recent patents on anti-infective drug discovery*, 6(2):77–83, 5 2011.

[3] Jeremy A. Horst, Adrian Laurenzi, Brady Bernard, and Ram Samudrala. Computational Multitarget Drug Discovery. In *Polypharmacology in Drug Discovery*, pages 263–301. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2 2012.

[4] Leonard V Sacks and Rachel E Behrman. Challenges, successes and hopes in the development of novel TB therapeutics. *Future Medicinal Chemistry*, 1(4):749–756, 7 2009.

[5] Sean Ekins, Antony J. Williams, Matthew D. Krasowski, and Joel S. Freundlich. In silico repositioning of approved drugs for rare and neglected diseases. *Drug Discovery Today*, 16(7-8):298–310, 4 2011.

[6] Kui. Xu and Timothy R. Coté. Database identifies FDA-approved drugs with potential to be repurposed for treatment of orphan diseases. *Briefings in Bioinformatics*, 12(4):341–345, 7 2011.

[7] Ekachai Jenwitheesuk and Ram Samudrala. Identification of Potential Multitarget Antimalarial Drugs. *JAMA*, 294(12):1487, 9 2005.

[8] Ekachai Jenwitheesuk, Jeremy A Horst, Kasey L Rivas, Wesley C Van Voorhis, and Ram Samudrala. Novel paradigms for drug discovery: computational multitarget screening. *Trends in pharmacological sciences*, 29(2):62–71, 2 2008.

[9] Mark Minie, Gaurav Chopra, Geetika Sethi, Jeremy Horst, George White, Ambrish Roy, Kaushik Hatti, and Ram Samudrala. CANDO and the infinite drug discovery frontier. *Drug Discovery Today*, 19(9):1353–1363, 9 2014.

[10] Joshua S. Swamidass. Mining small-molecule screens to repurpose drugs. *Briefings in Bioinformatics*, 12(4):327–335, 7 2011.

[11] Jingyuan Ren, Lei Xie, Wilfred W. Li, and Philip E. Bourne. SMAP-WS: A parallel web service for structural proteome-wide ligand-binding site comparison. *Nucleic Acids Research*, 38(SUPPL. 2):W441–W444, 7 2010.

[12] Joshua M. Costin, Ekachai Jenwitheesuk, Shee-Mei Lok, Elizabeth Hunsperger, Kelly A. Conrads, Krystal A. Fontaine, Craig R. Rees, Michael G. Rossmann, Sharon Isern, Ram Samudrala, and Scott F. Michael. Structural Optimization and De Novo Design of Dengue Virus Entry Inhibitory Peptides. *PLoS Neglected Tropical Diseases*, 4(6):e721, 6 2010.

[13] Cindo O. Nicholson, Joshua M. Costin, Dawne K. Rowe, Li Lin, Ekachai Jenwitheesuk, Ram Samudrala, Sharon Isern, and Scott F. Michael. Viral entry inhibitors block dengue antibody-dependent enhancement in vitro. *Antiviral Research*, 89(1):71–74, 1 2011.

[14] Jeremy A. Horst, Ursula Pieper, Andrej Sali, L. Zhan, Guarav Chopra, Ram Samudrala, and John D.B. Featherstone. Strategic Protein Target Analysis for Developing Drugs to Stop Dental Caries. *Advances in Dental Research*, 24(2):86–93, 9 2012.

[15] Geetika Sethi, Gaurav Chopra, and Ram Samudrala. Multiscale Modelling of Relationships between Protein Classes and Drug Behavior Across all Diseases Using the CANDO Platform. *Mini Reviews in Medicinal Chemistry*, 15(8):705–717, 2015.

[16] Xiaochu Ma, Jie Zhou, Changhao Wang, Brandon Carter-Cooper, Fan Yang, Elizabeth Larocque, Jonathan Fine, Genichiro Tsuji, Gaurav Chopra, Rena G. Lapidus, and Herman O. Sintim. Identification of New FLT3 Inhibitors That Potently Inhibit AML Cell Lines via an Azo Click-It/Staple-It Approach. *ACS Medicinal Chemistry Letters*, 8(5):492–497, 2017.

[17] Marimar Hernandez-Perez, Gaurav Chopra, Jonathan Fine, Abass M. Conteh, Ryan M. Anderson, Amelia K. Linnemann, Chanelle Benjamin, Jennifer B. Nelson, Kara S. Benninger, Jerry L. Nadler, David J. Maloney, Sarah A. Tersey, and Raghavendra G. Mirmira. Inhibition of 12/15-Lipoxygenase Protects Against β-Cell Oxidative Stress and Glycemic Deterioration in Mouse Models of Type 1 Diabetes. *Diabetes*, 66(11):2875–2887, 2017.

[18] Richard Pink, Alan Hudson, Marie Annick Mouriès, and Mary Bendig. Opportunities and challenges in antiparasitic drug discovery, 9 2005.

[19] Marcy J. Balunas and A. Douglas Kinghorn. Drug discovery from medicinal plants. *Life Sciences*, 78(5):431–441, 12 2005.

[20] Manhoi Hur, Alexis Ann Campbell, Marcia Almeida-de Macedo, Ling Li, Nick Ransom, Adarsh Jose, Matt Crispin, Basil J. Nikolau, and Eve Syrkin Wurtele. A global approach to analysis and interpretation of metabolic data for plant natural product discovery. *Natural Product Reports*, 30(4):565, 4 2013.

[21] Gordon M. Cragg, David J. Newman, and Kenneth M. Snader. Natural products in drug discovery and development, 1997.

[22] Eric Patridge, Peter Gareiss, Michael S. Kinch, and Denton Hoyer. An analysis of FDA-approved drugs: Natural products and their derivatives, 2 2016.

[23] Helen M Berman, John Westbrook, J Feng, Z Gilliland, Shindyalov, and I N Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):181–198, 1 2000.

[24] Chris Sander and Reinhard Schneider. Database of homology???derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics*, 9(1):56–68, 1 1991.

[25] Gaurav Chopra and Ram Samudrala. Exploring Polypharmacology in Drug Discovery and Repurposing Using the CANDO Platform. *Current pharmaceutical design*, 22(21):3109–23, 2016.

[26] Andrew C. R. Martin. Mapping PDB chains to UniProtKB entries. *Bioinformatics*, 21(23):4297–4301, 12 2005.

[27] Jonathan Fine, Janez Konc, Ram Samudrala, and Gaurav Chopra. CANDOCK: Chemical atomic network based hierarchical flexible docking algorithm using generalized statistical potentials. *bioRxiv*, page 442897, 1 2019.

[28] Brady Bernard and Ram Samudrala. A generalized knowledge-based discriminatory function for biomolecular interactions. *Proteins: Structure, Function and Bioinformatics*, 76(1):115–128, 2009.

[29] Gaurav Chopra, Sashank Kaushik, Peter L. Elkin, and Ram Samudrala. Combating Ebola with repurposed therapeutics using the CANDO platform. *Molecules*, 21(12):1537, 11 2016.

[30] Ekachai Jenwitheesuk and Ram Samudrala. Improved prediction of HIV-1 protease-inhibitor binding energies by molecular dynamics simulations. *BMC Structural Biology*, 3(1):1–9, 2003.

[31] Tianle Ma and Aidong Zhang. AffinityNet: semi-supervised few-shot learning for disease type prediction. *ArXiv*, 5 2018.

[32] Ram Samudrala and John Moult. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction 1 1Edited by F. Cohen. *Journal of Molecular Biology*, 275(5):895–916, 2 1998.

[33] Ram Samudrala and John Moult. A graph-theoretic algorithm for comparative modeling of protein structure. *Journal of Molecular Biology*, 279(1):287–302, 5 1998.

[34] Yu Xia, Enoch S. Huang, Michael Levitt, and Ram Samudrala. Ab initio construction of protein tertiary structures using a hierarchical approach. *Journal of Molecular Biology*, 300(1):171–185, 6 2000.

[35] Ram Samudrala, Yu Xia, Enoch Huang, and Michael Levitt. Ab initio protein structure prediction using a combined hierarchical approach. *Proteins*, Suppl 3:194–198, 1999.

[36] Feixiong Cheng, Yadi Zhou, Weihua Li, Guixia Liu, and Yun Tang. Prediction of chemical-protein interactions network with weighted network-based inference method. *PLoS ONE*, 7(7):e41064, 7 2012.

[37] Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Computational Biology*, 8(5):e1002503, 5 2012.

[38] Gary D Bader, Doron Betel, and Christopher W V Hogue. BIND: The Biomolecular Interaction Network Database. *Nucleic Acids Research*, 31(1):248–250, 1 2003.

[39] Péter Csermely, Vilmos Ágoston, and Sándor Pongor. The efficiency of multi-target drugs: The network approach might help drug design, 4 2005.

[40] American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders*. American Psychiatric Association, 5 2013.

[41] Zaakiyah Mohamed-Kaloo and Sumaya Laher. Perceptions of mental illness among Muslim general practitioners in South Africa. *South African Medical Journal*, 104(5):350, 3 2014.

[42] Nicolas Rüsch, Sara Evans-Lacko, and Graham Thornicroft. What is a mental illness? Public views and their effects on attitudes and disclosure. *Australian & New Zealand Journal of Psychiatry*, 46(7):641–650, 7 2012.

[43] L Skaer, M Robison, A Sclar, and S Galin. Treatment of depressive illness among children and adolescents in the United States. *Current Therapeutic Research*, 61(10):692, 2000.

[44] Annette Bauer, Martin Knapp, and Michael Parsonage. Lifetime costs of perinatal anxiety and depression. *Journal of Affective Disorders*, 192:83–90, 3 2016.

[45] Claire Henderson, Sara Evans-Lacko, and Graham Thornicroft. Mental illness stigma, help seeking, and public health programs. *American journal of public health*, 103(5):777–80, 5 2013.

[46] Medco. America's State of Mind. Technical report, Medco, 2011.

[47] Michael J. Sernyak, Douglas L. Leslie, Renato D. Alarcon, Miklos F. Losonczy, and Robert Rosenheck. Association of Diabetes Mellitus With Use of Atypical Neuroleptics in the Treatment of Schizophrenia. *American Journal of Psychiatry*, 159(4):561–566, 4 2002.

[48] National Alliance on Mental Illness. Mental Health Facts, 2018.

[49] National Alliance on Mental Illness. What is mental illness: Mental illness facts. *Www.Nami.Org*, pages 1–2, 2013.

[50] Georgina R. Cox, Patch Callahan, Rachel Churchill, Vivien Hunot, Sally N. Merry, Alexandra G. Parker, and Sarah E. Hetrick. Psychological therapies versus antidepressant medication, alone and in combination for depression in children and adolescents, 11 2014.

[51] Gil Sharon, Timothy R. Sampson, Daniel H. Geschwind, and Sarkis K. Mazmanian. The Central Nervous System and the Gut Microbiome. *Cell*, 167(4):915–932, 11 2016.

[52] Huda Akil, Joshua Gordon, Rene Hen, Jonathan Javitch, Helen Mayberg, Bruce McEwen, Michael J. Meaney, and Eric J. Nestler. Treatment resistant depression: A multi-scale, systems biology approach. *Neuroscience & Biobehavioral Reviews*, 84:272–288, 1 2018.

[53] Xiao Zheng, Xueli Zhang, Guangji Wang, and Haiping Hao. Treat the brain and treat the periphery: toward a holistic approach to major depressive disorder. *Drug Discovery Today*, 20(5):562–568, 5 2015.

[54] Marc Antoine Crocq. Historical and cultural aspects of man's relationship with addictive drugs, 2007.

[55] Alexander Shulgin and Ann Shulgin. Phenethylamines I Have Known And Loved: A Chemical Love Story. *Transform*, pages 1–1188, 1991.

[56] Alexander Shulgin and Ann Shulgin. *Tryptamines I Have Known And Loved*. Transform, 1997.

[57] Maria Chiara Paolino, Alessandro Ferretti, Laura Papetti, Maria Pia Villa, and Pasquale Parisi. Cannabidiol as potential treatment in refractory pediatric epilepsy, 2016.

[58] Joep Killestein. Cannabinoids in the Treatment of Epilepsy. *New England Journal of Medicine*, 374(1):94–95, 1 2016.

[59] Leor Roseman, Lysia Demetriou, Matthew B. Wall, David J. Nutt, and Robin L. Carhart-Harris. Increased amygdala responses to emotional faces after psilocybin for treatment-resistant depression. *Neuropharmacology*, 142:263–269, 11 2017.

[60] Thu H. Pham, Indira Mendez-David, Céine Defaix, Bruno P. Guiard, Laurent Tritschler, Denis J. David, and Alain M. Gardier. Ketamine treatment involves medial prefrontal cortex serotonin to induce a rapid antidepressant-like activity in BALB/cJ mice. *Neuropharmacology*, 112:198–209, 1 2017.

[61] James J.H. Rucker, Jonathan Iliff, and David J. Nutt. Psychiatry & the psychedelic drugs. Past, present & future, 12 2018.

[62] Juliana M Nascimento and Daniel Martins-de Souza. The proteome of schizophrenia. *npj Schizophrenia*, 1(1):14003, 12 2015.

[63] Jiyeong Lee, Eun-Jeong Joo, Hee-Joung Lim, Jong-Moon Park, Kyu Young Lee, Arum Park, AeEun Seok, HooKeun Lee, and Hee-Gyoo Kang. Proteomic Analysis of Serum from Patients with Major Depressive Disorder to Compare Their Depressive and Remission Statuses. *Psychiatry Investigation*, 12(2):249, 2015.

[64] Lucia Carboni. The contribution of proteomic studies in humans, animal models, and after antidepressant treatments to investigate the molecular neurobiology of major depression, 2015.

[65] Regina Taurines, Edward Dudley, Julia Grassl, Andreas Warnke, Manfred Gerlach, Andrew N. Coogan, and Johannes Thome. Review: Proteomic research in psychiatry, 2011.

[66] Firas H. Kobeissy, Shankar Sadasivan, Jing Liu, Mark S. Gold, and Kevin K.W. Wang. Psychiatric research: Psychoproteomics, degradomics and systems biology, 2008.

[67] Bruce E Bloom. Recent successes and future predictions on drug repurposing for rare diseases. *Expert Opinion on Orphan Drugs*, 4(1):1–4, 1 2016.

[68] T. I. Oprea and J. Mestres. Drug Repurposing: Far Beyond New Targets for Old Drugs. *The AAPS Journal*, 14(4):759–763, 12 2012.

[69] Rong Xu and Quanqiu Wang. Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing. *BMC bioinformatics*, 14(1):181, 6 2013.

[70] Antonio Lavecchia and Carmen Cerchia. In silico methods to address polypharmacology: Current status, applications and future perspectives, 2016.

[71] Brown R.A., Monti P.M., Myers M.G., Martin R.A., Rivinus T., and Dubreuil M.E. Depression among cocaine abusers in treatment: Relation to cocaine and alcohol use and treatment outcome. *American Journal of Psychiatry*, 155(2):220–225, 1998.

[72] John C M Brust. Seizures and substance abuse: treatment considerations. *Neurology*, 67(12 Suppl. 4):S45–S48, 12 2006.

[73] Kerry J Ressler and Charles B Nemeroff. Role of serotonergic and noradrenergic systems in the pathophysiology of depression and anxiety disorders, 2000.

[74] Eunice Y. Chen, Lauren Matthews, Charese Allen, Janice R. Kuo, and Marsha Marie Linehan. Dialectical behavior therapy for clients with binge-eating disorder or bulimia nervosa and borderline personality disorder. *International Journal of Eating Disorders*, 41(6):505–512, 9 2008.

[75] Ana B. Cerezo, Ángela Leal, M. Antonia Álvarez-Fernández, Ruth Hornedo-Ortega, Ana M. Troncoso, and M. Carmen García-Parrilla. Quality control and determination of melatonin in food supplements. *Journal of Food Composition and Analysis*, 45:80–86, 2016.

[76] Pawan Kumar Jha, Etienne Challet, and Andries Kalsbeek. Circadian rhythms in glucose and lipid metabolism in nocturnal and diurnal mammals, 2015.

[77] Kathryn J. Reid and Sabra M. Abbott. Jet lag and shift work disorder. *Sleep Medicine Clinics*, 10(4):523–535, 2015.

[78] Nava Zisapel and Moshe Laudon. Derivating of tryptamine and analgous compounds and pharamaceutical formulations containing them, 2004.

[79] Martin D. Schechter and Richard A. Glennon. Cathinone, cocaine and methamphetamine: similarity of behavioral effects. *Pharmacology, Biochemistry and Behavior*, 22(6):913–916, 1985.

[80] Kenneth Blum, M. Foster Olive, Kevin K.W. Wang, Marcelo Febo, Joan Borsten, John Giordano, Mary Hauser, and Mark S. Gold. Hypothesizing that designer drugs containing cathinones ("bath salts") have profound neuroinflammatory effects and dangerous neurotoxic response following human consumption. *Medical Hypotheses*, 81(3):450–455, 2013.

[81] Gaylord Ellison. Neural degeneration following chronic stimulant abuse reveals a weak link in brain, fasciculus retroflexus, implying the loss of forebrain control circuitry, 2002.

[82] F. Ivy Carroll, Bruce E. Blough, S. Wayne Mascarella, Hernán A. Navarro, Ronald J. Lukas, and M. Imad Damaj. Bupropion and bupropion analogs as treatments for CNS disorders. *Advances in Pharmacology*, 69:177–216, 2014.

[83] Kristin J. Holm and Caroline M. Spencer. Bupropion: A review of its use in the management of smoking cessation. *Drugs*, 59(4):1007–1024, 2000.

[84] Alessandro E. Vento, Fabrizio Schifano, Federica Gentili, Francesco Pompei, John M. Corkery, Georgios D. Kotzalidis, and Paolo Girardi. Bupropion perceived as a stimulant by two patients with a previous history of cocaine misuse. *Annali dell'Istituto Superiore di Sanita*, 49(4):402–405, 2013.

[85] Maria Sullivan and Elizabeth Evans. Abuse and misuse of antidepressants. *Substance Abuse and Rehabilitation*, page 107, 2014.

[86] Chulathida Chomchai and Boonying Manaboriboon. Stimulant Methamphetamine and Dextromethorphan Use Among Thai Adolescents: Implications for Health of Women and Children. *Journal of Medical Toxicology*, 8(3):291–294, 2012.

[87] Eric C. Strain, Maxine L. Stitzer, Ira A. Liebson, and George E. Bigelow. Dose-response effects of methadone in the treatment of opioid dependence. *Annals of Internal Medicine*, 119(1):23–27, 7 1993.

[88] Hyoung Chun Kim, Guoying Bing, Eun Joo Shin, Hyun Seon Jhoo, Mi Ae Cheon, Seung Hyun Lee, Ki Hwan Choi, Joo Il Kim, and Wang Kee Jhoo. Dextromethorphan affects cocaine-mediated behavioral pattern in parallel with a long-lasting Fos-related antigen-immunoreactivity. *Life Sciences*, 69(6):615–624, 2001.

[89] Luigi Pulvirenti, Claudia Balducci, and George F. Koob. Dextromethorphan reduces intravenous cocaine self-administration in the rat. *European Journal of Pharmacology*, 321(3):279–283, 1997.

[90] Wang Kee Jhoo, Eun Joo Shin, Young Ho Lee, Mi Ae Cheon, Ki Wan Oh, Seog Youn Kang, Chaeyoung Lee, Byung Cheon Yi, and Hyoung Chun Kim. Dual effects of dextromethorphan on cocaine-induced conditioned place preference in mice. *Neuroscience Letters*, 288(1):76–80, 2000.

[91] H C Kim, B K Park, S Y Hong, and W K Jhoo. Dextromethorphan alters the reinforcing effect of cocaine in the rat. *Methods Find Exp Clin Pharmacol*, 19(9):627–631, 1997.

[92] Eun-Joo Shin, Jae-Hyung Bach, Sung Youl Lee, Jeong Min Kim, Jinhwa Lee, Jau-Shyong Hong, Toshitaka Nabeshima, and Hyoung-Chun Kim. Neuropsychotoxic and Neuroprotective Potentials of Dextromethorphan and Its Analogs. *Journal of Pharmacological Sciences*, 116(2):137–148, 2011.

[93] Helene Perrotin-Brunel, Maaike C. Kroon, Maaike J.E. Van Roosmalen, Jaap Van Spronsen, Cor J. Peters, and Geert Jan Witkamp. Solubility of non-psychoactive cannabinoids in supercritical carbon dioxide and comparison with psychoactive cannabinoids. *Journal of Supercritical Fluids*, 55(2):603–608, 2010.

[94] George Marsaglia, Wai Wan Tsang, and Jingbo Wang. Evaluating Kolmogorov's Distribution. *Journal of Statistical Software*, 8(18), 2015.

[95] Peter W. Rose, Andreas Prlić, Chunxiao Bi, Wolfgang F. Bluhm, Cole H. Christie, Shuchismita Dutta, Rachel Kramer Green, David S. Goodsell, John D. Westbrook, Jesse Woo, Jasmine Young, Christine Zardecki, Helen M. Berman, Philip E. Bourne, and Stephen K. Burley. The RCSB Protein Data Bank: Views of structural biology for basic and applied research and education. *Nucleic Acids Research*, 43(D1):D345–D356, 1 2015.

[96] Michael J. Hartshorn, Marcel L. Verdonk, Gianni Chessari, Suzanne C. Brewerton, Wijnand T.M. Mooij, Paul N. Mortenson, and Christopher W. Murray. Diverse, high-quality test set for the validation of protein-ligand docking performance. *Journal of Medicinal Chemistry*, 50(4):726–741, 2007.

[97] Zhihai Liu, Minyi Su, Li Han, Jie Liu, Qifan Yang, Yan Li, and Renxiao Wang. Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Accounts of Chemical Research*, 50(2):302–309, 2017.

[98] 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, 7 2012.

[99] Gaurav Chopra, Christopher M. Summa, and Michael Levitt. Solvent dramatically affects protein structure refinement. *Proceedings of the National Academy of Sciences of the United States of America*, 105(51):20239–20244, 12 2008.

[100] Anthony R. Bradley, Alexander S. Rose, Antonín Pavelka, Yana Valasatava, Jose M. Duarte, Andreas Prlić, and Peter W. Rose. MMTF—An efficient file format for the transmission, visualization, and analysis of macromolecular structures. *PLoS Computational Biology*, 13(6):e1005575, 6 2017.

[101] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Technical report, Google Inc, 2004.

[102] Guillaume Fraux. chemfiles/chemfiles: 0.9.2. *Zenodo*, 1 2020.

[103] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 11 2009.

[104] Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 5 2012.

[105] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

[106] Anas Younes, Stephen Ansell, Nathan Fowler, Wyndham Wilson, Sven De Vos, John Seymour, Ranjana Advani, Andres Forero, Franck Morschhauser, Marie Jose Kersten, Kensei Tobinai, Pier Luigi Zinzani, Emanuele Zucca, Jeremy Abramson, and Julie Vose. The landscape of new drugs in lymphoma, 6 2017.

[107] Andrew M. Intlekofer and Anas Younes. Precision therapy for lymphoma—current state and future directions, 10 2014.

[108] Daniel Merrill, Ran An, Hao Sun, Bakhtiyor Yakubov, Daniela Matei, John Turek, and David Nolte. Intracellular Doppler Signatures of Platinum Sensitivity Captured by Biodynamic Profiling in Ovarian Xenografts. *Scientific Reports*, 6, 1 2016.

[109] Hao Sun, Daniel Merrill, Ran An, John Turek, Daniela Matei, and David D. Nolte. Biodynamic imaging for phenotypic profiling of three-dimensional tissue culture. *Journal of Biomedical Optics*, 22(1):016007, 1 2017.

[110] Ran An, Dan Merrill, Larisa Avramova, Jennifer Sturgis, Maria Tsiper, J. Paul Robinson, John Turek, and David D. Nolte. Phenotypic profiling of raf inhibitors and mitochondrial toxicity in 3D tissue using biodynamic imaging. *Journal of Biomolecular Screening*, 19(4):526–537, 4 2014.

[111] Ran An, Chunmin Wang, John Turek, Zoltan Machaty, and David D. Nolte. Biodynamic imaging of live porcine oocytes, zygotes and blastocysts for viability assessment in assisted reproductive technologies. *Biomedical Optics Express*, 6(3):963, 3 2015.

[112] M R Custead, R An, J J Turek, G E Moore, D D Nolte, and M O Childress. Predictive value of ex vivo biodynamic imaging in determining response to chemotherapy in dogs with spontaneous non-Hodgkin's lymphomas: a preliminary study. *Convergent Science Physical Oncology*, 1(1):015003, 10 2015.

[113] Honggu Choi, Zhe Li, Hao Sun, Dan Merrill, John Turek, Michael Childress, and David Nolte. Biodynamic digital holography of chemoresistance in a preclinical trial of canine B-cell lymphoma. *Biomedical Optics Express*, 9(5):2214, 5 2018.

[114] Jonathan Fine, Anand Rasjashekar, and Gaurav Chopra. Accurate and Automated de novo Identification of Molecular Functional Groups Using Deep Learning Architectures. *ChemRxiv*, 2019.

[115] Michal Brylinski and Jeffrey Skolnick. Cross-reactivityvirtual profiling of the human kinome by X-ReactKIN – a Chemical Systems Biology approach. *Molecular pharmaceutics*, 7(6):2324–2333, 2010.

[116] Hossam M. Ashtawy and Nihar R. Mahapatra. A comparative assessment of predictive accuracies of conventional and machine learning scoring functions for protein-ligand binding affinity prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(2):335–347, 2015.

[117] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 1 2013.

[118] Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), 12 2014.

[119] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 10 1968.

[120] Bernd Kuhn, Wolfgang Guba, Jérôme Hert, David Banner, Caterina Bissantz, Simona Ceccarelli, Wolfgang Haap, Matthias Körner, Andreas Kuglstatter, Christian Lerner, Patrizio Mattei, Werner Neidhart, Emmanuel Pinard, Markus G. Rudolph, Tanja Schulz-Gasch, Thomas Woltering, and Martin Stahl. A Real-World Perspective on Molecular Design. *Journal of Medicinal Chemistry*, 59(9):4087–4102, 2016.

[121] Ahmed Allam, Mate Nagy, George Thoma, and Michael Krauthammer. Neural networks versus Logistic regression for 30 days all-cause readmission prediction. *Scientific Reports*, 9(1), 12 2019.

[122] Evangelia Christodoulou, Jie Ma, Gary S. Collins, Ewout W. Steyerberg, Jan Y. Verbakel, and Ben Van Calster. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models, 6 2019.

[123] Jonathan Fine, Rachel Lackner, Ram Samudrala, and Gaurav Chopra. Computational chemoproteomics to understand the role of selected psychoactives in treating mental health indications. *Scientific Reports*, 9(1):1–15, 12 2019.

[124] William Mangione and Ram Samudrala. Identifying Protein Features Responsible for Improved Drug Repurposing Accuracies Using the CANDO Platform: Implications for Drug Design. *Molecules*, 24(1):167, 2019.

[125] Lissette Gomez, Jason R. Kovac, and Dolores J. Lamb. CYP17A1 inhibitors in castration-resistant prostate cancer. *Steroids*, 95:80–87, 3 2015.

[126] Tea Lanišnik Rižner and Trevor M Penning. Role of aldo-keto reductase family 1 (AKR1) enzymes in human steroid metabolism. *Steroids*, 79:49–63, 1 2014.

[127] Isabel Coutinho, Tanya K Day, Wayne D Tilley, and Luke A Selth. Androgen receptor signaling in castration-resistant prostate cancer: a lesson in persistence. *Endocrine-Related Cancer*, 23(12):T179–T197, 12 2016.

[128] RuiQi Chen, Yue Yu, and Xuesen Dong. Progesterone receptor in the prostate: A potential suppressor for benign prostatic hyperplasia and prostate cancer. *The Journal of Steroid Biochemistry and Molecular Biology*, 166:91–96, 2 2017.

[129] Junjian Wang, June X. Zou, Xiaoqian Xue, Demin Cai, Yan Zhang, Zhijian Duan, Qiuping Xiang, Joy C. Yang, Maggie C. Louie, Alexander D. Borowsky, Allen C. Gao, Christopher P Evans, Kit S. Lam, Jianzhen Xu, Hsing-Jien Kung, Ronald M. Evans, Yong Xu, and Hong-Wu Chen. ROR-γ drives androgen receptor expression and represents a therapeutic target in castration-resistant prostate cancer. *Nature Medicine*, 22(5):488–496, 5 2016.

[130] Krapcho M Miller D Brest A Yu M Ruhl J Tatalovich Z Mariotto A Lewis DR Chen HS Feuer EJ Cronin KA (eds). Noone AM, Howlader N. Prostate Cancer - Cancer Stat Facts, 2018.

[131] M. Kirby, C. Hirst, and E. D. Crawford. Characterising the castration-resistant prostate cancer population: A systematic review, 11 2011.

[132] William T. Lowrance, Mohammad Hassan Murad, William K. Oh, David F. Jarrard, Matthew J. Resnick, and Michael S. Cookson. Castration-Resistant Prostate Cancer: AUA Guideline Amendment 2018. *Journal of Urology*, 200(6):1264–1272, 12 2018.

[133] Maha Hussain, Karim Fizazi, Fred Saad, Per Rathenborg, Neal Shore, Ubirajara Ferreira, Petro Ivashchenko, Eren Demirhan, Katharina Modelska, De Phung, Andrew Krivoshik, and Cora N. Sternberg. Enzalutamide in men with nonmetastatic, castration-resistant prostate cancer. *New England Journal of Medicine*, 378(26):2465–2474, 2018.

[134] Andrew Anighoro and Jürgen Bajorath. Three-Dimensional Similarity in Molecular Docking: Prioritizing Ligand Poses on the Basis of Experimental Binding Modes. *Journal of Chemical Information and Modeling*, 56(3), 2016.

[135] Jin Xu and Yun Qiu. Current opinion and mechanistic interpretation of combination therapy for castration-resistant prostate cancer, 5 2019.

[136] Si-sheng Ou-Yang, Jun-yan Lu, Xiang-qian Kong, Zhong-jie Liang, Cheng Luo, and Hualiang Jiang. Computational drug discovery. *Acta Pharmacologica Sinica*, 33(9):1131–1140, 9 2012.

[137] Nataraj S. Pagadala, Khajamohiddin Syed, and Jack Tuszynski. Software for molecular docking: a review, 2017.

[138] Caterina Bissantz, Philippe Bernard, Marcel Hibert, and Didier Rognan. Protein-based virtual screening of chemical databases. II. Are homology models of g-protein coupled receptors suitable targets? *Proteins: Structure, Function, and Bioinformatics*, 50(1):5–25, 11 2002.

[139] Yu Chian Chen. Beware of docking! *Trends in Pharmacological Sciences*, 36(2):78–95, 2015.

[140] Yvonne Y. Li, Jianghong An, and Steven J. M. Jones. A Computational Approach to Finding Novel Targets for Existing Drugs. *PLoS Computational Biology*, 7(9):e1002139, 9 2011.

[141] Francis Gaudreault and Rafael J. Najmanovich. FlexAID: Revisiting Docking on Non-Native-Complex Structures. *Journal of Chemical Information and Modeling*, 55(7):1323–1336, 7 2015.

[142] Janaina Cruz Pereira, Ernesto Raúl Caffarena, and Cicero Nogueira dos Santos. Boosting Docking-Based Virtual Screening with Deep Learning. *Journal of Chemical Information and Modeling*, 2016.

[143] Jonathan Fine, Janez Konc, Ram Samudrala, and Gaurav Chopra. CANDOCK: Chemical Atomic Network-based Hierarchical Flexible Docking Algorithm using Generalized Statistical Potentials. *Journal of chemical information and modeling*, 60(3):1509–1527, 3 2020.

[144] Yankang Jing, Yuemin Bian, Ziheng Hu, Lirong Wang, and Xiang-Qun Sean Xie. Deep Learning for Drug Design: an Artificial Intelligence Paradigm for Drug Discovery in the Big Data Era. *The AAPS Journal*, 20(3):58, 5 2018.

[145] 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.

[146] Erik Gawehn, Jan A. Hiss, and Gisbert Schneider. Deep Learning in Drug Discovery, 2016.

[147] Jianlong Peng, Jing Lu, Qiancheng Shen, Mingyue Zheng, Xiaomin Luo, Weiliang Zhu, Hualiang Jiang, and Kaixian Chen. In silico site of metabolism prediction for human UGT-catalyzed reactions. *Bioinformatics*, 30(3):398–405, 2014.

[148] Sourav Das, Michael P. Krein, and Curt M. Breneman. Binding affinity prediction with property-encoded shape distribution signatures. *Journal of Chemical Information and Modeling*, 50(2):298–308, 2010.

[149] Hongdong Li, Yizeng Liang, and Qingsong Xu. Support vector machines and its applications in chemistry. *Chemometrics and Intelligent Laboratory Systems*, 95(2):188–198, 2009.

[150] Derek T. Ahneman, Jesús G. Estrada, Shishi Lin, Spencer D. Dreher, and Abigail G. Doyle. Predicting reaction performance in C–N cross-coupling using machine learning. *Science*, 360(6385):186–190, 4 2018.

[151] Cheng Wang and Yingkai Zhang. Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest. *Journal of Computational Chemistry*, 38(3):169–177, 2017.

[152] Florbela Pereira, Kaixia Xiao, Diogo A.R.S. Latino, Chengcheng Wu, Qingyou Zhang, and Joao Aires-De-Sousa. Machine Learning Methods to Predict Density Functional Theory B3LYP Energies of HOMO and LUMO Orbitals. *Journal of Chemical Information and Modeling*, 57(1):11–21, 2017.

[153] Junshui Ma, Robert P. Sheridan, Andy Liaw, George E. Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015.

[154] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole Von Lilienfeld. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *Journal of Chemical Theory and Computation*, 13(11):5255–5264, 2017.

[155] Jennifer N. Wei, David Duvenaud, and Alán Aspuru-Guzik. Neural Networks for the Prediction of Organic Chemistry Reactions. *ACS Central Science*, 2(10):725–732, 10 2016.

[156] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L Guimaraes, and Alán Aspuru-Guzik. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). *ChemRxiv*, pages 1–18, 2017.

[157] Mostapha Benhenda. ChemGAN challenge for drug discovery: can AI reproduce natural chemical diversity? *ArXiv*, 8 2017.

[158] Evgeny Putin, Arip Asadulaev, Yan Ivanenkov, Vladimir Aladinskiy, Benjamin Sanchez-Lengeling, Alán Aspuru-Guzik, and Alex Zhavoronkov. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, 58(6):1194–1204, 2018.

[159] Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *ChemRxiv*, pages 1–7, 5 2017.

[160] Artur Kadurin, Sergey Nikolenko, Kuzma Khrabrov, Alex Aliper, and Alex Zhavoronkov. DruGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Molecular Pharmaceutics*, 14(9):3098–3104, 2017.

[161] Marwin H.S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 1 2018.

[162] Zheng Xu, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Seq2seq Fingerprint. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics - ACM-BCB '17*, pages 285–294, New York, New York, USA, 2017. ACM Press.

[163] Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low Data Drug Discovery with One-Shot Learning. *ACS Central Science*, 3(4):283–293, 2017.

[164] Kathrin Heikamp and Jürgen Bajorath. Support vector machines for drug discovery. *Expert Opinion on Drug Discovery*, 9(1):93–104, 1 2014.

[165] Richard A. Friesner, Jay L. Banks, Robert B. Murphy, Thomas A. Halgren, Jasna J. Klicic, Daniel T. Mainz, Matthew P. Repasky, Eric H. Knoll, Mee Shelley, Jason K. Perry, David E. Shaw, Perry Francis, and Peter S. Shenkin. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 3 2004.

[166] Emanuele Perola, W. Patrick Walters, and Paul S. Charifson. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. *Proteins: Structure, Function, and Bioinformatics*, 56(2):235–249, 4 2004.

[167] Zhan Deng, Claudio Chuaqui, and Juswinder Singh. Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein−Ligand Binding Interactions. *Journal of Medicinal Chemistry*, 47(2):337–344, 1 2004.

[168] Y Pouliot, A P Chiang, and A J Butte. Predicting adverse drug reactions using publicly available PubChem bioassay data. *Clinical Pharmacology and Therapeutics*, 90(1):90–99, 7 2011.

[169] H A Carlson and J A McCammon. Accommodating protein flexibility in computational drug design. *Molecular pharmacology*, 57(2):213–8, 2 2000.

[170] 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, 6 2009.

[171] Jacek Biesiada, Aleksey Porollo, Prakash Velayutham, Michal Kouril, and Jaroslaw Meller. Survey of public domain software for docking simulations and virtual screening. *Human genomics*, 5(5):497–505, 7 2011.

[172] Suzanne C Brewerton. The use of protein-ligand interaction fingerprints in docking. *Current opinion in drug discovery & development*, 11(3):356–64, 5 2008.

[173] Sheng-You Huang and Xiaoqin Zou. Ensemble docking of multiple protein structures: Considering protein structural variations in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, 66(2):399–421, 11 2006.

[174] S.-Y. Huang and Xiaoqin Zou. Efficient molecular docking of NMR structures: Application to HIV-1 protease. *Protein Science*, 16(1):43–51, 1 2006.

[175] Arsen V. Grigoryan, Hong Wang, and Timothy J. Cardozo. Can the Energy Gap in the Protein-Ligand Binding Energy Landscape Be Used as a Descriptor in Virtual Ligand Screening? *PLoS ONE*, 7(10):e46532, 10 2012.

[176] Oleg Trott and Arthur J Olson. Software news and update 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, 1 2010.

[177] Marcel L. Verdonk, Jason C. Cole, Michael J. Hartshorn, Christopher W. Murray, and Richard D. Taylor. Improved protein-ligand docking using GOLD. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 8 2003.

[178] Shuangye Yin, Lada Biedermannova, Jiri Vondrasek, and Nikolay V Dokholyan. MedusaScore: An accurate force field-based scoring function for virtual drug screening. *Journal of Chemical Information and Modeling*, 48(8):1656–1662, 8 2008.

[179] Feng Ding, Shuangye Yin, and Nikolay V. Dokholyan. Rapid Flexible Docking Using a Stochastic Rotamer Library of Ligands. *Journal of Chemical Information and Modeling*, 50(9):1623–1632, 9 2010.

[180] Jian Wang and Nikolay V. Dokholyan. MedusaDock 2.0: Efficient and Accurate Protein-Ligand Docking with Constraints. *Journal of Chemical Information and Modeling*, 59(6):2509–2515, 6 2019.

[181] Elizabeth Yuriev and Paul A. Ramsland. Latest developments in molecular docking: 2010-2011 in review. *Journal of Molecular Recognition*, 26(5):215–239, 5 2013.

[182] Kelly L. Damm-Ganamet, Richard D. Smith, James B. Dunbar, Jeanne A. Stuckey, and Heather A. Carlson. CSAR benchmark exercise 2011-2012: Evaluation of results from docking and relative ranking of blinded congeneric series. *Journal of Chemical Information and Modeling*, 53(8):1853–1870, 8 2013.

[183] 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, Giovanna Tedesco, Ian D. Wall, James M. Woolven, Catherine E. Peishoff, and Martha S. Head. A critical assessment of docking programs and scoring functions. *Journal of Medicinal Chemistry*, 49(20):5912–5931, 2006.

[184] Esther Kellenberger, Jordi Rodrigo, Pascal Muller, and Didier Rognan. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins: Structure, Function and Genetics*, 57(2):225–242, 8 2004.

[185] Zhe Wang, Huiyong Sun, Xiaojun Yao, Dan Li, Lei Xu, Youyong Li, Sheng Tian, and Tingjun Hou. Comprehensive evaluation of ten docking programs on a diverse set of protein-ligand complexes: the prediction accuracy of sampling power and scoring power. *Physical chemistry chemical physics : PCCP*, 18:12964–12975, 2016.

[186] Holger Claußen, Christian Buning, Matthias Rarey, and Thomas Lengauer. FLEXE: Efficient molecular docking considering protein structure variations. *Journal of Molecular Biology*, 308(2):377–395, 4 2001.

[187] Andrew R. Leach. Ligand docking to proteins with discrete side-chain flexibility. *Journal of Molecular Biology*, 235(1):345–356, 1 1994.

[188] Joannis Apostolakis, Andreas Plückthun, and Amedeo Caflisch. Docking small ligands in flexible binding sites. *Journal of Computational Chemistry*, 19(1):21–37, 1998.

[189] Elaine C. Meng, Daniel A. Gschwend, Jeffrey M. Blaney, and Irwin D. Kuntz. Orientational sampling and rigid-body minimization in molecular docking. *Proteins: Structure, Function, and Genetics*, 17(3):266–278, 11 1993.

[190] Hongtao Zhao and Amedeo Caflisch. Discovery of ZAP70 inhibitors by high-throughput docking into a conformation of its kinase domain generated by molecular dynamics. *Bioorganic & Medicinal Chemistry Letters*, 23(20):5721–5726, 10 2013.

[191] Gaurav Chopra, Nir Kalisman, and Michael Levitt. Consistent refinement of submitted models at CASP using a knowledge-based potential. *Proteins: Structure, Function, and Bioinformatics*, 78(12):n/a–n/a, 6 2010.

[192] João P G L M Rodrigues, Michael Levitt, and Gaurav Chopra. KoBaMIN: A knowledge-based minimization web server for protein structure refinement. *Nucleic Acids Research*, 40(W1):323–8, 7 2012.

[193] Panida Lertkiatmongkol, Anunchai Assawamakin, George White, Gaurav Chopra, Pornpimol Rongnoparut, Ram Samudrala, and Sissades Tongsima. Distal effect of amino acid substitutions in CYP2C9 polymorphic variants causes differences in interatomic interactions against (S)-warfarin. *PloS one*, 8(9):e74053, 9 2013.

[194] M. I. Zavodszky and Leslie A. Kuhn. Side-chain flexibility in protein-ligand binding: The minimal rotation hypothesis. *Protein Science*, 14(4):1104–1114, 3 2005.

[195] Sergio Ruiz-Carmona, Daniel Alvarez-Garcia, Nicolas Foloppe, A. Beatriz Garmendia-Doval, Szilveszter Juhos, Peter Schmidtke, Xavier Barril, Roderick E. Hubbard, and S. David Morley. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Computational Biology*, 10(4), 2014.

[196] Praveen Nedumpully-Govindan, Domen B. Jemec, and Feng Ding. CSAR Benchmark of Flexible MedusaDock in Affinity Prediction and Nativelike Binding Pose Selection. *Journal of Chemical Information and Modeling*, 56(6):1042–1052, 6 2016.

[197] Heather A. Carlson, Richard D. Smith, Kelly L. Damm-Ganamet, Jeanne A. Stuckey, Aqeel Ahmed, Maire A. Convery, Donald O. Somers, Michael Kranz, Patricia A. Elkins, Guanglei Cui, Catherine E. Peishoff, Millard H. Lambert, and James B. Dunbar. CSAR 2014: A Benchmark Exercise Using Unpublished Data from Pharma. *Journal of Chemical Information and Modeling*, 56(6):1063–1077, 2016.

[198] Kenji Onodera, Kazuhito Satou, and Hiroshi Hirota. Evaluations of molecular docking programs for virtual screening. *Journal of Chemical Information and Modeling*, 47(4):1609–1618, 2007.

[199] Miklos Feher and Christopher I. Williams. Effect of input differences on the results of docking calculations. *Journal of Chemical Information and Modeling*, 49(7):1704–1714, 7 2009.

[200] Yuri Pevzner, Emilie Frugier, Vinushka Schalk, Amedeo Caflisch, and H Lee Woodcock. Fragment-based docking: Development of the CHARMMing web user interface as a platform for computer-aided drug design. *Journal of Chemical Information and Modeling*, 54(9):2612–2620, 9 2014.

[201] Richard K. Belew, Stefano Forli, David S. Goodsell, T. J. O'Donnell, and Arthur J. Olson. Fragment-Based Analysis of Ligand Dockings Improves Classification of Actives. *Journal of Chemical Information and Modeling*, 56(8):1597–1607, 8 2016.

[202] Santiago Vilar, Giorgio Cozza, and Stefano Moro. Medicinal Chemistry and the Molecular Operating Environment (MOE): Application of QSAR and Molecular Docking to Drug Discovery. *Current Topics in Medicinal Chemistry*, 8(18):1555–1572, 12 2008.

[203] Irene Luque and Ernesto Freire. Structural Stability of Binding Sites: Consequences for Binding Affinity and Allosteric Effects. *Proteins: Structure, Function and Genetics*, 4:63–71, 2000.

[204] Michael T. Zimmermann, Sumudu P. Leelananda, Andrzej Kloczkowski, and Robert L. Jernigan. Combining statistical potentials with dynamics-based entropies improves selection from protein decoys and docking poses. *Journal of Physical Chemistry B*, 116(23):6725–6731, 6 2012.

[205] Elaine C. Meng and Richard A. Lewis. Determination of molecular topology and atomic hybridization states from heavy atom coordinates. *Journal of Computational Chemistry*, 12(7):891–898, 9 1991.

[206] William J. Allen, Trent E. Balius, Sudipto Mukherjee, Scott R. Brozell, Demetri T. Moustakas, P. Therese Lang, David A. Case, Irwin D. Kuntz, and Robert C. Rizzo. DOCK 6: Impact of new features and current docking performance. *Journal of Computational Chemistry*, 36(15):1132–1156, 6 2015.

[207] 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, 7 2004.

[208] Junmei Wang, Wei Wang, Peter A. Kollman, and David A. Case. Automatic atom type and bond type perception in molecular mechanical calculations. *Journal of Molecular Graphics and Modelling*, 25(2):247–260, 10 2006.

[209] 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):e1005659, 7 2017.

[210] Colin R. Groom, Ian J. Bruno, Matthew P. Lightfoot, Suzanna C. Ward, and IUCr. The Cambridge Structural Database. *Acta Crystallographica Section B Structural Science, Crystal Engineering and Materials*, 72(2):171–179, 4 2016.

[211] Janez Konc and Dušanka Janežič. An improved branch and bound algorithm for the maximum clique problem. *MATCH Communications in Mathematical and in Computer Chemistry MATCH Commun. Math. Comput. Chem*, 58:569–590, 2007.

[212] Coen Bron and Joep Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph [H]. *Communications of the ACM*, 16(9):575–577, 9 1973.

[213] Zsolt Zsoldos, Darryl Reid, Aniko Simon, Sayyed Bashir Sadjad, and A. Peter Johnson. eHiTS: A new fast, exhaustive flexible ligand docking system. *Journal of Molecular Graphics and Modelling*, 26(1):198–212, 7 2007.

[214] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 8 1989.

[215] Jonathan Fine and Gaurav Chopra. Lemon: a framework for rapidly mining structural information from the Protein Data Bank. *Bioinformatics*, 35(20):4165–4167, 10 2019.

[216] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling*, 59(2):895–913, 2 2019.

[217] Marvinbeans Chemaxon. Molecule File Converter, version 5.10. 1,(C) 1999−2012 ChemAxon Ltd.

[218] Ann E. Cleves and Ajay N. Jain. Knowledge-guided docking: Accurate prospective prediction of bound configurations of novel ligands using Surflex-Dock. *Journal of Computer-Aided Molecular Design*, 29(6):485–509, 6 2015.

[219] Jianing Lu, Xuben Hou, Cheng Wang, and Yingkai Zhang. Incorporating Explicit Water Molecules and Ligand Conformation Stability in Machine-Learning Scoring Functions. *Journal of Chemical Information and Modeling*, 59(11):4540–4549, 11 2019.

[220] Ajay N Jain. Scoring noncovalent protein-ligand interactions: A continuous differentiable function tuned to compute binding affinities. *Journal of Computer-Aided Molecular Design*, 10(5):427–440, 1996.

[221] Jim Ruppert, Will Welch, and Ajay N. Jain. Automatic identification and representation of protein binding sites for molecular docking. *Protein Science*, 6(3):524–533, 12 1996.

[222] Mohamed A. Khamis and Walid Gomaa. Comparative assessment of machine-learning scoring functions on PDBbind 2013. *Engineering Applications of Artificial Intelligence*, 45:136–151, 2015.

[223] Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *Journal of Chemical Information and Modeling*, 59(9):3981–3988, 9 2019.

[224] Anita K. Nivedha, David F. Thieker, Spandana Makeneni, Huimin Hu, and Robert J. Woods. Vina-Carb: Improving Glycosidic Angles during Carbohydrate Docking. *Journal of Chemical Theory and Computation*, 12(2):892–901, 2016.

[225] Kiumars Shahrokh, Anita Orendt, Garold S. Yost, and Thomas E. Cheatham. Quantum mechanically derived AMBER-compatible heme parameters for various states of the cytochrome P450 catalytic cycle. *Journal of computational chemistry*, 33(2):119–33, 1 2012.

[226] Simon S.J. Cross. Improved FlexX docking using FlexS-determined base fragment placement. *Journal of Chemical Information and Modeling*, 45(4):993–1001, 2005.

[227] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open Babel: An Open chemical toolbox. *Journal of Cheminformatics*, 3(10):33, 10 2011.

[228] ChemAxon. MarvinSketch, 2016.

[229] V M Bezhentsev, O A Tarasova, A V Dmitriev, A V Rudik, A A Lagunin, D A Filimonov, and V V Poroikov. Computer-aided prediction of xenobiotic metabolism in the human body. *Russian Chemical Reviews*, 85(8):854–879, 8 2016.

[230] Johannes Kirchmair, Mark J. Williamson, Jonathan D. Tyzack, Lu Tan, Peter J. Bond, Andreas Bender, and Robert C. Glen. Computational prediction of metabolism: Sites, products, SAR, P450 enzyme dynamics, and mechanisms, 2012.

[231] Johannes Kirchmair, Andreas H. Göller, Dieter Lang, Jens Kunze, Bernard Testa, Ian D. Wilson, Robert C. Glen, and Gisbert Schneider. Predicting drug metabolism: Experiment and/or computation?, 2015.

[232] Graeme L. Card, Bruce P. England, Yoshihisa Suzuki, Daniel Fong, Ben Powell, Byunghun Lee, Catherine Luu, Maryam Tabrizizad, Sam Gillette, Prabha N. Ibrahim, Dean R. Artis, Gideon Bollag, Michael V. Milburn, Sung Hou Kim, Joseph Schlessinger, and Kam Y.J. Zhang. Structural basis for the activity of drugs that inhibit phosphodiesterases. *Structure*, 12(12):2233–2247, 12 2004.

[233] Hartmuth C Kolb and K. Barry Sharpless. The growing impact of click chemistry on drug discovery, 12 2003.

[234] Shunsuke Chatani, Devatha P. Nair, and Christopher N. Bowman. Relative reactivity and selectivity of vinyl sulfones and acrylates towards the thiol-Michael addition reaction and polymerization. *Polymer Chemistry*, 4(4):1048–1055, 2013.

[235] Vera L.S. Freitas and Maria D.M.C. Ribeiro da Silva. Influence of hydroxyl functional group on the structure and stability of xanthone: A computational approach. *Molecules*, 23(11):2962, 11 2018.

[236] Bennett D. Marshall and Constantinos P. Bokis. A PC-SAFT model for hydrocarbons II: General model development. *Fluid Phase Equilibria*, 478:34–41, 12 2018.

[237] Yi Min Dai, Zhi Ping Zhu, Cao Zhong, Yue Fei Zhang, Ju Lan Zeng, and Li Xun. Prediction of boiling points of organic compounds by QSPR tools. *Journal of Molecular Graphics and Modelling*, 44:113–119, 7 2013.

[238] Michael Withnall, Hongming Chen, and Igor V. Tetko. Matched Molecular Pair Analysis on Large Melting Point Datasets: A Big Data Perspective. *ChemMedChem*, 13(6):599–606, 3 2018.

[239] Takashi Takei, Mayaka Nakada, Norinobu Yoshikawa, Yoshihisa Hiroe, and Hirohisa Yoshida. Effect of organic functional groups on the phase transition of organic liquids in silica mesopores. *Journal of Thermal Analysis and Calorimetry*, 123(3):1787–1794, 3 2016.

[240] Paula Yurkanis Bruice. *Essential Organic Chemistry*. Pearson, Upper Saddle Reiver, New Jersey, 3 edition, 2016.

[241] Fernanda B. Cordeiro, Christina R. Ferreira, Tiago Jose P. Sobreira, Karen E. Yannell, Alan K. Jarmusch, Agnaldo P. Cedenho, Edson G. Lo Turco, and R. Graham Cooks. Multiple reaction monitoring (MRM)-profiling for biomarker discovery applied to human polycystic ovarian syndrome. *Rapid Communications in Mass Spectrometry*, 31(17):1462–1470, 9 2017.

[242] A. Minai-Tehrani, N. Jafarzadeh, and K. Gilany. Metabolomics: a state-of-the-art technology for better understanding of male infertility, 8 2016.

[243] Andrew V. Ewing and Sergei G. Kazarian. Infrared spectroscopy and spectroscopic imaging in forensic science. *The Analyst*, 142(2):257–272, 1 2017.

[244] R. Risoluti, S. Materazzi, A. Gregori, and L. Ripani. Early detection of emerging street drugs by near infrared spectroscopy and chemometrics. *Talanta*, 153:407–413, 6 2016.

[245] Jeremy Manheim, Kyle C Doty, Gregory McLaughlin, and Igor K Lednev. Forensic Hair Differentiation Using Attenuated Total Reflection Fourier Transform Infrared (ATR FT-IR) Spectroscopy. *Applied Spectroscopy*, 70(7):1109–1117, 2016.

[246] Paul T. Anastas, Miriam Fontalvo Gómez, Boris Johnson Restrepo, Torsten Stelzer, and Rodolfo J. Romañach. Process Analytical Chemistry and Nondestructive Analytical Methods: The Green Chemistry Approach for Reaction Monitoring, Control, and Analysis. In *Handbook of Green Chemistry*, pages 257–288. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 1 2019.

[247] Matthew J Baker, Júlio Trevisan, Paul Bassan, Rohit Bhargava, Holly J Butler, Konrad M Dorling, Peter R Fielden, Simon W Fogarty, Nigel J Fullwood, Kelly A Heys, Caryn Hughes, Peter Lasch, Pierre L Martin-hirsch, Blessing Obinaju, Ganesh D Sockalingum, Josep Sulé-suso, and Rebecca J Strong. Using Fourier transform IR spectroscopy to analyze biological materials. *Nature Protocols*, 9(8):1771–1791, 2014.

[248] Jianfeng Li, D. Brynn Hibbert, Stephen Fuller, and Gary Vaughn. A comparative study of point-to-point algorithms for matching spectra. *Chemometrics and Intelligent Laboratory Systems*, 82(1-2 SPEC. ISS):50–58, 5 2006.

[249] Jennifer Griffiths. A Brief History of Mass Spectrometry. *Analytical Chemistry*, 80(15):5678–5683, 8 2008.

[250] Kai Dührkop, Huibin Shen, Marvin Meusel, Juho Rousu, and Sebastian Böcker. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 10 2015.

[251] Tobias Kind, Hiroshi Tsugawa, Tomas Cajka, Yan Ma, Zijuan Lai, Sajjan S. Mehta, Gert Wohlgemuth, Dinesh Kumar Barupal, Megan R. Showalter, Masanori Arita, and Oliver Fiehn. Identification of small molecules using accurate mass MS/MS search. *Mass Spectrometry Reviews*, 37(4):513–532, 7 2018.

[252] Emma L. Schymanski, Junho Jeon, Rebekka Gulde, Kathrin Fenner, Matthias Ruff, Heinz P. Singer, and Juliane Hollender. Identifying Small Molecules via High Resolution Mass Spectrometry: Communicating Confidence. *Environmental Science & Technology*, 48(4):2097–2098, 2 2014.

[253] Karsten Levsen and Helmut Schwarz. Gas-phase chemistry of collisionally activated ions. *Mass Spectrometry Reviews*, 2(1):77–148, 3 1983.

[254] Raymond E. March. An Introduction to Quadrupole Ion Trap Mass Spectrometry. *Journal of Mass Spectrometry*, 32(4):351–369, 4 1997.

[255] Shibdas Banerjee and Shyamalava Mazumdar. Electrospray Ionization Mass Spectrometry: A Technique to Access the Information beyond the Molecular Weight of the Analyte. *International Journal of Analytical Chemistry*, 2012:1–40, 2012.

[256] Franziska Hufsky and Sebastian Böcker. Mining molecular structure databases: Identification of small molecules based on fragmentation mass spectrometry data. *Mass Spectrometry Reviews*, 36(5):624–633, 9 2017.

[257] Junqi Li, Steven G Ballmer, Eric P Gillis, Seiko Fujii, Michael J Schmidt, Andrea M E Palazzolo, Jonathan W Lehmann, Greg F Morehouse, and Martin D Burke. Automated Process. *Organic Synthesis*, 347(6227):1221, 2015.

[258] Jarosław M. Granda, Liva Donina, Vincenza Dragone, De Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, 2018.

[259] Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic Reaction Prediction Using Neural Sequence-to-Sequence Models. *ACS Central Science*, 3(10):1103–1113, 10 2017.

[260] Tyler B. Hughes, Grover P. Miller, and S. Joshua Swamidass. Site of reactivity models predict molecular reactivity of diverse chemicals with glutathione. *Chemical Research in Toxicology*, 28(4):797–809, 4 2015.

[261] Tyler B. Hughes, Na Le Dang, Grover P. Miller, and S. Joshua Swamidass. Modeling reactivity to biological macromolecules with a deep multitask network. *ACS Central Science*, 2(8):529–537, 8 2016.

[262] 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.

[263] Mario Krenn, Florian Häse, Akshatkumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. *ArXiv*, 5 2019.

[264] Rushikesh Nalla, Rajdeep Pinge, Manish Narwaria, and Bhaskar Chaudhury. Priority based functional group identification of organic molecules using machine learning. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data - CoDS-COMAD '18*, pages 201–209, 2018.

[265] Sylvio Barbon, Ana Paula Ayub da Costa Barbon, Rafael Gomes Mantovani, and Douglas Fernandes Barbin. Machine Learning Applied to Near-Infrared Spectra for Chicken Meat Classification. *Journal of Spectroscopy*, 2018:1–12, 8 2018.

[266] Weiqiang Fu and W. Scott Hopkins. Applying Machine Learning to Vibrational Spectroscopy. *The Journal of Physical Chemistry A*, 122(1):167–171, 1 2018.

[267] Ralph J. Fessenden and László Györgyi. Identifying functional groups in IR spectra using an artificial neural network. *J. Chem. Soc., Perkin Trans. 2*, pages 1755–1762, 1991.

[268] Ernest W. Robb and Morton E. Munk. A neural network approach to infrared spectrum interpretation. *Mikrochimica Acta*, 100(3-4):131–155, 5 1990.

[269] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 10 2010.

[270] D.A. Case, D.S. Cerutti, III T.E. Cheatham, T.J. Giese H. Gohlke A.W. Goetz D. Greene N. Homeyer S. Izadi A. Kovalenko T.S. Lee S. LeGrand P. Li C. Lin J. Liu T. Luchko R. Luo D. Mermelstein K.M. Merz G. Monard H. D.M. York T.A. Darden, R.E. Duke, and P.A. Kollman. AMBER 2017, 2017.

[271] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem substance and compound databases. *Nucleic Acids Research*, 44(D1), 2016.

[272] Gregory A. Landrum. RDKit: Open-source cheminformatics, 2011.

[273] Gert Wohlgemuth, Sajjan S. Mehta, Ramon F. Mejia, Steffen Neumann, Diego Pedrosa, Tomáš Pluskal, Emma L. Schymanski, Egon L. Willighagen, Michael Wilson, David S. Wishart, Masanori Arita, Pieter C. Dorrestein, Nuno Bandeira, Mingxun Wang, Tobias Schulze, Reza M. Salek, Christoph Steinbeck, Venkata Chandrasekhar Nainala, Robert Mistrik, Takaaki Nishioka, and Oliver Fiehn. SPLASH, a hashed identifier for mass spectra, 11 2016.

[274] Sara Szymkuć, Ewa P. Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A. Grzybowski. Computer-Assisted Synthetic Planning: The End of the Beginning, 2016.

[275] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *ArXiv*, 5 2018.

[276] Claudio Loda, Elena Bernabe, Anna Nicoletti, Sergio Bacchi, and Riet Dams. Determination of epichlorohydrin in active pharmaceutical ingredients by gas chromatography-mass spectrometry. *Organic Process Research and Development*, 15(6):1388–1391, 11 2011.

[277] Edouard Niyonsaba, Jeremy M. Manheim, Ravikiran Yerabolu, and Hilkka I. Kenttämaa. Recent Advances in Petroleum Analysis by Mass Spectrometry. *Analytical Chemistry*, 91(1):156–177, 1 2019.

[278] Donald J. Douglas, Aaron J. Frank, and Dunmin Mao. Linear ion traps in mass spectrometry. *Mass Spectrometry Reviews*, 24(1):1–29, 1 2005.

[279] John Y. Kong, Zaikuan Yu, McKay W. Easton, Edouard Niyonsaba, Xin Ma, Ravikiran Yerabolu, Huaming Sheng, Tiffany M. Jarrell, Zhoupeng Zhang, Arun K. Ghosh, and Hilkka I. Kenttämaa. Differentiating Isomeric Deprotonated Glucuronide Drug Metabolites via Ion/Molecule Reactions in Tandem Mass Spectrometry. *Analytical Chemistry*, 90(15):9426–9433, 2018.

[280] Minli Zhang, Ryan Eismin, Hilkka Kenttämaa, Hui Xiong, Ye Wu, Doug Burdette, and Rebecca Urbanek. Identification of 2-aminothiazolobenzazepine metabolites in human, rat, dog, and monkey microsomes by ion-molecule reactions in linear quadrupole ion trap mass spectrometry. *Drug Metabolism and Disposition*, 43(3):358–366, 3 2015.

[281] Huaming Sheng, Weijuan Tang, Ravikiran Yerabolu, John Y. Kong, Peggy E. Williams, Minli Zhang, and Hilkka I. Kenttämaa. Mass spectrometric identification of the N -monosubstituted N -hydroxylamino functionality in protonated analytes via ion/molecule reactions in tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 29(8):730–734, 4 2015.

[282] Karinna M. Campbell, Michael A. Watkins, Sen Li, Marc N. Fiddler, Brian Winger, and Hilkka I. Kenttämaa. Functional Group Selective Ion/Molecule Reactions: Mass Spectrometric Identification of the Amido Functionality in Protonated Monofunctional Compounds. *The Journal of Organic Chemistry*, 72(9):3159–3165, 4 2007.

[283] Scott Gronert. Mass Spectrometric Studies of Organic Ion/Molecule Reactions. *Chemical Reviews*, 101(2):329–360, 2 2001.

[284] Tracy Donovan McCarley and Jennifer Brodbelt. Structurally Diagnostic Ion-Molecule Reactions and Collisionally Activated Dissociation of 1,4-Benzodiazepines in a Quadrupole Ion Trap Mass Spectrometer. *Analytical Chemistry*, 65(17):2380–2388, 1993.

[285] Steven C. Habicht, Nelson R. Vinueza, Enada F. Archibold, Penggao Duan, and Hilkka I. Kenttämaa. Identification of the carboxylic acid functionality by using electrospray ionization and ion-molecule reactions in a modified linear quadrupole ion trap mass spectrometer. *Analytical Chemistry*, 80(9):3416–3421, 2008.

[286] Garrett B. Goh, Nathan O. Hodas, and Abhinav Vishnu. Deep learning for computational chemistry. *Journal of Computational Chemistry*, 38(16):1291–1307, 6 2017.

[287] Alex T. Müller, Jan A. Hiss, and Gisbert Schneider. Recurrent Neural Network Model for Constructive Peptide Design. *Journal of Chemical Information and Modeling*, 58(2):472–479, 2018.

[288] 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.

[289] Connor W. Coley, William H. Green, and Klavs F. Jensen. Machine Learning in Computer-Aided Synthesis Planning. *Accounts of Chemical Research*, 51(5), 2018.

[290] Ola Engkvist, Per-Ola Norrby, Nidhal Selmi, Yu-hong Lam, Zhengwei Peng, Edward C. Sherer, Willi Amberg, Thomas Erhard, and Lynette A. Smyth. Computational prediction of chemical reactions: current status and outlook. *Drug Discovery Today*, 23(6):1203–1218, 6 2018.

[291] Wengong Jin, Connor W. Coley, Regina Barzilay, and Tommi Jaakkola. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. *Advances in Neural Information Processing Systems*, 2017-Decem:2608–2617, 9 2017.

[292] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, 1 2019.

[293] Rong En Fan, Kai Wei Chang, Cho Jui Hsieh, Xiang Rui Wang, and Chih Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(2008):1871–1874, 2008.

[294] John R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 3 1986.

[295] Pierre Geurts, Alexandre Irrthum, and Louis Wehenkel. Supervised learning with decision tree-based methods in computational and systems biology. *Molecular BioSystems*, 5(12):1593, 11 2009.

[296] Bjørn Helge Mevik and Ron Wehrens. The pls package: Principal component and partial least squares regression in R. *Journal of Statistical Software*, 18(2):1–23, 2007.

[297] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2 2010.

[298] Klaus Hechenbichler and Klaus Schliep. Weighted k-Nearest-Neighbor Techniques and Ordinal Classification. *Molecular Ecology*, 399:17, 2004.

[299] John B. O. Mitchell. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(5):468–481, 9 2014.

[300] Harry L. Morgan. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5(2):107–113, 5 1965.

[301] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 5 2010.

[302] Jennifer N. Wei, David Belanger, Ryan P. Adams, and D. Sculley. Rapid Prediction of Electron-Ionization Mass Spectrometry Using Neural Networks. *ACS Central Science*, 5(4):700–708, 4 2019.

[303] Connor W. Coley, Regina Barzilay, Tommi S. Jaakkola, William H. Green, and Klavs F. Jensen. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Central Science*, 3(5):434–443, 2017.

[304] Hanyu Gao, Thomas J Struble, Connor W Coley, Yuran Wang, William H Green, and Klavs F Jensen. Using Machine Learning to Predict Suitable Conditions for Organic Reactions. *ACS Central Science*, 4(11):1465–1476, 2018.

[305] Alexander Kensert, Jonathan Alvarsson, Ulf Norinder, and Ola Spjuth. Evaluating parameters for ligand-based modeling with random forest on sparse data sets. *Journal of Cheminformatics*, 10(1), 10 2018.

[306] Huaming Sheng, Peggy E. Williams, Weijuan Tang, Minli Zhang, and Hilkka I. Kenttämaa. Identification of the sulfoxide functionality in protonated analytes via ion/molecule reactions in linear quadrupole ion trap mass spectrometry. *Analyst*, 139(17):4296–4302, 2014.

[307] Penggao Duan, Todd A. Gillespie, Brian E. Winger, and Hilkka I. Kenttämaa. Identification of the aromatic tertiary N-oxide functionality in protonated analytes via ion/molecule reactions in mass spectrometers. *Journal of Organic Chemistry*, 73(13):4888–4894, 2008.

[308] Ravikiran Yerabolu, John Kong, McKay Easton, Raghavendhar R. Kotha, Joann Max, Huaming Sheng, Minli Zhang, Chungang Gu, and Hilkka I. Kenttämaa. Identification of Protonated Sulfone and Aromatic Carboxylic Acid Functionalities in Organic Molecules by Using Ion–Molecule Reactions Followed by Collisionally Activated Dissociation in a Linear Quadrupole Ion Trap Mass Spectrometer. *Analytical Chemistry*, 89(14):7398–7405, 7 2017.

[309] Edward P.L. Hunter and Sharon G. Lias. Evaluated gas phase basicities and proton affinities of molecules: An update. *Journal of Physical and Chemical Reference Data*, 27(3):413–656, 1998.

[310] Scott Gronert. Quadrupole ion trap studies of fundamental organic reactions. *Mass Spectrometry Reviews*, 24(1):100–120, 2005.

[311] Max Kuhn. Building Predictive Models in R Using the caret Package. *Journal of Statistical Software, Articles*, 28(5):1–26, 2008.

[312] M J Frisch, G W Trucks, H B Schlegel, G E Scuseria, M A Robb, J R Cheeseman, G Scalmani, V Barone, G A Petersson, H Nakatsuji, X Li, M Caricato, A V Marenich, J Bloino, B G Janesko, R Gomperts, B Mennucci, H P Hratchian, J V Ortiz, A F Izmaylov, J L Sonnenberg, D Williams-Young, F Ding, F Lipparini, F Egidi, J Goings, B Peng, A Petrone, T Henderson, D Ranasinghe, V G Zakrzewski, J Gao, N Rega, G Zheng, W Liang, M Hada, M Ehara, K Toyota, R Fukuda, J Hasegawa, M Ishida, T Nakajima, Y Honda, O Kitao, H Nakai, T Vreven, K Throssell, J A Montgomery Jr., J E Peralta, F Ogliaro, M J Bearpark, J J Heyd, E N Brothers, K N Kudin, V N Staroverov, T A Keith, R Kobayashi, J Normand, K Raghavachari, A P Rendell, J C Burant, S S Iyengar, J Tomasi, M Cossi, J M Millam, M Klene, C Adamo, R Cammi, J W Ochterski, R L Martin, K Morokuma, O Farkas, J B Foresman, and D J Fox. Gaussian16 Revision B.01, 2016.

[313] Yan Zhao and Donald G. Truhlar. Density functional theory for reaction energies: Test of meta and hybrid meta functionals, range-separated functionals, and other high-performance functionals. *Journal of Chemical Theory and Computation*, 7(3):669–676, 2011.

[314] A G Yurieva, Oleg Kh Poleshchuk, and Victor D Filimonov. Comparative analysis of a full-electron basis set and pseudopotential for the iodine atom in DFT quantum-chemical calculations of iodine-containing compounds. *Journal of Structural Chemistry*, 49(3):548–552, 5 2008.

[315] Roy Dennington, Todd A Keith, and John M Millam. GaussView Version 6, 2016.

[316] E. J. Corey and W Todd Wipke. Computer-Assisted Design of Complex Organic Syntheses Published by : American Association for the Advancement of Science Stable URL : http://www.jstor.org/stable/1727162 digitize , preserve and extend access to Science of New Computer-Assisted Design of C. *Science*, 166(3902):178–192, 1969.

[317] Antonio Lavecchia. Machine-learning approaches in drug discovery: Methods and applications. *Drug Discovery Today*, 20(3):318–331, 2015.

[318] Raquel Rodríguez-Pérez, Tomoyuki Miyao, Swarit Jasial, Martin Vogt, and Jürgen Bajorath. Prediction of Compound Profiling Matrices Using Machine Learning. *ACS Omega*, 3(4):4713–4723, 2018.

[319] David Fooshee, Aaron Mood, Eugene Gutman, Mohammadamin Tavakoli, Gregor Urban, Frances Liu, Nancy Huynh, David Van Vranken, and Pierre Baldi. Deep learning for chemical reaction prediction. *Molecular Systems Design and Engineering*, 3(3):442–452, 2018.

[320] Marwin H.S. Segler and Mark P. Waller. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chemistry - A European Journal*, 2017.

[321] Matthew K. Nielsen, Derek T. Ahneman, Orestes Riera, and Abigail G. Doyle. Deoxyfluorination with sulfonyl fluorides: Navigating reaction space with machine learning. *Journal of the American Chemical Society*, 140(15):5004–5008, 2018.

[322] Hideto Miyabe, Masafumi Ueda, and Takeaki Naito. N-Sulfonylimines as an excellent acceptor for intermolecular radical reactions. *Chemical Communications*, pages 2059–2060, 2000.

[323] Javier Izquierdo and Miquel A. Pericàs. A Recyclable, Immobilized Analogue of Benzotetramisole for Catalytic Enantioselective Domino Michael Addition/-Cyclization Reactions in Batch and Flow. *ACS Catalysis*, 6(1):348–356, 1 2016.

[324] Louise A. Stubbing, G. Paul Savage, and Margaret A. Brimble. N-alkylsulfonylimines as dipolarophiles in cycloaddition reactions, 1 2013.

[325] Yunfei Luo, Hamish B. Hepburn, Nawasit Chotsaeng, and Hon Wai Lam. Enantioselective rhodium-catalyzed nucleophilic allylation of cyclic imines with allylboron reagents. *Angewandte Chemie - International Edition*, 51(33):8309–8313, 8 2012.

[326] Xiao Feng Xiong, Hang Zhang, Jing Peng, and Ying Chun Chen. Direct asymmetric Michael addition of cyclic N-sulfonylimines to α,β-unsaturated aldehydes. *Chemistry - A European Journal*, 17(8):2358–2360, 2011.

[327] Joydev K. Laha and Krupal P. Jethava. Access to Imidazolidine-Fused Sulfamidates and Sulfamides Bearing a Quaternary Center via 1,3-Dipolar Cycloaddition of Nonstabilized Azomethine Ylides. *Journal of Organic Chemistry*, 82(7):3597–3604, 2017.

[328] Joydev K. Laha, Krupal P. Jethava, K. S.Satyanarayana Tummalapalli, and Sheetal Sharma. Synthesis of Mono-N-sulfonylimidazolidines by a 1,3-Dipolar Cycloaddition Strategy, as an Alternative to Selective N-Sulfonylation, and Their Ring Cleavage To Afford 1,2-Diamines. *European Journal of Organic Chemistry*, 2017(31):4617–4624, 2017.

[329] Pengfei Hu, Jian Hu, Jiajun Jiao, and Xiaofeng Tong. Amine-promoted asymmetric (4+2) annulations for the enantioselective synthesis of tetrahydropyridines: A traceless and recoverable auxiliary strategy. *Angewandte Chemie - International Edition*, 52(20):5319–5322, 2013.

[330] Alberto G. Kravina, Jessada Mahatthananchai, and Jeffrey W. Bode. Enantioselective, NHC-catalyzed annulations of trisubstituted enals and cyclic n-sulfonylimines via α,β-unsaturated acyl azoliums. *Angewandte Chemie - International Edition*, 51(37):9433–9436, 2012.

[331] Xiang Yu Chen, Ruo Chen Lin, and Song Ye. Catalytic [2+2] and [3+2] cycloaddition reactions of allenoates with cyclic ketimines. *Chemical Communications*, 48(9):1317–1319, 2012.

[332] Kim Spielmann, Arie Van Der Lee, Renata Marcia De Figueiredo, and Jean Marc Campagne. Diastereoselective Palladium-Catalyzed (3 + 2)-Cycloadditions from Cyclic Imines and Vinyl Aziridines. *Organic Letters*, 20(5):1444–1447, 2018.

[333] Takahiro Nishimura, Akira Noishiki, Gavin Chit Tsui, and Tamio Hayashi. Asymmetric synthesis of (Triaryl)methylamines by rhodium-catalyzed addition of arylboroxines to cyclic N-sulfonyl ketimines. *Journal of the American Chemical Society*, 134(11):5056–5059, 2012.

[334] Hui Wang, Tao Jiang, and Ming Hua Xu. Simple branched sulfur-olefins as chiral ligands for Rh-catalyzed asymmetric arylation of cyclic ketimines: Highly enantioselective construction of tetrasubstituted carbon stereocenters. *Journal of the American Chemical Society*, 135(3):971–974, 2013.

[335] Yi Li, Yue Na Yu, and Ming Hua Xu. Simple Open-Chain Phosphite-Olefin as Ligand for Rh-Catalyzed Asymmetric Arylation of Cyclic Ketimines: Enantioselective Access to gem-Diaryl α-Amino Acid Derivatives. *ACS Catalysis*, 6(2):661–665, 2016.

[336] Lode De Munck, Alicia Monleón, Carlos Vila, and José R. Pedro. Diarylprolinol as a Ligand for Enantioselective Alkynylation of Cyclic Imines. *Advanced Synthesis and Catalysis*, 359(9):1582–1587, 2017.

[337] Yuan Huang, Rui Zhi Huang, and Yu Zhao. Cobalt-Catalyzed Enantioselective Vinylation of Activated Ketones and Imines. *Journal of the American Chemical Society*, 138(20):6571–6576, 2016.

[338] Mao Quan, Xiaoxiao Wang, Liang Wu, Ilya D. Gridnev, Guoqiang Yang, and Wanbin Zhang. Ni(II)-catalyzed asymmetric alkenylations of ketimines. *Nature Communications*, 9(1):1–11, 2018.

[339] H. Khalilullah, M. J. Ahsan, Md. Hedaitullah, S. Khan, and B. Ahmed. 1,3,4-Oxadiazole: A Biologically Active Scaffold. *Mini-Reviews in Medicinal Chemistry*, 12(8):789–801, 5 2012.

[340] Garima Verma, Mohemmed F. Khan, Wasim Akhtar, Mohammad Mumtaz Alam, Mymoona Akhter, and Mohammad Shaquiquzzaman. A Review Exploring Therapeutic Worth of 1,3,4-Oxadiazole Tailored Compounds. *Mini-Reviews in Medicinal Chemistry*, 19(6):477–509, 3 2019.

[341] Alan R. Katritzky and Charles W. Rees. *Comprehensive Heterocyclic Chemistry*, volume 1-7. Elsevier, 2009.

[342] Hui Zhen Zhang, Zhi Long Zhao, and Cheng He Zhou. Recent advance in oxazole-based medicinal chemistry, 2018.

[343] Kinjal D. Patel, Shraddha M. Prajapati, Shyamali N. Panchal, and Hitesh D. Patel. Review of synthesis of 1,3,4-oxadiazole derivatives. *Synthetic Communications*, 44(13):1859–1875, 7 2014.

[344] Bayard R. Huck, Lisa Kötzner, and Klaus Urbahns. Small Molecules Drive Big Improvements in Immuno-Oncology Therapies, 4 2018.

[345] Dong Jo Chang, Mi Young Jeong, Jiho Song, Chang Yun Jin, Young Ger Suh, Hyun Jung Kim, and Kyung Hoon Min. Discovery of small molecules that enhance astrocyte differentiation in rat fetal neural stem cells. *Bioorganic and Medicinal Chemistry Letters*, 2011.

[346] Benjamin H. Rotstein, Serge Zaretsky, Vishal Rai, and Andrei K. Yudin. Small heterocycles in multicomponent reactions, 2014.

[347] Alexander Dömling, Wei Wang, and Kan Wang. Chemistry and biology of multicomponent reactions, 6 2012.

[348] Cecilia Saiz, Peter Wipf, Eduardo Manta, and Graciela Mahler. Reversible thiazolidine exchange: A new reaction suitable for dynamic combinatorial chemistry. *Organic Letters*, 11(15):3170–3173, 8 2009.

[349] Taber S. Maskrey, Madeline C. Frischling, Mikhaila L. Rice, and Peter Wipf. A Five-Component Biginelli-Diels-Alder Cascade Reaction. *Frontiers in Chemistry*, 6(AUG):376, 8 2018.

[350] Ali Ramazani and Aram Rezaei. Novel One-Pot, Four-Component Condensation Reaction: An Efficient Approach for the Synthesis of 2,5-Disubstituted 1,3,4-Oxadiazole Derivatives by a Ugi-4CR/aza-Wittig Sequence. *Organic Letters*, 12(12):2852–2855, 2010.

[351] Solomon D. Appavoo, Takuya Kaji, John R. Frost, Conor C.G. Scully, and Andrei K. Yudin. Development of Endocyclic Control Elements for Peptide Macrocycles. *Journal of the American Chemical Society*, 140(28):8763–8770, 2018.

[352] John R. Frost, Conor C.G. Scully, and Andrei K. Yudin. Oxadiazole grafts in peptide macrocycles. *Nature Chemistry*, 8(12):1105–1111, 2016.

[353] Renato R. Contreras, Patricio Fuentealba, Marcelo Galván, and Patricia Pérez. A direct evaluation of regional Fukui functions in molecules. *Chemical Physics Letters*, 304(5-6):405–413, 5 1999.

[354] Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Central Science*, 3(12):1237–1245, 2017.

[355] Cyrus Levinthal. Molecular Model-building by Computer. *Scientific American*, 214(6):42–53, 1966.

[356] John C. Kendrew. Three dimensional structure of Globular Proteins. *Reviews of Modern Physics*, 31(1):94–99, 1959.

[357] Mohd Ahmar Rauf, Swaleha Zubair, and Asim Azhar. Ligand docking and binding site analysis with pymol and autodock/vina. *International Journal of Basic and Applied Sciences*, 4(2):168, 3 2015.

[358] Romelia Salomon-Ferrer, David A. Case, and Ross C. Walker. An overview of the Amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 3(2):198–210, 2013.

[359] Markus A. Lill and Matthew L. Danielson. Computer-aided drug design platform using PyMOL. *Journal of Computer-Aided Molecular Design*, 25(1):13–19, 2011.

[360] Andreas Moll, Andreas Hildebrandt, Hans Peter Lenhof, and Oliver Kohlbacher. BALLView: An object-oriented molecular visualization and modeling framework. *Journal of Computer-Aided Molecular Design*, 19(11):791–800, 2005.

[361] Alexander S. Rose and Peter W. Hildebrand. NGL Viewer: A web application for molecular visualization. *Nucleic Acids Research*, 43(W1):W576–W579, 2015.

[362] John L. Moreland, Apostol Gramada, Oleksandr V. Buzko, Qing Zhang, and Philip E. Bourne. The Molecular Biology Toolkit (MBT): A modular platform fro developing molecular visualization applications. *BMC Bioinformatics*, 6:1–7, 2005.

[363] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. Mouse, tactile, and tangible input for 3D manipulation. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 2017-May, pages 4727–4740, 2017.

[364] Hsin Kai Wu, Joseph S. Krajcik, and Elliot Soloway. Promoting understanding of chemical representations: Students' use of a visualization tool in the classroom. *Journal of Research in Science Teaching*, 38(7):821–842, 2001.

[365] Andrew Woods. Stereoscopic Presentations – Taking the Difficulty out of 3D. *Computer*, pages 1–6, 2000.

[366] George M. Bodner and Daniel S. Domin. Mental Models: The Role of Representations in Problem Solving in Chemistry. *University Chemistry Education*, 4(1):24–30, 2000.

[367] Andrew Johnson, Thomas Moher, Stellan Ohlsson, and Jason Leigh. Exploring multiple representations in elementary school science education. In *Proceedings IEEE Virtual Reality 2001*, pages 201–208. IEEE Comput. Soc, 2001.

[368] Abraham Anderson and Zhiping Weng. VRDD: Applying virtual reality visualization to protein docking and design. *Journal of Molecular Graphics and Modelling*, 17(3-4):180–186, 1999.

[369] Jorge Trindade, Carlos Fiolhais, and Leandro Almeida. Science learning in virtual environments: A descriptive study. *British Journal of Educational Technology*, 33(4):471–488, 2002.

[370] Stefan Birmanns and Willy Wriggers. Interactive fitting augmented by force-feedback and virtual reality. *Journal of Structural Biology*, 144(1-2):123–131, 2003.

[371] Aidin R. Balo, Merry Wang, and Oliver P. Ernst. Accessible virtual reality of biomolecular structural models using the Autodesk Molecule Viewer. *Nature Methods*, 14(12):1122–1123, 2017.

[372] Zhihan Lv, Alex Tek, Franck Da Silva, Charly Empereur-mot, Matthieu Chavent, and Marc Baaden. Game On, Science - How Video Game Technology May Help Biologists Tackle Visualization Challenges. *PLoS ONE*, 8(3), 2013.

[373] Thomas D. Goddard, Alan A. Brilliant, Thomas L. Skillman, Steven Vergenz, James Tyrwhitt-Drake, Elaine C. Meng, and Thomas E. Ferrin. Molecular Visualization on the Holodeck, 2018.

[374] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work? - A literature review of empirical studies on gamification. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 3025–3034. IEEE, 2014.

[375] Deise Albertazzi, Marcelo Gitirana Gomes Ferreira, and Fernando Antônio Forcellini. A Wide View on Gamification. *Technology, Knowledge and Learning*, 24(2):191–202, 2019.

[376] Damien Djaouti, Julian Alvarez, and Jean-pierre Jessel. Classifying Serious Games. In *Handbook of research on improving learning and motivation through educational games: Multidisciplinary approaches*, pages 118–136. IGI Global, 2011.

[377] Marc Prensky. Digital game-based learning. *Computers in Entertainment*, 1(1):21, 2003.

[378] Thibault Carron, Jean Charles Marty, and Jean Mathias Heraud. Teaching with game-based learning management systems: Exploring a pedagogical dungeon. *Simulation and Gaming*, 39(3):353–378, 2008.

[379] Hyungsung Park. Relationship between Motivation and Student ' s Activity on Educational Game. *Journal of Grid and Distributed Computing*, 5(1):101–114, 2012.

[380] Martin Ebner and Andreas Holzinger. Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers and Education*, 49(3):873–890, 2007.

[381] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[382] Firas Khatib, Frank Dimaio, Seth Cooper, MacIej Kazmierczyk, Miroslaw Gilski, Szymon Krzywda, Helena Zabranska, Iva Pichova, James Thompson, Zoran Popović, Mariusz Jaskolski, and David Baker. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature Structural and Molecular Biology*, 18(10):1175–1177, 2010.

[383] Grigore C. Burdea. Haptic Feedback for Virtual Reality. *Virtual Reality and Prototyping Workshop*, 2(June):17–29, 1999.

[384] Shanhong Liu. Forecast for the number of active virtual reality users worldwide from 2014 to 2018, 2017.

[385] Magnus Norrby, Christoph Grebner, Joakim Eriksson, and Jonas Boström. Molecular Rift: Virtual Reality for Drug Designers. *Journal of Chemical Information and Modeling*, 55(11):2475–2484, 2015.

[386] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm, 1987.

[387] Jay W Ponder and David A Case. Force Fields for Protein Simulations. In *Advances in Protein Chemistry*, volume 66, pages 27–85. Elsevier, 2003.

[388] Francesco Di Natale, Harsh Bhatia, Timothy S Carpenter, Chris Neale, Sara Kokkila Schumacher, Tomas Oppelstrup, Liam Stanton, Xiaohua Zhang, Shiv Sundram, Thomas R. W. Scogland, Gautham Dharuman, Michael P Surh, Yue Yang, Claudia Misale, Lars Schneidenbach, Carlos Costa, Changhoan Kim,

Bruce D'Amora, Sandrasegaram Gnanakaran, Dwight V Nissley, Fred Streitz, Felice C Lightstone, Peer-Timo Bremer, James N Glosli, and Helgi I Ingólfsson. A massively parallel infrastructure for adaptive multiscale simulations. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, New York, NY, USA, 11 2019. ACM.

[389] J. Jiménez, S. Doerr, G. Martínez-Rosell, A. S. Rose, and G. De Fabritiis. DeepSite: Protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, 2017.

[390] José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, 58(2), 2018.

[391] Miha Skalic, Davide Sabbadin, Boris Sattarov, Simone Sciabola, and Gianni De Fabritiis. From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design. *Molecular Pharmaceutics*, 16(10):4282–4291, 8 2019.

[392] J S Smith, O Isayev, and A E Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost †. *Chemical Science*, 8, 2017.

[393] Justin S Smith, Olexandr Isayev, Adrian E Roitberg, and Sample Characteristic. Data Descriptor: ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules Background & Summary. *Scientific Data*, 2017.

A. INTRODUCTION TO SUPERVISED MACHINE LEARNING

Since machine learning (ML) plays a pivotal role throughout the works presented in this document, it is important to outline the exact definition used in my works. Additionally, one must provide a succinct mathematical foundation written with a physical chemist in mind. Herein, a description of supervised machine learning is given for supervised machine learning. Or, more simply, a form of ML where there is an explicit set of input features and a desire to predict an outcome.

A.1 Expert-based solutions

To start, we frame any given prediction problem as simply mapping a 'feature space' ($\vec{x}$) to a desired outcome which we wish to model (y). Traditionally, one attempts to find a relationship such that $y = f(\vec{x})$ where the function f can be arbitrary as long as it can predict y given x. A simple example is an application of the Beer-Lambert law where one can predict the absorbance (A, also our y) of a solution given the concentration of a species C, its molar extinction coefficient ϵ, and the length of the container holding the solution b. Here, C, ϵ, and b are all input features (our $\vec{x}$). We know from the Beer-Lambert law that:

$$A = y = f(b, C, \epsilon) = b \cdot C \cdot \epsilon$$

This is an example of an expert-based method where the relationship between y and x is known and one can program a simple algorithm to compute A given the

proper inputs. While a simple equation works great for this problem, such a simple solution does not typically work for more complex problems which may depend on hundreds to thousands of features. Additionally, it may be difficult to impossible to create an exact mathematical expression which relate these variables. To address this, mathematicians and computer scientists, among others, have created powerful generic algorithms which attempt to find the function f from just y and $\vec{x}$.

A.2 Defining a machine learning problem

As opposed to an expert–based model, supervised ML methods attempt to define the function $f(\vec{x})$ using a set of known outcomes (y) with a set of known features ($\vec{x}$). Before this can be done, the nature of the outcome variable (y) must be determined. The first question that must be asked is whether y is numeric (e.g. absorbance, retention time, cLogP) or a category (e.g. its functional group classification, does it react with a set of reagents). If y is the former, then the problem is defined as a regression problem. If y is the latter, then it is a classification problem. Additionally, if y can only be one of two categories (e.g. resistant or sensitive to chemotherapy, reacts or does not react), then the problem is that of binary classification. Once the nature of y is determined, one must select an appropriate loss function (ℓ) which fits the problem one wishes to solve.

A.2.1 Loss functions for regression

For the case of regression, several loss functions (ℓ) are available, which will be described in no particular order. The true measured value that we wish to predict is referred to as the ground truth ($\hat{y}$). The Mean Absolute Error (MAE) is one such ℓ that is commonly used and is described below.

$$MAE(y, \hat{y}) = \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{n}$$

A popular alternative is that of Root Mean-Squared Deviation (MSE), which is defined below. The major difference between MAE and MSE is that MSE places a greater penalty on values farther from the ground truth than MAE due to the quadratic growth in the numerator of the sum. Related to MSE is the root mean-squared deviation ($RMSD$), which is the square root of the MSE.

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

$$RMSD(y, \hat{y}) = \sqrt{\sum_{i=i}^{n} \frac{(y_i - \hat{y}_i)^2}{n}}$$

The $RMSD$ is similar to the Cartesian distance (D_w) which is the following:

$$D_w(y, \hat{y}) = \sqrt{\sum_{i=i}^{n} (y_i - \hat{y}_i)^2}$$

An alternative to these measures of error, one can measure the dissimilarity between the prediction and ground truth. A common way of commuting dissimilarity is to calculate the similarity of the prediction and ground truth and subtracting the value from 1. An example of such a similarity measure is the cosine similarity:

$$cos\theta(y, \hat{y}) = \frac{y \cdot \hat{y}}{||y||\,||\hat{y}||} = \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{n} y_i}\sqrt{\sum_{i=1}^{n} \hat{y}_i}}$$

This measure is similar to that of the Pearson correlation which is defined below and is used to measure the linear correlation of two values.

$$r(y, \hat{y}) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})}}$$

Unlike MAE, MSE, $RMSD$, D_w, the $cos\theta$ and r similarity metrics are required to be between $[-1, 1]$. We can build similarity metrics from these distance functions to create custom and more complex similarity functions. These functions are typically used to compare the outputs of two different networks (as opposed to the ground truth directly). One such example is a Siamese neural network. These applications are important for one-shot learning, which is mentioned in the outlook chapter. For now, we will give one example of such a loss function, called the contrastive loss function defined below. This function takes two arguments, the distance between two vectors (d, as calculated by a distance metric) and a label which designates whether or not the two vectors describe the same outcome (s). It also takes a parameter called the margin (m) to control the penalty assigned to cases where y is zero (dissimilarity).

$$\ell_C(d, s; m) = s \cdot d^2 + (1 - s)max(m - d, 0)^2$$

This function is interesting as it bridges regressive loss functions through the d argument and classification loss functions through the s argument.

A.2.2 Loss functions for classification

Shannon entropy and entropic models

For classification, one typically uses a metric derived from Shannon entropy (H, which is eta), which is defined below for an event X. This event is a discrete value (i.e. a category) can be any value in the set $\{x_0, x_1, x_2, ..., x_n\}$. This yields the following definition for H:

$$H(X) = -\sum_{i=0}^{n} P(X_i) \ln P(X_i)$$

This is in turn named for the Boltzmann-H theorem which describes the kinetic energy (E) of a molecule at time t as this integral has a similar concept where a function is multiplied by its logarithm.

$$H(t) = \int_0^t f(E,t) \ln\left(\frac{f(E,t)}{\sqrt{E}} - 1\right) dE$$

One can also define the Shannon entropy for a set given a previous event (Y) where $p(x_i, y_i)$ is probability that $X = x_i$ and $Y = y_i$.

$$H(X|Y) = -\sum_{i,j}^{n} p(x_i, y_j) \frac{\ln p(x_i, y_j)}{p(x_i)}$$

These definitions allow us to define the information gain of a given feature f_i from the feature set F:

$$IG(X, f) = H(X) - H(X|f)$$

The maximization of IG is typically performed directly by a machine learning model referred to as a decision tree (DT). This model depends on this metric explicitly and attempts to maximize it for a given feature set $\vec{X}$. The major disadvantage for the use of IG is that it only considers the binary condition that feature f_i is true or false and does not treat f_i as numeric. This causes DTs to typically over train to a given training set, but is typically useful for creating models which are easily understood by humans (see chapters 5 and 6).

Random forest (RF) attempts to overcome these over–training issues by training multiple DTs using a subset of the data and then uses the individual DT models to

predict the result through a process called 'voting'. Here, if a majority of the DT models agree on a given classification, then the RF model predicts that class for the input. During training, the RF algorithm selects subsets of the entire training to use for training a given DT in a process called bagging. Bagging results in multiple trees trained on small portions of the data. While each tree may be over–trained, the ensemble of all trees typically is not over–trained. RF models are used throughout chemistry and data science as a quick and easy *go to* solution. In recent years, eXtreme Gradient Boosting (XBG) has begun to replace RF in this regard, but a detailed difference between these methods is beyond this simple introduction.

Application of entropy to binary classification given a probability produced by a model

A major alternative to this loss function is the cross entropy loss function which is defined for a ground truth of 0 or 1 (the binary classifier case, referred to as $\hat{y}$) and the probability that the ground truth is 1 (called p). Typically, the probability p is produced by the classifier and is similar in concept to the value y produced by a regression model, a relationship that will be explored in the later section on logistic regression. The cross entropy is defined as:

$$CE(p, \hat{y}) = -\hat{y} \ln(p) - (1 - \hat{y}) \ln(1 - p)$$

This formulation is reminiscent of contrastive loss as noted in the previous section (note that historically CE came first). When used with more than two conditions (the multi–class case), the CE is written as the following where n is the number of classes:

$$CE(\vec{p}, \hat{y}) = -\sum_{i=1}^{n} \hat{y}_i \ln p_i$$

If the ground truth can only be a single category (e.g. is an atom an element as the atom can only be carbon or sulfur or nitrogen, etc), then this function is typically combined with the softmax function (SM) to form the categorical cross-entropy function (CCE). Softmax can be thought of as a way to normalize the vector $\vec{p}$ in a manner that considers all values of the vector. This idea is useful as the single–label case (but multi–class problem) only has one value in the ground truth which is 1 and the rest are zero. These functions are as follows where p_i is the value in the prediction vector which corresponds to the non-zero value in the ground truth vector.

$$SM(p_i) = \frac{e^{p_i}}{\sum_j^n e^{p_j}}$$

$$CCE(\vec{p}, \hat{y}) = CE(SM(\vec{p}), \hat{y}) = -\ln\left(\frac{e^{p_p}}{\sum_j^n e^{p_j}}\right)$$

While categorical cross entropy can be used for multi–class models, it is typically not used for the multi–label case where a single observation could be in multiple classes (e.g. all the functional groups in a single molecule). An alternative to CCE is binary cross-entropy (BCE) and this type of loss is created through the combination of the sigmoid function (σ) and cross-entropy. The use of the sigmoid function is logical because it normalizes each value of a vector independently and the ground truth vector may have multiple non-zero values. These functions are defined below:

$$\sigma(p_i) = \frac{1}{1 + e^{-p_i}}$$

$$BCE(\vec{p}, \hat{y}) = CE(\sigma(\vec{p}), \hat{y})$$

This has the effect of creating a binary classifier for each possible case in the ground truth vector, making it useful for multi–label classification.

Evaluating the performance of a classifier

Unlike a regression model, a classifier is typically not measured by its loss function (which is in turn only used for training). Instead, it is measured by other metrics based on criterion based solely on whether the classifier is correct or incorrect, thereby ignoring the actual prediction value. If the classifier is correct, then the prediction is 'true', otherwise it is 'false'. If the ground truth is the positive case, then the result is 'positive', otherwise it is 'negative'. This creates four cases defined in the table below.

Ground Truth	**Prediction**	**Abbreviation**	**Full name**
Positive	Positive	TP	True positive
Positive	Negative	FP	False positive
Negative	Positive	FN	False negative
Negative	Negative	TN	True negative

These can be used to define the following:

Recall/sensitivity/true positive rate (RE)

$$\frac{TP}{TP + FN}$$

Selectivity/specificity/true negative rate

$$\frac{TN}{TN + FP}$$

Precision/positive prediction value (PR)

$$\frac{TP}{TP + FP}$$

Negative prediction value

$$\frac{TN}{TN + FN}$$

Accuracy (AC)

$$\frac{TN + TP}{TN + TP + FN + FP}$$

F1 score

$$\frac{2 \cdot PR \cdot RE}{PR + RE}$$

Cohen κ

$$1 - \frac{1 - AC}{1 - \frac{1}{N^2} \sum_k n_{k1} n_{k2}}$$

The concept of $F1$ score and κ are referenced many times in this document, so their formal definition here is important, especially with regards to Chapters 2, 4, 5, and 6. These values allow one to determine how well a classifier is able to model a given problem.

A.3 Understanding logistic regression as a sample classifier

Logistic regression (LR) is an important classifier for the understanding of how other classifiers such as neural networks are built. It is built on the simple premise that the input features ($\vec{x}$) have a linear relationship to the probability of the outcome (y) being positive or negative. We call the result of this linear transformation the discriminator (D) and it is defined as the following where $\vec{W}$ is a weighting vector and β is a scalar bias term:

$$D = \vec{x} \cdot \vec{W} + \beta$$

If $D > 0$ then the prediction for $\vec{x}$ is positive, otherwise it the prediction is negative. The crucial next step for developing a logistic classifier is to assume that the probability of the model being positive ($p(\vec{x})$) follows a logistic curve. This yields the following relationship:

$$\ln\left(\frac{p(\vec{x})}{1-p(\vec{x})}\right) = \vec{x} \cdot \vec{W} + \beta$$

Rearranging yields:

$$p(\vec{x}; \vec{W}; \beta) = \frac{1}{1 + e^{-\beta + \vec{W} \cdot \vec{x}}}$$

If one sets the bias term to zero, this equation becomes reminiscent of the Boltzmann distribution function for a system with two states. Here, the positive state is the higher energy state and the negative state is the lower energy term. The weighting vector is similar to the energy difference between the two states and the feature space is similar to the temperature. This analogy to fundamentals of physical chemistry should provide a clear picture of what machine learning attempts to accomplish and how it goes about solving the classification problem from a similar starting point.

Given the probability of an observation being active given its feature space, one must now determine the weighting vector and bias term. To do so, we start with the likelihood function calculated for all observations ($\vec{X}$ with an individual feature component x_i:

$$\ell(\vec{X}, \hat{y}; \vec{W}, \beta) = \prod_{i=0}^{n} p(x_i; \vec{W}, \beta)^{\hat{y}_i} (1 - p(x_i; \vec{W}, \beta)^{1-\hat{y}_i})$$

Then, the negative logarithm of this function is taken:

$$-\ln \ell(\vec{X}, \hat{y}; \vec{W}, \beta) = \sum_{i=0}^{n} \left[\hat{y}_i \ln p(x_i; \vec{W}, \beta) + (1 - \hat{y}_i) \ln(1 - p(x_i; \vec{W}, \beta)) \right]$$

This final function is equivalent to the cross-entropy loss function (CE) defined previously and therefore forms the basis of classification methodology for most neural networks. Since we wish to minimize this loss function, we will take its derivative for a single weight in the weight vector $\vec{W}$ to yield the following:

$$\frac{\partial CE}{\partial w_j} = -\sum_{i=1}^{n} \left(\hat{y}_i - \ln p(x_i, \vec{W}, \beta) \right) x_{ij}$$

Unfortunately, it is not possible to solve the case where the above equation becomes zero, so numerical methods are used to find the values of $\vec{W}$ such that the CE function is minimized.

A.3.1 Regularization of machine learning models

In the previous section, the loss function is derived solely from the weights and biases in the model. Unfortunately, this formulation of the loss function can lead to optimization scenarios where a single weight dominates the expression, which prevents the model from performing well on data not used for training the model. To address this, one can add the values of the weights directly to the loss function. This technique is call regularization and it typically comes in two forms: Least Absolute Selection and Shrinkage Operator (LASSO) and Tikhonov/Ridge. They are also referred to as L1 and L2 regularization, respectively, due to the value of the exponent used in their expressions (given below). They are scaled by a factor called λ which adjusts the amount the network is penalized by the regularization method.

$$L1 \longrightarrow \lambda \sum_{i=1} |W_i|$$

$$L2 \longrightarrow \lambda \sum_{i=1} W_i^2$$

A.3.2 Expanding logistic regression to the multi–layer perceptron

With the mathematical formalism for logistic regression (LR) laid out, the next step in building up a machine learning model is to increase the number of model 'layers' to create a 'multi–layer perceptron' (MLP) network. The LR model can be considered to be an MLP model with only 2 layers, an input (our original $\vec{X}$) and an output layer (the scalar p). The relationship is the multiplication of $\vec{X}$ and the weighting vector $\vec{W}$. When one introduces an additional layer between $\vec{X}$ and p (what is referred to as a hidden layer), the weighting vector becomes the weighting *matrix* $\mathbf{W}^{(l)}$ where l denotes the location of the transformation in the network. If an MLP has 1 hidden layer, there will be two weighting matrices, $\mathbf{W}^{(1)}_{MN}$ with rows M and columns N where N is equal to the length of $\vec{X}$, and $\mathbf{W}^{(2)}_{PN}$ where P is equal to the length of the output layer $\hat{y}$. Additionally, there will be a biasing vector $\beta^{(i)}$ which is added to the product of $\mathbf{W}^{(1)}_{MN}$ and $\hat{X}$ to form an intermediate vector $\vec{h^{(1)}}$. A similar procedure can be done for the final output layer. The element–wise (for element j) calculation of $\vec{h^{(l)}}$ is given below. This value is referred to as an activation.

$$h_j^{(l)} = \sum_{k=1} \mathbf{w}_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}$$

Since an intermediate vector $\vec{h^{(l)}}$ is introduced, additional transformations can be introduced to improve the MLP. One such transformation is called batch normalization. This procedure normalizes these layers using the other values produced in subset

of the training set (also referred to as a batch). This technique, developed by Sergey Ioffe and Christian Szegedy at Google in 2015, is designed to reduce the variations in the activation values during training (called the covarience shift) to allow for better generalization and decreased training times. To define batch normalization, let the size of the batch be the integer value m and the mean and variance of an activation in a batch to be defined as follows:

$$\mu_{j,B}^{(l)}\left(\left[h_{j,1}^{(l)}, h_{j,2}^{(l)}, h_{j,3}^{(l)}, ..., h_{j,m}^{(l)}\right]\right) = \frac{1}{m}\sum_{i=1}^{m} h_{j,i}^{(l)}$$

$$\sigma_{j,B}^{(l)^2}\left(\left[h_{j,1}^{(l)}, h_{j,2}^{(l)}, h_{j,3}^{(l)}, ..., h_{j,m}^{(l)}\right]\right) = \frac{1}{m}\sum_{i=1}^{m}(h_{j,i}^{(l)} - \mu_B^{(l)})^2$$

Now, one can normalize the original activation similar to how one would calculate a z–score. Here, an additional scaling parameter γ and shifting parameter β are introduced are learned along with the $\mathbf{W}^{(l)}$ matrix. The final expression is written as the following:

$$z_j^{(l)} = \gamma_k^{(l)}\left(\frac{h_j^{(l)} - \mu_{j,B}^{(l)}}{\sigma_{j,B}^{(l)^2}}\right) + \beta_k^{(l)}$$

In addition to batch normalization, scalar operations can be introduced per layer and are referred to as *activation* functions, written here as $h(x)$, to produce an vector $\vec{a}^{(l)}$. The purpose of an activation function is to add non-linearity to the MLP model and several functions can be selected to fill the role of $h(X)$, such as the sigmoidal function, the hyperbolic tangent, or the rectified linear unit (ReLU) function. The latter is defined as $ReLU(x) = max(0, x)$ and is a popular choice given its computational efficiency. The scalar $h(x)$ function is calculated for element j as:

$$a_j^{(l)} = h(Z_j^{(l)})$$

Finally, to increase the ability of the model to generalize to data not included during training, random activations are set to zero with a preset probability. This feature is called 'dropout' and is only applied during the training of the network. It cannot be used on the final output layer as this would randomly force the model to incorrect in a manner that would not be beneficial to training.

With the additional features of the MLP defined, the derivative of the network can be calculated. To do so, we define the error of a single activation as the following:

$$\delta_j^{(l)} = \frac{\partial \ell}{\partial a_j^{(l)}}$$

For the final layer of the model L, we can compute this value via the chain rule:

$$\delta_j^{(L)} = \frac{\delta C}{\delta a_j^{(L)}} \sigma\prime(z_j^{(L)})$$

This expression can be rewritten as the following for a matrix operation:

$$\delta^L = \nabla_a C \odot \sigma\prime(z^{(L)})$$

Once the derivative of the final layer is obtained, the derivative of the penultimate layer can be calculated as follows:

$$\delta^l = \left(\left(\mathbf{W^{l+1}} \right)^{\mathbf{T}} \delta^{l+1} \right) \odot \sigma\prime(z^l)$$

One can continue to 'backpropagate' this derivative until the derivative of the first layer is calculated. Now that $\delta_j^{(l)}$ can be calculated for all layers, the derivative of the total cost function with respect to a bias term can be written as

$$\frac{\delta C}{\delta b_j^{(l)}} = \delta_j^{(l)}$$

Additionally, the following for the weighting terms:

$$\frac{\delta C}{\delta \mathbf{W}_{jk}^{(l)}} = a_k^{(l-1)} \delta_j^{(l)}$$

Note that the derivation of the derivatives for γ and β are left out for brevity.

A.3.3 Applications to graphs

With the mathematical background described for an MLP network, the foundations for Graph Neural Networks (GNNs) can be discussed. First, let a graph G be defined as a set of nodes and edges. Since we are interested in molecules, one can think of these nodes as atoms and the edges as bonds. In a GNN, nodes can send 'messages' to their neighbors in each layer of the network. At the first, or zeroth layer ($l = 0$), each node simply passes a message to itself which contains its element, hybridization, charge, and other useful information. The next layer, $\ell = 1$, is calculated by adding in the messages from the N neighboring atoms. This process is repeated for all the layers in the network and can be expressed using the following equation for atom i with neighbors j:

$$X_i^{(\ell+1)} = X^{(l)} + \sum_{j=1}^{N} f\left(X_i^{(l)}\right)$$

For the final layer (as determined by the hyper–parameter called radius), the final $\vec{y}$ output is calculated from all the M atoms using the following equation:

$$\vec{y} = \sum_{i=1}^{M} X_i^{(L)}$$

As for the MLP network, the errors for each weight in the network can be obtained using the chain rule (backpropogation). This description is for a simple GNN, but various graph filters can be used as well to create a graph convolutional neural network

(GCN). A great application of the GCN network is that the topology of the graph does not change during training, allowing one to recolor (re-weight) the graph and obtain node properties. However, such a discussion is beyond the work presented herein.

A.4 Evaluating how well a machine learning model performs

A.4.1 The training, validation, and test sets and cross–validation

When evaluating the performance of a model, one needs to split the known data into three different steps: training (observations used to explicitly train the model), validation (observations used to evaluate model performance given a set of hyper–parameters and tune them accordingly), and testing (observations never used to train or tune the model and used as a final test for the model). Typically, the test set observations are known before training begins, however, in the works presented in this work, the test set used is experimentally measured *after* the model has been trained (so they are referred to as prospective test sets). In many cases, the validation of the model (done with a training set and validation set) is done multiple times and the training and validation sets are varied in a method called cross-validation. The simplest, and most exhaustive, methodology is referred to as Leave-One Out Cross-Validation (LOOCV). In this method, the validation set is a single observation and the training set is the remaining observations. The model is trained on this training set and evaluated on the single validation observation. Then, the validation observation is swapped with an observation in the training set and the process is repeated until all points have been used as the validation observation. Various evaluation criteria (F1 score, Cohen Kappa, etc) are calculated on the evaluated results. This validation methodology is expensive as the number of models created is equal to the number of observations used for the training and validation sets. An alternative to LOOCV is k-

fold validation where the training data is partitioned into 'k' different folds (typically without duplication) and the model is trained on $(k-1)$ folds and evaluated on the held–out fold (the validation set in this case). This process is repeated until all folds have been used as the validation set. In both LOOCV and k-fold validation, a final model is trained on all the available data and this final model is evaluated using the test set (which has been held out during the entire procedure).

A.4.2 Leave One Out Testing (LOOT)

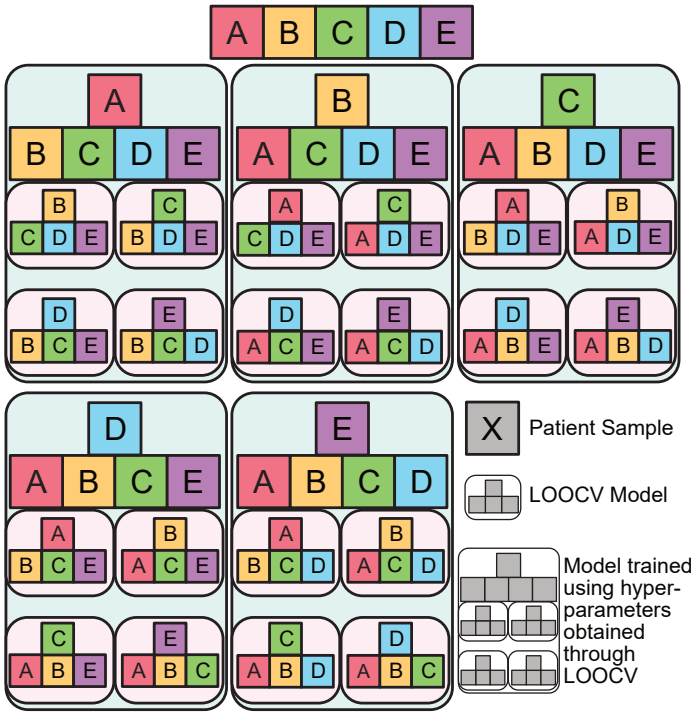


Figure A.1. Visual representation of Leave One Out Testing (LOOT) for five observations. Here, each observation is removed, and the remaining four observations are used to create a hyper-trained model through Leave One Out Cross-Validation (LOOCV).

Traditionally, evaluating a machine learning model requires one to partition their data into three parts: a training set, a validation set, and a testing set. The training set is directly introduced into the machine learning model and the weights and other

parameters of the model are identified from this set. This model is then applied to predict the results of the validation set and the model is allowed to be retrained multiple times using different training parameters (called hyper parameters) to optimize its performance on this set. Once an optimal set of hyperparameters has been found, a final model is created using a combination of the training and validation sets. The final model is then used to predict the results of the test set to obtain final statistics for the performance of the model. This scheme is typically used when a large amount of data is available to train and evaluate the model but is too stringent and open to bias for use in cases where a small amount of data is available. To address this, we have created a novel evaluation paradigm called Leave One Out Testing (LOOT). LOOT is similar in spirit to Leave One Out Cross-Validation (LOOCV) where a single observation is held out and the remaining observations are used to validate a potential model through LOOCV or other cross-validation methodology. The cross-validated model (hyper trained) obtained from the remaining observations is then used to predict the held-out member. This process is then repeated for all observations, allowing statistics to be calculated for the entire dataset that includes the effects of hyperparameter optimization. A visual diagram of this process for 5 observations is given in the above Figure where A, B, C, D, and E are the observations.

B. ADDITIONAL DATA FOR CHAPTER 1

Table B.1.: Indication ranks for mental health indications calculated for all top selections.

MESH	Drug count	Top10	Top25	Top40	Top100
D016574	1	100.00	100.00	100.00	67.68
D020178	1	100.00	100.00	100.00	67.68
D020179	1	100.00	100.00	100.00	67.68
D056912	2	83.33	75.00	73.08	68.18
D001039	1	44.44	58.33	69.23	56.57
D009290	7	40.74	38.89	42.31	44.11
D020186	3	40.74	33.33	28.21	24.24
D001308	1	33.33	54.17	56.41	57.58
D012148	10	33.33	39.58	40.00	35.96
D000856	3	29.63	25.00	23.93	25.93
D002385	12	25.93	23.61	24.57	25.17
D004831	12	25.56	25.83	23.85	21.72
D013981	5	25.00	16.67	17.31	16.16
D010842	4	22.22	22.92	22.44	22.73
D020922	1	22.22	8.33	5.13	4.04
D019958	5	19.44	21.88	30.13	29.04
D001289	22	17.28	16.67	17.66	19.14
D052018	4	16.67	17.71	21.15	25.00
D019263	6	16.67	15.28	14.10	16.16
D007172	26	16.34	14.46	15.69	16.64
D009771	13	16.16	15.15	15.62	19.93
D019964	14	14.44	12.92	12.56	17.27
D000647	44	14.38	16.05	17.72	19.55
D001321	32	12.26	13.65	15.47	17.55
D010554	30	11.56	13.50	13.64	14.75
D019973	3	11.11	20.83	23.08	17.68
D002658	4	11.11	13.54	16.03	21.21
D003072	26	11.11	13.29	14.04	15.34
D000379	5	11.11	12.50	12.31	14.14

continued on next page

Table B.1.: *continued*

MESH	Drug count	Top10	Top25	Top40	Top100
D020774	1	11.11	12.50	10.26	11.11
D001068	3	11.11	6.94	10.26	15.49
D003865	43	10.84	13.72	14.95	16.21
D007859	26	10.58	13.10	14.41	14.57
D001008	46	10.56	12.19	14.10	15.73
D019969	27	10.26	10.58	10.85	11.97
D019955	15	10.19	10.42	11.97	12.96
D019970	75	10.14	13.16	13.71	14.27
D016584	19	9.80	9.56	11.16	16.22
D003693	8	9.72	13.02	16.35	17.80
D012640	94	8.58	9.76	10.91	12.71
D017029	14	8.33	10.07	9.62	11.62
D002659	5	8.33	7.29	12.82	15.40
D004827	38	7.94	9.05	9.52	10.62
D001714	42	7.94	7.74	8.21	10.42
D000341	9	7.94	7.74	8.06	8.08
D019956	16	7.64	9.11	8.81	10.16
D019966	11	7.41	6.48	7.98	8.98
D019305	4	7.41	4.17	5.13	6.40
D013226	29	7.25	8.15	8.47	9.27
D003866	65	6.97	7.35	8.55	10.58
D006556	11	6.67	11.67	12.05	12.53
D013375	48	6.61	8.63	9.52	11.52
D000430	8	6.35	8.33	10.99	9.38
D005879	15	5.93	9.17	9.57	13.13
D020190	3	5.56	12.50	15.38	30.30
D015140	2	5.56	10.42	15.38	28.28
D013064	2	5.56	10.42	15.38	18.69
D004414	2	5.56	10.42	6.41	19.19
D000544	32	5.56	9.46	12.43	15.31
D014029	10	5.56	7.29	7.69	13.76
D003704	13	5.56	5.83	7.18	12.53
D007319	27	5.09	8.16	10.47	9.05
D002653	14	5.05	4.55	7.23	9.46
D009293	10	4.94	9.26	11.97	14.14
D003244	9	4.94	7.87	14.25	16.84
D004830	20	4.86	5.73	5.61	6.57
D004832	12	4.63	6.94	7.91	10.94
D011618	47	4.50	4.62	6.58	8.54
D011605	14	4.44	8.75	11.54	14.14

continued on next page

Table B.1.: *continued*

MESH	Drug count	Top10	Top25	Top40	Top100
D012559	65	4.00	4.75	5.33	6.26
D010698	7	3.70	5.56	5.13	4.71
D004833	11	3.70	4.63	4.84	7.86
D000435	3	3.70	4.17	2.56	3.37
D012893	14	3.42	3.85	6.11	8.00
D003294	8	3.17	1.79	2.20	4.47
D004829	7	3.17	1.79	2.20	3.03
D020018	11	3.03	4.17	3.73	7.35
D020324	4	2.78	10.42	16.03	19.44
D013313	21	2.78	4.17	5.77	7.89
D006970	5	2.22	6.67	8.72	12.32
D004828	19	2.08	3.39	3.21	5.30
D012563	22	1.75	1.75	2.43	3.72
D000437	17	1.48	3.89	4.27	5.59
D006816	10	1.11	4.58	6.67	12.63
D005715	1	0.00	12.50	15.38	40.40
D009497	3	0.00	8.33	14.53	15.49
D012734	1	0.00	8.33	5.13	2.02
D015161	1	0.00	4.17	7.69	10.10
D019052	1	0.00	4.17	7.69	4.04
D020195	2	0.00	4.17	2.56	3.03
D009021	5	0.00	3.13	1.92	2.27
D006998	2	0.00	2.08	3.85	6.06
D013036	8	0.00	1.79	1.47	1.30
D007174	8	0.00	1.39	5.56	8.92
D004775	7	0.00	1.39	1.28	5.72
D053206	4	0.00	1.04	5.13	8.08
D020270	4	0.00	1.04	2.56	6.57
D019957	1	0.00	0.00	10.26	17.17
D001883	4	0.00	0.00	2.56	3.03
D011604	1	0.00	0.00	2.56	2.02
D014256	1	0.00	0.00	2.56	2.02
D012560	5	0.00	0.00	1.92	2.53
D020187	2	0.00	0.00	1.28	2.53
D005329	1	0.00	0.00	0.00	3.03
D010262	3	0.00	0.00	0.00	2.53
D012561	2	0.00	0.00	0.00	2.53
D020961	2	0.00	0.00	0.00	2.53
D012562	2	0.00	0.00	0.00	2.02
D008607	2	0.00	0.00	0.00	1.52

continued on next page

Table B.1.: *continued*

MESH	Drug count	Top10	Top25	Top40	Top100
D020191	3	0.00	0.00	0.00	1.52
D020817	2	0.00	0.00	0.00	1.52
D019967	4	0.00	0.00	0.00	1.35
D003130	1	0.00	0.00	0.00	1.01
D012569	2	0.00	0.00	0.00	1.01
D012892	2	0.00	0.00	0.00	1.01
D014899	1	0.00	0.00	0.00	1.01
D057180	2	0.00	0.00	0.00	1.01
D009357	2	0.00	0.00	0.00	0.51

Compound	Category	Top 10	Top 25	Top 40	Top 100
2-ethylamino-1-(3,4-methylenedioxyphenyl)pentane	Phenethylamine	100.	100	13.3	14.7
3,4-methylenedioxy-n-isopropylamphetamine	Amphetamine	100.	100	6.8	9.0
3,4-methylenedioxy-n-propargylamphetamine	Amphetamine	100.	100	3.7	7.1
pentedrone	Cathinone	100.	57.1	21.9	24.4
cloforex	Amphetamine	100.	38.1	29.6	14.9
methylenedioxyhydroxymethamphetamine	Amphetamine	100.	33.3	7.9	11.9
methiopropamine	Other	100.	23.4	24.2	20.7
4-methylthioamphetamine	Amphetamine	100.	23.0	25.1	26.8
3,4-methylenedioxy-n-_allylamphetamine	Amphetamine	100.	9.1	9.1	13.1
n-hydroxy-n-methyl-3,4-methylenedioxyamphetamine	Amphetamine	100	6.7	6.7	4.1
buphedrone	Cathinone	84.6	23.3	24.4	25.3
methoxyphenamine	Amphetamine	75.0	8.9	26.8	35.2
scopolamine	Other	66.7	19.0	9.5	10.4
isopropylamphetamine	Amphetamine	62.9	40.0	34.8	28.2
2-fluoromethamphetamine	Amphetamine	57.1	30.9	25.3	27.2
pyrovalerone	Cathinone	53.8	40.2	34.9	25.1
alpha-pyrrolidinopentiophenone	Cathinone	50.0	39.5	28.7	21.7
n-[2-(1h-indol-3-yl)ethyl]-n-methyl-1-butanamine	Tryptamine	50.0	12.3	10.9	10.6
methylethyltryptamine	Tryptamine	50.0	9.5	11.3	10.2
4-methylamphetamine	Amphetamine	45.5	30.2	21.0	25.6
4-fluoroamphetamine	Amphetamine	44.7	33.0	23.5	26.6
bupropion	Cathinone	44.2	32.6	34.1	34.7
6-(2-aminopropyl)-5-methoxy-2-methyl-2,3-dihydrobenzofuran	Amphetamine	43.8	35.7	31.5	27.7
para-chloroamphetamine	Amphetamine	41.9	28.1	20.4	22.7
n-allyl-n-[2-(1h-indol-3-yl)ethyl]-2-propen-1-amine	Tryptamine	40.0	27.8	10.3	8.1
dipropyltryptamine	Tryptamine	40.0	13.3	10.3	8.9
n-allyl-n-[2-(5-methoxy-1h-indol-3-yl)ethyl]-2-propen-1-amine	Tryptamine	40.0	11.2	10.1	9.7

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
dimethylamphetamine	Amphetamine	38.7	25.3	22.9	23.1
meta-chlorophenylpiperazine	Other	36.7	35.1	36.7	42.1
3,6-dimethoxy-4-(2-aminopropyl)benzonorbornane	Amphetamine	35.7	35.6	31.2	21.1
ketamine	Other	34.9	29.2	30.6	27.1
3-methylamphetamine	Amphetamine	34.0	28.0	19.0	24.4
para-fluorophenylpiperazine	Other	33.3	37.3	36.5	29.7
2-fluoroamphetamine	Amphetamine	33.3	29.7	23.4	21.1
ortetamine	Amphetamine	33.3	26.0	34.2	25.4
lysergic_acid_diethylamide	Tryptamine	33.3	16.9	10.3	7.4
1h-indole-3-ethanamine,_5-methoxy-n,n-dipropyl-	Tryptamine	33.3	9.8	9.9	9.5
metamfepramone	Cathinone	32.1	29.7	32.0	25.2
gepefrine	Amphetamine	31.4	28.6	23.6	23.3
4-fluoromethamphetamine	Amphetamine	31.3	23.6	20.9	23.9
3-fluoroamphetamine	Amphetamine	30.2	28.0	20.7	24.4
benzylpiperazine	Other	30.0	27.1	29.5	28.1
dextromethorphan	Phenethylamine	29.8	29.3	20.8	15.2
phenethylamine	Phenethylamine	29.6	37.5	30.5	28.9
n-ethyl-6-methyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	29.6	21.4	22.0	22.6
3-methoxy-4,5-ethylenedioxyamphetamine	Amphetamine	28.6	28.6	5.3	4.1
n-[2-(6h-[1,3]dioxolo[4,5-e]indol-8-yl)ethyl]-n-isopropyl-2-propanamine	Tryptamine	28.6	7.4	5.7	7.4
ergoline-8beta-carboxamide,_1-acetyl-9,10-didehydro-n,n-diethyl-6-methyl-	Phenethylamine	28.6	7.0	7.6	6.7
4-methylmethamphetamine	Amphetamine	28.2	23.6	20.9	23.9
ethylamphetamine	Amphetamine	28.0	26.4	23.0	25.8
cannabinol	Cannabinoid	27.6	30.0	17.3	13.3

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
3,4-dimethylmethcathinone	Cathinone	26.7	23.6	24.2	25.0
methcathinone	Cathinone	26.5	15.9	23.4	25.6
para-bromoamphetamine	Amphetamine	25.0	26.3	30.9	26.5
clortermine	Amphetamine	25.0	25.0	39.7	27.6
3-methoxyamphetamine	Amphetamine	25.0	25.0	12.3	22.8
3-methoxy-4-methylamphetamine	Amphetamine	25.0	25.0	4.7	20.3
(8beta)-n,n-diethyl-6-propyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	25.0	23.8	22.0	19.3
5-methoxy-3-((r)-1-methyl-pyrrolidin-2-ylmethyl)-1h-indole	Tryptamine	25.0	7.7	7.4	12.1
amfepramone	Cathinone	23.7	17.8	23.6	29.0
amphetamine	Amphetamine	23.7	21.7	27.4	25.9
4-methyl-2,5-bis-(methylthio)amphetamine	Amphetamine	23.5	45.5	30.5	30.1
alpha-pyrrolidinopropiophenone	Cathinone	23.2	19.6	21.1	21.3
3-[2-(dipropylamino)ethyl]-1h-indol-4-ol	Tryptamine	23.1	10.6	10.1	8.3
(8beta)-n,n,6-trimethyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	21.1	18.2	18.0	15.0
ethcathinone	Cathinone	20.7	18.8	24.5	27.0
(2r)-1-(1h-indol-3-yl)-n-methyl-2-propanamine	Tryptamine	20.0	45.0	35.5	26.1
tryptamine	Tryptamine	20.0	25.0	27.3	37.5
alpha-pyrrolidinobutiophenone	Cathinone	20.0	24.7	24.0	25.2
methylbenzylpiperazine	Other	20.0	23.9	23.5	30.4
tetrahydrocannabinol	Cannabinoid	20.0	22.2	29.0	23.9
alpha-methyltryptamine	Tryptamine	20.0	20.0	47.4	32.1
2,alpha-dimethyltryptamine	Tryptamine	20.0	20.0	42.9	37.3
n-methyltryptamine	Tryptamine	20.0	20.0	32.1	30.4
nabilone	Cannabinoid	20.0	20.0	29.6	13.8

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
etolorex	Amphetamine	20.0	20.0	4.8	16.9
(8beta)-n,n-diethyl-6-propyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	20.0	12.7	15.4	17.8
dimethyltryptamine	Tryptamine	20.0	11.2	11.3	14.0
1-(5-fluoro-1h-indol-3-yl)propan-2-amine	Tryptamine	20.0	7.1	36.4	27.3
mephedrone	Cathinone	18.6	18.0	28.8	25.9
flephedrone	Cathinone	18.6	12.7	19.1	28.6
amfecloral	Amphetamine	18.4	33.7	30.4	31.6
(8beta)-6-butyl-n,n-diethyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	18.2	23.3	22.9	23.3
ergine	Tryptamine	18.2	21.5	18.3	17.4
n,n-diethyl-6-methyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	18.2	20.0	20.0	20.0
(8beta)-n,n-diethyl-6-(prop-2-en-1-yl)-9,10-didehydroergoline-8-carboxamide	Tryptamine	16.7	22.2	24.2	17.7
lysergic_acid_3-pentyl_amide	Tryptamine	16.7	19.7	19.3	21.8
4-methylphenylisobutylamine	Phenethylamine	16.7	17.5	24.6	26.6
4-chlorophenylisobutylamine	Phenethylamine	16.7	17.5	21.8	24.5
phenylisobutylamine	Phenethylamine	16.7	15.5	23.1	23.9
lysergic_acid_2-butyl_amide	Tryptamine	16.7	12.2	14.8	22.3
5-fluoro-n,n-dimethyltryptamine	Tryptamine	16.7	10.0	8.9	14.8
levomethamphetamine	Amphetamine	16.1	30.6	26.8	23.1
xylopropamine	Amphetamine	16.0	25.3	22.9	25.8
dexfenfluramine	Amphetamine	15.8	21.1	20.9	24.7
tiflorex	Amphetamine	15.8	14.9	20.3	23.1

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
(8beta)-n,n-diethyl-6-(prop-2-en-1-yl)-9,10-didehydroergoline-8-carboxamide	Tryptamine	15.4	23.2	21.9	20.2
(8beta)-n,n,6-triethyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	15.2	14.3	14.1	12.9
benzphetamine	Amphetamine	14.9	13.0	16.7	12.8
dextrorphan	Phenethylamine	14.3	21.5	16.9	16.6
n,n-dibutyltryptamine	Tryptamine	13.3	9.7	7.9	7.4
2,5-dimethoxy-beta-hydroxy-4-methylphenethylamine	Phenethylamine	13.3	9.1	5.8	4.3
cryogenine	Other	12.7	12.7	13.0	11.1
methysergide	Phenethylamine	12.5	12.5	10.0	19.5
5-methoxydimethyltryptamine	Tryptamine	12.5	7.8	8.4	10.0
5-methoxy-1h-indole-3-propane-2-amine	Tryptamine	12.5	5.6	10.4	16.8
elemicin	Other	12.5	4.5	4.0	5.1
5-methoxy-n,n-methylisopropyltryptamine	Tryptamine	12.1	11.4	11.1	8.8
dibutyltryptamine	Tryptamine	11.7	9.8	10.2	7.4
diethyltryptamine	Tryptamine	11.6	8.0	11.9	17.2
2-(2,5-dimethoxy-4-butylphenyl)ethan-1-amine	Phenethylamine	11.5	25.6	27.4	25.9
levoamphetamine	Amphetamine	11.5	22.6	25.7	25.9
oxilofrine	Phenethylamine	11.1	10.0	9.5	14.8
n,n-dimethyl-2-(2-methyl-1h-indol-3-yl)ethanamine	Tryptamine	11.1	9.3	9.1	14.5
1-(5-methoxy-1h-indol-3-yl)-n-methyl-2-propanamine	Tryptamine	11.0	9.3	8.4	7.7
5-methoxy-n,n-diethyltryptamine	Tryptamine	10.7	10.4	10.6	7.5
para-iodoamphetamine	Amphetamine	10.0	24.2	29.3	26.3
ergoline-8beta-carboxamide, 9,10-didehydro-n,n-diethyl-1,6-dimethyl-	Phenethylamine	10.0	20.0	6.3	9.6
ergometrine	Tryptamine	10.0	6.5	4.9	7.4

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
1-(5-methoxy-1h-indol-3-yl)-2-butanamine	Tryptamine	9.8	9.2	9.1	14.9
melatonin	Tryptamine	9.8	8.5	8.8	16.0
5,n-dimethyl-n-isopropyltryptamine	Tryptamine	9.7	10.6	9.9	13.5
methylisopropyltryptamine	Tryptamine	9.7	6.3	11.9	14.1
amphetaminil	Amphetamine	9.7	12.3	14.0	15.3
2-(5-methoxy-2-methyl-1h-indol-3-yl)-n,n-dimethylethanamine	Tryptamine	9.6	8.3	10.4	12.8
5-methoxy-diisopropyltryptamine	Tryptamine	9.5	11.9	11.1	9.9
cabergoline	Tryptamine	8.7	8.0	10.8	2.7
norpholedrine	Amphetamine	8.3	24.3	15.8	11.9
methylenedioxypyrovalerone	Cathinone	8.3	2.8	9.3	10.4
methylergometrine	Tryptamine	8.0	4.8	2.6	4.4
harmaline	Tryptamine	7.4	10.6	15.9	13.4
n,n-diethyltryptamine	Tryptamine	7.4	8.4	17.9	17.9
harmine	Other	7.4	11.3	16.4	13.4
6,7-dihydro-5h-indeno(5,6-d)-1,3-dioxol-6-amine	Phenethylamine	7.1	8.3	7.7	6.8
1-naphthyl(1-pentyl-1h-indol-3-yl)methanone	Other	7.1	12.6	14.3	15.0
tetrahydroharmine	Tryptamine	6.8	13.8	15.9	16.9
alpha-methylserotonin	Tryptamine	6.3	5.0	9.1	9.3
mescaline	Phenethylamine	5.4	4.2	3.6	4.9
n-(2-fluorobenzyl)-2-(4-iodo-2,5-dimethoxyphenyl)ethanamine	Phenethylamine	4.0	10.1	10.1	12.0
bromocriptine	Tryptamine	3.2	3.2	2.7	4.7
epicriptine	Tryptamine	3.1	2.0	2.0	5.0
levonordefrin	Phenethylamine	3.1	4.9	5.8	4.8
isomescaline	Phenethylamine	3.0	5.5	5.5	5.4
(1-butyl-1h-indol-3-yl)(1-naphthyl)methanon	Other	2.6	9.8	17.5	13.9

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
ergotamine	Tryptamine	2.0	1.7	2.1	2.0
dihydroergocryptine	Tryptamine	2.0	2.0	4.2	9.6
2-(5-methoxy-2-methyl-1h-indol-3-yl)-n,n-dimethylethanamine	Tryptamine	1.5	9.3	12.3	13.3
dihydroergotamine	Tryptamine	1.3	1.5	1.3	5.5
1-[(7r)-3-bromo-2,5-dimethoxybicyclo[4.2.0]octa-1,3,5-trien-7-yl]methanamine	Phenethylamine	1.1	1.1	2.0	6.3
3,4-methylenedioxy-n-cyclopropylmethylamphetamine	Amphetamine	0.0	100.0	100.0	16.7
3,4-methylenedioxy-n-butylamphetamine	Amphetamine	0.0	100.0	16.7	9.1
2-ethylamino-1-(3,4-methylenedioxyphenyl)butane	Phenethylamine	0.0	100.0	11.0	9.5
methylenedioxyethylamphetamine	Amphetamine	0.0	100.0	7.7	17.5
2-methylamino-1-(3,4-methylenedioxyphenyl)pentane	Phenethylamine	0.0	100.0	7.1	12.9
para-methoxyethylamphetamine	Amphetamine	0.0	87.5	50.0	24.1
4-methoxy-n-methylamphetamine	Amphetamine	0.0	50.0	41.9	26.2
homopiperonylamine	Phenethylamine	0.0	50.0	2.2	7.2
alpha-ethyltryptamine	Tryptamine	0.0	47.4	41.5	25.7
2,3-methylenedioxyamphetamine	Amphetamine	0.0	43.1	44.2	16.1
jimscaline	Phenethylamine	0.0	42.9	35.7	3.9
3,4-ethylenedioxy-n-methylamphetamine	Amphetamine	0.0	42.9	23.1	18.2
tetralinylaminopropane	Amphetamine	0.0	41.9	46.5	27.7
6-(2-aminopropyl)-5-methoxy-1,3-benzoxathiol	Amphetamine	0.0	38.1	27.3	32.1
1-(3,5-dimethoxy-4-propoxyphenyl)-2-propanamine	Amphetamine	0.0	37.5	9.3	6.2
4-methyl-alpha-ethyltryptamine	Tryptamine	0.0	36.4	43.6	24.9
6-benzofuranethanamine, _2,3-dihydro-5-methoxy-alpha,2-dimethyl-	Amphetamine	0.0	35.7	30.2	28.5
1-(1-benzofuran-6-yl)-2-propanamine	Amphetamine	0.0	35.0	11.8	34.7

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
2,4-dimethoxy-5-methylthioamphetamine	Amphetamine	0.0	33.3	33.3	17.8
2-bromo-4,5-methylenedioxyamphetamine	Amphetamine	0.0	33.3	13.1	12.1
ethylisopropyltryptamine	Tryptamine	0.0	33.3	12.8	13.6
n-ethyl-n-[2-(1h-indol-3-yl)ethyl]-2-propanamine	Tryptamine	0.0	33.3	10.5	14.0
5-(2-aminopropyl)benzofuran	Amphetamine	0.0	30.8	31.2	29.7
trifluoromethylphenylpiperazine	Other	0.0	30.0	35.0	29.6
cathine	Phenethylamine	0.0	29.8	29.1	30.7
methedrone	Cathinone	0.0	29.5	22.4	18.8
3,6-dimethoxy-4-(2-aminoethyl)benzonorbornane	Phenethylamine	0.0	29.0	24.6	18.4
2,3-dimethoxy-4,5-methylenedioxyamphetamine	Amphetamine	0.0	28.6	28.6	28.6
1,4-dimethoxynaphthyl-2-isopropylamine	Amphetamine	0.0	28.6	22.2	14.9
2,5-dimethoxy-3,4-(trimethylene)phenethylamine; _5-(2-aminoethyl)-4,7-dimethoxyindane)	Phenethylamine	0.0	28.6	22.2	6.8
2,5-dimethoxy-3,4-(trimethylene)amphetamine	Amphetamine	0.0	28.6	22.2	5.6
3-[2-(diisopropylamino)ethyl]-1h-indol-4-ol	Tryptamine	0.0	26.3	10.7	7.4
propylisopropyltryptamine	Tryptamine	0.0	26.3	10.3	13.3
pholedrine	Amphetamine	0.0	25.6	19.6	15.2
ecstasy	Amphetamine	0.0	25.0	16.0	14.8
n-[2-(5,6-dimethoxy-1h-indol-3-yl)ethyl]-n-methyl-2-propanamine	Tryptamine	0.0	25.0	15.0	9.4
ethacetin	Tryptamine	0.0	25.0	10.0	18.3
3-methoxymethamphetamine	Amphetamine	0.0	25.0	4.7	25.2
eden	Phenethylamine	0.0	25.0	3.3	12.5
2,5-dimethoxy-3,4-(tetramethylene)phenethylamine; _6-(2-aminoethyl)-5,8-dimethoxy-tetralin	Phenethylamine	0.0	22.2	5.9	11.3
prenylamine	Amphetamine	0.0	20.0	18.1	14.1

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
(2s)-1-(3,4-dimethoxyphenyl)-2-propanamine	Amphetamine	0.0	20.0	7.1	16.2
o-acetylpsilocin	Tryptamine	0.0	18.2	22.3	21.4
morphinan	Phenethylamine	0.0	16.7	27.0	24.2
3-{2-[isopropyl(methyl)amino]ethyl}-1h-indol-4-yl_acetate	Tryptamine	0.0	16.7	16.2	18.0
n-ethyltryptamine	Tryptamine	0.0	16.7	14.3	22.3
(8beta)-6-cyclopropyl-n,n-diethyl-9,10-didehydroergoline-8-carboxamide	Tryptamine	0.0	15.4	15.4	14.1
3,4-methylenedioxyphentermine	Amphetamine	0.0	14.3	17.8	20.1
4-thiometaescaline	Phenethylamine	0.0	14.3	14.3	16.7
3-[2-(dibutylamino)ethyl]-1h-indol-4-ol	Tryptamine	0.0	14.3	10.7	8.3
2-(4-iodo-2,5-dimethoxyphenyl)ethanamine	Phenethylamine	0.0	13.3	11.1	2.2
nexus	Phenethylamine	0.0	13.3	7.4	6.1
n,n-dimethyl-2-[5-(methylsulfanyl)-1h-indol-3-yl]ethanamine	Tryptamine	0.0	12.8	11.6	15.9
etafedrine	Phenethylamine	0.0	12.5	16.7	22.9
methylone	Cathinone	0.0	11.1	22.2	12.5
methylenedioxyhydroxyamphetamine	Amphetamine	0.0	10.7	4.3	7.4
atropine	Other	0.0	10.7	10.2	12.6
[(8beta)-6-methyl-9,10-didehydroergolin-8-yl](4-morpholinyl)methanone	Tryptamine	0.0	10.5	10.0	7.8
(2s)-1-(1,3-benzodioxol-5-yl)-2-butanamine	Phenethylamine	0.0	10.1	11.9	9.8
indole,_3-[2-(dimethylamino)ethyl]-5-methyl-	Tryptamine	0.0	10.0	8.9	15.1
3-[2-(1-pyrrolidinyl)ethyl]-1h-indole	Tryptamine	0.0	9.8	8.8	20.1
aldosterone-stimulating_hormone	Tryptamine	0.0	9.7	11.4	12.4
lysergic_acid_hydroxyethylamide	Tryptamine	0.0	9.4	6.1	7.6
cafedrine	Phenethylamine	0.0	9.3	8.8	7.7
5h-1,3-dioxolo[4,5-f]indole-7-ethanamine,_n,n-dimethyl-	Tryptamine	0.0	9.3	8.0	12.3

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
dibenzylpiperazine	Other	0.0	9.1	6.8	4.3
3,4-methylenedioxy-n-hydroxyamphetamine	Amphetamine	0.0	8.9	4.3	7.4
propylamphetamine	Amphetamine	0.0	8.5	16.4	25.9
1-(5-methoxy-1h-indol-3-yl)-n-methyl-2-propanamine	Tryptamine	0.0	8.4	7.9	14.7
[3-[2-(diisopropylamino)ethyl]-1h-indol-4-yl] acetate	Tryptamine	0.0	8.3	9.6	11.6
n,n-diethyl-2-(2-methyl-1h-indol-3-yl)ethanamine	Tryptamine	0.0	8.3	20.0	22.6
diisopropyltryptamine	Tryptamine	0.0	7.9	11.6	14.2
naphyrone	Cathinone	0.0	7.6	10.4	7.5
metaproscaline	Phenethylamine	0.0	7.4	5.8	4.7
5-methoxy-3-[2-(pyrrolidin-1-yl)ethyl]-1h-indole	Tryptamine	0.0	7.4	6.3	9.4
phenescaline	Phenethylamine	0.0	7.3	7.5	10.1
4-methoxy-n-methyl-n-isopropyltryptamine	Tryptamine	0.0	7.1	10.3	13.8
allylescaline	Phenethylamine	0.0	7.1	7.1	7.1
1,4-dimethoxynaphthyl-2-ethylamine	Phenethylamine	0.0	6.9	12.3	10.0
methoxamine	Phenethylamine	0.0	6.9	2.5	3.3
phencyclidine	Other	0.0	6.3	33.7	34.9
2-bromo-lsd	Tryptamine	0.0	6.3	10.4	5.0
n-[[(7r)-3-bromo-2,5-dimethoxy-7-bicyclo[4.2.0] octa-1,3,5-trienyl]methyl]-1-(2-methoxyphenyl)methanamine	Phenethylamine	0.0	6.0	4.0	3.0
para-methoxymethamphetamine	Amphetamine	0.0	5.3	5.5	26.2
2-(4-iodo-2,5-dimethoxyphenyl)-n-(2-methoxybenzyl)ethanamine	Phenethylamine	0.0	5.3	1.5	5.0
[(8beta)-6-methyl-9,10-didehydroergolin-8-yl](pyrrolidin-1-yl)methanone	Tryptamine	0.0	5.1	19.8	9.5
norbaeocystin	Other	0.0	5.0	5.6	3.9

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
3,4-methylenedioxy-n-(2-hydroxyethyl)amphetamine	Amphetamine	0.0	5.0	2.6	3.0
alpha-methyldopamine	Amphetamine	0.0	4.9	5.1	4.9
para-methoxyphenylpiperazine	Other	0.0	4.8	21.1	22.0
escaline	Phenethylamine	0.0	4.6	4.6	3.2
pentorex	Phenethylamine	0.0	4.3	25.0	31.4
psilocin	Tryptamine	0.0	4.3	3.9	4.4
isoproscaline	Phenethylamine	0.0	4.2	4.5	4.0
phenescaline	Phenethylamine	0.0	4.2	7.3	2.8
asymbescaline	Phenethylamine	0.0	4.2	4.5	5.4
methylbenzodioxolylbutanamine	Phenethylamine	0.0	4.0	3.3	14.7
lysergic_acid_2,4-dimethylazetidide	Tryptamine	0.0	3.9	9.1	7.7
4-ethylamphetamine	Amphetamine	0.0	3.6	25.0	30.1
myristicin	Other	0.0	3.3	4.4	3.9
3,4,5-trimethoxyamphetamine	Amphetamine	0.0	3.1	5.5	4.5
proscaline	Phenethylamine	0.0	3.1	4.6	5.6
3,5-dimethoxy-4-(2-propynyloxy)phenethylamine	Phenethylamine	0.0	3.1	2.0	5.6
5h-1,3-dioxolo[4,5-f]indole-7-ethanamine, _n,n-bis(1-methylethyl)-	Tryptamine	0.0	3.1	5.5	9.4
4-bromo-2,5-dimethoxy-1-benzylpiperazine	Other	0.0	3.0	7.5	10.6
2,3,4-trimethoxyamphetamine	Amphetamine	0.0	3.0	4.3	5.5
symbescaline	Phenethylamine	0.0	3.0	4.1	3.9
fencamine	Amphetamine	0.0	2.7	3.0	8.5
4-methoxyamphetamine	Amphetamine	0.0	2.7	3.7	16.9

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
ergoloid	Tryptamine	0.0	2.7	2.1	3.9
4-hydroxy-5-methoxydimethyltryptamine	Tryptamine	0.0	2.1	2.3	5.1
4-allyloxy-3,5-dimethoxyphenethylamine	Phenethylamine	0.0	2.0	2.0	6.7
formoterol	Amphetamine	0.0	1.9	1.9	1.2
2-(4-chloro-2,5-dimethoxyphenyl)-n-(2-methoxybenzyl)ethanamine	Phenethylamine	0.0	1.6	3.5	4.8
2-(4-bromo-2,5-dimethoxyphenyl)-n-(2-methoxybenzyl)ethanamine	Phenethylamine	0.0	1.6	3.5	4.4
amfepentorex	Amphetamine	0.0	1.5	13.8	23.6
fenethylline	Amphetamine	0.0	1.3	3.7	4.2
2,5-dimethoxy-4-nitrophenethylamine	Phenethylamine	0.0	1.1	1.1	2.3
2,5-dimethoxy-4-nitrophenethylamine	Phenethylamine	0.0	1.1	1.0	2.3
3,4-methylenedioxy-2-methylthioamphetamine	Amphetamine	0.0	0.0	100.0	9.5
3,4-methylenedioxy-n,n-dimethylamphetamine	Amphetamine	0.0	0.0	50.0	7.9
2-methoxy-4-methyl-5-methylthioamphetamine	Amphetamine	0.0	0.0	36.8	35.4
1-(4,7-dimethoxy-1,3-benzodioxol-5-yl)propan-2-amine	Amphetamine	0.0	0.0	28.6	4.3
2,5-dimethoxy-n,n-dimethyl-4-iodoamphetamine	Amphetamine	0.0	0.0	25.0	34.8
2,5-dimethoxy-4-ethylthio-n-hydroxyphenethylamine	Phenethylamine	0.0	0.0	25.0	7.7
1-(2,3-dihydro-1-benzofuran-5-yl)propan-2-amine	Amphetamine	0.0	0.0	24.6	25.2
2,5-dimethoxy-3,4-(tetramethylene)amphetamine	Amphetamine	0.0	0.0	22.2	32.9
4-ethyl-2-methoxy-5-methylthioamphetamine	Amphetamine	0.0	0.0	21.6	15.7
4,5-dimethoxy-2-methylthioamphetamine	Amphetamine	0.0	0.0	21.4	13.2
indanylaminopropane	Amphetamine	0.0	0.0	20.9	30.8
4-ethyl-5-methoxy-2-methylthioamphetamine	Amphetamine	0.0	0.0	17.6	23.6
5-thioasymbescaline	Phenethylamine	0.0	0.0	16.7	13.5
ethylidenedioxyamphetamine	Amphetamine	0.0	0.0	15.4	13.6

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
2,4-dimethoxyamphetamine	Amphetamine	0.0	0.0	14.3	25.9
2-methoxy-n-methyl-4,5-methylenedioxyamphetamine	Amphetamine	0.0	0.0	14.3	23.7
para-methoxyamphetamine	Amphetamine	0.0	0.0	14.3	19.9
2,4,5-trimethoxyamphetamine	Amphetamine	0.0	0.0	14.3	7.4
4,5-dimethoxy-2-ethoxyamphetamine	Phenethylamine	0.0	0.0	14.3	5.8
3,4-methylenedioxy-n-benzylamphetamine	Amphetamine	0.0	0.0	14.0	13.2
3,5-dimethoxy-4-methylphenethylamine	Phenethylamine	0.0	0.0	13.3	6.3
2,5-dimethoxy-4-ethylphenethylamine	Phenethylamine	0.0	0.0	13.3	4.2
5-methoxy-4-methyl-2-methylthioamphetamine	Amphetamine	0.0	0.0	12.5	25.9
4-bromo-3,5-dimethoxyamphetamine	Amphetamine	0.0	0.0	11.1	12.8
benzodioxolylbutanamine	Phenethylamine	0.0	0.0	10.0	16.0
3,5-dimethoxy-4-ethoxyamphetamine	Amphetamine	0.0	0.0	9.3	6.7
3-{2-[methyl(propyl)amino]ethyl}-1h-indol-4-ol	Tryptamine	0.0	0.0	7.7	3.5
2,5-dimethoxy-4-phenylthioamphetamine	Amphetamine	0.0	0.0	7.1	6.3
methylenedioxymethylphenethylamine	Phenethylamine	0.0	0.0	5.9	8.3
6h-1,3-dioxolo[4,5-e]indole-8-ethanamine,_n,n-dimethyl-	Tryptamine	0.0	0.0	5.9	5.2
(6-methyl-9,10-didehydroergolin-8-yl) (1-piperidinyl)methanone	Tryptamine	0.0	0.0	5.8	7.8
baeocystin	Tryptamine	0.0	0.0	5.7	3.2
metaescaline	Phenethylamine	0.0	0.0	5.5	4.1
2-methoxy-4-methyl-5-methylsulfinylamphetamine	Amphetamine	0.0	0.0	5.0	11.8
1h-indol-4-ol,_3-[2-(ethylmethylamino)ethyl]-	Tryptamine	0.0	0.0	3.8	6.2
bufotenin	Tryptamine	0.0	0.0	3.4	7.1
4-benzyloxy-3,5-dimethoxyamphetamine	Amphetamine	0.0	0.0	3.0	7.6
buscaline	Phenethylamine	0.0	0.0	2.9	6.6

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
beta-methoxy-2c-b;_4-bromo-2,5-beta-trimethoxyphenethylamine	Phenethylamine	0.0	0.0	2.6	5.8
cyclopropylmescaline	Phenethylamine	0.0	0.0	2.0	6.6
trisescaline	Phenethylamine	0.0	0.0	2.0	6.2
benfluorex	Amphetamine	0.0	0.0	1.4	11.1
thiobuscaline	Phenethylamine	0.0	0.0	0.0	100.0
4-methyl-2,5-methoxyphenylcyclopropylamine	Phenethylamine	0.0	0.0	0.0	48.6
3-thiomescaline	Phenethylamine	0.0	0.0	0.0	33.3
1-(2,6-dimethoxy-4-methylphenyl)propan-2-amine	Amphetamine	0.0	0.0	0.0	29.6
(8beta)-n,n-diethyl-6-(2-propyn-1-yl)-9,10-didehydroergoline-8-carboxamide	Tryptamine	0.0	0.0	0.0	27.8
para-ethoxyamphetamine	Amphetamine	0.0	0.0	0.0	27.7
(2s)-1-(2,5-dimethoxyphenyl)-2-propanamine	Amphetamine	0.0	0.0	0.0	26.8
2,5-dimethoxy-4-ethylamphetamine	Amphetamine	0.0	0.0	0.0	26.4
2,5-dimethoxyphenethylamine	Phenethylamine	0.0	0.0	0.0	26.3
beatrice_(psychedelic)	Amphetamine	0.0	0.0	0.0	24.2
2,5-dimethoxy-4-chloroamphetamine	Amphetamine	0.0	0.0	0.0	23.2
2,5-dimethoxy-4-(n)-amylamphetamine	Amphetamine	0.0	0.0	0.0	21.6
2,5-dimethoxy-4-fluoroamphetamine	Amphetamine	0.0	0.0	0.0	20.5
2,5-dimethoxy-n-methylamphetamine	Amphetamine	0.0	0.0	0.0	20.2
2,n-dimethyl-4,5-methylenedioxyamphetamine	Amphetamine	0.0	0.0	0.0	20.0
4-bromo-2,5-dimethoxy-n-methylamphetamine	Amphetamine	0.0	0.0	0.0	19.8
2,5-dimethoxy-4-bromoamphetamine	Amphetamine	0.0	0.0	0.0	18.5
2,5-dimethoxy-4-methylamphetamine	Amphetamine	0.0	0.0	0.0	17.4
ecstasy	Amphetamine	0.0	0.0	0.0	16.9
2,5-dimethoxy-4-iodoamphetamine	Amphetamine	0.0	0.0	0.0	16.7

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
2-[4-(isopropylsulfanyl)-2,6-dimethoxyphenyl]ethanamine	Phenethylamine	0.0	0.0	0.0	16.7
3,4-methylenedioxy-n-(2-methoxyethyl)amphetamine	Amphetamine	0.0	0.0	0.0	16.7
furfenorex	Amphetamine	0.0	0.0	0.0	16.0
4-thiosymbescaline	Phenethylamine	0.0	0.0	0.0	15.4
1-(6-methyl-1,3-benzodioxol-5-yl)-2-propanamine	Amphetamine	0.0	0.0	0.0	14.8
2-methoxy-4,5-methylenedioxyamphetamine	Amphetamine	0.0	0.0	0.0	13.3
5-thiometaescaline	Phenethylamine	0.0	0.0	0.0	12.7
4-thiotrescaline	Phenethylamine	0.0	0.0	0.0	12.5
3,4-methylenedioxy-n-methyoxyamphetamine	Amphetamine	0.0	0.0	0.0	12.0
3-thioasymbescaline	Phenethylamine	0.0	0.0	0.0	11.9
3-methoxy-4-ethoxyphenethylamine	Phenethylamine	0.0	0.0	0.0	11.7
4-thioasymbescaline	Phenethylamine	0.0	0.0	0.0	10.5
3,4-methylenedioxyamphetamine	Amphetamine	0.0	0.0	0.0	9.7
2-[2,5-dimethoxy-4-(trifluoromethyl) phenyl]-n-(2-methoxybenzyl)ethanamine	Phenethylamine	0.0	0.0	0.0	9.4
2,5-dimethoxy-4-methylthiophenethylamine	Phenethylamine	0.0	0.0	0.0	9.4
1-(4-methyl-1,3-benzodioxol-6-yl)-2-aminopropane	Amphetamine	0.0	0.0	0.0	9.1
2-methyl-3,4-methylenedioxyamphetamine	Amphetamine	0.0	0.0	0.0	9.0
3-thiotrescaline	Phenethylamine	0.0	0.0	0.0	8.7
3-[2-(diethylamino)ethyl]-1h-indol-4-ol	Tryptamine	0.0	0.0	0.0	8.7
5-bromo-2,4-dimethoxyamphetamine	Amphetamine	0.0	0.0	0.0	7.7
3-{[(2r)-1-methyl-2-pyrrolidinyl]methyl}-1h-indol-4-ol	Tryptamine	0.0	0.0	0.0	7.2
2,5-dimethoxy-n-hydroxy-4-(n)-propylthiophenethylamine	Phenethylamine	0.0	0.0	0.0	7.1
1-(4-ethynyl-2,5-dimethoxyphenyl)-2-aminoethane	Phenethylamine	0.0	0.0	0.0	6.7
2,5-dimethoxy-4-(i)-propylthioamphetamine	Amphetamine	0.0	0.0	0.0	6.7
3-[2-(pyrrolidin-1-yl)ethyl]-1h-indol-4-ol	Tryptamine	0.0	0.0	0.0	6.6

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
benzeneethanamine,_4-(ethylthio)-3,5-dimethoxy-	Phenethylamine	0.0	0.0	0.0	6.1
methylenedioxyphenethylamine	Phenethylamine	0.0	0.0	0.0	5.8
2,4,5-trimethoxyphenethylamine	Phenethylamine	0.0	0.0	0.0	5.8
methallylescaline	Phenethylamine	0.0	0.0	0.0	5.6
3-thioescaline	Phenethylamine	0.0	0.0	0.0	5.6
n-[2-(5h-[1,3]dioxolo[4,5-f]indol-7-yl) ethyl]-n-methyl-2-propanamine	Tryptamine	0.0	0.0	0.0	5.3
2,5-dimethoxy-4-nitroamphetamine	Amphetamine	0.0	0.0	0.0	5.1
2,5-dimethoxy-4-(t)-butylthiophenethylamine	Phenethylamine	0.0	0.0	0.0	5.0
benzeneethanamine,_3,5-dimethoxy-4-(methylthio)-	Phenethylamine	0.0	0.0	0.0	4.9
2,5-dimethoxy-4-(n)-propylthioamphetamine	Amphetamine	0.0	0.0	0.0	4.8
3-{2-[methyl(propan-2-yl)amino]ethyl}-1h-indol-4-ol	Tryptamine	0.0	0.0	0.0	4.5
3-thiomescaline	Phenethylamine	0.0	0.0	0.0	4.5
ethocybin	Tryptamine	0.0	0.0	0.0	4.3
4-thioisomescaline	Phenethylamine	0.0	0.0	0.0	4.1
2,5-dimethoxy-4-ethoxyamphetamine	Amphetamine	0.0	0.0	0.0	3.8
lophophine	Phenethylamine	0.0	0.0	0.0	3.8
2,5-dimethoxy-4-(2-methoxyethylthio)phenethylamine	Phenethylamine	0.0	0.0	0.0	3.7
3-thiometaescaline	Phenethylamine	0.0	0.0	0.0	3.6
3-{2-[methyl(propan-2-yl)amino]ethyl}-1h-indol-4-ol	Tryptamine	0.0	0.0	0.0	3.5
4-bromomethcathinone	Cathinone	0.0	0.0	0.0	3.5
3-thiosymbescaline	Phenethylamine	0.0	0.0	0.0	3.4
2-thioisomescaline	Phenethylamine	0.0	0.0	0.0	3.3
2,5-diethoxy-4-methoxyamphetamine	Phenethylamine	0.0	0.0	0.0	3.0
4-methylthio-2-5-dimethoxyamphetamine	Amphetamine	0.0	0.0	0.0	2.9
aeruginascin	Tryptamine	0.0	0.0	0.0	2.9

continued on next page

Table B.2.: *continued*

Compound	Category	Top 10	Top 25	Top 40	Top 100
3,4-dimethoxyphenethylamine	Phenethylamine	0.0	0.0	0.0	2.8
thioproscaline	Phenethylamine	0.0	0.0	0.0	2.8
2,3,6-trimethoxyamphetamine	Amphetamine	0.0	0.0	0.0	2.3
2,5-dimethoxy-3,4-dimethylamphetamine	Amphetamine	0.0	0.0	0.0	2.1

Indication	Category	Top 10	Top 25	Top 40	Top 100
Jet Lag Syndrome	Tryptamine	100	100	100	63.6
Seasonal Affective Disorder	Tryptamine	100	100	100	63.6
Sleep Disorders, Circadian Rhythm	Tryptamine	100	100	100	63.6
Binge-Eating Disorder	Amphetamine	93.8	100	100	77.0
Aphasia, Broca	Tryptamine	44.4	58.3	69.2	55.6
Narcolepsy	Amphetamine	35.6	39.8	45.6	58.3
Anorexia Nervosa	Amphetamine	29.6	22.2	19.7	18.5
Auditory Perceptual Disorders	Cathinone	22.2	16.7	18.0	9.09
Sleep-Wake Transition Disorders	Other	22.2	8.33	5.13	2.02
Epilepsies, Myoclonic	Amphetamine	20.0	24.4	21.0	19.2
Pica	Cathinone	19.4	10.4	6.41	3.08
Sleep Bruxism	Amphetamine	18.5	22.1	18.4	17.3
Sleep Bruxism	Tryptamine	18.5	8.82	5.50	3.46
Cataplexy	Amphetamine	17.0	17.3	19.0	23.7
Restless Legs Syndrome	Tryptamine	16.7	22.1	24.4	25.4
Dysthymic Disorder	Cathinone	14.8	8.46	5.50	4.13
Tic Disorders	Tryptamine	13.9	7.29	7.05	6.19
Bulimia Nervosa	Cathinone	12.1	6.98	6.67	4.97
ADHD	Tryptamine	11.1	14.6	17.3	15.4
Pick Disease of the Brain	Cathinone	11.1	8.33	7.69	4.04
Alcohol-Related Disorders	Cathinone	11.1	8.33	6.58	5.78
Auditory Perceptual Disorders	Other	11.1	8.33	5.13	5.05
ADHD	Tryptamine	10.8	11.8	14.9	20.5
Conduct Disorder	Amphetamine	9.89	10.7	12.1	13.2
Mood Disorders	Cathinone	9.88	6.12	5.26	3.59
Obsessive-Compulsive Disorder	Amphetamine	9.78	10.6	11.6	18.85
Restless Legs Syndrome	Cathinone	9.52	5.88	5.06	3.41
Personality Disorders	Amphetamine	8.84	9.87	10.4	15.4
Panic Disorder	Amphetamine	8.18	10.5	12.2	21.4
Depressive Disorder, Major	Amphetamine	7.42	11.8	15.1	23.9
Alcohol Withdrawal Delirium	Tryptamine	7.41	10.3	12.4	11.7
Eating Disorders	Cathinone	7.41	5.56	5.98	5.54
Restless Legs Syndrome	Amphetamine	7.14	13.2	14.6	18.3
SIMD	Tryptamine	6.80	10.0	13.0	13.8
Narcolepsy	Cathinone	6.67	4.85	6.71	14.2
Agoraphobia	Cathinone	6.67	3.57	3.45	2.69
Cataplexy	Tryptamine	6.38	5.00	4.91	7.68

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Child Development Disorders, Pervasive	Amphetamine	6.25	5.88	9.29	9.91
Epilepsy, Complex Partial	Amphetamine	6.12	7.98	7.31	11.5
Amnesia	Amphetamine	5.95	7.87	9.27	17.3
Autistic Disorder	Tryptamine	5.91	8.42	11.3	15.8
Depressive Disorder, Major	Cathinone	5.86	4.29	4.22	5.45
Erectile Dysfunction	Tryptamine	5.80	4.28	3.89	5.83
Delirium	Cathinone	5.63	3.76	4.56	2.82
Dementia, Vascular	Amphetamine	5.56	6.25	5.13	12.1
Dyspareunia	Cathinone	5.56	6.25	3.85	4.04
ADHD	Amphetamine	5.56	5.21	8.33	8.23
Myoclonic Epilepsy, Juvenile	Cathinone	5.56	4.17	6.76	4.44
Speech Disorders	Phenethylamine	5.56	4.17	2.56	2.02
Developmental Disabilities	Other	5.56	3.26	4.08	2.69
Tic Disorders	Other	5.56	3.13	5.13	3.35
Epilepsy	Amphetamine	5.42	6.57	7.44	11.4
Cognition Disorders	Tryptamine	5.41	6.03	6.41	10.1
Affective Disorders, Psychotic	Amphetamine	5.17	3.68	3.88	4.10
Learning Disorders	Tryptamine	5.08	7.92	10.54	14.8
Bipolar Disorder	Amphetamine	5.08	6.10	6.03	10.6
Depressive Disorder	Tryptamine	5.02	4.99	6.44	10.9
Autistic Disorder	Amphetamine	5.00	6.11	6.61	16.2
Anxiety Disorders	Amphetamine	4.94	7.06	9.55	20.3
Stereotypic Movement Disorder	Tryptamine	4.80	4.21	5.13	6.50
Seizures	Amphetamine	4.59	8.31	11.24	22.3
Panic Disorder	Cathinone	4.55	2.94	4.17	5.01
Narcolepsy	Other	4.44	2.91	4.03	5.21
Epilepsy, Tonic-Clonic	Amphetamine	4.42	6.87	6.65	7.74
Cocaine-Related Disorders	Cathinone	4.39	3.67	3.50	5.19
Erectile Dysfunction	Phenethylamine	4.35	4.28	3.69	5.10
Obsessive-Compulsive Disorder	Cathinone	4.35	2.75	3.45	4.83
Amphetamine Disorders	Tryptamine	4.21	5.62	5.98	10.4
Cocaine-Related Disorders	Amphetamine	4.17	8.02	9.98	18.96
Phobic Disorders	Cathinone	4.00	2.61	2.33	3.07
Heroin Dependence	Cathinone	3.90	3.55	3.26	3.63
ADHD	Other	3.85	4.30	5.38	5.59
Anxiety Disorders	Tryptamine	3.80	5.16	8.10	11.6
Anxiety Disorders	Cathinone	3.80	3.44	3.99	5.33
Epilepsies, Myoclonic	Tryptamine	3.75	3.98	2.85	2.99

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Amphetamine Disorders	Cathinone	3.74	2.92	2.91	3.00
Epilepsy, Rolandic	Phenethylamine	3.70	2.82	1.83	1.55
Alcoholic Intoxication	Cathinone	3.70	2.78	1.71	1.56
Sleep Bruxism	Cathinone	3.70	1.47	3.67	2.69
Epilepsy, Rolandic	Amphetamine	3.70	1.41	0.92	0.78
Eating Disorders	Amphetamine	3.70	1.39	1.71	7.01
Depressive Disorder	Amphetamine	3.68	4.99	7.13	14.3
Erectile Dysfunction	Amphetamine	3.62	4.89	8.20	12.7
Tourette Syndrome	Cathinone	3.54	3.17	3.60	2.64
Tobacco Use Disorder	Cathinone	3.45	2.58	2.58	3.26
Tobacco Use Disorder	Other	3.45	1.94	2.15	2.65
Amnesia	Tryptamine	3.35	5.03	6.29	11.0
Personality Disorders	Cathinone	3.31	2.78	3.24	3.91
Sexual Dysfunctions, Psychological	Cannabinoid	3.30	1.32	0.84	0.39
Cognition Disorders	Cathinone	3.24	2.68	2.50	2.21
Cataplexy	Other	3.19	3.18	3.68	3.84
Child Development Disorders, Pervasive	Other	3.13	1.18	2.14	2.10
Schizophrenia	Amphetamine	3.10	3.60	4.27	7.11
ADHD	Phenethylamine	3.08	2.51	2.20	2.93
Cocaine-Related Disorders	Tryptamine	3.07	6.90	10.24	20.8
Seizures	Tryptamine	3.06	2.93	4.33	12.6
Bulimia Nervosa	Amphetamine	3.03	9.30	11.85	18.2
Bulimia Nervosa	Phenethylamine	3.03	3.49	2.22	3.31
Amnesia	Phenethylamine	2.97	4.86	6.86	9.56
Amnesia	Cathinone	2.97	2.68	2.86	4.20
Opioid-Related Disorders	Cathinone	2.94	3.28	2.72	3.17
Opioid-Related Disorders	Other	2.94	2.73	2.38	2.06
Dementia	Amphetamine	2.90	3.83	2.69	7.81
Erectile Dysfunction	Cathinone	2.90	3.67	3.69	3.23
Dementia	Cathinone	2.90	1.64	1.35	1.65
Learning Disorders	Cathinone	2.82	2.48	2.17	2.19
Pica	Amphetamine	2.78	10.42	12.8	11.3
Developmental Disabilities	Cathinone	2.78	6.52	6.12	5.69
Tic Disorders	Cathinone	2.78	4.17	3.85	3.09
Developmental Disabilities	Amphetamine	2.78	3.26	2.72	11.68
Amnesia, Anterograde	Phenethylamine	2.78	3.19	3.47	3.64
Tic Disorders	Phenethylamine	2.78	2.08	1.28	0.52

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
ADHD	Other	2.78	1.04	1.92	2.06
Bipolar Disorder	Tryptamine	2.73	2.15	3.85	8.06
SWS	Amphetamine	2.73	4.57	5.24	11.47
Status Epilepticus	Amphetamine	2.69	3.99	5.32	9.79
Tourette Syndrome	Cannabinoid	2.65	1.06	0.90	0.74
Consciousness Disorders	Amphetamine	2.60	3.28	6.01	7.17
Heroin Dependence	Other	2.60	2.54	2.28	1.74
Substance-Related Disorders	Cathinone	2.56	2.09	2.05	2.19
Substance-Related Disorders	Other	2.56	1.57	2.05	1.72
Epilepsies, Myoclonic	Phenethylamine	2.50	2.84	4.27	3.58
Psychoses, Substance-Induced	Cathinone	2.47	1.96	2.61	1.49
Mood Disorders	Phenethylamine	2.47	1.53	1.75	4.27
Mood Disorders	Other	2.47	1.53	1.05	1.71
Restless Legs Syndrome	Phenethylamine	2.38	3.43	3.80	6.51
Psychotic Disorders	Amphetamine	2.34	2.47	3.82	6.65
ADHD	Cathinone	2.31	3.94	3.67	5.46
Autistic Disorder	Other	2.27	1.68	2.64	2.56
Narcolepsy	Phenethylamine	2.22	6.80	10.07	8.68
Agoraphobia	Amphetamine	2.22	5.36	6.32	11.02
Disorders of Excessive Somnolence	Phenethylamine	2.22	3.54	2.33	1.98
Agoraphobia	Phenethylamine	2.22	2.68	1.72	1.88
Seizures, Febrile	Phenethylamine	2.22	1.72	1.13	1.06
Seizures, Febrile	Amphetamine	2.22	0.86	0.56	0.80
Obsessive-Compulsive Disorder	Other	2.17	2.29	3.45	3.74
Cognition Disorders	Amphetamine	2.16	3.57	5.16	9.67
Child Behavior Disorders	Amphetamine	2.15	2.20	3.78	5.33
Status Epilepticus	Phenethylamine	2.15	2.11	1.99	2.95
Child Behavior Disorders	Phenethylamine	2.15	1.32	1.16	2.02
Cataplexy	Phenethylamine	2.13	3.18	4.60	4.15
SWS	Other	2.12	2.29	2.30	2.56
Epilepsy	Phenethylamine	2.08	4.50	5.18	7.76
Epilepsy, Absence	Tryptamine	2.06	2.13	1.62	3.98
Epilepsy, Complex Partial	Cathinone	2.04	1.41	2.99	2.50
Depressive Disorder	Cathinone	2.01	1.77	2.30	3.07
Alzheimer Disease	Other	1.90	1.73	1.54	1.56
Alzheimer Disease	Cathinone	1.90	1.35	1.41	1.34
Dysthymic Disorder	Amphetamine	1.85	6.92	9.50	9.63
Alcoholism	Cathinone	1.77	1.06	0.71	1.01

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
PTSD	Cathinone	1.74	1.22	1.12	1.17
Affective Disorders, Psychotic	Phenethylamine	1.72	2.94	1.94	3.13
Affective Disorders, Psychotic	Cathinone	1.72	1.47	1.46	1.20
Schizophrenia, Paranoid	Amphetamine	1.71	1.19	1.68	2.23
Learning Disorders	Amphetamine	1.69	2.48	2.84	5.93
Epilepsy, Generalized	Phenethylamine	1.69	1.42	0.97	0.91
Epilepsy, Generalized	Amphetamine	1.69	0.71	0.49	0.45
Personality Disorders	Phenethylamine	1.66	5.06	4.95	6.09
Status Epilepticus	Tryptamine	1.61	1.64	1.50	2.85
Stereotypic Movement Disorder	Phenethylamine	1.60	1.62	1.71	1.92
Stereotypic Movement Disorder	Cathinone	1.60	1.29	1.07	1.60
ADHD	Amphetamine	1.54	3.23	4.16	10.9
Seizures	Other	1.53	2.15	3.26	4.79
Anxiety Disorders	Other	1.52	2.07	2.66	4.21
Erectile Dysfunction	Other	1.45	0.92	1.84	2.08
Dementia	Other	1.45	0.55	0.34	0.75
Delirium	Amphetamine	1.41	4.30	4.21	6.80
Delirium	Other	1.41	2.15	2.46	1.82
Delirium	Phenethylamine	1.41	1.08	1.75	2.65
Amphetamine Disorders	Amphetamine	1.40	3.60	5.82	8.12
Psychotic Disorders	Tryptamine	1.40	2.47	4.46	12.11
Psychotic Disorders	Phenethylamine	1.40	2.02	1.75	3.28
Amphetamine Disorders	Phenethylamine	1.40	1.80	1.94	3.71
Schizophrenia	Tryptamine	1.38	2.70	4.00	9.50
Autistic Disorder	Cathinone	1.36	2.53	3.52	5.04
Seizures	Cathinone	1.34	2.54	3.19	5.59
Epilepsy, Temporal Lobe	Phenethylamine	1.33	3.49	2.53	3.11
Epilepsy, Temporal Lobe	Cathinone	1.33	1.16	2.11	1.86
Epilepsy, Temporal Lobe	Amphetamine	1.33	1.16	0.84	4.97
Cocaine-Related Disorders	Other	1.32	2.90	3.67	4.44
Consciousness Disorders	Phenethylamine	1.30	1.64	1.77	2.80
Consciousness Disorders	Other	1.30	1.09	2.12	2.02
Heroin Dependence	Cannabinoid	1.30	0.51	1.30	1.02
Substance-Related Disorders	Amphetamine	1.28	0.52	1.03	2.82
Substance-Related Disorders	Cannabinoid	1.28	0.52	1.03	0.47
Epilepsies, Myoclonic	Other	1.25	2.84	2.49	2.39
Epilepsy	Cathinone	1.25	1.31	1.86	2.48
Epilepsies, Myoclonic	Cathinone	1.25	1.14	2.49	3.88
Epilepsy	Cannabinoid	1.25	0.56	0.53	0.47

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Mood Disorders	Amphetamine	1.23	5.61	7.72	12.8
Psychoses, Substance-Induced	Other	1.23	1.47	2.61	1.64
Psychoses, Substance-Induced	Phenethylamine	1.23	0.98	1.31	2.08
SWS	Cathinone	1.21	1.43	2.30	3.47
Seizures	Phenethylamine	1.15	2.15	3.49	7.61
Huntington Disease	Phenethylamine	1.14	1.34	1.12	1.59
Learning Disorders	Other	1.13	2.48	2.51	2.37
Amnesia	Other	1.12	1.51	1.49	2.31
Conduct Disorder	Cathinone	1.10	1.46	2.30	3.59
Conduct Disorder	Other	1.10	0.98	1.64	2.77
Obsessive-Compulsive Disorder	Phenethylamine	1.09	2.75	2.19	3.27
Sleep Disorders	Tryptamine	1.09	1.49	3.62	7.42
Sleep Disorders	Other	1.09	1.49	3.29	3.87
Sleep Disorders	Phenethylamine	1.09	1.49	1.64	1.94
Sleep Disorders	Amphetamine	1.09	1.00	0.99	2.26
Status Epilepticus	Other	1.08	1.41	1.99	1.81
Child Behavior Disorders	Other	1.08	0.44	1.74	1.87
Cataplexy	Cathinone	1.06	2.27	3.07	6.45
Epilepsy, Absence	Amphetamine	1.03	2.98	4.05	6.80
Epilepsy, Absence	Cathinone	1.03	0.85	1.35	1.16
Epilepsy, Absence	Other	1.03	0.43	1.08	1.03
Epilepsy, Complex Partial	Phenethylamine	1.02	3.76	2.99	4.51
Alzheimer Disease	Tryptamine	0.95	3.85	6.94	13.1
Alzheimer Disease	Amphetamine	0.95	2.50	3.60	7.94
Psychotic Disorders	Other	0.93	1.35	3.66	4.37
Psychotic Disorders	Cathinone	0.93	0.90	1.43	2.09
Panic Disorder	Phenethylamine	0.91	2.52	2.08	4.86
SWS	Phenethylamine	0.91	1.86	2.41	4.59
Tourette Syndrome	Other	0.88	1.41	1.57	1.48
Epilepsy, Tonic-Clonic	Other	0.88	1.29	1.21	2.17
Epilepsy, Tonic-Clonic	Cathinone	0.88	0.43	1.51	1.55
PTSD	Phenethylamine	0.87	2.44	1.68	3.22
PTSD	Amphetamine	0.87	1.22	1.12	2.34
Schizophrenia, Paranoid	Phenethylamine	0.85	1.58	1.12	1.93
Stereotypic Movement Disorder	Amphetamine	0.80	2.91	2.78	5.97
Depressive Disorder, Major	Tryptamine	0.78	3.17	6.40	12.6
Epilepsies, Partial	Phenethylamine	0.78	3.03	2.07	2.52
Depressive Disorder, Major	Other	0.78	2.80	3.27	4.01
Epilepsies, Partial	Cathinone	0.78	1.14	1.55	1.59

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Bipolar Disorder	Other	0.78	1.08	1.15	1.96
Epilepsies, Partial	Amphetamine	0.78	0.76	0.52	3.45
Bipolar Disorder	Cathinone	0.78	0.54	1.54	2.64
Schizophrenia	Other	0.69	1.62	2.67	2.80
Schizophrenia	Cathinone	0.69	0.72	0.93	2.64
SIMD	Other	0.68	2.51	3.99	3.20
Cocaine-Related Disorders	Phenethylamine	0.66	2.67	3.75	6.28
Learning Disorders	Phenethylamine	0.56	0.99	1.67	2.28
Personality Disorders	Other	0.55	1.27	1.37	2.36
Cognition Disorders	Other	0.54	2.46	2.66	1.87
Status Epilepticus	Cathinone	0.54	1.41	1.83	2.57
Alzheimer Disease	Phenethylamine	0.47	1.92	2.70	5.05
Amphetamine Disorders	Other	0.47	0.90	1.13	1.77
Epilepsy	Other	0.42	0.94	1.59	2.48
Depressive Disorder, Major	Phenethylamine	0.39	2.99	3.13	6.57
Bipolar Disorder	Phenethylamine	0.39	1.80	1.79	3.84
Depressive Disorder, Major	Cannabinoid	0.39	0.19	0.41	0.24
Anxiety Disorders	Phenethylamine	0.38	2.41	2.30	5.41
Schizophrenia	Phenethylamine	0.34	1.62	2.00	2.64
SWS	Tryptamine	0.30	2.14	3.77	8.98
SWS	Cannabinoid	0.30	0.14	0.31	0.33
Cocaine-Related Disorders	Cannabinoid	0.22	0.11	0.34	0.58
Auditory Perceptual Disorders	Amphetamine	0.00	29.17	28.21	37.5
Binge-Eating Disorder	Phenethylamine	0.00	12.50	22.73	13.5
Gambling	Amphetamine	0.00	12.50	7.69	26.3
Disorders of Sex Development	Other	0.00	8.33	5.13	2.02
Speech Disorders	Tryptamine	0.00	6.25	11.54	15.2
Alcohol-Related Disorders	Other	0.00	6.25	6.58	4.05
Amnesia, Anterograde	Tryptamine	0.00	5.32	9.03	15.2
Heroin Dependence	Tryptamine	0.00	4.57	5.86	6.68
Alcohol-Related Disorders	Tryptamine	0.00	4.17	9.21	8.09
Depression, Postpartum	Other	0.00	4.17	7.69	3.03
Myoclonic Epilepsy, Juvenile	Amphetamine	0.00	4.17	5.41	16.1
Dementia, Vascular	Phenethylamine	0.00	4.17	5.13	4.55
Neurotic Disorders	Tryptamine	0.00	4.17	5.13	2.87
Myoclonic Epilepsy, Juvenile	Phenethylamine	0.00	4.17	4.05	7.22
Dyspareunia	Amphetamine	0.00	4.17	2.56	5.05
Dementia, Multi-Infarct	Tryptamine	0.00	4.17	2.56	4.04
Epilepsy, Reflex	Cathinone	0.00	4.17	2.56	3.03

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Pick Disease of the Brain	Amphetamine	0.00	4.17	2.56	3.03
Psychoses, Substance-Induced	Amphetamine	0.00	3.43	3.59	6.10
Morphine Dependence	Phenethylamine	0.00	3.37	2.48	1.65
Tourette Syndrome	Amphetamine	0.00	3.17	2.92	6.86
Neurotic Disorders	Cathinone	0.00	2.78	2.56	2.51
Consciousness Disorders	Tryptamine	0.00	2.73	6.36	8.72
Opioid-Related Disorders	Tryptamine	0.00	2.73	5.78	7.61
Disorders of Excessive Somnolence	Tryptamine	0.00	2.65	5.81	10.6
Substance-Related Disorders	Tryptamine	0.00	2.62	2.74	4.70
Phobic Disorders	Other	0.00	2.61	2.91	2.45
Spasms, Infantile	Phenethylamine	0.00	2.59	1.76	0.80
Psychoses, Substance-Induced	Tryptamine	0.00	2.45	4.25	9.08
Delirium	Tryptamine	0.00	2.15	4.91	9.12
Amnesia, Anterograde	Other	0.00	2.13	3.47	1.52
Epilepsy, Absence	Phenethylamine	0.00	2.13	1.62	3.47
Alcoholism	Phenethylamine	0.00	2.12	1.42	2.53
Hypochondriasis	Tryptamine	0.00	2.08	2.56	4.04
Alcohol-Related Disorders	Amphetamine	0.00	2.08	1.32	2.31
Heroin Dependence	Phenethylamine	0.00	2.03	1.30	1.31
Restless Legs Syndrome	Other	0.00	1.96	1.58	1.55
Tobacco Use Disorder	Tryptamine	0.00	1.94	2.58	8.35
Agoraphobia	Other	0.00	1.79	1.72	1.61
Huntington Disease	Amphetamine	0.00	1.79	1.40	5.16
Depressive Disorder	Phenethylamine	0.00	1.77	1.84	5.52
PTSD	Other	0.00	1.63	3.35	3.36
Personality Disorders	Tryptamine	0.00	1.52	2.73	5.36
Affective Disorders, Psychotic	Other	0.00	1.47	2.91	2.41
Sleep Bruxism	Phenethylamine	0.00	1.47	1.83	3.46
Sleep Bruxism	Other	0.00	1.47	0.92	0.77
Conduct Disorder	Phenethylamine	0.00	1.46	2.30	2.28
Tourette Syndrome	Tryptamine	0.00	1.41	2.02	6.12
Tourette Syndrome	Phenethylamine	0.00	1.41	1.57	2.75
Neurotic Disorders	Amphetamine	0.00	1.39	5.13	7.53
Anorexia Nervosa	Phenethylamine	0.00	1.39	1.71	3.03
Anorexia Nervosa	Other	0.00	1.39	0.85	1.01
Alcoholic Intoxication	Phenethylamine	0.00	1.39	0.85	0.78
Huntington Disease	Tryptamine	0.00	1.34	3.35	7.94

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Sexual Dysfunctions, Psychological	Phenethylamine	0.00	1.32	1.12	1.44
Sexual Dysfunctions, Psychological	Amphetamine	0.00	1.32	0.84	3.66
Stereotypic Movement Disorder	Other	0.00	1.29	1.07	1.07
Tobacco Use Disorder	Amphetamine	0.00	1.29	1.72	6.11
Tobacco Use Disorder	Phenethylamine	0.00	1.29	0.86	1.02
Autistic Disorder	Phenethylamine	0.00	1.26	1.62	3.25
SIMD	Cathinone	0.00	1.25	1.26	1.17
Nocturnal Enuresis	Amphetamine	0.00	1.23	2.99	3.26
Child Development Disorders, Pervasive	Cathinone	0.00	1.18	2.86	3.00
Dementia	Tryptamine	0.00	1.09	3.03	6.31
Opioid-Related Disorders	Amphetamine	0.00	1.09	2.38	5.23
Opioid-Related Disorders	Phenethylamine	0.00	1.09	0.68	1.27
Developmental Disabilities	Phenethylamine	0.00	1.09	2.04	1.50
Alcoholism	Tryptamine	0.00	1.06	1.89	3.54
ADHD	Cathinone	0.00	1.04	2.56	2.57
Alcohol Withdrawal Seizures	Amphetamine	0.00	1.04	1.32	1.65
Pica	Tryptamine	0.00	1.04	1.28	4.36
Pica	Other	0.00	1.04	0.64	0.77
Mood Disorders	Tryptamine	0.00	1.02	1.40	6.84
Heroin Dependence	Amphetamine	0.00	1.02	1.30	3.63
Depressive Disorder	Other	0.00	0.97	1.84	3.42
Enuresis	Amphetamine	0.00	0.93	1.16	2.63
Enuresis	Phenethylamine	0.00	0.93	0.58	0.48
Disorders of Excessive Somnolence	Other	0.00	0.88	1.74	0.99
Sexual Dysfunctions, Psychological	Tryptamine	0.00	0.88	1.68	4.06
Child Behavior Disorders	Tryptamine	0.00	0.88	1.16	3.75
Phobic Disorders	Amphetamine	0.00	0.87	0.58	0.92
Phobic Disorders	Phenethylamine	0.00	0.87	0.58	0.31
Epilepsy, Tonic-Clonic	Phenethylamine	0.00	0.86	0.91	2.48
Dysthymic Disorder	Phenethylamine	0.00	0.77	1.00	2.75
Dysthymic Disorder	Other	0.00	0.77	0.50	1.83
Impulse Control Disorders	Tryptamine	0.00	0.76	4.35	7.93
Impulse Control Disorders	Other	0.00	0.76	0.97	0.63
Alcoholism	Cannabinoid	0.00	0.71	0.71	0.76

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
SIMD	Amphetamine	0.00	0.63	1.68	2.99
Dementia	Phenethylamine	0.00	0.55	1.68	1.80
Consciousness Disorders	Cathinone	0.00	0.55	1.41	2.49
Sleep Disorders	Cathinone	0.00	0.50	0.66	1.13
Epilepsy, Complex Partial	Other	0.00	0.47	1.33	1.84
Huntington Disease	Other	0.00	0.45	1.40	1.19
Child Behavior Disorders	Cathinone	0.00	0.44	1.16	1.87
Panic Disorder	Other	0.00	0.42	2.08	3.95
Schizophrenia, Paranoid	Other	0.00	0.40	1.68	2.52
Epilepsy	Tryptamine	0.00	0.38	0.66	3.96
SIMD	Phenethylamine	0.00	0.31	0.63	1.81
Cognition Disorders	Phenethylamine	0.00	0.22	1.25	3.05
Motor Skills Disorders	Phenethylamine	0.00	0.00	5.13	13.1
Dementia, Vascular	Tryptamine	0.00	0.00	5.13	10.6
Gambling	Tryptamine	0.00	0.00	5.13	7.07
Auditory Perceptual Disorders	Phenethylamine	0.00	0.00	5.13	6.06
Motor Skills Disorders	Amphetamine	0.00	0.00	5.13	3.03
Binge-Eating Disorder	Other	0.00	0.00	4.55	2.38
Borderline Personality Disorder	Other	0.00	0.00	3.53	2.13
Nocturnal Enuresis	Tryptamine	0.00	0.00	2.99	3.56
Bulimia Nervosa	Tryptamine	0.00	0.00	2.96	4.64
PTSD	Tryptamine	0.00	0.00	2.79	8.19
Gambling	Phenethylamine	0.00	0.00	2.56	4.04
Dementia, Multi-Infarct	Other	0.00	0.00	2.56	3.03
Dementia, Multi-Infarct	Amphetamine	0.00	0.00	2.56	2.02
Trichotillomania	Tryptamine	0.00	0.00	2.56	2.02
Psychoses, Alcoholic	Cathinone	0.00	0.00	2.56	1.01
Binge-Eating Disorder	Cathinone	0.00	0.00	2.27	11.90
Schizophrenia, Catatonic	Other	0.00	0.00	2.13	1.38
Developmental Disabilities	Tryptamine	0.00	0.00	2.04	3.59
Epilepsy, Rolandic	Other	0.00	0.00	1.83	2.33
Anorexia Nervosa	Cathinone	0.00	0.00	1.71	3.03
Eating Disorders	Phenethylamine	0.00	0.00	1.71	1.48
Panic Disorder	Tryptamine	0.00	0.00	1.49	6.22
Amnesia, Anterograde	Amphetamine	0.00	0.00	1.39	2.73
Alcohol Withdrawal Delirium	Other	0.00	0.00	1.38	1.49
REM Sleep Behavior Disorder	Other	0.00	0.00	1.30	2.86
Hypochondriasis	Cannabinoid	0.00	0.00	1.28	1.01
Speech Disorders	Other	0.00	0.00	1.28	0.51

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Epilepsy, Temporal Lobe	Other	0.00	0.00	1.27	2.07
Alcoholism	Other	0.00	0.00	1.18	1.39
Seizures, Febrile	Other	0.00	0.00	1.13	2.93
Epilepsy, Generalized	Other	0.00	0.00	0.97	1.59
Epilepsy, Rolandic	Tryptamine	0.00	0.00	0.92	2.71
Neurotic Disorders	Phenethylamine	0.00	0.00	0.85	2.51
Eating Disorders	Other	0.00	0.00	0.85	1.11
Neurotic Disorders	Other	0.00	0.00	0.85	1.08
Epilepsies, Partial	Other	0.00	0.00	0.78	1.46
Bulimia Nervosa	Other	0.00	0.00	0.74	1.66
Substance-Related Disorders	Phenethylamine	0.00	0.00	0.68	0.63
Alcohol Withdrawal Seizures	Tryptamine	0.00	0.00	0.66	2.75
Alcohol Withdrawal Seizures	Cathinone	0.00	0.00	0.66	0.83
Pica	Phenethylamine	0.00	0.00	0.64	3.08
Pica	Cannabinoid	0.00	0.00	0.64	0.51
Spasms, Infantile	Other	0.00	0.00	0.59	1.59
Phobic Disorders	Tryptamine	0.00	0.00	0.58	1.84
Agoraphobia	Tryptamine	0.00	0.00	0.57	1.61
Seizures, Febrile	Tryptamine	0.00	0.00	0.56	3.19
Affective Disorders, Psychotic	Tryptamine	0.00	0.00	0.49	2.65
Epilepsy, Generalized	Tryptamine	0.00	0.00	0.49	1.59
Impulse Control Disorders	Cannabinoid	0.00	0.00	0.48	0.63
Impulse Control Disorders	Amphetamine	0.00	0.00	0.48	0.42
Tobacco Use Disorder	Cannabinoid	0.00	0.00	0.43	0.81
Epilepsy, Temporal Lobe	Tryptamine	0.00	0.00	0.42	2.48
Opioid-Related Disorders	Cannabinoid	0.00	0.00	0.34	0.63
Dementia	Cannabinoid	0.00	0.00	0.34	0.30
Epilepsy, Complex Partial	Tryptamine	0.00	0.00	0.33	2.67
Psychoses, Substance-Induced	Cannabinoid	0.00	0.00	0.33	0.45
Obsessive-Compulsive Disorder	Tryptamine	0.00	0.00	0.31	3.12
Epilepsy, Tonic-Clonic	Tryptamine	0.00	0.00	0.30	2.17
Schizophrenia, Paranoid	Tryptamine	0.00	0.00	0.28	2.97
Schizophrenia, Paranoid	Cathinone	0.00	0.00	0.28	0.74
Epilepsy, Absence	Cannabinoid	0.00	0.00	0.27	0.26
Epilepsies, Partial	Tryptamine	0.00	0.00	0.26	2.12
Amnesia	Cannabinoid	0.00	0.00	0.11	0.20
Dyspareunia	Phenethylamine	0.00	0.00	0.00	6.57
Myoclonic Epilepsy, Juvenile	Tryptamine	0.00	0.00	0.00	5.00
Narcolepsy	Tryptamine	0.00	0.00	0.00	4.51

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Enuresis	Tryptamine	0.00	0.00	0.00	3.83
Dysthymic Disorder	Tryptamine	0.00	0.00	0.00	3.67
Tic Disorders	Amphetamine	0.00	0.00	0.00	3.35
Conduct Disorder	Tryptamine	0.00	0.00	0.00	3.26
Gambling	Cathinone	0.00	0.00	0.00	3.03
Pick Disease of the Brain	Tryptamine	0.00	0.00	0.00	3.03
Borderline Personality Disorder	Tryptamine	0.00	0.00	0.00	2.66
Binge-Eating Disorder	Tryptamine	0.00	0.00	0.00	2.38
Schizophrenia, Childhood	Tryptamine	0.00	0.00	0.00	2.06
Paraphilias	Tryptamine	0.00	0.00	0.00	2.04
Lewy Body Disease	Tryptamine	0.00	0.00	0.00	2.03
Dyspareunia	Other	0.00	0.00	0.00	2.02
Fetishism (Psychiatric)	Phenethylamine	0.00	0.00	0.00	2.02
Jet Lag Syndrome	Phenethylamine	0.00	0.00	0.00	2.02
Seasonal Affective Disorder	Phenethylamine	0.00	0.00	0.00	2.02
Sleep Disorders, Circadian Rhythm	Phenethylamine	0.00	0.00	0.00	2.02
Sleep-Wake Transition Disorders	Tryptamine	0.00	0.00	0.00	2.02
Eating Disorders	Tryptamine	0.00	0.00	0.00	1.85
Child Development Disorders, Pervasive	Tryptamine	0.00	0.00	0.00	1.80
Schizophrenia, Catatonic	Tryptamine	0.00	0.00	0.00	1.72
Morphine Dependence	Tryptamine	0.00	0.00	0.00	1.65
Dyspareunia	Tryptamine	0.00	0.00	0.00	1.52
Child Development Disorders, Pervasive	Phenethylamine	0.00	0.00	0.00	1.50
Impulse Control Disorders	Phenethylamine	0.00	0.00	0.00	1.46
ADHD	Phenethylamine	0.00	0.00	0.00	1.29
Alcoholism	Amphetamine	0.00	0.00	0.00	1.27
Nocturnal Enuresis	Other	0.00	0.00	0.00	1.19
Alcoholic Intoxication	Amphetamine	0.00	0.00	0.00	1.17
Schizophrenia, Disorganized	Phenethylamine	0.00	0.00	0.00	1.16
Alcohol Withdrawal Seizures	Phenethylamine	0.00	0.00	0.00	1.10
Myoclonic Epilepsies, Progressive	Other	0.00	0.00	0.00	1.03
Aphasia, Broca	Other	0.00	0.00	0.00	1.01
Combat Disorders	Phenethylamine	0.00	0.00	0.00	1.01
Dementia, Multi-Infarct	Phenethylamine	0.00	0.00	0.00	1.01
Dementia, Vascular	Cathinone	0.00	0.00	0.00	1.01

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Depression, Postpartum	Tryptamine	0.00	0.00	0.00	1.01
Fetishism (Psychiatric)	Tryptamine	0.00	0.00	0.00	1.01
Jet Lag Syndrome	Amphetamine	0.00	0.00	0.00	1.01
Jet Lag Syndrome	Other	0.00	0.00	0.00	1.01
Motor Skills Disorders	Tryptamine	0.00	0.00	0.00	1.01
Pick Disease of the Brain	Other	0.00	0.00	0.00	1.01
Psychoses, Alcoholic	Phenethylamine	0.00	0.00	0.00	1.01
Schizotypal Personality	Other	0.00	0.00	0.00	1.01
Seasonal Affective Disorder	Amphetamine	0.00	0.00	0.00	1.01
Seasonal Affective Disorder	Other	0.00	0.00	0.00	1.01
Sleep Deprivation	Other	0.00	0.00	0.00	1.01
Sleep Disorders, Circadian Rhythm	Amphetamine	0.00	0.00	0.00	1.01
Sleep Disorders, Circadian Rhythm	Other	0.00	0.00	0.00	1.01
Speech Disorders	Amphetamine	0.00	0.00	0.00	1.01
Wernicke Encephalopathy	Phenethylamine	0.00	0.00	0.00	1.01
Enuresis	Other	0.00	0.00	0.00	0.96
Schizophrenia and Psychotic Disorder	Phenethylamine	0.00	0.00	0.00	0.94
Nocturnal Enuresis	Phenethylamine	0.00	0.00	0.00	0.89
Alcohol Withdrawal Seizures	Other	0.00	0.00	0.00	0.83
Disorders of Excessive Somnolence	Cathinone	0.00	0.00	0.00	0.74
Asperger Syndrome	Amphetamine	0.00	0.00	0.00	0.67
Asperger Syndrome	Phenethylamine	0.00	0.00	0.00	0.67
Asperger Syndrome	Tryptamine	0.00	0.00	0.00	0.67
Intellectual Disability	Amphetamine	0.00	0.00	0.00	0.67
Intellectual Disability	Phenethylamine	0.00	0.00	0.00	0.67
Intellectual Disability	Tryptamine	0.00	0.00	0.00	0.67
Huntington Disease	Cathinone	0.00	0.00	0.00	0.66
Nocturnal Enuresis	Cathinone	0.00	0.00	0.00	0.59
Schizophrenia, Disorganized	Amphetamine	0.00	0.00	0.00	0.58
Schizophrenia, Disorganized	Cathinone	0.00	0.00	0.00	0.58
Myoclonic Epilepsy, Juvenile	Other	0.00	0.00	0.00	0.56
Sexual Dysfunctions, Psychological	Cathinone	0.00	0.00	0.00	0.52
Schizophrenia, Childhood	Other	0.00	0.00	0.00	0.52

continued on next page

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Myoclonic Epilepsies, Progressive	Cathinone	0.00	0.00	0.00	0.51
Paraphilias	Other	0.00	0.00	0.00	0.51
Lewy Body Disease	Other	0.00	0.00	0.00	0.51
Frontotemporal Dementia	Phenethylamine	0.00	0.00	0.00	0.51
Frontotemporal Dementia	Tryptamine	0.00	0.00	0.00	0.51
Hypochondriasis	Amphetamine	0.00	0.00	0.00	0.51
Hypochondriasis	Phenethylamine	0.00	0.00	0.00	0.51
Neonatal Abstinence Syndrome	Tryptamine	0.00	0.00	0.00	0.51
Disorders of Excessive Somnolence	Amphetamine	0.00	0.00	0.00	0.50
Schizophrenia and Psychotic Disorder	Amphetamine	0.00	0.00	0.00	0.47
Schizophrenia and Psychotic Disorder	Tryptamine	0.00	0.00	0.00	0.47
Alcohol Withdrawal Delirium	Phenethylamine	0.00	0.00	0.00	0.43
Morphine Dependence	Cannabinoid	0.00	0.00	0.00	0.41
Sexual Dysfunctions, Psychological	Other	0.00	0.00	0.00	0.39
Alcoholic Intoxication	Other	0.00	0.00	0.00	0.39
Schizophrenia, Catatonic	Phenethylamine	0.00	0.00	0.00	0.34
Anorexia Nervosa	Tryptamine	0.00	0.00	0.00	0.34
Amnesia, Anterograde	Cannabinoid	0.00	0.00	0.00	0.30
Seizures, Febrile	Cathinone	0.00	0.00	0.00	0.27
Disorders of Excessive Somnolence	Cannabinoid	0.00	0.00	0.00	0.25
Enuresis	Cathinone	0.00	0.00	0.00	0.24
Epilepsy, Generalized	Cathinone	0.00	0.00	0.00	0.23
Alzheimer Disease	Cannabinoid	0.00	0.00	0.00	0.22
Alcohol Withdrawal Delirium	Cathinone	0.00	0.00	0.00	0.21
Erectile Dysfunction	Cannabinoid	0.00	0.00	0.00	0.21
Amphetamine Disorders	Cannabinoid	0.00	0.00	0.00	0.18
Autistic Disorder	Cannabinoid	0.00	0.00	0.00	0.17
Cognition Disorders	Cannabinoid	0.00	0.00	0.00	0.17
Delirium	Cannabinoid	0.00	0.00	0.00	0.17
Consciousness Disorders	Cannabinoid	0.00	0.00	0.00	0.16
Cataplexy	Cannabinoid	0.00	0.00	0.00	0.15
Stereotypic Movement Disorder	Cannabinoid	0.00	0.00	0.00	0.11
Status Epilepticus	Cannabinoid	0.00	0.00	0.00	0.10

continued on next page

Table B.4.
One-tailed KS-Test p-values for the statistical tests. The alternative hypothesis for all tests that the distribution tested has a greater cumulative distribution function than the randomized distributions.

	Top 10	Top 25	Top 40	Top100
Normalized indication rank against randomized indications	1.622e-14	7.392e-11	4.030e-10	4.449e-08
Normalized indication rank against randomized compounds	1.805e-35	1.805e-35	1.694e-29	8.553e-21
Normalized compound rank against randomized indications	1.960e-06	6.955e-10	2.445e-15	8.796e-13
Normalized compound rank against randomized compounds	1.504e-07	7.772e-17	1.476e-23	4.665e-26

Table B.3.: *continued*

Indication	Category	Top 10	Top 25	Top 40	Top 100
Learning Disorders	Cannabinoid	0.00	0.00	0.00	0.09
Schizophrenia	Cannabinoid	0.00	0.00	0.00	0.08
Bipolar Disorder	Cannabinoid	0.00	0.00	0.00	0.08

Table B.5.
One-tailed paired T-Test p-values for the statistical tests introduced in Fig 1.2. The alternative hypothesis for all tests that the distribution tested has a greater cumulative distribution function than the randomized distributions.

	Top 10	Top 25	Top 40	Top100
Normalized compound rank against randomized indications	7.846e-12	3.121e-18	2.250e-21	8.811e-37
Normalized compound rank against randomized compounds	1.397e-08	1.345e-10	1.904e-08	8.247e-16

Table B.6.
Indication-Indication association counts for the Top10 predictions.

Indication 1	Indication 2	Association
Depressive Disorder, Major	Binge-Eating Disorder	2
Personality Disorders	Binge-Eating Disorder	2
Depressive Disorder, Major	Bipolar Disorder	2
Epilepsies, Myoclonic	Bipolar Disorder	2
Amnesia	Cataplexy	3
Depressive Disorder, Major	Cocaine-Related Disorders	5
Restless Legs Syndrome	Cocaine-Related Disorders	3
Seizures	Cocaine-Related Disorders	2
Anxiety Disorders	Depressive Disorder, Major	2
Epilepsies, Myoclonic	Depressive Disorder, Major	2
Amnesia	Epilepsy	2
Amnesia	Narcolepsy	2
Cataplexy	Narcolepsy	2
Epilepsies, Myoclonic	Narcolepsy	2
Bipolar Disorder	Personality Disorders	4
Depressive Disorder, Major	Personality Disorders	3
Epilepsies, Myoclonic	Personality Disorders	2
Depressive Disorder, Major	Seizures	2
Epilepsies, Myoclonic	Seizures	2
Cocaine-Related Disorders	Substance Withdrawal Syndrome	3

Table B.7.
Indication-Indication association counts for the Top25 predictions

Indication 1	Indication 2	Assoc.
Amnesia	Anxiety Disorders	2
Depressive Disorder, Major	Anxiety Disorders	2
Cocaine-Related Disorders	ADHD	2
Depressive Disorder, Major	Cocaine-Related Disorders	10
Epilepsies, Myoclonic	Cocaine-Related Disorders	7
Narcolepsy	Cocaine-Related Disorders	2
Seizures	Cocaine-Related Disorders	9
Depressive Disorder, Major	Epilepsies, Myoclonic	7
Depressive Disorder, Major	Narcolepsy	3
Epilepsies, Myoclonic	Narcolepsy	3
Depressive Disorder, Major	Seizures	8
Epilepsies, Myoclonic	Seizures	10
Narcolepsy	Seizures	2
Seizures	SIMD	2
Substance Withdrawal Syndrome	SIMD	2
Cocaine-Related Disorders	Substance Withdrawal Syndrome	3
Seizures	Substance Withdrawal Syndrome	4

Table B.8.
Indication-Indication association counts for the Top40 predictions

Indication 1	Indication 2	Assoc.
Cocaine-Related Disorders	Alzheimer Disease	2
Seizures	Anxiety Disorders	2
Cocaine-Related Disorders	ADHD	2
Anxiety Disorders	Cocaine-Related Disorders	2
Depressive Disorder, Major	Cocaine-Related Disorders	10
Seizures	Cocaine-Related Disorders	13
Anxiety Disorders	Depressive Disorder, Major	4
Seizures	Depressive Disorder, Major	9
Cocaine-Related Disorders	Learning Disorders	3
Seizures	Learning Disorders	3
Substance Withdrawal Syndrome	Learning Disorders	3
Depressive Disorder, Major	Narcolepsy	2
Cocaine-Related Disorders	Psychotic Disorders	3
Depressive Disorder, Major	Psychotic Disorders	2
Seizures	Psychotic Disorders	2
Anxiety Disorders	Schizophrenia	2
Cocaine-Related Disorders	Schizophrenia	2
Depressive Disorder, Major	Schizophrenia	2
Psychotic Disorders	Schizophrenia	2
Seizures	Schizophrenia	2
SIMDs	Schizophrenia	2
Cocaine-Related Disorders	SIMD	2
Depressive Disorder, Major	SIMD	2
Psychotic Disorders	SIMD	2
Seizures	SIMD	2
Cocaine-Related Disorders	Substance Withdrawal Syndrome	5
Depressive Disorder, Major	Substance Withdrawal Syndrome	2
Psychotic Disorders	Substance Withdrawal Syndrome	2
Schizophrenia	Substance Withdrawal Syndrome	2
Seizures	Substance Withdrawal Syndrome	5
SIMDs	Substance Withdrawal Syndrome	2

Table B.9.: Indication-Indication association counts for the Top1000 predictions

Indication 1	Indication 2	Assoc.
Amnesia	Alzheimer Disease	2
Cocaine-Related Disorders	Alzheimer Disease	3
Seizures	Alzheimer Disease	2
Cocaine-Related Disorders	Amnesia	4
Amnesia	Anxiety Disorders	2
Cocaine-Related Disorders	Anxiety Disorders	11
Psychotic Disorders	Anxiety Disorders	2
Cocaine-Related Disorders	ADHD	5
Seizures	ADHD	3
Amnesia	Autistic Disorder	3
Anxiety Disorders	Autistic Disorder	4
Cocaine-Related Disorders	Autistic Disorder	8
Anxiety Disorders	Depressive Disorder	2
Cocaine-Related Disorders	Depressive Disorder	4
Depressive Disorder, Major	Depressive Disorder	2
Psychotic Disorders	Depressive Disorder	2
Schizophrenia	Depressive Disorder	2
Seizures	Depressive Disorder	4
Substance Withdrawal Syndrome	Depressive Disorder	2
Anxiety Disorders	Depressive Disorder, Major	7
Autistic Disorder	Depressive Disorder, Major	2
Cocaine-Related Disorders	Depressive Disorder, Major	12
Psychotic Disorders	Depressive Disorder, Major	2
Seizures	Depressive Disorder, Major	6
Cocaine-Related Disorders	Psychotic Disorders	7
Anxiety Disorders	Schizophrenia	3
ADHD	Schizophrenia	2
Cocaine-Related Disorders	Schizophrenia	5

continued on next page

Table B.9.: *continued*

Indication 1	Indication 2	Assoc.
Psychotic Disorders	Schizophrenia	3
Seizures	Schizophrenia	4
Amnesia	Seizures	7
Anxiety Disorders	Seizures	9
Autistic Disorder	Seizures	6
Cocaine-Related Disorders	Seizures	23
Psychotic Disorders	Seizures	5
Anxiety Disorders	Sleep Disorders	2
Cocaine-Related Disorders	Sleep Disorders	2
Depressive Disorder	Sleep Disorders	2
Psychotic Disorders	Sleep Disorders	2
Schizophrenia	Sleep Disorders	2
Seizures	Sleep Disorders	2
SIMD	Sleep Disorders	2
Anxiety Disorders	SIMDs	2
Cocaine-Related Disorders	SIMD	2
Depressive Disorder	SIMDs	2
Psychotic Disorders	SIMD	2
Schizophrenia	SIMD	2
Seizures	SIMD	2
Cocaine-Related Disorders	Substance Withdrawal Syndrome	5
Depressive Disorder, Major	Substance Withdrawal Syndrome	2
Psychotic Disorders	Substance Withdrawal Syndrome	2
Schizophrenia	Substance Withdrawal Syndrome	2
Seizures	Substance Withdrawal Syndrome	5

C. ADDITIONAL FIGURES AND LISTINGS FOR CHAPTER 2.1

C.1 Results of Lemon Workflows

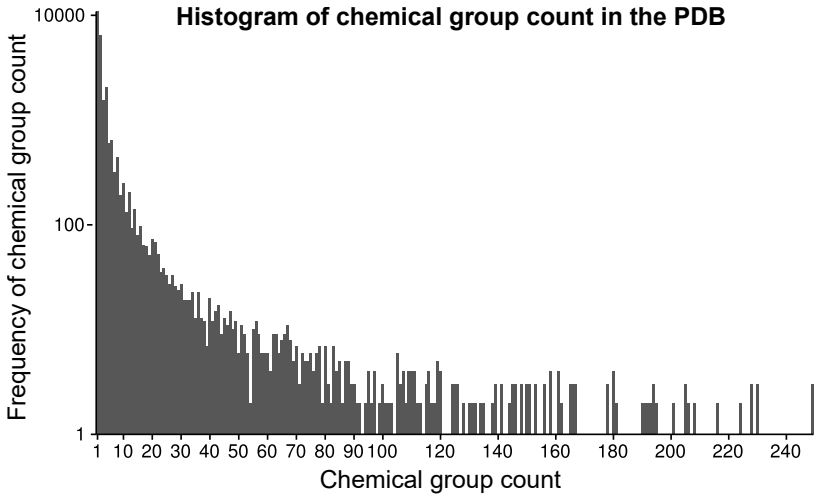


Figure C.1. Histogram showing the frequency of a given chemical group count (maximum of 250). The X-axis is the chemical group count. This count is independent of chemical environment is determined from the three-letter code given to chemical groups in the PDB. For example, if the residue 'CFF' occurs once in PDBID 142N and thrice in PDBID 1L59, and occurs nowhere else in the PDB, then it has a count of 4. The Y-axis gives the frequency for all chemical group counts in the PDB. From this data we can conclude that the majority of chemical groups occur only once in throughout the entire PDB.

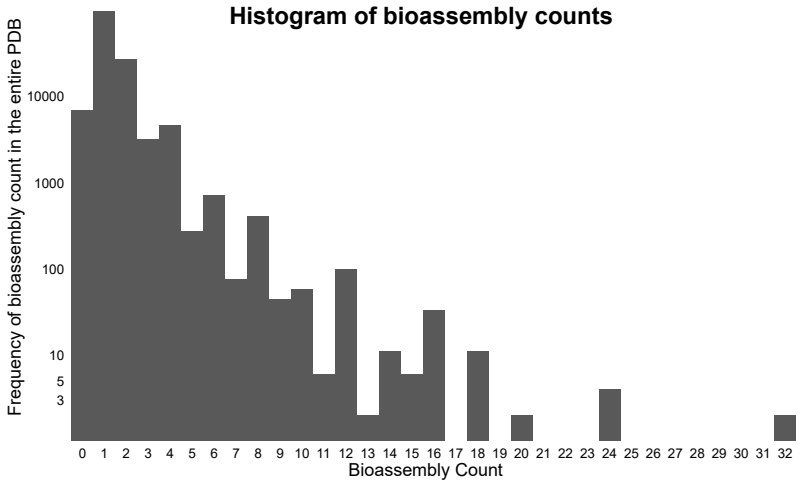


Figure C.2. Histogram showing the frequency of bioassemblies (as defined by the depositor of a PDB file) throughout the PDB.

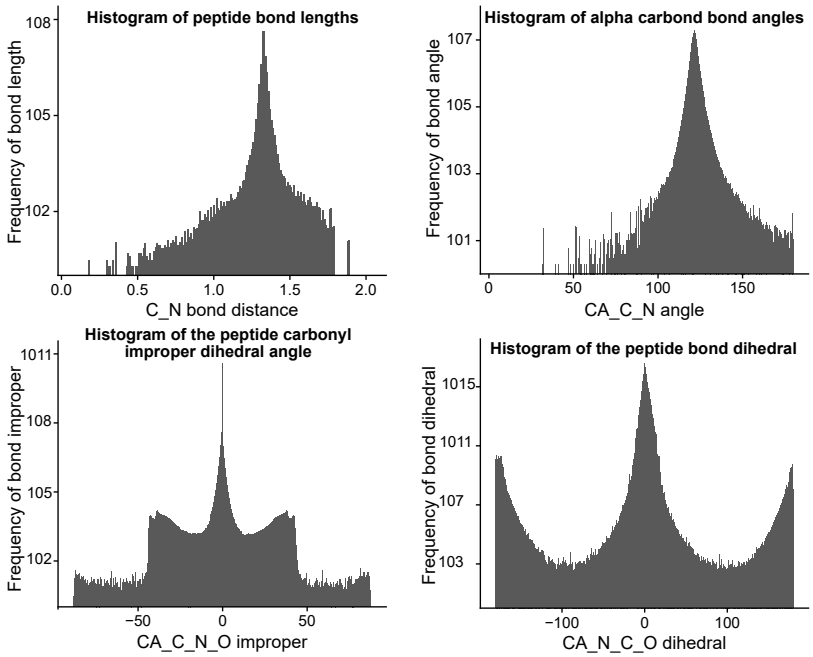


Figure C.3. Histograms of various geometries centered around the peptide bond. These plots illustrate Lemon's ability to mine geometrical data from the PDB.

C.2 Lemon Program Listings

C.2.1 Simple Workflows

Listing 1

C++ Lambda function to count the number of biological assemblies in the PDB. This example illustrates how to obtain information about a residue/group property (in this case symmetry) which could be used to determine if the user wishes to continue calculation.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);
    auto worker = [](chemfiles::Frame entry,
                     const std::string& pdbid) {
        // Desired info is obtained directly, no pruning
        auto result =
            lemon::count::residue_property(entry,
                                           "assembly");

        // Output phase
        return pdbid + "␣" + std::to_string(result) + "\n";
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 2

C++ Lambda function to determine the number of alternative atom locations in all PDB entries. This example illustrates the use of an atomic property to potentially screen entries which do not contain alternative locations.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);
    auto worker = [](chemfiles::Frame entry,
                     const std::string& pdbid) {
        // Desired info is obtained directly
        auto result =
            lemon::count::atom_property(entry, "altloc");

        // Output phase
```

```
        return pdbid + "␣" + std::to_string(result) + "\n";
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 3

C++ Lambda function to select metal ions in the PDB. This workflow shows how the selection phase of a Lemon workflow works by filling a generic STL container with the desired residue ids. The output is the pdbid followed by all the metal ions found in the corresponding entry.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);
    auto worker = [](chemfiles::Frame entry,
                     const std::string& pdbid) {

        // Selection phase
        std::list<size_t> metal_ids;
        lemon::select::metal_ions(entry, metal_ids);

        // No pruning, straight to output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 metal_ids);
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 4

C++ Lambda function to determine the occurrence of all residues in the PDB. The purpose of this workflow is to show that one can return more than strings from a C++ lambda function as long they use a different 'combine' function object to handle this return value. The concept of 'combine' functions is detailed in the online documentation along with this example to illustrate it. It outputs all three-letter residue names and the number of times each is found throughout all entries in the

PDB. Note that residues may occur multiple times in a single entry and this is reflected in this lambda.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);

    auto worker = [](chemfiles::Frame entry,
                     const std::string&) {
        // Desired info is calculated directly, no pruning,
        // output is done later
        lemon::ResidueNameCount rnc;
        lemon::count::residues(entry, rnc);
        return rnc;
    };

    lemon::ResidueNameCount resn_total;
    auto collector =
         lemon::map_combine<lemon::ResidueNameCount>
         (resn_total);
    lemon::launch(o, worker, collector);

    for (auto i : resn_total) {
        std::cout << i.first << "\t" << i.second << "\n";
    }
}
```

Listing 5

This example workflow combines concepts from the past two workflows to show that 'selection' can be combined with other workflow concepts via the separate functionality. Separate allows one to create a subset of an entry and perform further calculations on just the subset. This workflow is similar to above listing, but only prints residues with peptide linkage.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);
    auto worker = [](chemfiles::Frame entry,
                     const std::string&) {
        lemon::ResidueNameCount rnc;
        // Selection phase
        chemfiles::Frame protein_only;
        auto peptides = lemon::select::peptides(entry);

        if (peptides.size() == 0) {
```

```
            return rnc;
        }
        lemon::separate::residues(entry,
                                  peptides, protein_only);
        // Output phase
        lemon::count::residues(protein_only, rnc);
        return rnc;
    };
    lemon::ResidueNameCount resn_total;
    auto collector =
         lemon::map_combine<lemon::ResidueNameCount>
         (resn_total);
    lemon::launch(o, worker, collector);
    for (auto i : resn_total) {
        std::cout << i.first << "\t" << i.second << "\n";
    }
}
```

Listing 6

This workflow is designed to introduce pruning to the user. In this specific example, selected small molecules are pruned by removing common cofactors and common fatty acids. No detailed calculations are performed yet, but such calculations will be introduced in the next workflows.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);

    auto worker = [](chemfiles::Frame entry,
                     const std::string& pdbid) {

        // Selection phase
        std::list<size_t> smallm;
        if (lemon::select::small_molecules(entry,
                                           smallm) == 0) {
            return std::string("");
        }

        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
```

```
                                        lemon::common_fatty_acids);

        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

C.2.2 Distance–Based Workflows

Listing 7

C++ Lambda function to determine the number of small molecules which interact with a metal ion within a distance cutoff. This workflow is designed to show how to select two different groups and perform a distance-based pruning operation on the two groups. It also introduces the concept of obtaining command-line arguments.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
    o.parse_command_line(argc, argv);
    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto metals = lemon::select::metal_ions(entry);
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::keep_interactions(entry, smallm,
                                metals, distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
```

```
                                                    smallm);
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 8

C++ Lambda function to determine the number of small molecules which interact with a Heme group within a distance cutoff. This is similar to the last workflow and illustrates how selectors can be used to find cofactors instead of metal ion.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
    o.parse_command_line(argc, argv);
    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto hemegs = lemon::select::specific_residues(
            entry, {"HEM", "HEA", "HEB", "HEC"});
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::keep_interactions(entry, smallm,
                                        hemegs, distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 9

C++ Lambda function to determine the number of small molecules which interact with a SAM molecule within a distance cutoff. This workflow is similar in spirit to the previous one. It was written by request of a user interested in the interaction of ligands with this cofactor.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
    o.parse_command_line(argc, argv);
    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto sam =
             lemon::select::specific_residues(entry,
                                              {"SAM"});
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::keep_interactions(entry, smallm,
                                        sam, distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 10

C++ Lambda function to find small molecules which do not interact with any water molecules within a distance cutoff. Water is an important consideration when

predicting the pose of a ligand in a binding site and therefore many users may wish to find ligands which are within a given proximity to water.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
    o.parse_command_line(argc, argv);

    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto waters =
            lemon::select::specific_residues(entry, {"HOH"});
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::remove_interactions(entry, smallm,
                                          waters, distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 11

C++ Lambda function to find small molecules which interact with an amino acid chemical group. These interactions are crucial to developing small-molecule therapeutics and are thus of great important to the medicinal chemistry community.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
```

```
    o.parse_command_line(argc, argv);
    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto peptides = lemon::select::peptides(entry);
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::keep_interactions(entry, smallm,
                                        peptides, distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };

    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 12

C++ Lambda function to find small molecules which interact with a nucleic acid chemical group. This example was written by request from a user wishing to study the interactions between RNA and small-molecules.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto distance = 6.0;
    o.add_option("--distance,-d", distance,
                 "Largest distance");
    o.parse_command_line(argc, argv);
    auto worker = [distance](chemfiles::Frame entry,
                             const std::string& pdbid) {
        // Selection phase
        auto nucleic_acids =
             lemon::select::nucleic_acids(entry);
        auto smallm = lemon::select::small_molecules(entry);
        // Pruning phase
```

```
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        lemon::prune::keep_interactions(entry, smallm,
                                        nucleic_acids,
                                        distance);
        // Output phase
        return pdbid +
               lemon::count::print_residue_names(entry,
                                                 smallm);
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

C.2.3 Complex Workflows

Listing 13

Lemon C++ Workflow to align all structures to a given reference structure using the TMalign algorithm and print the corresponding scores.

```
int main(int argc, char* argv[]) {
    lemon::Options o;
    auto reference = std::string("reference.pdb");
    o.add_option("--reference,-r", reference,
                 "Protein or DNA to align to.")->
        check(CLI::ExistingFile);

    o.parse_command_line(argc, argv);
    chemfiles::Trajectory traj(reference);
    chemfiles::Frame native = traj.read();
    auto worker = [&native](chemfiles::Frame entry,
                            const std::string& pdbid) {
        std::vector<chemfiles::Vector3D> junk;
        auto tm = lemon::tmalign::TMscore(entry,
                                          native, junk);
        return pdbid + "\t" +
               std::to_string(tm.score) + "\t" +
               std::to_string(tm.rmsd) + "\t" +
               std::to_string(tm.aligned) + "\n";
```

```
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 14

Lemon C++ Workflow to calculate the docking score of all small-molecules with the surrounding environment using the scoring function published with AutoDOCK Vina.

```
int main(int argc, char* argv[]) {
    lemon::Options o(argc, argv);
    auto worker = [](chemfiles::Frame entry,
                     const std::string& pdbid) {
        // Selection phase
        std::list<size_t> smallm;
        if (lemon::select::small_molecules(entry,
                                           smallm) == 0) {
            return std::string("");
        }
        // Pruning phase
        lemon::prune::identical_residues(entry, smallm);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_cofactors);
        lemon::prune::cofactors(entry, smallm,
                                lemon::common_fatty_acids);
        // Output phase
        const auto& residues = entry.topology().residues();
        std::list<size_t> proteins;
        for (size_t i = 0;
             i < entry.topology().residues().size();
             ++i) {
            proteins.push_back(i);
        }
        std::string result;
        for (auto smallm_id : smallm) {
            auto prot_copy = proteins;
            lemon::prune::keep_interactions(entry, smallm,
                                            prot_copy, 8.0);
            prot_copy.erase(std::remove(prot_copy.begin(),
                                        prot_copy.end(),
                                        smallm_id));
```

```
            auto vscore =
                lemon::xscore::vina_score(entry, smallm_id,
                                          prot_copy);
            result += pdbid + "\t" +
                residues[smallm_id].name() + "\t" +
                std::to_string(vscore.g1) + "\t" +
                std::to_string(vscore.g2) + "\t" +
                std::to_string(vscore.hydrogen) + "\t" +
                std::to_string(vscore.hydrophobic) + "\t" +
                std::to_string(vscore.rep) + "\n";
        }
        return result;
    };
    auto collector = lemon::print_combine(std::cout);
    return lemon::launch(o, worker, collector);
}
```

Listing 15

C++ Lambda function to calculate all bond distances in the PDB.

```
// typedefs for binned data
typedef std::pair<std::string, size_t> BondStretchBin;
typedef std::map<BondStretchBin, size_t> StretchCounts;

using lemon::geometry::protein::bond_name;

int main(int argc, char* argv[]) {
    lemon::Options o;
    auto bin_size = 0.01;
    o.add_option("--bin_size,-b", bin_size,
                 "Size of the length(stretch) bin.");
    o.parse_command_line(argc, argv);

    auto worker = [bin_size](chemfiles::Frame entry,
                             const std::string& pdbid) {
        StretchCounts bins;
        // Selection phase
        chemfiles::Frame protein_only;
        std::list<size_t> peptides;
        if (lemon::select::specific_residues(
                entry, peptides,
```

```
                lemon::common_peptides) == 0) {
            return bins;
        }
        lemon::separate::residues(entry, peptides,
                                  protein_only);
        const auto& bonds = protein_only.topology().bonds();
        for (const auto& bond : bonds) {
            std::string bondnm;
            try {
                bondnm = bond_name(protein_only, bond);
            }
            catch (lemon::geometry::geometry_error& e) {
                auto msg = pdbid + ": " + e.what() + '\n';
                std::cerr << msg;
            }
            auto distance = protein_only.distance(bond[0],
                                                  bond[1]);
            size_t bin = static_cast<size_t>(
                             std::floor(distance /
                             bin_size));
            BondStretchBin sbin = {bondnm, bin};
            auto bin_iterator = bins.find(sbin);
            if (bin_iterator == bins.end()) {
                bins[sbin] = 1;
                continue;
            }
            ++(bin_iterator->second);
        }
        return bins;
    };
    StretchCounts sc_total;
    auto collector =
        lemon::map_combine<StretchCounts>(sc_total);
    lemon::launch(o, worker, collector);
    for (const auto& i : sc_total) {
        std::cout << i.first.first << "\t"
                  << static_cast<double>(i.first.second) *
                     bin_size << "\t"
                  << i.second << "\n";
    }
    return 0;
}
```

Listing 16

C++ Lambda function to calculate all bond angles in the PDB.

```
// typedefs for binned data
typedef std::pair<std::string, size_t> BondAngleBin;
typedef std::map<BondAngleBin, size_t> AngleCounts;

using lemon::geometry::protein::angle_name;

int main(int argc, char* argv[]) {
    lemon::Options o;
    auto bin_size = 0.01;
    o.add_option("--bin_size,-b", bin_size,
                 "Size of the angle bin.");
    o.parse_command_line(argc, argv);
    auto worker = [bin_size](chemfiles::Frame entry,
                             const std::string& pdbid) {
        AngleCounts bins;
        // Selection phase
        chemfiles::Frame protein_only;
        std::list<size_t> peptides;
        if (lemon::select::specific_residues(
                entry, peptides,
                lemon::common_peptides) == 0) {
            return bins;
        }
        lemon::separate::residues(entry, peptides,
                                  protein_only);
        const auto& angles =
              protein_only.topology().angles();
        for (const auto& angle : angles) {
            std::string anglenm;
            try {
                anglenm = angle_name(protein_only, angle);
            }
            catch (lemon::geometry::geometry_error& e) {
                auto msg = pdbid + ": " + e.what() + '\n';
                std::cerr << msg;
            }
            auto theta = protein_only.angle(angle[0],
                                            angle[1],
                                            angle[2]);
```

```
            size_t bin = static_cast<size_t>
                    (std::floor(theta / bin_size));
            BondAngleBin sbin = {anglenm, bin};
            auto bin_iterator = bins.find(sbin);
            if (bin_iterator == bins.end()) {
                bins[sbin] = 1;
                continue;
            }
            ++(bin_iterator->second);
        }
        return bins;
    };
    AngleCounts sc_total;
    auto collector =
         lemon::map_combine<AngleCounts>(sc_total);
    lemon::launch(o, worker, collector);
    for (const auto& i : sc_total) {
        std::cout << i.first.first << "\t"
                  << static_cast<double>(i.first.second) *
                     bin_size << "\t"
                  << i.second << "\n";
    }
    return 0;
}
```

Listing 17

C++ Lambda function to calculate all bond improper dihedrals in the PDB.

```
// typedefs for binned data
typedef std::pair<std::string, int> BondImproperBin;
typedef std::map<BondImproperBin, size_t> ImproperCounts;

using lemon::geometry::protein::improper_name;

int main(int argc, char* argv[]) {
    lemon::Options o;
    auto bin_size = 0.01;
    o.add_option("--bin_size,-b", bin_size,
                 "Size of the improper-dihedral bin.");
    o.parse_command_line(argc, argv);
    auto worker = [bin_size](chemfiles::Frame entry,
                             const std::string& pdbid) {
```

```
    ImproperCounts bins;
    // Selection phase
    chemfiles::Frame protein_only;
    std::list<size_t> peptides;
    if (lemon::select::specific_residues(
            entry, peptides,
            lemon::common_peptides) == 0) {
        return bins;
    }
    lemon::separate::residues(entry, peptides,
                              protein_only);
    protein_only.set_cell(entry.cell());
    const auto& impropers =
          protein_only.topology().impropers();
    for (const auto& improper : impropers) {
        std::string impropernm;
        try {
            impropernm = improper_name(protein_only,
                                       improper);
        }
        catch (lemon::geometry::geometry_error& e) {
            auto msg = pdbid + ": " + e.what() + '\n';
            std::cerr << msg;
        }
        auto theta = protein_only.out_of_plane(
            improper[0],
            improper[1],
            improper[2],
            improper[3]);

        int bin = static_cast<int>
           (std::floor(theta / bin_size));
        BondImproperBin sbin = {impropernm, bin};
        auto bin_iterator = bins.find(sbin);
        if (bin_iterator == bins.end()) {
            bins[sbin] = 1;
            continue;
        }
        ++(bin_iterator->second);
    }
    return bins;
};
ImproperCounts sc_total;
```

```
    auto collector =
        lemon::map_combine<ImproperCounts>(sc_total);

    lemon::launch(o, worker, collector);
    for (const auto& i : sc_total) {
        std::cout << i.first.first << "\t"
                  << static_cast<double>(i.first.second) *
                     bin_size << "\t"
                  << i.second << "\n";
    }
    return 0;
}
```

Listing 18

C++ Lambda function to calculate all bond dihedrals in the PDB.

```
// typedefs for binned data
typedef std::pair<std::string, int> BondDihedralBin;
typedef std::map<BondDihedralBin, size_t> DihedralCounts;

using lemon::geometry::protein::dihedral_name;

int main(int argc, char* argv[]) {
    lemon::Options o;
    auto bin_size = 0.01;
    o.add_option("--bin_size,-b", bin_size,
                 "Size of the dihedral bin.");
    o.parse_command_line(argc, argv);
    auto worker = [bin_size](chemfiles::Frame entry,
                             const std::string& pdbid) {
        DihedralCounts bins;

        // Selection phase
        chemfiles::Frame protein_only;
        std::list<size_t> peptides;
        if (lemon::select::specific_residues(
                entry, peptides,
                lemon::common_peptides) == 0) {
            return bins;
        }
        lemon::separate::residues(
                entry, peptides, protein_only);
```

```
    protein_only.set_cell(entry.cell());

    const auto& dihedrals =
          protein_only.topology().dihedrals();

    for (const auto& dihedral : dihedrals) {
        std::string dihedralnm;
        try {
            dihedralnm = dihedral_name(protein_only,
                                       dihedral);
        }
        catch (lemon::geometry::geometry_error& e) {
            auto msg = pdbid + ": " + e.what() + '\n';
            std::cerr << msg;
        }
        auto theta = protein_only.dihedral(dihedral[0],
                                           dihedral[1],
                                           dihedral[2],
                                           dihedral[3]);

        int bin = static_cast<int>
                  (std::floor(theta / bin_size));

        BondDihedralBin sbin = {dihedralnm, bin};
        auto bin_iterator = bins.find(sbin);
        if (bin_iterator == bins.end()) {
            bins[sbin] = 1;
            continue;
        }
        ++(bin_iterator->second);
    }
    return bins;
};
DihedralCounts sc_total;
auto collector = lemon::map_combine<DihedralCounts>
                 (sc_total);
lemon::launch(o, worker, collector);
for (const auto& i : sc_total) {
    std::cout << i.first.first << "\t"
              << static_cast<double>(i.first.second) *
                 bin_size << "\t"
              << i.second << "\n";
```

```
    }
    return 0;
}
```

C.2.4 Workflows written in Python

Listing 19

This workflow is a port of Listing 6 and is an example of a 'simple' workflow that includes the 'selection' and 'pruning' of chemical groups. It illustrates how easy converting between Python and C++ implementations of lemon can be if one follows the recommend workflow development pipeline.

```
import lemon

class MyWorkflow(lemon.Workflow):
    def worker(self, entry, pdbid):
        import lemon
        smallm = lemon.select_small_molecules(
            entry, lemon.small_molecule_types, 10)

        # Pruning phase
        lemon.prune_identical_residues(entry, smallm)
        lemon.prune_cofactors(entry, smallm,
                              lemon.common_cofactors)
        lemon.prune_cofactors(entry, smallm,
                              lemon.common_fatty_acids)

        # Output phase
        return pdbid +
               lemon.count_print_residue_names(entry,
                                               smallm)
```

Listing 20

This workflow is a Python port of Listing 10 and again illustrates the similarities shared between the C++ and Python APIs.

```
import lemon

class MyWorkflow(lemon.Workflow):
    def worker(self, entry, pdbid):
        import lemon
```

```
    wat_name = lemon.ResidueNameSet()
    wat_name.append(lemon.ResidueName("HOH"))
    waters = lemon.select_specific_residues(entry,
                                            wat_name)
    smallm = lemon.select_small_molecules(
        entry, lemon.small_molecule_types, 10)

    # Pruning phase
    lemon.prune_identical_residues(entry, smallm)
    lemon.prune_cofactors(entry, smallm,
                          lemon.common_cofactors)
    lemon.prune_cofactors(entry, smallm,
                          lemon.common_fatty_acids)
    lemon.keep_interactions(entry, smallm, waters, 6.0)

    # Output phase
    return pdbid +
           lemon.count_print_residue_names(entry,
                                           smallm) +
           '\n'
```

Listing 21

This workflow is a Python port of Listing 17 and is an example of a 'complex' workflow implemented in Python. It is also an example of how to implement more functionality in the Python derived subclass.

```
from __future__ import print_function
import lemon

class MyWorkflow(lemon.Workflow):
    def __init__(self):
        import lemon

        # This line is very important!
        lemon.Workflow.__init__(self)
        self.dihedral_dict = {}

    def worker(self, entry, pdbid):
        import lemon
        import math

        protein_only = lemon.Frame()
```

```
        peptides = lemon.ResidueIDs()
        if (lemon.select_specific_residues(
                entry, peptides,
                lemon.common_peptides) == 0):
            return ""

        lemon.separate_residues(entry, peptides,
                                protein_only)
        dihedrals = protein_only.topology().dihedrals()
        for dihedral in dihedrals:
            dihedralnm = ""
            try:
                dihedralnm = lemon.protein_dihedral_name(
                                protein_only, dihedral,
                                lemon.proline_res)

            except lemon.GeometryError as error:
                return pdbid + ": " + 'error' + '\n'
            theta = protein_only.dihedral(dihedral[0],
                                          dihedral[1],
                                          dihedral[2],
                                          dihedral[3])

            dbin = int(math.floor(theta / 0.01))
            sbin = (dihedralnm, dbin)
            if sbin in self.dihedral_dict:
                self.dihedral_dict[sbin] =
                    self.dihedral_dict[sbin] + 1
            else:
                self.dihedral_dict[sbin] = 1
        return ""
    def finalize(self):
        for sbin, count in self.dihedral_dict.items():
            print(sbin[0], '\t', sbin[1], '\t', count)
```

D. ADDITIONAL TABLES FOR CHAPTER 2.2

D.1 Additional tables for LR model results

On the following pages, the Leave-one out cross-validation (LOOCV) results will be give for all the logistic regression (LR) models trained on on differing input features (BDI, RNA-seq, and subsets and combinations thereof). The hyper–parameters used below are the cost used for training the LR model, the type of regularization used and the epsilon value used for minimizing the loss function. The accuracy and Cohen-kappa values for the LOOCV validation of each model is given.

Table D.1.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained only using the BDI variables.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.5789	-0.0704
0.5	L1	0.01	0.5789	0.0256
0.5	L1	0.1	0.5263	-0.2667
0.5	L2_dual	0.001	0.5789	0.0256
0.5	L2_dual	0.01	0.5789	0.0256
0.5	L2_dual	0.1	0.5789	0.0256
0.5	L2_primal	0.001	0.5789	0.0256
0.5	L2_primal	0.01	0.5789	0.0256
0.5	L2_primal	0.1	0.5789	0.0256
1	L1	0.001	0.5789	-0.0704
1	L1	0.01	0.5789	-0.0704
1	L1	0.1	0.5789	-0.0704
1	L2_dual	0.001	0.5789	0.0256
1	L2_dual	0.01	0.5789	0.0256
1	L2_dual	0.1	0.5789	0.0256
1	L2_primal	0.001	0.5789	0.0256
1	L2_primal	0.01	0.5789	0.0256
1	L2_primal	0.1	0.5789	0.0256
2	L1	0.001	0.6316	0.1074
2	L1	0.01	0.6316	0.1074
2	L1	0.1	0.5263	-0.1477
2	L2_dual	0.001	0.5789	0.0256
2	L2_dual	0.01	0.5789	0.0256
2	L2_dual	0.1	0.5789	0.0256
2	L2_primal	0.001	0.5789	0.0256
2	L2_primal	0.01	0.5789	0.0256
2	L2_primal	0.1	0.5789	0.0256

Table D.2.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained only using the Best 3 BDI variables.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.6842	0.2692
0.5	L1	0.01	0.6842	0.2692
0.5	L1	0.1	0.6316	0.1074
0.5	L2_dual	0.001	0.7368	0.4172
0.5	L2_dual	0.01	0.7368	0.4172
0.5	L2_dual	0.1	0.7368	0.4172
0.5	L2_primal	0.001	0.7368	0.4172
0.5	L2_primal	0.01	0.7368	0.4172
0.5	L2_primal	0.1	0.7368	0.4172
1	L1	0.001	0.6842	0.2692
1	L1	0.01	0.6842	0.2692
1	L1	0.1	0.6842	0.2692
1	L2_dual	0.001	0.7368	0.4172
1	L2_dual	0.01	0.7368	0.4172
1	L2_dual	0.1	0.7368	0.4172
1	L2_primal	0.001	0.7368	0.4172
1	L2_primal	0.01	0.7368	0.4172
1	L2_primal	0.1	0.7368	0.4172
2	L1	0.001	0.6842	0.2692
2	L1	0.01	0.6842	0.2692
2	L1	0.1	0.6842	0.2692
2	L2_dual	0.001	0.7368	0.4172
2	L2_dual	0.01	0.7368	0.4172
2	L2_dual	0.1	0.7368	0.4172
2	L2_primal	0.001	0.7368	0.4172
2	L2_primal	0.01	0.7368	0.4172
2	L2_primal	0.1	0.7368	0.4172

Table D.3.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained only using the RNA-seq variables.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.6842	0.1972
0.5	L1	0.01	0.6316	0.0148
0.5	L1	0.1	0.6316	0.0148
0.5	L2_dual	0.001	0.7895	0.4062
0.5	L2_dual	0.01	0.7895	0.4062
0.5	L2_dual	0.1	0.7895	0.4062
0.5	L2_primal	0.001	0.7895	0.4062
0.5	L2_primal	0.01	0.7895	0.4062
0.5	L2_primal	0.1	0.7895	0.4062
1.0	L1	0.001	0.7895	0.4648
1.0	L1	0.01	0.7368	0.2963
1.0	L1	0.1	0.7895	0.4648
1.0	L2_dual	0.001	0.7895	0.4062
1.0	L2_dual	0.01	0.7895	0.4062
1.0	L2_dual	0.1	0.7895	0.4062
1.0	L2_primal	0.001	0.7895	0.4062
1.0	L2_primal	0.01	0.7895	0.4062
1.0	L2_primal	0.1	0.7895	0.4062
2.0	L1	0.001	0.6316	0.1074
2.0	L1	0.01	0.6842	0.1972
2.0	L1	0.1	0.7368	0.3624
2.0	L2_dual	0.001	0.7895	0.4062
2.0	L2_dual	0.01	0.7895	0.4062
2.0	L2_dual	0.1	0.7895	0.4062
2.0	L2_primal	0.001	0.7895	0.4062
2.0	L2_primal	0.01	0.7895	0.4062
2.0	L2_primal	0.1	0.7895	0.4062

D.2 Additional tables for combined variable models

Table D.4.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained using both the BDI and RNA-seq variables without selection.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.4737	-0.2179
0.5	L1	0.01	0.4737	-0.2179
0.5	L1	0.1	0.5263	-0.1477
0.5	L2_dual	0.001	0.6842	0.1972
0.5	L2_dual	0.01	0.6842	0.1972
0.5	L2_dual	0.1	0.6842	0.1972
0.5	L2_primal	0.001	0.6842	0.1972
0.5	L2_primal	0.01	0.6842	0.1972
0.5	L2_primal	0.1	0.6842	0.1972
1.0	L1	0.001	0.5789	0.1059
1.0	L1	0.01	0.5789	0.1059
1.0	L1	0.1	0.6316	0.1074
1.0	L2_dual	0.001	0.6842	0.1972
1.0	L2_dual	0.01	0.6842	0.1972
1.0	L2_dual	0.1	0.6842	0.1972
1.0	L2_primal	0.001	0.6842	0.1972
1.0	L2_primal	0.01	0.6842	0.1972
1.0	L2_primal	0.1	0.6842	0.1972
2.0	L1	0.001	0.5789	0.1059
2.0	L1	0.01	0.6316	0.1074
2.0	L1	0.1	0.6842	0.1972
2.0	L2_dual	0.001	0.6842	0.1972
2.0	L2_dual	0.01	0.6842	0.1972
2.0	L2_dual	0.1	0.6842	0.1972
2.0	L2_primal	0.001	0.6842	0.1972
2.0	L2_primal	0.01	0.6842	0.1972
2.0	L2_primal	0.1	0.6842	0.1972

Table D.5.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained using both the best 3 BDI biomarkers and the RNA-seq variables without selection.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.4737	-0.1176
0.5	L1	0.01	0.5263	-0.0491
0.5	L1	0.1	0.5263	-0.0491
0.5	L2_dual	0.001	0.7895	0.4062
0.5	L2_dual	0.01	0.7895	0.4062
0.5	L2_dual	0.1	0.7895	0.4062
0.5	L2_primal	0.001	0.7895	0.4062
0.5	L2_primal	0.01	0.7895	0.4062
0.5	L2_primal	0.1	0.7895	0.4062
1.0	L1	0.001	0.6316	0.1840
1.0	L1	0.01	0.6842	0.2692
1.0	L1	0.1	0.7895	0.4648
1.0	L2_dual	0.001	0.7895	0.4062
1.0	L2_dual	0.01	0.7895	0.4062
1.0	L2_dual	0.1	0.7895	0.4062
1.0	L2_primal	0.001	0.7895	0.4062
1.0	L2_primal	0.01	0.7895	0.4062
1.0	L2_primal	0.1	0.7895	0.4062
2.0	L1	0.001	0.7368	0.4172
2.0	L1	0.01	0.7368	0.4172
2.0	L1	0.1	0.7895	0.4648
2.0	L2_dual	0.001	0.7895	0.4062
2.0	L2_dual	0.01	0.7895	0.4062
2.0	L2_dual	0.1	0.7895	0.4062
2.0	L2_primal	0.001	0.7895	0.4062
2.0	L2_primal	0.01	0.7895	0.4062
2.0	L2_primal	0.1	0.7895	0.4062

Table D.6.
LOOCV results for regularized logistic regression models trained using various hyper–parameters trained using both the BDI and RNA-seq variables with only the seven variables ALLF1pred, SDIP1dox, LOF0chop, ENSCAFG00000011225, SH2D4A, KIAA1217, FGFR4.

Cost	Loss	Epsilon	Accuracy	Kappa
0.5	L1	0.001	0.736842	0.463277
0.5	L1	0.01	0.736842	0.463277
0.5	L1	0.1	0.736842	0.463277
0.5	L2_dual	0.001	0.894737	0.776471
0.5	L2_dual	0.01	0.894737	0.776471
0.5	L2_dual	0.1	0.894737	0.776471
0.5	L2_primal	0.001	0.894737	0.776471
0.5	L2_primal	0.01	0.894737	0.776471
0.5	L2_primal	0.1	0.894737	0.776471
1.0	L1	0.001	0.894737	0.776471
1.0	L1	0.01	0.894737	0.776471
1.0	L1	0.1	0.894737	0.776471
1.0	L2_dual	0.001	0.894737	0.776471
1.0	L2_dual	0.01	0.894737	0.776471
1.0	L2_dual	0.1	0.894737	0.776471
1.0	L2_primal	0.001	0.894737	0.776471
1.0	L2_primal	0.01	0.894737	0.776471
1.0	L2_primal	0.1	0.894737	0.776471
2.0	L1	0.001	0.894737	0.776471
2.0	L1	0.01	0.894737	0.776471
2.0	L1	0.1	0.894737	0.776471
2.0	L2_dual	0.001	0.947368	0.883436
2.0	L2_dual	0.01	0.947368	0.883436
2.0	L2_dual	0.1	0.947368	0.883436
2.0	L2_primal	0.001	0.947368	0.883436
2.0	L2_primal	0.01	0.947368	0.883436
2.0	L2_primal	0.1	0.947368	0.883436

D.3 LOOT Results

For the following tables, the prediction column gives the predicted sensitive of the 'removed' dog and the ground truth is the actual outcome of the 'removed' dog. The self score gives how well the model is able to predict the dogs used to train it and the validation kappa is the kappa value obtained from cross validation of the model.

Table D.7.
LOOT performed for all patent samples with the BDI biomarkers

Prediction	Ground truth	'Removed' patient ID	Self-score	Validation kappa
Sensitive	Resistant	LY09	18	0.0526
Sensitive	Resistant	case786-844	18	0.2603
Resistant	Resistant	LY02	17	0.0526
Sensitive	Resistant	LY06	18	0.1692
Resistant	Resistant	Ly58BD	18	0.0137
Sensitive	Resistant	LY01	18	0.5068
Sensitive	Sensitive	LY05	18	0.1600
Resistant	Sensitive	Ly42GJ	18	0.5385
Sensitive	Sensitive	Ly51CL	17	0.0870
Sensitive	Sensitive	LY04	18	0.0870
Sensitive	Sensitive	LY03	16	0.1600
Sensitive	Sensitive	LY07	18	0.2500
Resistant	Sensitive	LY08	16	0.2500
Sensitive	Sensitive	Ly43BD	18	0.3478
Sensitive	Sensitive	case785-528	17	0.0000
Sensitive	Sensitive	Ly83MS	18	0.0769
Sensitive	Sensitive	case782-104	17	0.2500
Resistant	Sensitive	Ly01YB	18	0.2500
Sensitive	Sensitive	LY10	17	0.0870

Table D.8.
LOOT performed for all patent samples (see the methods section in the main text for a description of this technique) with the RNA-seq variables.

Prediction	Ground truth	'Removed' patient ID	Self-score	Validation kappa
Sensitive	Sensitive	LY04	18	0.4000
Sensitive	Sensitive	LY05	18	0.4000
Resistant	Resistant	LY09	18	0.0526
Resistant	Resistant	LY02	18	0.2653
Sensitive	Sensitive	Ly83MS	18	0.3478
Sensitive	Sensitive	case782-104	18	0.4545
Sensitive	Sensitive	Ly43BD	18	0.2857
Sensitive	Sensitive	LY07	18	0.4545
Sensitive	Sensitive	Ly51CL	18	0.4545
Sensitive	Sensitive	LY08	18	0.4545
Sensitive	Resistant	LY01	17	0.4906
Sensitive	Sensitive	Ly42GJ	18	0.4000
Sensitive	Sensitive	LY10	18	0.4545
Sensitive	Resistant	case786-844	18	0.3684
Sensitive	Sensitive	Ly01YB	18	0.4545
Sensitive	Resistant	LY06	16	0.3684
Resistant	Sensitive	LY03	17	0.7273
Sensitive	Resistant	Ly58BD	18	0.3684
Sensitive	Sensitive	case785-528	18	0.4000

Table D.9.
LOOT performed for all patent samples with the best 3 BDI biomarkers.

Prediction	Ground truth	'Removed' patient ID	Self-score	Validation kappa
Sensitive	Sensitive	case782-104	15	0.4000
Resistant	Sensitive	case785-528	17	0.6400
Resistant	Resistant	case786-844	14	0.1692
Sensitive	Resistant	LY01	15	0.5068
Resistant	Sensitive	Ly01YB	17	0.7692
Sensitive	Resistant	LY02	15	0.3478
Sensitive	Sensitive	LY03	15	0.4000
Sensitive	Sensitive	LY04	14	0.4000
Sensitive	Sensitive	LY05	14	0.4000
Resistant	Resistant	LY06	14	0.3478
Sensitive	Sensitive	LY07	15	0.4000
Sensitive	Sensitive	LY08	15	0.4000
Resistant	Resistant	LY09	14	0.3478
Sensitive	Sensitive	LY10	15	0.4000
Resistant	Sensitive	Ly42GJ	15	0.6400
Sensitive	Sensitive	Ly43BD	15	0.4000
Sensitive	Sensitive	Ly51CL	15	0.4000
Resistant	Resistant	Ly58BD	14	0.3478
Sensitive	Sensitive	Ly83MS	14	0.4000

Table D.10.
LOOT performed for all patent samples with both the top 4 RNA-seq and the best 3 unnormalized BDI biomarkers (seven feature model).

Prediction	Ground truth	'Removed' patient ID	Self-score	Validation kappa
Sensitive	Sensitive	case782-104	18	0.8800
Sensitive	Sensitive	case785-528	18	1.0000
Resistant	Resistant	case786-844	18	0.7231
Resistant	Resistant	LY01	18	0.8696
Sensitive	Sensitive	Ly01YB	18	0.8800
Resistant	Resistant	LY02	18	0.8696
Resistant	Sensitive	LY03	18	1.0000
Sensitive	Sensitive	LY04	18	0.8800
Sensitive	Sensitive	LY05	18	0.8800
Resistant	Resistant	LY06	18	0.8696
Sensitive	Sensitive	LY07	18	0.8800
Sensitive	Sensitive	LY08	17	0.8800
Resistant	Resistant	LY09	18	0.8696
Sensitive	Sensitive	LY10	18	0.8800
Sensitive	Sensitive	Ly42GJ	18	0.8800
Sensitive	Sensitive	Ly43BD	18	0.8800
Sensitive	Sensitive	Ly51CL	18	0.8800
Resistant	Resistant	Ly58BD	18	0.7231
Sensitive	Sensitive	Ly83MS	18	0.8800

E. ADDITIONAL FIGURES AND LISTINGS FOR CHAPTER 2.4

E.1 Additional computational details

E.1.1 Docking of CANDO predictions to selected CRPC targets

We retrieved the structures for all FDA compounds using the 'Name-to-Structure' feature available in ChemAxon's MarvinSketch and these structures were prepared for docking using the 'Clean in 3D' feature. All ten molecules are saved in their 3D form as separate structures in the TRIPOS MOL2 format and submitted to CANDOCK v0.4.3's prep_fragrments module, resulting in 27 unique rigid fragments.

The eighteen targets can be obtained from the Protein Data Bank (see Table E.2). Binding sites for all targets were predicted using the find centroids module available in CANDOCK and saved to corresponding centroid files. For all targets, we used the default CANDOCK parameters to perform rigid fragment docking and linking. We used the radial-mean-reduced objective function with a cutoff of 6.0Å for scoring and employed iterative linking for linking fragments.

An independent docking calculation job was launched for each protein target, and the highest ranked scored pose in the binding pocket was extracted for further calculations. The name of the target and the compound docked to it are saved in long CSV format along with the score of best-docked pose, and the results from all targets were concatenated into one file. If CANDOCK predicted a compound to be a non-binder, a score of 10000 was inserted. The interactions of all CANDO predicted compounds is given in Table E.3.

E.1.2 Docking of designed compounds to selected CRPC targets

Each additional design was drawn using MarvinSketch, and the 3D coordinates were assigned using the clean in a 3D feature available in this software package. We reused the binding sites predicted for the original ten compounds and the same docking procedure was employed to obtain docking scores for all new designs and the selected CRPC targets. A separate CSV file was created from this docking data.

E.1.3 Classification of active compounds using Support Vector Machine

The CSV file created from the docking the experimentally tested was loaded in R v3.3.2 using the read_csv function, and the spread function was used to create a matrix representation of the data. An additional 'activity' column was appended to the matrix to contain the experimental cell assay result where a value of 'A' was used to indicate a compound as active, and a value of 'I' was used to indicate inactive compounds. Support Vector Machine C–Classification implemented via the e1071 package along with a radial basis function was used to create a model with all ten compounds where the activity column is the dependent on all eighteen features produced by the docking.

We created a feature matrix from the new designs was obtained similarly to the one used to create the training feature matrix. Using the SVM model created from the active and inactive compounds, we classified these new designs as being in the active or inactive class. To avoid synthesizing and experimentally verifying the multitude of compounds predicted as active, we extracted the SVM decision values calculated using this model and preferred compound that are significantly more likely to be active than other compounds in the distribution. Therefore, we only considered decision values greater than one standard devotion than the mean. Nine compounds are greater than this cut off, three are the compounds used to train the SVM model, and seven

are new compounds with unknown activity in the treatment of CRPC. These seven compounds, termed **2**, **4**, **11**, **29**, **40**, and **42** were used in a second round of cell assays to verify that these compounds are indeed active.

Cell assays revealed that **2**, **4**, **40**, and **42** are active compounds in treating CRPC. Thus, the performance of the initial training is 67% with a recall of 100% yielding an F1-Score of 60%. After the second round of in vitro testing, we wished to ensure the new experimental results did not lead to additional compounds being classified as active. To accomplish this, we created a second SVM model using the ten original compounds and the seven-compounds identified by the first network. We then used this network to predict the activities of all compounds presented in this paper to check if any other compounds may now be classified as active. See that this was not the case, i.e. no new compounds were identified by the new model.

E.1.4 Identification of a unique protein network

We want to narrow down the long list of targets we examined (eighteen) to a smaller set of targets that can be used to model the more extensive system. To do so, we used all steroidal compounds (both experimentally active and inactive) to create a third SVM model using all eighteen features. Then, we calculated the correlation between these eighteen features by multiplying the transpose of the support vector matrix by the support vector matrix to obtain a matrix representing how each feature correlations in any other feature. We then took the mean correlation each feature has (i.e. the mean of each row of the matrix) to determine how independent each feature is from the other features. The features with the highest degree of independence were selected one by one to create new SVM models until the smaller network produced the same result as the network containing all possible features. The smallest feature set created with this method was then tested on all compounds to ensure no new compounds are predicted to be active with this smaller feature set.

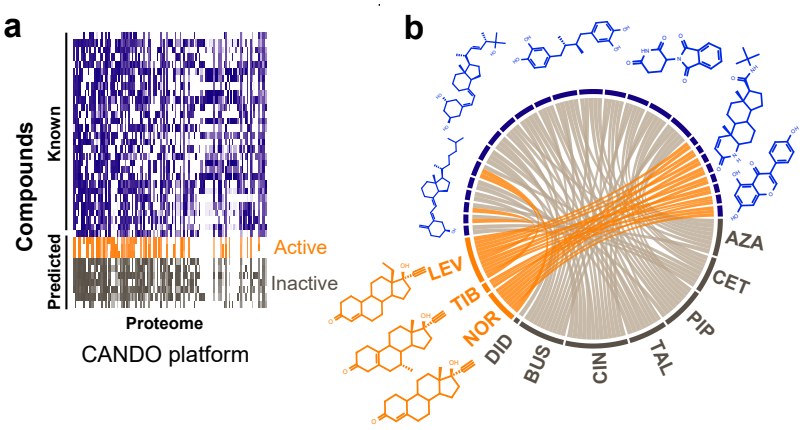


Figure E.1. Prediction of initial leads using the CANDO platform. (a) Proteome wide signatures for all known prostate cancer therapeutics (blue) and initial leads compounds (grey and orange). The signatures of the unknown compounds are compared to the known signatures to produce the initial leads presented in this paper. The orange signatures are for the active initial leads while the grey signatures are of inactive initial leads. (b) Chord diagram showing relationship between known prostate cancer drugs and the initial lead, the connections between the initial predictions and the known prostate cancer therapies.

New models were created using features from the most independent to the most correlated. If the predictive capability of the model improved, then the feature was accepted and added to the increasing model. If it does not improve the model, the feature is rejected. To create a disease specific network, only the ability of the model to classify a compound as active vs inactive was considered in ranking a model. For compound specific networks, an additional criterion of increasing the rank of a given compound was included, thus the new model must increase the number of predicted actives, or, if the number of predicted actives did not change, then the model must improve the rank of a compound in question.

Table E.1.
Viability IC_{50} values of the predicted drugs in different cancer cell lines.

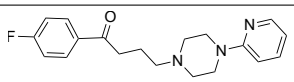
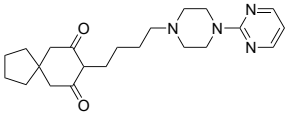
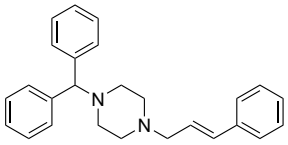
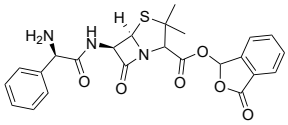
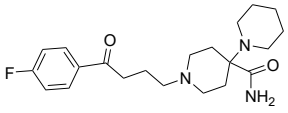
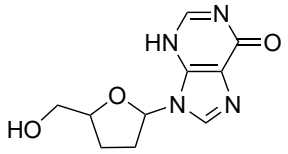
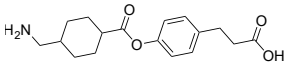
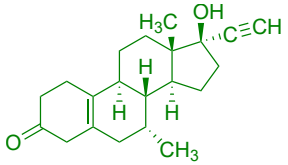
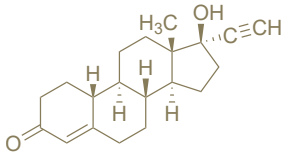
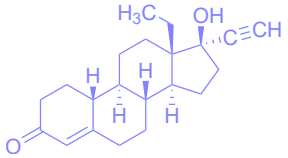
Name	Structure	LNCaP cell	C4-2 cell
Azaperone (AZA)		>10.0 μM	>10.0 μM
Buspirone (BUS)		>10.0 μM	5.25 μM
Cinnarizine (CIN)		>10.0 μM	9.12 μM
Pipamperone (PIP)		>10.0 μM	>10.0 μM
Didanosine (DID)		>10.0 μM	>10.0 μM
Cetraxate (CET)		>10.0 μM	>10.0 μM
Talampicillin (TAL)		>10.0 μM	>10.0 μM
Tibolone (TIB)		24.86 nM	3.12 nM
Norethisterone (NOR)		32.52 nM	7.04 nM
Levonorgestrel (LEV)		181.0 nM	41.78 nM

Table E.2.: Target selection for the initial active leads.

UniProt	Gene	Protein name	Association with CRPC
P14061	HSD17B1*	Estradiol 17-beta-dehydrogenase 1	Involved in the androgen synthesis such as testosterone biosynthesis from cholesterol
P51449	RORG*	Nuclear receptor ROR-gamma	Drives AR expression through recruitment of coactivators (SRC-1/3) and binding to AR-ROR response elements on DNA
P04278	SHBG*	Sex hormone-binding globulin	Involved in transportation of androgens such as testosterone from blood stream to cancer cells
P28845	HSD11B1*	Corticosteroid 11-beta-dehydrogenase isozyme 1	Involved in the androgen synthesis such as testosterone biosynthesis from cholesterol
P03372	ESR1*	Estrogen receptor	Known to involve in the development and progression of prostate cancer
P06401	PGR*	Progesterone receptor	Involved in the activation of AR-dimerization and its translocation in nucleus.
P08235	MCR*	Mineralocorticoid receptor	Involved in the activation of AR-dimerization and its translocation in nucleus.
P08185	CBG*	Corticosteroid-binding globulin	The major carrier of the hormone such as androgens and cortisol
P10275	AR*	Androgen receptor	Upon activation or mutation drives contact independent growth of prostate cancer cells
O75469	NR1I2*	Nuclear receptor subfamily 1 group I member 2 (PXR)	Shown to interact with AR and repress AR-regulated transcription in the presence of AR antagonists.
P49888	SULT1E1*	Estrogen sulfotransferase	Associated in outcomes of abiraterone Acetate therapies in men with mCRPC.
O60218	AKR1B10*	Aldo-keto reductase family 1 member B10	Involved in the synthesis of active form of androgen such as DHT from testosterone.
P52895	AKR1C2	Aldo-keto reductase family 1 member C2	Functions as a DHT reductase.

continued on next page

Table E.2.: *continued*

UniProt	Gene	Protein name	Association with CRPC
P04150	GR	Glucocorticoid receptor	Transcription factor resulting in multiple signaling pathway inhibition and tumor suppression.
P42330	AKR1C3	Aldo-keto reductase family 1 member C3	Involved in the synthesis of androgens (such as testosterone) from cholesterol.
P18405	SRD5A1	3-oxo-5-alpha-steroid 4-dehydrogenase 1	Catalyses the conversion of testosterone to DHT.
P17516	AKR1C4	Aldo-keto reductase family 1 member C4	Liver-specific metabolic enzyme which catalyzes the reduction of 5α-pregnane-3,20-dione to yield 3α-hydroxy-5α-pregnane-20-one, a precursor of androsterone and thus it plays a critical role in the "backdoor pathway" of androgen synthesis in prostate cancer.
P05093	CYP17A1	Steroid 17-alpha-hydroxylase/17,20 lyase	Involved in the synthesis of androgens (such as testosterone) from cholesterol.

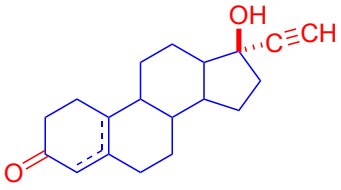
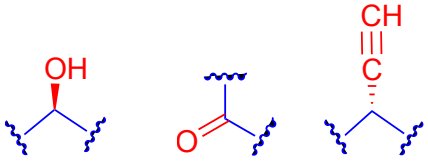
Table E.3.
Protein-compound interaction scores for initial set of experimentally tested compounds used for machine learning

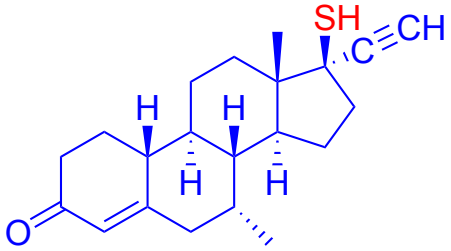
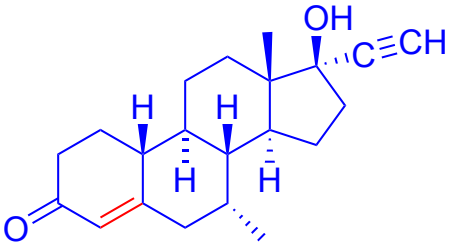
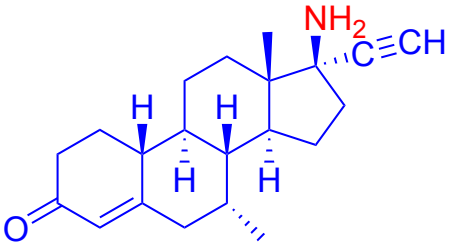
PDBID	UniProt	AZA	BUS	CET	CIN	DID	LEV	NOR	PIP	TAL	TIB
4xo6	P52895	-36.16	10000	-38.52	-49.98	-30.2	-28.50	-26.07	-30.85	-42.5	-27.03
4ga8	O60218	-50.47	-62.43	-53.83	-71.51	-44.21	-22.93	-29.97	-50.87	-47.63	-29.77
4xve	P42330	-56.44	-60.58	-48.79	-60.18	-34.68	-39.34	-36.45	-55.20	-55.91	-30.93
2fvl	P17516	-53.27	-60.12	-56.30	-68.28	-34.85	-40.59	-36.58	-55.68	-54.35	-43.84
4oha	P10275	-31.35	-43.01	-38.98	-52.16	-28.84	-25.70	-22.68	-37.24	-40.52	-22.86
4c41	P08185	-17.13	-33.35	-31.68	-32.68	-28.05	-15.62	-17.33	-32.26	-23.96	-11.17
4nkw	P05093	-52.87	-54.26	-48.86	-68.21	-39.04	-40.88	-40.54	-57.96	-54.27	-38.33
5kct	P03372	-45.13	-41.47	-54.37	-69.15	-35.97	-27.93	-30.86	-45.46	-40.59	-41.54
4udd	P04150	-45.84	-45.86	-55.21	-61.23	-35.10	-44.24	-43.29	-48.95	-49.96	-36.21
4c7k	P28845	-52.75	-68.81	-56.41	-64.75	-48.53	-42.24	-41.30	-59.94	-63.21	-42.32
1jtv	P14061	-46.15	-58.44	-51.97	-62.35	-36.55	-36.30	-41.66	-52.82	-50.58	-32.88
4pf3	P08235	-33.05	-43.90	-45.28	-44.86	-26.87	-26.02	-23.26	-37.15	-40.27	-23.77
4xhd	O75469	-51.22	-32.92	-41.80	-55.93	-36.55	-33.43	-31.95	-50.57	-54.62	-37.64
1sqn	P06401	-31.73	10000	-38.70	-36.32	-31.63	-36.10	-34.20	-39.10	-30.27	-24.00
5ayg	P51449	-47.75	-50.03	-47.59	-56.66	-35.61	-38.15	-37.11	-44.39	-51.35	-42.78
1d2s	P04278	-23.44	10000	-29.52	-42.19	-30.51	-29.75	-24.21	-15.29	-21.26	-28.84
Modeled	P18405	-31.40	-29.13	-30.62	-42.57	-25.48	-24.99	-22.20	-30.65	-35.31	-24.60
1hy3	P49888	-43.92	-39.60	-49.24	-51.45	-34.54	-40.30	-36.56	-54.04	-44.87	-38.43

Table E.4.
Protein-compound interaction scores for experimentally active compounds.

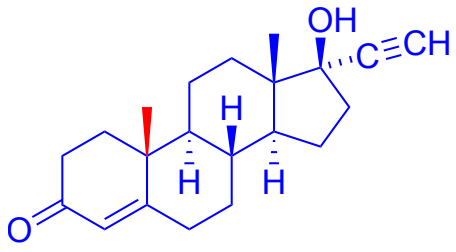
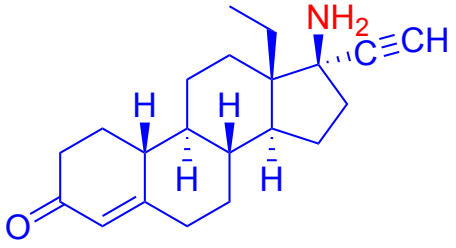
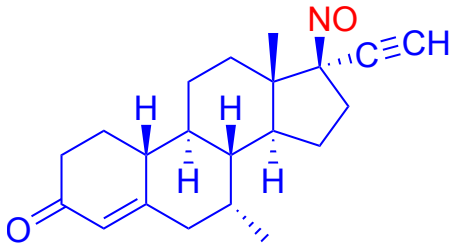
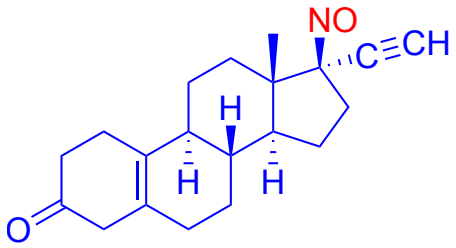
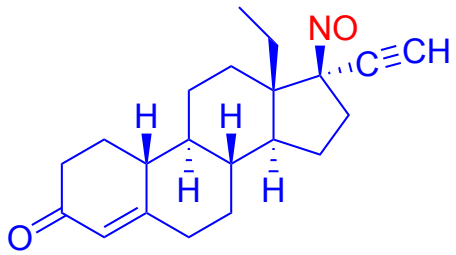
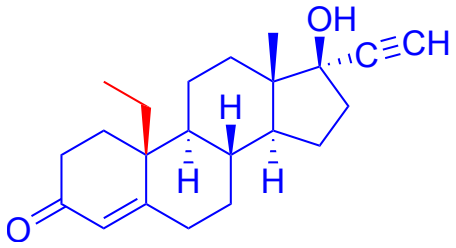
PDBID	UniProt	2	4	40	42	LEV	NOR	TIB
4xo6	P52895	-26.48	-24.37	-33.93	-33.41	-28.50	-26.07	-27.03
4ga8	O60218	-28.35	-25.87	-36.74	-35.91	-22.93	-29.97	-29.77
4xve	P42330	-24.68	-29.93	-38.14	-38.65	-39.34	-36.45	-30.93
2fvl	P17516	-40.97	-38.31	-40.10	-39.50	-40.59	-36.58	-43.84
4oha	P10275	-27.70	-24.52	-27.33	-27.30	-25.70	-22.68	-22.86
4c41	P08185	-10.88	-16.19	-16.61	-18.26	-15.62	-17.33	-11.17
4nkw	P05093	-36.88	-38.33	-38.03	-39.20	-40.88	-40.54	-38.33
5kct	P03372	-20.61	-25.69	-33.22	-33.98	-27.93	-30.86	-41.54
4udd	P04150	-37.13	-39.88	-44.94	-44.83	-44.24	-43.29	-36.21
4c7k	P28845	-41.49	-41.44	-43.81	-45.31	-42.24	-41.30	-42.32
1jtv	P14061	-35.83	-38.21	-40.16	-37.55	-36.30	-41.66	-32.88
4pf3	P08235	-28.08	-24.57	-26.91	-27.79	-26.02	-23.26	-23.77
4xhd	O75469	-34.02	-35.11	-37.93	-37.96	-33.43	-31.95	-37.64
1sqn	P06401	-39.24	-34.56	-48.50	-47.52	-36.10	-34.20	-24.00
5ayg	P51449	-37.76	-37.20	-38.79	-38.48	-38.15	-37.11	-42.78
1d2s	P04278	-34.04	-28.60	-36.47	-38.23	-29.75	-24.21	-28.84
Modeled	P18405	-23.94	-22.83	-24.60	-24.79	-24.99	-22.20	-24.60
1hy3	P49888	-36.45	-37.33	-37.95	-38.30	-40.30	-36.56	-38.43

Table E.5.: List of the designed molecules using the common scaffold and groups/fragments of the initial leads including decision values obtained after the first round of machine learning:

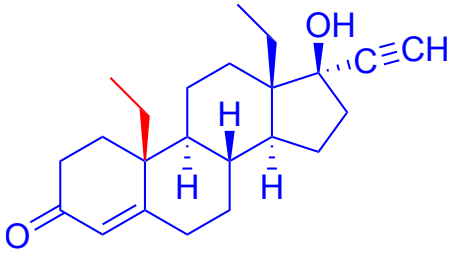
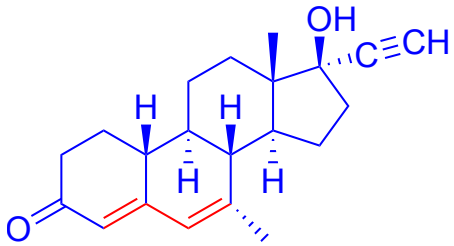
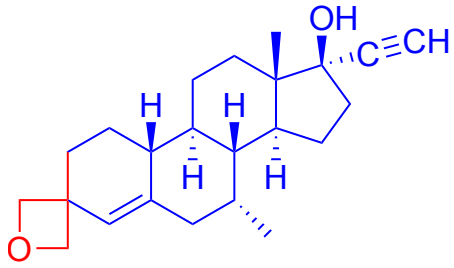
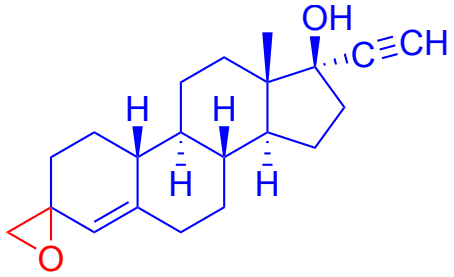
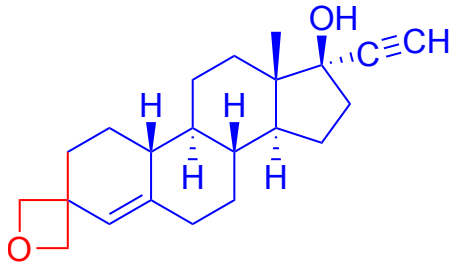
Common initial lead scaffold	
Common initial lead FGs	

Iso-stereo and isomer based designs		
Code	Structure	SVM distance
01		-0.156873513
02		-0.881368758
03		-0.533115698

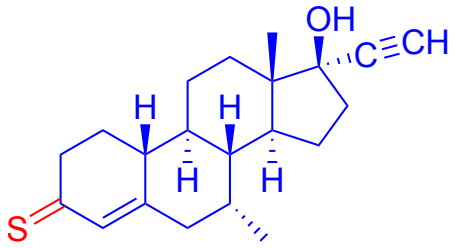
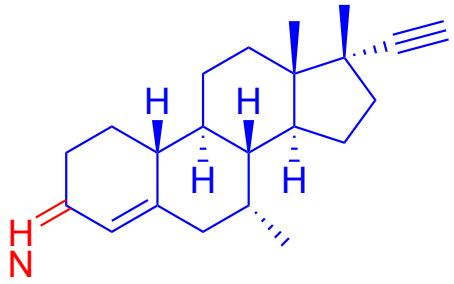
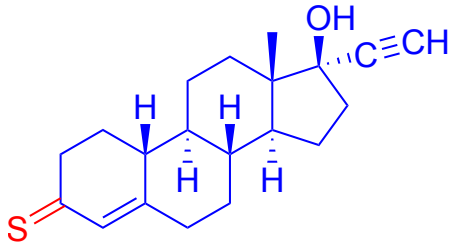
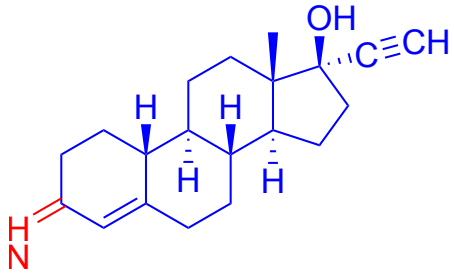
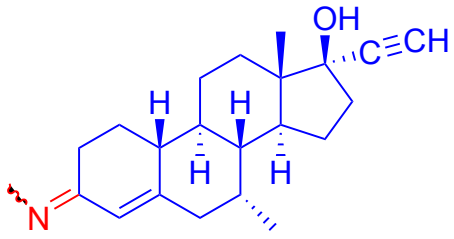
continued on next page

Code	Structure	SVM distance
04		-1.010892492
05		-0.047654449
06		-0.354780662
07		-0.427391831
08		-0.614704024
09		-0.218538216

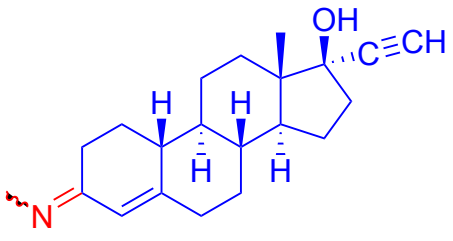
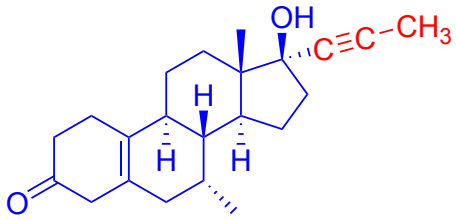
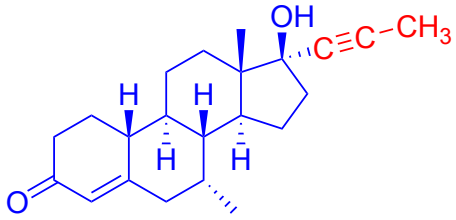
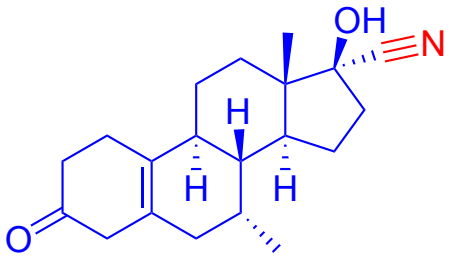
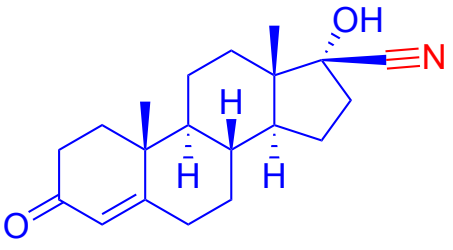
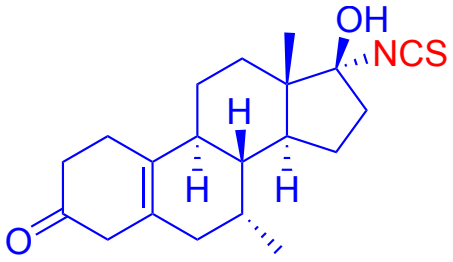
continued on next page

Code	Structure	SVM distance
10		-0.188698997
11		-0.852335189
12		0.215775616
13		0.136362802
14		0.421769484

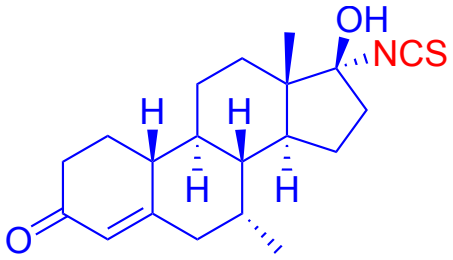
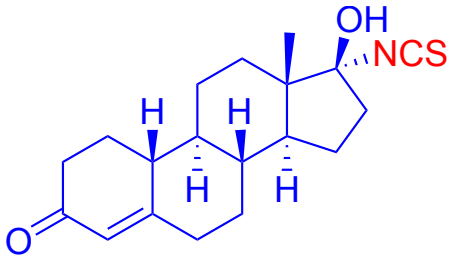
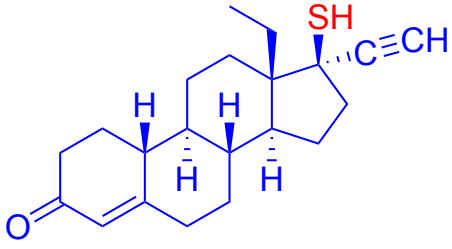
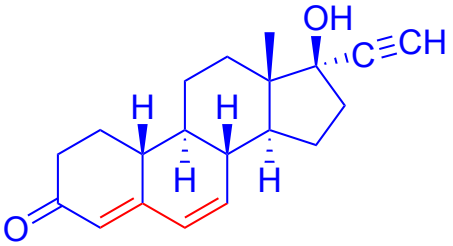
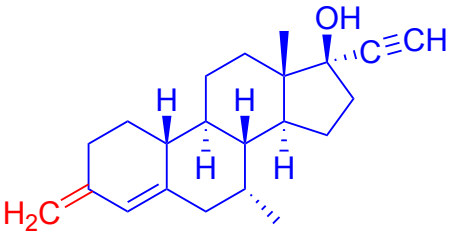
continued on next page

Code	Structure	SVM distance
15		0.01972264
16		0.078050193
17		-0.148133161
18		0.37967144
19		0.35951977

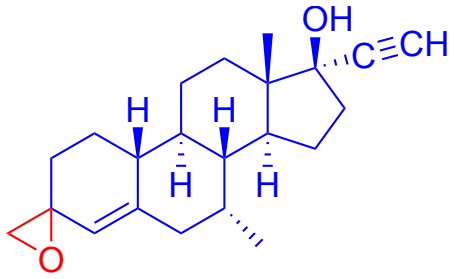
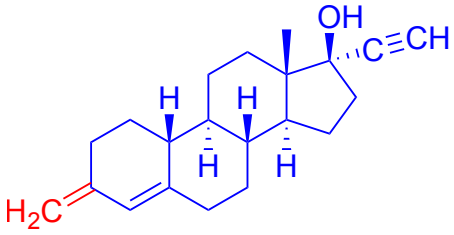
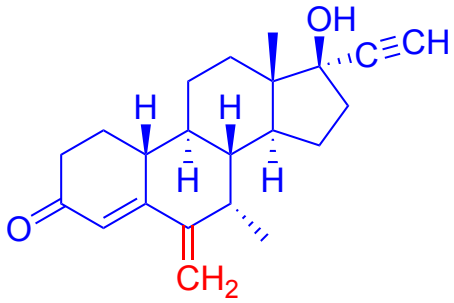
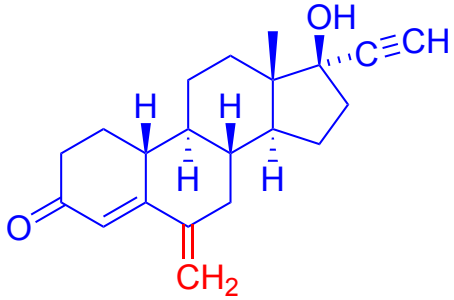
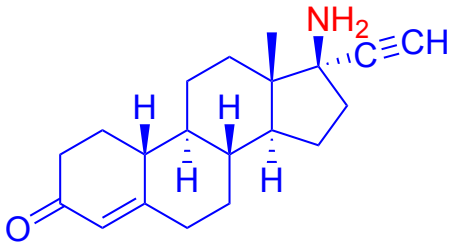
continued on next page

Code	Structure	SVM distance
20		0.18039821
21		-0.13211674
22		0.49061446
23		-0.014083785
24		0.49061445
25		0.045484673

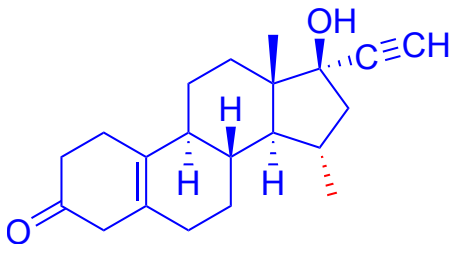
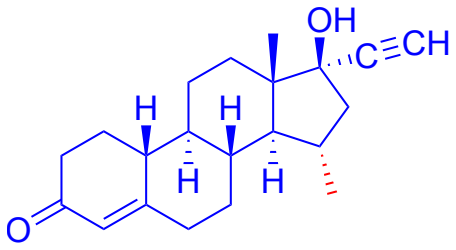
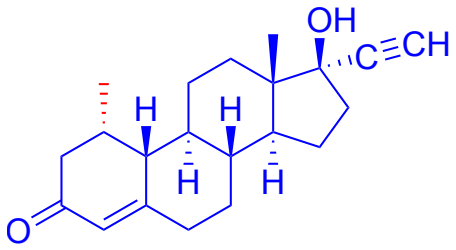
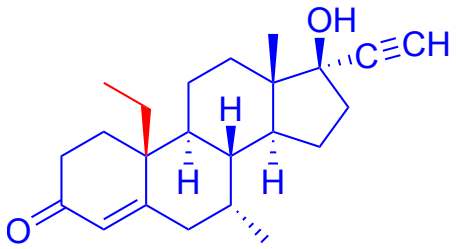
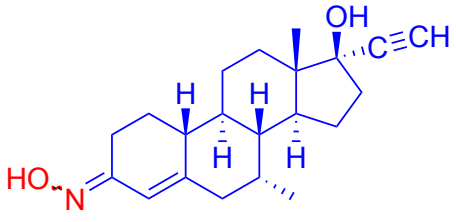
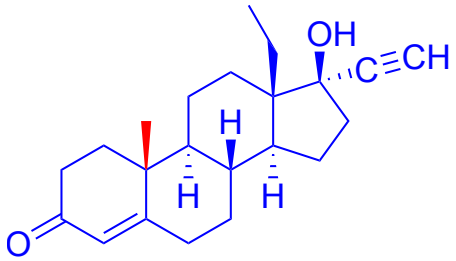
continued on next page

Code	Structure	SVM distance
26		0.082019865
27		-0.067820763
28		-0.504004393
29		-0.97760207
30		0.323527962

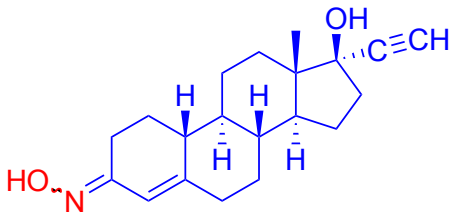
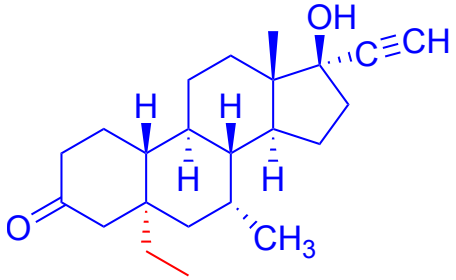
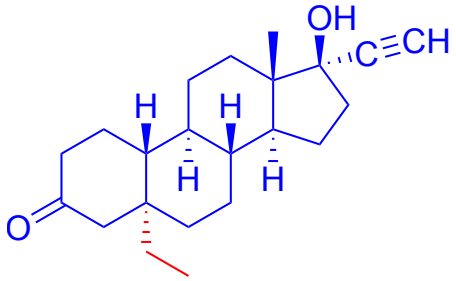
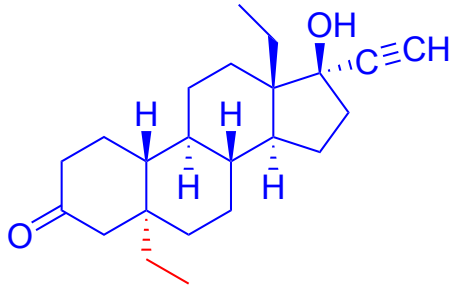
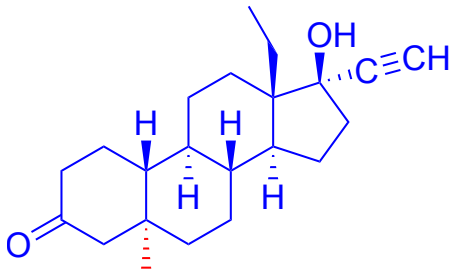
continued on next page

Code	Structure	SVM distance
31		0.315092791
32		0.078039826
33		-0.29541342
34		-0.371547573
35		-0.251111223

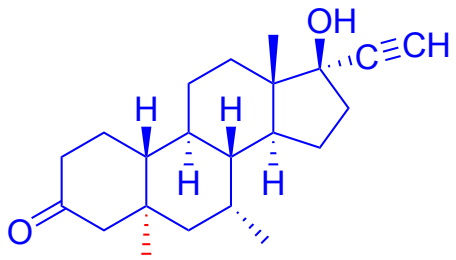
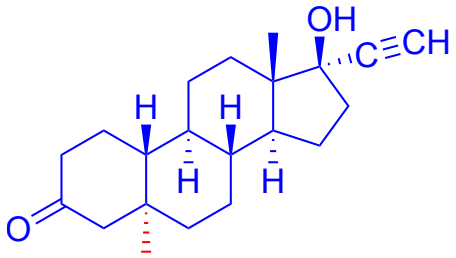
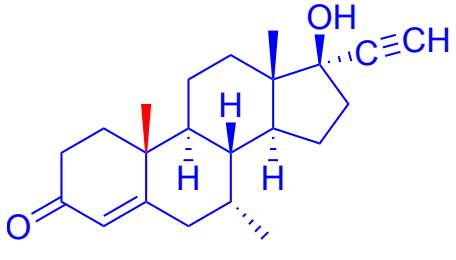
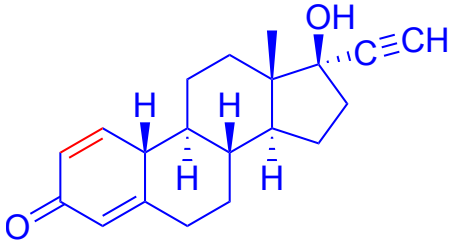
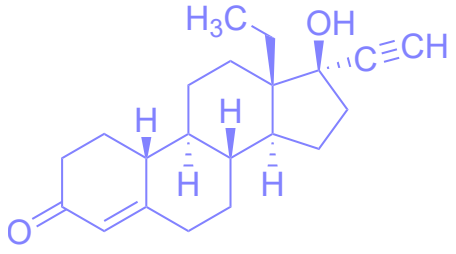
continued on next page

Code	Structure	SVM distance
36		-0.579352455
37		-0.276284051
38		-0.085681749
39		-0.321823682
40		-0.734332919
41		-0.421222024

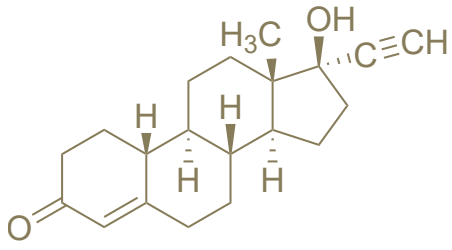
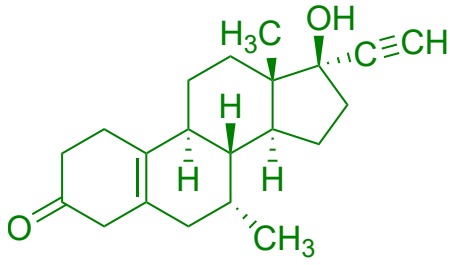
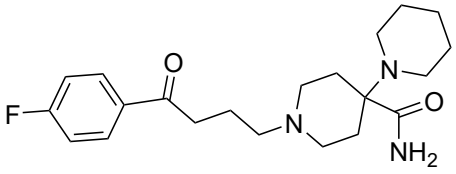
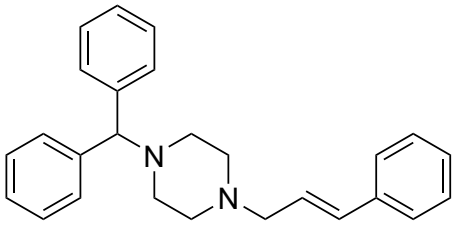
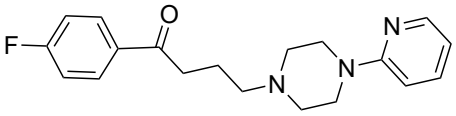
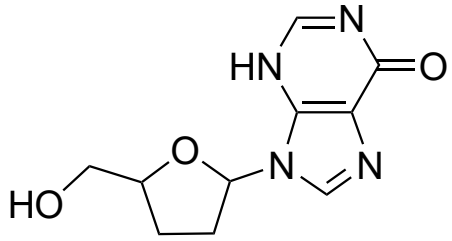
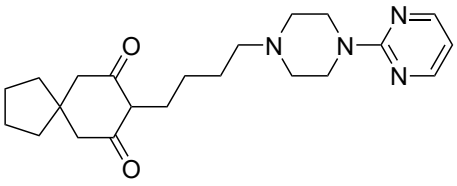
continued on next page

Code	Structure	SVM distance
42		-0.676726886
43		-0.058029637
44		0.180398212
45		0.021217893
46		-0.201125913

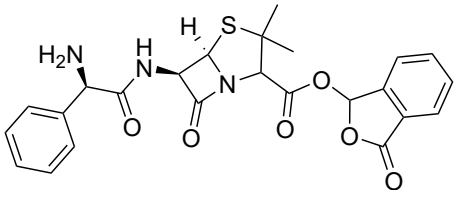
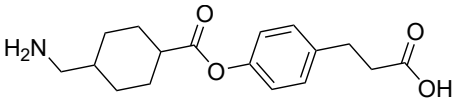
continued on next page

Code	Structure	SVM distance
47		-0.116213968
48		0.038055715
49		-0.520016982
50		-0.294051197
LEV		-1.000120655

continued on next page

Code	Structure	SVM distance
NOR		-0.978720295
TIB		-0.929651231
DID		-0.062700102
CIN		0.834403191
AZA		0.92903296
CET		0.999765416
BUS		1.000249382

continued on next page

Code	Structure	SVM distance
PIP		1.078020185
TAL		1.181821863

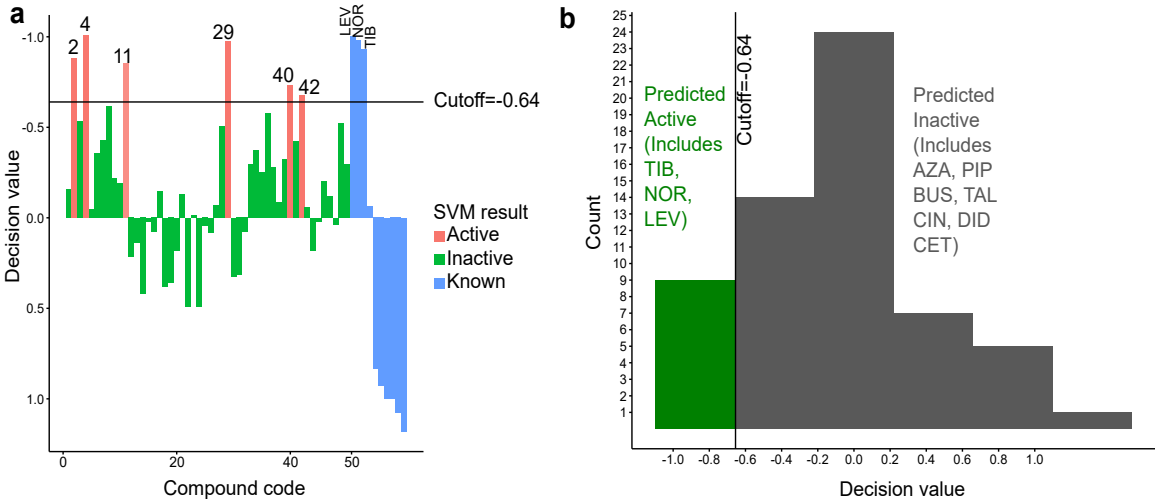


Figure E.2. CANDOCK machine learning for designing new leads. (a) CANDOCK machine learning score for the all designed molecules 1-50 shown in Table S2 along with the training data. Molecules having score less than -0.64 were selected to synthesise and named as 2, 4, 11, 29, 40, 42. (b) Distribution of decision values for the training and prediction set with the selected cut off.

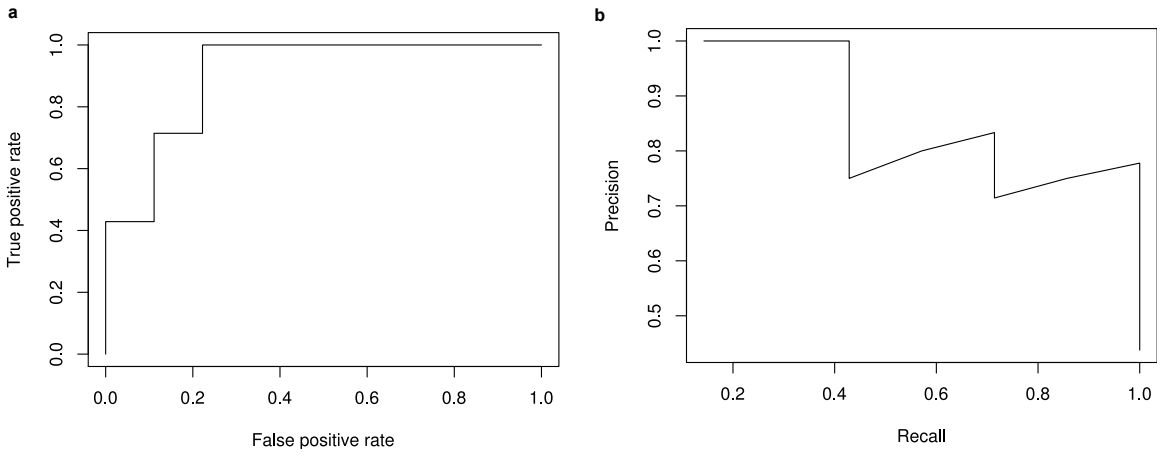


Figure E.3. (a) Receiver Operator Curve for the first round of machine learning. The AUROC is 0.9048, suggesting a highly successful machine learning model. (b) Precision vs Recall plot for the first machine learning. These plots include information gathered from testing the initial predictions from A and B. The F1 score is 0.875 This confirms that the selection of 18 targets is valid.

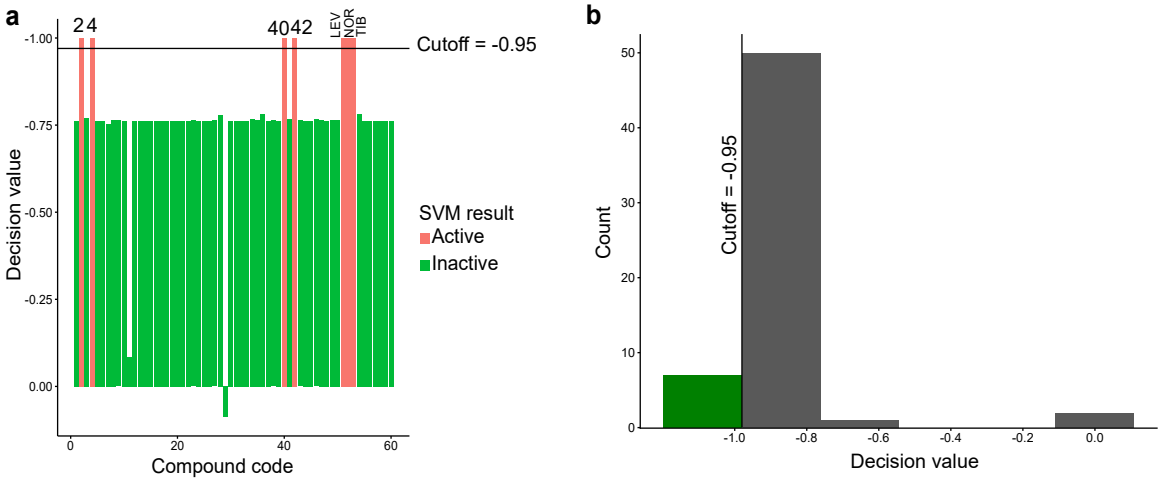


Figure E.4. (a) CANDOCK machine learning score given all molecules (designed and from the original experimental predictions) from the second round of machine learning (2, 4, 40, 41 are included as active, and 11, 29 are included as inactive). These data suggest that no new compounds need to be tested experimentally because it does not predict any new active compounds.

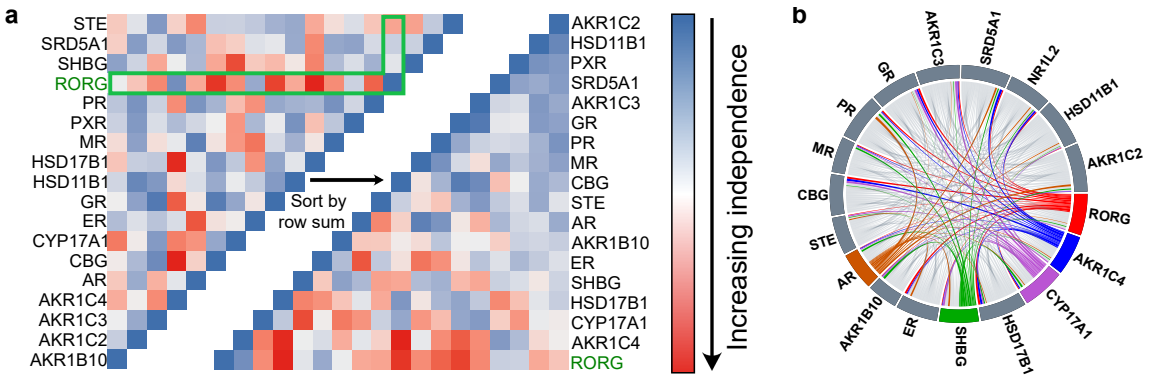


Figure E.5. (a) Ranking of the correlation matrix to obtain the most independent features (bottom is most independent, top is least independent). (b) Chord diagram representation of the independence interactions shown in (a).

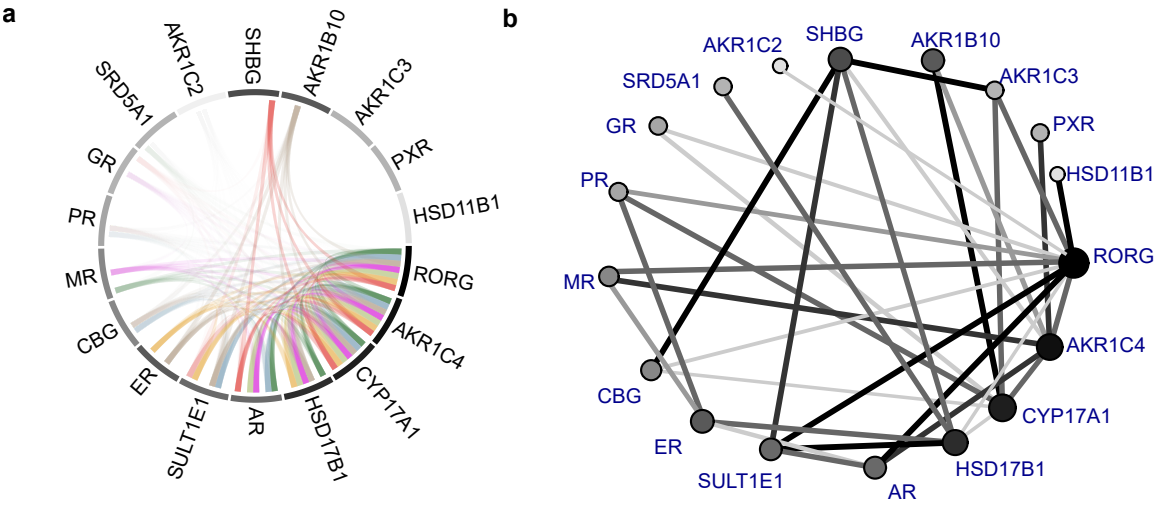


Figure E.6. (a) Chord diagram representation for all compounds over laid on top of one another. (b) Network representation of all features with nodes representing features and edges representing the independence between the nodes they connect. Shading represents the independence value of a given feature or between two features.

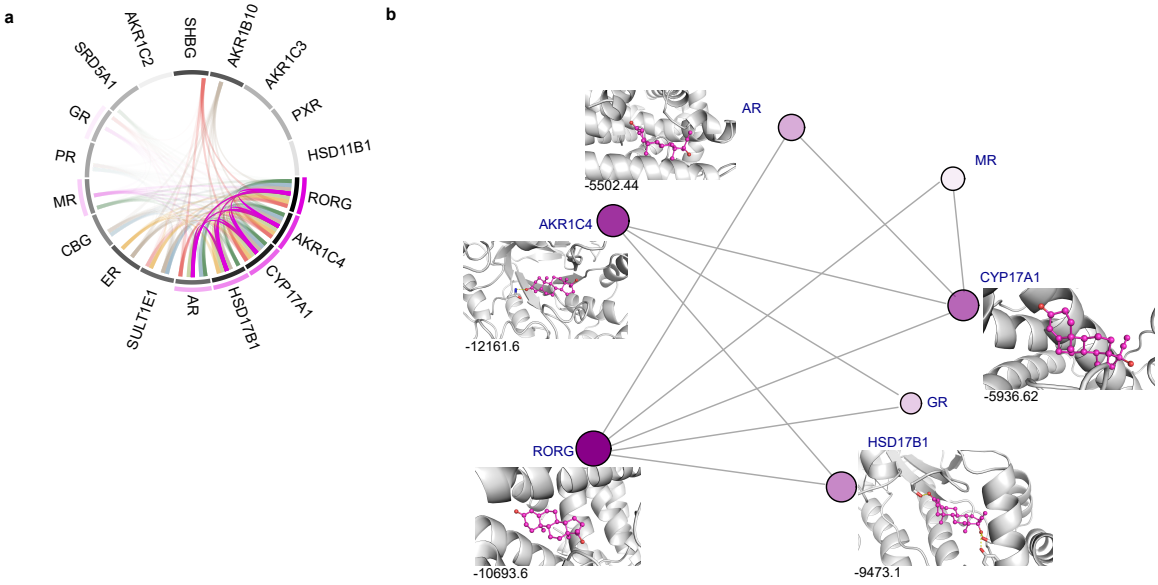


Figure E.7. Specific networks for **4**.

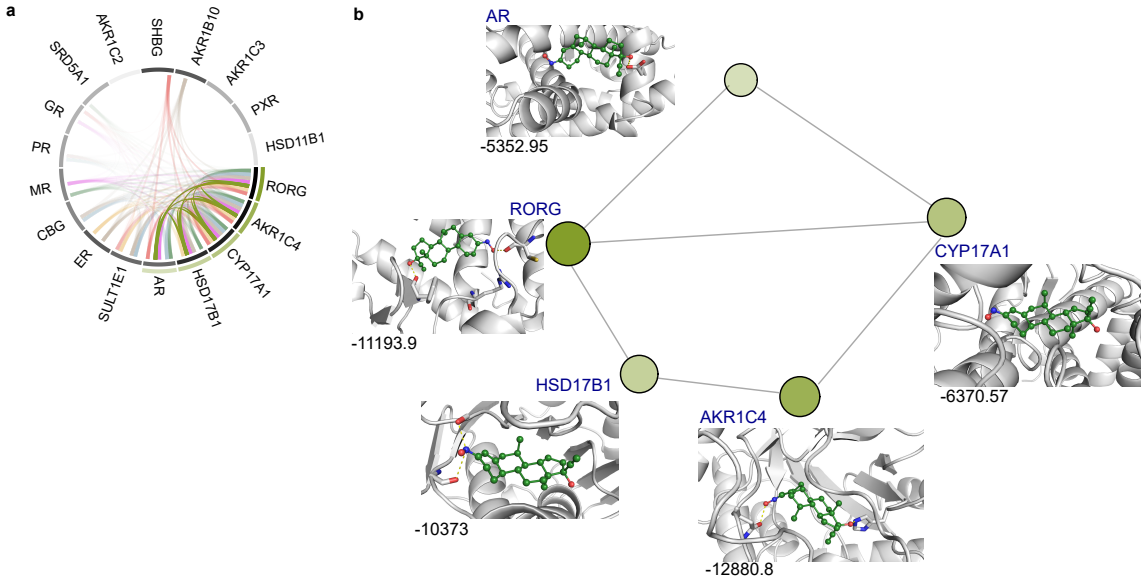


Figure E.8. Specific networks for **40**.

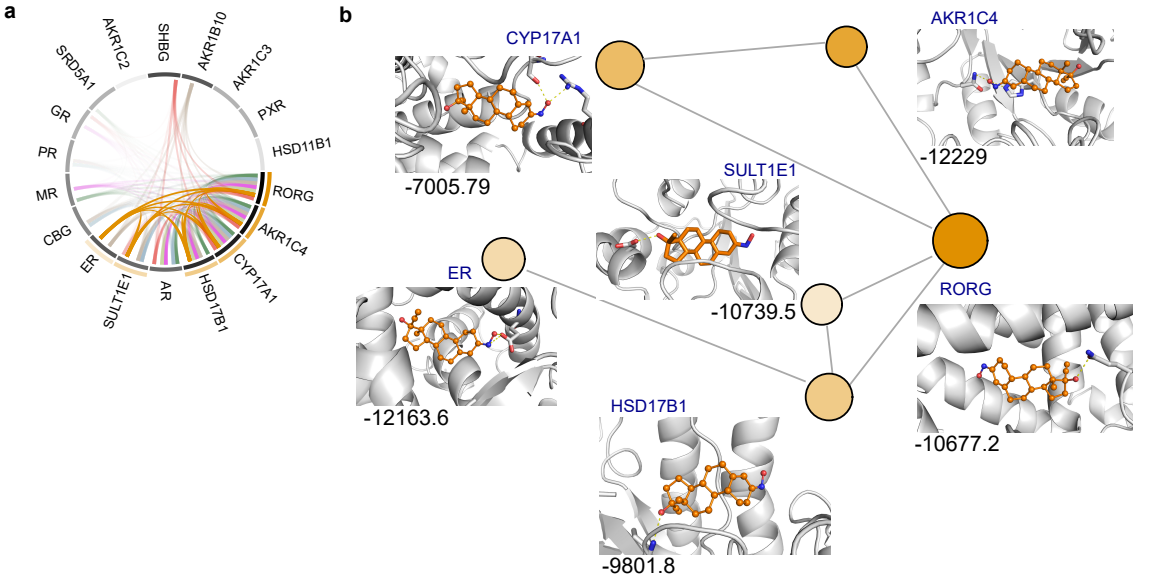


Figure E.9. Specific networks for **42**.

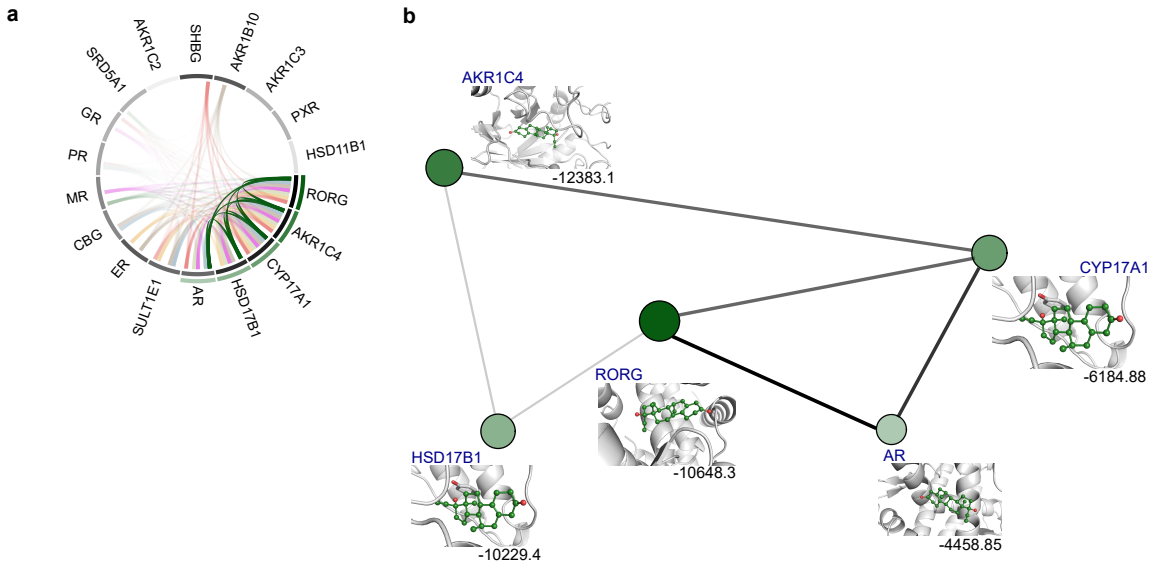


Figure E.10. Specific networks for tibolone.

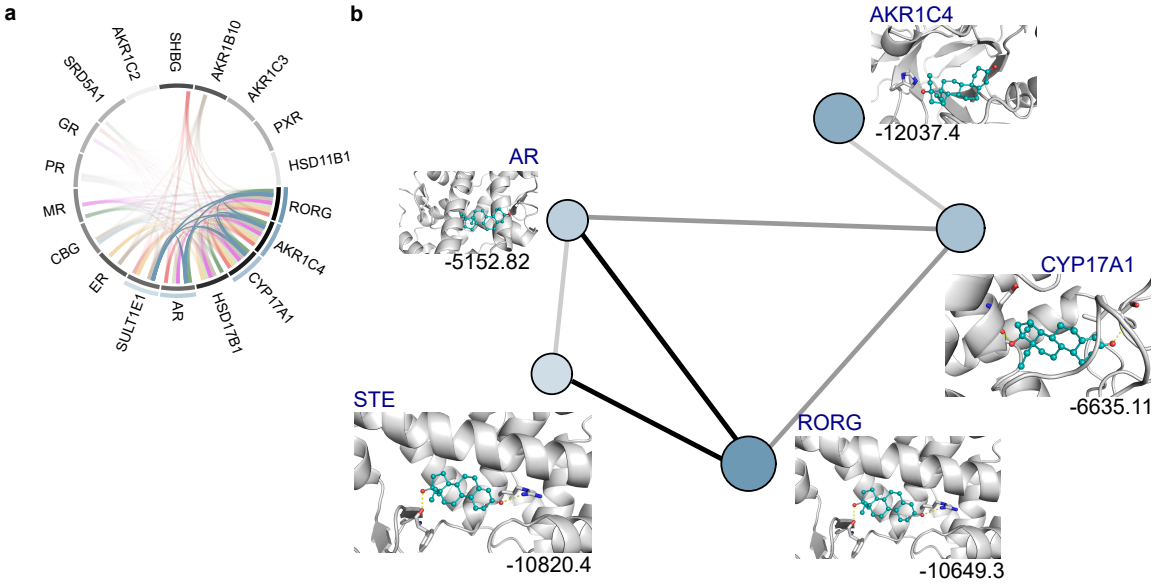


Figure E.11. Specific networks for norethesterone.

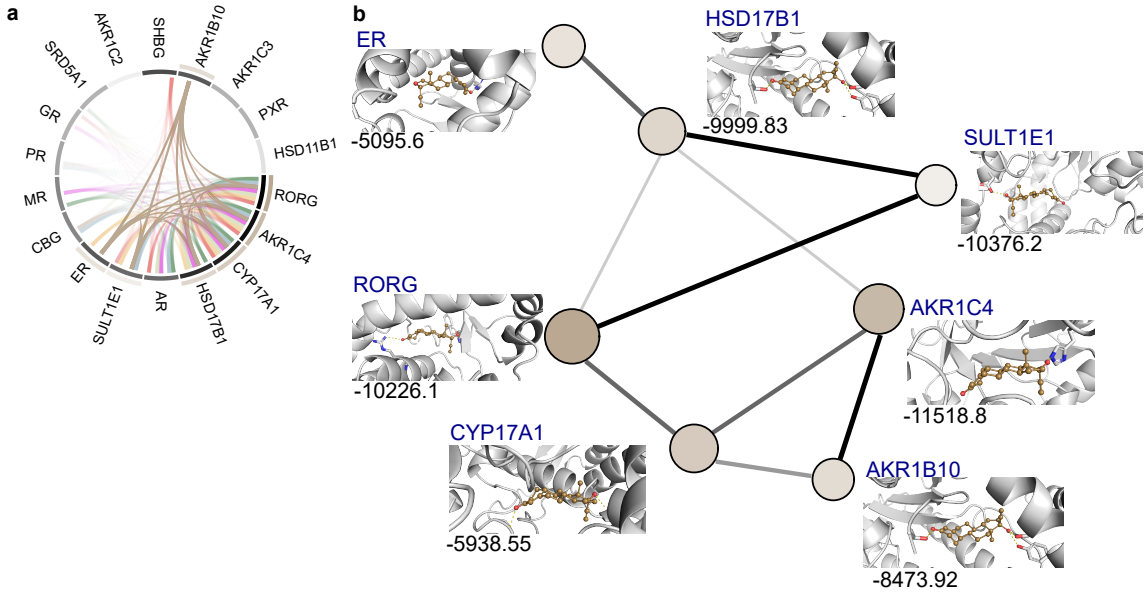


Figure E.12. Specific networks for levonorgestrel.

F. ADDITIONAL INFORMATION FOR CHAPTER 3

F.1 Timing and pose generation benchmarking

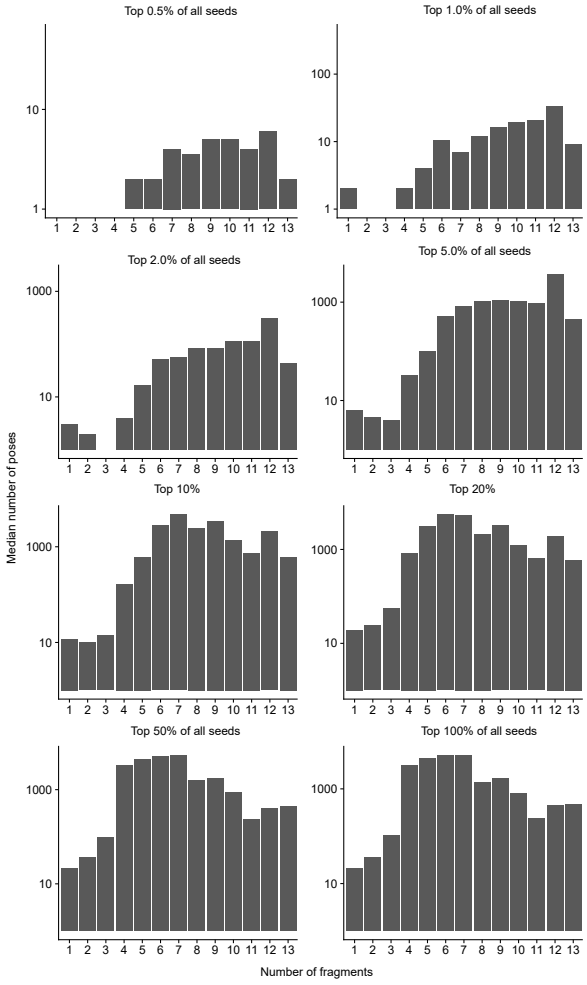


Figure F.1. Median number of poses generated for ligands containing 1-13 fragments divided by the 'Top Seed Percent' parameter.

F.2 Scoring Function atom types

Table F.1.: Atom types considered by the IDATM algorithm implemented in CANDOCK.

Name	Geometry	Expected Bonds	Description
Car	Planar	3	aromatic carbon
C3	Tetrahedral	4	sp3-hybridized carbon
C2	Planar	3	sp2-hybridized carbon
C1	Linear	2	sp-hybridized carbon bonded to 2 other atoms
C1-	Linear	1	sp-hybridized carbon bonded to 1 other atom
Cac	Planar	3	carboxylate carbon
N3+	Tetrahedral	4	sp3-hybridized nitrogen
N3	Tetrahedral	3	sp3-hybridized nitrogen
Npl	Planar	3	sp2-hybridized nitrogen
N2+	Planar	3	sp2-hybridized nitrogen
N2	Planar	2	sp2-hybridized nitrogen
N1+	Linear	2	sp-hybridized nitrogen bonded to 2 other atoms
N1	Linear	1	sp-hybridized nitrogen bonded to 1 other atom
Ntr	Planar	3	nitro nitrogen
Nox	Tetrahedral	4	N-oxide amine
Ng+	Planar	3	guanidinium/amidinium nitrogen
O3	Tetrahedral	2	sp3-hybridized oxygen
O3-	Tetrahedral	1	phosphate or sulfate oxygen sharing formal negative charge
Oar	Planar	2	aromatic oxygen
Oar+	Planar	2	aromatic oxygen
O2	Planar	1	sp2-hybridized oxygen

continued on next page

Table F.1.: *continued*

Name	Geometry	Expected Bonds	Description
O2-	Planar	1	carboxylate oxygen sharing formal negative charge
O1	Linear	1	nitro group oxygen
O1+	Linear	1	sp-hybridized oxygen
S3+	Tetrahedral	3	sp-hybridized oxygen
S3	Tetrahedral	2	sp3-hybridized sulfur
S3-	Tetrahedral	1	sp3-hybridized sulfur
S2	Planar	1	thiophosphate sulfur
Sar	Planar	2	sp2-hybridized sulfur
Sac	Tetrahedral	4	aromatic sulfur
Son	Tetrahedral	4	sulfate
Sxd	Tetrahedral	3	sulfone sulfur
S	Tetrahedral	4	sulfoxide sulfur
Pac	Tetrahedral	4	other sulfur
Pox	Tetrahedral	4	phosphate phosphorus
P3+	Tetrahedral	4	P-oxide phosphorus
P	Trigonal Bipyramidal	5	sp3-hybridized phosphorus
HC	Single	1	other phosphorus
H	Single	1	hydrogen bonded to carbon
DC	Single	1	other hydrogen
D	Single	1	deuterium bonded to carbon
F	Single	1	other deuterium
Cl	Single	1	fluoride
Br	Single	1	chloride
I	Single	1	bromide
Si	Tetrahedral	4	iodide
Mg	Ion	0	silicon
Mn	Ion	0	magnesium
Zn	Ion	0	manganese
Ca	Ion	0	zinc
Na	Ion	0	calcium
K	Ion	0	sodium
			potassium

continued on next page

Table F.1.: *continued*

Name	Geometry	Expected Bonds	Description
Fe	Ion	0	iron
Co	Ion	0	cobalt
Cu	Ion	0	copper
Ni	Ion	0	nickel

F.3 Scoring function correlation to pose deviations

Table F.2.: Correlations between score and small molecule RMSD calculated and summarized over the entire CASF-2016 benchmarking set. Results are provided for poses generated from the top 20% of seeds.

SF	Rigid protein		Semi-flexible protein		Fully-flexible protein	
	Average	Median	Average	Median	Average	Median
rmr4	0.095	0.049	0.140035	0.079644	0.140	0.080
rmr5	0.145	0.102	0.179655	0.112795	0.180	0.113
rmr6	0.176	0.115	0.18871	0.130094	0.189	0.130
rmr7	0.178	0.112	0.207084	0.139238	0.207	0.139
rmr8	0.190	0.112	0.211746	0.165313	0.212	0.165
rmr9	0.189	0.123	0.214307	0.18227	0.214	0.182
rmr10	0.203	0.141	0.236163	0.212437	0.236	0.212
rmr11	0.216	0.149	0.252629	0.230083	0.253	0.230
rmr12	0.225	0.165	0.262296	0.246224	0.262	0.246
rmr13	0.222	0.181	0.261284	0.256056	0.261	0.256
rmr14	0.210	0.169	0.248759	0.235548	0.249	0.236
rmr15	0.183	0.121	0.217107	0.195525	0.217	0.196
rmc4	0.307	0.279	0.361189	0.359725	0.361	0.360
rmc5	0.413	0.416	0.424266	0.425995	0.424	0.426
rmc6	0.459	0.473	0.462102	0.470914	0.462	0.471
rmc7	0.476	0.501	0.494936	0.497759	0.495	0.498
rmc8	0.498	0.519	0.517914	0.528137	0.518	0.528
rmc9	0.523	0.542	0.537665	0.555256	0.538	0.555
rmc10	0.537	0.560	0.550387	0.558608	0.550	0.559
rmc11	0.539	0.562	0.554441	0.564875	0.554	0.565
rmc12	0.546	0.571	0.561478	0.579258	0.561	0.579
rmc13	0.546	0.576	0.562225	0.584367	0.562	0.584
rmc14	0.556	0.587	0.56007	0.586941	0.560	0.587
rmc15	0.559	0.581	0.558598	0.58669	0.559	0.587
fmr4	0.081	0.027	0.120917	0.070754	0.121	0.071

continued on next page

Table F.2.: *continued*

SF	Rigid protein		Semi-flexible protein		Fully-flexible protein	
	Average	Median	Average	Median	Average	Median
fmr5	0.107	0.063	0.14436	0.094075	0.144	0.094
fmr6	0.125	0.064	0.154699	0.106345	0.155	0.106
fmr7	0.132	0.069	0.165974	0.108041	0.166	0.108
fmr8	0.152	0.079	0.176905	0.101773	0.177	0.102
fmr9	0.149	0.078	0.183293	0.112085	0.183	0.112
fmr10	0.149	0.070	0.182985	0.11354	0.183	0.114
fmr11	0.146	0.075	0.178826	0.118613	0.179	0.119
fmr12	0.135	0.062	0.164336	0.114359	0.164	0.114
fmr13	0.116	0.048	0.140079	0.083963	0.140	0.084
fmr14	0.093	0.023	0.110206	0.053444	0.110	0.053
fmr15	0.062	-0.014	0.070699	0.008513	0.071	0.009
fmc4	0.307	0.270	0.359536	0.353402	0.360	0.353
fmc5	0.424	0.427	0.428663	0.429932	0.429	0.430
fmc6	0.468	0.480	0.468468	0.480115	0.468	0.480
fmc7	0.486	0.510	0.501306	0.508722	0.501	0.509
fmc8	0.505	0.527	0.523356	0.530233	0.523	0.530
fmc9	0.535	0.549	0.541273	0.563213	0.541	0.563
fmc10	0.539	0.560	0.551955	0.563304	0.552	0.563
fmc11	0.539	0.562	0.553678	0.569521	0.554	0.570
fmc12	0.545	0.569	0.560537	0.581049	0.561	0.581
fmc13	0.546	0.580	0.561942	0.582873	0.562	0.583
fmc14	0.556	0.587	0.558451	0.586431	0.558	0.586
fmc15	0.559	0.582	0.556559	0.583299	0.557	0.583
rcr4	0.050	-0.015	0.115444	0.052879	0.115	0.053
rcr5	0.053	-0.015	0.103434	0.042769	0.103	0.043
rcr6	0.056	-0.016	0.095697	0.036087	0.096	0.036
rcr7	0.047	-0.018	0.08391	0.02515	0.084	0.025
rcr8	0.057	-0.010	0.090313	0.022349	0.090	0.022
rcr9	0.053	-0.013	0.084556	0.032991	0.085	0.033
rcr10	0.039	-0.025	0.06555	0.00743	0.066	0.007
rcr11	0.031	-0.035	0.055276	0.009913	0.055	0.010
rcr12	0.031	-0.034	0.054034	0.007386	0.054	0.007
rcr13	0.038	-0.031	0.064184	0.023671	0.064	0.024
rcr14	0.057	-0.017	0.088855	0.036508	0.089	0.037
rcr15	0.081	-0.006	0.119669	0.068518	0.120	0.069
rcc4	0.067	0.001	0.118727	0.047455	0.119	0.047
rcc5	0.075	0.007	0.112314	0.046986	0.112	0.047
rcc6	0.073	0.013	0.099047	0.039596	0.099	0.040
rcc7	0.056	-0.005	0.078413	0.021166	0.078	0.021

continued on next page

Table F.2.: *continued*

SF	Rigid protein		Semi-flexible protein		Fully-flexible protein	
	Average	Median	Average	Median	Average	Median
rcc8	0.051	-0.012	0.066573	-1.97E-04	0.067	0.000
rcc9	0.036	-0.028	0.047623	-0.0104	0.048	-0.010
rcc10	0.020	-0.053	0.024173	-0.02832	0.024	-0.028
rcc11	0.011	-0.073	0.013487	-0.04264	0.013	-0.043
rcc12	0.013	-0.072	0.013389	-0.04284	0.013	-0.043
rcc13	0.021	-0.061	0.02453	-0.03049	0.025	-0.030
rcc14	0.040	-0.044	0.049441	-0.00867	0.049	-0.009
rcc15	0.065	-0.010	0.080644	0.019915	0.081	0.020
fcr4	0.050	-0.015	0.114892	0.061528	0.115	0.062
fcr5	0.062	-0.002	0.1126	0.060021	0.113	0.060
fcr6	0.071	-0.001	0.116733	0.055982	0.117	0.056
fcr7	0.074	0.003	0.122048	0.056749	0.122	0.057
fcr8	0.084	0.014	0.129655	0.070466	0.130	0.070
fcr9	0.086	0.007	0.131851	0.072608	0.132	0.073
fcr10	0.091	0.011	0.137708	0.071559	0.138	0.072
fcr11	0.102	0.033	0.153892	0.105595	0.154	0.106
fcr12	0.116	0.047	0.171489	0.114962	0.171	0.115
fcr13	0.133	0.063	0.193219	0.14408	0.193	0.144
fcr14	0.157	0.091	0.221047	0.180568	0.221	0.181
fcr15	0.177	0.116	0.243947	0.208698	0.244	0.209
fcc4	0.063	-0.006	0.112292	0.05447	0.112	0.054
fcc5	0.075	0.002	0.110992	0.055188	0.111	0.055
fcc6	0.084	0.019	0.116043	0.056761	0.116	0.057
fcc7	0.088	0.026	0.12244	0.058866	0.122	0.059
fcc8	0.098	0.021	0.130706	0.063725	0.131	0.064
fcc9	0.101	0.029	0.134228	0.069798	0.134	0.070
fcc10	0.105	0.036	0.140838	0.075632	0.141	0.076
fcc11	0.117	0.053	0.157337	0.100401	0.157	0.100
fcc12	0.132	0.071	0.175309	0.121073	0.175	0.121
fcc13	0.146	0.086	0.192998	0.13832	0.193	0.138
fcc14	0.163	0.107	0.214633	0.165242	0.215	0.165
fcc15	0.179	0.126	0.232661	0.192197	0.233	0.192

F.4 Correlation between score and binding affinity for each protein in CASF-2016

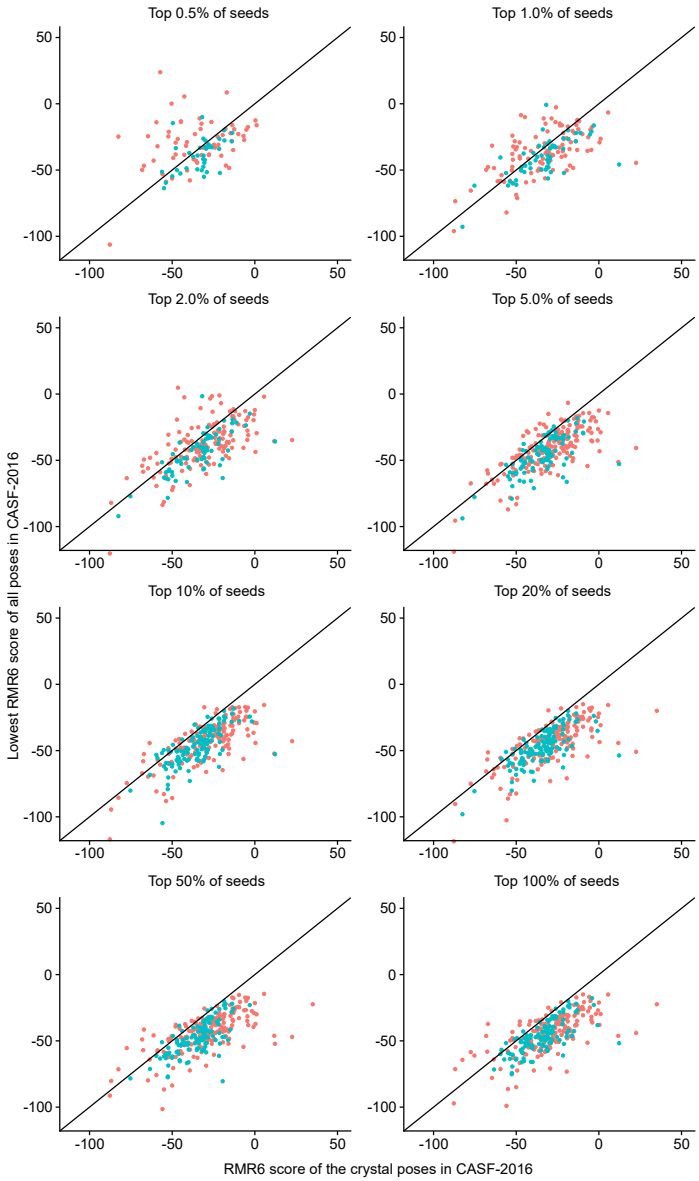


Figure F.2. The lowest RMR6 score obtained for each cocrystal is plotted against the RMR6 score of the crystal pose. Poses within 2.0 Å of the crystal pose are shown in blue (success) while poses with RMSD $>$ 2.0 Å are shown in red. The majority of points on this graph cluster below the y=x line, indicating that the RMR6 scoring function incorrectly scores several poses more favorably than the crystal pose, regardless of if the pose is close to the crystal pose. Therefore, there are potential improvements to be made for this scoring function.

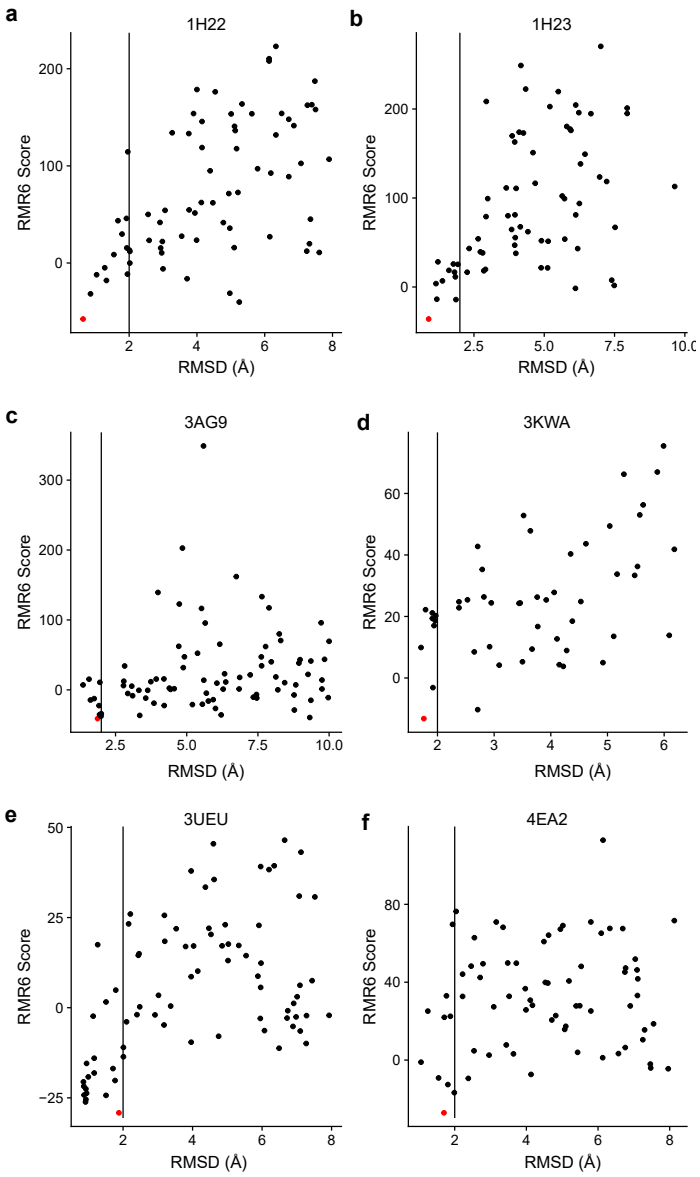


Figure F.3. Sheep plots for the 6 failure cases detailed in the results and discussion section. In each plot, the RMSD of a CASF-2016 decoy pose is plotted against its RMR6 score where the pose with the lowest RMR6 score is shown in red.

Table F.3.: Pearson correlations for all ligands in CASF-2016 using various scoring functions to select the representative pose for the protein-ligand complex and rank the activity of the ligand versus other ligands for the same protein. Here, the poses are generated by CANDOCK and not supplied by the benchmark. Results are provided for poses generated from the top 20% of seeds.

Selector:	RMSD		RMR6		RMC15	
Ranker	RMR6	RMC15	RMR6	RMC15	RMR6	RMC15
3-DEHYDROQUINATE DEHYDRATASE	-0.716	-0.876	-0.852	-0.875	0.591	-0.874
ACETYLCHOLINE RECEPTOR	0.335	0.339	0.476	0.350	0.772	0.326
ACETYLCHOLINE BINDING PROTEIN	0.482	-0.193	-0.004	-0.195	0.554	-0.212
ACHE	-0.269	-0.664	-0.474	-0.688	-0.300	-0.652
FUCO2	-0.692	-0.372	-0.606	-0.307	-0.666	-0.304
MA2A1	-0.271	-0.581	0.575	-0.580	0.855	-0.599
AR	-0.919	0.734	-0.738	0.730	-0.776	0.736
TrpD	0.652	-0.905	-0.328	-0.832	0.539	-0.920
β-GLUCOSIDASE A	0.751	0.140	-0.337	0.119	-0.953	0.147
β-LACTAMASE	-0.495	-0.894	-0.681	-0.908	0.835	-0.886
β-LACTOGLOBULIN	-0.981	-0.991	-0.978	-0.997	-0.959	-0.993
β-SECRETASE 1	0.695	-0.116	-0.210	-0.370	0.673	-0.121
BRD4	-0.644	-0.981	-0.579	-0.988	0.393	-0.955
KAP0	0.696	-0.905	-0.619	-0.996	0.785	-0.864
CAII	-0.694	-0.883	0.770	-0.770	0.976	-0.856
COMT	-0.858	-0.870	-0.749	-0.839	-0.687	-0.781
CELL DIVISION PROTEIN KINASE 2	-0.800	-0.899	0.713	-0.918	0.969	-0.879
P53	-0.739	-0.719	0.796	-0.719	0.925	-0.648
PDE5A	-0.649	0.052	-0.138	0.103	-0.711	0.074
CHITINASE A	-0.915	-0.725	-0.833	-0.725	-0.704	-0.682
FACTOR XA	-0.963	-0.753	-0.992	-0.671	0.560	-0.596
FACTOR XI	-0.805	-0.916	-0.864	-0.885	0.989	-0.887
DEHYDROSQUALENE SYNTHASE	-0.353	-0.439	-0.727	-0.468	0.173	-0.432
ENDOTHIAPEPSIN	-0.975	-0.993	-0.903	-0.992	0.971	-0.989
ER	-0.843	0.764	0.501	0.764	-0.293	0.761
GRIA2	-0.814	-0.457	-0.741	-0.458	-0.854	-0.454
GRIK2	-0.977	-0.646	-0.702	-0.632	0.872	-0.577

continued on next page

Table F.3.: *continued*

Selector:	RMSD		RMR6		RMC15	
Ranker	RMR6	RMC15	RMR6	RMC15	RMR6	RMC15
GLYCOGEN PHOSPHORYLASE	0.716	0.383	-0.186	0.407	-0.833	0.341
HSP82	-0.418	0.112	-0.574	0.105	-0.771	0.166
HSP90-ALPHA	-0.728	-0.879	-0.832	-0.882	0.778	-0.873
HIV-1 INTEGRASE	-0.843	-0.954	-0.954	-0.960	0.935	-0.952
HIV-1 PROTEASE	-0.916	-0.573	-0.907	-0.468	-0.926	-0.541
(MMP-1)	-0.680	-0.815	-0.801	-0.808	0.966	-0.796
MK14	-0.680	-0.902	-0.691	-0.903	0.916	-0.900
SAH NUCLEOSIDASE	0.601	0.203	0.601	0.203	0.657	0.206
O-GLCNACASE	-0.974	-0.128	-0.471	-0.129	-0.392	-0.074
PANTOTHENATE SYNTHETASE	-0.873	-0.960	-0.764	-0.961	0.835	-0.962
PPARG	-0.974	-0.992	-0.971	-0.989	0.902	-0.983
PROTEIN-TYROSINE PHOSPHATASE 1B	0.304	-0.947	-0.080	-0.805	0.759	-0.874
QTRT2	-0.875	-0.697	-0.875	-0.697	-0.676	-0.703
RIBONUCLEASE A	-0.701	-0.842	-0.908	-0.849	0.708	-0.836
RNA-DIRECTED RNA POLYMERASE	-0.794	-0.831	-0.740	-0.828	0.981	-0.856
SERINE/THREONINE PROTEIN KINASE 6	-0.093	-0.706	0.091	-0.885	-0.603	-0.619
CHK1	0.509	0.545	-0.204	0.637	0.963	0.386
PIM-1	0.656	0.447	0.423	0.436	-0.568	0.424
TANKYRASE-2	-0.918	-0.854	-0.970	-0.881	0.491	-0.825
THERMOLYSIN	-0.610	0.178	-0.546	-0.272	-0.700	0.149
THROMBIN	-0.798	0.154	0.473	0.188	0.562	0.155
TRANSCRIPTION POLYPEPTIDE 2	0.789	-0.772	0.784	-0.756	0.908	-0.770
TRANSPORTER	-0.376	0.497	-0.284	0.509	-0.352	0.510
TRYPSIN BETA	-0.905	-0.805	-0.927	-0.804	0.653	-0.769
ABL1	0.929	-0.119	0.187	0.093	0.847	-0.221
ITK/TSK	0.879	0.749	0.183	0.747	-0.786	0.755
JAK1	-0.552	-0.093	-0.292	-0.154	0.234	-0.079
JAK2	-0.653	-0.883	-0.718	-0.876	0.759	-0.931
UROKINASE-TYPE ACTIVATOR	-0.909	-0.927	-0.908	-0.922	0.882	-0.924

F.5 AutoDOCK Vina results for the PINC Benchmark

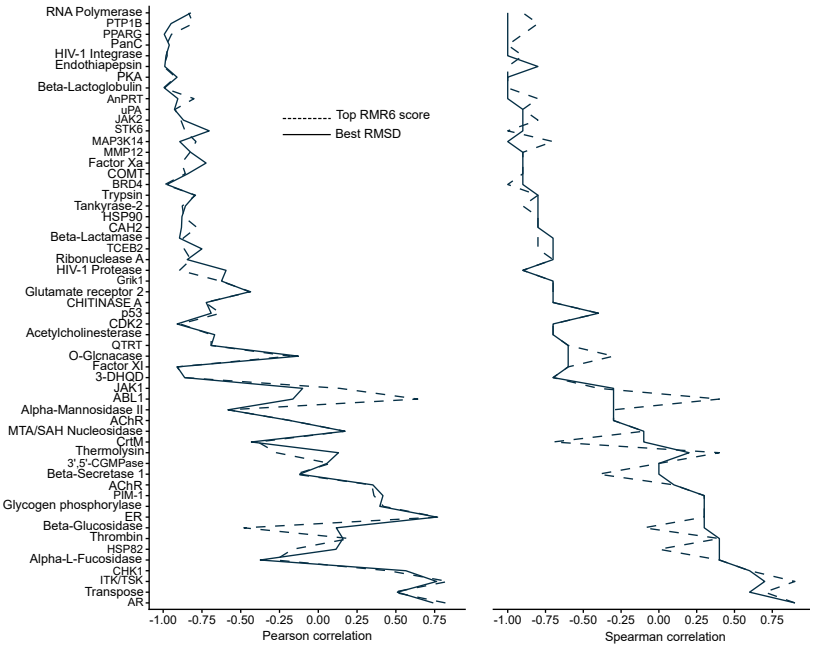


Figure F.4. Rigid protein docking correlations between the RMC15 score and the measured pKd/pKi of the compounds in CASF-2016 for each protein target. A negative correlation is expected as a decrease in score (an estimation of free energy change) should result in an increase in the negative log of the binding coefficient. The representative docked ligand pose for ranking was selected with either the lowest RMSD or the best RMR6 score criterion. Results are provided for poses generated from the top 20% of seeds.

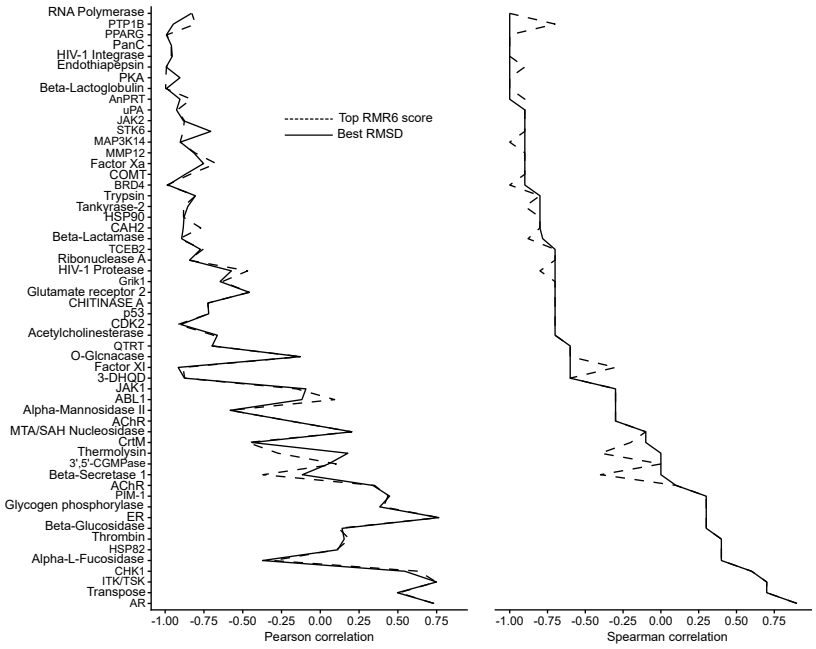


Figure F.5. Semi-flexible protein docking correlations between the RMC15 score and the measured pKd/pKi of the compounds in CASF-2016 for each protein target. The representative docked ligand pose for ranking was selected with either the lowest RMSD or the best RMR6 score criterion. Results are provided for poses generated from the top 20% of seeds.

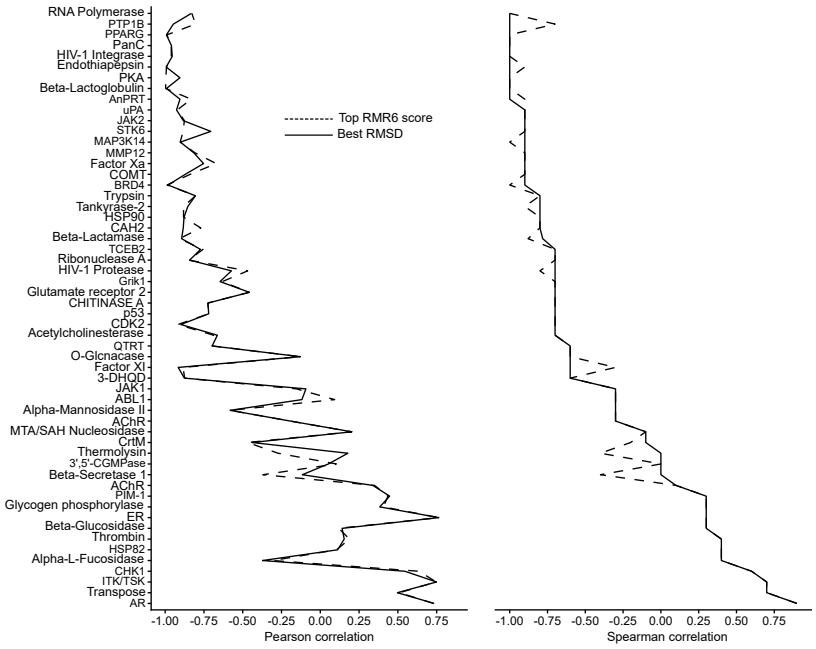


Figure F.6. Fully flexible protein docking correlations between the RMC15 score and the measured pKd/pKi of the compounds in CASF-2016 for each protein target. The representative docked ligand pose for ranking was selected with either the lowest RMSD or the best RMR6 score criterion. Results are provided for poses generated from the top 20% of seeds.

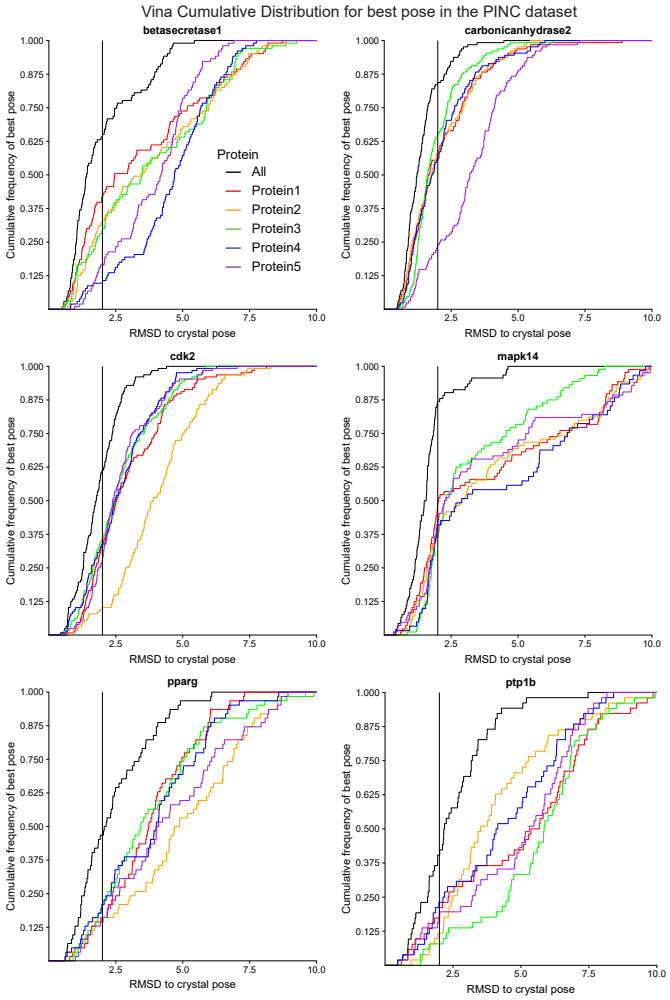


Figure F.7. . Cumulative distributions for the best pose produced by AutoDOCK Vina on the PINC benchmarking set using the top 20% of all seeds.

G. ADDITIONAL INFORMATION FOR CHAPTER 4

G.1 Details of neural networks

Table G.1.
The final optimization parameters for the IR+MS model

Layer size	Dropout
237	0.457866692938781
170	0.26437107014663824
Batch size	178

Table G.2.
For the IR model

Layer size	Dropout
240	0.3820803111613069
200	0.38822353533309584
131	0.008815281710900874
Batch size	153

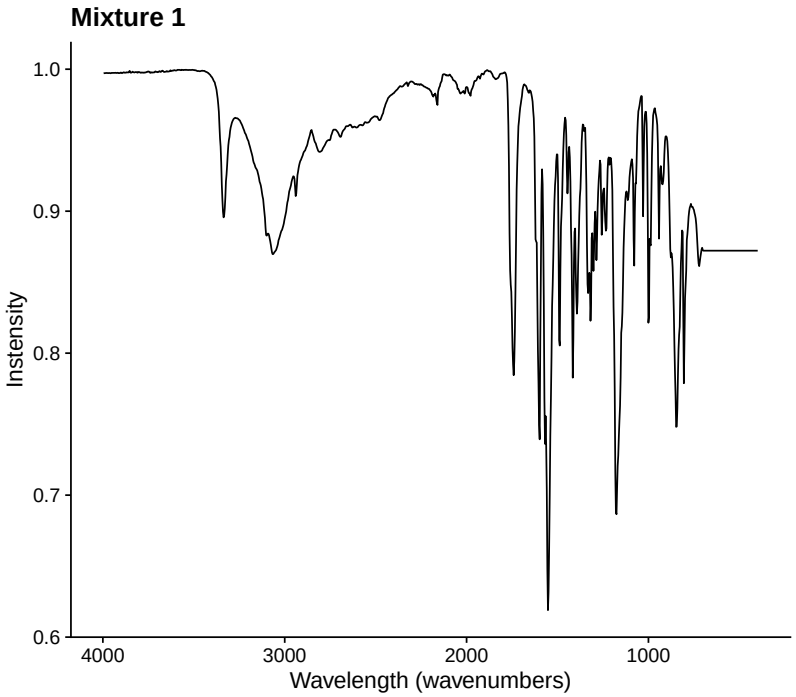


Figure G.1. IR Spectra for Mixture 1

G.2 IR Spectra for testing the model

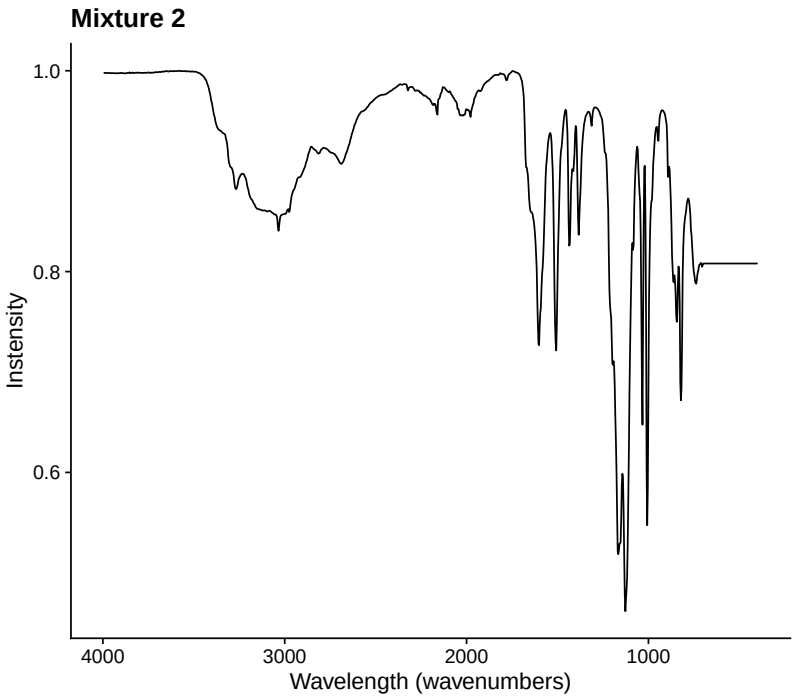


Figure G.2. IR Spectra for Mixture 2

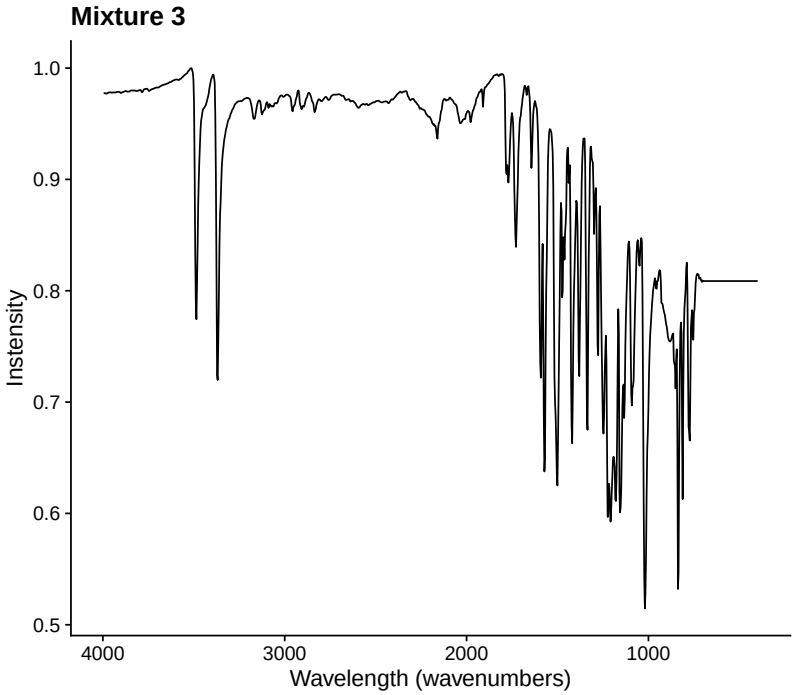


Figure G.3. IR Spectra for Mixture 3

G.3 Performance details

Table G.3.
Functional group F-1 scores for the random forest model.

Functional Group	Fold 1	Fold2	Fold 3	Fold 4	Fold 5
Alkane	0.979	0.976	0.975	0.978	0.982
Alkene	0.648	0.616	0.578	0.634	0.689
Alkyne	0.367	0.629	0.509	0.458	0.676
Alcohols	0.902	0.900	0.919	0.922	0.924
Amines	0.647	0.695	0.692	0.695	0.696
Nitriles	0.246	0.222	0.181	0.200	0.172
Aromatics	0.957	0.958	0.959	0.965	0.966
Alkyl halides	0.679	0.701	0.723	0.665	0.667
Esters	0.805	0.842	0.860	0.816	0.855
Ketones	0.757	0.720	0.736	0.725	0.704
Carboxylic acids	0.931	0.944	0.928	0.921	0.961
Acyl halides	0.222	0.105	0.100	0.400	0.111
Amides	0.050	0.258	0.071	0.303	0.148

Table G.4.
Functional group F-1 scores for the neural network model trained on only IR spectra

Functional Group	Training set F1	Validation set F1
Alkane	0.983057	0.962597
Alkene	0.866956	0.771962
Alkyne	0.891495	0.824410
Alcohols	0.978567	0.946291
Amines	0.916645	0.829724
Nitriles	0.682049	0.493131
Aromatics	0.987723	0.971455
Alkyl halides	0.883381	0.794842
Esters	0.945287	0.906326
Ketones	0.933960	0.851401
Carboxylic acids	0.970379	0.938528
Acyl halides	0.901172	0.767982
Amides	0.726499	0.562125

Table G.5.
Functional group F-1 scores for the neural network model trained on only MS spectra

Functional Group	Training set F1	Validation set F1
Alkane	0.999203	0.988438
Alkene	0.854542	0.629076
Alkyne	0.823007	0.586774
Alcohols	0.944003	0.738918
Amines	0.940923	0.686877
Nitriles	0.821942	0.431075
Aromatics	0.999599	0.990165
Alkyl halides	0.990067	0.916458
Esters	0.844197	0.561388
Ketones	0.843226	0.565298
Carboxylic acids	0.981144	0.574958
Acyl halides	0.589059	0.210796
Amides	0.701243	0.282036

Table G.6.
Functional group F-1 scores for the neural network model trained on both IR and MS spectra

Functional Group	Training set F1	Validation set F1
Alkane	0.992912	0.969812
Alkene	0.934595	0.820889
Alkyne	0.928958	0.833759
Alcohols	0.982569	0.943450
Amines	0.964915	0.867276
Nitriles	0.846275	0.617405
Aromatics	0.993621	0.979649
Alkyl halides	0.950192	0.855821
Esters	0.974097	0.913863
Ketones	0.956765	0.855171
Carboxylic acids	0.985346	0.932786
Acyl halides	0.938061	0.778668
Amides	0.813684	0.563190

Table G.7.
Functional group F-1 scores for single neural networks trained on both IR and MS spectra

Functional Group	Fold1	Fold2	Fold3	Fold4	Fold5
Alkane	0.975	0.974	0.977	0.984	0.971
Alkene	0.820	0.796	0.813	0.782	0.833
Alkyne	0.818	0.852	0.857	0.900	0.707
Alcohols	0.951	0.931	0.932	0.931	0.929
Amines	0.879	0.879	0.879	0.853	0.861
Nitriles	0.647	0.715	0.533	0.637	0.632
Aromatics	0.979	0.982	0.978	0.984	0.982
Alkyl halides	0.871	0.849	0.878	0.881	0.875
Esters	0.929	0.900	0.914	0.924	0.923
Ketones	0.853	0.892	0.817	0.881	0.828
Carboxylic acids	0.920	0.900	0.955	0.940	0.897
Acyl halides	0.785	0.857	0.833	0.838	0.709
Amides	0.612	0.640	0.478	0.612	0.690

Table G.8.
MPR and MF1 values for a multitask model trained on only IR spectra

	Molecular Perfection Rate	Molecular F-1
Training set	85.2767%	0.963177
Validation set	72.5011%	0.923357

Table G.9.
MPR and MF1 values for a multitask model trained on IR and MS spectra

	Molecular Perfection Rate	Molecular F-1
Training set	92.5571%	0.982041
Validation set	74.9085%	0.931506

Table G.10.
MPR and MF1 values for a multitask model trained on only IR spectra with the new definitions of functional groups

	Molecular Perfection Rate	Molecular F-1
Training set	79.1323%	0.955077
Validation set	64.0335%	0.909212

Table G.11.
MPR and MF1 values for a multitask model trained on IR and MS spectra

	Molecular Perfection Rate	Molecular F-1
Training set	87.8871%	0.975642
Validation set	65.2510%	0.912017

Table G.12.
Functional group F-1 scores for a model trained on only IR with the new definitions of functional groups

Functional Group	Training set F1	Validation set F1
Alkanes	0.966563	0.932969
Alkenes	0.898341	0.823709
Alkynes	0.946598	0.847545
Alcohols	0.981538	0.957765
Amines	0.948083	0.877436
Aitriles	0.739183	0.525128
Aromatics	0.991503	0.976025
Alkyl halides	0.907264	0.825761
Esters	0.978914	0.933366
Ketones	0.952114	0.882585
Aldehydes	0.982074	0.927797
Carboxylic acids	0.974353	0.944752
Acyl halides	0.936764	0.822867
Amides	0.783791	0.620740
Methyl	0.962598	0.928545
Ether	0.977310	0.935875
Nitro	0.986419	0.953173

Table G.13.
Functional group F-1 scores for a model trained on IR and MS spectra with the new definitions of functional groups

Functional Group	Training set F1	Validation set F1
Alkane	0.983110	0.935748
Alkene	0.951914	0.825343
Alkyne	0.966157	0.869274
Alcohols	0.985552	0.935951
Amines	0.969121	0.873207
Nitriles	0.887506	0.598101
Aromatics	0.997007	0.981913
Alkyl halides	0.966182	0.865727
Esters	0.970721	0.912860
Ketones	0.965129	0.867477
Aldehydes	0.979790	0.903850
Carboxylic acids	0.977540	0.930756
Acyl halides	0.945896	0.788083
Amides	0.832065	0.595560
Methyls	0.977781	0.932062
Ethers	0.984980	0.923053
Nitros	0.990951	0.931536

Table G.14.
MPR and MF1 values for a model trained using an autoencoder on only IR and with the new definitions of functional groups

	Molecular Perfection Rate	Molecular F-1
Training set	78.8955%	0.955907
Validation set	62.5593%	0.904820

Table G.15.
MPR and MF1 values for a model trained using an autoencoder on IR and MS and with the new definitions of functional groups

	Molecular Perfection Rate	Molecular F-1
Training set	86.8895%	0.973454
Validation set	62.5726%	0.905013

Table G.16.
Functional group F-1 scores for a model trained using an autoencoder on only IR with the new definitions of functional groups

Functional Group	Training set F1	Validation set F1
Alkane	0.968777	0.93169
Alkene	0.907346	0.812864
Alkyne	0.945042	0.851205
Alcohols	0.97892	0.944236
Amines	0.946405	0.852841
Nitriles	0.717182	0.488428
Aromatics	0.992644	0.974879
Alkyl halides	0.907742	0.810426
Esters	0.979923	0.922709
Ketones	0.951888	0.867387
Aldehydes	0.976048	0.918015
Carboxylic acids	0.971139	0.941297
Acyl halides	0.920298	0.791876
Amides	0.788451	0.597016
Methysl	0.963815	0.932059
Ethers	0.973455	0.923417
Nitros	0.98336	0.946973

Table G.17.
Functional group F-1 scores for a model trained using an autoencoder on IR and MS with the new definitions of functional groups

Functional Group	Training set F1	Validation set F1
Alkane	0.984136	0.932257
Alkene	0.947195	0.819603
Alkyne	0.958650	0.848086
Alcohols	0.978334	0.910960
Amines	0.960173	0.852991
Nitriles	0.854644	0.553305
Aromatics	0.996893	0.982649
Alkyl halides	0.963728	0.855594
Esters	0.969606	0.913754
Ketones	0.964384	0.857152
Aldehydes	0.979850	0.866663
Carboxylic acids	0.979510	0.917079
Acyl halides	0.952464	0.736802
Amides	0.844232	0.557778
Methyls	0.978014	0.930977
Ethers	0.980859	0.919104
Nitros	0.987188	0.933380

H. ADDITIONAL INFORMATION FOR CHAPTER 5

H.1 Addition machine learning model results

Table H.1.: Additional diagnostic product branching ratio cutoffs and fingerprint radii for the decision tree model. Here, radius refers to the radius parameter of the Morgan algorithm.

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
01	0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50
02		0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
03		1.00	1.00	0.50	0.00	0.00	0.00	0.00	0.00
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50
07		1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08		1.00	1.00	1.00	1.00	0.84	0.84	0.66	0.33
09		1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50
13		1.00	1.00	1.00	1.00	0.84	0.84	0.66	0.33
14		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
κ		0.22	0.46	0.59	0.59	0.59	0.59	0.59	0.53
01	1	0.50	0.51	0.54	0.50	0.47	1.00	1.00	0.50
02		0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00
03		0.11	0.00	0.08	0.00	0.00	0.33	0.00	0.00
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		0.52	0.59	0.58	0.50	0.44	0.04	0.00	0.00
07		0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00
08		1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.50
09		0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50

continued on next page

Table H.1.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.50
13		1.00	1.00	1.00	1.00	0.88	1.00	1.00	0.50
14		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
κ		0.44	0.59	0.59	0.59	0.57	0.72	0.72	0.53
01	2	0.50	0.51	0.50	0.50	0.47	1.00	1.00	0.25
02		0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00
03		0.11	0.00	0.00	0.00	0.00	0.35	0.00	0.00
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		0.52	0.59	0.50	0.50	0.44	0.04	0.00	0.00
07		0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00
08		1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.25
09		0.00	0.00	0.00	0.00	0.00	0.33	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.25
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.25
13		1.00	1.00	1.00	1.00	0.88	1.00	1.00	0.25
14		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
κ		0.44	0.59	0.46	0.59	0.57	0.72	0.72	0.53
01	3	0.50	0.49	0.49	0.50	1.00	1.00	1.00	0.25
02		0.57	0.50	1.00	0.50	0.42	0.39	0.38	0.40
03		0.67	0.00	0.28	0.48	0.67	0.68	0.33	0.13
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		1.00	0.66	0.49	0.50	0.00	0.00	0.00	0.13
07		0.50	0.00	0.00	0.95	1.00	1.00	0.67	0.00
08		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.25
09		0.00	0.00	0.00	0.95	1.00	1.00	0.67	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.25
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.25
13		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.25
14		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
κ		0.22	0.44	0.44	0.31	0.34	0.34	0.46	0.53

Table H.2.: Regularized logistic regression results for various cutoffs and fingerprint radii. Here, radius refers to the radius parameter of the Morgan algorithm.

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
01	0	0.63	0.44	0.55	0.69	0.77	0.33	0.27	0.21
02		0.82	0.41	0.38	0.22	0.67	0.33	0.27	0.11
03		0.89	0.37	0.21	0.19	0.23	0.28	0.27	0.03
04		0.44	0.81	0.72	0.81	0.80	0.29	0.33	0.39
05		0.13	0.12	0.10	0.15	0.10	0.28	0.27	0.03
06		0.78	0.64	0.56	0.71	0.65	0.28	0.27	0.18
07		0.36	0.34	0.21	0.19	0.12	0.28	0.27	0.03
08		0.79	0.70	0.75	0.75	0.65	0.33	0.27	0.11
09		0.36	0.34	0.21	0.19	0.12	0.28	0.27	0.03
10		0.90	0.70	0.75	0.75	0.93	0.33	0.27	0.17
11		0.81	0.85	0.86	0.84	0.80	0.34	0.33	0.70
12		0.90	0.70	0.75	0.75	0.93	0.33	0.27	0.17
13		0.79	0.70	0.75	0.75	0.65	0.33	0.27	0.11
14		0.98	0.79	0.72	0.81	0.90	0.29	0.33	0.14
κ		0.19	0.44	0.59	0.59	0.46	0.00	0.00	0.19
01	1	0.56	0.48	0.10	0.36	0.56	0.33	0.27	0.20
02		0.65	0.48	0.44	0.22	0.22	0.33	0.27	0.08
03		0.65	0.37	0.06	0.18	0.08	0.28	0.27	0.02
04		0.79	0.71	0.62	0.80	0.69	0.29	0.33	0.55
05		0.13	0.10	0.04	0.16	0.08	0.28	0.27	0.02
06		0.62	0.41	0.12	0.36	0.26	0.28	0.27	0.18
07		0.33	0.17	0.10	0.18	0.08	0.28	0.27	0.04
08		0.83	0.67	0.83	0.75	0.74	0.33	0.27	0.12
09		0.29	0.17	0.10	0.18	0.08	0.28	0.27	0.03
10		0.86	0.74	0.89	0.75	0.82	0.33	0.27	0.15
11		0.83	0.80	0.91	0.84	0.82	0.34	0.33	0.77
12		0.86	0.72	0.89	0.75	0.82	0.33	0.27	0.31
13		0.84	0.68	0.83	0.75	0.74	0.33	0.27	0.24
14		0.95	0.79	0.55	0.80	0.60	0.29	0.33	0.18
κ		0.34	0.57	0.57	0.57	0.72	0.00	0.00	0.36
01	2	0.53	0.48	0.25	0.39	0.23	0.33	0.27	0.18
02		0.63	0.49	0.43	0.22	0.21	0.33	0.27	0.18
03		0.64	0.36	0.21	0.18	0.10	0.28	0.27	0.17
04		0.82	0.72	0.67	0.80	0.67	0.29	0.33	0.32
05		0.12	0.10	0.08	0.16	0.07	0.28	0.27	0.15
06		0.56	0.42	0.25	0.39	0.11	0.28	0.27	0.17
07		0.26	0.16	0.21	0.18	0.08	0.28	0.27	0.17

continued on next page

Table H.2.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
08		0.81	0.67	0.74	0.75	0.75	0.33	0.27	0.17
09		0.19	0.13	0.20	0.18	0.08	0.28	0.27	0.15
10		0.87	0.76	0.74	0.75	0.80	0.33	0.27	0.17
11		0.84	0.81	0.85	0.84	0.84	0.34	0.33	0.35
12		0.85	0.74	0.75	0.75	0.80	0.33	0.27	0.18
13		0.82	0.69	0.75	0.75	0.75	0.33	0.27	0.18
14		0.94	0.78	0.68	0.80	0.67	0.29	0.33	0.32
κ		0.34	0.57	0.57	0.57	0.57	0.00	0.00	0.00
01	3	0.60	0.49	0.25	0.32	0.32	0.42	0.52	0.23
02		0.85	0.75	0.58	0.49	0.54	0.60	0.46	0.39
03		0.49	0.31	0.29	0.27	0.31	0.35	0.26	0.17
04		0.87	0.91	0.84	0.87	0.75	0.76	0.68	0.62
05		0.26	0.12	0.08	0.13	0.10	0.01	0.02	0.07
06		0.76	0.73	0.38	0.45	0.36	0.56	0.26	0.18
07		0.50	0.38	0.35	0.29	0.33	0.35	0.26	0.18
08		0.71	0.68	0.75	0.75	0.59	0.81	0.75	0.09
09		0.43	0.31	0.21	0.20	0.17	0.06	0.06	0.07
10		0.57	0.53	0.51	0.75	0.49	0.64	0.52	0.09
11		0.80	0.85	0.88	0.87	0.83	0.87	0.95	0.68
12		0.76	0.68	0.75	0.75	0.59	0.82	0.76	0.23
13		0.76	0.68	0.75	0.75	0.59	0.81	0.75	0.23
14		0.86	0.81	0.80	0.85	0.73	0.76	0.68	0.60
κ		0.46	0.31	0.44	0.57	0.57	0.31	0.72	0.53

Table H.3.: Generalized Linear Model (GLM) results for various cutoffs and fingerprint radii. Here, radius refers to the radius parameter of the Morgan algorithm.

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
01	0	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
02		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
03		0.00	0.98	1.00	1.00	1.00	0.00	0.00	0.00
04		0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		0.00	0.20	1.00	1.00	1.00	1.00	1.00	0.00
07		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08		0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
09		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

continued on next page

Table H.3.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
13		0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
14		0.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
κ		-0.22	0.31	0.34	0.34	0.34	0.16	0.16	0.53
01	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
02		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
03		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
07		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
09		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
κ		0.16	0.16	0.16	0.16	0.29	0.29	0.29	0.42
01	2	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
02		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
03		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
04		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
07		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
08		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
09		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
κ		0.31	0.31	0.31	0.31	0.42	0.42	0.42	0.55
01	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
02		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
03		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
04		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
05		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
06		0.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00

continued on next page

Table H.3.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
07		1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
08		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
09		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
14		1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
κ		0.31	0.19	0.19	0.19	0.13	0.39	0.39	0.39

Table H.4.: Partial Least Squares (PLS) results for various cutoffs and fingerprint radii. Here, radius refers to the radius parameter of the Morgan algorithm.

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
01	0	0.60	0.42	0.47	0.51	0.59	0.65	0.38	0.38
02		0.68	0.53	0.60	0.51	0.60	0.59	0.43	0.30
03		0.77	0.62	0.49	0.36	0.49	0.50	0.44	0.31
04		0.41	0.56	0.52	0.53	0.50	0.46	0.54	0.60
05		0.21	0.14	0.17	0.26	0.14	0.14	0.24	0.19
06		0.66	0.56	0.53	0.48	0.52	0.61	0.46	0.41
07		0.33	0.35	0.40	0.40	0.32	0.26	0.35	0.24
08		0.64	0.63	0.69	0.66	0.66	0.60	0.46	0.32
09		0.33	0.35	0.40	0.40	0.32	0.26	0.35	0.24
10		0.79	0.68	0.70	0.62	0.73	0.73	0.46	0.35
11		0.68	0.74	0.71	0.69	0.71	0.64	0.62	0.68
12		0.79	0.68	0.70	0.62	0.73	0.73	0.46	0.35
13		0.64	0.63	0.69	0.66	0.66	0.60	0.46	0.32
14		0.91	0.85	0.74	0.54	0.74	0.70	0.59	0.58
κ		0.19	0.19	0.31	0.59	0.31	0.19	0.53	0.53
01	1	0.55	0.50	0.48	0.46	0.45	0.57	0.60	0.33
02		0.58	0.47	0.55	0.48	0.48	0.42	0.48	0.33
03		0.57	0.50	0.38	0.37	0.38	0.38	0.37	0.28
04		0.61	0.67	0.57	0.60	0.58	0.57	0.57	0.58
05		0.27	0.18	0.23	0.25	0.22	0.21	0.23	0.23
06		0.49	0.43	0.39	0.40	0.37	0.44	0.43	0.35
07		0.30	0.22	0.32	0.35	0.29	0.23	0.24	0.29
08		0.71	0.64	0.69	0.65	0.69	0.67	0.67	0.41
09		0.30	0.23	0.32	0.34	0.31	0.23	0.22	0.28
10		0.78	0.70	0.72	0.63	0.72	0.77	0.80	0.41

continued on next page

Table H.4.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
11		0.70	0.75	0.69	0.73	0.72	0.72	0.69	0.68
12		0.75	0.67	0.68	0.61	0.67	0.74	0.75	0.45
13		0.71	0.64	0.68	0.64	0.66	0.67	0.66	0.44
14		0.83	0.81	0.68	0.65	0.66	0.61	0.61	0.54
κ		0.46	0.44	0.44	0.57	0.57	0.72	0.72	0.53
01	2	0.50	0.45	0.47	0.46	0.42	0.40	0.40	0.31
02		0.54	0.51	0.52	0.50	0.46	0.39	0.45	0.30
03		0.55	0.43	0.34	0.36	0.36	0.39	0.34	0.31
04		0.66	0.65	0.58	0.59	0.56	0.52	0.54	0.47
05		0.27	0.22	0.25	0.24	0.23	0.22	0.22	0.26
06		0.50	0.45	0.38	0.37	0.35	0.37	0.34	0.34
07		0.33	0.29	0.31	0.30	0.30	0.29	0.27	0.31
08		0.69	0.65	0.68	0.67	0.65	0.52	0.59	0.33
09		0.30	0.27	0.30	0.30	0.29	0.27	0.25	0.28
10		0.75	0.69	0.71	0.67	0.65	0.50	0.63	0.30
11		0.71	0.68	0.70	0.72	0.74	0.70	0.74	0.59
12		0.70	0.65	0.65	0.63	0.61	0.50	0.58	0.35
13		0.67	0.64	0.64	0.64	0.61	0.52	0.54	0.36
14		0.78	0.72	0.61	0.61	0.60	0.55	0.56	0.44
κ		0.31	0.44	0.44	0.57	0.57	0.57	0.57	0.19
01	3	0.51	0.48	0.50	0.48	0.45	0.44	0.47	0.34
02		0.55	0.57	0.56	0.56	0.52	0.42	0.51	0.33
03		0.57	0.49	0.45	0.42	0.41	0.43	0.35	0.33
04		0.67	0.67	0.68	0.62	0.60	0.53	0.61	0.48
05		0.26	0.22	0.24	0.25	0.24	0.22	0.26	0.26
06		0.56	0.53	0.51	0.44	0.44	0.42	0.44	0.35
07		0.47	0.40	0.42	0.38	0.38	0.35	0.41	0.33
08		0.64	0.63	0.65	0.66	0.63	0.50	0.60	0.34
09		0.33	0.27	0.32	0.31	0.30	0.26	0.32	0.29
10		0.64	0.61	0.59	0.59	0.57	0.46	0.58	0.30
11		0.71	0.70	0.74	0.74	0.78	0.72	0.73	0.62
12		0.63	0.63	0.64	0.61	0.59	0.49	0.61	0.36
13		0.64	0.64	0.64	0.63	0.61	0.53	0.59	0.38
14		0.75	0.72	0.70	0.64	0.62	0.55	0.54	0.43
κ		0.34	0.31	0.31	0.44	0.44	0.85	0.44	0.19

Table H.5.: K-Nearest Neighbor (KNN) results for various cutoffs and fingerprint radii. Here, radius refers to the radius parameter of the Morgan algorithm.

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
01	0	0.33	0.33	0.33	0.33	0.33	0.33	0.25	0.17
02		0.45	0.36	0.36	0.36	0.36	0.27	0.27	0.18
03		0.56	0.22	0.11	0.11	0.11	0.11	0.11	0.00
04		0.42	0.38	0.38	0.38	0.42	0.33	0.33	0.25
05		0.40	0.20	0.20	0.20	0.20	0.20	0.20	0.10
06		0.55	0.43	0.29	0.29	0.27	0.27	0.27	0.18
07		0.42	0.14	0.14	0.14	0.25	0.17	0.17	0.08
08		0.56	0.56	0.56	0.56	0.56	0.44	0.33	0.22
09		0.42	0.14	0.14	0.14	0.25	0.17	0.17	0.08
10		0.45	0.36	0.36	0.36	0.36	0.27	0.27	0.18
11		0.33	0.33	0.33	0.33	0.33	0.22	0.22	0.22
12		0.45	0.36	0.36	0.36	0.36	0.27	0.27	0.18
13		0.56	0.56	0.56	0.56	0.56	0.44	0.33	0.22
14		0.71	0.57	0.43	0.43	0.43	0.36	0.36	0.29
κ		0.26	0.53	0.36	0.36	0.36	0.00	0.00	0.00
01	1	0.33	0.33	0.40	0.27	0.20	0.20	0.20	0.13
02		0.45	0.36	0.40	0.36	0.36	0.27	0.27	0.18
03		0.33	0.11	0.00	0.00	0.00	0.00	0.00	0.00
04		0.40	0.40	0.40	0.40	0.40	0.30	0.30	0.20
05		0.25	0.08	0.00	0.08	0.08	0.08	0.08	0.08
06		0.50	0.30	0.20	0.30	0.30	0.30	0.30	0.20
07		0.40	0.20	0.00	0.20	0.20	0.10	0.10	0.10
08		0.40	0.40	0.50	0.40	0.40	0.30	0.30	0.20
09		0.40	0.20	0.00	0.20	0.20	0.10	0.10	0.10
10		0.45	0.36	0.50	0.36	0.36	0.27	0.27	0.18
11		0.40	0.40	0.40	0.40	0.40	0.30	0.30	0.20
12		0.45	0.36	0.50	0.36	0.36	0.27	0.27	0.18
13		0.33	0.33	0.50	0.33	0.33	0.25	0.25	0.17
14		0.55	0.45	0.80	0.36	0.36	0.27	0.27	0.27
κ		0.19	0.00	0.19	0.00	0.00	0.00	0.00	0.00
01	2	0.30	0.20	0.20	0.20	0.20	0.20	0.20	0.10
02		0.33	0.14	0.14	0.14	0.33	0.33	0.33	0.22
03		0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.00
04		0.50	0.40	0.40	0.40	0.40	0.30	0.30	0.20
05		0.22	0.00	0.00	0.00	0.11	0.11	0.11	0.11
06		0.42	0.20	0.20	0.20	0.25	0.25	0.25	0.17
07		0.33	0.00	0.00	0.00	0.17	0.08	0.08	0.08

continued on next page

Table H.5.: *continued*

Compound	Radius	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
08		0.36	0.36	0.36	0.36	0.36	0.27	0.27	0.18
09		0.40	0.00	0.00	0.00	0.20	0.10	0.10	0.10
10		0.38	0.43	0.43	0.43	0.31	0.23	0.23	0.15
11		0.30	0.40	0.40	0.40	0.30	0.30	0.30	0.20
12		0.30	0.33	0.33	0.33	0.30	0.30	0.30	0.20
13		0.33	0.29	0.29	0.29	0.33	0.25	0.25	0.17
14		0.38	0.50	0.50	0.50	0.38	0.31	0.31	0.23
κ		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
01	3	0.40	0.56	0.44	0.40	0.33	0.33	0.33	0.22
02		0.20	0.40	0.40	0.20	0.40	0.30	0.30	0.30
03		0.40	0.36	0.29	0.40	0.27	0.27	0.27	0.18
04		0.56	0.56	0.44	0.44	0.44	0.44	0.44	0.33
05		0.00	0.10	0.00	0.00	0.10	0.10	0.10	0.10
06		0.29	0.36	0.29	0.29	0.27	0.27	0.27	0.18
07		0.17	0.22	0.11	0.17	0.11	0.11	0.11	0.11
08		0.40	0.33	0.33	0.40	0.33	0.22	0.22	0.22
09		0.40	0.22	0.22	0.20	0.22	0.11	0.11	0.11
10		0.40	0.33	0.43	0.40	0.33	0.33	0.33	0.22
11		0.50	0.40	0.38	0.50	0.40	0.40	0.40	0.30
12		0.40	0.42	0.38	0.40	0.25	0.25	0.25	0.17
13		0.43	0.44	0.29	0.29	0.33	0.33	0.33	0.22
14		0.50	0.56	0.44	0.50	0.44	0.33	0.33	0.22
κ		0.19	0.53	0.00	0.00	0.00	0.00	0.00	0.00

H.2 MS validation of test set analytes

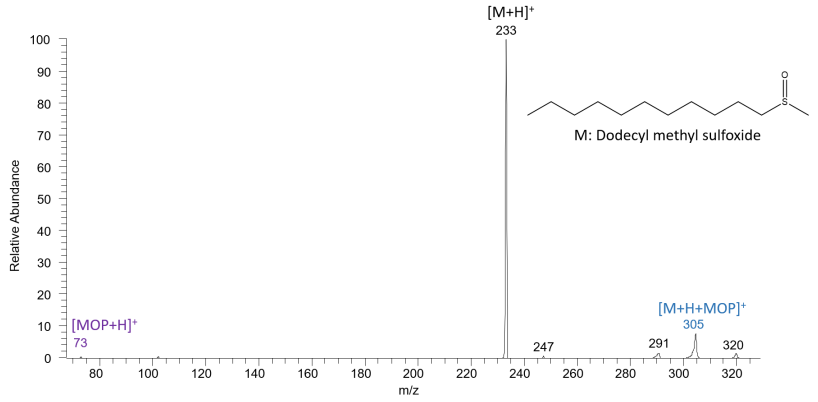


Figure H.1. MS/MS spectrum measured after 3,000 ms reaction of protonated dodecyl methyl sulfoxide with MOP, indicating the formation of a diagnostic addition product (M+H+MOP). Credit: Judy Liu

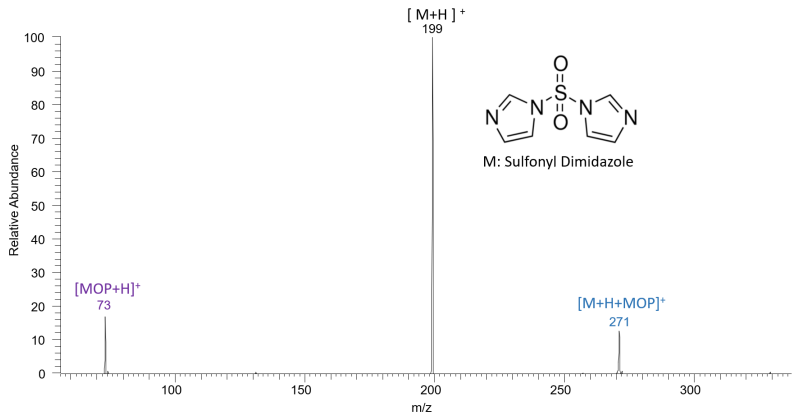


Figure H.2. MS/MS spectrum measured after 3,000 ms reaction of protonated sulfonyl dimidazole with MOP, indicating the formation of a diagnostic addition product. Credit: Judy Liu

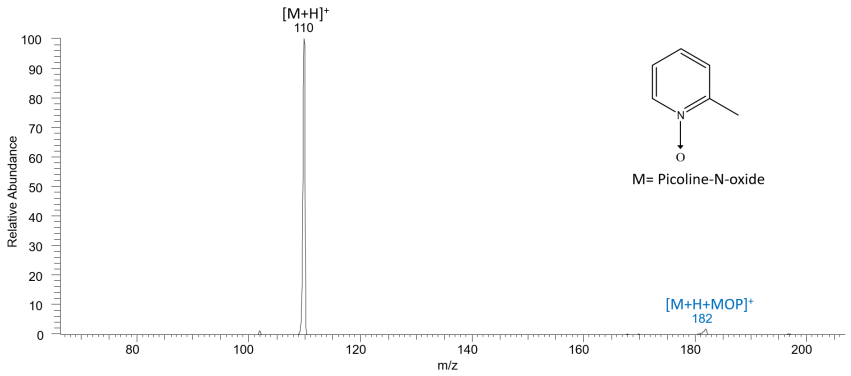


Figure H.3. MS/MS spectrum measured after 10,000 ms reaction of protonated picoline N-oxide with MOP, indicating the formation of a diagnostic addition product. No proton transfer product was observed. Credit: Judy Liu

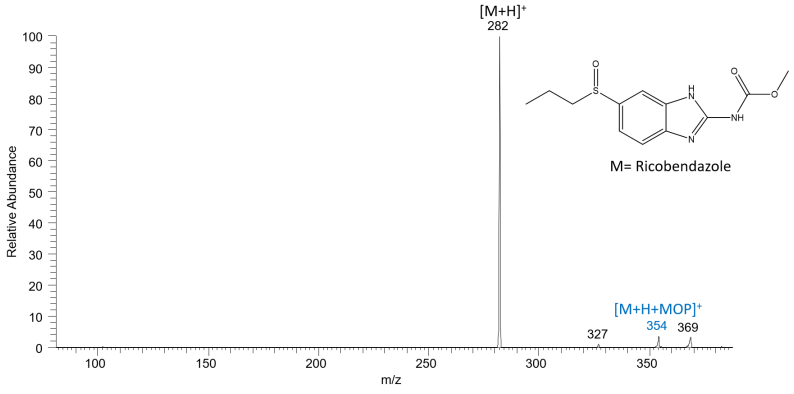


Figure H.4. MS/MS spectrum measured after 10,000 ms reaction of protonated ricobendazole with MOP, indicating the formation of a diagnostic addition product. No proton transfer product was observed. Credit: Judy Liu

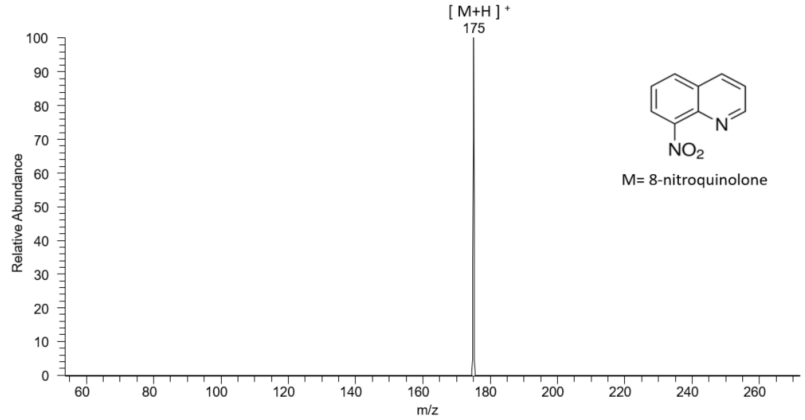


Figure H.5. MS/MS spectrum measured after 10,000 ms reaction of protonated 8-nitroquinolone with MOP, indicating that no diagnostic addition product was formed. No proton transfer product was formed, either. Credit: Judy Liu

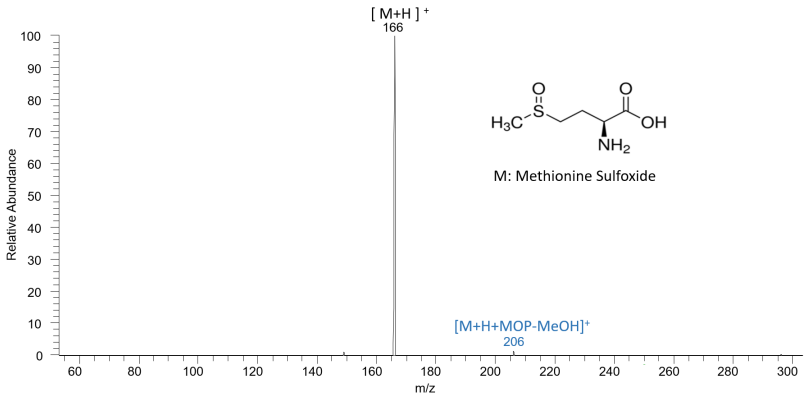


Figure H.6. MS/MS spectrum measured after 10,000 ms reaction of protonated methionine sulfoxide with MOP, indicating the formation of a diagnostic addition product. No proton transfer product was formed. Credit: Judy Liu

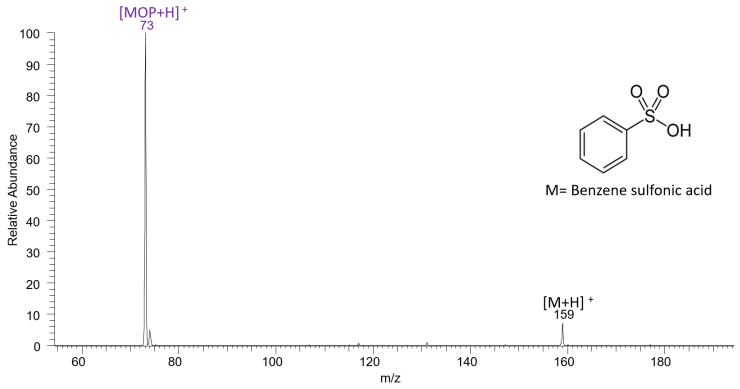


Figure H.7. MS/MS spectrum measured after 3,000 ms reaction of protonated benzene sulfonic acid with MOP, indicating that a diagnostic addition product was not formed. Instead, a proton transfer product was observed (MOP+H). Credit: Judy Liu

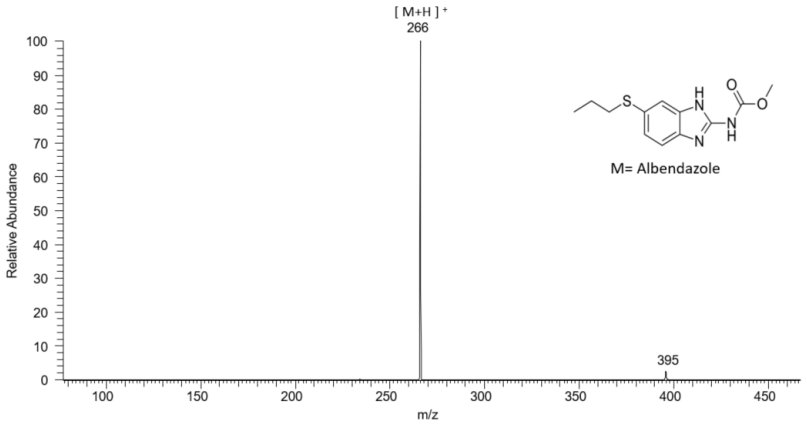


Figure H.8. MS/MS spectrum measured after 10,000 ms reaction of protonated albendazole with MOP, indicating that a diagnostic addition product was not formed. No proton transfer product was observed, either. Credit: Judy Liu

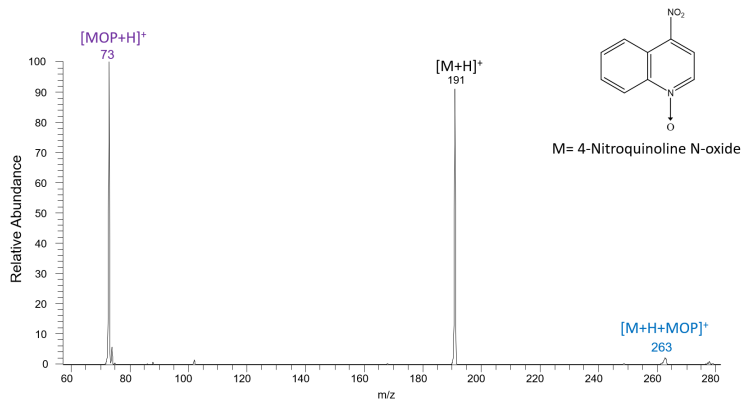


Figure H.9. MS/MS spectrum measured after 3,000 ms reaction of protonated 4–nitroquinoline N-oxide with MOP. Although evidence of a diagnostic addition product is seen, the presence of a major proton transfer product indicates that this reaction is not suitable for diagnostic applications. Credit: Judy Liu

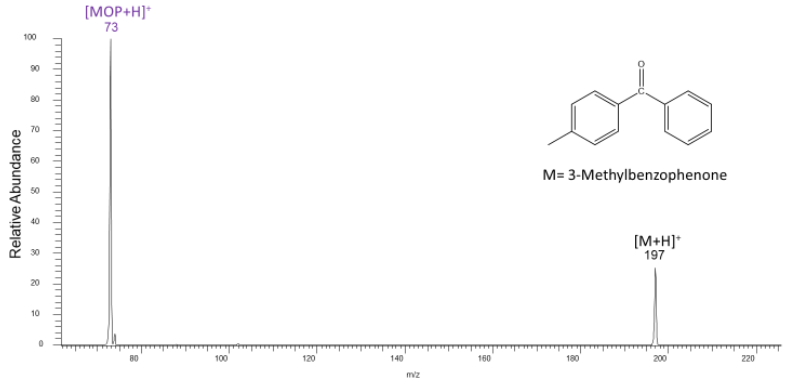


Figure H.10. MS/MS spectrum measured after 3,000 ms reaction of protonated 3–methylbenzophenone with MOP, indicating that a diagnostic addition product was not formed. Instead, a proton transfer product was observed. Credit: Judy Liu

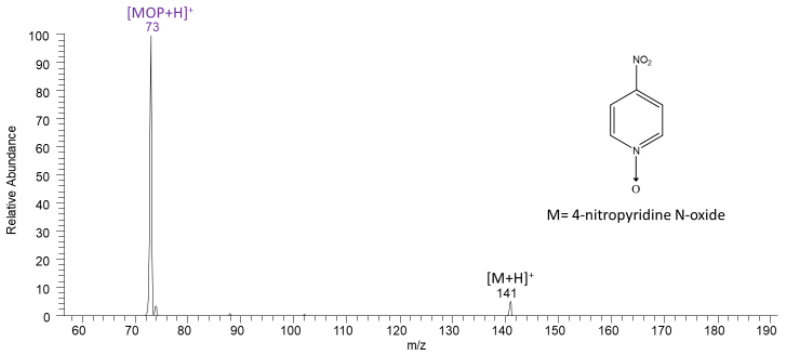


Figure H.11. MS/MS spectrum measured after 3,000 ms reaction of protonated 4–nitropyridine N–oxide with MOP, indicating that a diagnostic addition product was not formed. Instead, a proton transfer product was observed. Credit: Judy Liu

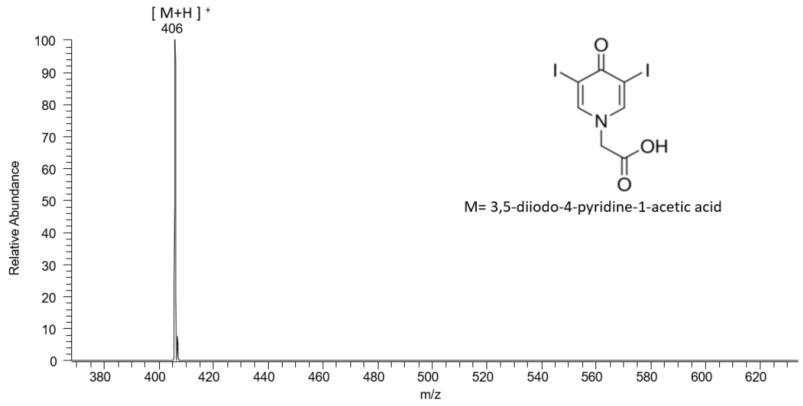


Figure H.12. MS/MS spectrum measured after 10,000 ms reaction of protonated 3,5-diiodo-4-pyridine-1-acetic acid with MOP, indicating that a diagnostic addition product was not formed. No proton transfer product was observed, either. Credit: Judy Liu

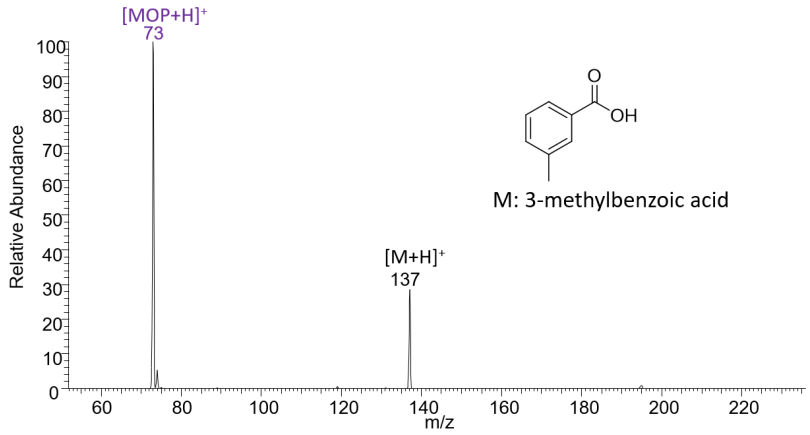


Figure H.13. MS/MS spectrum measured after 30 ms reaction of protonated 3-methylbenzoic acid with MOP, indicating that a diagnostic addition product was not formed. Instead, a proton transfer product was observed. Credit: Judy Liu

VITA

Education

Doctor of Philosophy in Chemistry — June 2015 – May 2020
Purdue University — 4.00 GPA

Bachelors of Science in Chemistry — August 2011 – August 2014
Rensselaer Polytechnic Institute — 3.97 GPA

Honors / Awards

- Eagle Scout — Awarded 12/2010
- John and Mary Cloke Prize for Undergraduate Research in Chemistry Awarded 05/2015
- Purdue Graduate Student Government Travel Award — Awarded 12/2017
- Computational Interdisciplinary Graduate Program Research Spotlight Awarded 03/2018
- Lynn Fellowship for computational life sciences — Awarded 04/2015
- Purdue Center for Cancer Research Fellowship — Awarded 12/2018
- Merck Rising Stars in Analytical Chemistry and Materials Science Awarded 10/2019

Research Experience

Chopra Lab
Graduate Student — October 2015 – Present

- Developed machine learning methodologies to identify functional groups using Infrared and Mass spectra
- Spearheaded the creation of a reagent reaction prediction scheme using cheminformatics and entropic machine learning algorithms
- Mentored a graduate student in the creation of a Multi-Layered Perceptron (MLP) for the prediction of reaction solvent conditions
- Collaborated with experimental biologists, synthetic, and analytical chemists to develop therapeutics targeting the immune system
- Utilized a combination of machine learning and molecular docking to identify targets in Castration Resistant Prostate Cancer
- Lead developer of the Chemical Algorithms for Network based Decisions on Interactions for modeling reactivitY (CANDIY) Suite
- Executed proteome wide docking using CANDOCK to identify protein targets in Myeloid Derived Suppressor Cells
- Oversaw the development of a knowledge-based force field package for running CUDA accelerated molecular dynamics simulations

Slipchenko Research Group

Undergraduate Researcher June 2015 – September 2015

- Determined parameters for fitting exchange repulsion Effective Fragment Potential calculations using the GAMESS software package

Rensselaer Exploratory Center for Cheminformatics Research Undergraduate Student | August 2013 – May 2015

- Investigated the protein-protein binding interface of mutant insulin variants using property encoded shape distributions with Novo-Nordisk

- Optimized a methodology for the calculation of dielectric properties for silica polymers using density functional theory

First author Publications

Fine J, Rasjashekar A, Jethava K, Chopra G. Accurate and automated *de novo* identification of molecular functional groups using keep learning architectures. *Chemical Communications*, **Submitted**.

Fine J, Liu J, Beck A, Alzarieni K, Ma X, Boulos V, Kenttämaa H, Chopra G. Graph Based Machine Learning Interprets Diagnostic Isomer-Selective Ion-Molecule Reactions in Tandem Mass Spectrometry. *ChemRxiv*, **2019**.

Fine J, Chopra G, Childress M, Nolte D, Lanman N. Combining Biodynamic Imaging and RNA-sequencing yields an improved machine-learning model for predicting lymphoma resistance to chemotherapy. *In preparation*, **2019**.

Fine J, Chopra G. Lemon: a framework for rapidly mining structural information from the Protein Data Bank. *Bioinformatics*, **2019**.

Fine J, Lackner R, Chopra G, Samudrala R. Computation Chemoproteomics to Understand the Role of Selected Psychoactives in Treating Mental Health Indications. *Scientific Reports*, **2019**.

Fine J, Konc J, Samudrala R, Chopra G. CANDOCK: Chemical atomic network based hierarchical flexible docking algorithm using generalized statistical potentials. *Journal of Chemical Information and Modeling*, **2020**.

Fine J, Majumder J, Prakash P, Lantz T, Chopra G. Protein-target identification of molecular functional groups using deep learning architectures. *In preparation.*

Co-first author Publications

Zhang W, **Fine J**, Sculley C, McGraw J, Chopra G. MINT: A virtual reality platform to visualize, manipulate, and explore molecular structures. *ChemRxiv*, **2019**.

Beck A, **Fine J**, Jethava K, Chopra G. Machine learning to improve the selection of reaction solvent in organic chemistry. *In preparation.*

Jethava K, **Fine J**, Chen Y, Chopra G. Heterogeneous reactivity prediction of N–sulfonylimine electrophilicity for fast multi-component synthesis of 1,3,4-Oxadiazole explained Using machine learning. *In preparation.*

Wijewardhane P, Jethava K, **Fine J**, Chopra G. Combined Molecular Graph Neural Network and Structural Docking Selects Potent Programmable Cell Death Protein 1/Programmable Death-Ligand 1 (PD-1/PD-L1) Small Molecule Inhibitors. *Journal of Medicinal Chemistry*, **Submitted**.

Co-author Publications

Rojas C, **Fine J**, Slipchenko L. Exchange-repulsion energy in QM/EFP. *The Journal of Chemical Physics*, **2018**.

Ma X, Zhou J, Wang C, Carter-Cooper B, Yang F, Larocque L, **Fine J**, Tsuji G, Chopra G, Lapidus R, Sintim H. Identification of new FLT3 inhibitors that potently inhibit AML cell lines, via an azo click-it/staple-it approach. *ACS Medicinal Chemistry Letters.*, **2017**.

Hernandez-Perez M, Chopra G, **Fine J**, Anderson RM, Benjamin C, Nadler J, Holman TR, Maloney D, Tersey S, Mirmira R. Inhibition of 12/15-lipoxygenase protects against β cell oxidative stress and glycemic deterioration in mouse models of type I diabetes. *Diabetes*, **2017**.

Published Abstracts

Fine J, Chopra G. CANDOCK: Conformational Entropy Driven Analytics for Class-Specific Proteome-Wide Docking. *Biophysical Society Abstracts*, **2018**.

Majumder, J, Lantz TC, **Fine J**, Chopra G. Drug repurposing for castration resistant prostate cancer based on disease-disease relationships. *AACR Cancer Research*, **2017**.

Stewart B, **Fine J**, Chopra G. Parallelization of Molecular Docking algorithms using CUDA for use in Drug Discovery. *The Summer Undergraduate Research Fellowship (SURF) Symposium*, **2017**.

Collaborations

Hilkka Kenttämaa, Created a graph-based decision tree classifier for the determining whether an analyte will form an adduct with a neutral reagent.

Herman Sintim, Evaluated the potency of novel kinase inhibitors for the treatment of acute myeloid leukemia using an in-house docking software.

Ram Samudrala, Investigated the potential therapeutic value of psychoactive compounds for the treatment of mental health indications using the Computational Analytics for Novel Drug Opportunities (CANDO) platform.

Ragu Mirmira, Calculated the toxicity of lipooxygenase inhibitors using proteome wide docking software.

Skills

Computational Chemistry QSAR • Quantum Chemistry • DFT • Docking • Molecular Dynamics • PyMOL • MOE • High Performance Computing

Programming Languages C • C++ • Fortran • R • Python • Julia

Machine Learning Platforms Keras • PyTorch • Caret • TensorFlow

Architectures SVM • Random Forest • Decision Tree • MLP • Long Short Term Memory

PRESENTATIONS

Selected Talks

- CSESC, Society for Industrial and Applied Mathematics 04/14/2017
- The Hitchhiker's guide to the Biomolecular Galaxy 05/11/2017

- Biophysical Society 02/18/2018
- American Chemical Society 04/01/2019
- Molecule Sciences Software Institute workshop on Machine Learning in Chemistry 11/17/2019
- Merck Rising Stars in Analytical Chemistry and Materials Science Symposia 11/22/2019

Workshops Instructed

- CIGP Workshop on Molecular Dynamics
- Docking in Medicinal Chemistry

Mentoring

Brandon Stewart, *Undergraduate Student*, Developed a GPU implementation of the statistical forcefield used for scoring in the CANDOCK software

Jean-Michael Diei, *Undergraduate Student*, Implemented a neuronal fingerprint machine learning architecture for the prediction of reaction sites in small molecules

Anand Rasjashekar, *Undergraduate Student*, Produced an MLP neural network to identify functional groups in Infrared Spectroscopy and a Recurrent Neural Network to calculate the molecular mass of a compound in Mass Spectroscopy

Armen Beck, *Graduate Student*, Improved a machine learning network that predicts solvent reaction conditions given the topology of a reaction

Prageeth Rajitha, *Graduate Student*, Initiated a new machine learning architecture that combines graph and docking features of molecule to predict its binding towards PDL-1

Connor Beveridge, *Graduate Student*, Mentored a new graduate student by helping him create a variational autoencoder to identify fingerprint features in tandem mass spectra

Dawood Mohideen, *Graduate Student*, Oversaw the creation of a Generative Adversarial Network for the creation of new drug-like molecules

Leadership

- $\Phi\Sigma K$ **President**

  Created a $50,000 micro-grant program for the Troy Mount Ida community

- **CHM 125 Course Supervisor**

  Oversaw other CHM 125 Teaching Assistants, planned laboratory experiments and exams

- **Graduate Student Advisory Board Representative**

  Represented the interests of physical chemistry graduate students

- **Mental Health Committee Chair**

  Arranged events and support groups to help students understand mental health issues amongst graduate students