

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Developing Artificial Intelligence Tools for Biologists

Permalink

<https://escholarship.org/uc/item/2768w8wt>

Author

Shub, Laura

Publication Date

2024

Peer reviewed|Thesis/dissertation

Developing Artificial Intelligence Tools for Biologists


by
Laura Shub

DISSERTATION
Submitted in partial satisfaction of the requirements for degree of
DOCTOR OF PHILOSOPHY

in
Biological and Medical Informatics


in the
GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:

DocuSigned by:

4DF1BD06D670465... Michael Keiser
Chair

DocuSigned by:

brian shoichet

DocuSigned by:

582727E949C7441... William DeGrado

Committee Members

Copyright 2024

by

Laura Shub

For Mom, Dad, Allie, and Neko.

ACKNOWLEDGEMENTS

As cliché as it may be, we do indeed stand on the shoulders of giants, and exist in the context of all that came before us. This work would not have been possible without the support of so many incredible people at UCSF and beyond.

First and foremost, I would like to thank my advisor, Michael Keiser. Mike is an incredible resource of computational expertise, interesting philosophical discussions about the ethics of artificial intelligence, and knowledge of Russian and sci-fi literature. I was not initially planning on doing anything with machine learning, but his enthusiasm and mentorship convinced me to try it out. Mike has provided invaluable guidance over these past few years and I know he'll go on to do incredible work in the next stage of his career.

Second, I'd like to thank the professors at UCSF and elsewhere for their support. Thank you to the members of my thesis committee: Brian Shoichet and Bill DeGrado. Brian was a source of stability at UCSF, starting from my very first rotation through graduation and beyond. Bill provided incredible insight into how my work could interact with that of biologists and chemists to make a greater impact. Thank you to Tony Capra for being on my qualifying exam committee and good-naturedly letting me distract his lab members as I borrowed a desk in his space. Thank you to Georgios Skiniotis for his useful project discussions. Additionally, special thanks to Michael J. Robertson, who somehow managed to mentor me through a structural biology project while landing a faculty job. Hope you're enjoying Texas!

From the moment I joined Mike's lab, I benefited from the guidance provided by other lab members. Luca Ponzoni was a great rotation mentor, and through him, I learned about chemical fingerprints, variational autoencoders, hyperparameter optimization, and more that would be pivotal to my doctoral studies. Jessica McKinley was a role model for how I could succeed in

higher education and provided useful advice throughout her time in the lab. Alexandre Fassio developed the LUNA software, which was integral to many of my projects. Thank you to the all members of the former Keiser group, postdocs, students, RDAs, system administrators, and administrative assistants alike, whose time in the lab overlapped with my own, however briefly: Kangway Chuang, Mahdi Ghorbani, Elena Caceres, Garrett Gaskins, Douglas Myers-Turnbull, Dan Wong, Laura Gunsalus, Will Connell, Wren Saylor, Parker Grosjean, Ben Orr, Brendan Hall, Duncan Muir, Umair Khan, Victor Rabesquine, Halimat Afolabi, Chimno Nnadi, Mikio Tada, Varun Sharma, Rish Sharma, Noah Weber, Sina Ghandian, Liane Albargouthi, Lise Minaud, Taline Mardirossian, Nick Mew, John Gallias, Sahru Keiser, and Fedelle Austria. Thank you to Selina Liu for doing all the grunt work I assigned her, and good luck in graduate school! To my original iPQB cohort, Adamo Mancino, Aiden Winters, Connor Galvin, Erin Gilbertson, Hasan Alkhairo, Hersh Bhargava, Matthew Hancock, Matthew Smith, Muziyue Wang, Scott Nanda, Tianna Grant, Wilson Vasquez, and Zach Cutts: We'll all make it.

Thank you to my parents, Lisa Horvath and Ben Shub, for their unending support and for encouraging my decision to move myself and my cat all the way to San Francisco. I know the last few years have been tumultuous, but my love for both of you will never change. Thank you especially to my sister Allie, who was always a source of levity when I was rethinking my choices in life and needed to get out of my own head. And, of course, thank you to my extended families on both sides, who were always willing to entertain me talking about my projects whether they cared or not: Grandma and Grandpa, Grand Alice, Aunt Sara, Aunt Stephanie, Aunt Sam, Uncle Doug and Aunt Jennifer, Will and Hannah, Olivia and Woodrow, Celeste, Dominic, Maddie, Genna, Hannah, Hailey, and many, many more.

Thank you to the Deep Apple family for my first real foray into working in biotech: Elissa Fink, who was a great rotation mentor and introduced me to virtual screening. Stefan Gahbauer, for this great feedback and suggestions during our machine learning meetings. Rishikesh Magar, for dealing with my questionable code. Jazon Zbieg, for letting me play around in AI generative space with little oversight. I'm looking forward to joining y'all soon.

I'd like to thank the friends that have reminded me that a world exists beyond the lab: Erin Gilbertson (again) and Mark Richard for trivia nights, wine tastings, baseball games, country concerts, and showing me how great Minnesota and its people are; Luke Masters, for being the ultimate concert buddy and just as obsessed with Joanna Newsom as I am; Gabrielle Rabadam, for adopting an introvert and somehow convincing me to go backpacking multiple times; and Justin Salonia, for late-night gaming and walking me home. From my days at the University of Texas, thank you to Rachel Rapagnani for hard carrying me through organic chemistry lab all those years ago, and Justin Kang for late nights commiserating about computer science projects, along with the full Dean's Scholars family. I wouldn't have made it this far without all of you.

Thank you to Joanna Newsom for being the soundtrack of my graduate career.

And finally, thank you to my cat, Neko, for keeping me relatively sane. You deserve all the cat treats in the world.

CONTRIBUTIONS

This dissertation was supervised by Dr. Michael Keiser. Additional guidance was provided by Dr. Brian Shoichet and Dr. Michael J. Robertson.

Chapter 2 contains material from a manuscript currently under review and available in open-access format:

Shub L, Liu W, Skiniotis G, Keiser MJ, Robertson MJ. Metric Ion Classification (MIC): A Deep Learning Tool for Assigning Ions and Waters in Cryo-EM and X-Ray Crystallography Structures. bioRxiv March 19, 2024, <https://doi.org/10.1101/2024.03.18.585639>.

Chapter 3 contains material from a manuscript in preparation:

Shub L, Lin F-Y, Muir D, Mathiowetz A, Keiser MJ. Autoparty: Machine Learning-guided Visual Inspection of Molecular Docking Results.

We seek our name, we seek our fame, and our credentials,

Paned in glass, trained to master incidentals.

- *Joanna Newsom, Leaving the City*

Developing Artificial Intelligence Tools for Biologists

Laura Shub

ABSTRACT

With the growth of biological and chemical datasets and the development of novel computational techniques, applications of artificial intelligence (AI) and machine learning (ML) methods that leverage these datasets to assist experimentalists become more critical than ever. This dissertation presents an overview of commonly used AI/ML tools for molecular biology and introduces two novel tools, as well as details their specific use cases. In Chapter 1, I provide a review of traditional techniques and their machine learning counterparts for ligand- and structure-based drug discovery and protein structure elucidation and design. In Chapter 2, I introduce Metric Ion Classification (MIC), a method for determining the identity of experimentally identified waters and ions in biomolecular structures. MIC builds upon recent advancements in protein-ligand interface representations and metric learning techniques to introduce a novel classification scheme with extensive validation on a variety of experimental structures. In Chapter 3, we present Autoparty, a tool for AI-assisted human-in-the-loop molecule annotation designed to facilitate the manual assessment of virtual screening results. Autoparty uses the principles of active learning to direct chemists toward useful compounds and limit the amount of labor required when evaluating compounds. These applications do not attempt to replace existing techniques; rather, they act in service of scientists to accelerate both structure determination and drug discovery pipelines. This work broadly highlights the utility of these tools and others like them and encourages their adoption alongside classical approaches.

TABLE OF CONTENTS

Chapter 1: Artificial Intelligence in Structural Biology: From Small Molecules to Proteins 1

Introduction.....	1
Artificial Intelligence vs. Machine Learning vs. Deep Learning: What’s the difference?	2
Artificial Intelligence for Small Molecules	4
<i>Chemical representation</i>	4
<i>Molecular property prediction</i>	7
<i>Computer-assisted drug discovery</i>	8
<i>Machine learning in drug discovery</i>	10
<i>Addressing expanding library sizes</i>	12
<i>De novo molecule generation</i>	14
Artificial Intelligence for Proteins	15
<i>Protein structure determination</i>	15
<i>Machine learning in protein structure elucidation</i>	16
<i>Protein structure prediction</i>	18
<i>De novo protein design</i>	20
Machine Learning-related Controversy: What Could Go Wrong?.....	21
Conclusion	22
Chapter 2: Deep learning assigns identities to ions and waters in cryo-EM and x-ray crystallography structures	23

Abstract.....	23
Introduction.....	23
Results.....	26
<i>Architecture and Performance of MIC</i>	26
<i>Manual Inspection of Disagreeing Sites</i>	30
<i>Validation of MIC on Structures Derived from Cryo-EM Maps</i>	32
<i>RNA/Ribosomal structure evaluation</i>	35
<i>Extended set model training, performance, and manual review</i>	36
<i>Comparison with existing methods</i>	37
Discussion.....	40
Methods.....	41
<i>Dataset Curation</i>	41
<i>Density Fingerprint Representation</i>	42
<i>Initial atomic features and included interactions</i>	43
<i>Shell Number and Radius</i>	44
<i>Training and test datasets of curated densities for MIC</i>	44
<i>Model Training</i>	45
<i>Feature Attribution</i>	45
<i>Undowser Comparison</i>	46

<i>Statistical Analysis</i>	46
Data and Code Availability.....	46
Acknowledgments.....	47
Author Contributions	47
Figures.....	48
Supplemental Figures.....	55
Chapter 3: Machine Learning-guided Visual Inspection of Molecular Docking Results with Autoparty	63
Abstract.....	63
Introduction.....	63
Results.....	66
<i>Autoparty: A tool for automated human-in-the-loop molecule inspection</i>	66
<i>Interaction Calculation and Representation Generation</i>	67
<i>Model Architecture and Training</i>	68
Throwing a Hit-picking Party	71
<i>Uploading a screen</i>	71
<i>Getting the party started</i>	71
<i>Annotating Molecules</i>	71
<i>Uploading existing annotations (optional)</i>	72
<i>Training a Model</i>	72

Autoparty Implementation	72
Acknowledgments.....	73
Author Contributions	73
Figures.....	74
Supplemental Methods.....	77
<i>Uncertainty Quantification Analysis with AA2AR</i>	77
<i>Ordinal Classification Testing with Dopamine D4</i>	80
Supplemental Figures.....	82
Chapter 4: Final and Future Thoughts.....	89
References.....	91

LIST OF FIGURES

Figure 2.1. Overview of MIC workflow.....	48
Figure 2.2. MIC learned embeddings, performance, and validation.	49
Figure 2.3. MIC predictions on Cryo-EM structures of MC4R and apoferritin.	50
Figure 2.4. Predictions on RNA/Ribosomal structures.....	51
Figure 2.5. MIC Extended ion set performance and manual review.	52
Figure 2.6. MIC, CheckMyMetal, and Undowser performance examples.....	53
Supplemental Figure 2.1. MIC dataset preparation and exploration.	55
Supplemental Figure 2.2. MIC prevalent-ion set additional results.	56
Supplemental Figure 2.3. Manual review of test set discrepant sites.	57
Supplemental Figure 2.4. MIC extended set additional results.	58
Supplemental Figure 2.5. Manual review of discrepant sites from extended test set.....	59
Figure 3.1. Conceptual Overview of the Virtual Screening Pipeline and Active Learning.....	74
Figure 3.2. Schematic showing Autoparty workflow and user interaction.....	75
Figure 3.3. Autoparty Web Interface	76
Supplemental Figure 3.1. Investigating Uncertainty Quantification with AA2AR.....	82
Supplemental Figure 3.2. Ordinal Label Training with D4 MM/GBSA-based Grades	83
Supplemental Figure 3.3. Schema for Autoparty SQL database.	84

LIST OF TABLES

Table 2.1. Summary of RNA/Ribosome structure performance.....	54
Supplemental Table 2.1. Initial features for each atom by type and fingerprint type	60
Supplemental Table 2.2. Hyperparameters explored and final values.....	61
Supplemental Table 2.3. P-values and number of sites for all statistical analyses.	62
Supplemental Table 3.1. Autoparty hit-picking settings, options, and default values.....	85
Supplemental Table 3.2. Uncertainty quantification settings, options, and default values.	86
Supplemental Table 3.3. LUNA available interactions and associated colors for visualization. .	87
Supplemental Table 3.4. Calibration test architectures, hyperparameters, and metrics.	88

LIST OF ABBREVIATIONS

AF	AlphaFold
AI	Artificial Intelligence
AL	Active Learning
CASP	Critical Assessment of Protein Structure Prediction
CNN	Convolutional Neural Network
Cryo-EM	Cryogenic Electron Microscopy
DL	Deep Learning
ECFP	Extended-Connectivity Fingerprint
EGNN	Equivariant Graph Neural Network
FF	Feed-Forward
GNN	Graph Neural Network
LLM	Large Language Model
MIC	Metric Ion Classification
ML	Machine Learning
MPNN	Message-Passing Neural Network
MSA	Multiple Sequence Alignment
NN	Neural Network
PDB	Protein Data Bank
QSAR	Quantitative Structure-Activity Relationship
QSPR	Quantitative Structure-Property Relationship
RL	Reinforcement Learning
RNN	Recurrent Neural Network

RF RoseTTAFold

SF Scoring Function

SMILES Simplified Molecular Input Line Entry System

SVC Support Vector Classifier

CHAPTER 1: ARTIFICIAL INTELLIGENCE IN STRUCTURAL BIOLOGY: FROM SMALL MOLECULES TO PROTEINS

Introduction

Artificial intelligence (AI) as a field has existed since as early as the 1950s, and has continued to see development despite periodic “winters” of depressed activity and funding.¹ Growing interest in the applications of AI to biology specifically has corresponded with expanding data sets, prompting the development of techniques to match.² AI, and more specifically machine learning (ML), has the capacity to vastly improve upon current methods if properly employed. However, these approaches require communication between computational researchers, chemists, and biologists to design these models, from the initial training sets to the final implementations and beyond. This ensures that AI/ML tools do not just serve as interesting technical demonstrations but provide actionable predictions that can be validated experimentally.

In this dissertation, we seek to demonstrate how machine learning, and specifically deep neural networks, can be applied to various structural biology problems to help accelerate existing pipelines and assist scientists in a variety of ways. In this chapter, we motivate this work by summarizing the current state of AI/ML for two major areas of molecular biology: small molecules and proteins. In addition, we provide an overview of common cheminformatics and machine learning techniques. We describe major advances in the application of ML to these questions and the current state-of-the-art technologies, as well as the potential areas for further improvement in these fields.

Artificial Intelligence vs. Machine Learning vs. Deep Learning: What's the difference?

Machine learning refers to a broad array of techniques united by the fundamental concept of self-learning rules from large databases of examples.^{3,4} ML is a subset of a broader class of algorithms known as artificial intelligence (AI) concerned with the science and engineering of machines with similar decision-making capabilities to humans.⁵⁻⁷ However, the two are often conflated in media and popular culture, both due to sensationalism and a lack of field-specific expertise, adding to confusion regarding the terms in the broader public consciousness.⁸⁻¹⁰

Both AI and its subset ML have wide utility in real-world problems. Examples of AI that are not ML are still common and include more traditional computer programs that follow predefined protocols given input data, such as pathing algorithms like those used by Google Maps.¹¹ The fields of robotics and self-driving cars similarly have continued to use non-ML AI to direct movement and decision-making due to a need for pre-programmed responses in specific situations (i.e., stopping if a pedestrian is detected in front of the car).¹² In the clinic, AI has long been used in the form of medical diagnosis expert systems that assist physicians with difficult tasks, including bacterial infection identification¹³⁻¹⁶ and determining post-operative care.¹⁷ In each of these cases, the algorithms follow *a priori* rule systems programmed by a human operator, with no updates to these rules over time, classifying them as artificial intelligence but not machine learning.

By contrast, machine learning refers specifically to algorithms that can adapt their own performance without following specific instructions. Generally stated, a supervised ML model consists of three major components: 1) a decision process, in which the algorithm produces an output in response to input, often a class label; 2) an error function that evaluates the quality of this output by comparison with the known label or desired output; and 3) an update or optimization

process, in which the decision process is modified to minimize the calculated error. ML applications have increased dramatically in recent years in virtually every field, from facial recognition,^{18,19} spam detection,^{20,21} content recommendation,²²⁻²⁴ and machine translation²⁵⁻²⁸ to more controversial uses such as customer service^{29,30} and advertising.³¹⁻³⁶ In each of these cases, the ML model leverages large bodies of existing data (faces, customer behavior, emails) to learn ideal behavior. ML problems typically fall into either supervised, where labels are known and the model attempts to learn how to classify new data, or unsupervised, where the goal is to learn patterns in the absence of existing labels. Examples of techniques for supervised ML include decision trees and random forests,³⁷ gradient boosting variants such as AdaBoost and XGBoost,³⁸ support vector machines/classifiers,³⁹ and Bayesian classifiers.⁴⁰

Many applications of ML discussed above are examples of a specific ML paradigm known as deep learning (DL). Deep learning approaches utilize multi-layer neural networks (NNs) to learn to make highly accurate predictions from increasingly large bodies of data. The classic example of a DL paradigm is the feedforward (FF) neural network, first published in 1965.⁴¹ FF-NNs operate on vector inputs through matrix multiplication followed by a non-linear activation function, such as rectified linear unit (ReLU) or Swish.⁴² The values of the matrices, otherwise known as the model weights, are updated to minimize model error (loss) by stochastic gradient descent with backpropagation.⁴³

Since the introduction of standard FF-NNs, there have been numerous advances to apply this same logic to alternative input data structures. Convolutional neural networks (CNNs) operate on multidimensional inputs such as 2D images, 3D voxels, and videos, capturing spatial relationships between pixels in the input data.⁴⁴ Recurrent neural networks (RNNs) were developed to work on sequential data of variable lengths such as text input.⁴⁵ Graph neural

networks (GNNs)⁴⁶ operate on graph structures to output either graph-level or node-level predictions. Most recently, the concept of transformers based on the multi-head attention mechanism⁴⁷ has been incorporated into a wide variety of architectures, enabling the development of large language models (LLMs)⁴⁸ that achieve unprecedented success in text prediction. Each of these architectures has been adapted to address outstanding problems in biology and chemistry, which we will cover in the following sections.

Artificial Intelligence for Small Molecules

Chemical representation

One of the most critical questions when designing a model architecture to operate on small molecules is the choice of representation, typically either string- vector, or graph-based. The most ubiquitous string representation of molecular structure is the Simplified Molecular Input Line Entry System (SMILES), first proposed by Weininger in 1998⁴⁹ and expanded to canonical SMILES in 1989.⁵⁰ SMILES are constructed by “breaking” all cycles in a molecule followed by a full traversal, appending a character at each node of the tree representing the corresponding atom. This results in a one-to-many relationship between a molecule and its SMILES strings depending on the specific path traveled, a potential problem that is addressed in canonical SMILES by imposing an order to this traversal based on initial atomic features. Since their introduction, SMILES have seen widespread use for storing databases of molecules, for example recording known actives for protein targets⁵¹ or in virtual screening libraries.^{52,53} SMILES and similar string-based representations (InChi,⁵⁴ SELFIES⁵⁵) provide an opportunity for utilizing RNNs that operate on sequences, typically used in the field of natural language processing. There are a variety of these architectures, such as long short-term memory and gated recurrent units, with the unifying logic that these approaches maintain a hidden state calculated from the prior sequence.⁵⁶ This

hidden state, together with the current element, determine the next prediction. RNNs can operate on inputs of any length to learn complex syntaxes, such as those that govern SMILES construction and validity. More recently, transformers have emerged as the state-of-the-art approach for text processing. Both of these architectures have seen use in molecular generation and property prediction, discussed further in the following sections.

While line inputs such as SMILES have seen widespread use, they remain unideal for certain ML approaches. They cannot be used with standard FF architectures that require fixed-length vector inputs. Furthermore, distance calculations between molecules using SMILES is difficult, as similar molecules can encode to very different sequences based on decisions made during construction by molecular traversal. This problem persists despite the invention of canonical smiles. The original canonicalization algorithm was proprietary, and many cheminformatics packages including RDKit⁵⁷ and OpenEye⁵⁸ implement their own variant of this canonicalization protocol. Handling chirality also remains difficult, with many databases including Zinc 2D opting to remove any chiral information from the stored representation. Rather, for standard ligand-based approaches, an ideal representation would be invariant to traversal order and capture increasingly large substructures of the molecule. Early methods to create these vector molecular representations, referred to as fingerprints, include Molecular ACCess keys fingerprints (MACCS)⁵⁹ and PubChem⁶⁰ fingerprints. The former is a vector of 166 selected input features, while the latter encodes 881 potential elements. Both these formulations required expert feature engineering to design, thus introducing the potential for the exclusion of relevant properties in the final representation. This was the problem addressed by extended-connectivity fingerprints (ECFPs) for molecules, introduced by Rogers and Hahn in 2010.⁶¹ ECFPs use the Daylight atomic invariants and circular expansion based on the Morgan algorithm⁶² along with a fixed hash function

that encodes individual atoms or neighborhoods of atoms into a fixed-length vector of 1s and 0s. The 1 at a given position thus indicates the presence of a specific substructure that is consistent across fingerprints constructed by the same algorithm. ECFPs are used primarily for rapid similarity measurements between two molecules and are often employed in virtual screening to find nearby molecules with a high chance of displaying similar biological activities against a given target,^{63,64} but they also provide an ideal input for typical ML architectures and have been used as the input representation in a variety of molecular property prediction tasks. This concept has been expanded to include three-dimensional information about molecule structure.⁶⁵ Similarly, some efforts have been made to extend the concept of molecular fingerprints to protein-small molecule interfaces,⁶⁶⁻⁶⁸ capturing the contacting geometries and non-covalent intermolecular interactions contributing to binding.

The most recent advancement in molecular representation is the development of learned molecular fingerprints, proposed by Duvenaud *et al.*⁶⁹ Rather than being generated by a fixed predetermined algorithm, these task-specific representations perform graph convolutions on molecular topology before typically feeding into a standard linear predictor. This process improves on standard fingerprints in a few critical ways. ECFPs weigh all input features equally through the formulation as a string of bits, while this may not be appropriate for the given task. Graph-based fingerprints allow the model to determine the most salient input features, resulting in improved accuracy and improving model interpretability. These learned representations also circumvent the known problem of bit-collisions in fingerprints, wherein multiple features encode to the same bit, complicating property prediction.⁷⁰ This approach was formalized into the current message-passing neural network (MPNN) terminology and proven to be the state-of-the-art for molecular property prediction in Gilmer *et al.*⁷¹ discussed further in the following section.

Molecular property prediction

As mentioned previously, one of the most common applications for ML in chemistry is that of molecular property prediction. The obvious example is the use of ML in Quantitative Structure-Property Relationship (QSPR) and Quantitative Structure-Activity Relationship (QSAR) studies that seek to quantify the interaction between input molecular features and downstream effects.^{72,73}

In their initial iterations, QSPR/QSAR analyses were performed by fitting a linear correlation between the relevant value and 2D- or 3D-molecular features engineered by medicinal chemists.⁷⁴⁻

⁷⁷ Early work in using neural networks and molecular fingerprints for this process showed improved results over traditional QSAR, representing an important early step in ML ligand-based drug discovery.^{78,79} Recently, it's been demonstrated that learned GNN-based representations typically outperform traditional fingerprinting techniques for property prediction tasks, leading to widespread interest and adoption of this architecture.^{80,81} In just the past few years, GNNs have been used to predict a variety of molecular properties including polar surface area,^{82,83} bioavailability,⁸⁴ octanol solubility,⁸⁵ aqueous solubility,⁸⁵⁻⁹⁰ blood-brain barrier permeability,⁸⁹⁻⁹⁴ hydrophobicity,^{82,83,95,96} toxicity,^{85,90,93,97-103}, synthetic accessibility,⁸² and cost,¹⁰⁴ to name a few.

One advantage of GNNs/MPNNs over traditional fingerprinting methods is their ability to operate directly on molecular structure, and in many cases, incorporate geometric information. Early iterations of MPNNs operated on graph structure alone, analogous to the generation of 2D fingerprints that do not consider chirality. Variants of architectures that place greater emphasis on node connectivity include directed message passing NNs as implemented in Heid *et al.*¹⁰⁵ and edge-memory networks as presented by Withnall *et al.*⁸⁹ One of the earliest architectures to include explicit positional information was the Equivariant Graph Neural Network (EGNN),¹⁰⁶ designed specifically to address the need for rotation and translation invariance in 3D graphs by

incorporating the relative distance between nodes into the message passing steps. EGNNs have been used for toxicity prediction,¹⁰³ molecule conformer minimization,¹⁰⁷ and in various other applications including ML-based docking software^{108,109} and target-aware *de novo* molecule generation,^{110,111} both discussed in greater detail below.

The area where GNNs for property prediction have seen the greatest success is quantum chemistry. With the release of datasets such as QM9,¹¹² MD17,^{113,114} GEOM,¹¹⁵ and QMugs¹¹⁶ that record the result of very computationally expensive quantum chemistry calculations, many models have been trained to predict the approximate energies of molecular conformational states. SchNet introduced continuous convolutional filters based on radial basis functions to reproduce interatomic forces and predict total energy of molecules.¹¹⁷ DimeNet¹¹⁸ and DimeNet++¹¹⁹ build upon this advancement to incorporate spherical Bessel functions using both radial and angle information during message-passing convolutions for more accurate predictions. SphereNet¹²⁰ similarly provides angle information through a spherical coordinate system used during message passing. Architectures like GEM¹²¹ and AliGNN¹²² incorporate geometric information by running message passing on two graphs simultaneously, one representing atoms and bonds and the other representing bonds and the angles between them. These increasingly feature-aware architectures enable further predictive power and more accurate models.

Computer-assisted drug discovery

There is a commonly cited figure that the total cost of developing a novel pharmaceutical exceeds two billion US dollars.^{123,124} Virtual screening, the process by which libraries of compounds are pre-screened against a given receptor, has been used since the 1990s as an upstream step to discover lead molecules with novel chemotypes, resulting in many successful drug

candidates.^{125,126} Virtual screening methods differ in their exact implementations and intended use cases. Each docking program must solve two major problems: 1) how to *sample* molecule conformations within a protein pocket and 2) how to rank, or *score*, the generated pose. UCSF DOCK employs a graph-matching algorithm to place rigid fragments based on precalculated molecule conformers into a static protein structure.^{127,128} A similar approach is taken by the rigid docking protocols in Schrodinger's Glide¹²⁹ and FlexX.¹³⁰ Alternatively, some implementations that allow ligand flexibility arrive at a final pose through optimization techniques such as Monte Carlo Minimization or genetic algorithms.¹³¹ This includes approaches such as AutoDock and AutoDock Vina¹³² as well as GOLD,¹³³ ICM,¹³⁴ and flexible protocols in FlexX. Finally, some programs consider receptor flexibility as well. RosettaLigand, part of the larger Rosetta protein-modeling suite, uses a two-staged sequence with flexible small-molecule and protein bond angles to identify the predicted pose of a given ligand, a much more time-intensive process.^{135,136} Recent modifications to the AutoDock program have similarly allowed for limited receptor flexibility but only in the case where the user has significant available computational resources.¹³⁷

Like sampling algorithms, there are multiple categories of scoring functions (SFs) utilized to rank ligands that vary in implementation, accuracy, and computational expense. Historically, these have been divided into force field-based, empirical, and knowledge-based methods.^{138,139} Recent developments in the field and confusion over the boundaries of these terms have resulted in reconsideration of these categories, leading some scientists to propose an updated categorization scheme: physics-based, regression-based, potential-based, and machine learning-based.¹⁴⁰ Physics-based (or force-field) SFs evaluate a ligand using fundamental molecular mechanical calculations such as van der Waals and electrostatics, with many iterations incorporating a term for desolvation.^{141,142} DOCK and GOLD both use this approach, with the latter incorporating an

additional explicit term for hydrogen bonding. Empirical (regression-based) SFs compute a linear combination of individual terms to fit experimentally measured binding affinities.^{143,144} Despite assuming similar forms, this differs from physics-based approaches as the individual contributions of each term are calculated using methods such as multivariate linear regression or partial least squares, leading to some overlap between empirical and ML scores. ChemScore¹⁴⁵ (implemented in GOLD), X-score,¹⁴⁶ and GlideScore-XP¹⁴⁷ SFs all fall under this approach. Knowledge (potential)-based SFs calculate the sum of the distance-dependent pairwise statistical potential between atoms within a given interaction radius.¹⁴⁴ The exact parameters of these potential functions are determined by analyzing large protein-ligand structure databases, primarily the Protein Data Bank.^{148,149} Finally, as the name suggests, approaches belonging to the newly-introduced category of ML-based SFs use machine learning to predict scores and binding affinities for protein-ligand interfaces, discussed in more detail below. The boundaries between these categories are not exact; the Rosetta scoring function is a weighted combination of both physics-based and knowledge-based terms,¹⁵⁰ and many empirical methods could arguably be a type of machine learning scoring function.

Machine learning in drug discovery

ML SFs have emerged in recent years as an alternative to these more classical approaches. Traditional SFs like those above require extensive domain knowledge to develop and evaluate. Furthermore, they often omit critical terms such as entropy and require many simplifying assumptions to be tractable, especially regarding quantum mechanical calculations.¹⁵¹ There has been significant interest in the creation of a generic ML SF that is able to take as input any protein-small molecule complex and predict an accurate binding affinity. Such a score would massively

accelerate the drug-discovery process but remains elusive; however, that does not mean there has not been significant progress made towards that goal. One of the earliest iterations of this was RFscore, which uses a random forest regressor on the frequency of pairs of heavy atoms at the protein-ligand interface.¹⁵² Das *et al.* used support vector machines on shape-based representations to predict binding affinity from property-encoded shape distributions, achieving R^2 values in the 0.5-0.7 range.¹⁵³ NNScore¹⁵⁴ and its successor NNScore 2.0¹⁵⁵ represent some of the earliest efforts to apply deep learning to selected interface features for distinguishing between potent and weak binders or predicting binding affinity, respectively. These methods showed improved performance over classical scoring functions in some cases but still suffered from scarce training data, simple architectures, failure to generalize to new systems, and overall concerns about the validity of the learned logic of the model.¹⁵⁶ As mentioned above, protein-ligand IFPs attempt to encode the full range or potential protein interactions into fixed-length vector representations, allowing for similarity analysis and ML model training.^{66-68,157} Alternately, Kdeep¹⁵⁸ utilizes 3D-CNNs centered on the binding pocket as input. GNINA,¹⁵⁹ one of the first attempts to bridge traditional docking methods with ML scoring, uses exhaustive sampling of ligand orientations along with Monte-Carlo minimization of a 3D-CNN-based SF to dock molecules. OnionNet¹⁶⁰ uses a 3D-CNN that covers both local and longer-form interactions for binding affinity prediction, while OnionNet-2¹⁶¹ uses a 2D-CNN on atom pairs in distance shells, an extension to the original RFscore feature set. MPNNs have been applied to the tasks of scoring as well, using graphs to represent either the protein and ligand separately or the bound complex structure. Many of these methods achieve high accuracy on the PDBbind¹⁶² and CASF¹⁶³ test sets for binding affinity prediction. These include Proximity Graph Networks,¹⁶⁴ SIGN,¹⁶⁵ PLANET,¹⁶⁶ SS-GNN,¹⁶⁷ GIaNT,¹⁶⁸ and graphLambda,¹⁶⁹ among others.

Historically, ML applications in virtual screening have focused on scoring. In the past few years, ML-based sampling and protein-ligand complex pose generation methods have emerged as well, corresponding with advancements in generative architectures. Among the most notable of these are EquiBind¹⁰⁸ and EquiDock.¹⁰⁹ Both approaches use graph-matching neural networks to directly predict the bound structures of an input protein-ligand or protein-protein complex, respectively. TANKbind¹⁷⁰ uses angle-aware message passing to predict both structure and binding affinity. DiffDock¹⁷¹ uses diffusion models that learn translational, rotational, and torsional transformations to arrive at a predicted protein-ligand pose, achieving greater accuracy than traditional GLIDE docking in some cases. These methods still struggle to recapitulate realistic binding modes and can often produce strained or invalid geometries, but they provide significant speedup over exhaustive searches. In time, they may come to consistently outperform traditional methods.

Addressing expanding library sizes

Advancements in combinatorial chemistry have resulted in a significant expansion in the size of virtual libraries available for screening. Zinc22 has grown to over 30 billion compounds,⁵² while eMolecule's eXplore boasts having over 4.9 trillion individual molecules in its database.¹⁷² While this opens up the possibility of screening for drugs with novel chemotypes and still only represents a small fraction of the total drug-like chemical space ($>10^{63}$ compounds!¹⁷³), it introduces significant technical challenges. Current virtual screening methods can evaluate billions of compounds, but the looming explosion in potential chemical matter necessitates methodologies that better identify important regions of chemical space to explore for a given target. Some traditional similarity-based library traversal strategies have been proposed to address this problem

and have shown very promising results in retrospective screens.¹⁷⁴ Still, ML represents an appealing potential avenue due to its accuracy and high throughput. Active learning (AL) has already been applied to virtual screens to successfully recover top compounds from screening a fraction of the full compound library.^{175,176} This same concept opens the possibility for rapid prediction of more computationally intense scores such as free energy perturbation (FEP).¹⁷⁷ In Chapter 3 of this work, we discuss another potential application for AL in the context of human-in-the-loop training during hit-picking, another bottleneck in the overall drug discovery process.

Alternatively, *implicit* molecular libraries have seen significant development in recent years. Under this paradigm, rather than explicitly enumerating millions to billions (and beyond) of compounds, a model is trained that can generate novel, valid molecules property-matched to those in the training set. This model is queried when new molecules are needed, such as in preparation for a docking screen. Early iterations of these implicit libraries used SMILES with recurrent neural networks^{178,179} or autoencoders^{180,181} to generate valid chemical structures in similar topological and property space to the training sets. Graph-based methods were developed to address the problems inherent in this approach, specifically the potential disconnect between string representation and molecular structure.¹⁸² These generators offer some advantages over traditional chemical library approaches, removing the need to combinatorically build molecules and requiring fewer computational resources to store. However, they introduce some potential pitfalls, such as the possibility of invalid structures, no enforced synthesizability constraints, and a lack of guaranteed enumeration of all possible molecules.

De novo molecule generation

Building on this idea of implicit libraries opens the potential for target-specific library generation. Approaches following this paradigm take a generic molecule generator and employ reinforcement learning (RL) with a target-specific reward function to bias the model towards producing well-scoring compounds by this external metric. Zhavoronkov *et al.* used this technique along with a novel generative tensorial reinforcement-learning algorithm to discover DDR1 kinase inhibitors in only 21 days.¹⁸³ One benefit of this technique is that it is applicable to multiple architectures, including transformers: Mazuz *et al.*¹⁸⁴ and Noutahi *et al.*¹⁸⁵ both used RL with transformers to generate molecules with desired properties (predicted binding affinity for Beta-secretase 1 and central nervous system penetrance, respectively). These techniques have historically been used to directly produce compounds for experimental testing but could be used in combination with standard docking programs to generate full libraries of high-quality candidate molecules to use in virtual screening.

These 2-stage approaches still rely on classical virtual screening and ligand-based SFs to tune the reward function used in RL to modulate the properties of generated molecules. Furthermore, they typically do not produce a three-dimensional complex structure, a critical step for evaluating compounds *in silico*. This has encouraged the development of pocket-based *de novo* drug design, in which models are able to directly build a new small molecule into the selected protein. This concept of “pocket-to-lead” design was used by Kojima *et al.*¹⁸⁶ in designing HIV-1 protease inhibitors by manually optimizing weak hits from a fragment screen to better match the pharmacophoric properties of the pocket. Since then, many DL approaches have been developed that condition molecular generation on 3D ligand or pocket structure. Skalic *et al.* created LigDream¹⁸⁷ and LiGANN¹⁸⁸ which use a combination of CNNs and RNNs to output compounds

matching a desired 3D pharmacophore from the ligand and the pocket, respectively. Other approaches include the structure-conditioned RNN from Xu *et al.*,¹⁸⁹ Monte Carlo tree search with a graph-based molecule generator in DeepLigBinder,¹⁹⁰ encoder-decoders trained on interface properties as in RELATION,¹⁹¹ and many more.^{192–196}

Diffusion models have gained significant attention recently for their ability to generate strikingly realistic images.¹⁹⁷ These models are trained to reconstruct the original input following perturbation, a process that has been applied to both design linkers for existing fragments¹⁹⁸ and generate 3D molecules in full.¹⁹⁹ These approaches can similarly be conditioned on an existing protein pocket for fragment-based¹¹⁰ or whole-molecule^{111,200,201} generation from the binding site alone, an idea that has the potential to revolutionize the standard virtual screening pipeline. Similar to ML-based sampling, these approaches can suffer from unrealistic and strained geometries, but new variants will almost be developed that address these problems by incorporating explicit hybridization states and other chemically-motivated features.

Artificial Intelligence for Proteins

Protein structure determination

It is commonly accepted in the field of protein biology that structure determines function²⁰² but it is up to structural biologists to determine structure. There are three primary methods used for protein structure elucidation. The most common by far is x-ray crystallography, used to obtain macromolecular structures by constructing electron density maps from the x-ray diffraction patterns of crystals.²⁰³ Crystallizing a protein is much easier said than done and is by far the rate-limiting step of this protocol, with many proteins proving unable to crystallize at all.²⁰⁴ Nuclear magnetic resonance (NMR) spectroscopy exists as an alternative method of structure determination.²⁰⁵ Protein NMR can determine secondary structures by measuring the difference in

chemical shift between folded and individual residues, and coupling between nuclei can provide information about peptide dihedral angles. These data are then used with molecular dynamics simulations to arrive at the protein's folded structure.²⁰⁶ Cryogenic electron microscopy (cryo-EM) has gained traction as an alternative to x-ray crystallography and NMR.²⁰⁷ In cryo-EM, rather than requiring crystallization, structures are frozen down to approximately -180°C. Multiple 2D images are collected in various orientations and combined to create a three-dimensional view of the molecule. Historically, cryo-EM has been the lower resolution method,²⁰⁸ though this is changing rapidly. Within the past few years, cryo-EM was used to solve the structure of apoferritin at sub-2Å resolution, allowing for the visualization of even ordered waters.^{209,210} Cryo-EM is becoming increasingly popular; in 2015, the ratio of new x-ray crystallography structures to cryo-EM structures deposited in the PDB was 40:1 (8,578:216), while in 2024 that number has reduced to 1.7:1 (2,924:1,743) structures (so far).²¹¹ These advances in experimental methodologies have encouraged the development of computational techniques to similarly accelerate the process of biomolecular structure elucidation, discussed in the following section.

Machine learning in protein structure elucidation

There are a variety of ML methods that have been used to assist with the modeling of biomolecular structures from x-ray crystallography and cryo-EM maps.^{212,213} Many applications of ML to x-ray crystallography have focused on automating image analysis to determine the presence or absence of crystals, a typically laborious process that requires expert annotation. Bruno *et al*²¹⁴ developed MARCO, a CNN-based method for classifying biomolecular images into clear, precipitate, crystal, and other categories. DeepFreak²¹⁵ was similarly developed to classify x-ray diffraction images into five potential categories based on successful crystallization, denoting the quality of the

diffraction pattern. Other similar methods include the AlexNet-based model presented by Ke *et al.*²¹⁶ and CrystalNet.²¹⁷ Further downstream, QAEmap²¹⁸ used 3D-CNNs to estimate the local quality of protein modeling in low-resolution electron density maps, building on the statistically informed methods of real-space correlation coefficient²¹⁹ and MolProbity score.²²⁰

By comparison, cryo-EM has seen wide interest in ML for modeling. Early methods in ML for cryo-EM include CryoSPARC²²¹ which uses a stochastic gradient descent algorithm for structure determination and 3D classification, and Emap2sec(+)^{222,223} to predict nucleic acids and protein secondary structures from low resolution (5-10Å) cryo-EM maps. For higher resolution maps, Si *et al.*²²⁴ applied a cascaded-CNN approach to predict protein backbone structure that outperformed more classical techniques like Rosetta and MAINMAST.²²⁵ Terashi *et al.*²²⁶ introduced the deep-learning-based amino-acid-wise model quality (DAQ) score that uses 3D-CNNs to estimate the quality of protein modeling from cryo-EM maps at the secondary structure, amino acid, and alpha-carbon levels. DeepMainmast²²⁷ built upon the original minimum-spanning-tree-based MAINMAST for main-chain protein modeling, incorporating AlphaFold (discussed below) predictions for improved accuracy. Most notably, ModelAngelo,²²⁸ a graph-neural network approach to predict the refined structure from Cryo-EM maps, has demonstrated performance on par with human experts for proteins and nucleic acids. Additionally, it demonstrated high accuracy in determining the amino acid of unknown sequences. These methods can work with classical techniques to facilitate their use and remove some of the subjectivity of human analysis (or even the chance for fraud²²⁹).

Protein structure prediction

Any discussion of machine learning in protein structure prediction must begin with AlphaFold (AF)²³⁰ and, more specifically, its successor AlphaFold2 (AF2).²³¹ Both AlphaFold variants use multiple sequence alignment (MSA) of natural proteins with known structures to determine evolutionarily correlated residues, inferring putative contacts from which 3-dimensional structure is calculated. AlphaFold generated significant interest when it won the 13th Critical Assessment of Protein Structure Prediction (CASP) competition by a wide margin.²³² AlphaFold2 utilizes an SE(3)-invariant transformer architecture (“evoformer”) with self-attention that allows it to capture longer-range interactions, leading to significantly improved performance: AF2 won the following CASP14 by an even wider margin (cumulative z-score of 244.0 by human assessment, compared to the next highest score of 90.8).²³³ While these approaches can use templates for an approach closer to homology modeling, removing these templates for *ab initio* prediction (free modeling) often results in a sharp decrease in accuracy.²³⁴ Still, both approaches and their demonstrated success revolutionized the way that we think about protein structure prediction and led to a flood of similar methods and widespread application. Nature Methods named protein structure prediction as its 2021 method of the year²³⁵.

Shortly after CASP13, Yang *et al.* released trRosetta²³⁶ using similar approaches as AF to generate contacts from MSA, though it incorporated relative residue orientation information along with a constrained relaxation step to ensure more realistic geometries. RoseTTAFold (RF),²³⁷ the Rosetta community’s response to AF2, uses a three-track approach that simultaneously operates on sequences, distances, and coordinates to learn relationships between these structural properties. OpenFold²³⁸ was created as an open-source, trainable, less memory-intensive variant of AlphaFold2 to improve accessibility to the larger scientific community. Efforts have been made

to predict structure from sequence alone, without the need for MSAs. Facebook's research team produced ESMfold,²³⁹ a language-based model that operates on input amino acid sequences. Other similar approaches include RGN2²⁴⁰ and EMBER2,²⁴¹ all of which require no MSA information.

The impacts of AF and similar techniques have been far-reaching.^{242,243} DeepMind and the European Bioinformatics Institute released the predicted structures of over 200M sequences, covering a wide range of proteins across species.²⁴⁴ AF predictions have been used in combination with x-ray crystallography²⁴⁵ and cryo-EM²⁴⁶⁻²⁴⁸ to solve protein structures. Databases of binding pockets from both solved and AF protein structures have been created for use in drug repurposing and to gain a greater understanding of protein structure-function relationships.^{249,250} AF predictions have seen success in drug discovery, performing comparably to known experimental structures in GLIDE in some studies²⁵¹ (though it should be noted that this is not always the case²⁵²). AF2-predicted structures of the serotonin receptor 5-HT2A were used in a successful virtual screening campaign, demonstrating that AF2 can potentially identify druggable structures not discovered through traditional structural biology.²⁵³ Similarly, AF2 was used to predict the structures of the unresolved hERG closed and inactivated states, providing an explanation for known anti-target activity.²⁵⁴ These uses suggest that despite known issues in AF (need for MSA information, lack of protein dynamics, lack of ability to predict proteins in complex with small molecules, difficulties with disordered regions), these predictions can be an incredibly useful resource for biologists as they approach these problems. During the writing of this thesis, AlphaFold3²⁵⁵ was released (controversially without an associated codebase²⁵⁶). AF3 uses a diffusion-based architecture to allow for the prediction of proteins, nucleic acids, small molecules, and ions with a single model. AF3 displays high accuracy in a variety of tasks from protein-ligand complex

structure prediction to antibody-antigen prediction, and there is little doubt that it will see widespread use across related fields, though its true impact on biology remains to be seen.

De novo protein design

Another area of interest for ML application is *de novo* protein design, a longstanding goal of synthetic biology.²⁵⁷⁻²⁶⁰ Similar to the number of potential drug-like molecules, the space of all possible protein sequences is vast. For any 200-residue protein, there are 20^{200} potential sequences considering natural amino acids alone. This expands further if one includes unnatural amino acids, i.e. for potential therapeutic development.²⁶¹ Deep-learning protein structure prediction technologies have been applied to protein design by predicting structures of random amino acid sequences, then modifying the sequence to optimize the divergence between these starting structures and known naturally occurring geometries to result in novel folds, a process known as hallucination.²⁶² Direct sequence design from a provided backbone was presented in ProteinMPNN,²⁶³ which uses the message-passing architecture on an input graph of protein backbone atoms to predict the amino acid identity of each residue. ProteinMPNN designs from AF-generated backbones showed high success rates for solubility and stability, exceeding that of designs produced by hallucination with AF. Multiple approaches have used modifications of RoseTTAFold for protein design. Wang *et al.*²⁶⁴ used a finetuned RF approach to optimize sequences that contain a specific functional site. RFdiffusion²⁶⁵ modifies the RF structure prediction module to instead denoise 3D coordinates and predict new potential backbone orientations. Additionally, RFdiffusion allows for guided conformational generation, such as designing sequences to bind other proteins or small molecules. The most recent iterations of both RF and RFdiffusion include all-atom representations to allow for the modeling of more complex

assemblies that contain nucleic acids, small molecules, and metals.²⁶⁶ Future iterations of these methods could begin to address outstanding problems, including prediction of conformational dynamics and the design of engineerable functions.

Machine Learning-related Controversy: What Could Go Wrong?

The introduction of ML into the pharmacology and structural biology realms has not been without its detractors, and for good reason. There are numerous examples of AI/ML models recapitulating problematic trends in training datasets across fields, including hiring,²⁶⁷ healthcare,^{36,268} facial recognition,^{269,270} and financial services.²⁷¹ ChatGPT, the revolutionary LLM released by OpenAI, has seen widespread adoption^{272,273} in spite of the known tendency of the chatbot to reference spurious publications or court rulings.²⁷⁴ Similarly, Google's AI chatbot Bard (now Gemini) has been used to assist scientific research despite an infamous factual error made during its debut.^{275,276} These types of faulty outputs are relatively simple to fact-check, but models used in fields requiring extensive domain knowledge may have hidden artifacts that are yet to be uncovered. In the field of drug discovery specifically, many ligand-based classifiers have used DUD-E²⁷⁷ as a training set to distinguish between actives and decoys despite known hidden bias making it particularly unsuitable for this task.²⁷⁸ Even the widely used PDBbind dataset for binding affinity prediction has been subject to additional scrutiny due to concerns over data leakage and limited generalization.^{279,280} In protein structure prediction, analysis of the AF2 performance for different protein groups has revealed significant discrepancies in the success of the model structure across amino acid types, secondary structures, and overall protein size.²⁸¹ While AF predictions have been successfully used in drug discovery, they cannot yet replace experimental structure determination.^{282,283} On the other hand, some of these methods are subject to potential misuse by bad actors, a problem highlighted by the successful rapid generation of novel biochemical

weapons.²⁸⁴ Despite these concerns over its application, AI is likely here to stay. In the future, to ensure proper use of these techniques, AI/ML models and the datasets they are trained on need to be thoroughly interrogated. Furthermore, experimental validation of all ML predictions remains critical.

Conclusion

We live in the midst of the golden age of machine learning in biology. New applications for AI/ML are developed near daily, demonstrating successes across a variety of fields, including the aforementioned uses in drug discovery, de novo protein design, and structure prediction. These applications will only grow more accurate with the development of new databases and computational architectures to better leverage them. I believe that they will have a profound impact on how we approach biological questions as they arise, whether that be drugging new targets or understanding novel mechanisms of action. This highlights the need for dialogue between biologists, chemists, and computer scientists to better identify problems to address and design the corresponding solutions. Additionally, it is critical that the resulting tools are available to biologists and chemists working on these projects. Many useful computational tools require command-line familiarity to install and use and thus go underutilized among the people who would otherwise be the primary beneficiaries. Growing lists of software dependencies require more and more sophisticated package managers and environments to support them, which can quickly become prohibitively difficult to work with for many. The integration of techniques like AlphaFold and ESMfold into widely-used tools like ChimeraX²⁸⁵ represents a step in the right direction, as does ColabFold²⁸⁶ for browser-based use of AlphaFold. However, many tools remain trapped behind the technical know-how required to run them. As developers, we must endeavor not to lock our colleagues out of this brave new world.

CHAPTER 2: DEEP LEARNING ASSIGNS IDENTITIES TO IONS AND WATERS IN CRYO-EM AND X-RAY CRYSTALLOGRAPHY STRUCTURES

Abstract

At sufficiently high resolution, x-ray crystallography and cryogenic electron microscopy are capable of resolving small spherical map features corresponding to either water or ions. Correct classification of these sites provides crucial insight for understanding structure and function as well as guiding downstream design tasks, including structure-based drug discovery and de novo biomolecule design. However, direct identification of these sites from experimental data can prove extremely challenging, and existing empirical approaches leveraging the local environment can only characterize limited ion types. We present a novel representation of chemical environments using interaction fingerprints and develop a machine-learning model to predict the identity of input water and ion sites. We validate the method, named Metric Ion Classification (MIC), on a wide variety of biomolecular examples to demonstrate its utility, identifying many probable mismodeled ions deposited in the PDB. Finally, we collect all steps of this approach into an easy-to-use open-source package that can integrate with existing structure determination pipelines.

Introduction

Hydration and ion binding are vital for biomolecular function, contributing to structure,^{287,288} ligand binding,²⁸⁹⁻²⁹¹ enzymatic catalysis,²⁹² and dynamics.^{287,293} Proper rationalization of their effects in structures requires accurate identification of these sites as either water or a specific ion bound. However, pinpointing the identity of spherical features in experimental maps can be challenging, as the experimental data may be of insufficient quality or interpretability to definitively classify the scatterer. In structures derived from x-ray crystallography, individual ions and water can often

be distinguished by examining the Fo-Fc difference map or OMIT map.²⁹⁴ However, this depends on data quality and can be difficult if the scattering is similar, for example when differentiating between water, sodium, and magnesium. Cryogenic electron microscopy data is even more problematic due to intrinsic challenges in generating meaningful difference maps.²⁹⁵ While scattering differences between atoms of different charges in certain resolution ranges can be used to discriminate atomic charges from cryo-EM data in theory,²⁹⁶ this often proves difficult in practice.

Rather than directly determining the identity of the spherical feature from the experimental data, one can also consider the environment around the feature responsible for its coordination. Cations generally have a coordination shell of several partial or formal negatively charged atoms with well-defined geometry and coordination distances.²⁹⁷ Water has an ideal tetrahedral coordination with two hydrogen bond donors and two hydrogen bond acceptors. Anions tend to have a less well-defined coordination shell than cations but with positive interaction partners, specifically the guanidinium of arginine.²⁹⁸ While computational tools exist for classification based on the local environment, these methods tend to be focused on more specific subsets, for example, comparing different metal ions^{299–302} or validating waters.³⁰³

In recent years, machine learning (ML) has been successfully applied to various biomolecular modeling tasks including structure prediction,^{231,237} protein design,²⁶⁵ molecular docking,^{108,171} and molecular dynamics simulations.³⁰⁴ The field of cryo-EM specifically has seen increased interest in ML for modeling proteins^{222,224,227,228} and nucleic acids,³⁰⁵ as well as improving and evaluating map and model quality.^{226,306,307} ML methods have been explored for identifying potential binding sites for specific ions or ion subclasses,^{308–310} but assigning the identity of experimentally determined sites remains underexplored. One complicating factor is the relative

scarcity of high-quality experimental structures with the full set of possible ions bound compared to what is needed for the 3-dimensional convolutional architectures typically used in these applications, highlighting the need to consider both alternate site representations and model architectures for this task.

In cheminformatics, molecular fingerprints are vector representations that encode chemical structure.^{61,70} They have been used in mapping chemical space,³¹¹ virtual drug screening,³¹² and as input to quantitative structure-activity relationship models.³¹³ In recent years, this concept has been expanded to interface fingerprints that capture the geometry of ligand-receptor complexes.^{67,68,314,315} These representations have been used to filter virtual screening results by binding mode and as inputs to ML models for binding affinity prediction.^{316,317} One such example is the extended interaction fingerprint presented as part of the LUNA Python toolkit,⁶⁶ designed for calculating and encoding protein-ligand interactions. In this work, we implement an extension to LUNA to generate identity-blinded geometric fingerprints that capture the chemical microenvironment of ion and water coordination sites.³¹⁸

Deep metric learning is an ML framework employed for facial recognition, anomaly detection, and signature verification applications.^{18,319–321} In contrast to classic ML approaches trained to predict a 1:1 label for each example, deep metric learning models learn an efficient virtual landscape that maximizes the distance between objects of different classes. It has been used in cheminformatics to learn molecular similarity and improve molecular property prediction.^{322,323} Metric models are often trained on triplets of examples sampled from each batch, consisting of an anchor, a positive example from the same class as the anchor, and a negative example of a different class.^{324–326} While this near-cubic increase in potential training examples can introduce implementation challenges, it is a useful property when operating in a data-sparse regime. We

exploit this property by training a metric learning model on a relatively small dataset of ions in structures from the Protein Data Bank (PDB) to learn a low-dimensional embedding (landscape) conditioned on deposited ion identity and use these embeddings for downstream identity classification.

Here, we present Metric Ion Classifier (MIC), an open-source tool for assigning identities to sites in PDB structures. MIC utilizes a novel ion-fingerprint representation and deep metric learning approach to predict the class of placed ions and waters in input structures. We demonstrate MIC's accuracy on a test set of structures from the PDB and use explainable AI feature attribution techniques to understand the biophysical rationale behind these predictions. Finally, we evaluate the performance on diverse x-ray crystallography and cryo-EM structures of both proteins and RNA, demonstrating the widespread utility of this approach. We hope this will prove a vital verification tool in structural biology workflows and represent an important step towards interpretable machine learning in the field.

Results

Architecture and Performance of MIC

The MIC tool assigns identities to waters and ions modeled in PDB structures. The overall workflow consists of three steps: 1) generating the fingerprint representation for the chemical environment of the density, 2) condensing this representation into a lower-dimensional embedding using a trained deep metric model, and 3) passing this embedding through a support vector classifier (SVC) to obtain final probabilities for all classes as well as prediction confidence (**Figure 2.1**). The class with the highest probability is then taken as the final prediction.

Prior approaches to conceptually similar tasks have used voxel representations as input to neural networks, necessitating large 3D-convolutional architectures that are both orientation-

dependent and rely on abundant training data to tune properly^{309,310}. We overcome these limitations in two ways. To represent each density, we use a modified version of the LUNA toolkit developed to calculate intermolecular interactions at protein-ligand interfaces and encode them into a fixed-length vector representation known as a "fingerprint." This greatly reduces the size of our models while meaningfully capturing information required for downstream classification. The generation process for these ion fingerprints is closely related to interface fingerprints with a few key exceptions (**Figure 2.1**). First, a proximity graph is constructed comprising all atoms $< 6\text{\AA}$ from the center of mass of the ion or water of interest. Each atom in this graph is assigned a set of atomic identifiers depending on its chemical identity and user-selected fingerprint type (**Supplemental Table 2.1**). Crucially, we remove the initial features of both the density itself and any additional waters or ions in the graph, effectively blinding the representation to any existing label to protect against data leakage during downstream prediction. The final list of modified atomic identifiers and all interactions are passed through a distance-dependent hash function that converts these input features into numeric values, which are folded down to 4,096 dimensions following standard molecular fingerprinting procedures.^{61,318}

These fingerprints are further condensed using a deep metric model. This model, constructed as a small feed-forward network, is trained to learn low-dimensional embeddings that maximize the distance between members of different classes (**Figure 2.1**). This step establishes the discriminative capabilities of the model, enabling accurate differentiation between closely related density types. The final predictions are generated by an SVC that uses these learned latent embeddings to calculate a probability for each class, the maximum of which is taken as the label (**Figure 2.1**). Full details for fingerprint generation and model training are provided in Methods, and the full list of sites used for training and testing are provided in **Supplementary Table 2.1**

and **Supplemental Figure 2.1**. The model was trained for 1000 epochs (**Supplemental Figure 2.2**).

In **Figure 2.2**, we present the performance of the MIC protocol trained on the six most prevalent classes from our curation of the Protein Data Bank: water, magnesium, sodium, zinc, calcium, and chloride. The model achieves an initial accuracy of 76.7% on a held-out test set and displays notable trends in performance by class, specifically showing high accuracy for zinc, magnesium, calcium, and water recovery. (**Figure 2.2**). A particularly interesting property of these learned embeddings is the organization by charge, visualized here with UMAP and confirmed by both PCA and the learned latent embeddings (**Figure 2.2, Supplemental Figure 2.2**). This constraint was not explicitly included in the representation or loss function during training, and the model was provided with no information about class relationships. Rather, this is an emergent learned quality of the transition of the chemical microenvironment of the sites themselves. The model's ability to learn the underlying structure inherent to the dataset supports the utility of our representation in capturing relevant information. Additionally, this reasonably organized continuous landscape also allows for confidence estimation through proximity to the classifier decision boundary, discussed below.

One potential drawback to MIC, and in fact most machine learning-based approaches, is a lack of interpretability of the resulting models, also known as the “black box” problem. We aimed to address this and provide further validation of the model through pairwise feature attribution with integrated gradients, a technique used to quantify the importance of input features to the model’s output.^{327,328} By calculating the attribution of fingerprints near the centroid of an ion cluster in embedding space, we can form hypotheses about which bits in the input fingerprint are most salient for a given class. Furthermore, we can use LUNA to trace back these features to their origin

in the input structure's atoms or interactions, allowing us to support the predictions with a biophysical rationale (The full details of feature attribution protocol as implemented by L. Ponzoni, PhD are provided in the Methods section).

To investigate the model's rationale behind the emergent organization by chemical microenvironment in the embedding space, we used pairwise attribution to probe the features most useful to the model for differentiating between closely related classes. Comparing two representative zinc and magnesium fingerprints (4L9P:B:ZN:601 and 4OKE:A:MG:202, **Supplemental Figure 2.2**) provides insight into how the model separates these embeddings despite similar charges. The nearby Cys367 sulfur appears in the top features by importance for zinc when compared against magnesium along with the short distance to the Asp365 sidechain carboxylate group. Visualizing the embedding space by the value of the corresponding fingerprint bit (2497) shows strong localization in the zinc cluster, following known properties of zinc binding sites and likely contributing to the high confidence prediction for this example (**Figure 2.2**).³²⁹ Conversely, our analysis showed that salient features for magnesium similarly prioritized the slightly longer distances of nearby carboxylates (Asp6, Glu8) and the number of nearby waters, commonly observed features of magnesium sites³³⁰. Comparing this same zinc against a calcium example (3BMV:CA:A:684, **Supplemental Fig. 2.2**) also yields known important features. In addition to the Cys367 and Asp365 recovered against magnesium, the attribution value of bit 3541 corresponding to the nearby His433 imidazole nitrogen is higher, indicating additional importance of this feature for the model in distinguishing calcium and zinc examples (**Figure 2.2**). In comparing the calcium and magnesium fingerprints, both assign high attribution to oxygens in their top features, but calcium includes backbone carbonyl oxygens while magnesium again includes the Asp6 carboxyl and the AMP phosphate oxygen, agreeing with known properties of

these sites.^{330,331} Interestingly, one feature that consistently returned high attribution was the null shell corresponding to the ion itself and any proximal waters and ions. The magnitude of the value at this index is consistently high and indicates the number of total sites encoded in the representation, a feature that is evidently useful to the model in structuring the learned latent space.

In addition to predicting the identity of a site, MIC provides a measure of confidence through the probability estimates output by the SVC. Because the latent representation transitions smoothly between chemically related classes, we can use the proximity to the decision boundary to measure confidence in a given prediction. Indeed, we found that our model was well-calibrated such that this simple metric showed statistically significant separation between test set predictions that agreed and disagreed with the deposited PDB label (P-value $\lll 1e^{-10}$, **Figure 2.2, Supplemental Table 3**). This property could assist the user in interpreting MIC results and encouraged us to further investigate these high-confidence disagreeing examples from the test set.

Manual Inspection of Disagreeing Sites

Following model prediction, we manually reviewed 455 disagreeing test examples and considered what the correct label should be based upon several factors including favorable/unfavorable interactions, experimental map agreement (x-ray structures were re-refined with the alternative density and Fo-Fc maps were inspected in both cases), and coordination geometric features (**Supplemental Figure 2.3**). We assigned each structure a score between -3 and 3, with increasingly positive scores denoting more support for the MIC prediction and increasingly negative scores support for the original label. We identified 135 sites where we believe the provided label in the PDB to be incorrect and MIC accurate in its assignment and 176 sites where MIC is likely incorrect and the deposited label is correct. A further 80 sites were scored as 0,

reflecting that even after manual inspection and re-refinement it was unclear which of the two labels were correct. 64 sites were also labeled as having unusual issues that would prevent proper prediction, including extended densities indicating the site represents a larger chemical entity than an ion or water, extensive heterogeneity and/or partial occupancy, the presence of an unusual multi-ion cluster, or that the likely correct identity of the ion did not fall within the set of predicted ions; indicated by a manual label of 30 (**Supplemental Figure 2.3**). In the manually annotated cases where the MIC assignment was correct over the deposited label, the average confidence was $78.0 \pm 17.2\%$, while the confirmed incorrect MIC predictions had an average confidence of $61.5 \pm 15.4\%$ (**Figure 2.2**). The revised overall test set accuracy following manual annotation is 83.3% with an average confidence of $84.8 \pm 15.3\%$ for correct predictions (**Figure 2.2**). The most common corrections made by MIC were reassigning spurious sodium and chloride ions to water (53 and 18 examples, respectively), followed by reassigning sodium to chloride and calcium to magnesium (11 examples) (**Supplemental Figure 2.2**). Given that 52 of the 258 total sodium sites in the test set were changed upon manual review, up to 20% of the sodium in the PDB may instead be water and up to 25% of all sodium in the PDB could be misannotated. Manually reviewing these examples additionally allowed us to provide an estimated accuracy cutoff by confidence (**Figure 2.2**). Except for sodium, the confidence of correctly predicted examples was significantly higher ($P\text{-value} < 1e-5$, **Supplemental Table 2.3**) than mispredicted examples. Overall, we found that a confidence of 70% was a useful cutoff in practice for most classes, and predictions below this cutoff typically require further review. Sodium is more challenging to predict confidently, likely due to the modest quality of annotated sodium ions in the dataset, and these predictions often require additional inspection.

Four diverse examples of high-confidence probable mismodeling captured by MIC are presented in **Figure 2.2**, showing sodium to chloride (PDB:1JG8), magnesium to chloride (PDB:3S70), chloride to water (PDB:2RL1), and sodium to water (PDB:6JIZ) substitution. In each case, there is at least one short-range (3.0-3.2 Å) unfavorable interaction and often several modest-range (3.5-4.0 Å) unfavorable charge interactions while lacking any opposite charge/partial charge interactions that would support the original assignment (for example, carbonyl interactions with a cation). None of the three deposited cations has the extended coordination shell or short coordination distances one would expect of a cation. Further, experimental difference maps were typically improved upon re-refinement with the MIC ion (**Supplemental Figure 2.3**), providing additional support for the corrected label.

Validation of MIC on Structures Derived from Cryo-EM Maps

As all but 9 structures in the training set derive from x-ray crystallography, we wanted to examine how well MIC would work on cryo-EM structures. For this purpose, we examined two disparate cases, representing the lower bound of resolution where an ion can still be resolved in a cryo-EM map (structures of melanocortin receptor 4 MC4R with bound calcium, nominal reported resolutions ranging from 2.6 Å to 3.1 Å) and the upper bound of resolutions currently possible with cryo-EM (apoferritin, 1.15-1.27 Å nominal resolution). In the first case, three different groups have determined several structures of MC4R bound to various ligands, resolving in each a spherical feature in the map thought by all three groups to be the calcium that has been biochemically demonstrated to be necessary for MC4R ligand binding.³³²⁻³³⁵ Further, some structures also resolve water molecules providing additional coordination for calcium ion binding. In the single structure from Israeli *et al.*³³² of MC4R bound to setmelanotide (PDB:7AUE, **Figure 2.3**), MIC correctly

identified calcium with 56.0% confidence, followed by sodium with 22.8% and magnesium with 13.8% confidence. In contrast, in the only other structure with an identical ligand, PDB:7PIU³³³ (**Figure 2.3**), the site was predicted to be either water (63.1%) or sodium (31.0%). This likely stems from the unexpectedly long carboxylate-calcium interaction distances modeled (**Figure 2.3**), which at 2.9-3.4Å are substantially longer than the ~2.4Å average one would expect for a carboxylate-calcium interaction.³⁰⁰ These coordination distances are similar to those of the other structure from Heyder *et al.*, PDB:7PIV³³³ (**Figure 2.3**), which MIC predicted to be sodium (40.4%) or calcium (35.9%), with the improved classification likely due to the presence of an additional carbonyl interaction. All four structures from Zhang *et al.*³³⁴ (PDB:7F53, 7F54, 7F55, 7F58; **Fig. 2.3**) are predicted to have a calcium ion at this site with high confidence (96.4%, 77.3%, 84.9%, 90.8%). Given the biochemical demonstration in Yu *et al.*³³⁵ that this is the site responsible for the calcium-dependence of ligand binding, all structures almost certainly did make the correct assignment as calcium, a result typically correctly predicted by MIC. In the case of 7PIV and particularly 7PIU, the discrepancy can be attributed to unusual coordination modeling, which is not unexpected in the ~2.5-3.0Å nominal resolution range where ions can begin to be resolved but extremely precise placement of sidechain atoms remains challenging. Thus, MIC in this resolution range also provides some level of audit on the overall modeling of the ion/water coordination site.

On the other end of the resolution spectrum are the atomic-resolution structures of apoferritin determined by several labs,³³⁶⁻³³⁹ generally producing superimposable structures (**Figure 2.3**), although not without some disagreements in ion modeling. In four examples (PDB:7A4M, 7RRP, 7A6A, 8J5A) a common coordination site near glutamate 27 and 62 is modeled as either sodium (7A6A, 8J5A) or zinc (7A4M, 7RRP) (**Figure 2.3**). Interestingly, in 3 of these cases (7A6A, 8J5A, 7A4M), MIC suggests a 70% or greater probability of zinc, while in

7RRP, where this site is modeled as zinc, MIC predicts a 71.2% chance of magnesium. Although the generally short coordination distances (1.9-2.1 Å) of two glutamates and a histidine support the choice of zinc in 7A4M, 7A6A, and 8J5A, the slight outward rotation and imidazole flip of histidine 65 in 7RRP weakens the case for zinc substantially as this interaction is abolished (it should be noted that in the case of 7A4M there is an alternative conformation for histidine 65 that matches 7RRP, however MIC only considers the first alternate conformation for a residue). 7RRP also includes several other ions not found in the other structures, including a zinc interacting with arginine 22 that, given the mismatched charges, should likely be a water or chloride and is predicted by MIC as water with 92.8% confidence (**Figure 2.3**). A sodium ion is also modeled interacting with the same arginine in 7RRP (**Figure 2.3**), similarly predicted by MIC to be a water with 97.8% confidence. These structures also have numerous waters modeled, and at this extremely high resolution, it is even possible at some sites to observe the slight deformation of the spherical densities due to the water hydrogens, providing experimental evidence for the water in some cases. Examining the 110 water molecules modeled in 7A4M, 106 (96.4%) are predicted to be water by MIC with an average confidence of $87.8 \pm 13.4\%$. Two sites are labeled as chloride at modest confidence (51.8% Cl, 47.3% water for A:HOH:380 and 78.9% Cl, 19.3% water for A:HOH:391), which is possible given their interactions but there is not enough evidence for the swap. The other two discrepant sites are immediately adjacent to the zinc site, and are also assigned to be cations (sodium and magnesium) at low confidence (43.7-49.8%). This is a consistent pathology we have observed with MIC for proteins, which is that water molecules that are part of the coordination sphere of a cation are often annotated as cations with low confidence (**Supplemental Figure 2.2**). This likely stems from the fact that the model is blinded to the identity of the other nearby sites, and waters that are part of a cation coordination shell often have relatively

short distances to several anionic side chains and potential ion sites themselves. To account for this, MIC warns when a site is part of a dense cluster of other sites to examine the central, high confidence site as the probable ion. Overall, the MIC method performs well for the cryo-EM structures, especially those obtained at very high resolution.

RNA/Ribosomal structure evaluation

We wanted to examine the performance of MIC on structures of RNA, where ion binding is also pivotal,²⁸⁸ but only 72 of the 10,364 individual structures in the prevalent-ion training set contained RNA or RNA/protein complexes, corresponding to 122 ion/water sites. In general, MIC was still able to perform reasonably well on RNA-bound ions in simple high resolution RNA structures, likely correctly predicting 8/9 ions in 8D2B, 2/2 ions in 5HNJ, and 5/5 non-potassium ions in 1L2X (**Table 2.1**). This includes in some cases probable corrections, for example predicting the three sodium ions in 1L2X to have a strong potential to be water (**Figure 2.4**). This result is consistent with the overall long coordination distances for a sodium (generally 2.7-2.8 Å vs 2.4 Å expected) and the lack of more than 2 definitive hydrogen bond acceptors or 4 interaction partners total. However, where the model has more difficulties in RNA-bound structures are water molecules, which tend to be overpredicted as cations. For 1L2X, MIC had 73.8% accuracy over the 160 waters with 75.6±1.6% confidence for correct assignments and 56.5±0.13% confidence for incorrect, demonstrating both less accurate and less confident guesses, with every misassignment either sodium or magnesium. Indeed, even the sodium ions in 1L2X likely correctly predicted to be water only have ~50% confidence.

This trend persists when evaluating MIC on ribosomal structures. In the case of 8CGV, the bacterial 50S ribosome at 1.66 Å resolution, MIC correctly predicts 212/219 magnesium with an

average of $94.5 \pm 8.3\%$ confidence (although some, such as MG:V:102 and MG:A:3263 which are predicted to be water, are likely mismodeled, **Figure 2.4**), the sole zinc correctly with 99.9% confidence, but only 4,227/6,570 water molecules with an average confidence of $70.7 \pm 16.0\%$. We anticipate this is likely due to the relative paucity of training data (only 56 waters in the training set are from RNA-containing structures) and will improve with further model training on additional deposited structures.

Extended set model training, performance, and manual review

Another potential pitfall highlighted in the RNA work is the lack of inclusion of potassium or other less well-represented ions in the PDB that nevertheless can be found in structures, as the prevalent-ion model is incapable of producing the correct answer in these cases. We trained an additional model that includes potassium, iron, manganese, bromide, and iodide in addition to the prevalent ions, although there were less than 1,000 examples of each of these new classes (**Supplemental Figure 2.1, Supplemental Figure 2.4**). This extended-set model achieves an initial accuracy of 69.3% against the deposited test labels and displays similar results to the prevalent-ion model in embedding space organization and accuracy by class. The embedding space is again organized primarily by charge as visualized by the UMAP and confirmed by PCA, transitioning smoothly from the halides to water, to monocations, and ending with the transition metals (**Figure 2.5, Supplemental Figure 2.5**). Mis-predictions on the test set were often chemically reasonable, such as predicting bromide as either chloride or iodide, iron as zinc, or manganese as magnesium (**Supplemental Fig. 2.4**). Among the added classes, iodide shows high AUROC and AUPRC values as well as separation between the confidence values of agreeing and disagreeing predictions (**Supplemental Figure 2.4**).

Similar to the prevalent set, we manually reviewed the set of discrepant ions in the extended test set using the protocol described above (**Supplemental Figure 2.5**). This included 415 examples that were predicted to belong to a class different from the deposited label by both the prevalent and extended models as well as an additional 161 disagreeing sites belonging to the added extended classes. We observed a number of similar trends, such as a large number of sodium sites and 12 of the 86 potassium sites corrected to water in our dataset, suggesting that potassium may also be misannotated throughout the PDB (**Figure 2.5, Supplemental Figure 2.5**). Even when the MIC prediction is incorrect, it can often still help identify likely changes, such as predicting the magnesium in 4AK8 to be a bromide (63.3% confidence), while the true identity is likely chloride. The final accuracy of the extended set model following manual review was 76.3%, and confidence was once again a strong measure of correctness for many classes in the prevalent set (zinc, magnesium, water, calcium) and newly introduced classes (potassium, iodide, iron, and bromine) (**Figure 2.5**). We observe worse chloride performance compared to the prevalent-only model, likely from the inclusion of additional halide classes that remain difficult to differentiate due to the low number of training examples. Despite this overall slight decrease in accuracy from the prevalent-only model, it is still able to successfully classify sites belonging to many different ions and can be used when one of these additional ion classes is likely.

Comparison with existing methods

The most ubiquitous method currently used to assign identities to ion sites is the CheckMyMetal (CMM) web server.^{299–302} CMM uses a combination of known binding site properties to evaluate each input structure. Each property (atomic contacts, valence, and geometry) contributes a score between 0 and 2 resulting in a maximum score of 6 for a given ion identity at a particular site. The

score of each potential metal is reported, often leading to multiple ions receiving comparably high scores. During manual inspection of the disagreeing test examples, we ran CMM on all structures to identify cases where CMM and MIC differ in either their predicted class or from the deposited label in the PDB.

CMM and MIC produce concordant results for many sites. In 3FOB, MIC predicts the deposited sodium ion to be a magnesium (88.6%), consistent with the respective scores of 4 and 6 for these metals from CMM, although cobalt and manganese also receive a CMM score of 6 (**Figure 2.6**). Similarly, both MIC and CMM predict the calcium site in 1XPH to be a magnesium, receiving a CMM score of 6 and a MIC confidence of 82.4%. In sites where manual inspection revealed the deposited metal was likely water or chloride, CMM generally gave poor scores for either all metals or all metals except potassium. The sites of magnesium to chloride corrections in 4KP1, 3S70, and 3A4X, and the magnesium to water correction in 5VX0, receive a 0 from CMM for all ions, but no additional distinction between these cases is provided as CMM is specifically designed for validating metals. Additionally, there are cases where MIC likely identifies the correct label while CMM does not, notably in correcting ions to water: 6RJ4 is one such example containing a deposited sodium ion. CMM gives this site a potassium score of 6, followed by a sodium score of 4, showing significant disagreement with the high-confidence MIC water prediction (95.8%) that was accepted upon manual review, as re-refinement with water at this position improves both the difference map and interactions at this site (**Fig. 2.6b**). Conversely, because CMM predicts the output for multiple classes, it can in some cases succeed where MIC fails; the identity of the magnesium ion in 2ICJ was confirmed experimentally³⁴⁰ and agrees with the high CMM assignment of 5 for magnesium, while MIC predicts this site to contain a high-confidence zinc ion, which only scored a 4 by CMM. Finally, CMM and MIC have distinct output

classes that make each more suitable for specific cases. CMM predicts the score for several metals that are not in the MIC class set, including copper, cobalt, and nickel, while MIC includes an explicit prediction for water and halides.

Another tool with some overlapping use is Undowser, which is intended to find waters that clash with nearby atoms as these could indicate that the site would better be modeled as a metal. We ran MIC and Undowser on a selection of identity-blinded waters and ions to compare the results (see Methods). Like CMM, Undowser often agrees with MIC predictions. Both tools identify the zinc sites in 2C1I (A:1465,1466) with a MIC zinc confidence of 98.6% and 91.4%, and an Undowser cumulative clash severity score of 2.797 and 2.121, respectively, each comprising of multiple $>0.5\text{\AA}$ polar clashes strongly indicating the presence of an ion (**Figure 2.6**). Similarly, the magnesium ion in 4RKQ is caught by both tools (MIC: 98.1%, Undowser: 2.098). Even when MIC is unable to predict the correct identity, it is often able to distinguish what should be an ion binding site, such as the iron sites in 1YFU, predicted as zinc by the MIC prevalent-only model with 98.6% confidence and an Undowser clash severity score of 1.985. MIC does show a tendency to over-predict ions compared to Undowser, though similar to the RNA/ribosomal predictions these assignments typically have a lower confidence ($57.1\pm 15.1\%$) than water predictions that agree with Undowser ($85.5\pm 13\%$), helping the user handle these cases. Undowser and MIC both fail where the modeling is questionable, as is the case for 6E27:C:HOH:201, which both MIC and Undowser flag with high zinc confidence and clash score (88.7%, 1.854 \AA). However, 6E27 shows no major positive difference density and lacks density in the 2Fo-Fc map for much of the protein at this site (**Figure 2.6**). Undowser provides information about the charge of clashing atoms that can assist the user in interpreting the results, but does not explicitly attempt to predict the true ion identity. Undowser does not calculate any clashes for halides such as the

chlorines in 3MUJ and 4RKQ, which are predicted correctly by MIC. Ultimately, these are complementary but not overlapping methods of confirming correct modeling, and users should choose the tool that best aligns with their specific requirements.

Discussion

MIC is a novel method to classify water/ion sites in biomolecular structures and provide confidence estimation in these predictions. It uses structured embeddings from a deep metric learning model to generate accurate predictions for structures from both x-ray crystallography and cryo-EM. Notably, this representation includes no angle information and only coarse distance information through constructing the initial microenvironment fingerprint by shell expansion. Despite this, our results show that the fingerprints contain sufficient data about the atomic environment to correctly distinguish ions in over 80% of our observed cases. We demonstrate that MIC achieves incredible accuracy for extremely high-resolution cryo-EM structures. Furthermore, the results for MC4R suggest that ambiguity and inaccurate assignments by MIC may result from problematic protein modeling. The output probabilities can be used to estimate model confidence, further enabling MIC's use as a validation tool. We use MIC to identify mismodeled ions throughout the PDB and show that the tool can be used to evaluate the modeling of coordinating side-chains in low-resolution structures when the desired ion identity is known. Finally, we show that MIC performs comparably to field-standard approaches and offers further utility over these methods by including additional output classes.

There are limitations to MIC's use. Our dataset was restricted to ions for which at least a few hundred high-quality structures exist in the PDB, limiting the possible classes we could include when training the deep-learning model and SVC portions. As mentioned previously, there are concerns about the quality of the training dataset, specifically the inclusion of misannotated

ions from the PDB. Sodium in particular achieves lower performance and even correct sodium predictions are often low confidence, likely due to mislabeled sodium sites introducing noise into the dataset. MIC also tends to predict waters that are very close to a cation and participate in a larger coordination cluster as cations, necessitating an additional flag to account for these cases. MIC is designed to work for sites centered in experimentally determined density, and is not expected to give accurate results for sites with few or no interactions.

The current MIC workflow uses a deterministic hash function to convert the site microenvironment to a fixed-length fingerprint as input to the deep-learning model. This introduces the potential for bit collisions that occur when multiple features hash to the same index in the fingerprint, complicating both model training and feature attribution analyses. A future expansion on this method could replace this fingerprinting method with a task-specific representation learned from this proximity graph, similar to those used for molecular property prediction^{69,105,164}, further enabling downstream classification and limiting the problem of bit collisions. The metric model was trained using a triplet loss to maximize inter-class distances and successfully learns class relationships, but this could be explicitly enforced using a hierarchical loss function. As new structures are added to the Protein Data Bank and ion sites are subject to more careful validation and scrutiny, this additional training data will further improve MIC's accuracy for future iterations.

Methods

Dataset Curation

Our final dataset consisted of 23,101 examples split across 11 classes. In decreasing order of number of examples: water, magnesium, sodium, zinc, calcium, chloride, potassium, manganese, iodine, iron, and bromine. Candidate structures were restricted to < 2.0 Å resolution except for

potassium, bromide, and iodide, which were relaxed to $< 3.0 \text{ \AA}$. Structures were restricted to have no more than 95% sequence homology to another example in the dataset to avoid overrepresentation. Only the first example of each ion type and a single water was taken from a given structure to prevent redundant symmetry-related ions being added to the dataset and potential modeler-related biases. Finally, to prevent the inclusion of water molecules and/or ions modeled into empty space with no interaction partners, sites were filtered to have at least 2 other atoms within 3.5 \AA . The full list of ions and their associated counts and resolutions is provided in **Supplemental Figure 2.1**.

Density Fingerprint Representation

We represented each density using a modified version of the interaction fingerprint available in the LUNA toolkit. LUNA fingerprints were developed to capture the interactions at the protein-ligand interface of a bound complex. This is accomplished by assigning biochemical properties of individual atoms and atomic groups, then defining interactions as pairs of atoms/atomic groups in proximity that meet certain geometric and chemical properties. The ligand properties and final list of intermolecular interactions are then converted into a fixed length vector using the MurmurHash3 algorithm, referred to as a fingerprint.^{61,341}

We made two crucial modifications to the generation protocol of these interface fingerprints for our use case. First, to ensure that any previous density assignment was not encoded in the representation, trivializing any downstream classification task, the initial atomic representation for each density of interest and nearby spherical densities was programmatically set to a vector of zeros. All atoms belonging to the protein or nearby small molecules are given their standard initial feature set (described in more detail below). The second modification was

limiting all calculated interactions to be proximal-only, defined as simply being located between 2Å and 6Å from the ion. Proximal interactions are unique in that they do not rely on the chemical features of either participating atomic group, and thus the resulting representation continues to be identity-agnostic. The final fingerprint is effectively an identity-blinded representation of the atomic environment surrounding the density of interest.

Initial atomic features and included interactions

LUNA provides extended interaction fingerprint (EIFP) and functional interaction fingerprint (FIFP) featurization options for the user during fingerprint generation. EIFPs use the Daylight atomic invariants for the initial atomic feature set, consisting of 7 fields: number of heavy atom neighbors, valence minus the number of bound hydrogens, atomic number, isotope number, formal charge, number of hydrogen neighbors, and aromaticity.⁵⁰ Functional fingerprints use pharmacophore-like features, such as whether an atom is aromatic, hydrophobic, or a hydrogen donor or acceptor. **Supplemental Table 2.1** contains the full list of initial atomic features for each fingerprint type. In addition to these options, we also considered whether or not to include interactions between neighbors in our representation versus a “pruned” representation with interactions limited only to those between the density of interest and neighbors. We evaluated the four fingerprint types (non-prune/eifp, prune/eifp, non/prune-fifp, prune/fifp) by generating the specified fingerprint type and training an SVC to predict ion identity. We found that prune-eifp fingerprints performance exceeded that of the other types, especially in multi-class classification tasks, and used that type for all presented work (**Supplemental Figure 2.1**).

Shell Number and Radius

During the fingerprint creation process, interactions in shells around the ligand atoms are iteratively converted to fixed integers, similar to the process for generating molecular Morgan fingerprints.⁶¹ The user sets the size and number of these shells during creation, with the default LUNA values being 2 shells of 6Å radius step each. We hypothesized that these values, optimized for longer intermolecular protein-ligand interfaces, would not provide sufficient granularity to differentiate between ions. To address this, we explored a range of different shell radii and depths by randomly selecting up to 2000 examples from each ion class in the dataset, generating fingerprints with the specific radii and number of shells, and training an SVC³⁴² with 5-fold cross-validation to predict ion identity (**Supplemental Figure 2.1**). Accuracies ranged from 0.44 to 0.63 for all fingerprint types, with a general rule that the product of the radius and number of shells should be between 3Å and 5Å for best performance. Lower radii performed better overall, suggesting that the additional discrimination provided by finer shells is useful for downstream identity classification, though this does make the representation more sensitive to slight changes in atom position. All final fingerprints were generated with 18 shells of radius step 0.25Å, comprising a total volume of 4.5Å around each atom in the proximity graph. Count fingerprints of length 4,096 were used for all experiments, consistent with the original LUNA manuscript.

Training and test datasets of curated densities for MIC

Each site in our curated dataset was randomly assigned to training or testing with a 90%/10% split. We chose random splits because of the strict criteria during dataset curation, limiting the similarity between all examples. All ions belonging to a class with fewer than 1000 examples were dropped for the prevalent-only models, resulting in final datasets of 18,420 training and 2037 testing

examples. The extended set included all of these sites plus examples from potassium, manganese, iron, bromide, and iodide, bringing the total number of training examples to 20,801 with 2,300 examples used for testing. Training and testing splits were consistent across all fingerprint types and hyperparameter optimization.

Model Training

We present two metric learning networks one trained on the prevalent set of ions and one trained on the extended set. Models were trained using the Pytorch Metric Learning library with triplet margin sampling and triplet loss.^{343,344} Hyperparameter optimization was performed with the Optuna library to evaluate the effect of learning rate, dropout, loss and miner margins, and embedding dimension.³⁴⁵ The full list of evaluated hyperparameters, ranges, and final values are displayed in **Supplemental Table 2**. Models were optimized to maximize two downstream metrics: the average area under the receiver operating characteristic (ROC) curve and the F1-score for a linear-kernel SVC trained on the embeddings from the metric learning model. Due to significant class imbalance, each batch was generated by weighted random sampling with replacement, resulting in approximately balanced batches. The architecture of all final models consists of 1 hidden layer of 4000 neurons each with output size of 32 for the resulting embeddings. The model was trained for 1000 epochs (**Supplemental Figure 2.2**).

Feature Attribution

Pairwise feature attribution was calculated between representative examples for each class using a modification of the popular integrated gradients technique implemented in the Captum library.^{327,328} The specific examples were chosen by selecting those close to the cluster centroid for

a given class in the learned embedding space. A baseline fingerprint of zeroes was used for all calculations. Following default global attribution rules, the resulting attribution vector is multiplied by the input fingerprint. While this is known in practice to result in cleaner attribution features and improve the ease of interpretation of the results, it is an important limitation to note as only features that are turned “on” for a given fingerprint will be assigned a non-zero attribution value. The features corresponding to the top ten bits with the highest attribution for both comparisons are available in **Supplementary Table 2.3**.

Undowser Comparison

The comparison with Undowser was performed by randomly selecting structures from the PDB that fit our resolution requirements and contained at least one non-water density, converting all of the ions to water, and running Undowser to determine if the clashing “waters” matched with the non-water MIC predictions.

Statistical Analysis

All statistical analyses performed were two-sided t-tests for independence, as implemented in the Scipy³⁴⁶ statistics module. The full list of comparisons, number of examples, and P-values are provided in **Supplemental Table 2.3**.

Data and Code Availability

The complete source code for MIC, all training data, trained models, and associated tutorial Jupyter Notebooks are freely available under the open-source MIT license at <https://github.com/keiserlab/metric-ion-classification>.

Acknowledgments

This work was supported by NIH T32 GM067547 and the UCSF Graduate Division (L.S), CZI grant DAF2018-191905 (DOI 10.37921/550142lkcjzw) from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation (funder DOI 10.13039/100014989) (M.J.K.), NIH K99/R00 HD107581 (M.J.R.), and CPRIT award RR230042 (M.J.R.). We would like to thank Alexandre Fassio for his assistance in adapting the LUNA package, Luca Ponzoni for his work and advice in implementing feature attribution, and Mahdi Ghorbani, Brendan Hall, and Zachary Gale-Day for their useful advice throughout this project.

Author Contributions

L.S. designed and wrote the MIC software, trained and evaluated the machine learning models, and generated predictions for all structures. M.J.R. conceived the project, curated training data, and performed manual validation and structure re-refinement. W.L. provided code review and software validation. L.S. and M.J.R. wrote the manuscript with input from G.S., M.J.K., and W.L. M.J.R., G.S., and M.J.K. supervised the project.

Figures

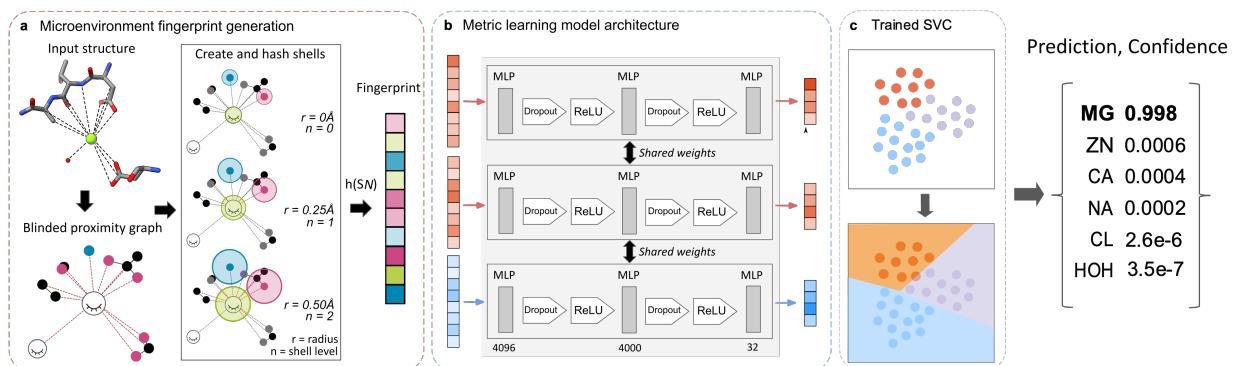


Figure 2.1. Overview of MIC workflow.

MIC is a multi-step ML workflow for classifying experimental water and ion sites. **a**, Ion fingerprints are generated by first constructing a proximal interaction graph containing all atoms within 6Å for the density of interest. The fingerprint generation protocol iteratively captures local chemical information by hashing the atomic invariants and interactions within consecutive shells originating from each atom. The example structure shown here is 4KU4:A:Mg:302. **b**, The fingerprints are embedded into a lower dimensional embedding space by a metric learning model consisting of a 4096-dimensional input layer, a single hidden layer with 4000 neurons, and an output layer of 32. ReLU is an activation function $x = \max(x, 0)$. **c**, The final step of MIC is using an SVC on the generated fingerprints to output probabilities for each class. The class with the highest probability is taken as the predicted label.

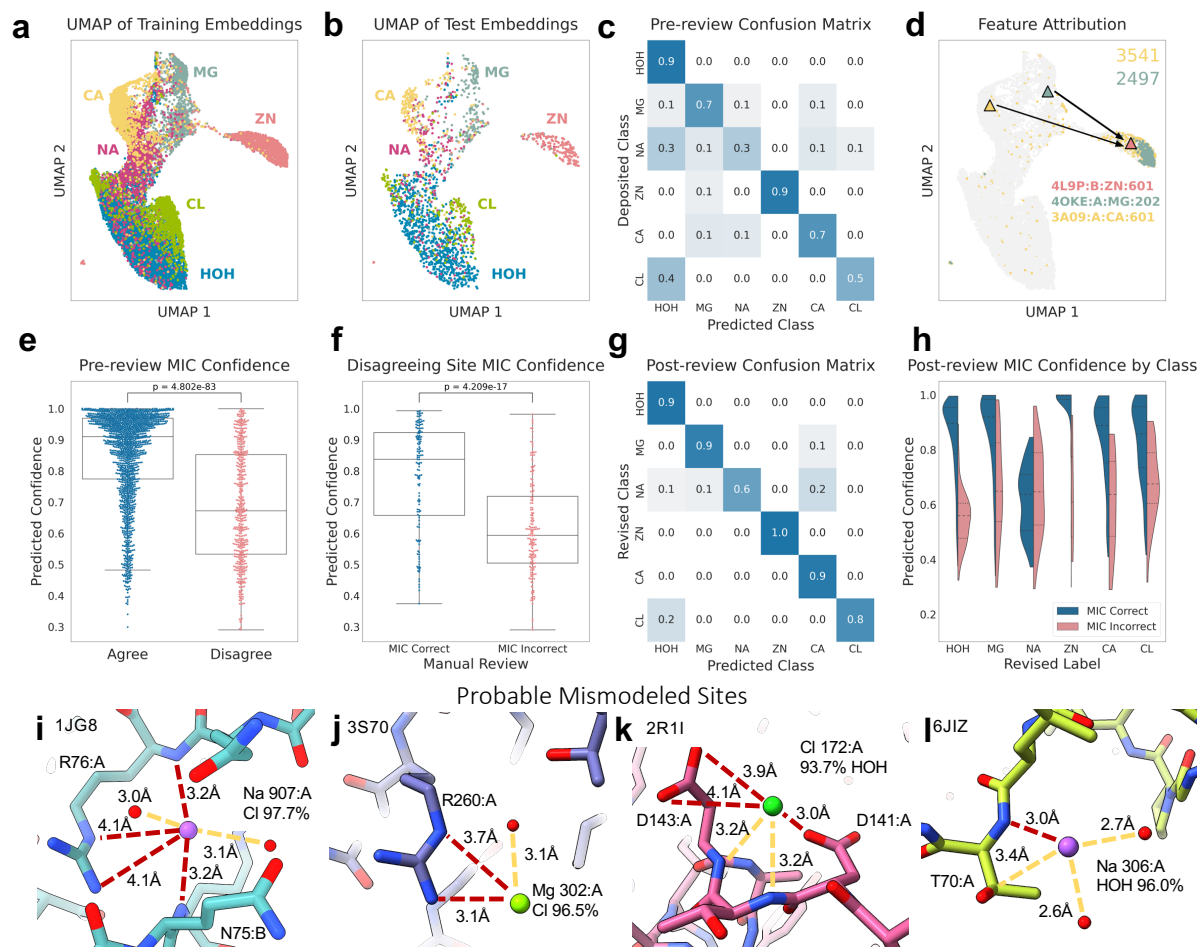


Figure 2.2. MIC learned embeddings, performance, and validation.

a-b, UMAP visualization of training and test set embeddings from the MIC prevalent-set model, colored by deposited class. **c**, Confusion matrix of the deposited labels and MIC predicted labels for the test set. **d**, UMAP visualization of training set embeddings, colored by the value of the bits 2497 (green) and 3541 (yellow), corresponding to the presence of a cysteine sulfur and imidazole nitrogen, respectively. The triangles indicate the position of specific examples used to perform feature attribution: 4OKE:A:Mg:202 (green), 3A09:A:Ca:601 (yellow), and 4L9P:B:Zn:601 (pink). **e**, Comparison of the confidence values for MIC predictions that agree vs disagree with the deposited label. **f**, Comparison of confidence values for manually inspected disagreeing examples with accurate vs inaccurate MIC-predicted labels. **g**, Confusion matrix of revised labels and MIC predictions following manual review of disagreeing test examples. **h**, Violin plots of the confidence of correct vs incorrect MIC test set predictions, split by class. **i-l**, Examples of disagreeing annotations with probable mismodeling. **i**, Sodium in 1J86 corrected to a chloride, 97.7% confidence. **j**, Magnesium in 3S70 corrected to chloride, 96.5% confidence. **k**, Chloride in 2RL1 corrected to water, 93.7% confidence. **l**, Sodium in 6JIZ corrected to water, 96.0% confidence. Red dashed lines depict unfavorable interactions in the originally modeled structure.

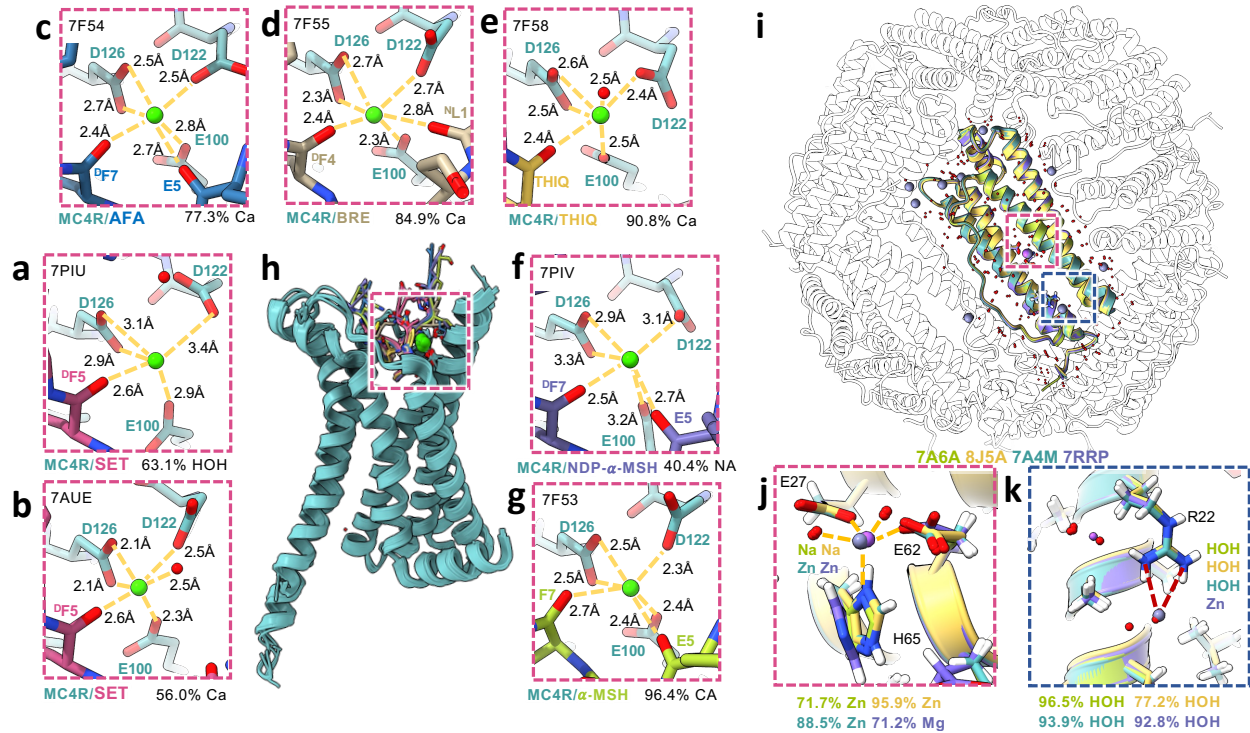


Figure 2.3. MIC predictions on Cryo-EM structures of MC4R and apoferritin.

a-h, MC4R Ca^{2+} -coordination site in complex with various ligands: setmelanotide (SET, **a,b**), afamelanotide (AFA, **c**), bremelanotide (BRE, **d**), THIQ (**e**), NDP- α -MSH (**f**), and α -MSH (**g**). **i-k**, Superimposed ion coordination sites in four apoferritin structures: 7A4M (green), 7RRP (purple), 7A6A (teal), 8J5A (yellow). **j**, For three structures, the ion is predicted to be zinc with confidence exceeding 70%. The 7RRP outwardly turned histidine imidazole shifts the prediction from zinc to a high confidence magnesium. **k**, Superimposed ion coordination site in four apoferritin structures: 7A4M (green), 7RRP (purple), 7A6A (teal), 8J5A (yellow). An additional site is shown in the top left, assigned sodium in 7RRP and water in all other structures.

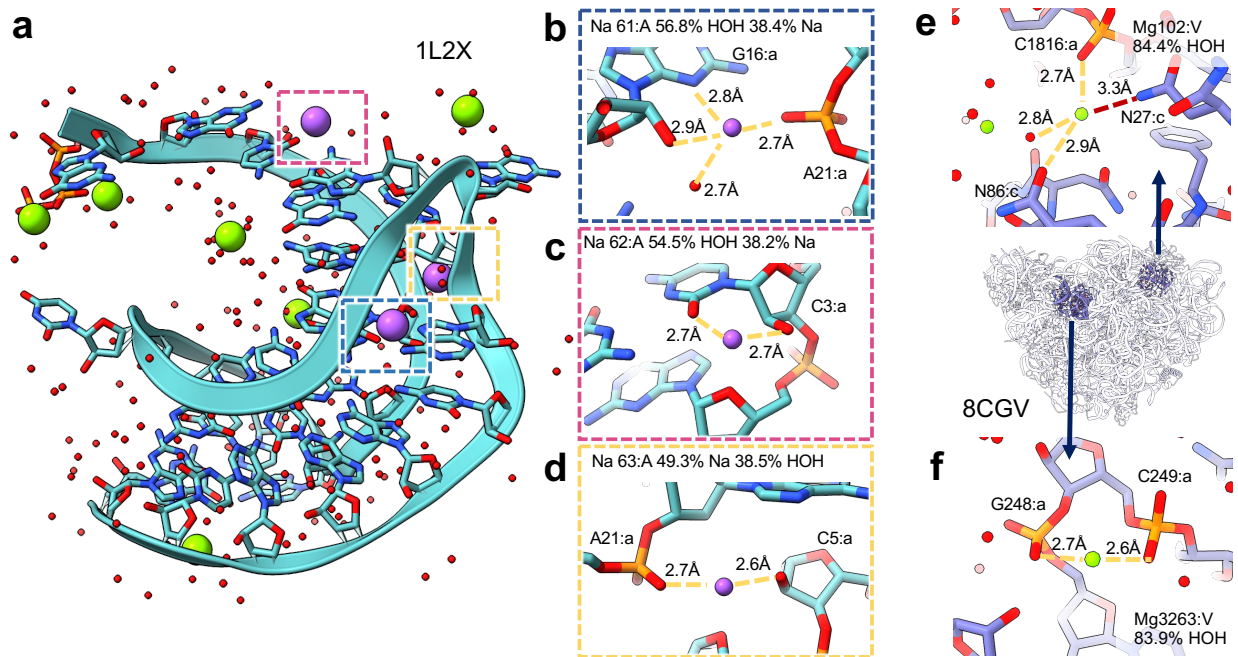


Figure 2.4. Predictions on RNA/Ribosomal structures.

a, Structure of viral RNA pseudoknot (PDB: 1L2X). **b-d**, Sodium sites with either low-confidence water (**b,c**) or low-confidence sodium (**d**) MIC predictions. **e,f**, Potentially mismodeled magnesium ions in PDB 8CGV, predicted to be water with high confidence. **e**, MG:V:102. **f**, MG:A:3263.

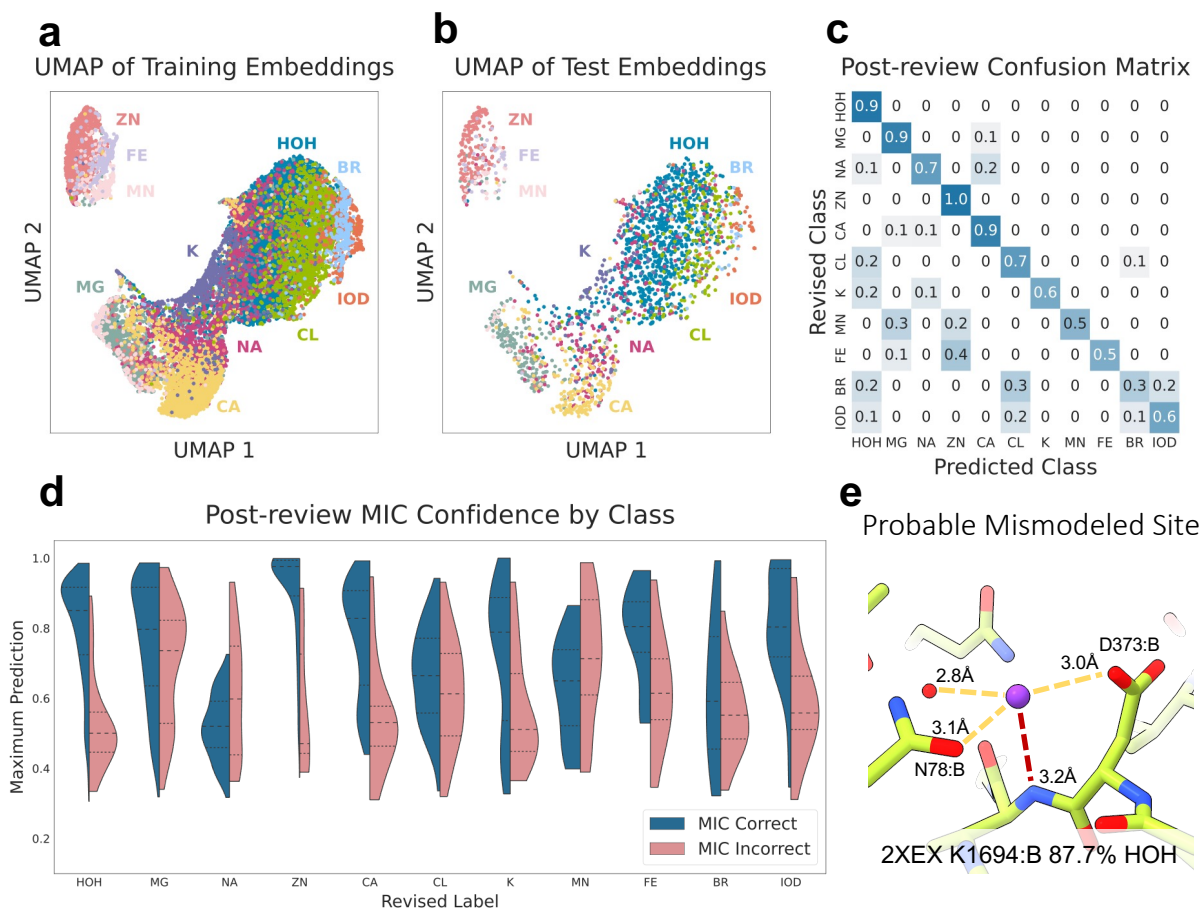


Figure 2.5. MIC Extended ion set performance and manual review.

a-b, UMAP visualization of training (**a**) and test (**b**) set embeddings from trained extended-set MIC model. **c**, Confusion matrix of MIC predictions vs revised label following manual review. **d**, Violin plots of the confidence of correct vs incorrect MIC test set predictions by class. **e**, Probable mismodeling of a potassium site in PDB2XEX, predicted as water by MIC with 87.7% confidence.

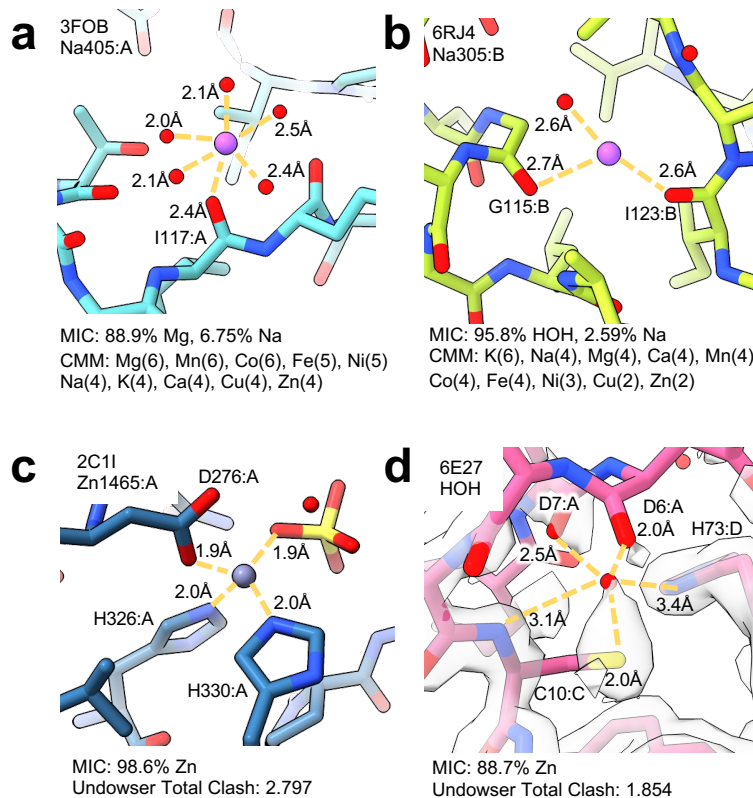


Figure 2.6. MIC, CheckMyMetal, and Undowser performance examples.

a, Example of CMM and MIC both correcting sodium (3FOB:A:Na:405) to magnesium (MIC confidence: 88.9%, CMM Mg Score: 6). **b**, Example of a site (6RJ4:A:Na:305) MIC likely predicts correctly as a water (95.8% confidence) over CMM, which assigns a score of 6 for potassium and 4 for five other metals. **c**, Zinc coordination site (2C11:A:Zn:1465) identified by both MIC and Undowser with high confidence (MIC zinc confidence: 98.6%, Undowser clash score: 2.797). **d**, Example of Undowser and MIC results at a questionably modeled site (6E27:C:HOH:201). This site likely does not contain either an ion or water, but is predicted to be an ion by both Undowser (Clash score: 1.854) and MIC (zinc confidence: 88.7%).

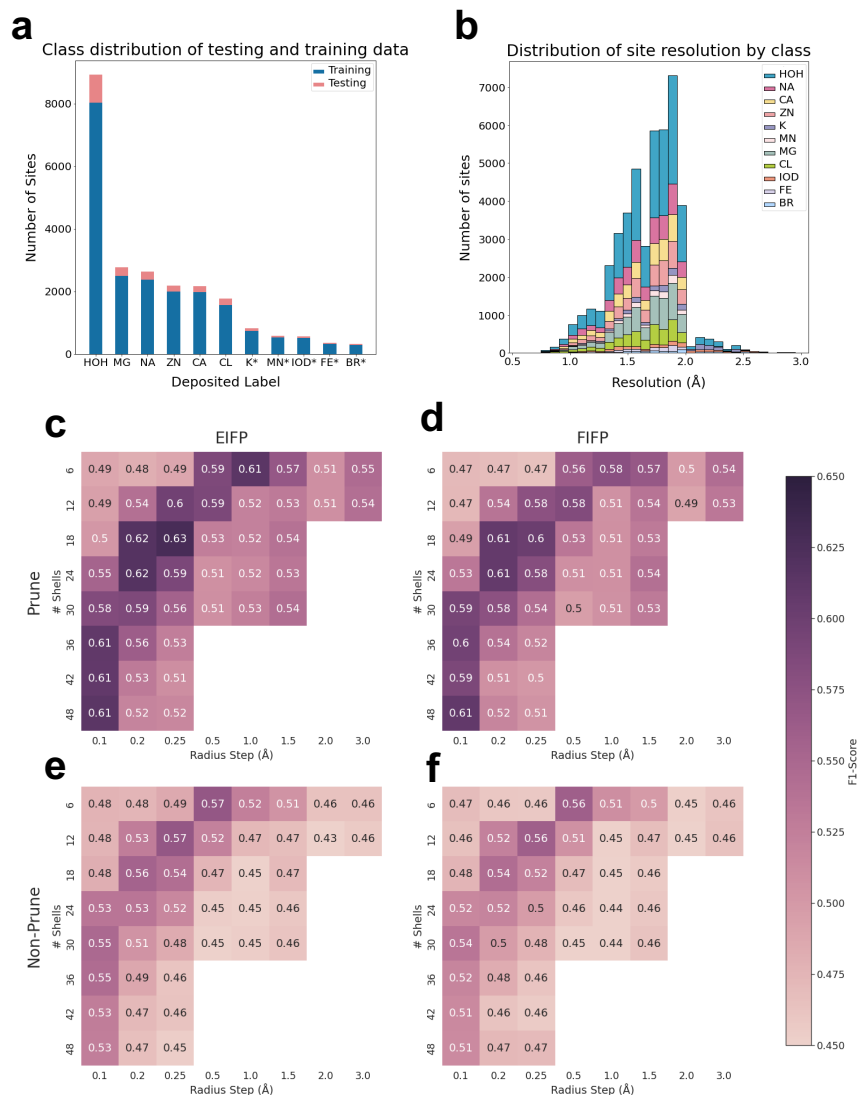
Table 2.1. Summary of RNA/Ribosome structure performance.

* indicates structures for which all non-water sites were manually examined

+ averages shown, mean \pm standard deviation

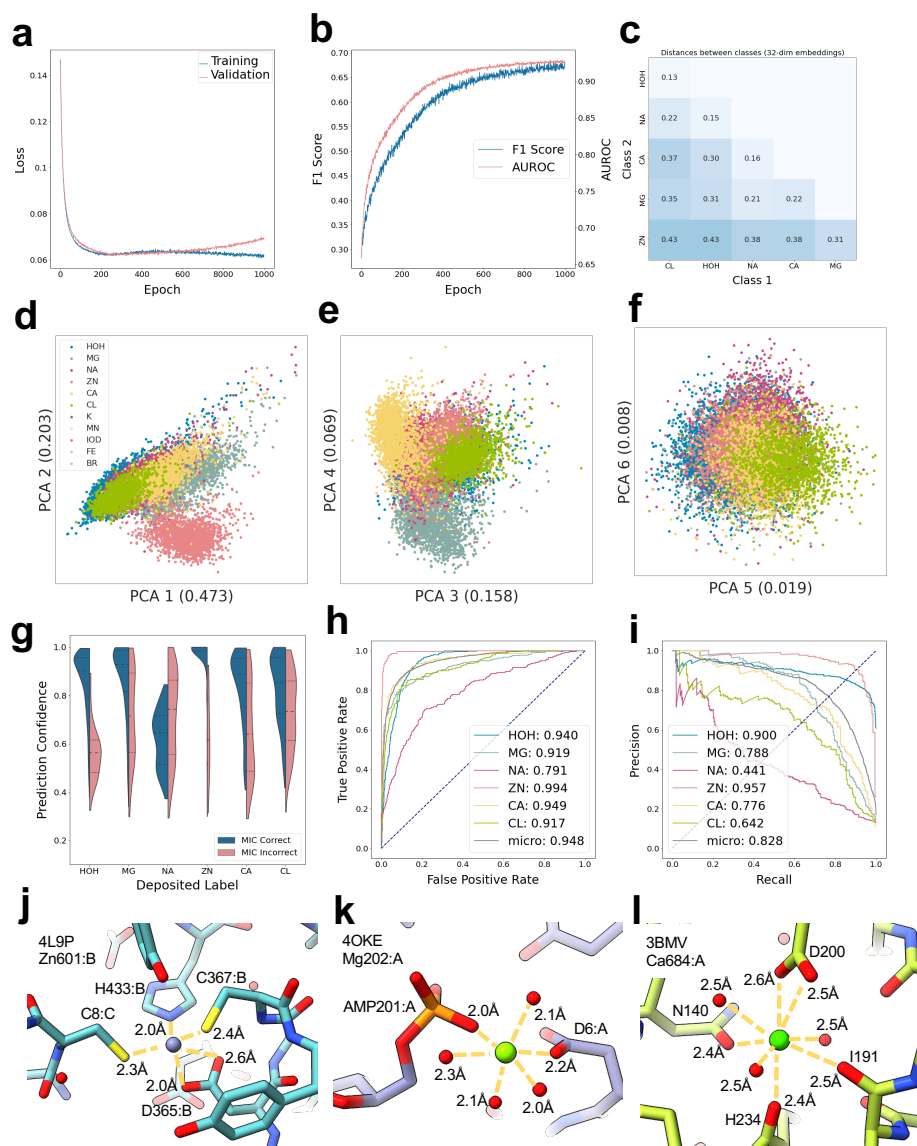
PDB	Class	Res (Å)	Ions			Waters		
			Correct (%)	Correct Confidence ⁺	Incorrect Confidence ⁺	Correct (%)	Correct Confidence ⁺	Incorrect Confidence ⁺
5HNJ*	RNA	1.24	2/2 (100%)	0.997 \pm 0.02	-	143/248 (57.7%)	0.688 \pm 0.16	0.612 \pm 0.16
8D2B*	RNA	1.44	8/9 (88.9%)	0.703 \pm 0.19	0.400 \pm 0	235/319 (73.7%)	0.738 \pm 0.15	0.576 \pm 0.12
1L2X*	RNA	2.25	5/5 (100%)	0.920 \pm 0.10	-	112/148 (75.7%)	0.756 \pm 0.16	0.560 \pm 0.13
8CGV	Ribosome	1.66	213/220 (96.8%)	0.946 \pm 0.08	0.593 \pm 0.22	4227/6570 (64.3%)	0.707 \pm 0.16	0.607 \pm 0.18
7ZHG	Ribosome	2.25	85/96 (88.5%)	0.809 \pm 0.18	0.511 \pm 0.12	1533/2083 (73.6%)	0.737 \pm 0.16	0.578 \pm 0.15
1YHQ	Ribosome	2.40	114/190 (60%)	0.743 \pm 0.20	0.629 \pm 0.16	5573/7763 (71.8%)	0.724 \pm 0.16	0.557 \pm 0.14
3CC2	Ribosome	2.40	133/224 (53.4%)	0.765 \pm 0.21	0.634 \pm 0.17	5783/7823 (73.9%)	0.729 \pm 0.16	0.559 \pm 0.15

Supplemental Figures



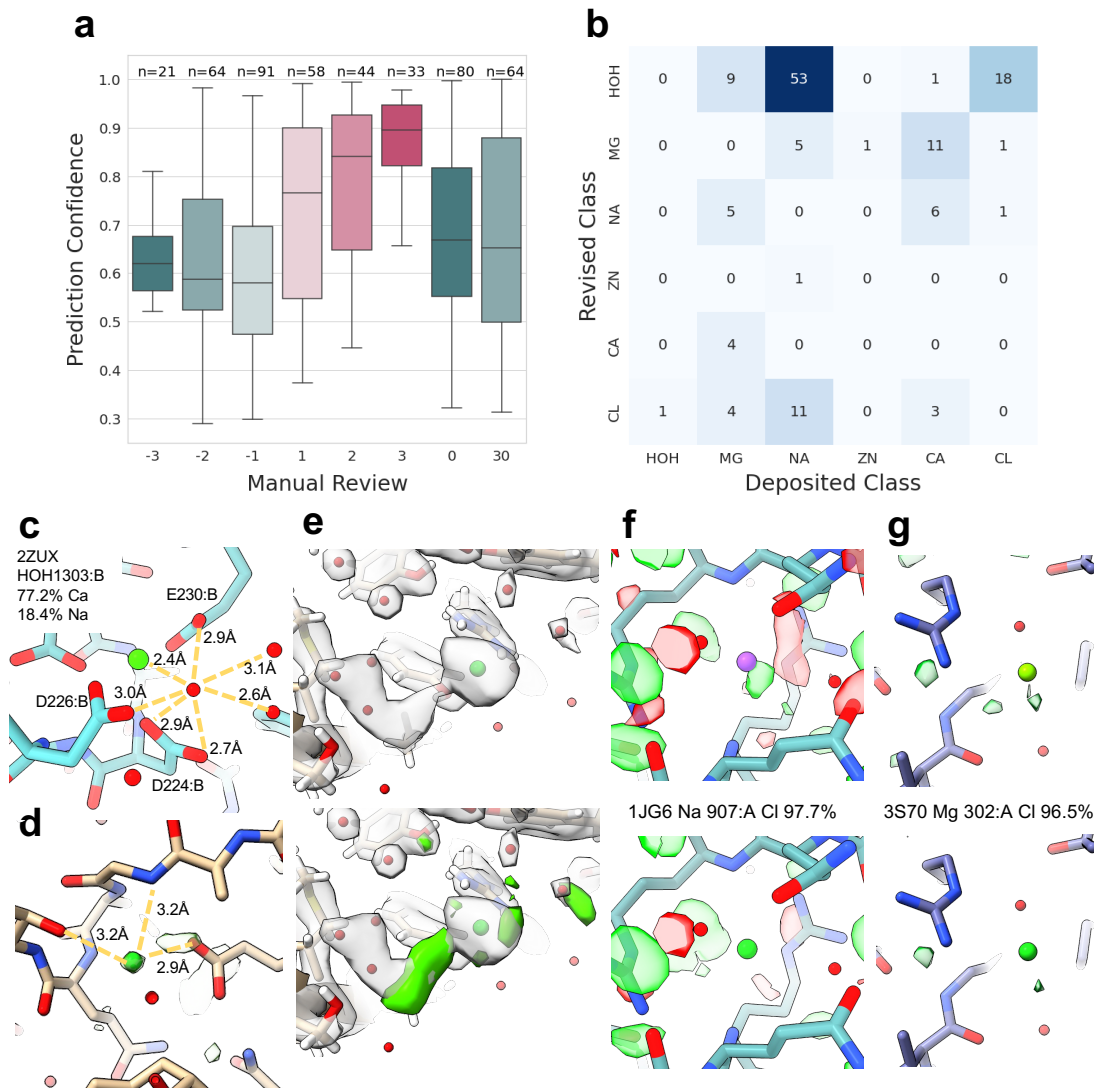
Supplemental Figure 2.1. MIC dataset preparation and exploration.

a, Class distribution used for training and testing MIC models. Extended-set only ions are marked with *. **b**, Distribution of the resolutions of the structures each site was extracted from, colored by the deposited label. Potassium, chloride, and bromide (and matching waters) were restricted to $<3\text{\AA}$, all other classes were restricted to $<2\text{\AA}$. **c-f**, Heatmaps of the accuracy of an SVC trained directly on fingerprints generated with a given radius and shell number for the evaluated fingerprint types. **c**, Prune/EIFP. **d**, Non-prune/EIFP. **e** Prune/FIFP. **f**, Non-prune/FIFP.



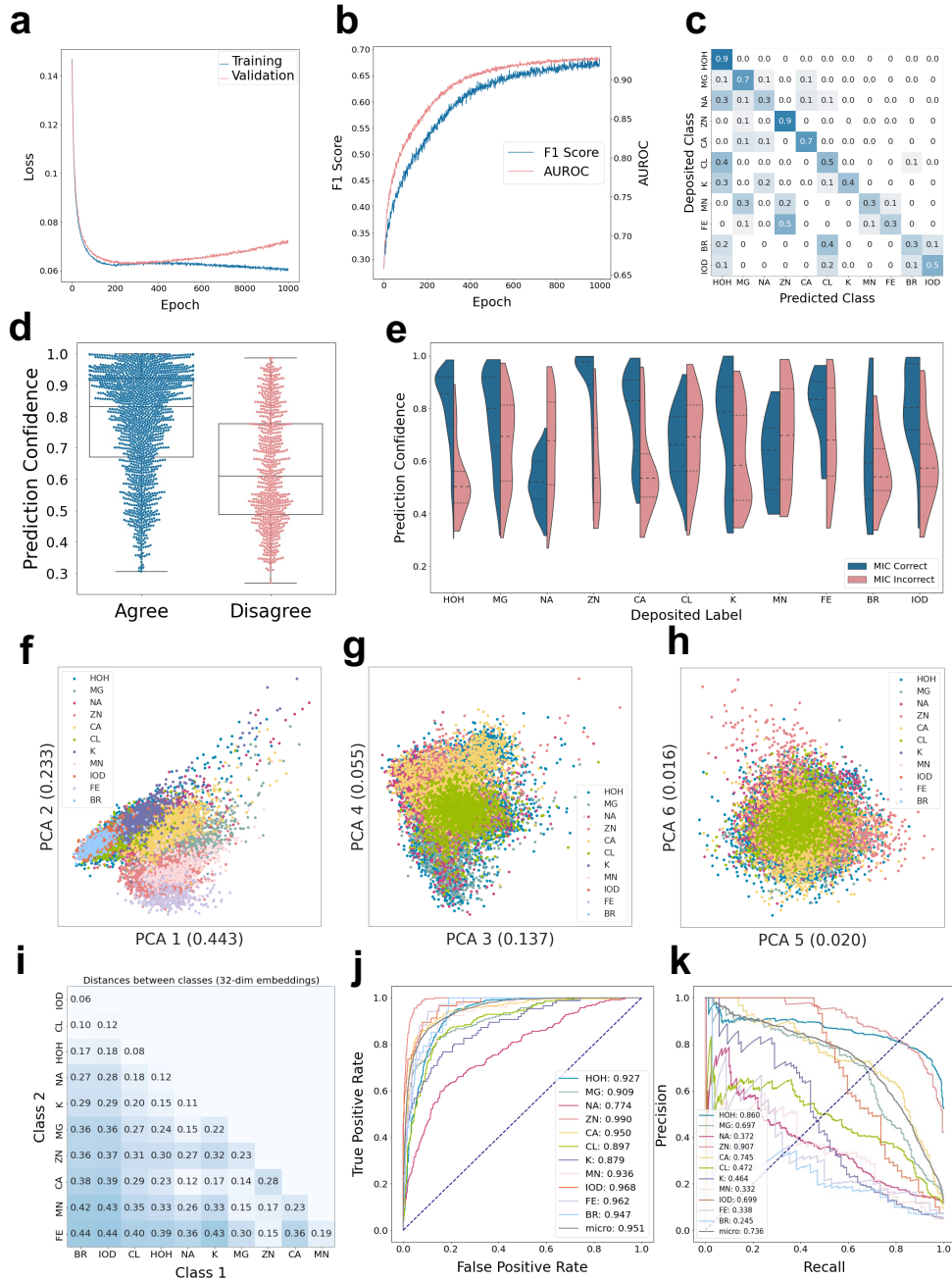
Supplemental Figure 2.2. MIC prevalent-ion set additional results.

a, Training and validation loss. **b**, F1-score and micro-AUROC of SVC trained on intermediate training set embeddings and labels, evaluated on validation set. **c**, Average distance in 32-dim latent embedding spaces between classes as confirmation of the trends observed in the Fig. 2 Low-dimensional visualizations. **d-f**, PCA plots for of the first 6 dimensions by variance explained, shown alongside the axis labels. **g**, Confidence of agreeing vs disagreeing predictions and deposited labels of prevalent-ion test set, split by class. **h**, ROC curves of individual classes and micro-average of MIC predictions on the prevalent-ion test set. **i**, PRCs of individual classes and micro-average of MIC predictions on the prevalent-ion test set. **j-l**, Sites used to perform feature attribution analysis. **j**, 4L9P:B:ZN:601. **k**, 4OKE:A:MG:202. **l**, 3BMV:CA:A:684.



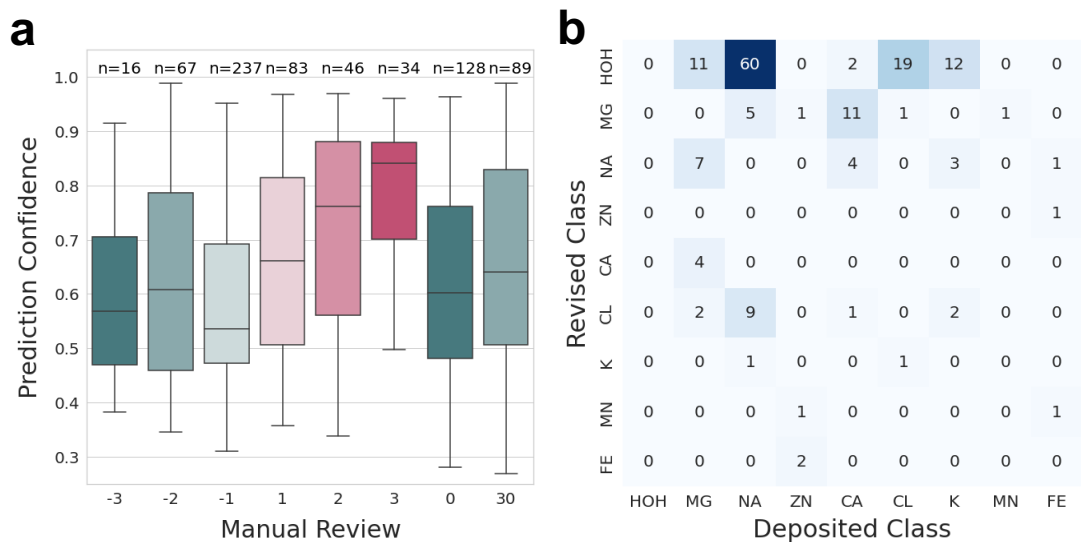
Supplemental Figure 2.3. Manual review of test set discrepant sites.

a, Confidence of each discrepant example vs the manual review value. Manual review values ranged from -3, indicating an incorrect MIC prediction to 3, indicating a correct MIC prediction. 0 marks examples for which it was unclear which of the two labels was correct, and 30 marks sites with other issues preventing proper prediction. **b**, Heatmap of corrected sites; deposited label is shown on the x-axis and the revised MIC-predicted label is shown on the y-axis. Each cell is labeled with the count of sites. **c**, Example of common MIC pathology of predicting coordinating waters as cations. **d-e**, Examples of structures that received a 0 and a 30 during manual review. (d) 4DWD. **e**, 4JO5. **f-g**, Re-refinement of difference maps contoured at $\pm 3\sigma$ with deposited (top) and MIC-predicted label (bottom). **f**, 1J68. **g**, 3S70.



Supplemental Figure 2.4. MIC extended set additional results.

a, Training and validation loss. **b**, AUROC and F1-score during training. **c**, Confusion matrix of test set deposited and predicted label. **d**, Agreeing vs discrepant prediction confidence, pre-revision. **e**, Agreeing vs discrepant prediction confidence by class, pre-revision. **f-h**, PCA plots of learned latent embeddings, extended set. **i**, Distance between classes in embedding space. **j**, ROC curves of individual classes and micro-average of MIC predictions, extended set. **k**, PRCs of individual classes and micro-average of MIC predictions, extended set.



Supplemental Figure 2.5. Manual review of discrepant sites from extended test set.

a, Confidence of each discrepant example vs the manual review value. Manual review values ranged from -3, indicating an incorrect MIC prediction to 3, indicating a correct MIC prediction. 0 marks examples for which it was unclear which of the two labels was correct, and 30 marks sites with other issues preventing proper prediction. **b**, Heatmap of corrected sites; deposited label is shown on the x-axis and the revised MIC-predicted label is shown on the y-axis. Each cell is labeled with the count of sites.

Supplemental Table 2.1. Initial features for each atom by type and fingerprint type.

Type	Descriptors	Length
Ions and waters	[0, 0, 0, 0, 0, 0, 0]	7
Extended Interaction Fingerprint (EIFP)	[# Heavy Atom Neighbors, Valence - # Hydrogen Neighbors, Atomic number, Isotope number, Formal Charge, # Hydrogen Neighbors, Is_In_Ring]	7
Functional Interaction Fingerprint (FIFP)	Aromatic, Acceptor, Donor, Hydrophobe, Hydrophobic, Negative, Positive, Negatively ionizable, Positively ionizable, Halogen donor, Metal, Lumped hydrophobe, Weak donor, Weak acceptor, Electrophile, Nucleophile, Chalcogen donor, Amide, Atom	≤ 19

Supplemental Table 2.2. Hyperparameters explored and final values.

Hyperparameter	Options	Final
hidden_layers	[1,4]	1
n_neurons	1000,2000,3000,4000	4000
embedding_dim	2, 8, 16, 32	32
learning_rate	[1e-8, 1e-4]	1e-7
dropout	[0.05, 0.4]	0.35
weight_decay	[0, 0.1]	0.005
loss_margin	[0.01, 0.1]	0.06
miner_margin	[0.1, 0.3]	0.2

Supplemental Table 2.3. P-values and number of sites for all statistical analyses.

Statistically significant comparisons are indicated.

Comparison	Number of examples	p-value	t-statistic	Degrees of Freedom	Effect Size (Cohen's d)
Confidence of agreeing vs disagreeing predictions, initial*	2037	4.802e-83	20.227	2035	1.00
Confidence of agreeing vs disagreeing predictions, revised*	1893	8.596e-74	18.997	1891	2.46
Confidence of agreeing vs disagreeing water predictions, revised*	975	8.166e-49	15.540	973	1.42
Confidence of agreeing vs disagreeing magnesium predictions, revised*	245	3.450e-10	6.547	243	1.17
Confidence of agreeing vs disagreeing sodium predictions, revised	151	0.178	-1.353	149	-0.22
Confidence of agreeing vs disagreeing zinc predictions, revised*	193	5.952e-17	9.208	191	2.23
Confidence of agreeing vs disagreeing calcium predictions, revised*	165	1.536e-06	4.990	163	1.35
Confidence of agreeing vs disagreeing chlorine predictions, revised*	164	3.197e-07	5.333	162	1.01

CHAPTER 3: MACHINE LEARNING-GUIDED VISUAL INSPECTION OF MOLECULAR DOCKING RESULTS WITH AUTOPARTY

Abstract

Human inspection of potential drug compounds is a crucial step in the virtual drug screening pipeline. However, there is a pressing need to accelerate this process as the number of molecules humans can realistically be expected to examine is extremely limited relative to the scale of virtual screens. Furthermore, medicinal chemists are often inconsistent in evaluating different poses, and there remains no standard way of recording annotations. We propose Autoparty, a tool to address both of these potential problems. Autoparty builds on recent work in active learning applications for drug discovery to facilitate human-in-the-loop training of models that recapitulate human intuition. We use a variety of uncertainty quantification techniques to present the user with the most informative example for model training, limiting the amount of necessary labels. Additionally, these annotations are saved into a database for persistence that can be later exported for further quantification of these medicinal chemical rules.

Introduction

Computer-aided drug design (CADD) techniques are often utilized in the early stages of drug discovery to identify lead molecules with high predicted binding affinity against a specific target.^{347,348} Virtual screening (VS) programs such as DOCK^{127,128} and Glide¹²⁹ computationally dock and score libraries of compounds, producing ranked lists of molecules and associated predicted complex structures. The top-scoring molecules are typically clustered by chemical structure and often filtered against known binders to discover novel chemotypes. Additional filtering steps based on interaction fingerprints or alternative scoring methods may be applied

during this step as well. The remaining best-scoring cluster heads are visually inspected (“hit-picked”) by expert medicinal chemists to evaluate the quality of the generated poses before selecting final compounds for experimental testing (**Figure 3.1**).^{349,350}

This current VS pipeline suffers from several bottlenecks where potential binders may be filtered out, leading to false negatives. The first is docking itself, wherein good candidate molecules either score poorly by the running function or are not included in the screen at all. The latter problem is only growing more relevant as the size of available molecular libraries increases into the order of billions and beyond.^{52,53,351} Several ultra-large screens have evaluated multiple hundreds of millions to billions of molecules,^{316,352–355} but docking the total available chemical space remains intractable.³⁵⁶ Efforts to accelerate docking programs for brute force screening rely on parallelizable operations performed with large supercomputing clusters and are thus inaccessible to the wider research community.^{357–359}

One alternative approach to address this issue is active learning (AL).^{360,361} Active learning is a semi-supervised machine learning technique for training models from a minimal amount of labeled data, particularly when labels are expensive. In AL, the model is first trained on a random subset of the data. The model then predicts labels for and selects examples from the remaining unlabeled data for submission to the “oracle” to obtain true labels. These examples and labels are then added to the training set for the next iteration (**Figure 3.1**). Intuitively, this approach can identify the most informative examples at each iteration, allowing for an accurate model with minimal data. Often, the “oracle” is a human, hence why this is sometimes referred to as human-in-the-loop training. In virtual screening, AL has been used to determine which regions of chemical space to dock from large libraries. The oracle is the docking program itself, and the model is trained to select compounds to build and dock. Yang *et al.*¹⁷⁵ demonstrated the success of this approach

across multiple receptors both retrospectively and prospectively for UCSF DOCK. Graff *et al.*¹⁷⁶ applied AL to screens from small libraries (Enamine 10K) up to ultra-large libraries (>100M). They showed improved recovery of top hits over random acquisition for both Glide and DOCK scores. Thompson *et al.*¹⁷⁷ explored various AL parameters, including acquisition function, number of molecules sampled at each iteration, ML architecture, and initial sampling strategy, for predicting the output of free energy perturbation (FEP) calculations. Their best-performing models recovered 75% of the top 100 molecules from running FEP on 6% of the dataset, representing a substantial decrease in the required number of calculations. These studies suggest that AL is a useful paradigm for learning complex scoring functions.

Another less studied rate-limiting step is the final human inspection stage, wherein medicinal chemists manually evaluate predicted structures. A review of drug discovery protocols³⁶² found that visual inspection of docked poses was used in 50% of the 250 publications surveyed, often as the final step in compound prioritization before experimental testing. It has been suggested that any sufficiently experienced human could outperform existing scoring functions for compound selection.³⁶³ Indeed, previous studies have found molecules chosen by expert medicinal chemists achieve greater potencies than those selected by scoring functions alone.³¹⁶ However, there are limits to the number of poses that a given scientist can realistically examine. This is especially troubling given recent reports of a hit-rate plateau indicating a significant amount of false negatives further down in the screening results.^{316,352} Moreover, despite efforts to codify rules for visual inspection^{364,365} there remains no high-quality database of expert annotations for learners to use as a reference. This problem is further complicated by the fact that even expert medicinal chemists often disagree with one another.³⁶⁶

In this application note, we introduce Autoparty, a tool for AI-assisted accelerated hit-picking on molecular docking results. Autoparty is a browser-based tool for visualizing the protein-ligand interface and intermolecular interactions. The user assigns grades to each given molecule, which are stored in a database and available for export. Autoparty allows for active learning with the user as the human for human-in-the-loop training, displaying the poses that are most useful to the user for review. This trained model can then be applied further down in the docking screen to potentially rescue false negatives that would not be found by standard inspection procedure. We hope this tool will bring together recent ML advancements and medicinal chemical expertise, facilitating pose evaluation and potentially allowing for the recovery of false negatives further down in the screening hit list.

Results

Autoparty: A tool for automated human-in-the-loop molecule inspection

Autoparty is a Python-based containerized application developed to assist scientists in analyzing the results of virtual docking screens using active deep learning. The user uploads the protein target and docked ligands to visualize their structure and relevant intermolecular interactions. These molecules can then be graded by the user. These assigned grades are saved to an SQL database to maintain a robust record of structures and human annotations. During this human-in-the-loop training, a machine learning (ML) model is trained to both predict these annotations for new molecules and to determine which molecules are the most informative for the user to see. This process can continue until the user finishes grading molecules, after which the full list of grades and model predictions can be exported for further review (**Figure 3.2**). In addition, the trained models can be extracted and applied to new molecules not present in the initially uploaded screen. In the next few paragraphs, we describe each of these steps and their associated configuration

options in greater detail. A full list of Autoparty settings with their defaults is provided in **Supplemental Table 3.1**.

Interaction Calculation and Representation Generation

When a screen is uploaded to Autoparty, the LUNA toolkit⁶⁶ calculates intermolecular interactions between all input molecules and the protein structure. This process consists of two steps: First, the biochemical properties of both the ligand and protein atoms at the interface are determined using openbabel³⁶⁷ or RDKit.⁵⁷ This includes the determination of atomic groups, such as aromatic rings, that are considered to participate as a single interacting group. Groups are visualized in Autoparty as white spheres at the group's center of mass. Second, interactions are then defined as pairs of atoms or atomic groups in proximity that meet certain geometric and chemical criteria. Common interaction types include Van der Waals interactions, hydrogen bonds, hydrophobic interactions, halogen bonds, and others. These interactions are shown on the complex as dashed cylinders. All available interaction types and associated display colors are provided in **Supplemental Table 3.3**.

For the input to the backend ML model, we chose to use interface fingerprints (IFPs). In contrast to molecular fingerprints that encode the chemical topology of molecules, IFPs capture the geometric and biophysical properties of the full protein-ligand interface. They have previously been used to predict docking scores and binding affinities^{66,67} and offer a way to address the known problem of activity cliffs in ligand-based machine learning scoring functions.³⁶⁸ IFPs have been shown to perform comparably to more complex message-passing neural networks at small dataset sizes.¹⁷⁶

To convert the calculated interaction graph into a fixed-size vector representation of the protein-ligand interface, the ligand and interactions are divided into multiple three-dimensional

spherical shells centered on each ligand atom. The first shell at radius 0 contains the initial atomic identifiers for each ligand atom. These invariants can either be explicit, including the number of non-hydrogen neighbors, total bond order, atomic number, atomic mass, atomic charges, and number of attached hydrogens) or functional/pharmacophoric (aromatic, acceptor, donor, hydrophobic, etc), referred to as EIFP and FIFP, respectively. The subsequent shells contain interactions within their boundaries. Each shell is passed through a hash function to result in a series of "on" indices, which are collected into the final fingerprint following the standard extended-connectivity fingerprint protocol introduced by Rogers and Hahn.⁶¹ This calculation occurs before the molecule is saved to the database, and the user can export the calculated IFPs. By default, Autoparty uses EIFP count fingerprints calculated with 2 shells of radius approximately 6Å and folded to a length of 4096, as these were found to be the optimal settings for DOCK score prediction in previous studies.⁶⁶ A full summary of this fingerprint generation protocol and initial chemical features is available from Fassio *et al.*⁶⁶

Model Architecture and Training

In standard AL workflows, it is common to employ model architectures that provide both a predicted label and a measure of uncertainty quantification (UQ) in this prediction. These outputs are used in various acquisition strategies, such as greedy and least-confidence sampling, to determine the examples to include in the next AL iteration. Uncertainty quantification methods in molecular property prediction have been previously explored, but the performance of these approaches remains strongly dependent on the dataset itself.^{369,370} By default, Autoparty uses ensemble-based approaches that train a committee of individual models. Uncertainty is then measured as the variance across predictions from individual committee members.³⁷¹ The default

model trained during a hitpicking party consists of three separate and unique models that together make up the ensemble. Each member is a deep neural network that consists of an input layer followed by two hidden layers of 1024 neurons each. The size of the output layer is equal to the number of unique possible grades as determined by the user. Each ensemble member generates initial unique training data by sampling with replacement from the full set of available graded molecules. This approach provides data augmentation through this sampling while adding limited overhead as the training time for each individual model remains short, given the dataset size. Additionally, we found that ensemble UQ provided high accuracy and good calibration in initial studies (**Supplemental Figure 3.1**).

Due to the known variability of different UQ methods across datasets, Autoparty includes other uncertainty estimation architectures as options for the user. Specifically, Autoparty allows for dropout-based uncertainty³⁷² in which a single model repeatedly predicts a label for a single example with random dropout. Similar to ensemble methods, this results in multiple predictions for the same example that are averaged for the final prediction. Uncertainty is the variance in these predictions. Autoparty also includes the option of distance-based uncertainty, where the distance between a new example and its closest neighbors in the model training set is used as a proxy for uncertainty.^{373,374}

Grades differ from typical classification labels in that they are ordinal, meaning that the potential output classes have an ordered relationship; an A is closer to a B than an F. For ideal training, it is necessary to include this property in the formulation of output classes and the loss function of the model. Some approaches simply treat ordinal labels as a regression task and bin the predictions into different classes, though this does require additional tuning of these cutoff values. We used a cumulative one-encoding strategy wherein each label was adjusted to a vector

of 1s followed by 0s. The number of 1s indicated the value of the grade (Additional detail provided in **Supplemental Methods**). Models are trained with binary cross-entropy loss. Evaluating this on a small dataset of MM/GBSA scores of complexes from a virtual screen against the Dopamine D4 receptor showed successful generalization between training and testing along with a higher proportion of “close” mispredictions, or those that were a single grade away from the true value (**Supplemental Figure 3.2**). Autoparty also allows for nominal class labels in which case the output classes are simply one-hot encoded.

One of the most critical design choices in AL is the selection of the acquisition function. These are the criteria by which examples are selected at each iteration for the oracle. Historically, AL has focused on including molecules with the greatest predicted uncertainty under the assumption that these would be maximally informative for the model.³⁷⁵⁻³⁷⁷ Previous studies using AL for virtual screening have instead used greedy acquisition strategies that select the examples with the best-predicted score by the model regardless of confidence. This approach successfully found the true top-scoring molecules from large libraries of compounds, a more useful metric for their purposes than model accuracy alone.^{175,176} Gusev *et al.* combined similarity clustering with greedy cluster-head selection to train a relative binding free energy prediction model.³⁷⁸ By contrast, Thompson *et al.* found that the number of top compounds recovered was surprisingly indifferent to acquisition strategy, though all strategies did outperform random acquisition.¹⁷⁷ We provide greedy, maximum uncertainty, and random orderings as options to the user with maximum uncertainty as the default value.

Throwing a Hit-picking Party

Uploading a screen

The first step to using Autoparty is uploading the result of a virtual docking screen (**Figure 3.2, Figure 3.3**). To begin, the user must provide both the protein and docked molecule files. Additionally, the user is able to indicate a property within the initial molecules file by which the compounds are ordered for annotation before any active learning component. Upon uploading, molecules and their three-dimensional coordinates are read from the provided file and sent in batches to calculate intermolecular interactions and IFPs. Hitpicking can begin as soon as the interactions for the first molecule are calculated.

Getting the party started

Once the screen has started uploading and calculating interactions, the hit-picking session can begin. The user can either start a new party by selecting the uploaded screen or resume an existing party to recover existing grades, models, and predictions from a prior session. The available settings and options are discussed in **Supplemental Table 3.1** and **Supplemental Table 3.2**.

Annotating Molecules

Upon beginning annotation, molecules are sorted by the field provided by the user during screen upload (**Figure 3.3**). The full list of molecules is displayed on the right. The available grades appear at the bottom of the screen. In 'Annotation' mode, molecules that have previously been assigned grades are hidden to allowing the user to view only new molecules and sort them by either provided score or, if a model has been trained, predicted grade or uncertainty. In 'Review' mode, the user can view molecules that they have already seen, offering the potential for regrading if

their mind has changed. In this mode, there is an additional sorting method known as 'Disagreement' which shows the molecules with high certainty for which the predicted grade differs from the user's previously assigned grade for that molecule.

Uploading existing annotations (optional)

Autoparty allows for uploading previous annotations for screened molecules. Users can upload a CSV file containing molecule names and grade columns. The molecule names are then used to match uploaded molecules with new grades and update the database. These new grades count towards the trigger for required total grades to begin model training.

Training a Model

After a set number of annotations have been submitted (default 100), the user can train a model and begin the human-in-the-loop active learning process. On the backend, all of the fingerprints and corresponding annotations from the current party are recovered from the database to use as a training set, and a model is trained to predict these labels (**Figure 3.2**). The trained model is then applied to all remaining molecules, predicting 1) the annotation (grade) for that molecule and 2) the model's confidence in that predicted grade. Upon completion, the molecules are reordered based on the selected acquisition function. This process continues until stopped by the user. The loss curves and model history are updated in real-time and visible in the Training dashboard.

Autoparty Implementation

Autoparty is written primarily in Python for the backend service, which handles processing the provided input files, calculating protein-ligand intermolecular interactions, reading and writing to the SQL database, training models, generating model predictions, and selecting examples to

service to the frontend. The full SQL database schema is shown in **Supplemental Figure 3.3**. All ML models were implemented in the Pytorch deep learning library.³⁴⁴ The frontend interface that the user interacts with is implemented in JavaScript, HTML, and CSS. The molecular visualization panel uses 3Dmol.js for on-page structure viewing and displaying calculated interactions.³⁷⁹ The Celery queue³⁸⁰ and Redis³⁸¹ broker handle asynchronous tasks such as updating the database, training models, and creating output prediction summaries. The application was containerized for ease of portability across machines with Singularity (now Apptainer).^{382,383} The container and all application code are available for download at <https://github.com/keiserlab/autoparty>.

Acknowledgments

We would like to thank Alexandre Fassio for his work on the LUNA package and assistance in interaction visualization code and John Gallias for his useful discussions on container lifecycles. We would also like to thank Magdalena Korczynska and Meihua Tu for their suggestions regarding potential features to add to the application, and Helena Qi for her help installing and testing the software.

Author Contributions

L.S. designed and wrote the Autoparty software with input from F.L., A.M. and M.J.K. D.M. performed initial uncertainty quantification studies. L.S. wrote the manuscript with input from D.M. and M.J.K. M.J.K. supervised the project.

Figures

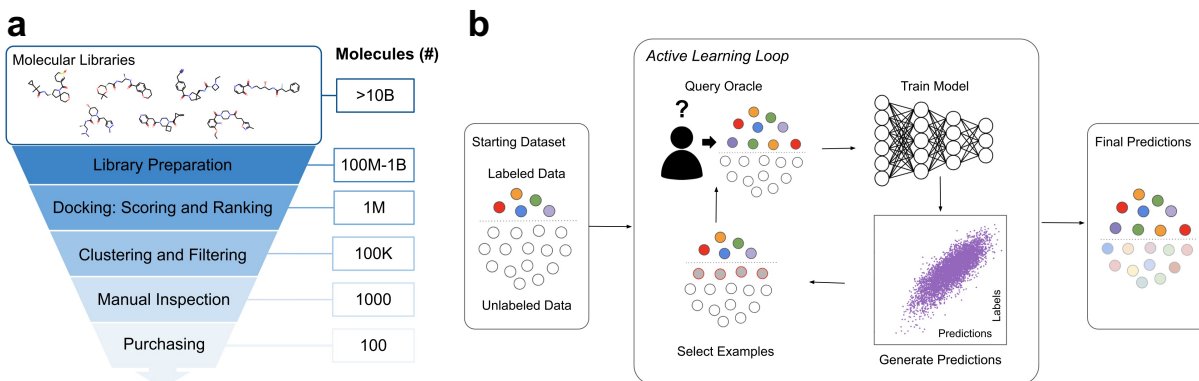


Figure 3.1. Conceptual Overview of the Virtual Screening Pipeline and Active Learning

a, The drug-discovery “funnel” shows standard steps in the process along with the number of molecules remaining at each step (approximate). This highlights the need for tools to address these bottlenecks and ensure promising molecules make it through to experimental testing. **b**, Human-in-the-loop active learning training paradigm. The starting dataset consists of a large pool of data with limited or no true available labels. A random subset of that data is presented to the oracle (human) for labeling, after which a model is trained on this initial data. The model is used to predict labels and confidence for the remaining data. Selected examples are fed to the oracle. This process repeats until we reach a quota of labels or achieve the desired accuracy.

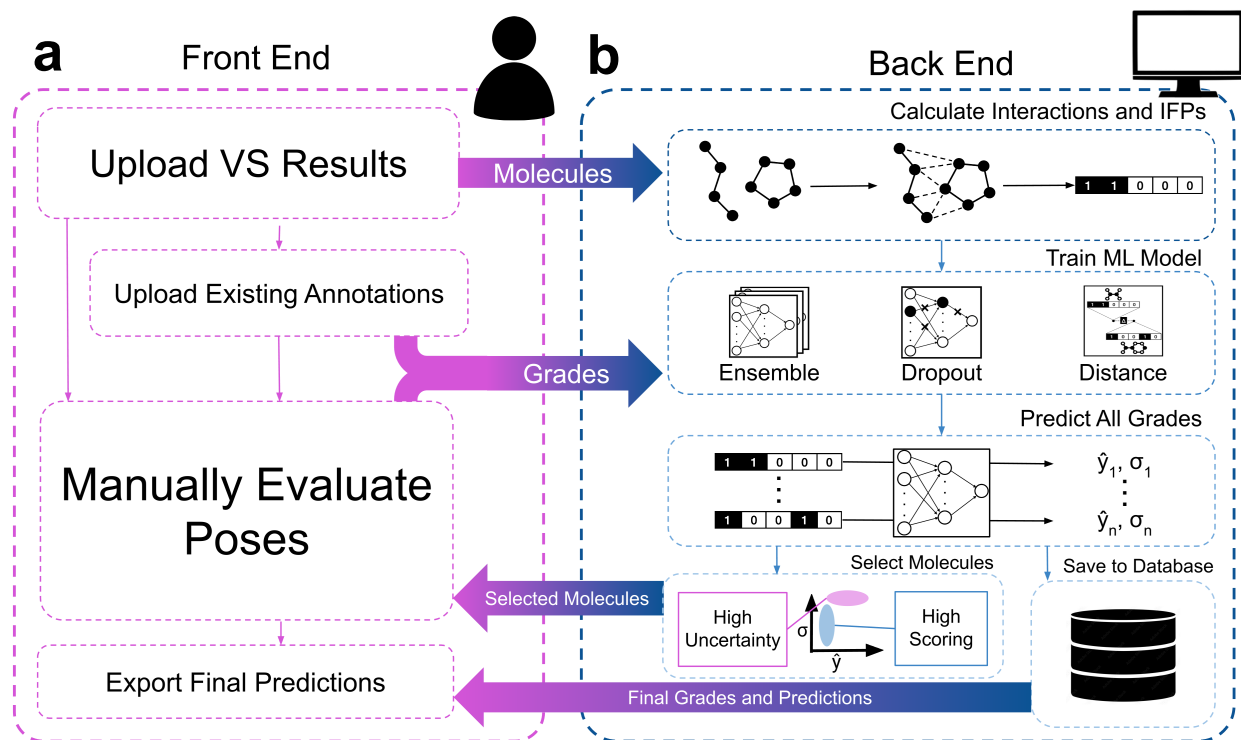


Figure 3.2. Schematic showing Autoparty workflow and user interaction.

a, Autoparty functions that are performed by the user. These include uploading the initial screening results and any potential preexisting annotations and grading new molecules. The user can also recover all existing grades and predictions from the database. **b**, Autoparty functions performed automatically on the back end to assist with hit picking. For each complex, the interactions and LUNA IFPs are calculated upon upload. The back end also handles saving the grades provided by the user, training models to try to predict human labels, and ordering the compounds to show the user the most informative examples.

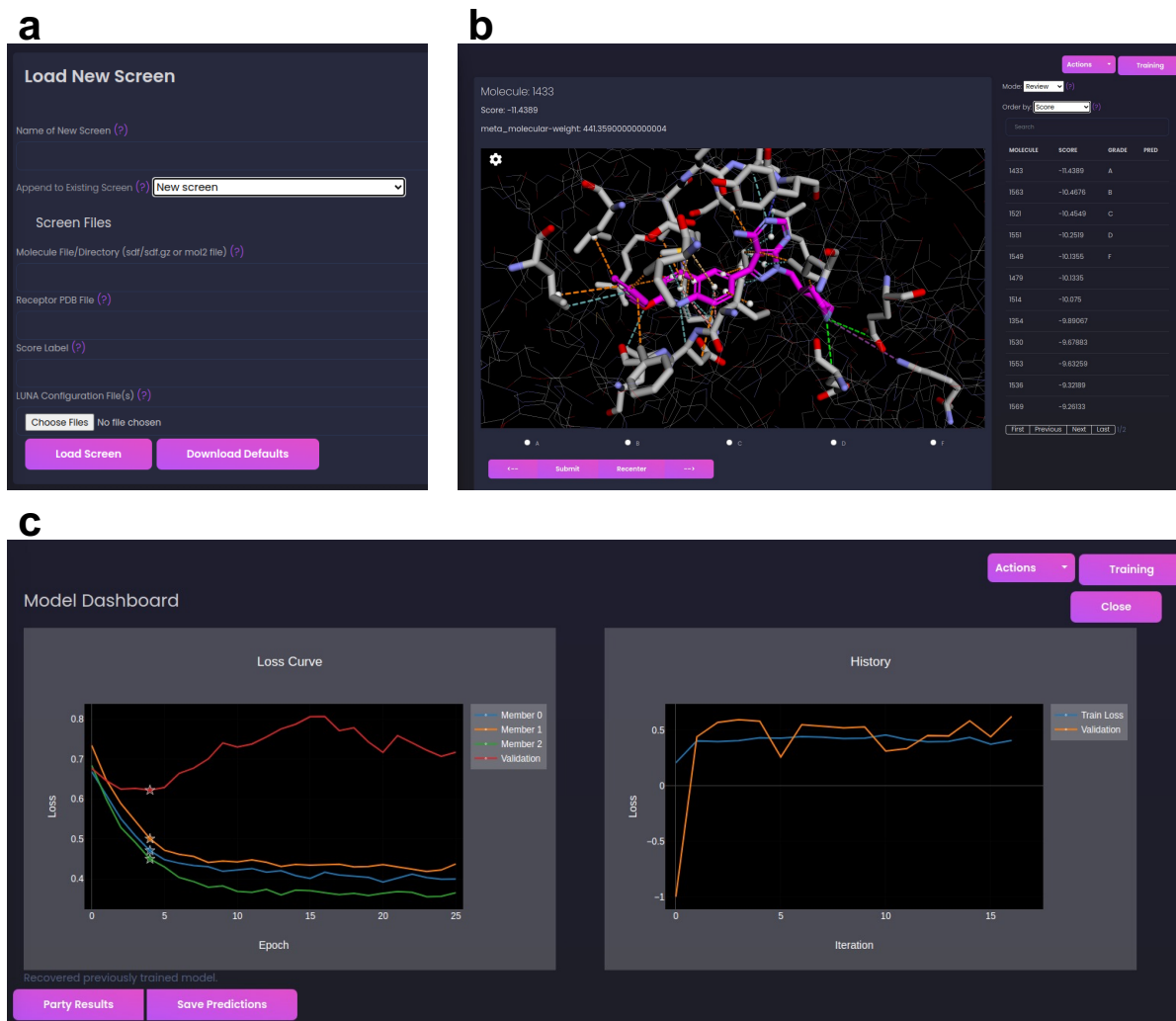


Figure 3.3. Autoparty Web Interface

a, The screen upload interface. The user must provide a protein file and a separate molecules file. They also have the option of providing an attribute to sort molecules for initial annotation. This screen is where the user may also provide specific LUNA configurations, otherwise the default options are used. **b**, Hit picking interface for human-in-the-loop training. This screen shows the current molecule being reviewed along with the calculated interactions. Potential grades are shown along the bottom of the screen. The top right shows current mode (Annotate vs Review) and current sorting method (Score, Uncertainty, Disagreement, Random) along with the options to upload a screen or view the model training panel. **c**. Model Training Panel. The left half shows the individual loss curve of each committee member along with the overall validation loss during training. The epoch with the best loss value is indicated. The right half shows the training and loss over time throughout multiple iterations of model training.

Supplemental Methods

Uncertainty Quantification Analysis with AA2AR

Dataset and Model Training

To explore uncertainty quantification methods for DOCK score prediction from interface fingerprints, we used a dataset of 311,705 complexes from a screen of molecules against the adenosine A2A receptor (AA2AR) structure³⁸⁴ using the DUD-EZ docking benchmark³⁸⁵ set starting files. Each complex was represented by the LUNA⁶⁶ extended interaction fingerprint with default settings. Data was split randomly into training/validation/testing with an 80:10:10 split and kept consistent across all trials. All models trained consisted of four hidden layers of 5000, 4500, 4050, and 3645 neurons respectively, 10^{-4} weight decay, and 20% dropout. Training was performed with MSE loss with an additional KL divergence term for the Bayesian networks.

Uncertainty Quantification Methods

Ensemble: Ensemble uncertainty was estimated by training a committee of models, each with the same architecture. Starting data were sampled from the total available training data to introduce additional variability between the ensemble members and as a form of data augmentation. The prediction and uncertainty for a single example were the mean and variance of all committee member predictions, respectively. We evaluated the effect of committee size and initial sampling strategy.

Dropout: Dropout uncertainty was calculated by predicting the score for a single example multiple times with non-zero random dropout, resulting in multiple outputs for an individual example. The prediction and uncertainty for a single example were the mean and variance of all predictions. We evaluated the number of replicates and the proportion of dropped neurons.

*Bayesian:*³⁸⁶ In contrast to standard point-wise neural networks with deterministic outputs, Bayesian networks learn probability distributions over their weights that are sampled when predicting the label for a new example. Similar to dropout uncertainty, predicting multiple times for a single examples results in differing outputs, with the final prediction and uncertainty calculated as the mean and variance of all predictions.

Feature Similarity: Feature similarity was calculated as the average Tanimoto coefficient³⁸⁷ between a given fingerprint and the closest k neighbors in the training set. This was used as a proxy for model confidence in the corresponding prediction.

Metrics

For architecture evaluation, we focused on two metrics: the regression accuracy of the model (R^2 correlation coefficient between true DOCK scores and model predictions) and a novel metric, the calibration coefficient (CC). A well-calibrated model shows a correlation between the residuals and the predicted uncertainty, commonly visualized using a calibration curve that plots the expected and observed accuracy against the response rate.³⁸⁸ We extend this concept further by measuring calibration success as a function of the difference in area between these two curves. The calibration curve of an ideal model (1:1 residual/uncertainty ranking) provides an upper bound for this area, while random ordering provides a lower bound (**Supplemental Figure 3.1**). The plot for trained models falls between these two curves. The relative area of the model against the ideal calibration then provides a quantitative measure of true calibration. Formally, the area under the curve for a given ordering is defined as:

$$AUC_{order} = \int_{RR_{order}=0}^1 R^2 d(RR)$$

For a given ranking of examples, where RR is the response rate for a given ordering of the test set, sorted either by residual (ideal), uncertainty (observed), or random. The calibration coefficient formula is then:

$$CC = \frac{AUC_{observed} - AUC_{baseline}}{AUC_{ideal} - AUC_{baseline}}$$

where the baseline is the average over 5 random orderings. CC provides additional information compared to rank-order correlation between residuals and uncertainty alone, primarily regarding the range of data for which the model is well or poorly calibrated (**Supplemental Figure 3.2**).

Results

For each architecture, we calculated both regression accuracy and calibration coefficients for various sets of hyperparameters. The full table of conditions and metrics evaluated is available in **Supplemental Table 3.4**. All architectures showed improved calibration over random ordering. Repeated sampling increased calibration and accuracy for both Bayesian neural networks and dropout-based uncertainty. Distance-based uncertainty showed the best calibration at higher confidence but was outperformed by ensemble-based sampling for lower confidence examples. Overall, ensemble uncertainty with five members and bootstrap sampling resulted in the highest observed calibration coefficient and accuracy, particularly among the least confident examples. This is the Autoparty default uncertainty quantification method.

Ordinal Classification Testing with Dopamine D4

Dataset and Model Training

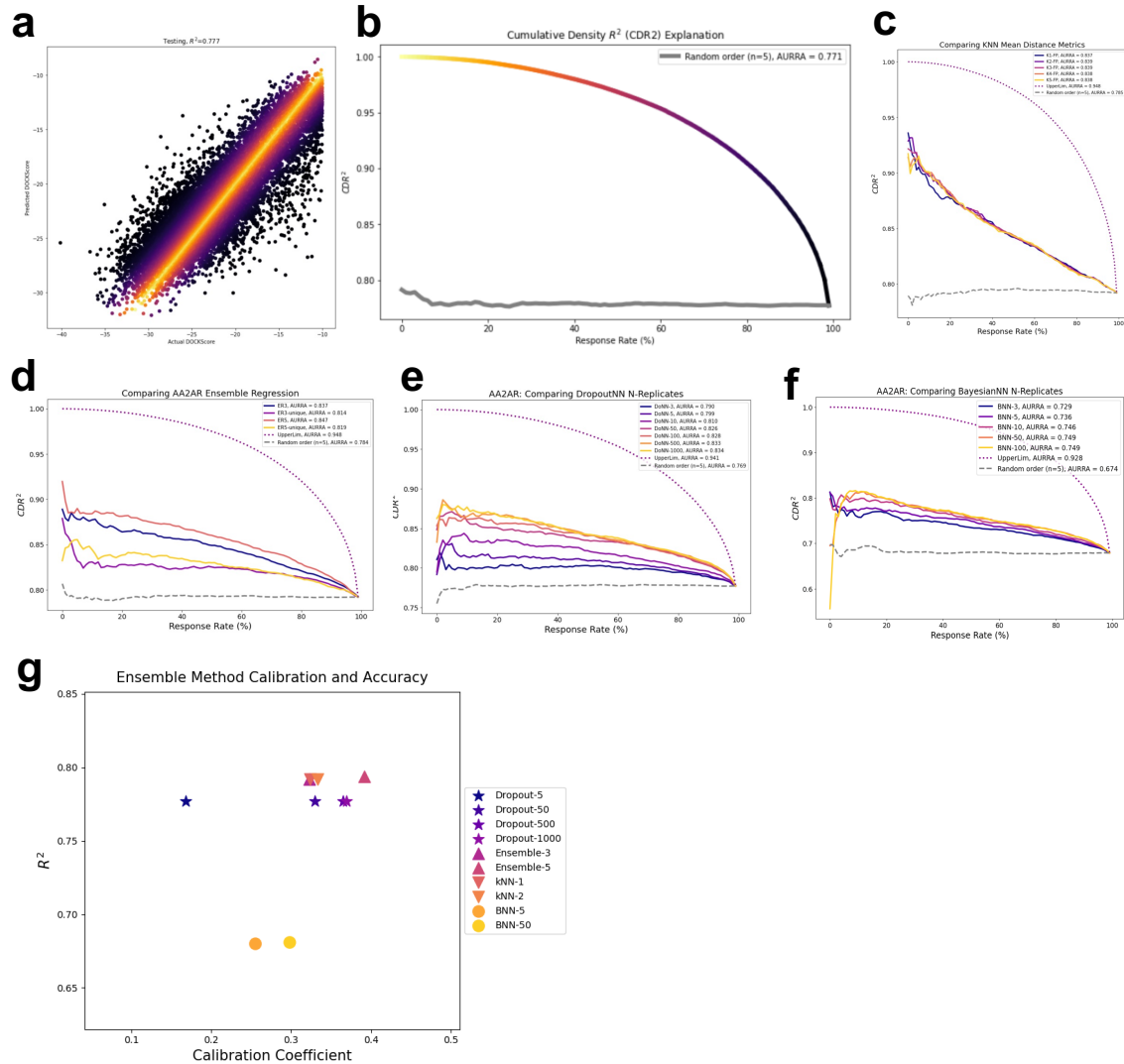
We used a dataset of poses and MM/GBSA scores from a screen against the dopamine D4 receptor³¹⁶ for initial ordinal classification experiments. To convert these regression labels to ordinal, examples were binned by score into grades A through F with an approximately equal number of complexes in each bin. The ordinal labels were encoded through cumulative one-hot encoding to preserve the ordered relationship between grades (**Supplemental Figure 3.2**). Based on previous work, we trained an ensemble of three neural networks with bootstrap sampling on a subset of randomly selected examples ($n = 2000$ and $n = 50,000$) to evaluate the ability of the model to predict ordinal labels and model calibration. Testing was performed on a consistent held out set of 10,000 molecules. Each ensemble member had two hidden layers of 4,096 neurons each, and training was performed with binary cross-entropy loss on the encoded labels.

Results

The full results for both the 2K and 50K models are shown in (**Supplemental Figure 3.2**). Overall, both models were largely successful at learning to predict the assigned grades but did show some overfitting, especially at lower data regimes. The average root mean square error (RMSE) between test examples for the 2K model was 0.88, indicating that the predicted grade was rarely more than a single class away. This value dropped to 0.714 with the addition of more data. The main difference in performance between the two models was at the extremes, with the 2K model underpredicting both A and F. However, the majority of the examples belonging to those classes were predicted as B (956) or D (905), respectively, indicating that the model is successfully identifying better and worse poses by our toy grades. Even at 2,000 grades, the ensemble model

was able to recover 61% of the true 'As' in the test dataset. Crucially, the least confidence examples were more likely to be incorrect for both cases, following the expected properties of well-calibrated models (**Supplemental Figure 3.2**). We used this same ordinality prediction method for the Autoparty implementation.

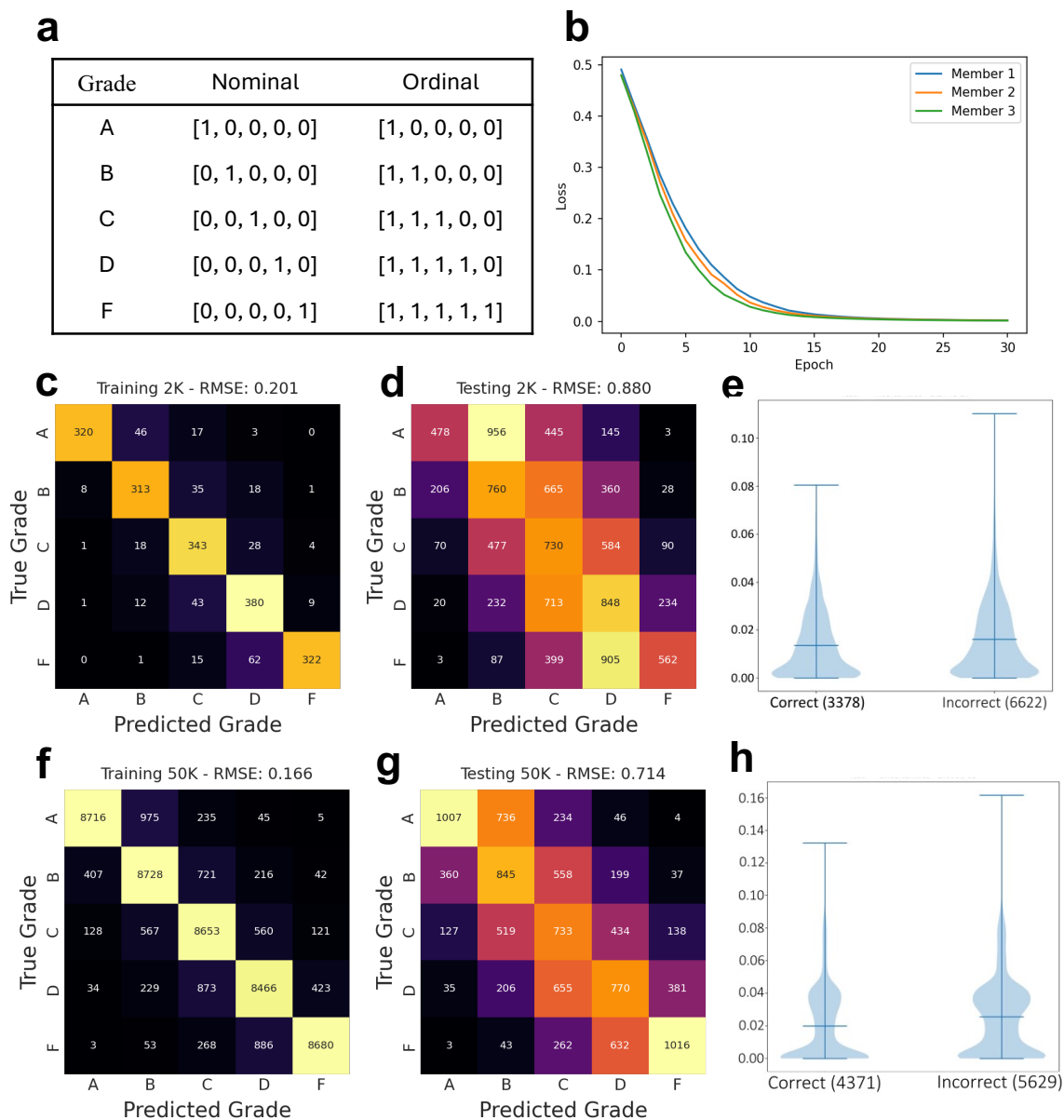
Supplemental Figures



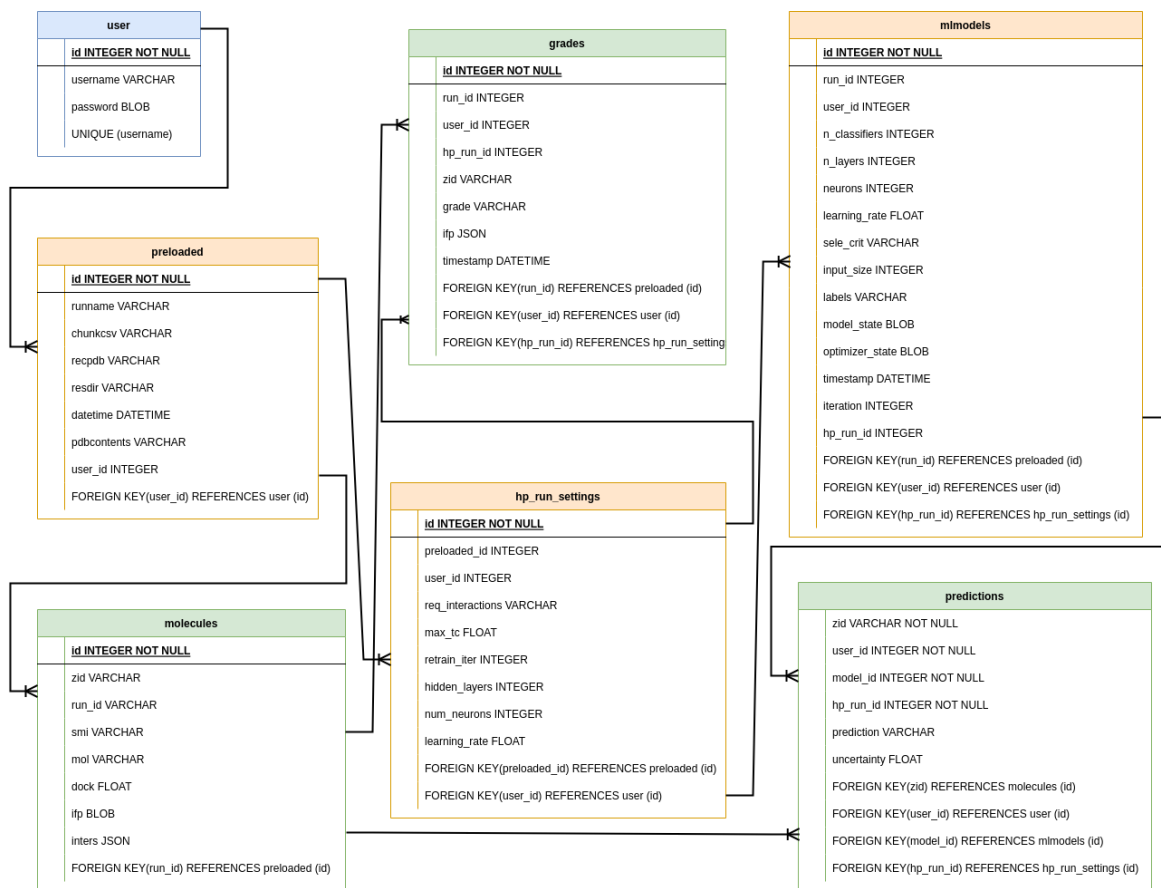
Supplemental Figure 3.1. Investigating Uncertainty Quantification with AA2AR

a-b, Visual representation of calibration coefficient metric. **a**, Example regression plot, colored by the value of the residual. **b**, Cumulative R^2 plot. R^2 is shown on the y-axis. The x-axis shows response rate, the proportion of examples included in the R^2 calculation. **c-f**, Calibration curves obtained for the four metrics evaluated with various hyperparameters. **c**, Distance, evaluating n nearest neighbors. **d**, Ensemble, evaluating initial sampling and committee size. **e**, Dropout, evaluating number of replicates. **f**, Bayesian, evaluating number of replicates. **g**, R^2 s and calibration coefficients for the best-performing models of all methods evaluated.

Supplemental Figure 3.2. Ordinal Label Training with D4 MM/GBSA-based Grades



a, Comparison of nominal vs ordinal encoding for labels. Ordinal encoding maintains a relationship between grades. **b**, Training curves for the committee members, 2000 examples. **c-e**. Results for 2,000 randomly selected complexes. **c**, Training set confusion matrix. **d**, Test set confusion matrix. **e**, Variance of ensemble predictions of correctly predicted and incorrectly predicted test examples. **f-h**, Results for 50,000 random complexes. **f**, Training set confusion matrix. **g**, Test set confusion matrix. **h**, Variance of ensemble predictions of correctly predicted and incorrectly predicted test examples.



Supplemental Figure 3.3. Schema for Autoparty SQL database.

The color of each table indicates the data stored: blue for users, orange for screens, settings, and models, and green for molecules, grades, and predictions. The forked connector denotes a one-to-many connection.

Supplemental Table 3.1. Autoparty hit-picking settings, options, and default values.

	Setting	Description	Type	Default	Options
General	learning_rate	Learning rate for the model, how quickly weights are updated	float	1e-4	(0, 1)
	n_neurons	Number of neurons per layer	int	1024	(0, inf)
	hidden_layers	Number of hidden layers in model	int	2	(0, inf)
	weight_decay	Penalty for large model weights	float	1e-2	(0, 1)
	dropout	Probability given neuron will be zeroed during training	float	0.2	[0, 1)
	output_options	List of potential grades to assign	list	a,b,c,d,f	Comma-separated unique values
	output_type	Relation of provided grades	string	ordinal	{ordinal, classes}
	uncertainty	Uncertainty calculation method	string	ensemble	{ensemble, dropout, distance}
	retrain_freq	How many new grades are required to start model training	string	300	(0, inf)
	max_epochs	Maximum number of epochs to train models	int	100	(0, inf)
	patience	Epochs to wait before stopping training (requires validation set)	int	20	(0, inf)

Supplemental Table 3.2. Uncertainty quantification settings, options, and default values.

	Setting	Description	Type	Default	Options
Ensemble	committee_size	Number of models in committee	int	3	(0, inf)
	data_split	Sampling method for ensemble training datasets	string	bootstrap	{bootstrap, full-split}
Dropout	passes	Number of predictions to generate per example	int	50	(0, inf)
Distance	distance_method	Distance metric to use for uncertainty	string	tanimoto	{tanimoto}
	kNN	Number of nearest neighbors to use for kNN distance calculation	int	5	(0, inf)

Supplemental Table 3.3. LUNA available interactions and associated colors for visualization.

Proximal	gray60
Hydrogen bond	tv_blue
Water-bridged hydrogen bond	lightblue
Weak hydrogen bond	lightteal
Ionic	green
Salt bridge	forest
Cation-pi	salmon
Amide-aromatic stacking	raspberry
Hydrophobic	orange
Halogen bond	aquamarine
Halogen-pi	aquamarine
Chalcogen bond	lightorange
Chalcogen-pi	lightorange
Repulsive	violetpurple
Covalent bond	black
Atom overlap	gray40
Van der Waals clash	gray90
Van der Waals	gray50
Orthogonal multipolar	paleyellow
Parallel multipolar	paleyellow
Antiparallel multipolar	paleyellow
Tilted multipolar	paleyellow
Multipolar	paleyellow
Unfavorable nucleophile-nucleophile	dirtyviolet
Unfavorable electrophile-electrophile	dirtyviolet
Cation-nucleophile	palegreen
Anion-electrophile	palegreen
Unfavorable anion-nucleophile	dirtyviolet
Unfavorable cation-electrophile	dirtyviolet
Pi-stacking	tv_red
Face-to-face pi-stacking	tv_red
Face-to-edge pi-stacking	tv_red
Face-to-slope pi-stacking	tv_red
Edge-to-edge pi-stacking	tv_red
Edge-to-face pi-stacking	tv_red
Edge-to-slope pi-stacking	tv_red
Displaced face-to-face pi-stacking	tv_red
Displaced face-to-edge pi-stacking	tv_red
Displaced face-to-slope pi-stacking	tv_red
Single bond	black
Double bond	black
Triple bond	black
Aromatic bond	black
Other bond	black
Metal coordination	olive

Supplemental Table 3.4. Calibration test architectures, hyperparameters, and metrics.

*CC = Calibration Coefficient

Architecture	Hyperparameter(s)	Value(s)	R ²	CC*
Ensemble	Sampling, Number of members	Unique, 3	0.77	0.22
		Unique, 5	0.76	0.26
		Bootstrap, 3	0.79	0.32
		Bootstrap, 5	0.79	0.38
Dropout	Number of predictions	5	0.78	0.17
		50	0.78	0.33
		500	0.78	0.37
		1000	0.78	0.37
Bayesian	Number of predictions	5	0.68	0.26
		50	0.68	0.30
Distance	Number of nearest neighbors	1	0.79	0.32
		2	0.79	0.33

CHAPTER 4: FINAL AND FUTURE THOUGHTS

This dissertation provides an overview of various AI methods for addressing outstanding problems in chemical biology. Chapter 1 provides a non-comprehensive literature review of existing technologies, highlighting potential improvements from machine learning for two major areas: 1) small molecules focusing on property prediction and drug discovery, and 2) proteins, describing structure prediction and *de novo* protein design. Chapter 2 provides a concrete example of how standard chemical informatics techniques like interface fingerprints can be modified and combined with machine learning architectures to develop additional validation techniques for assisting experimentalists. Chapter 3 continues this discussion with the introduction of Autoparty, a tool to improve the process of manual inspection following molecular docking and extract additional information from human annotations. Both tools assist existing workflows for biologists and chemists, a design philosophy that should be prioritized when developing new ML methods.

Finally, I want to highlight a few additional areas for drug design specifically that I believe will have a large impact in the coming years. The first of these is target-specific *de novo* molecular generation. While I touched on the recent advancements briefly in chapter 1, existing technologies based on diffusion, though promising, do not achieve the level of success that image generation achieves, likely due to a relative lack of training data. Better feature engineering and additional datasets, including the incorporation of structures from virtual screening, have the potential to significantly improve performance and revolutionize how we approach new drug targets. Second, the concept of a federated learning training paradigm that is able to use existing pharmaceutical data across companies while maintaining private proprietary information is intriguing, and could provide an avenue to use the vast amounts of collected chemical data in a way that is accessible to all. This could assist particularly in the models in predicting downstream ADMET properties and

off-target effects, allowing chemists to filter out unsuccessful compounds before buying and testing. Finally, the idea of transfer learning, or taking weights trained from one dataset and fine-tuning them on another, remains an underexplored opportunity to leverage large virtual screening data for binding affinity prediction. I firmly believe that each of these approaches could result in large leaps forward for the field of drug discovery, and I'm excited to see what the next AlphaFold-level advancement will be.

REFERENCES

1. Floridi, L. (2020). AI and Its New Winter: From Myths to Realities. *Philosophy & Technology*, 33(1), 1–3. <https://doi.org/10.1007/s13347-020-00396-6>
2. Bhardwaj, A., Kishore, S., & Pandey, D. K. (2022). Artificial Intelligence in Biological Sciences. *Life*, 12(9), 1430. <https://doi.org/10.3390/life12091430>
3. Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>
4. Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
5. McCarthy, J. (n.d.). *WHAT IS ARTIFICIAL INTELLIGENCE?*
6. Kersting, K. (2018). Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behavior in Machines. *Frontiers in Big Data*, 1, 6. <https://doi.org/10.3389/fdata.2018.00006>
7. Kühl, N., Schemmer, M., Goutier, M., & Satzger, G. (2022). Artificial intelligence and machine learning. *Electronic Markets*, 32(4), 2235–2244. <https://doi.org/10.1007/s12525-022-00598-0>
8. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489. <https://doi.org/10.1038/nature16961>
9. *Transcript of Mark Zuckerberg’s Senate hearing—The Washington Post*. (n.d.). Retrieved April 15, 2024, from <https://www.washingtonpost.com/news/the-switch/wp/2018/04/10/transcript-of-mark-zuckerbergs-senate-hearing/>

10. Miller, G. (2023, September 13). *Transcript: US Senate Judiciary Hearing on Oversight of A.I.* | *TechPolicy.Press*. Tech Policy Press. <https://techpolicy.press/transcript-us-senate-judiciary-hearing-on-oversight-of-a-i>
11. Lanning, D. R., Harrell, G. K., & Wang, J. (2014). Dijkstra's algorithm and Google maps. *Proceedings of the 2014 ACM Southeast Regional Conference*, 1–3. <https://doi.org/10.1145/2638404.2638494>
12. Qiu, T. M. (2023). A Review of Motion Planning for Urban Autonomous Driving. *2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, 32–36. <https://doi.org/10.1109/AINIT59027.2023.10212775>
13. van Melle, W. (1978). MYCIN: A knowledge-based consultation program for infectious disease diagnosis. *International Journal of Man-Machine Studies*, 10(3), 313–322. [https://doi.org/10.1016/S0020-7373\(78\)80049-2](https://doi.org/10.1016/S0020-7373(78)80049-2)
14. Aronson, A. R. (1997). DiagnosisPro: The Ultimate Differential Diagnosis Assistant. *JAMA*, 277(5), 426. <https://doi.org/10.1001/jama.1997.03540290078040>
15. Edberg, S. C. (2005). Global Infectious Diseases and Epidemiology Network (GIDEON): A world wide Web-based program for diagnosis and informatics in infectious diseases. *Clinical Infectious Diseases: An Official Publication of the Infectious Diseases Society of America*, 40(1), 123–126. <https://doi.org/10.1086/426549>
16. Nkuma-Udah, K. I., Chukwudebe, G. A., & Ekwonwune, E. N. (2018). Medical Diagnosis Expert System for Malaria and Related Diseases for Developing Countries. *E-Health Telecommunication Systems and Networks*, 7(2), Article 2. <https://doi.org/10.4236/etsn.2018.72002>
17. *An Expert System for PostOperative Care (POEMS)* | *Semantic Scholar*. (n.d.). Retrieved April 15, 2024, from [https://www.semanticscholar.org/paper/An-Expert-System-for-PostOperative-Care-\(POEMS\)-Sawar-Brennan/b16ad267f603eddc6ef260876696bfefcd6759b](https://www.semanticscholar.org/paper/An-Expert-System-for-PostOperative-Care-(POEMS)-Sawar-Brennan/b16ad267f603eddc6ef260876696bfefcd6759b)

18. Guillaumin, M., Verbeek, J., & Schmid, C. (2009). Is that you? Metric learning approaches for face identification. *2009 IEEE 12th International Conference on Computer Vision*, 498–505.
<https://doi.org/10.1109/ICCV.2009.5459197>
19. Teoh, K., Ismail, R., Naziri, S., Hussin, R., Isa, M., & Basir, M. (2021). Face Recognition and Identification using Deep Learning Approach. *Journal of Physics: Conference Series*, 1755(1), 012006. <https://doi.org/10.1088/1742-6596/1755/1/012006>
20. Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon*, 5(6), e01802. <https://doi.org/10.1016/j.heliyon.2019.e01802>
21. Kaddoura, S., Chandrasekaran, G., Elena Popescu, D., & Duraisamy, J. H. (2022). A systematic literature review on spam content detection and classification. *PeerJ Computer Science*, 8, e830. <https://doi.org/10.7717/peerj-cs.830>
22. Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization* (pp. 325–341). Springer. https://doi.org/10.1007/978-3-540-72079-9_10
23. Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). *Wide & Deep Learning for Recommender Systems* (arXiv:1606.07792). arXiv.
<http://arxiv.org/abs/1606.07792>
24. Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1), 59. <https://doi.org/10.1186/s40537-022-00592-5>
25. Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., & Jain, S. (2017). Machine translation using deep learning: An overview. *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 162–167. <https://doi.org/10.1109/COMPTLIX.2017.8003957>
26. Popel, M., Tomkova, M., Tomek, J., Kaiser, Ł., Uszkoreit, J., Bojar, O., & Žabokrtský, Z. (2020). Transforming machine translation: A deep learning system reaches news translation quality

- comparable to human professionals. *Nature Communications*, *11*(1), 4381.
<https://doi.org/10.1038/s41467-020-18073-9>
27. Stahlberg, F. (2020). Neural Machine Translation: A Review. *Journal of Artificial Intelligence Research*, *69*, 343–418. <https://doi.org/10.1613/jair.1.12007>
28. Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., & Liu, Y. (2020). Neural machine translation: A review of methods, resources, and tools. *AI Open*, *1*, 5–21.
<https://doi.org/10.1016/j.aiopen.2020.11.001>
29. Hollebeek, L. D., Sprott, D. E., & Brady, M. K. (2021). Rise of the Machines? Customer Engagement in Automated Service Interactions. *Journal of Service Research*, *24*(1), 3–8.
<https://doi.org/10.1177/1094670520975110>
30. Peruchini, M., da Silva, G. M., & Teixeira, J. M. (2024). Between artificial intelligence and customer experience: A literature review on the intersection. *Discover Artificial Intelligence*, *4*(1), 4.
<https://doi.org/10.1007/s44163-024-00105-8>
31. Kietzmann, J., Paschen, J., & Treen, E. (2018). Artificial Intelligence in Advertising: How Marketers Can Leverage Artificial Intelligence Along the Consumer Journey. *Journal of Advertising Research*, *58*(3), 263–267. <https://doi.org/10.2501/JAR-2018-035>
32. Choi, J.-A., & Lim, K. (2020). Identifying machine learning techniques for classification of target advertising. *ICT Express*, *6*(3), 175–180. <https://doi.org/10.1016/j.icte.2020.04.012>
33. Ullal, M. S., Hawaldar, I. T., Soni, R., & Nadeem, M. (2021). The Role of Machine Learning in Digital Marketing. *Sage Open*, *11*(4), 21582440211050394.
<https://doi.org/10.1177/21582440211050394>
34. De Mauro, A., Sestino, A., & Bacconi, A. (2022). Machine learning and artificial intelligence use in marketing: A general taxonomy. *Italian Journal of Marketing*, *2022*(4), 439–457.
<https://doi.org/10.1007/s43039-022-00057-w>

35. Haleem, A., Javaid, M., Asim Qadri, M., Pratap Singh, R., & Suman, R. (2022). Artificial intelligence (AI) applications for marketing: A literature-based study. *International Journal of Intelligent Networks*, 3, 119–132. <https://doi.org/10.1016/j.ijin.2022.08.005>
36. Gao, B., Wang, Y., Xie, H., Hu, Y., & Hu, Y. (2023). Artificial Intelligence in Advertising: Advancements, Challenges, and Ethical Considerations in Targeting, Personalization, Content Creation, and Ad Optimization. *Sage Open*, 13(4), 21582440231210759. <https://doi.org/10.1177/21582440231210759>
37. Esmaily, H., Tayefi, M., Doosti, H., Ghayour-Mobarhan, M., Nezami, H., & Amirabadizadeh, A. (2018). A Comparison between Decision Tree and Random Forest in Determining the Risk Factors Associated with Type 2 Diabetes. *Journal of Research in Health Sciences*, 18(2), 412.
38. Azmi, S. S., & Baliga, S. (2020). *An Overview of Boosting Decision Tree Algorithms utilizing AdaBoost and XGBoost Boosting strategies*. 07(05).
39. Shmilovici, A. (n.d.). SUPPORT VECTOR MACHINES. *DATA MINING AND KNOWLEDGE DISCOVERY HANDBOOK*.
40. Langley, P., Iba, and, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 223–228.
41. Schmidhuber, J. (2022). *Annotated History of Modern AI and Deep Learning* (arXiv:2212.11279). arXiv. <http://arxiv.org/abs/2212.11279>
42. Ramachandran, P., Zoph, B., & Le, Q. V. (2017). *Searching for Activation Functions* (arXiv:1710.05941). arXiv. <http://arxiv.org/abs/1710.05941>
43. Han, J., Moraga, C., & Sinne, S. (1996). Optimization of feedforward neural networks. *Engineering Applications of Artificial Intelligence*, 9(2), 109–119. [https://doi.org/10.1016/0952-1976\(95\)00001-1](https://doi.org/10.1016/0952-1976(95)00001-1)
44. O’Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (arXiv:1511.08458). arXiv. <http://arxiv.org/abs/1511.08458>

45. Marhon, S. A., Cameron, C. J. F., & Kremer, S. C. (2013). Recurrent Neural Networks. In M. Bianchini, M. Maggini, & L. C. Jain (Eds.), *Handbook on Neural Information Processing* (pp. 29–65). Springer. https://doi.org/10.1007/978-3-642-36657-4_2
46. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. <https://doi.org/10.1016/j.aiopen.2021.01.001>
47. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. <http://arxiv.org/abs/1706.03762>
48. De Angelis, L., Baglivo, F., Arzilli, G., Privitera, G. P., Ferragina, P., Tozzi, A. E., & Rizzo, C. (2023). ChatGPT and the rise of large language models: The new AI-driven infodemic threat in public health. *Frontiers in Public Health*, *11*, 1166120. <https://doi.org/10.3389/fpubh.2023.1166120>
49. Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, *28*(1), 31–36. <https://doi.org/10.1021/ci00057a005>
50. Weininger, D., Weininger, A., & Weininger, J. L. (1989). SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, *29*(2), 97–101. <https://doi.org/10.1021/ci00062a008>
51. Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., & Overington, J. P. (2012). ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(Database issue), D1100–D1107. <https://doi.org/10.1093/nar/gkr777>
52. Tingle, B. I., Tang, K. G., Castanon, M., Gutierrez, J. J., Khurelbaatar, M., Dandarchuluun, C., Moroz, Y. S., & Irwin, J. J. (2023). ZINC-22—A Free Multi-Billion-Scale Database of Tangible

- Compounds for Ligand Discovery. *Journal of Chemical Information and Modeling*, 63(4), 1166–1176. <https://doi.org/10.1021/acs.jcim.2c01253>
53. Ruddigkeit, L., van Deursen, R., Blum, L. C., & Reymond, J.-L. (2012). Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, 52(11), 2864–2875. <https://doi.org/10.1021/ci300415d>
54. Heller, S. R., McNaught, A., Pletnev, I., Stein, S., & Tchekhovskoi, D. (2015). InChI, the IUPAC International Chemical Identifier. *Journal of Cheminformatics*, 7(1), 23. <https://doi.org/10.1186/s13321-015-0068-4>
55. Krenn, M., Häse, F., Nigam, A., Friederich, P., & Aspuru-Guzik, A. (2020). Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4), 045024. <https://doi.org/10.1088/2632-2153/aba947>
56. Cahuantzi, R., Chen, X., & Güttel, S. (2023). *A comparison of LSTM and GRU networks for learning symbolic sequences* (Vol. 739, pp. 771–785). https://doi.org/10.1007/978-3-031-37963-5_53
57. *RDKit*. (n.d.). Retrieved April 22, 2024, from <https://www.rdkit.org/>
58. *Molecular Modeling Software | OpenEye Scientific*. (n.d.). Retrieved April 29, 2024, from <https://www.eyesopen.com>
59. Durant, J. L., Leland, B. A., Henry, D. R., & Nourse, J. G. (2002). Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42(6), 1273–1280. <https://doi.org/10.1021/ci010132r>
60. Kim, S. (2021). Exploring Chemical Information in PubChem. *Current Protocols*, 1(8), e217. <https://doi.org/10.1002/cpz1.217>
61. Rogers, D., & Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5), 742–754. <https://doi.org/10.1021/ci100050t>
62. Morgan, H. L. (1965). 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. <https://doi.org/10.1021/c160017a018>

63. Sastry, M., Lowrie, J. F., Dixon, S. L., & Sherman, W. (2010). Large-Scale Systematic Analysis of 2D Fingerprint Methods and Parameters to Improve Virtual Screening Enrichments. *Journal of Chemical Information and Modeling*, *50*(5), 771–784. <https://doi.org/10.1021/ci100062n>
64. Muegge, I., & Mukherjee, P. (2016). An overview of molecular fingerprint similarity search in virtual screening. *Expert Opinion on Drug Discovery*, *11*(2), 137–148. <https://doi.org/10.1517/17460441.2016.1117070>
65. Axen, S. D., Huang, X.-P., Cáceres, E. L., Gendele, L., Roth, B. L., & Keiser, M. J. (2017). A simple representation of three-dimensional molecular structure. *Journal of Medicinal Chemistry*, *60*(17), 7393–7409. <https://doi.org/10.1021/acs.jmedchem.7b00696>
66. Fassio, A. V., Shub, L., Ponzoni, L., McKinley, J., O'Meara, M. J., Ferreira, R. S., Keiser, M. J., & de Melo Minardi, R. C. (2022). Prioritizing Virtual Screening with Interpretable Interaction Fingerprints. *Journal of Chemical Information and Modeling*, *62*(18), 4300–4318. <https://doi.org/10.1021/acs.jcim.2c00695>
67. Wójcikowski, M., Kukielka, M., Stepińska-Dziubińska, M. M., & Siedlecki, P. (2019). Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. *Bioinformatics (Oxford, England)*, *35*(8), 1334–1341. <https://doi.org/10.1093/bioinformatics/bty757>
68. Da, C., & Kireev, D. (2014). Structural Protein–Ligand Interaction Fingerprints (SPLIF) for Structure-Based Virtual Screening: Method and Benchmark Study. *Journal of Chemical Information and Modeling*, *54*(9), 2555–2561. <https://doi.org/10.1021/ci500319f>
69. Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). *Convolutional Networks on Graphs for Learning Molecular Fingerprints* (arXiv:1509.09292). arXiv. <http://arxiv.org/abs/1509.09292>
70. Capecchi, A., Probst, D., & Reymond, J.-L. (2020). One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome. *Journal of Cheminformatics*, *12*(1), 43. <https://doi.org/10.1186/s13321-020-00445-4>

71. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). *Neural Message Passing for Quantum Chemistry* (arXiv:1704.01212). arXiv. <https://doi.org/10.48550/arXiv.1704.01212>
72. *QSAR Modeling: Where Have You Been? Where Are You Going To?* | *Journal of Medicinal Chemistry*. (n.d.). Retrieved April 26, 2024, from <https://pubs.acs.org/doi/10.1021/jm4004285>
73. Toropov, A. A., & Toropova, A. P. (2020). QSPR/QSAR: State-of-Art, Weirdness, the Future. *Molecules*, 25(6), Article 6. <https://doi.org/10.3390/molecules25061292>
74. Hansch, Corwin., & Fujita, Toshio. (1964). p - σ - π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *Journal of the American Chemical Society*, 86(8), 1616–1626. <https://doi.org/10.1021/ja01062a035>
75. Leo, A., Jow, P. Y. C., Silipo, C., & Hansch, C. (1975). Calculation of hydrophobic constant (log P) from π . And f constants. *Journal of Medicinal Chemistry*, 18(9), 865–868. <https://doi.org/10.1021/jm00243a001>
76. Cruz-Montegudo, M., Borges, F., & Cordeiro, M. N. D. S. (2008). Desirability-based multiobjective optimization for global QSAR studies: Application to the design of novel NSAIDs with improved analgesic, antiinflammatory, and ulcerogenic profiles. *Journal of Computational Chemistry*, 29(14), 2445–2459. <https://doi.org/10.1002/jcc.20994>
77. Verma, J., Khedkar, V. M., & Coutinho, E. C. (n.d.). 3D-QSAR in Drug Design—A Review. *Current Topics in Medicinal Chemistry*, 10(1), 95–115.
78. Myint, K.-Z., Wang, L., Tong, Q., & Xie, X.-Q. (2012). Molecular Fingerprint-based Artificial Neural Networks QSAR for Ligand Biological Activity Predictions. *Molecular Pharmaceutics*, 9(10), 2912–2923. <https://doi.org/10.1021/mp300237z>
79. Gao, K., Nguyen, D. D., Sresht, V., Mathiowetz, A. M., Tu, M., & Wei, G.-W. (2020). Are 2D fingerprints still valuable for drug discovery? *Physical Chemistry Chemical Physics*, 22(16), 8373–8390. <https://doi.org/10.1039/D0CP00305K>

80. Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., & Langer, T. (2020). A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37, 1–12. <https://doi.org/10.1016/j.ddtec.2020.11.009>
81. Tang, M., Li, B., & Chen, H. (2023). Application of message passing neural networks for molecular property prediction. *Current Opinion in Structural Biology*, 81, 102616. <https://doi.org/10.1016/j.sbi.2023.102616>
82. Ryu, S., Lim, J., Hong, S. H., & Kim, W. Y. (2018). *Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network* (arXiv:1805.10988). arXiv. <http://arxiv.org/abs/1805.10988>
83. Feinberg, E. N., Joshi, E., Pande, V. S., & Cheng, A. C. (2020). Improvement in ADMET Prediction with Multitask Deep Featurization. *Journal of Medicinal Chemistry*, 63(16), 8835–8848. <https://doi.org/10.1021/acs.jmedchem.9b02187>
84. Kireev, D. B. (1995). ChemNet: A Novel Neural Network Based Method for Graph/Property Mapping. *Journal of Chemical Information and Computer Sciences*, 35(2), 175–180. <https://doi.org/10.1021/ci00024a001>
85. Coley, C. W., Barzilay, R., Green, W. H., Jaakkola, T. S., & Jensen, K. F. (2017). Convolutional Embedding of Attributed Molecular Graphs for Physical Property Prediction. *Journal of Chemical Information and Modeling*, 57(8), 1757–1772. <https://doi.org/10.1021/acs.jcim.6b00601>
86. Lusci, A., Pollastri, G., & Baldi, P. (2013). Deep Architectures and Deep Learning in Chemoinformatics: The Prediction of Aqueous Solubility for Drug-Like Molecules. *Journal of Chemical Information and Modeling*, 53(7), 1563–1575. <https://doi.org/10.1021/ci400187y>
87. Wang, X., Li, Z., Jiang, M., Wang, S., Zhang, S., & Wei, Z. (2019). Molecule Property Prediction Based on Spatial Graph Embedding. *Journal of Chemical Information and Modeling*, 59(9), 3817–3828. <https://doi.org/10.1021/acs.jcim.9b00410>

88. Meng, M., Wei, Z., Li, Z., Jiang, M., & Bian, Y. (2019). Property Prediction of Molecules in Graph Convolutional Neural Network Expansion. *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 263–266.
<https://doi.org/10.1109/ICSESS47205.2019.9040723>
89. Withnall, M., Lindelöf, E., Engkvist, O., & Chen, H. (2020). Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *Journal of Cheminformatics*, *12*(1), 1. <https://doi.org/10.1186/s13321-019-0407-y>
90. Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T., Jensen, K., & Barzilay, R. (2019). Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, *59*(8), 3370–3388. <https://doi.org/10.1021/acs.jcim.9b00237>
91. Ma, H., Bian, Y., Rong, Y., Huang, W., Xu, T., Xie, W., Ye, G., & Huang, J. (2020). *Multi-View Graph Neural Networks for Molecular Property Prediction* (arXiv:2005.13607). arXiv.
<http://arxiv.org/abs/2005.13607>
92. Chen, B., Bécigneul, G., Ganea, O.-E., Barzilay, R., & Jaakkola, T. (2021). *Optimal Transport Graph Neural Networks* (arXiv:2006.04804). arXiv. <http://arxiv.org/abs/2006.04804>
93. Wu, Z., Ramsundar, B., N. Feinberg, E., Gomes, J., Geniesse, C., S. Pappu, A., Leswing, K., & Pande, V. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, *9*(2), 513–530. <https://doi.org/10.1039/C7SC02664A>
94. Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., & Leskovec, J. (2020). *Strategies for Pre-training Graph Neural Networks* (arXiv:1905.12265). arXiv. <http://arxiv.org/abs/1905.12265>
95. Li, R., Wang, S., Zhu, F., & Huang, J. (2018). Adaptive Graph Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), Article 1.
<https://doi.org/10.1609/aaai.v32i1.11691>

96. Tang, B., Kramer, S. T., Fang, M., Qiu, Y., Wu, Z., & Xu, D. (2020). A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *Journal of Cheminformatics*, 12(1), 15. <https://doi.org/10.1186/s13321-020-0414-z>
97. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1), 61–80. <https://doi.org/10.1109/TNN.2008.2005605>
98. Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018). An End-to-End Deep Learning Architecture for Graph Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Article 1. <https://doi.org/10.1609/aaai.v32i1.11782>
99. Niepert, M., Ahmed, M., & Kutzkov, K. (n.d.). *Learning Convolutional Neural Networks for Graphs*.
100. Simonovsky, M., & Komodakis, N. (2017). *Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs* (arXiv:1704.02901). arXiv. <http://arxiv.org/abs/1704.02901>
101. Altae-Tran, H., Ramsundar, B., Pappu, A. S., & Pande, V. (2017). Low Data Drug Discovery with One-Shot Learning. *ACS Central Science*, 3(4), 283–293. <https://doi.org/10.1021/acscentsci.6b00367>
102. Li, J., Cai, D., & He, X. (2017). *Learning Graph-Level Representation for Drug Discovery* (arXiv:1709.03741). arXiv. <http://arxiv.org/abs/1709.03741>
103. Cremer, J., Medrano Sandonas, L., Tkatchenko, A., Clevert, D.-A., & De Fabritiis, G. (2023). Equivariant Graph Neural Networks for Toxicity Prediction. *Chemical Research in Toxicology*, 36(10), 1561–1573. <https://doi.org/10.1021/acs.chemrestox.3c00032>
104. Sanchez-Garcia, R., Havasi, D., Takács, G., Robinson, M. C., Lee, A., & Deane, C. M. (n.d.). *CoPriNet: Graph Neural Networks provide accurate and rapid compound price prediction for molecule prioritisation*.
105. Heid, E., Greenman, K. P., Chung, Y., Li, S.-C., Graff, D. E., Vermeire, F. H., Wu, H., Green, W. H., & McGill, C. J. (2024). Chemprop: A Machine Learning Package for Chemical Property

- Prediction. *Journal of Chemical Information and Modeling*, 64(1), 9–17.
<https://doi.org/10.1021/acs.jcim.3c01250>
106. Satorras, V. G., Hoogeboom, E., & Welling, M. (2022). *E(n) Equivariant Graph Neural Networks* (arXiv:2102.09844). arXiv. <http://arxiv.org/abs/2102.09844>
107. Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., & Tang, J. (2022). *GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation* (arXiv:2203.02923). arXiv.
<http://arxiv.org/abs/2203.02923>
108. Stärk, H., Ganea, O.-E., Pattanaik, L., Barzilay, R., & Jaakkola, T. (2022). *EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction* (arXiv:2202.05146). arXiv.
<https://doi.org/10.48550/arXiv.2202.05146>
109. Ganea, O.-E., Huang, X., Bunne, C., Bian, Y., Barzilay, R., Jaakkola, T., & Krause, A. (2022). *Independent SE(3)-Equivariant Models for End-to-End Rigid Protein Docking* (arXiv:2111.07786). arXiv. <http://arxiv.org/abs/2111.07786>
110. Ghorbani, M., Gendele, L., Beroza, P., & Keiser, M. J. (2023). *Autoregressive fragment-based diffusion for pocket-aware ligand design* (arXiv:2401.05370). arXiv.
<http://arxiv.org/abs/2401.05370>
111. Guan, J., Qian, W. W., Peng, X., Su, Y., Peng, J., & Ma, J. (2023). *3D EQUIVARIANT DIFFUSION FOR TARGET-AWARE MOLECULE GENERATION AND AFFINITY PREDICTION*.
112. Ramakrishnan, R., Dral, P. O., Rupp, M., & von Lilienfeld, O. A. (2014). Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1), 140022.
<https://doi.org/10.1038/sdata.2014.22>
113. Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., & Müller, K.-R. (2017). Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5), e1603015. <https://doi.org/10.1126/sciadv.1603015>

114. Christensen, A. S., & Von Lilienfeld, O. A. (2020). On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1(4), 045018. <https://doi.org/10.1088/2632-2153/abba6f>
115. Axelrod, S., & Gómez-Bombarelli, R. (2022). GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1), 185. <https://doi.org/10.1038/s41597-022-01288-4>
116. Isert, C., Atz, K., Jiménez-Luna, J., & Schneider, G. (2022). QMugs, quantum mechanical properties of drug-like molecules. *Scientific Data*, 9(1), 273. <https://doi.org/10.1038/s41597-022-01390-7>
117. Schütt, K. T., Kindermans, P.-J., Saucedo, H. E., Chmiela, S., Tkatchenko, A., & Müller, K.-R. (2017). *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions* (arXiv:1706.08566). arXiv. <http://arxiv.org/abs/1706.08566>
118. Gasteiger, J., Groß, J., & Günnemann, S. (2022). *Directional Message Passing for Molecular Graphs* (arXiv:2003.03123). arXiv. <http://arxiv.org/abs/2003.03123>
119. Gasteiger, J., Giri, S., Margraf, J. T., & Günnemann, S. (2022). *Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules* (arXiv:2011.14115). arXiv. <http://arxiv.org/abs/2011.14115>
120. Liu, Y., Wang, L., Liu, M., Zhang, X., Oztekin, B., & Ji, S. (2022). *Spherical Message Passing for 3D Graph Networks* (arXiv:2102.05013). arXiv. <http://arxiv.org/abs/2102.05013>
121. Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wang, F., Wu, H., & Wang, H. (2022). Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2), 127–134. <https://doi.org/10.1038/s42256-021-00438-4>
122. Choudhary, K., & DeCost, B. (2021). Atomistic Line Graph Neural Network for improved materials property predictions. *Npj Computational Materials*, 7(1), 1–8. <https://doi.org/10.1038/s41524-021-00650-1>

123. DiMasi, J. A., Hansen, R. W., & Grabowski, H. G. (2003). The price of innovation: New estimates of drug development costs. *Journal of Health Economics*, 22(2), 151–185.
[https://doi.org/10.1016/S0167-6296\(02\)00126-1](https://doi.org/10.1016/S0167-6296(02)00126-1)
124. DiMasi, J. A., Grabowski, H. G., & Hansen, R. W. (2016). Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47, 20–33.
<https://doi.org/10.1016/j.jhealeco.2016.01.012>
125. Mayr, L. M., & Bojanic, D. (2009). Novel trends in high-throughput screening. *Current Opinion in Pharmacology*, 9(5), 580–588. <https://doi.org/10.1016/j.coph.2009.08.004>
126. Keserü, G. M., & Makara, G. M. (2009). The influence of lead discovery strategies on the properties of drug candidates. *Nature Reviews Drug Discovery*, 8(3), 203–212.
<https://doi.org/10.1038/nrd2796>
127. Ewing, T. J. A., & Kuntz, I. D. (1997). Critical evaluation of search algorithms for automated molecular docking and database screening. *Journal of Computational Chemistry*, 18(9), 1175–1189. [https://doi.org/10.1002/\(SICI\)1096-987X\(19970715\)18:9<1175::AID-JCC6>3.0.CO;2-O](https://doi.org/10.1002/(SICI)1096-987X(19970715)18:9<1175::AID-JCC6>3.0.CO;2-O)
128. Coleman, R. G., Carchia, M., Sterling, T., Irwin, J. J., & Shoichet, B. K. (2013). Ligand Pose and Orientational Sampling in Molecular Docking. *PLOS ONE*, 8(10), e75992.
<https://doi.org/10.1371/journal.pone.0075992>
129. Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., & Shenkin, P. S. (2004). 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.
<https://doi.org/10.1021/jm0306430>
130. Rarey, M., Kramer, B., Lengauer, T., & Klebe, G. (1996). A Fast Flexible Docking Method using an Incremental Construction Algorithm. *Journal of Molecular Biology*, 261(3), 470–489.
<https://doi.org/10.1006/jmbi.1996.0477>

131. Bursulaya, B. D., Totrov, M., Abagyan, R., & Brooks, C. L. (2003). Comparative study of several algorithms for flexible ligand docking. *Journal of Computer-Aided Molecular Design*, *17*(11), 755–763. <https://doi.org/10.1023/b:jcam.0000017496.76572.6f>
132. Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., & Olson, A. J. (1998). Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, *19*(14), 1639–1662. [https://doi.org/10.1002/\(SICI\)1096-987X\(19981115\)19:14<1639::AID-JCC10>3.0.CO;2-B](https://doi.org/10.1002/(SICI)1096-987X(19981115)19:14<1639::AID-JCC10>3.0.CO;2-B)
133. Jones, G., Willett, P., Glen, R. C., Leach, A. R., & Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of Molecular Biology*, *267*(3), 727–748. <https://doi.org/10.1006/jmbi.1996.0897>
134. Totrov, M., & Abagyan, R. (1997). Flexible protein-ligand docking by global energy optimization in internal coordinates. *Proteins, Suppl 1*, 215–220. [https://doi.org/10.1002/\(sici\)1097-0134\(1997\)1+<215::aid-prot29>3.3.co;2-i](https://doi.org/10.1002/(sici)1097-0134(1997)1+<215::aid-prot29>3.3.co;2-i)
135. Lemmon, G., & Meiler, J. (2012). RosettaLigand docking with flexible XML protocols. *Methods in Molecular Biology (Clifton, N.J.)*, *819*, 143–155. https://doi.org/10.1007/978-1-61779-465-0_10
136. DeLuca, S., Khar, K., & Meiler, J. (2015). Fully Flexible Docking of Medium Sized Ligand Libraries with RosettaLigand. *PLOS ONE*, *10*(7), e0132508. <https://doi.org/10.1371/journal.pone.0132508>
137. Ravindranath, P. A., Forli, S., Goodsell, D. S., Olson, A. J., & Sanner, M. F. (2015). AutoDockFR: Advances in Protein-Ligand Docking with Explicitly Specified Binding Site Flexibility. *PLoS Computational Biology*, *11*(12), e1004586. <https://doi.org/10.1371/journal.pcbi.1004586>
138. Böhm, H. J. (1994). The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *Journal of Computer-Aided Molecular Design*, *8*(3), 243–256. <https://doi.org/10.1007/BF00126743>

139. Muegge, I., & Rarey, M. (2001). Small Molecule Docking and Scoring. In *Reviews in Computational Chemistry* (pp. 1–60). John Wiley & Sons, Ltd.
<https://doi.org/10.1002/0471224413.ch1>
140. Liu, J., & Wang, R. (2015). Classification of Current Scoring Functions. *Journal of Chemical Information and Modeling*, *55*(3), 475–482. <https://doi.org/10.1021/ci500731a>
141. Zou, X., Yaxiong, & Kuntz, I. D. (1999). Inclusion of Solvation in Ligand Binding Free Energy Calculations Using the Generalized-Born Model. *Journal of the American Chemical Society*, *121*(35), 8033–8043. <https://doi.org/10.1021/ja984102p>
142. Gilson, M. K., Given, J. A., & Head, M. S. (1997). A new class of models for computing receptor-ligand binding affinities. *Chemistry & Biology*, *4*(2), 87–92. [https://doi.org/10.1016/S1074-5521\(97\)90251-9](https://doi.org/10.1016/S1074-5521(97)90251-9)
143. Pason, L. P., & Sotriffer, C. A. (2016). Empirical Scoring Functions for Affinity Prediction of Protein-ligand Complexes. *Molecular Informatics*, *35*(11–12), 541–548.
<https://doi.org/10.1002/minf.201600048>
144. Guedes, I. A., Pereira, F. S. S., & Dardenne, L. E. (2018). Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges. *Frontiers in Pharmacology*, *9*. <https://doi.org/10.3389/fphar.2018.01089>
145. Eldridge, M. D., Murray, C. W., Auton, T. R., Paolini, G. V., & Mee, R. P. (1997). Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. *Journal of Computer-Aided Molecular Design*, *11*(5), 425–445. <https://doi.org/10.1023/a:1007996124545>
146. Wang, R., Lai, L., & Wang, S. (2002). Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design*, *16*(1), 11–26. <https://doi.org/10.1023/a:1016357811882>
147. Friesner, R. A., Murphy, R. B., Repasky, M. P., Frye, L. L., Greenwood, J. R., Halgren, T. A., Sanschagrin, P. C., & Mainz, D. T. (2006). Extra Precision Glide: Docking and Scoring

- Incorporating a Model of Hydrophobic Enclosure for Protein–Ligand Complexes. *Journal of Medicinal Chemistry*, 49(21), 6177–6196. <https://doi.org/10.1021/jm051256o>
148. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1), 235–242.
149. Berman, H., Henrick, K., Nakamura, H., & Markley, J. L. (2007). The worldwide Protein Data Bank (wwPDB): Ensuring a single, uniform archive of PDB data. *Nucleic Acids Research*, 35(Database issue), D301. <https://doi.org/10.1093/nar/gkl971>
150. Smith, S. T., & Meiler, J. (2020). Assessing multiple score functions in Rosetta for drug discovery. *PLoS ONE*, 15(10), e0240450. <https://doi.org/10.1371/journal.pone.0240450>
151. Gilson, M. K., & Zhou, H.-X. (2007). Calculation of Protein-Ligand Binding Affinities*. *Annual Review of Biophysics*, 36(Volume 36, 2007), 21–42. <https://doi.org/10.1146/annurev.biophys.36.040306.132550>
152. Ballester, P. J., & Mitchell, J. B. O. (2010). A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics (Oxford, England)*, 26(9), 1169–1175. <https://doi.org/10.1093/bioinformatics/btq112>
153. Das, S., Krein, M. P., & Breneman, C. M. (2010). Binding Affinity prediction with Property Encoded Shape Distribution signatures. *Journal of Chemical Information and Modeling*, 50(2), 298–308. <https://doi.org/10.1021/ci9004139>
154. Durrant, J. D., & McCammon, J. A. (2010). NNScore: A Neural-Network-Based Scoring Function for the Characterization of Protein–Ligand Complexes. *Journal of Chemical Information and Modeling*, 50(10), 1865–1871. <https://doi.org/10.1021/ci100244v>
155. Durrant, J. D., & McCammon, J. A. (2011). NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function. *Journal of Chemical Information and Modeling*, 51(11), 2897–2903. <https://doi.org/10.1021/ci2003889>

156. Gabel, J., Desaphy, J., & Rognan, D. (2014). Beware of Machine Learning-Based Scoring Functions—On the Danger of Developing Black Boxes. *Journal of Chemical Information and Modeling*, *54*(10), 2807–2815. <https://doi.org/10.1021/ci500406k>
157. Deng, Z., Chuaqui, C., & Singh, J. (2004). Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein–Ligand Binding Interactions. *Journal of Medicinal Chemistry*, *47*(2), 337–344. <https://doi.org/10.1021/jm030331x>
158. Jiménez, J., Škalič, M., Martínez-Rosell, G., & De Fabritiis, G. (2018). KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, *58*(2), 287–296. <https://doi.org/10.1021/acs.jcim.7b00650>
159. McNutt, A. T., Francoeur, P., Aggarwal, R., Masuda, T., Meli, R., Ragoza, M., Sunseri, J., & Koes, D. R. (2021). GNINA 1.0: Molecular docking with deep learning. *Journal of Cheminformatics*, *13*(1), 43. <https://doi.org/10.1186/s13321-021-00522-2>
160. Zheng, L., Fan, J., & Mu, Y. (2019). OnionNet: A Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction. *ACS Omega*, *4*(14), 15956–15965. <https://doi.org/10.1021/acsomega.9b01997>
161. Wang, Z., Zheng, L., Liu, Y., Qu, Y., Li, Y.-Q., Zhao, M., Mu, Y., & Li, W. (2021). OnionNet-2: A Convolutional Neural Network Model for Predicting Protein-Ligand Binding Affinity Based on Residue-Atom Contacting Shells. *Frontiers in Chemistry*, *9*. <https://doi.org/10.3389/fchem.2021.753002>
162. Wang, R., Fang, X., Lu, Y., & Wang, S. (2004). The PDBbind Database: Collection of Binding Affinities for Protein–Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry*, *47*(12), 2977–2980. <https://doi.org/10.1021/jm0305801>
163. Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., & Wang, R. (2019). Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling*, *59*(2), 895–913. <https://doi.org/10.1021/acs.jcim.8b00545>

164. Gale-Day, Z. J., Shub, L., Chuang, K. V., & Keiser, M. J. (2024). *Proximity Graph Networks: Predicting Ligand Affinity with Message Passing Neural Networks*. ChemRxiv.
<https://doi.org/10.26434/chemrxiv-2024-hznxh>
165. Li, S., Zhou, J., Xu, T., Huang, L., Wang, F., Xiong, H., Huang, W., Dou, D., & Xiong, H. (2021). Structure-aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 975–985. <https://doi.org/10.1145/3447548.3467311>
166. Zhang, X., Gao, H., Wang, H., Chen, Z., Zhang, Z., Chen, X., Li, Y., Qi, Y., & Wang, R. (2024). PLANET: A Multi-objective Graph Neural Network Model for Protein–Ligand Binding Affinity Prediction. *Journal of Chemical Information and Modeling*, 64(7), 2205–2220.
<https://doi.org/10.1021/acs.jcim.3c00253>
167. Zhang, S., Jin, Y., Liu, T., Wang, Q., Zhang, Z., Zhao, S., & Shan, B. (2023). SS-GNN: A Simple-Structured Graph Neural Network for Affinity Prediction. *ACS Omega*, 8(25), 22496–22507.
<https://doi.org/10.1021/acsomega.3c00085>
168. Li, S., Zhou, J., Xu, T., Huang, L., Wang, F., Xiong, H., Huang, W., Dou, D., & Xiong, H. (2024). GIaNt: Protein-Ligand Binding Affinity Prediction via Geometry-Aware Interactive Graph Neural Network. *IEEE Transactions on Knowledge and Data Engineering*, 36(05), 1991–2008.
<https://doi.org/10.1109/TKDE.2023.3314502>
169. Mqawass, G., & Popov, P. (2024). graphLambda: Fusion Graph Neural Networks for Binding Affinity Prediction. *Journal of Chemical Information and Modeling*, 64(7), 2323–2330.
<https://doi.org/10.1021/acs.jcim.3c00771>
170. Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., & Zheng, S. (2022). *TANKBind: Trigonometry-Aware Neural Networks for Drug-Protein Binding Structure Prediction*.
<https://doi.org/10.1101/2022.06.06.495043>

171. Corso, G., Stärk, H., Jing, B., Barzilay, R., & Jaakkola, T. (2023). *DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking* (arXiv:2210.01776). arXiv.
<https://doi.org/10.48550/arXiv.2210.01776>
172. eMolecules. (n.d.). *Buy Research Compounds | Search CAS Number | eMolecules*. Retrieved April 21, 2024, from <https://www.emolecules.com>
173. Bohacek, R. S., McMartin, C., & Guida, W. C. (1996). The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal Research Reviews*, *16*(1), 3–50.
[https://doi.org/10.1002/\(SICI\)1098-1128\(199601\)16:1<3::AID-MED1>3.0.CO;2-6](https://doi.org/10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6)
174. Hall, B. W., & Keiser, M. J. (n.d.). *Retrieval Augmented Docking using Hierarchical Navigable Small Worlds*.
175. Yang, Y., Yao, K., Repasky, M. P., Leswing, K., Abel, R., Shoichet, B. K., & Jerome, S. V. (2021). Efficient Exploration of Chemical Space with Docking and Deep Learning. *Journal of Chemical Theory and Computation*, *17*(11), 7106–7119. <https://doi.org/10.1021/acs.jctc.1c00810>
176. Graff, D. E., Shakhnovich, E. I., & Coley, C. W. (n.d.). Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, *12*(22), 7866–7881.
<https://doi.org/10.1039/d0sc06805e>
177. Thompson, J., Walters, W. P., Feng, J. A., Pabon, N. A., Xu, H., Maser, M., Goldman, B. B., Moustakas, D., Schmidt, M., & York, F. (2022). Optimizing active learning for free energy calculations. *Artificial Intelligence in the Life Sciences*, *2*, 100050.
<https://doi.org/10.1016/j.ails.2022.100050>
178. Bjerrum, E. J., & Threlfall, R. (2017). *Molecular Generation with Recurrent Neural Networks (RNNs)* (arXiv:1705.04612). arXiv. <http://arxiv.org/abs/1705.04612>
179. Segler, M. H. S., Kogej, T., Tyrchan, C., & Waller, M. P. (2018). Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Central Science*, *4*(1), 120–131. <https://doi.org/10.1021/acscentsci.7b00512>

180. Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., & Aspuru-Guzik, A. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2), 268–276. <https://doi.org/10.1021/acscentsci.7b00572>
181. Kusner, M. J., Paige, B., & Hernández-Lobato, J. M. (2017). *Grammar Variational Autoencoder* (arXiv:1703.01925). arXiv. <https://doi.org/10.48550/arXiv.1703.01925>
182. Jin, W., Barzilay, R., & Jaakkola, T. (2019). *Junction Tree Variational Autoencoder for Molecular Graph Generation* (arXiv:1802.04364). arXiv. <http://arxiv.org/abs/1802.04364>
183. Zhavoronkov, A., Ivanenkov, Y. A., Aliper, A., Veselov, M. S., Aladinskiy, V. A., Aladinskaya, A. V., Terentiev, V. A., Polykovskiy, D. A., Kuznetsov, M. D., Asadulaev, A., Volkov, Y., Zholus, A., Shayakhmetov, R. R., Zhebrak, A., Minaeva, L. I., Zagribelnyy, B. A., Lee, L. H., Soll, R., Madge, D., ... Aspuru-Guzik, A. (2019). Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, 37(9), 1038–1040. <https://doi.org/10.1038/s41587-019-0224-x>
184. Mazuz, E., Shtar, G., Shapira, B., & Rokach, L. (2023). Molecule generation using transformers and policy gradient reinforcement learning. *Scientific Reports*, 13(1), 8799. <https://doi.org/10.1038/s41598-023-35648-w>
185. Noutahi, E., Gabellini, C., Craig, M., Lim, J. S. C., & Tossou, P. (2023). *Gotta be SAFE: A New Framework for Molecular Design* (arXiv:2310.10773). arXiv. <http://arxiv.org/abs/2310.10773>
186. Kojima, E., Iimuro, A., Nakajima, M., Kinuta, H., Asada, N., Sako, Y., Nakata, Z., Uemura, K., Arita, S., Miki, S., Wakasa-Morimoto, C., & Tachibana, Y. (2022). Pocket-to-Lead: Structure-Based De Novo Design of Novel Non-peptidic HIV-1 Protease Inhibitors Using the Ligand Binding Pocket as a Template. *Journal of Medicinal Chemistry*, 65(8), 6157–6170. <https://doi.org/10.1021/acs.jmedchem.1c02217>

187. Skalic, M., Jiménez, J., Sabbadin, D., & De Fabritiis, G. (2019). Shape-Based Generative Modeling for de Novo Drug Design. *Journal of Chemical Information and Modeling*, *59*(3), 1205–1214. <https://doi.org/10.1021/acs.jcim.8b00706>
188. Skalic, M., Sabbadin, D., Sattarov, B., Sciabola, S., & De Fabritiis, G. (2019). From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design. *Molecular Pharmaceutics*, *16*(10), 4282–4291. <https://doi.org/10.1021/acs.molpharmaceut.9b00634>
189. Xu, M., Ran, T., & Chen, H. (2021). De Novo Molecule Design Through the Molecular Generative Model Conditioned by 3D Information of Protein Binding Sites. *Journal of Chemical Information and Modeling*, *61*(7), 3240–3254. <https://doi.org/10.1021/acs.jcim.0c01494>
190. Li, Y., Pei, J., & Lai, L. (2021). Structure-based de novo drug design using 3D deep generative models. *Chemical Science*, *12*(41), 13664–13675. <https://doi.org/10.1039/D1SC04444C>
191. Wang, M., Hsieh, C.-Y., Wang, J., Wang, D., Weng, G., Shen, C., Yao, X., Bing, Z., Li, H., Cao, D., & Hou, T. (2022). RELATION: A Deep Generative Model for Structure-Based De Novo Drug Design. *Journal of Medicinal Chemistry*, *65*(13), 9478–9492. <https://doi.org/10.1021/acs.jmedchem.2c00732>
192. Luo, S., Guan, J., Ma, J., & Peng, J. (2022). *A 3D Generative Model for Structure-Based Drug Design* (arXiv:2203.10446). arXiv. <https://doi.org/10.48550/arXiv.2203.10446>
193. McNaughton, A. D., Bontha, M. S., Knutson, C. R., Pope, J. A., & Kumar, N. (2022). *De novo design of protein target specific scaffold-based Inhibitors via Reinforcement Learning* (arXiv:2205.10473). arXiv. <http://arxiv.org/abs/2205.10473>
194. Chan, L., Kumar, R., Verdonk, M., & Poelking, C. (2022). *3D pride without 2D prejudice: Bias-controlled multi-level generative models for structure-based ligand design* (arXiv:2204.10663). arXiv. <http://arxiv.org/abs/2204.10663>
195. Krishnan, S. R., Bung, N., Vangala, S. R., Srinivasan, R., Bulusu, G., & Roy, A. (2022). De Novo Structure-Based Drug Design Using Deep Learning. *Journal of Chemical Information and Modeling*, *62*(21), 5100–5109. <https://doi.org/10.1021/acs.jcim.1c01319>

196. Ünlü, A., Çevrim, E., Sarıgün, A., Çelikbilek, H., Güvenilir, H. A., Koyaş, A., Kahraman, D. C., Olğaç, A., Rifaioğlu, A., & Doğan, T. (2023). *Target Specific De Novo Design of Drug Candidate Molecules with Graph Transformer-based Generative Adversarial Networks* (arXiv:2302.07868). arXiv. <https://doi.org/10.48550/arXiv.2302.07868>
197. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). *Zero-Shot Text-to-Image Generation* (arXiv:2102.12092). arXiv. <http://arxiv.org/abs/2102.12092>
198. Igashov, I., Stärk, H., Vignac, C., Satorras, V. G., Frossard, P., Welling, M., Bronstein, M., & Correia, B. (2022). *Equivariant 3D-Conditional Diffusion Models for Molecular Linker Design* (arXiv:2210.05274). arXiv. <https://doi.org/10.48550/arXiv.2210.05274>
199. Wu, L., Gong, C., Liu, X., Ye, M., & Liu, Q. (2022). *Diffusion-based Molecule Generation with Informative Prior Bridges* (arXiv:2209.00865). arXiv. <https://doi.org/10.48550/arXiv.2209.00865>
200. Lin, H., Huang, Y., Liu, M., Li, X., Ji, S., & Li, S. Z. (2022). *DiffBP: Generative Diffusion of 3D Molecules for Target Protein Binding* (arXiv:2211.11214). arXiv. <https://doi.org/10.48550/arXiv.2211.11214>
201. Schneuing, A., Du, Y., Harris, C., Jamasb, A., Igashov, I., Du, W., Blundell, T., Lió, P., Gomes, C., Welling, M., Bronstein, M., & Correia, B. (2023). *Structure-based Drug Design with Equivariant Diffusion Models* (arXiv:2210.13695). arXiv. <https://doi.org/10.48550/arXiv.2210.13695>
202. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). Analyzing Protein Structure and Function. In *Molecular Biology of the Cell. 4th edition*. Garland Science. <https://www.ncbi.nlm.nih.gov/books/NBK26820/>
203. Smyth, M. S., & Martin, J. H. J. (2000). X Ray crystallography. *Molecular Pathology*, 53(1), 8–14.
204. McPherson, A., & Gavira, J. A. (2013). Introduction to protein crystallization. *Acta Crystallographica. Section F, Structural Biology Communications*, 70(Pt 1), 2–20. <https://doi.org/10.1107/S2053230X13033141>

205. Hu, Y., Cheng, K., He, L., Zhang, X., Jiang, B., Jiang, L., Li, C., Wang, G., Yang, Y., & Liu, M. (2021). NMR-Based Methods for Protein Analysis. *Analytical Chemistry*, 93(4), 1866–1879. <https://doi.org/10.1021/acs.analchem.0c03830>
206. Poulsen, F. M. (n.d.). *Introduction to NMR spectroscopy of proteins*.
207. Savva, C. (2019). A beginner's guide to cryogenic electron microscopy. *The Biochemist*, 41(2), 46–52. <https://doi.org/10.1042/BIO04102046>
208. Dubach, V. R. A., & Guskov, A. (2020). The Resolution in X-ray Crystallography and Single-Particle Cryogenic Electron Microscopy. *Crystals*, 10(7), Article 7. <https://doi.org/10.3390/cryst10070580>
209. Kühlbrandt, W. (2014). The Resolution Revolution. *Science*, 343(6178), 1443–1444. <https://doi.org/10.1126/science.1251652>
210. Wu, M., Lander, G. C., & Herzik, M. A. (2020). Sub-2 Angstrom resolution structure determination using single-particle cryo-EM at 200 keV. *Journal of Structural Biology: X*, 4, 100020. <https://doi.org/10.1016/j.yjsbx.2020.100020>
211. Bank, R. P. D. (n.d.). *PDB Statistics: Number of Released PDB Structures per Year*. Retrieved April 28, 2024, from <https://www.rcsb.org/stats/all-released-structures>
212. Vollmar, M., & Evans, G. (2021). Machine learning applications in macromolecular X-ray crystallography. *Crystallography Reviews*. <https://www.tandfonline.com/doi/abs/10.1080/0889311X.2021.1982914>
213. Chung, J. M., Durie, C. L., & Lee, J. (2022). Artificial Intelligence in Cryo-Electron Microscopy. *Life*, 12(8), 1267. <https://doi.org/10.3390/life12081267>
214. Bruno, A. E., Charbonneau, P., Newman, J., Snell, E. H., So, D. R., Vanhoucke, V., Watkins, C. J., Williams, S., & Wilson, J. (2018). Classification of crystallization outcomes using deep convolutional neural networks. *PLOS ONE*, 13(6), e0198883. <https://doi.org/10.1371/journal.pone.0198883>

215. Souza, A., Oliveira, L. B., Hollatz, S., Feldman, M., Olukotun, K., Holton, J. M., Cohen, A. E., & Nardi, L. (2019). *DeepFreak: Learning Crystallography Diffraction Patterns with Automated Machine Learning* (arXiv:1904.11834). arXiv. <https://doi.org/10.48550/arXiv.1904.11834>
216. Ke, T.-W., Brewster, A. S., Yu, S. X., Ushizima, D., Yang, C., & Sauter, N. K. (2018). A convolutional neural network-based screening tool for X-ray serial crystallography. *Journal of Synchrotron Radiation*, 25(3), 655–670. <https://doi.org/10.1107/S1600577518004873>
217. Yann, M., & Tang, Y. (2016). Learning Deep Convolutional Neural Networks for X-Ray Protein Crystallization Image Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1). <https://doi.org/10.1609/aaai.v30i1.10150>
218. Miyaguchi, I., Sato, M., Kashima, A., Nakagawa, H., Kokabu, Y., Ma, B., Matsumoto, S., Tokuhisa, A., Ohta, M., & Ikeguchi, M. (2021). Machine learning to estimate the local quality of protein crystal structures. *Scientific Reports*, 11(1), 23599. <https://doi.org/10.1038/s41598-021-02948-y>
219. Tickle, I. J. (2012). Statistical quality indicators for electron-density maps. *Acta Crystallographica Section D: Biological Crystallography*, 68(Pt 4), 454–467. <https://doi.org/10.1107/S0907444911035918>
220. Chen, V. B., Arendall, W. B., Headd, J. J., Keedy, D. A., Immormino, R. M., Kapral, G. J., Murray, L. W., Richardson, J. S., & Richardson, D. C. (2010). MolProbity: All-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D: Biological Crystallography*, 66(Pt 1), 12–21. <https://doi.org/10.1107/S0907444909042073>
221. Punjani, A., Rubinstein, J. L., Fleet, D. J., & Brubaker, M. A. (2017). cryoSPARC: Algorithms for rapid unsupervised cryo-EM structure determination. *Nature Methods*, 14(3), 290–296. <https://doi.org/10.1038/nmeth.4169>
222. Maddhuri Venkata Subramaniya, S. R., Terashi, G., & Kihara, D. (2019). Protein secondary structure detection in intermediate-resolution cryo-EM maps using deep learning. *Nature Methods*, 16(9), Article 9. <https://doi.org/10.1038/s41592-019-0500-1>

223. Wang, X., Alnabati, E., Aderinwale, T. W., Maddhuri Venkata Subramaniya, S. R., Terashi, G., & Kihara, D. (2021). Detecting protein and DNA/RNA structures in cryo-EM maps of intermediate resolution using deep learning. *Nature Communications*, *12*(1), 2302.
<https://doi.org/10.1038/s41467-021-22577-3>
224. Si, D., Moritz, S. A., Pfab, J., Hou, J., Cao, R., Wang, L., Wu, T., & Cheng, J. (2020). Deep Learning to Predict Protein Backbone Structure from High-Resolution Cryo-EM Density Maps. *Scientific Reports*, *10*(1), Article 1. <https://doi.org/10.1038/s41598-020-60598-y>
225. *De novo main-chain modeling for EM maps using MAINMAST* | *Nature Communications*. (n.d.). Retrieved April 27, 2024, from <https://www.nature.com/articles/s41467-018-04053-7>
226. Terashi, G., Wang, X., Maddhuri Venkata Subramaniya, S. R., Tesmer, J. J. G., & Kihara, D. (2022). Residue-wise local quality estimation for protein models from cryo-EM maps. *Nature Methods*, *19*(9), 1116–1125.
227. Terashi, G., Wang, X., Prasad, D., Nakamura, T., & Kihara, D. (2024). DeepMainmast: Integrated protocol of protein structure modeling for cryo-EM with deep learning and structure prediction. *Nature Methods*, *21*(1), Article 1. <https://doi.org/10.1038/s41592-023-02099-0>
228. Jamali, K., Käll, L., Zhang, R., Brown, A., Kimanius, D., & Scheres, S. H. W. (2024). Automated model building and protein identification in cryo-EM maps. *Nature*, 1–2.
<https://doi.org/10.1038/s41586-024-07215-4>
229. Borrell, B. (2009). Fraud rocks protein community. *Nature*, *462*(7276), 970–970.
<https://doi.org/10.1038/462970a>
230. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Židek, A., Nelson, A. W. R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., & Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, *577*(7792), 706–710.
<https://doi.org/10.1038/s41586-019-1923-7>

231. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), Article 7873.
<https://doi.org/10.1038/s41586-021-03819-2>
232. Kryshchuk, A., Schwede, T., Topf, M., Fidelis, K., & Moult, J. (2019). Critical Assessment of Methods of Protein Structure Prediction (CASP) – Round XIII. *Proteins*, 87(12), 1011–1020.
<https://doi.org/10.1002/prot.25823>
233. Kryshchuk, A., Schwede, T., Topf, M., Fidelis, K., & Moult, J. (2021). Critical assessment of methods of protein structure prediction (CASP)—Round XIV. *Proteins: Structure, Function, and Bioinformatics*, 89(12), 1607–1617. <https://doi.org/10.1002/prot.26237>
234. Huang, B., Kong, L., Wang, C., Ju, F., Zhang, Q., Zhu, J., Gong, T., Zhang, H., Yu, C., Zheng, W.-M., & Bu, D. (2023). Protein Structure Prediction: Challenges, Advances, and the Shift of Research Paradigms. *Genomics, Proteomics & Bioinformatics*, 21(5), 913–925.
<https://doi.org/10.1016/j.gpb.2022.11.014>
235. *Method of the Year 2021: Protein structure prediction*. (2022, January 11). Nature.
<https://www.nature.com/collections/dfejabhghd>
236. Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., & Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences of the United States of America*, 117(3), 1496–1503.
<https://doi.org/10.1073/pnas.1914677117>
237. Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., ... Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557), 871–876. <https://doi.org/10.1126/science.abj8754>

238. Ahdritz, G., Bouatta, N., Floristean, C., Kadyan, S., Xia, Q., Gerecke, W., O'Donnell, T. J., Berenberg, D., Fisk, I., Zanichelli, N., Zhang, B., Nowaczynski, A., Wang, B., Stepniewska-Dziubinska, M. M., Zhang, S., Ojewole, A., Guney, M. E., Biderman, S., Watkins, A. M., ... AlQuraishi, M. (2023). *OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization* (p. 2022.11.20.517210). bioRxiv.
<https://doi.org/10.1101/2022.11.20.517210>
239. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 1123–1130. <https://doi.org/10.1126/science.ade2574>
240. Chowdhury, R., Bouatta, N., Biswas, S., Rochereau, C., Floristean, C., Ahdritz, G., Zhang, J., Church, G. M., Sorger, P. K., & AlQuraishi, M. (2022). Single-sequence protein structure prediction using language models and deep learning. *Nature Biotechnology*, 40(11), 1617–1623.
<https://doi.org/10.1038/s41587-022-01432-w>
241. Weissenow, K., Heinzinger, M., & Rost, B. (2022). Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. *Structure*, 30(8), 1169-1177.e4.
<https://doi.org/10.1016/j.str.2022.05.001>
242. Akdel, M., Pires, D. E. V., Pardo, E. P., Jänes, J., Zalevsky, A. O., Mészáros, B., Bryant, P., Good, L. L., Laskowski, R. A., Pozzati, G., Shenoy, A., Zhu, W., Kundrotas, P., Serra, V. R., Rodrigues, C. H. M., Dunham, A. S., Burke, D., Borkakoti, N., Velankar, S., ... Beltrao, P. (2022). A structural biology community assessment of AlphaFold2 applications. *Nature Structural & Molecular Biology*, 29(11), 1056–1067. <https://doi.org/10.1038/s41594-022-00849-w>
243. Yang, Z., Zeng, X., Zhao, Y., & Chen, R. (2023). AlphaFold2 and its applications in the fields of biology and medicine. *Signal Transduction and Targeted Therapy*, 8(1), 1–14.
<https://doi.org/10.1038/s41392-023-01381-z>

244. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Židek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., ... Velankar, S. (2022). AlphaFold Protein Structure Database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, *50*(D1), D439–D444.
<https://doi.org/10.1093/nar/gkab1061>
245. Hu, L., Salmen, W., Sankaran, B., Lasanajak, Y., Smith, D. F., Crawford, S. E., Estes, M. K., & Prasad, B. V. V. (2022). Novel fold of rotavirus glycan-binding domain predicted by AlphaFold2 and determined by X-ray crystallography. *Communications Biology*, *5*, 419.
<https://doi.org/10.1038/s42003-022-03357-1>
246. Hutin, S., Ling, W. L., Tarbouriech, N., Schoehn, G., Grimm, C., Fischer, U., & Burmeister, W. P. (2022). The Vaccinia Virus DNA Helicase Structure from Combined Single-Particle Cryo-Electron Microscopy and AlphaFold2 Prediction. *Viruses*, *14*(10), Article 10.
<https://doi.org/10.3390/v14102206>
247. Jin, Y., Fyfe, P. K., Gardner, S., Wilmes, S., Bubeck, D., & Moraga, I. (2022). Structural insights into the assembly and activation of the IL-27 signaling complex. *EMBO Reports*, *23*(10), e55450.
<https://doi.org/10.15252/embr.202255450>
248. Dai, X., Wu, L., Yoo, S., & Liu, Q. (2023). Integrating AlphaFold and deep learning for atomistic interpretation of cryo-EM maps. *Briefings in Bioinformatics*, *24*(6), bbad405.
<https://doi.org/10.1093/bib/bbad405>
249. Wang, S., Lin, H., Huang, Z., He, Y., Deng, X., Xu, Y., Pei, J., & Lai, L. (2022). CavitySpace: A Database of Potential Ligand Binding Sites in the Human Proteome. *Biomolecules*, *12*(7), Article 7. <https://doi.org/10.3390/biom12070967>
250. Velez Rueda, A. J., Bulgarelli, F. L., Palopoli, N., & Parisi, G. (2023). CaviDB: A database of cavities and their features in the structural and conformational space of proteins. *Database*, *2023*, baad010. <https://doi.org/10.1093/database/baad010>

251. Zhang, Y., Vass, M., Shi, D., Abualrous, E., Chambers, J., Chopra, N., Higgs, C., Kasavajhala, K., Li, H., Nandekar, P., Sato, H., Miller, E., Repasky, M., & Jerome, S. (2022). *Benchmarking Refined and Unrefined AlphaFold2 Structures for Hit Discovery*. ChemRxiv. <https://doi.org/10.26434/chemrxiv-2022-kcn0d-v2>
252. Scardino, V., Di Filippo, J. I., & Cavasotto, C. N. (2022). How good are AlphaFold models for docking-based virtual screening? *iScience*, 26(1), 105920. <https://doi.org/10.1016/j.isci.2022.105920>
253. Lyu, J., Kapolka, N., Gumpper, R., Alon, A., Wang, L., Jain, M. K., Barros-Álvarez, X., Sakamoto, K., Kim, Y., DiBerto, J., Kim, K., Tummino, T. A., Huang, S., Irwin, J. J., Tarkhanova, O. O., Moroz, Y., Skinotis, G., Kruse, A. C., Shoichet, B. K., & Roth, B. L. (2024). *AlphaFold2 structures template ligand discovery* (p. 2023.12.20.572662). bioRxiv. <https://doi.org/10.1101/2023.12.20.572662>
254. Ngo, K., Yarov-Yarovoy, V., Clancy, C. E., & Vorobyov, I. (2024). *Harnessing AlphaFold to reveal state secrets: Prediction of hERG closed and inactivated states* (p. 2024.01.27.577468). bioRxiv. <https://doi.org/10.1101/2024.01.27.577468>
255. Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., ... Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 1–3. <https://doi.org/10.1038/s41586-024-07487-w>
256. AlphaFold3—Why did Nature publish it without its code? (2024). *Nature*, 629(8013), 728–728. <https://doi.org/10.1038/d41586-024-01463-0>
257. Huang, P.-S., Boyken, S. E., & Baker, D. (2016). The coming of age of de novo protein design. *Nature*, 537(7620), 320–327. <https://doi.org/10.1038/nature19946>
258. Korendovych, I. V., & DeGrado, W. F. (2020). De novo protein design, a retrospective. *Quarterly Reviews of Biophysics*, 53, e3. <https://doi.org/10.1017/S0033583519000131>

259. Pan, X., & Kortemme, T. (2021). Recent advances in de novo protein design: Principles, methods, and applications. *Journal of Biological Chemistry*, 296. <https://doi.org/10.1016/j.jbc.2021.100558>
260. Kortemme, T. (2024). De novo protein design—From new structures to programmable functions. *Cell*, 187(3), 526–544. <https://doi.org/10.1016/j.cell.2023.12.028>
261. Adhikari, A., Bhattarai, B. R., Aryal, A., Thapa, N., KC, P., Adhikari, A., Maharjan, S., Chanda, P. B., Regmi, B. P., & Parajuli, N. (n.d.). Reprogramming natural proteins using unnatural amino acids. *RSC Advances*, 11(60), 38126–38145. <https://doi.org/10.1039/d1ra07028b>
262. Anishchenko, I., Pellock, S. J., Chidyausiku, T. M., Ramelot, T. A., Ovchinnikov, S., Hao, J., Bafna, K., Norn, C., Kang, A., Bera, A. K., DiMaio, F., Carter, L., Chow, C. M., Montelione, G. T., & Baker, D. (2021). De novo protein design by deep network hallucination. *Nature*, 600(7889), 547–552. <https://doi.org/10.1038/s41586-021-04184-w>
263. Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., ... Baker, D. (2022). Robust deep learning based protein sequence design using ProteinMPNN. *Science (New York, N.Y.)*, 378(6615), 49–56. <https://doi.org/10.1126/science.add2187>
264. Wang, J., Lianza, S., Juergens, D., Tischer, D., Watson, J. L., Castro, K. M., Ragotte, R., Saragovi, A., Milles, L. F., Baek, M., Anishchenko, I., Yang, W., Hicks, D. R., Expòsit, M., Schlichthaerle, T., Chun, J.-H., Dauparas, J., Bennett, N., Wicky, B. I. M., ... Baker, D. (2022). Scaffolding protein functional sites using deep learning. *Science (New York, N.Y.)*, 377(6604), 387–394. <https://doi.org/10.1126/science.abn2100>
265. Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., ... Baker, D. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976), Article 7976. <https://doi.org/10.1038/s41586-023-06415-8>

266. Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., McHugh, R., Vafeados, D., Li, X., Sutherland, G. A., Hitchcock, A., Hunter, C. N., Baek, M., DiMaio, F., & Baker, D. (2023). *Generalized Biomolecular Modeling and Design with RoseTTAFold All-Atom* (p. 2023.10.09.561603). bioRxiv. <https://doi.org/10.1101/2023.10.09.561603>
267. Chen, Z. (2023). Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications*, *10*(1), 1–12. <https://doi.org/10.1057/s41599-023-02079-x>
268. Huang, J., Galal, G., Etemadi, M., & Vaidyanathan, M. (2022). Evaluation and Mitigation of Racial Bias in Clinical Machine Learning Models: Scoping Review. *JMIR Medical Informatics*, *10*(5), e36388. <https://doi.org/10.2196/36388>
269. Grother, P. J., Quinn, G. W., & Phillips, P. J. (2011). *Report on the evaluation of 2D still-image face recognition algorithms* (NIST IR 7709; 0 ed., p. NIST IR 7709). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.IR.7709>
270. Buolamwini, J., & Gebru, T. (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, 77–91. <https://proceedings.mlr.press/v81/buolamwini18a.html>
271. Kelly, S., & Mirpourian, M. (n.d.). *Algorithmic Bias, Financial Inclusion, and Gender*.
272. Gray, A. (2024). *ChatGPT “contamination”: Estimating the prevalence of LLMs in the scholarly literature* (arXiv:2403.16887). arXiv. <https://doi.org/10.48550/arXiv.2403.16887>
273. Liang, W., Izzo, Z., Zhang, Y., Lepp, H., Cao, H., Zhao, X., Chen, L., Ye, H., Liu, S., Huang, Z., McFarland, D. A., & Zou, J. Y. (2024). *Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews* (arXiv:2403.07183). arXiv. <https://doi.org/10.48550/arXiv.2403.07183>

274. Walters, W. H., & Wilder, E. I. (2023). Fabrication and errors in the bibliographic citations generated by ChatGPT. *Scientific Reports*, *13*(1), 14045. <https://doi.org/10.1038/s41598-023-41032-5>
275. Mokrane, S. (n.d.). *The Promise and Perils of Google's Bard for Scientific Research*.
276. Thorbecke, C. (2023, February 8). *Google shares lose \$100 billion after company's AI chatbot makes an error during demo* | CNN Business. CNN. <https://www.cnn.com/2023/02/08/tech/google-ai-bard-demo-error/index.html>
277. Mysinger, M. M., Carchia, M., Irwin, John. J., & Shoichet, B. K. (2012). Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, *55*(14), 6582–6594. <https://doi.org/10.1021/jm300687e>
278. Chen, L., Cruz, A., Ramsey, S., Dickson, C. J., Duca, J. S., Hornak, V., Koes, D. R., & Kurtzman, T. (2019). Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLOS ONE*, *14*(8), e0220113. <https://doi.org/10.1371/journal.pone.0220113>
279. Li, J., Guan, X., Zhang, O., Sun, K., Wang, Y., Bagni, D., & Head-Gordon, T. (2023). Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *ArXiv*, arXiv:2308.09639v1.
280. Kanakala, G. C., Aggarwal, R., Nayar, D., & Priyakumar, U. D. (2023). Latent Biases in Machine Learning Models for Predicting Binding Affinities Using Popular Data Sets. *ACS Omega*, *8*(2), 2389–2397. <https://doi.org/10.1021/acsomega.2c06781>
281. Abbas, U., Chen, J., & Shao, Q. (2023). Assessing Fairness of AlphaFold2 Prediction of Protein 3D Structures. *bioRxiv*, 2023.05.23.542006. <https://doi.org/10.1101/2023.05.23.542006>
282. Terwilliger, T. C., Liebschner, D., Croll, T. I., Williams, C. J., McCoy, A. J., Poon, B. K., Afonine, P. V., Oeffner, R. D., Richardson, J. S., Read, R. J., & Adams, P. D. (2024). AlphaFold predictions are valuable hypotheses and accelerate but do not replace experimental structure determination. *Nature Methods*, *21*(1), 110–116. <https://doi.org/10.1038/s41592-023-02087-4>

283. Perrakis, A., & Sixma, T. K. (2021). AI revolutions in biology. *EMBO Reports*, 22(11), e54046.
<https://doi.org/10.15252/embr.202154046>
284. Urbina, F., Lentzos, F., Invernizzi, C., & Ekins, S. (2022). Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3), 189–191.
<https://doi.org/10.1038/s42256-022-00465-9>
285. Meng, E. C., Goddard, T. D., Pettersen, E. F., Couch, G. S., Pearson, Z. J., Morris, J. H., & Ferrin, T. E. (2023). UCSF ChimeraX: Tools for structure building and analysis. *Protein Science*, 32(11), e4792. <https://doi.org/10.1002/pro.4792>
286. Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., & Steinegger, M. (2022). ColabFold: Making protein folding accessible to all. *Nature Methods*, 19(6), 679–682.
<https://doi.org/10.1038/s41592-022-01488-1>
287. Bellissent-Funel, M.-C., Hassanali, A., Havenith, M., Henchman, R., Pohl, P., Sterpone, F., van der Spoel, D., Xu, Y., & Garcia, A. E. (2016). Water Determines the Structure and Dynamics of Proteins. *Chemical Reviews*, 116(13), 7673–7697. <https://doi.org/10.1021/acs.chemrev.5b00664>
288. Draper, D. E. (2004). A guide to ions and RNA structure. *RNA*, 10(3), 335–343.
<https://doi.org/10.1261/rna.5205404>
289. Lu, Y., Wang, R., Yang, C.-Y., & Wang, S. (2007). Analysis of Ligand-Bound Water Molecules in High-Resolution Crystal Structures of Protein–Ligand Complexes. *Journal of Chemical Information and Modeling*, 47(2), 668–675. <https://doi.org/10.1021/ci6003527>
290. Anderson, A. C. (2003). The Process of Structure-Based Drug Design. *Chemistry & Biology*, 10(9), 787–797. <https://doi.org/10.1016/j.chembiol.2003.09.002>
291. Mobley, D. L., & Dill, K. A. (2009). Binding of Small-Molecule Ligands to Proteins: “What You See” Is Not Always “What You Get.” *Structure*, 17(4), 489–498.
<https://doi.org/10.1016/j.str.2009.02.010>
292. Andreini, C., Bertini, I., Cavallaro, G., Holliday, G. L., & Thornton, J. M. (2008). Metal ions in biological catalysis: From enzyme databases to general principles. *Journal of Biological*

- Inorganic Chemistry: JBIC: A Publication of the Society of Biological Inorganic Chemistry*, 13(8), 1205–1218. <https://doi.org/10.1007/s00775-008-0404-5>
293. Hediger, M. A., Kanai, Y., You, G., & Nussberger, S. (1995). Mammalian ion-coupled solute transporters. *The Journal of Physiology*, 482(suppl), 7–17.
<https://doi.org/10.1113/jphysiol.1995.sp020559>
294. Liebschner, D., Afonine, P. V., Moriarty, N. W., Poon, B. K., Sobolev, O. V., Terwilliger, T. C., & Adams, P. D. (2017). Polder maps: Improving OMIT maps by excluding bulk solvent. *Acta Crystallographica Section D: Structural Biology*, 73(2), 148–157.
<https://doi.org/10.1107/S2059798316018210>
295. Yamashita, K., Palmer, C. M., Burnley, T., & Murshudov, G. N. (2021). Cryo-EM single-particle structure refinement and map calculation using Servalcat. *Acta Crystallographica Section D: Structural Biology*, 77(10), 1282–1291. <https://doi.org/10.1107/S2059798321009475>
296. Wang, J., Liu, Z., Frank, J., & Moore, P. B. (2018). Identification of ions in experimental electrostatic potential maps. *IUCrJ*, 5(4), 375–381. <https://doi.org/10.1107/S2052252518006292>
297. Yamashita, M. M., Wesson, L., Eisenman, G., & Eisenberg, D. (1990). Where metal ions bind in proteins. *Proceedings of the National Academy of Sciences*, 87(15), 5648–5652.
<https://doi.org/10.1073/pnas.87.15.5648>
298. Carugo, O. (2014). Buried chloride stereochemistry in the Protein Data Bank. *BMC Structural Biology*, 14(1), 19. <https://doi.org/10.1186/s12900-014-0019-8>
299. Gucwa, M., Lenkiewicz, J., Zheng, H., Cymborowski, M., Cooper, D. R., Murzyn, K., & Minor, W. (2023). CMM—An enhanced platform for interactive validation of metal binding sites. *Protein Science: A Publication of the Protein Society*, 32(1), e4525. <https://doi.org/10.1002/pro.4525>
300. Handing, K. B., Niedzialkowska, E., Shabalin, I. G., Kuhn, M. L., Zheng, H., & Minor, W. (2018). Characterizing metal binding sites in proteins with X-ray crystallography. *Nature Protocols*, 13(5), 1062–1090. <https://doi.org/10.1038/nprot.2018.018>

301. Zheng, H., Cooper, D. R., Porebski, P. J., Shabalin, I. G., Handing, K. B., & Minor, W. (2017). CheckMyMetal: A macromolecular metal-binding validation tool. *Acta Crystallographica Section D: Structural Biology*, 73(3), 223–233. <https://doi.org/10.1107/S2059798317001061>
302. Zheng, H., Chordia, M. D., Cooper, D. R., Chruszcz, M., Müller, P., Sheldrick, G. M., & Minor, W. (2014). Validation of metal-binding sites in macromolecular structures with the CheckMyMetal web server. *Nature Protocols*, 9(1), 156–170. <https://doi.org/10.1038/nprot.2013.172>
303. Prisant, M. G., Williams, C. J., Chen, V. B., Richardson, J. S., & Richardson, D. C. (2020). New tools in MolProbity validation: CaBLAM for CryoEM backbone, UnDowser to rethink “waters,” and NGL Viewer to recapture online 3D graphics. *Protein Science*, 29(1), 315–329. <https://doi.org/10.1002/pro.3786>
304. Doerr, S., Majewski, M., Pérez, A., Krämer, A., Clementi, C., Noe, F., Giorgino, T., & De Fabritiis, G. (2021). TorchMD: A Deep Learning Framework for Molecular Simulations. *Journal of Chemical Theory and Computation*, 17(4), 2355–2363. <https://doi.org/10.1021/acs.jctc.0c01343>
305. Wang, X., Terashi, G., & Kihara, D. (2023). CryoREAD: De novo structure modeling for nucleic acids in cryo-EM maps using deep learning. *Nature Methods*, 20(11), Article 11. <https://doi.org/10.1038/s41592-023-02032-5>
306. He, J., Li, T., & Huang, S.-Y. (2023). Improvement of cryo-EM maps by simultaneous local and non-local deep learning. *Nature Communications*, 14(1), Article 1. <https://doi.org/10.1038/s41467-023-39031-1>
307. Nakamura, T., Wang, X., Terashi, G., & Kihara, D. (2023). DAQ-Score Database: Assessment of map–model compatibility for protein structure models from cryo-EM maps. *Nature Methods*, 20(6), Article 6. <https://doi.org/10.1038/s41592-023-01876-1>
308. Sun, K., Hu, X., Feng, Z., Wang, H., Lv, H., Wang, Z., Zhang, G., Xu, S., & You, X. (2022). Predicting Ca²⁺ and Mg²⁺ ligand binding sites by deep neural network algorithm. *BMC Bioinformatics*, 22(12), 324. <https://doi.org/10.1186/s12859-021-04250-0>

309. *Metal3D: a general deep learning framework for accurate metal ion location prediction in proteins* | *Nature Communications*. (n.d.). Retrieved February 25, 2024, from <https://www.nature.com/articles/s41467-023-37870-6>
310. Mohamadi, A., Cheng, T., Jin, L., Wang, J., Sun, H., & Koochi-Moghadam, M. (2022). An ensemble 3D deep-learning model to predict protein metal-binding site. *Cell Reports Physical Science*, 3(9), 101046. <https://doi.org/10.1016/j.xcrp.2022.101046>
311. Boldini, D., Ballabio, D., Consonni, V., Todeschini, R., Grisoni, F., & Sieber, S. (2023). *Effectiveness of molecular fingerprints for exploring the chemical space of natural products* [Preprint]. Chemistry. <https://doi.org/10.26434/chemrxiv-2023-0m355-v2>
312. Willett, P. (2006). Similarity-based virtual screening using 2D fingerprints. *Drug Discovery Today*, 11(23), 1046–1053. <https://doi.org/10.1016/j.drudis.2006.10.005>
313. Soares, T. A., Nunes-Alves, A., Mazzolari, A., Ruggiu, F., Wei, G.-W., & Merz, K. (2022). The (Re)-Evolution of Quantitative Structure–Activity Relationship (QSAR) Studies Propelled by the Surge of Machine Learning Methods. *Journal of Chemical Information and Modeling*, 62(22), 5317–5320. <https://doi.org/10.1021/acs.jcim.2c01422>
314. Sánchez-Cruz, N., Medina-Franco, J. L., Mestres, J., & Barril, X. (2021). Extended connectivity interaction features: Improving binding affinity prediction through chemical description. *Bioinformatics*, 37(10), 1376–1382. <https://doi.org/10.1093/bioinformatics/btaa982>
315. Gainza, P., Sverrisson, F., Monti, F., Rodolà, E., Boscaini, D., Bronstein, M. M., & Correia, B. E. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2), 184–192. <https://doi.org/10.1038/s41592-019-0666-6>
316. Lyu, J., Wang, S., Balias, T. E., Singh, I., Levit, A., Moroz, Y. S., O’Meara, M. J., Che, T., Alga, E., Tolmachova, K., Tolmachev, A. A., Shoichet, B. K., Roth, B. L., & Irwin, J. J. (2019). Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743), 224–229.
317. Brewerton, S. C. (2008). The use of protein-ligand interaction fingerprints in docking. *Current Opinion in Drug Discovery & Development*, 11(3), 356–364.

318. Fassio, A. V., Shub, L., Ponzoni, L., McKinley, J., O'Meara, M. J., Ferreira, R. S., Keiser, M. J., & de Melo Minardi, R. C. (2022). Prioritizing Virtual Screening with Interpretable Interaction Fingerprints. *Journal of Chemical Information and Modeling*, 62(18), 4300–4318.
319. Yilmaz, S. F., & Kozat, S. S. (2020). *Unsupervised Anomaly Detection via Deep Metric Learning with End-to-End Optimization* (arXiv:2005.05865). arXiv. <http://arxiv.org/abs/2005.05865>
320. Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Säckinger, E., & Shah, R. (1993). Signature verification using a “siamese” time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 07(04), 669–688.
<https://doi.org/10.1142/S0218001493000339>
321. Kaya, M., & Bilge, H. (2019). Deep Metric Learning: A Survey. *Symmetry*, 11, 1066.
<https://doi.org/10.3390/sym11091066>
322. Coupry, D. E., & Pogány, P. (2022). Application of deep metric learning to molecular graph similarity. *Journal of Cheminformatics*, 14(1), 11. <https://doi.org/10.1186/s13321-022-00595-7>
323. Wu, F., Courty, N., Jin, S., & Li, S. Z. (2023). Improving molecular representation learning with metric learning-enhanced optimal transport. *Patterns*, 4(4), 100714.
<https://doi.org/10.1016/j.patter.2023.100714>
324. Ge, W., Huang, W., Dong, D., & Scott, M. R. (2018). *Deep Metric Learning with Hierarchical Triplet Loss* (arXiv:1810.06951). arXiv. <https://doi.org/10.48550/arXiv.1810.06951>
325. Hoffer, E., & Ailon, N. (2018). *Deep metric learning using Triplet network* (arXiv:1412.6622). arXiv. <http://arxiv.org/abs/1412.6622>
326. Ding, S., Lin, L., Wang, G., & Chao, H. (2015). *Deep Feature Learning with Relative Distance Comparison for Person Re-identification* (arXiv:1512.03622). arXiv.
<http://arxiv.org/abs/1512.03622>
327. Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-Richardson, O. (2020). *Captum: A unified and*

- generic model interpretability library for PyTorch* (arXiv:2009.07896). arXiv.
<http://arxiv.org/abs/2009.07896>
328. Sundararajan, M., Taly, A., & Yan, Q. (2017). *Axiomatic Attribution for Deep Networks* (arXiv:1703.01365). arXiv. <http://arxiv.org/abs/1703.01365>
329. Laitaoja, M., Valjakka, J., & Jänis, J. (2013). Zinc coordination spheres in protein structures. *Inorganic Chemistry*, 52(19), 10983–10991.
330. Piovesan, D., Profiti, G., Martelli, P. L., & Casadio, R. (2012). The human “magnesome”: Detecting magnesium binding sites on human proteins. *BMC Bioinformatics*, 13 Suppl 14(Suppl 14), S10.
331. Kirberger, M., Wang, X., Deng, H., Yang, W., Chen, G., & Yang, J. J. (2008). Statistical analysis of structural characteristics of protein Ca²⁺-binding sites. *Journal of Biological Inorganic Chemistry: JBIC: A Publication of the Society of Biological Inorganic Chemistry*, 13(7), 1169–1181.
332. Israeli, H., Degtjarik, O., Fierro, F., Chunilal, V., Gill, A. K., Roth, N. J., Botta, J., Prabakar, V., Peleg, Y., Chan, L. F., Ben-Zvi, D., McCormick, P. J., Niv, M. Y., & Shalev-Benami, M. (2021). Structure reveals the activation mechanism of the MC4 receptor to initiate satiation signaling. *Science*, 372(6544), 808–814. <https://doi.org/10.1126/science.abf7958>
333. Heyder, N. A., Kleinau, G., Speck, D., Schmidt, A., Paisdzior, S., Szczepek, M., Bauer, B., Koch, A., Gallandi, M., Kwiatkowski, D., Bürger, J., Mielke, T., Beck-Sickinger, A. G., Hildebrand, P. W., Spahn, C. M. T., Hilger, D., Schacherl, M., Biebermann, H., Hilal, T., ... Scheerer, P. (2021). Structures of active melanocortin-4 receptor–Gs-protein complexes with NDP- α -MSH and setmelanotide. *Cell Research*, 31(11), Article 11. <https://doi.org/10.1038/s41422-021-00569-8>
334. Zhang, H., Chen, L.-N., Yang, D., Mao, C., Shen, Q., Feng, W., Shen, D.-D., Dai, A., Xie, S., Zhou, Y., Qin, J., Sun, J.-P., Scharf, D. H., Hou, T., Zhou, T., Wang, M.-W., & Zhang, Y. (2021). Structural insights into ligand recognition and activation of the melanocortin-4 receptor. *Cell Research*, 31(11), Article 11. <https://doi.org/10.1038/s41422-021-00552-3>

335. Yu, J., Gimenez, L. E., Hernandez, C. C., Wu, Y., Wein, A. H., Han, G. W., McClary, K., Mittal, S. R., Burdsall, K., Stauch, B., Wu, L., Stevens, S. N., Peisley, A., Williams, S. Y., Chen, V., Millhauser, G. L., Zhao, S., Cone, R. D., & Stevens, R. C. (2020). Determination of the melanocortin-4 receptor structure identifies Ca²⁺ as a cofactor for ligand binding. *Science*, *368*(6489), 428–433. <https://doi.org/10.1126/science.aaz8995>
336. Yip, K. M., Fischer, N., Paknia, E., Chari, A., & Stark, H. (2020). Atomic-resolution protein structure determination by cryo-EM. *Nature*, *587*(7832), Article 7832. <https://doi.org/10.1038/s41586-020-2833-4>
337. Maki-Yonekura, S., Kawakami, K., Takaba, K., Hamaguchi, T., & Yonekura, K. (2023). Measurement of charges and chemical bonding in a cryo-EM structure. *Communications Chemistry*, *6*(1), Article 1. <https://doi.org/10.1038/s42004-023-00900-x>
338. Nakane, T., Kotecha, A., Sente, A., McMullan, G., Masiulis, S., Brown, P. M. G. E., Grigoras, I. T., Malinauskaite, L., Malinauskas, T., Miehl, J., Uchański, T., Yu, L., Karia, D., Pechnikova, E. V., de Jong, E., Keizer, J., Bischoff, M., McCormack, J., Tiemeijer, P., ... Scheres, S. H. W. (2020). Single-particle cryo-EM at atomic resolution. *Nature*, *587*(7832), Article 7832. <https://doi.org/10.1038/s41586-020-2829-0>
339. Zhang, K., Pintilie, G. D., Li, S., Schmid, M. F., & Chiu, W. (2020). Resolving individual atoms of protein complex by cryo-electron microscopy. *Cell Research*, *30*(12), Article 12. <https://doi.org/10.1038/s41422-020-00432-2>
340. Zheng, W., Sun, F., Bartlam, M., Li, X., Li, R., & Rao, Z. (2007). The Crystal Structure of Human Isopentenyl Diphosphate Isomerase at 1.7 Å Resolution Reveals its Catalytic Mechanism in Isoprenoid Biosynthesis. *Journal of Molecular Biology*, *366*(5), 1447–1458. <https://doi.org/10.1016/j.jmb.2006.12.055>
341. Maggiora, G., Vogt, M., Stumpfe, D., & Bajorath, J. (2014). Molecular Similarity in Medicinal Chemistry. *Journal of Medicinal Chemistry*, *57*(8), 3186–3204. <https://doi.org/10.1021/jm401411z>

342. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (n.d.). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research: JMLR*.
343. Musgrave, K., Belongie, S., & Lim, S.-N. (2020). *PyTorch Metric Learning* (arXiv:2008.09164). arXiv. <https://doi.org/10.48550/arXiv.2008.09164>
344. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (arXiv:1912.01703). arXiv. <http://arxiv.org/abs/1912.01703>
345. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework* (arXiv:1907.10902). arXiv. <http://arxiv.org/abs/1907.10902>
346. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), Article 3. <https://doi.org/10.1038/s41592-019-0686-2>
347. Lionta, E., Spyrou, G., Vassilatis, D. K., & Cournia, Z. (2014). Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. *Current Topics in Medicinal Chemistry*, 14(16), 1923–1938. <https://doi.org/10.2174/1568026614666140929124445>
348. Kontoyianni, M. (2017). Docking and Virtual Screening in Drug Discovery. In I. M. Lazar, M. Kontoyianni, & A. C. Lazar (Eds.), *Proteomics for Drug Discovery: Methods and Protocols* (pp. 255–266). Springer. https://doi.org/10.1007/978-1-4939-7201-2_18

349. Hughes, J., Rees, S., Kalindjian, S., & Philpott, K. (2011). Principles of early drug discovery. *British Journal of Pharmacology*, *162*(6), 1239–1249. <https://doi.org/10.1111/j.1476-5381.2010.01127.x>
350. Maia, E. H. B., Assis, L. C., de Oliveira, T. A., da Silva, A. M., & Taranto, A. G. (2020). Structure-Based Virtual Screening: From Classical to Artificial Intelligence. *Frontiers in Chemistry*, *8*. <https://doi.org/10.3389/fchem.2020.00343>
351. Lyu, J., Irwin, J. J., & Shoichet, B. K. (2023). Modeling the expansion of virtual screening libraries. *Nature Chemical Biology*, *19*(6), 712–718. <https://doi.org/10.1038/s41589-022-01234-w>
352. Alon, A., Lyu, J., Braz, J. M., Tummino, T. A., Craik, V., O'Meara, M. J., Webb, C. M., Radchenko, D. S., Moroz, Y. S., Huang, X.-P., Liu, Y., Roth, B. L., Irwin, J. J., Basbaum, A. I., Shoichet, B. K., & Kruse, A. C. (2021). Structures of the σ_2 receptor enable docking for bioactive ligand discovery. *Nature*, *600*(7890), 759–764. <https://doi.org/10.1038/s41586-021-04175-x>
353. Fink, E. A., Xu, J., Hübner, H., Braz, J. M., Seemann, P., Avet, C., Craik, V., Weikert, D., Schmidt, M. F., Webb, C. M., Tolmachova, N. A., Moroz, Y. S., Huang, X.-P., Kalyanaraman, C., Gahbauer, S., Chen, G., Liu, Z., Jacobson, M. P., Irwin, J. J., ... Gmeiner, P. (2022). Structure-based discovery of nonopioid analgesics acting through the α_2A -adrenergic receptor. *Science*, *377*(6614), eabn7065. <https://doi.org/10.1126/science.abn7065>
354. Fink, E. A., Bardine, C., Gahbauer, S., Singh, I., Detomasi, T. C., White, K., Gu, S., Wan, X., Chen, J., Ary, B., Glenn, I., O'Connell, J., O'Donnell, H., Fajtová, P., Lyu, J., Vigneron, S., Young, N. J., Kondratov, I. S., Alisoltani, A., ... Craik, C. S. (2023). Large library docking for novel SARS-CoV-2 main protease non-covalent and covalent inhibitors. *Protein Science*, *32*(8), e4712. <https://doi.org/10.1002/pro.4712>
355. Singh, I., Li, F., Fink, E. A., Chau, I., Li, A., Rodriguez-Hernández, A., Glenn, I., Zapatero-Belinchón, F. J., Rodriguez, M. L., Devkota, K., Deng, Z., White, K., Wan, X., Tolmachova, N. A., Moroz, Y. S., Kaniskan, H. Ü., Ott, M., García-Sastre, A., Jin, J., ... Shoichet, B. K. (2023). Structure-Based Discovery of Inhibitors of the SARS-CoV-2 Nsp14 N7-Methyltransferase.

- Journal of Medicinal Chemistry*, 66(12), 7785–7803.
<https://doi.org/10.1021/acs.jmedchem.2c02120>
356. *Keeping pace with the explosive growth of chemical libraries with structure-based virtual screening—Kuan—2023—WIREs Computational Molecular Science—Wiley Online Library.* (n.d.). Retrieved April 22, 2024, from <https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wcms.1678>
357. *GigadockTM Rapidly Identifies Novel Chemical Entities for GPCR Targets.* (n.d.). Retrieved April 22, 2024, from <https://www.eyesopen.com/news/openeye-helps-beacon-discovery-increase-speed-and-improve-virtual-screening>
358. Santos-Martins, D., Solis-Vasquez, L., Tillack, A. F., Sanner, M. F., Koch, A., & Forli, S. (2021). Accelerating AutoDock4 with GPUs and Gradient-Based Local Search. *Journal of Chemical Theory and Computation*, 17(2), 1060–1073. <https://doi.org/10.1021/acs.jctc.0c01006>
359. Yu, Y., Cai, C., Zhu, Z., & Zheng, H. (2022). *Uni-Dock: A GPU-Accelerated Docking Program Enables Ultra-Large Virtual Screening.* <https://doi.org/10.26434/chemrxiv-2022-5t5ts>
360. Settles, B. (2011). From Theories to Queries: Active Learning in Practice. *Active Learning and Experimental Design Workshop In Conjunction with AISTATS 2010*, 1–18.
<https://proceedings.mlr.press/v16/settles11a.html>
361. *Human-in-the-loop machine learning: A state of the art | Artificial Intelligence Review.* (n.d.). Retrieved April 22, 2024, from <https://link.springer.com/article/10.1007/s10462-022-10246-w>
362. Ripphausen, P., Stumpfe, D., & Bajorath, J. (2012). Analysis of structure-based virtual screening studies and characterization of identified active compounds. *Future Medicinal Chemistry*, 4(5), 603–613. <https://doi.org/10.4155/fmc.12.18>
363. Schulz-Gasch, T., & Stahl, M. (2004). Scoring functions for protein–ligand interactions: A critical perspective. *Drug Discovery Today: Technologies*, 1(3), 231–239.
<https://doi.org/10.1016/j.ddtec.2004.08.004>

364. Fischer, A., Smieško, M., Sellner, M., & Lill, M. A. (2021). Decision Making in Structure-Based Drug Discovery: Visual Inspection of Docking Results. *Journal of Medicinal Chemistry*, *64*(5), 2489–2500. <https://doi.org/10.1021/acs.jmedchem.0c02227>
365. Choung, O.-H., Vianello, R., Segler, M., Stiefl, N., & Jiménez-Luna, J. (2023). Extracting medicinal chemistry intuition via preference machine learning. *Nature Communications*, *14*(1), 6651. <https://doi.org/10.1038/s41467-023-42242-1>
366. Gomez, L. (2018). Decision Making in Medicinal Chemistry: The Power of Our Intuition. *ACS Medicinal Chemistry Letters*, *9*(10), 956–958. <https://doi.org/10.1021/acsmedchemlett.8b00359>
367. O’Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, *3*(1), 33. <https://doi.org/10.1186/1758-2946-3-33>
368. van Tilborg, D., Alenicheva, A., & Grisoni, F. (2022). Exposing the Limitations of Molecular Machine Learning with Activity Cliffs. *Journal of Chemical Information and Modeling*, *62*(23), 5938–5951. <https://doi.org/10.1021/acs.jcim.2c01073>
369. Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., & Green, W. H. (2019). *Evaluating Scalable Uncertainty Estimation Methods for DNN-Based Molecular Property Prediction* (arXiv:1910.03127). arXiv. <http://arxiv.org/abs/1910.03127>
370. Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R., & Coley, C. W. (2020). Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *Journal of Chemical Information and Modeling*, *60*(8), 3770–3780. <https://doi.org/10.1021/acs.jcim.0c00502>
371. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles* (arXiv:1612.01474). arXiv. <http://arxiv.org/abs/1612.01474>
372. Gal, Y., & Ghahramani, Z. (n.d.). *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*.

373. Sheridan, R. P., Feuston, B. P., Maiorov, V. N., & Kearsley, S. K. (2004). Similarity to Molecules in the Training Set Is a Good Discriminator for Prediction Accuracy in QSAR. *Journal of Chemical Information and Computer Sciences*, 44(6), 1912–1928. <https://doi.org/10.1021/ci049782w>
374. Liu, R., & Wallqvist, A. (2019). Molecular Similarity-Based Domain Applicability Metric Efficiently Identifies Out-of-Domain Compounds. *Journal of Chemical Information and Modeling*, 59(1), 181–189. <https://doi.org/10.1021/acs.jcim.8b00597>
375. Settles, B. (n.d.). *Active Learning Literature Survey*.
376. Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, 1070. <https://doi.org/10.3115/1613715.1613855>
377. Sharma, M., & Bilgic, M. (2017). Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, 31(1), 164–202. <https://doi.org/10.1007/s10618-016-0460-3>
378. Gusev, F., Gutkin, E., Kurnikova, M. G., & Isayev, O. (n.d.). *Active learning guided drug design lead optimization based on relative binding free energy modeling*.
379. Rego, N., & Koes, D. (2015). 3Dmol.js: Molecular visualization with WebGL. *Bioinformatics*, 31(8), 1322–1324. <https://doi.org/10.1093/bioinformatics/btu829>
380. *celery/celery: Distributed Task Queue (development branch)*. (n.d.). Retrieved April 22, 2024, from <https://github.com/celery/celery>
381. *Redis/redis*. (2024). [C]. Redis. <https://github.com/redis/redis> (Original work published 2009)
382. Kurtzer, G. M., Sochat, V., & Bauer, M. W. (2017). Singularity: Scientific containers for mobility of compute. *PLOS ONE*, 12(5), e0177459. <https://doi.org/10.1371/journal.pone.0177459>
383. Kurtzer, G. M., cclerget, Bauer, M., Kaneshiro, I., Trudgian, D., & Godlove, D. (2021). *hpcng/singularity: Singularity 3.7.3 (v3.7.3) [Computer software]*. Zenodo. <https://doi.org/10.5281/zenodo.4667718>

384. Carlsson, J., Yoo, L., Gao, Z.-G., Irwin, J. J., Shoichet, B. K., & Jacobson, K. A. (2010). Structure-Based Discovery of A2A Adenosine Receptor Ligands. *Journal of Medicinal Chemistry*, 53(9), 3748–3755. <https://doi.org/10.1021/jm100240h>
385. Stein, R. M., Yang, Y., Balius, T. E., O’Meara, M. J., Lyu, J., Young, J., Tang, K., Shoichet, B. K., & Irwin, J. J. (2021). Property-Unmatched Decoys in Docking Benchmarks. *Journal of Chemical Information and Modeling*, 61(2), 699–714. <https://doi.org/10.1021/acs.jcim.0c00598>
386. Lee, S., Kim, H., & Lee, J. (2021). *GradDiv: Adversarial Robustness of Randomized Neural Networks via Gradient Diversity Regularization* (arXiv:2107.02425). arXiv. <http://arxiv.org/abs/2107.02425>
387. Bero, S. A., Muda, A. K., Choo, Y. H., Muda, N. A., & Pratama, S. F. (2017). Similarity Measure for Molecular Structure: A Brief Review. *Journal of Physics: Conference Series*, 892, 012015. <https://doi.org/10.1088/1742-6596/892/1/012015>
388. Tran, K., Neiswanger, W., Yoon, J., Zhang, Q., Xing, E., & Ulissi, Z. W. (2020). *Methods for comparing uncertainty quantifications for material property predictions* (arXiv:1912.10066). arXiv. <http://arxiv.org/abs/1912.10066>

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

DocuSigned by:

C91A9F274F9E408... Author Signature

5/24/2024
Date