Development and Application of CDOCKER Docking Methodology

by

Yujin Wu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Chemistry)
in the University of Michigan
2022

Doctoral Committee:

Professor Charles L. Brooks III, Chair
Assistant Professor Timothy Cernak
Professor Anna K. Mapp
Professor Neil Marsh

Yujin Wu

wyujin@umich.edu

ORCID ID: 0000-0003-2773-9262

ACKNOWLEDGMENTS

First and foremost I would like to thank my thesis advisor professor Charles L. Brooks III for the guidance and help in all of my research projects. I have greatly benefited from his expertise and enthusiasm that consistenly provided great advice to me. Much of the work presented in the dissertation would have never happened without his guidance. I also want to express my gratitude to my dissertation committee – Neil Marsh, Anna Mapp, Timothy Cernak, Aaron Frank and Heather Carlson – for their suggestions on the research projects. I would also like to thank Professor Matthew Soellner together with Professor Anna Mapp for the amazing collaboration work on the TMPRSS2 project. I also want to thank Professor Brian Coppola for his help and guidance during my study here. I had a great time doing research in the Brooks lab and received enormous help from many Brooks lab members including Xinqiang Ding, Amanda Peiffer, Sara Tweedy, Josh Buckner, Xiaorong Liu, Ryan Hayes, Jonah Vilseck, Yanming Wang, Kathleen Dyki, Nicholas O'Hair and David Braun. I also want to thank Katie Foster, the chemistry doctoral program coordinator, for her help throughout my training. Finally, I want to thank my parents and my girlfriend for their patience and support.

Contents

List of Tables

List of Figures

ABSTRACT

The binding of small molecule ligands to receptor targets is important to numerous biological processes. When multiple ligands with different sizes are docked to a target receptor it is reasonable to assume the residues in the binding pocket may adopt alternative conformations. It has also been suggested that the entropic contribution to binding can be important. In the work presented here, we discuss a new physics-based scoring function that includes both enthalpic and entropic contributions to binding that consider the conformational variability of the flexible receptor side chains within the ensemble of docked poses. To accommodate the additional searching requirements for flexible receptor docking studies, we also describe a novel hybrid searching algorithm that combines both molecular dynamics (MD) based simulated annealing and a continuous genetic algorithm. These advances further the development of the Flexible CDOCKER docking methods in the CHARMM software package. We benchmark our developments using 6 varied receptor targets: thrombin, dihydrofolate reductase (DHFR), T4 L99A, T4 L99A/M102Q, PDE10A and the riboswitch. These cover a wide range of enzyme classes with different binding pocket environments and the novel aspects of RNA receptor targets. We demonstrate improved accuracy in flexible cross-docking experiments compared with rigid cross-docking. The largest improvement in top ranking accuracy is 22% for ligands binding to DHFR. Moreover, by using advances in GPU accelerated sampling, we reduce the wall-time required for such calculations so that flexible receptor ligand docking can be applied in high-throughput experiments.

As a practical example of this methodology, we worked with a team of experimental colleagues to identify potential therapeutics for the host transmembrane serine protease TMPRSS2, a promising antiviral target that plays a direct role in SARS-CoV-2 infections. We screened a database of 134,109 molecules that were collected from multiple sources of

molecules in clinical trials, on approved drug lists or readily available in local libraries. These compounds were filtered based on pharmacophore similarity. A total of 4,308 candidates were identified and docked with the flexible docking method noted above. From our in silico docking trials, novel non-covalent inhibitors were identified and verified with biochemical assays.

Targeted covalent inhibitors (TCIs) are an important component in the toolbox of drug discovery and about 30% of currently marketed drugs are TCIs. Although these drugs raise concerns about toxicity, their high potencies and prolonged effects result in less-frequent drug dosing and wide therapeutic margins for patients. On the other hand, a recent study shows that one of the known inhibitors, Nafamostat, forms a covalent bond with TMPRSS2. This motivated us to expand the application of Rigid CDOCKER to tethered docking. We describe the implementation and testing of a covalent docking methodology in Rigid CDOCKER and the optimization of the corresponding physics-based scoring function with an additional customizable covalent bond grid potential which represents the free energy change of bond formation between the ligand and the receptor. We show that our new covalent docking algorithm has comparable pose prediction accuracy with reduced computational cost and could identify lead compounds among a large chemical space.

Finally, leveraging the popularity and utility of the Python language, we introduce the python functionality in the CDOCKER family (i.e., pyCHARMM CDOCKER). This further accelerates CDOCKER docking algorithms and broadens the potential users who want to perform standard CDOCKER docking experiments with little knowledge of docking or the CDOCKER family.

Keywords:

- ***CHARMM:*** Chemistry at Harvard Macromolecular Mechanics (CHARMM) is the name of a widely used set of force fields for molecular dynamics, and the name for the molecular dynamics simulation and analysis computer software package associated with them.
- ***Docking:*** (Molecular) docking is a computational technique that places a small molecule (ligand) in the binding pocket (receptor) and estimates its binding affinity.

- ***CDOCKER:*** CHARMM input script based docking method.
- ***Rigid receptor docking:*** Receptor structure is fixed during docking.
- ***Flexible receptor docking:*** Typically, flexible receptor docking allows conformational changes of the binding pocket during docking.
- ***Covalent docking:*** Docking experiments designed for covalent inhibitors.
- ***Native-like poses:*** Docking poses that is within the RMSD cutoff with respect to the crystal structure. The RMSD cutoff is 2.0 for rigid receptor docking and 2.5 for flexible receptor docking.
- ***Docking accuracy:*** Docking accuracy, or top ranking accuracy, is defined as the percentage of docking experiments that successfully identifies native-like poses as the best pose.
- ***pyCHARMM:*** CHARMM python package.

Chapter 1

Introduction, Applications, and Outstanding Questions of Molecular Docking

The binding of small-molecule ligands to protein or nucleic acid targets is important to numerous biological processes. Accurate prediction of the binding modes between a ligand and a macromolecule is of fundamental importance in structure-based structure-function exploration. The field of (molecular) docking has emerged in late 1990s driven by the needs of structural molecular biology and structure-based drug discovery.[1] Generally speaking, docking predicts the orientation and conformation of a small molecule (ligand) in the binding site of the target protein and estimates its binding affinity.[2–4] It has been greatly facilitated by the dramatic growth in availability and power of computers, and the growing ease of access to small molecule and protein databases.[5]

1.1 An Overview of Docking Models and the Corresponding Application Scenario

1.1.1 Comparison of different docking models

Based on the molecular recognition modeling, molecular docking can be separated into three categories. Analogies to the "lock-and-key" concept to describe the specific interactions between ligands and receptors were first put forth by Emil Fischer.[6] In the rigid receptor docking methods, the internal configuration of the binding pocket is fixed, and the ligands need to possess specific complementary geometric shapes that fit exactly into the binding pocket. The ligand conformation and orientation then can be sampled in two different ways. In cases of rigid receptor – rigid ligand docking methods (rigid ligand docking methods), such as CDOCKER FFT[7] and DOCK3.7,[5] the ligand conformation is pre-computed and fixed during docking. This approach greatly simplifies the docking problem by reducing the number of degrees of freedom from several thousand to only six so that the docking problem can be considered as looking for structure complementarity between two rigid bodies.[5,7,8] However, one might not always sample the ligand bound conformation prior to docking. This results in the development of rigid receptor – flexible ligand docking methods (rigid receptor docking methods). Such a method allows the ligand to explore its conformation space and optimize its configuration upon docking.[6,9–11]

On the other hand, as shown in Figure 1.1A, when ligands with different sizes and shapes bind to the same binding pocket, it is reasonable to assume the binding pocket may undergo conformational changes to better accommodate the ligands (i.e., the induced fit theory).[6] The development of flexible receptor – flexible ligand docking methods (flexible receptor docking methods) is an attempt to sample both receptor and ligand conformational space upon binding and has demonstrated improved performance in pose prediction and lead compound identification.[12–14]

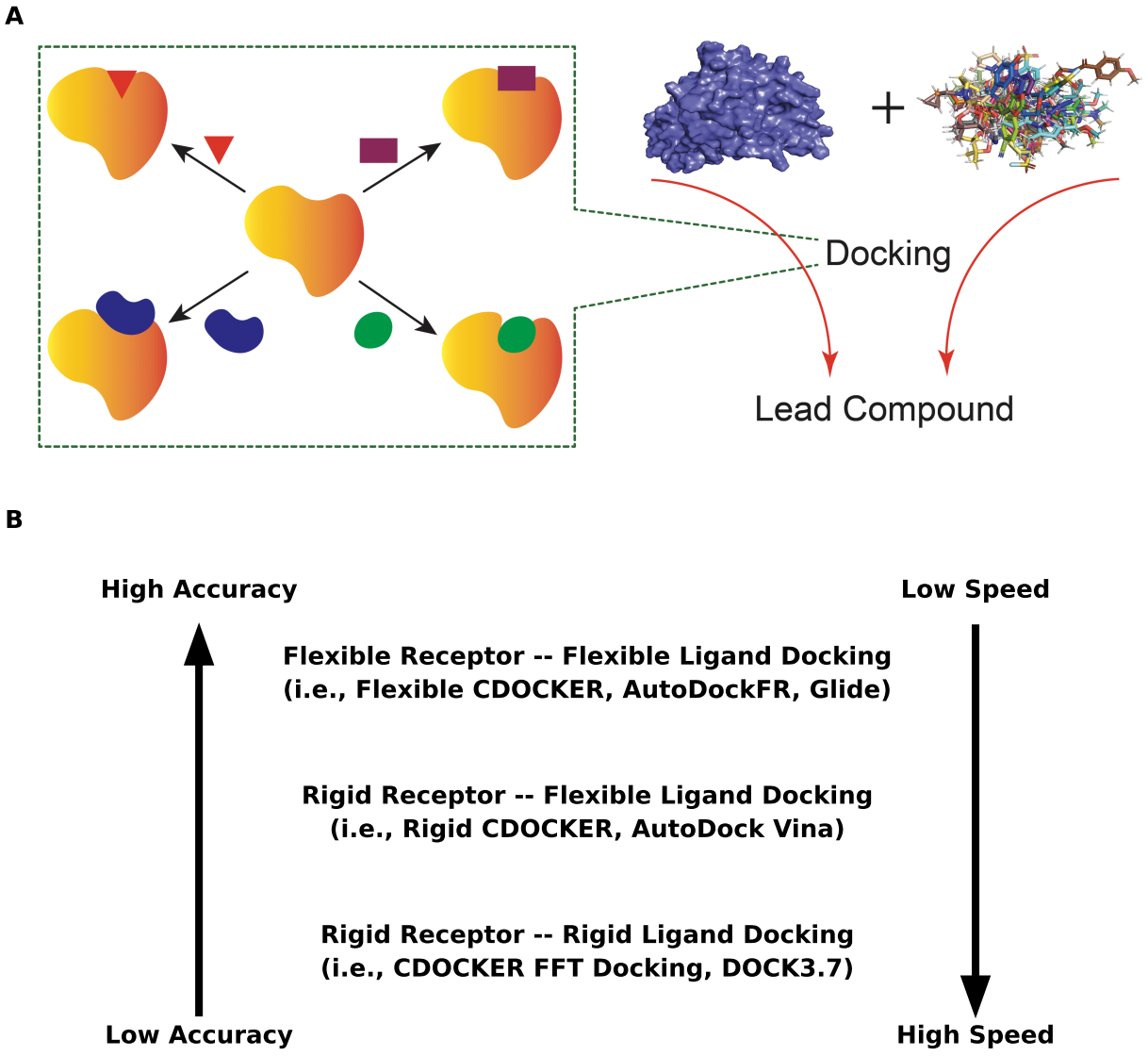

Figure 1.1: Overview of molecular docking models. (A) Binding pocket might adopt different configurations when ligands with different size and shape bound to it. (B) Comparison of docking accuracy and speed among different docking models.

1.1.2 Application scenario of different docking models

In small-scale (i.e., dozens of) docking experiments for pose prediction, it is always a good idea to use sophisticated docking methods, such as Flexible CDOCKER. However, in large-scale docking experiments, such as virtual screening for lead compound identification, direct

application of these methods might not be appropriate or realistic because of its relatively large computational cost, although these methods have higher docking accuracy.

Nowadays, several vendors and academic laboratories have introduced "make-on-demand" libraries based on relatively simple two- or three-component reactions where the final product is readily purified in high yields.[15] For example, the Enamine company could synthesize over 29 billion molecules based on >120,000 distinct and often highly stereogenic building blocks synthesized from >140 reactions.[16] The development of new drugs with traditional methods can cost anywhere in the range of 400 million to 2 billion dollars, with synthesis and testing of lead analogs being a large contributor to that sum.[17] Compared with experimentally screening such ultra large compound libraries (i.e., billions of compounds), docking and experimental validation of the lead compound is a more promising and beneficial approach. Docking such ultra large libraries also demonstrates its potential in identify unusually potent and selective molecules.[18–20] On the other hand, a recent study suggests that docking results and experimental molecular efficacies improve as we increase the compound library size.[18] A successful computer-aided drug design (CADD) protocol can save a large amount of money and time, and this has been a focus of development in the field for decades. A recent review study of the current trends in CADD published in 2021 shows an average of 36.5% of the studies included reports on experimental evaluations following virtual screening and docking methods are the most prominently used type of approach with an average of 57.6% usage.[21]

Docking ultra large compound libraries also brings new challenges in the docking methods. The docking calculation (searching efficiency) must be no more than one second per molecule per core for a billion-molecule library on moderately sized computer clusters (e.g., 500-1,000 cores).[5] This leads to the development and application of these simplified rigid ligand docking methods (i.e., CDOCKER FFT Docking and DOCK 3.7). These methods have a significantly smaller computational cost compared with other types of docking methods by ignoring the conformational changes of both ligand and receptor, ignoring important terms (e.g., ligand strain) and approximate terms that it does include (e.g., fixed potentials).[5,7] Because of such approximation, as shown in Figure 1.1B, these methods have relatively lower accuracy (i.e., more likely to miss ranking the non-binders prior to true inhibitors). Meanwhile, more sophisticated docking methods often give more accurate prediction in binding pose

and relative binding free energy with a computational cost ranging from several minutes to several hours.[12–14] Therefore, we proposed a hierarchical virtual screening workflow so that we can maintain high accuracy and high efficiency (Figure 1.2)

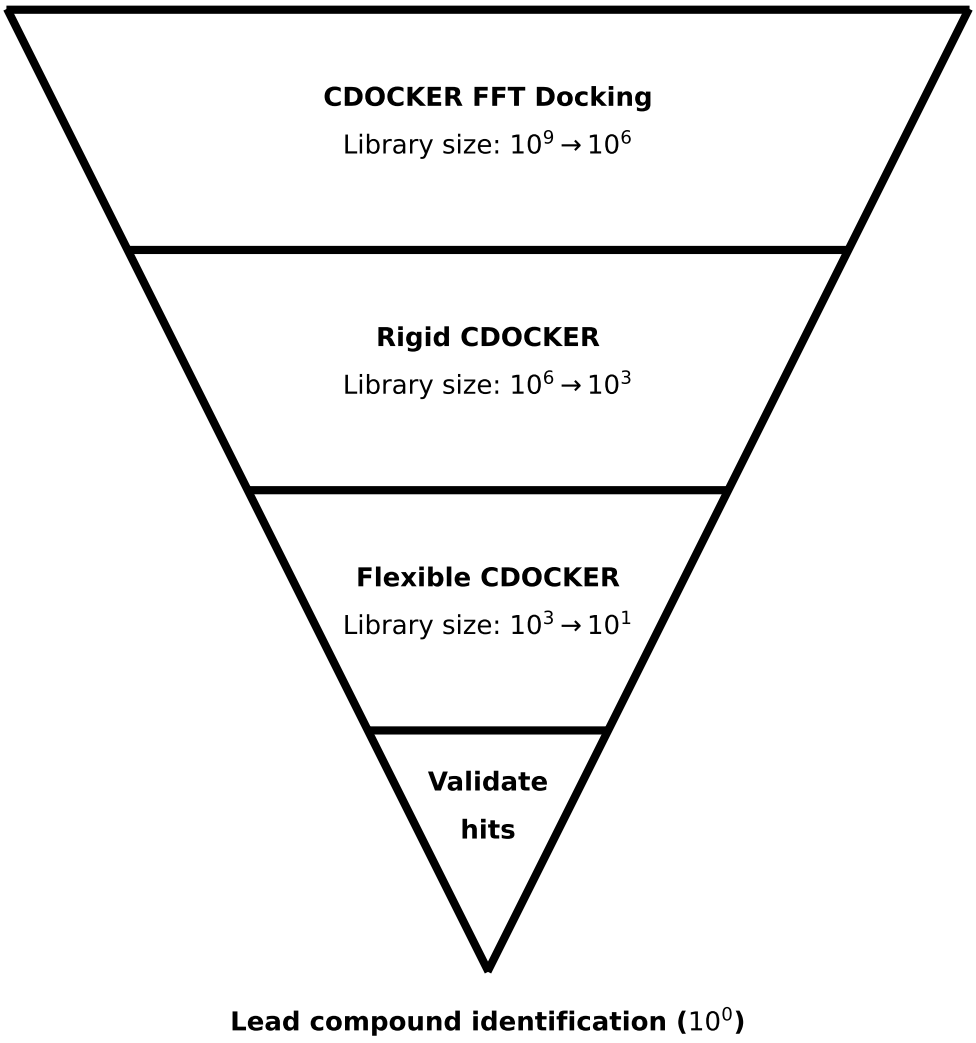

Figure 1.2: A hierarchical virtual screening workflow example.

In Chapter 4, as a practical example of this hierarchical virtual screening workflow concept, we worked with a team of experimental colleagues to identify potential therapeutics for the host transmembrane serine protease TMPRSS2, a promising antiviral target that plays a

direct role in SARS-CoV-2 infections.[22] A database of 134,109 molecules that were collected from multiple sources of molecules in clinical trials, on approved drug lists or readily available in local libraries. These compounds were filtered based on pharmacophore similarity. A total of 4,308 candidates were identified and docked with the flexible docking method described in Chapter 3. From our in silico docking trials, three novel non-covalent inhibitors were identified and verified with biochemical assays.

The needs of low computational cost in docking also motivates us to reduce the computational cost for all CDOCKER docking methods. In Chapter 2, I describe the development and implementation of the new feature parallel simulated annealing.[7] This significantly accelerate CDOCKER by more than an order of magnitude.

1.2 Outstanding Questions in Docking Methodology

We have mentioned briefly in the previous section that docking involves two main components: searching and scoring.[3] In one element, searching, one generates multiple structures of a ligand within the constraints of the receptor binding site. The application of a scoring function then ranks these conformations and is expected to differentiate the correct binding pose from incorrect ones through the assumption that the correct binding pose is at the top rank. Docking methodology has been well established in the past three decades. However, there are several questions in docking that need to be addressed, especially in the field of flexible receptor docking algorithms.

1.2.1 Can we further improve the docking scoring function?

The scoring functions can be classified as physics based, empirical, knowledge based or machine learning based.[23–26] Table 1.1 summarizes the types of scoring function of the popular used docking methods. The physics based scoring functions directly compute the interactions between the atoms of receptor and ligand based on the principles of physics. Therefore, this type of scoring function can be applied to various biological systems without further modification. However, the predictive accuracy for the binding energy is relative lower because it generally neglects entropy and solvent effects in its functional form. On the other hand,

the non-physics based scoring function is constructed by employing a training set of known ligands with binding affinities to optimize each term in the scoring function. Therefore, it is obvious that the non-physics based scoring function tends to have better performance in estimating the binding free energy. However, by the nature of using known data to train the scoring function, the docking performance may vary significantly from system to system, especially in the cases of novel targets, where there are no known inhibitors. This motivates us to further improve the physics based scoring function.

Table 1.1: Docking Methods and the Corresponding Scoring Function

Docking software	Scoring function
CDOCKER[9,27], DOCK[5,28]	Physics based
Autodock Vina[11]	Empirical
Glide[14,29]	Machine learning based
SILCS[30]	Knowledge based

Flexible receptor docking scoring function for pose prediction. Accurate prediction of the binding pose is the fundamental of any docking applications. It has been captured experimentally that the binding pocket might adopt different conformations for different ligands.[31–34] It has also been demonstrated in the literature that flexible receptor docking methods have better performance than rigid receptor docking methods.[12–14] However, there is a lack of well described scoring functions for the flexible receptor docking methods. One usually direct applies rigid receptor docking methods to analyze flexible receptor docking results, which do not consider side chain conformational variance upon binding.[7,13,14] This motivates us to optimize the flexible receptor docking scoring function.

As shown in Figure 1.3, when the ligand is docked to the same receptor binding pocket multiple times, we will generate a set of different docking poses including native-like and non-native like. The application of a scoring function then ranks these docking poses and is expected to differentiate the correct binding pose from incorrect ones through the assumption that the correct binding pose is at the top rank (i.e., $\Delta G_1 < \Delta G_2$ or $\Delta G_4 < 0$). Because we are docking the same ligand to the same binding pocket, the free energy of the protein and ligand in the unbound state are constant (i.e., $\Delta G_3 = 0$) and can be neglected. And we

assume the solvation free energy is approximately the same.

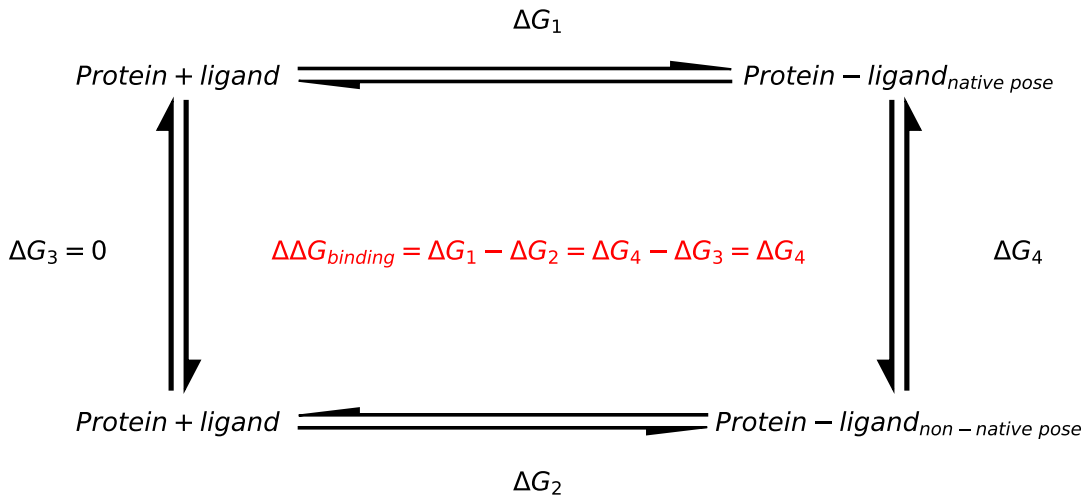

Figure 1.3: Thermodynamic equation for pose prediction. Relative binding free energy equation is shown in red.

Therefore, we only need to calculate the total energy difference among different docking poses (i.e., ΔG_4) with the scoring function. In Chapter 3, I describe the development of the scoring function in the flexible receptor docking method (Flexible CDOCKER).[12] The new physics-based scoring function includes both enthalpic and entropic contributions upon binding by considering the conformational variability of the flexible side chains within the ensemble of docked pose. We demonstrated significantly improved performance in pose prediction compared with other popular rigid and flexible receptor docking methods.

Scoring function for covalent ligand pose prediction. During the TMPRSS2 project, we realize a set of known compounds form a covalent bond with the active site serine of serine proteases.[35,36] On the other hand, targeted covalent inhibitors (TCIs) are considered to be an important component in the toolbox of drug discovery and about 30% of currently marketed drugs are TCIs.[37–39] Because of the covalent bond formation between the TCIs and the receptor target, the inhibitor will stay in the binding pocket for a longer time. Therefore, TCIs typically have high potencies and prolonged effects result in less frequent drug dosing and wide therapeutic margins for patients. This motivates us to develop

covalent docking method. In Chapter 5, I discuss the implementation and testing of a covalent docking methodology in Rigid CDOCKER and the optimization of the corresponding physics-based scoring function with an additional customizable covalent bond grid potential which represents the free energy change of bond formation between the ligand and the receptor. We show that our new covalent docking algorithm has comparable pose prediction accuracy with reduced computational cost.

Scoring function for comparing binding free energy for different compounds. In Chapter 3 and 5, we first developed the scoring function for pose prediction, which estimates the total energy of the protein-ligand complex upon binding. As shown in Figure 1.4, in order to compare different small molecules, we need to augment these scoring functions to consider the system in the unbound state (i.e., $\Delta G_3 \neq 0$). On the other hand, the solvation free energy cannot be neglected because different ligands will have a different contribution. Therefore, we optimized our scoring function by subtracting the total energy of the system in the unbound state and by incorporating solvation/desolvation effects using the FACTS implicit solvent model.[40] In the retrospective virtual screening experiments described in Chapter 3 and 5, we demonstrate that both Flexible CDOCKER and Covalent CDOCKER can identify novel inhibitors among a large chemical space.

1.2.2 What is the optimal searching algorithm for flexible receptor docking?

Flexible receptor docking methods have been developed in the last decades. The biggest advantage in these methods is the sampling of both ligand and receptor conformational space during docking, leading to a more accurate docking result. Flexible receptor docking methods have, however, been less adopted because of the relatively large computational cost. Therefore, one of the key questions in the flexible receptor docking methods is how to perform an efficient sampling in the increased sampling space. Different protein-ligand flexible receptor docking programs address this question in various ways (Table 1.2). However, all of these searching algorithms have potential problems. This motivates us to improve the Flexible CDOCKER searching performance.

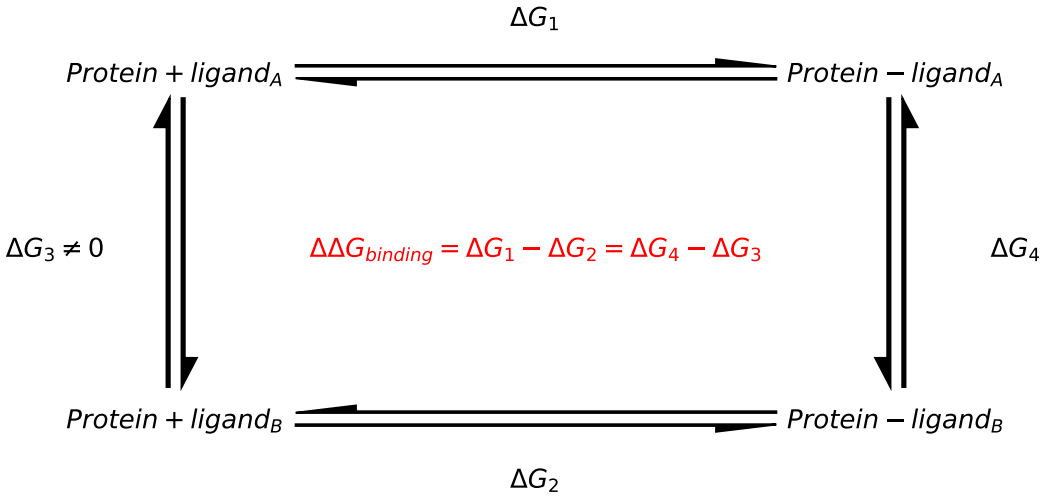

Figure 1.4: Thermodynamic equation for comparing different ligands. Relative binding free energy equation is shown in red.

Table 1.2: Disadvantages in the Flexible Receptor Docking Method Sampling Algorithms

Docking software	Searching method	Disadvantages
Glide[14]	Induced-fit	Receptor and ligand conformational space are sampled iteratively.
AutoDockFR[13]	Genetic algorithm	Relatively hard to overcome the energy barrier to identify global minimum.
Original Flexible CDOCKER[27]	Simulated annealing	Typically it needs more docking trials to identify correct binding pose.

In Chapter 3, I also describe the development of the novel hybrid searching algorithm in the Flexible CDOCKER.[12] This novel hybrid searching algorithm combines both molecular dynamics (MD) based simulated annealing and genetic algorithm crossovers to address enhanced sampling of the increased search space. This new feature significantly reduces the computational cost of Flexible CDOCKER. To our knowledge, Flexible CDOCKER is the fastest flexible receptor docking method.

1.2.3 Can we make docking as a more easily to access tool?

Docking is a powerful computational tool and has been applied for various structure-based structure-function explorations. However, the scripting languages of these docking methods are relatively complicated for ones with little knowledge of docking. On the other hand, to perform a successful docking experiment, one typically would need various cheminformatics tools either in preparation or post analyzing, OpenMM[41,42], Pybel[43], RDKit[44] and Scikit-learn.[45]

Leveraging the popularity and utility of the Python language, in Chapter 6, I introduce the Python functionality to the CDOCKER family (i.e., pyCHARMM CDOCKER). One of the aims of pyCHARMM CDOCKER is to provide simple one-liner code to perform standard CDOCKER docking experiments. This broadens the use by potential users who want to perform standard CDOCKER docking experiments with little knowledge of docking or the CDOCKER family. The Python interface can also further accelerate CDOCKER docking algorithms and make CDOCKER easier to integrate with other commonly used molecular simulation python packages.

1.3 Summary

In this dissertation, I describe the developments and application of CDOCKER docking methodology to advance the field of computer-aided drug design. The organization with Chapter 2, 3, 4, 5 and 6 follows the same structure. An introduction is given at the beginning of each chapter to provide an overview of the specific field. After the introduction, the novel computational methods developed or applied in the dissertation are described in detail. These methods are then evaluated on different systems. Then each chapter is ended by conclusions and discussions.

Chapter 2

Accelerated CDOCKER with GPUs and Parallel Simulated Annealing

Xinqiang Ding, Yujin Wu, Yanming Wang, Jonah Z. Vilseck, and Charles L. Brooks III. "Accelerated CDOCKER with GPUs, Parallel Simulated Annealing, and Fast Fourier Transforms." *J. Chem. Theory Comput.* 2020, 16, 6, 3910-3919. I focused on the methodology development of parallel simulated annealing in CDOCKER.

2.1 Introduction

Protein-ligand docking methods aim to predict how ligands bind with a target protein, i.e., binding poses of ligands and their binding affinities.[27] Many off-the-shelf protein-ligand docking programs, either commercial or free, are available for use,[46] such as CDOCKER,[9] Autodock,[10] Autodock Vina,[11] DOCK,[28] and Glide[29,47]. Most protein-ligand docking programs consist of two essential components — a scoring function for the assessment and ranking of ligand-binding poses and a search algorithm that facilitates the search for and discovery of those low free energy binding poses.[9] The scoring function quantifies the fit between a ligand's binding pose and the protein receptor and is expected to be able to differentiate the correct binding pose from incorrect ones through the assertion that the correct binding pose has the best score. When used to predict binding affinities, the scoring function is also expected to approximate the binding free energy between ligands and target proteins.

The search algorithm is utilized to sample potential ligand binding poses and identify the binding pose with the best score. Because the scoring functions used in protein-ligand docking programs are not convex functions and typically have multiple local minima, heuristic search algorithms such as genetic algorithms and simulated annealing are often used in search for optimal binding poses.[9,11]

CDOCKER[9], a CHARMM[48] module for protein-ligand docking, is one of the protein-ligand docking programs that are widely used in both academia and industry for drug discovery. It uses the interaction energies between proteins and ligands calculated with the CHARMM force field[48] for proteins and the CGenFF force field[49] for ligands as its scoring function. To search for the lowest energy pose of ligands, CDOCKER utilizes molecular dynamics (MD) based simulated annealing followed by energy minimization. In the MD based simulated annealing, MD is used to simulate the dynamics of protein-ligand interaction and the temperature of the MD first increases to a high value and then slowly decreases. As the temperature of the MD decreases, ligands are expected to adopt low energy poses. The resulting ligand poses from simulated annealing are further optimized by energy minimization. As the MD-based simulated annealing is a heuristic search approach, it is not guaranteed that the ligand will converge to the lowest energy pose in each trial of MD-based simulated annealing. To increase the chance that the lowest energy pose of the ligand is identified, multiple trials of simulated annealing are employed. In each trial, the ligand is first initialized with a random conformation,[50] a random orientation, and a random position within the binding pocket before going through the MD-based simulated annealing and energy minimization. After the energy minimization, the resulting poses, one from each trial, are ranked by their energies that include the intra-interaction energy of the ligand and the interaction energy between the ligand and the protein. The pose with the lowest energy is predicted to be the binding pose. In a typical application of CDOCKER, a large number of ligands need to be docked with a protein. Therefore, the docking procedure has to run fast enough to make the method practical. To accelerate the docking procedure and help search for the lowest energy pose of ligands, CDOCKER utilizes a cubic grid representation of the binding pocket and soft-core potentials,[9,27] respectively, both of which will be described in detail in following sections.

In this paper one new feature — parallel MD simulated annealing — is added to CDOCKER to further accelerate the search algorithm in CDOCKER. Both features are implemented such that they can take advantage of the parallel computing power of graphical processing units (GPUs). In addition, the original CDOCKER routine used for computing protein grid potentials is also updated such that it can run on GPUs.

2.2 Methods

2.2.1 Accelerated routine for computing soft-core grid potentials

In CDOCKER's docking protocol, most of the computational time is spent on calculating forces on ligand atoms and the ligand's interaction energy with the protein for a large number of ligand poses. To accelerate the force and energy calculation, a cubic grid representation of the binding pocket is used. Specifically, the binding pocket inside a protein is discretized onto a cubic grid. Probe atoms are placed on each of the grid points and their interaction energies with the protein are saved in a lookup table. Then the interaction energy of a ligand atom with the protein can be rapidly calculated by looking up values in the tables, instead of explicitly calculating its interaction with all of the protein atoms. When a ligand atom's position is not on any grid points, its interaction energy with protein atoms is calculated using a trilinear interpolation of the energy values on the eight grid points of the cell which contains the ligand atom. The force from protein atoms on a ligand atom is approximated using the local energy gradient. To accurately approximate the interaction energy between different ligand atoms and a protein, multiple grids are needed and calculated based on the protein. One of the grids is associated with the electrostatic interaction energy and the remaining grids are associated with the van der Waals interaction energy for ligand atoms with different atom types. In total, 27 grid potentials are computed based on the structure of the protein receptor in CDOCKER.

Soft-core potentials in CDOCKER are used to smooth the energy landscape, which can help the MD-based simulated annealing search escape from local minima and identify the ligand pose with the lowest energy. Specifically, when using soft-core potentials, the van der

Waals, electrostatic attractive, and electrostatic repulsive energies are approximated using the formula:

$$E_{ij} = E_{\max} - a \cdot r_{ij}^{b} \text{ if } |E_{ij}^{*}| > \frac{|E_{\max}|}{2}, \tag{2.1}$$

where E_{ij}^{*} is the regular interaction energy; $E_{\max}$ is a parameter controlling the "softness" of the potential; Given $E_{\max}$, a and b are automatically determined using the condition that the energy and the force calculated using Eq. 2.1 have to be equal to that calculate using the regular formula at the switch distance where $|E_{ij}^{*}| = |E_{\max}|/2$.

In the previous version of CDOCKER, the routine for computing soft-core grid potentials for the protein receptor ran on central processing units (CPUs), which was slow due to the large number of grid points and protein atoms. For instance, to compute 26 grid potentials, each of which has $43 \times 43 \times 43$ lattice points, for a target protein of 8757 atoms, the number of floating point operations is on the order of $43 \times 43 \times 43 \times 26 \times 8,757 = 18,102,312,774$. It took 2,100 seconds in the previous implementation running on a CPU(Intel® Xeon® Processor E5520) to compute these grid potentials. Here, we added a parallel implementation of the routine such that it can calculate the soft-core grid potentials on GPUs. To do the same computation, it only takes 2.5 seconds for the new implementation running on a GPU (NVIDIA® GEFORCE® GTX 1080). Therefore, our new implementation on GPUs is more than 800 times faster than the previous implementation on CPUs.

2.2.2 Parallel MD-based simulated annealing with GPUs

One of the advances in using MD simulations to study both chemical and biological systems has been the utilization of GPUs.[42,51–53] Compared with CPUs, the parallel computing power of GPUs enables us to run MD simulations orders of magnitude faster and simulate longer timescale dynamics of chemical and biological systems, which makes MD suitable to study processes that were not accessible before.[42,51–53] Although GPUs have been widely employed in running MD simulations of large chemical and biological systems, they are rarely used to accelerate protein-ligand docking methods. Here we investigated the utilization of GPU computing to accelerate CDOCKER for protein-ligand docking by running MD-based sim-

ulated annealing of multiple copies of ligands and any included flexible receptor regions in parallel on one GPU.

Because protein-ligand interaction energy landscapes have many local minima and the MD-based simulated annealing is a heuristic search method, multiple trials of simulated annealing have to be employed to search for the lowest energy pose. As the number of trials increases, the docking accuracy usually improves until it reaches a plateau. In addition, in a typical application, CDOCKER needs to dock a large number of ligands with a protein. Therefore, accelerating multiple trials of MD-based simulated annealing can help CDOCKER to dock a large number of ligands in a limited time while maintaining docking accuracy. Because trials of MD-based simulated annealing are independent, one way to accelerate the calculation is to run them in parallel with multiple processors. In the existing implementation of CDOCKER, multiple trials of MD-based simulated annealing can be run in parallel with multiple CPUs. Here we introduce a new feature into CDOCKER to enable it to run multiple trials of MD-based simulated annealing simultaneously on GPUs.

As there are already implementations of MD engines running on GPUs, instead of writing a new MD engine specifically for running multiple trials of MD-based simulated annealing on GPUs, we utilize the existing GPU-enabled MD engine that is part of the CHARMM/OpenMM interface.[41] To utilize the MD engine from OpenMM for our purpose, we make a customized system consisting of multiple copies of a ligand and any included flexible receptor regions and one copy of the potential grids of the protein. Atoms in each copy of the ligand and the flexible receptor group interact with atoms in the same copy and the potential grids, but do not interact with atoms in all other copies of flexible groups. Therefore, although the system includes multiple copies of the flexible atoms, these copies are independent from each other and the dynamics of each copy is the same as if there is just one copy. Running one trial of MD-based simulated annealing with this customized system on GPUs is equivalent to running multiple trials of simulated annealing using the previous implementation on CPUs.

The parallel MD-based simulated annealing with GPUs is implemented for both rigid receptor and flexible receptor docking.[27] In rigid receptor docking, protein grid potentials are computed using all protein atoms and the customized OpenMM system consists of multiple

copies of ligand atoms. In flexible receptor docking, side chains of multiple amino acids near the binding pocket are modeled as flexible regions. The protein atoms of these flexible side chains are modeled the same as ligand atoms, except that they are attached to fixed protein backbone atoms through bonded interactions. These protein atoms of flexible side chains are excluded when computing the protein grid potentials and the customized OpenMM system consists of multiple copies of both ligand atoms and atoms in flexible side chains. More detailed information on flexible receptor docking in CDOCKER is available in the original paper describing flexible CDOCKER.[27]

2.3 Results

2.3.1 Benchmark datasets and computational details

Three sets of protein-ligand complexes are used as benchmark datasets to evaluate the protein-ligand docking methods just described. The Astex diverse set[54] and the SB2012 set[55] are used for evaluating rigid receptor docking. The SEQ17 dataset[13] is used for evaluating flexible receptor docking. The Astex diverse set contains 85 diverse high-resolution protein-ligand complexes and has been widely used for benchmarking different protein-ligand docking methods.[54] In this study, 70 of the 85 protein-ligand complexes that do not include cofactors are used. Compared to the Astex diverse set, the SB2012 set[55] is a much larger set of protein-ligand complexes. It contains 1043 protein-ligand complexes, out of which the 1003 complexes that do not have cofactors and can be typed using CGenFF[56] are used in this study. The 1003 protein-ligand complexes from the SB2012 set overlap with 69 out of 70 complexes from the Astex diverse set. Protein-ligand complexes with cofactors are excluded because force field parameters of the cofactors are not readily available in the CHARMM force field. Unlike protein-ligand docking methods that use empirical scoring functions, CDOCKER uses physical interaction energies based on the all-atom CHARMM force field. Cofactors have to be parameterized based on the CHARMM force field in order to be included in docking. Because not all cofactors in the benchmark datasets are readily available using CGenFF, we excluded all protein-ligand complexes with cofactors to enable

us to treat the complexes in the benchmark datasets in a consistent manner. In focused studies on specific receptors, if cofactors were present, they could be consistently included or parameterized and incorporated into docking studies. However, for our purposes of benchmarking here, we have just ignored complexes with cofactors. The SEQ17 dataset, which was originally used to benchmark the flexible receptor docking method AutodockFR[13], contains 17 pairs of apo-holo structures. These 17 systems were selected to represent a wide range of receptors by the original authors and we include them here for our studies.

The MD-based simulated annealing in CDOCKER is conducted in several stages and different stages use protein grid potentials with different softness values, i.e., grid potentials calculated using different $E_{\max}$ (Eq. 2.1). Three sets of values for the parameter $E_{\max}$ are used for rigid receptor docking and they are summarized in Table 2.1. The stages of simulated annealing for rigid receptor docking are as follows. Using the soft-core grid potential I, the ligand is heated up from 300K to 700K over 3000 MD integration steps with a step size of 1.5 femtoseconds and then cooled down from 700K to 300K over 14000 steps. Then using the soft-core grid potential II, the ligand is further cooled down from 500K to 300K over 7000 steps and then from 400K to 50K over 3000 steps. Finally, the ligand is minimized using the soft-core grid potential III for 200 steps.

Table 2.1: Soft-core Potentials Used in Rigid Receptor Docking

name	$E^*_{\max}$(vdw)	$E^*_{\max}$(att)	$E^*_{\max}$(rep)
soft-core potential I	0.6	-0.4	8.0
soft-core potential II	3.0	-20.0	40.0
soft-core potential III	100	-100	100

*$E_{\max}$(vdw), $E_{\max}$(att) and $E_{\max}$(rep) in the unit of kcal/mol are parameters for the van der Waals, electrostatic attractive, and electrostatic repulsive interactions, respectively.

Similar stages of MD-based simulated annealing are used in flexible receptor docking except that soft-core grid potentials with different softness are used. Specifically, the soft-core grid potential I is changed and is adopted from the flexible docking protocol outlined by Gagnon et al.[27] to prevent ligands from leaving the binding pocket during the initial searching and heating stages of simulated annealing. The soft-core potential III is designed to give a more native-like energy landscape. Since there are more explicit atoms involved in flexible

receptor docking, these values are larger than those used for rigid receptor docking. Detailed values for the softness parameter $E_{\max}$ used in flexible receptor docking are summarized in Table 2.2. More details on the simulated annealing procedures used in the present study are available in the CHARMM scripts included in the Supporting Information.

Table 2.2: Soft-core Potentials Used in Flexible Receptor Docking

name	$E^*_{\max}$(vdw)	$E^*_{\max}$(att)	$E^*_{\max}$(rep)
soft-core potential I	15.0	-120.0	-2.0
soft-core potential III	10000	-10000	10000

*$E_{\max}$(vdw), $E_{\max}$(att) and $E_{\max}$(rep) in the unit of kcal/mol are parameters for the van der Waals, electrostatic attractive, and electrostatic repulsive interactions, respectively.

2.3.2 Parallel MD-based simulated annealing with GPUs

GPU accelerated parallel simulated annealing significantly accelerates CDOCKER

Compared with the original CDOCKER running serially on CPUs, the speedup of the parallel MD-based simulated annealing with GPUs is shown in Table 2.3. For the protein-ligand pairs in the Astex diverse set, when 100 and 500 docking trials are used, the average wall time used by the original CDOCKER with CPUs are 338.4 and 1692.0 seconds, respectively. In contrast, the average wall time used by the parallel MD-based simulated annealing with GPUs are 30.8 and 85.5 seconds, respectively, which is about 10 fold and 20 fold faster. The speedup becomes even larger when the number of trials increases, because the wall time used by the original CDOCKER on CPUs is proportional to the number of trials.

Comparison with other protein-ligand docking programs for rigid receptor docking.

The accelerated CDOCKER is compared with three other widely used protein-ligand docking programs: Autodock, Autodock Vina, and DOCK. The computational details for setting up docking in all the docking programs presented here are included in the Supporting Information. To make a fair comparison between different protein-ligand docking programs, the same settings are used for all docking programs whenever it is possible. For instance, for

Table 2.3: Acceleration of Parallel MD-based Simulated Annealing with GPUs Compared with the Original CDOCKER with CPUs on the Astex Diverse Set.

	CDOCKER with CPUs	CDOCKER with parallel MD-based simulated annealing with GPUs
accuracy[a]	0.623 ± 0.023	0.631 ± 0.029
wall time[b] (seconds)	338.4	30.8
wall time[c] (seconds)	1692.0	85.5

[a] The accuracy when 100 trials are used; The ligand native conformation is used as the starting conformation; The uncertainty is estimated using 10 independent repeats. [b] The wall time used when 100 trials are used; [c] The wall time used when 500 trials are used.

a protein-ligand complex, a docking grid box with the same position and size is used in all programs. The docking accuracy is calculated as the percentage of protein-ligand complexes in benchmark datasets for which the RMSD of the lowest energy docked pose is less than 2.0Å from the native pose.

The re-docking results on the Astex diverse set and the SB2012 set are shown in Table 2.4 and Table 2.5, respectively. With the acceleration achieved by the parallel MD-based simulated annealing with GPUs in CDOCKER, the average wall time required by CDOCKER for docking one protein-ligand complex is either faster than or on par with other programs. For CDOCKER, Autodock, and Autodock Vina, the docking accuracy one obtains depends on whether the ligands' native conformation or a random conformation is used as the starting conformation: all of these approaches perform better when the ligand native conformation is used as the starting conformation. Starting with ligands' native conformation makes the conformational search easier and the docking accuracies higher than those corresponding to using random starting conformations. Because the DOCK program uses the "anchor and grow" search method[28], its accuracy does not depend on the starting conformation.

Based on the results from the Astex diverse set, when ligand random conformations are used as starting conformations, DOCK and Autodock Vina have similar and higher docking accuracy. Autodock has the lowest docking accuracy and CDOCKER is in between. Increasing the parameter that controls the searching exhaustiveness in Autodock Vina from 8 to 20 proportionally increases the run time, but it does not change its docking accuracy significantly. Compared with the results on the Astex diverse set (Table 2.4), the relative

Table 2.4: Docking Accuracy of Multiple Protein-ligand Docking Programs on the Astex Diverse Set.

	CDOCKER[d]	Autodock v4.2.6	Autodock Vina[e]	Autodock Vina[f]	DOCK v6.7
accuracy (native[a])	0.664 (± 0.022[g])	0.600 (± 0.020)	0.701 (±0.019)	0.710 (± 0.009)	0.639 (± 0.016)
accuracy (random[b])	0.537 (±0.021)	0.530 (±0.029)	0.633 (±0.014)	0.623 (±0.011)	
wall time (sec)[c]	85.5	279.6	82.3	202.9	50.0

[a] Ligand native conformations are used as starting conformations. [b] Ligand random conformations are used as starting conformations. [c] CDOCKER is run on a GPU (NVIDIA GeForce GTX 980). All the other docking programs use one CPU (Intel Xeon Processor E5645 2.4GHz). [d] 500 trials are used in CDOCKER. [e] exhaustiveness = 8. [f] exhaustiveness = 20. [g] All uncertainties are estimated using 10 independent repeats.

Table 2.5: Docking Accuracy of Multiple Protein-ligand Docking Programs on the SB2012 Set.

	CDOCKER[c]	Autodock v4.2.6	Autodock Vina[d]	Autodock Vina[e]	DOCK v6.7
accuracy(native[a])	0.569 (± 0.006[f])	0.477 (± 0.009)	0.631 (±0.004)	0.642 (± 0.005)	0.553 (±0.005)
accuracy (random[b])	0.429 (± 0.007)	0.418 (±0.004)	0.532 (±0.004)	0.547 (±0.004)	

[a] Ligand native conformations are used as starting conformations. [b] Ligand random conformations are used as starting conformations. [c] 500 trials are used in CDOCKER. [d] exhaustiveness = 8. [e] exhaustiveness = 20. [f] All uncertainties are estimated using 10 independent repeats.

performance of the protein-ligand docking programs on the SB2012 set is the same in terms of docking accuracy (Table 2.5). However, for all of the programs, the docking accuracy is lower on the SB2012 set (Table 2.5). Although the Astex diverse set contains a diverse set of protein-ligand complexes, the number of protein-ligand complexes in the set is relatively small. Because the SB2012 dataset contains more than an order of magnitude more protein-ligand complexes, the performance on the SB2012 set should be a more objective measure of the protein-ligand docking programs' docking accuracy. The lower docking accuracy on the SB2012 set for all the tested protein-ligand docking programs can be attributed to either

Table 2.6: Docking Accuracy of SEQ17 Dataset Using Flexible CDOCKER

Receptor Structure	Searching Accuracy	Ranking Accuracy
Holo	95.29% ± 2.48%	78.82% ± 4.96%
Apo	80.00% ± 6.90%	57.06% ± 3.97%

search algorithms or scoring functions or both. This suggests that more efforts are required to further improve search algorithms and scoring functions including, both physics based scoring functions used in CDOCKER and DOCK and the empirical scoring functions used in Autodock and Autodock Vina. We note that the docking accuracies of both Autodock Vina and DOCK reported in this study are quite different from those reported in previous studies.[11,28,57] This is because of the fact, as shown in this study, that the docking accuracy of a protein-ligand docking program can vary significantly depending on ligand starting conformation and benchmark dataset.

Benchmarking flexible receptor docking

The implementation of parallel simulated annealing on GPUs also significantly accelerates flexible receptor docking in CDOCKER. To test both the docking accuracy and speed of flexible receptor docking in CDOCKER, we used the SEQ17 dataset which contains 17 pairs of apo-holo structures and represents a subset of protein-ligand complexes in the SEQ dataset that were able to be successfully docked using AutoDock (RMSD $<$ 2.0 Å) with rigid re-docking.[13] These 17 systems were selected to represent a wide range of receptors. For each of the 17 apo-protein structures, there is at least one amino acid side chain around the binding pocket whose conformation in the holo structure is different from that in the apo structure by at least 2.5 Å (RMSD). Therefore, there would be at least one severe clash between ligand atoms and the receptor side chain if the ligand native conformation from the protein-ligand complex is directly fit onto the apo structure. In what follows, we evaluate the overall performance (accuracy and speed) of flexible CDOCKER for flexible receptor docking using this dataset and compare it with AutodockFR[13].

In preparing each system for flexible receptor docking, the apo structure is superposed on the corresponding holo structure. The criteria used for selecting flexible side chains is

the same as that used in AutoDockFR, except that in flexible CDOCKER the heavy atoms beyond Cα are considered to be flexible, whereas the heavy atoms beyond Cβ are set to be flexible in AutoDockFR.[13] The RMSD cutoff to identify native-like poses is set to be 2.5 Å to be consistent with the evaluation criteria used by AutoDockFR.[13] We performed 10 repeated calculations, each of which consists of 500 trials of both flexible CDOCKER and rigid CDOCKER.

The docking results on the SEQ17 set are shown in Table 2.6. In each repeat, we first computed the searching accuracy. A searching success is positive when at least one native-like docking pose is identified among the docked poses from 500 trials of docking. The searching accuracy for rigid CDOCKER is 36.47% $\pm$ 5.41% when the apo structure is used as the receptor and 95.88% $\pm$ 2.84% when the holo structure is used. In contrast, for flexible CDOCKER, the search accuracies are 95.29% $\pm$ 2.48% and 80.00% $\pm$ 6.90% when the apo structures and the holo structure are used as receptors, respectively (Table 2.6). Therefore, flexible CDOCKER has a higher searching accuracy than rigid CDOCKER when the receptor binding pocket has a conformational change upon binding with ligands.

To further assess the accuracy of flexible CDOCKER for this set, we considered the following scoring protocal. The docked poses from 500 trials of docking are clustered using a K-means clustering algorithm from the MMTSB toolset (cluster.pl)[58] based on ligand heavy atoms with a RMSD cutoff of 2.0 Å. If the number of docked poses in a cluster is less than 10, then the cluster is discarded. This allows us to remove those less populated docking poses that are frequently away from the binding pocket. Then the minimum energy pose of each cluster is selected and ranked based on the scoring function. A scoring success is defined as at least one of the cluster representatives presents as a native-like pose.[13] The scoring accuracy of flexible docking in CDOCKER is about 57.06% when apo structures are used as receptors (Table 2.6) and the corresponding scoring accuracy of AutodockFR is 70.6%[13] which is higher than CDOCKER. However, as mentioned before, the SEQ17 dataset is intentionally designed to only include protein-ligand complexes that can be correctly re-docked (RMSD $<$ 2 Å) using the Autodock. If we do the same filtering process for CDOCKER, i.e., exclude protein-ligand complexes that can not be re-docked using CDOCKER, which include 3PTE and 2A78, the scoring accuracy of flexible docking in CDOCKER becomes 68.82% $\pm$ 3.97%,

which is similar to the accuracy achieved by the AutodockFR.

In high-throughput virtual screening, where a large number of ligands need to be docked, the major barrier for the use of flexible receptor docking is its speed. For flexible CDOCKER with parallel simulated annealing running on a GPU (NVIDIA® GEFORCE® GTX 1080), the average wall time required for one protein-ligand complex is about one hour. In contrast, AutodockFR running on a CPU (Intel® Xeon® Processor E5520) requires on average $50 \times 8.5 = 425$ hours for one protein-ligand complex[13]: it takes 8.5 hours for AutodockFR to run one round of genetic algorithm evolution and 50 rounds of evolution are required. Therefore, our flexible CDOCKER based on parallel MD-based simulated annealing on GPUs greatly reduces the amount of time required while maintaining similar docking accuracy for flexible receptor docking.

2.4 Conclusions and discussions

The parallel MD-based simulated annealing with GPUs enables CDOCKER to run about 20 times faster when 500 trials of simulated annealing are used. The speedup becomes even larger when more trials of simulated annealing are employed. With this acceleration, the speed of CDOCKER is on par with or faster than several other popular protein-ligand docking programs tested in this study. In addition, the parallel MD-based simulated annealing on GPUs also enables flexible CDOCKER to achieve significant speed advantages while maintaining similar accuracy for flexible receptor docking. We note that these additions to the CDOCKER and flexible CDOCKER modules described in this paper are part of the c43a1 development version of CHARMM and will be released for general use in the near feature.

Chapter 3

Flexible CDOCKER: Hybrid Searching Algorithm and Scoring Function with Side Chain Conformational Entropy

Yujin Wu and Charles L. Brooks III. "Flexible CDOCKER: Hybrid Searching Algorithm and Scoring Function with Side Chain Conformational Entropy." *J. Chem. Inf. Model.* 2021, 51, 11, 5535-5549.

3.1 Introduction

In high-throughput docking for hit identification, multiple small ligands with different sizes are docked to the target receptor. During this process, it is reasonable to assume the binding pocket undergoes conformational changes, and this has been captured experimentally.[31–34] Today, multiple off-the-shelf protein-ligand flexible docking programs, either commercial or free, are available for use, such as Glide[14,29], ROSETTALIGAND[59], Flexible CDOCKER[7,27] and AutodockFR[13]. As one notable example, the flexible docking method in Glide combines induced-fit docking and molecular docking (IFD-MD).[14,29] This method shows high accuracy in ranking native-like poses as top rank compared with rigid docking. However, the average computational cost for an IFD-MD calculation is 400 CPU hours and 50 GPU hours,[14] which is similar to applications of AutodockFR, making it less likely to be applied in high-

throughput experiments.

In the previous study (Chapter 2), we explored flexible receptor docking using the Seq17 dataset,[13] which comprises 17 pairs of apo-holo structures that were selected to represent a wide range of receptors, both Flexible CDOCKER and AutodockFR showed a higher accuracy in finding a native-like pose compared with rigid docking.[7,13] However, both docking algorithms performed less well in ranking the native-like pose as top rank.[7,13] For Flexible CDOCKER with parallel simulated annealing running on one GPU, the average wall time required to compute the ensemble of ligand-receptor poses necessary to identify a cluster of native-poses for one protein-ligand complex was about 1hr, more than a 10-fold improvement in computing time compared to AutodockFR.[7] Given this relatively high efficiency in finding a native-like pose in Flexible CDOCKER, in the current work we focus on further improvements in the search algorithm and on improving the scoring function for flexible docking.

Flexible CDOCKER[27] uses a physics-based scoring function (eq 3.1) and does not include entropic contributions, which have been reported to be important in many cases.[60–63] Machine learning based scoring functions are constructed by characterizing protein-ligand complexes using feature vectors comprising the number of occurrences of specific protein-ligand atom type pairs interacting within predefined distance thresholds,[64,65] using different biochemical descriptors to characterize the protein-ligand interaction,[66] or assigning different weights to physics-based scoring functions.[67] Empirical scoring functions, such as X-SCORE, propose an additional term for the calculation of conformational entropy based on the number of rotatable bonds in the ligand.[68] However, both empirical approaches and machine learning based approaches may not serve to capture the side chain conformational entropy change upon binding because the number of the flexible side chain rotatable bonds does not change when binding to two different ligands. In the current work, we seek to improve the general physics-based scoring function by including an entropy calculation based on the microscopic definition of entropy as described below. We test this idea with a range of different datasets that include multiple receptors and ligands.

$$\Delta G_{binding} = E_{protein\ internal\ energy} + E_{ligand\ internal\ energy} + E_{vdw} + E_{elec} \tag{3.1}$$

3.2 Methods

3.2.1 CDOCKER Algorithm Overview

There are three main elements to the Flexible CDOCKER algorithm: the receptor and ligand representation, the newly developed scoring function, which includes the calculation of side chain conformational entropy, and a corresponding updated hybrid searching algorithm that combines molecular dynamics (MD) based simulated annealing[2,3,7,9,27] and genetic algorithm crossovers.[2,3,11,13,69] In the current study, the rigid docking algorithm and MD based simulated annealing parameters are the same as reported in Chapter 2

3.2.2 Receptor and Ligand Representation

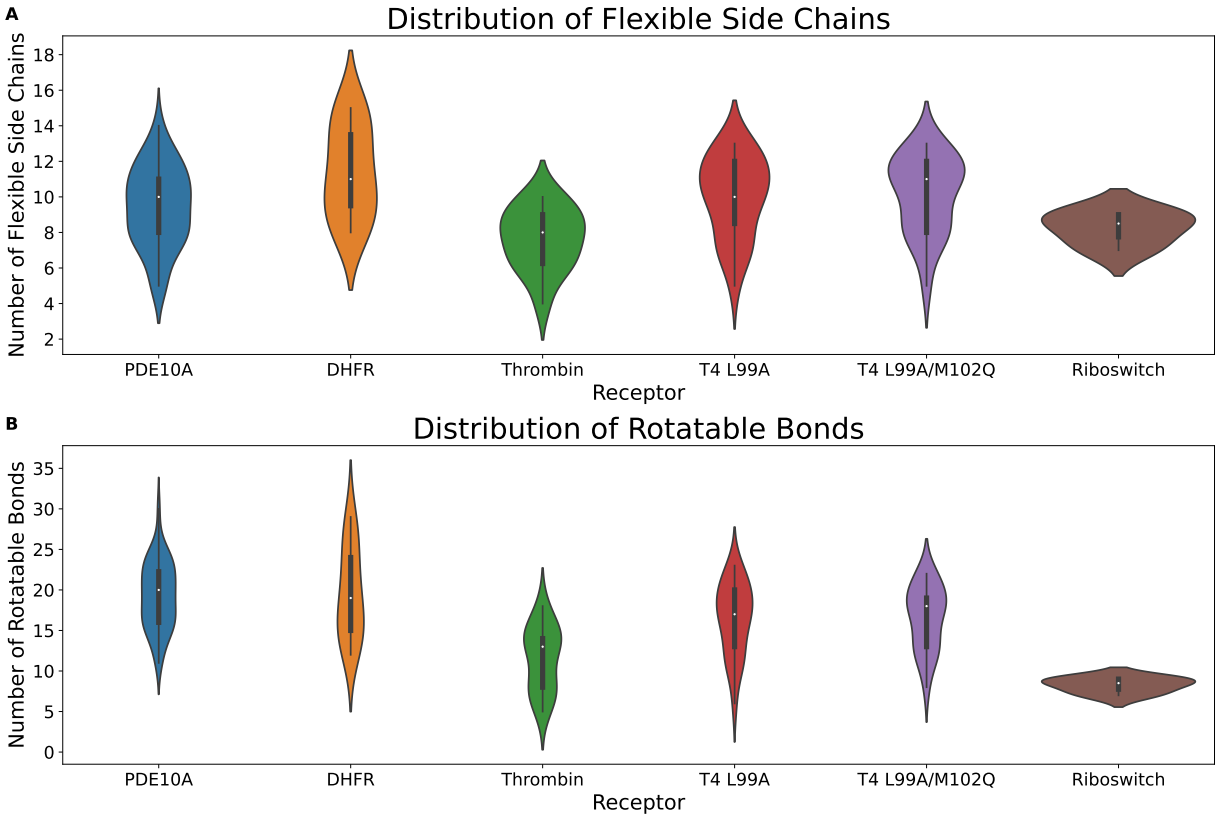

Figure 3.1: Receptor binding pocket complexity. **(A)** Distribution of number of receptor flexible side chains. **(B)** Distribution of number of receptor rotatable bonds.

All structure files were acquired from the PDB and had a structural resolution better than 2.0 Å. MOE (Molecular Operating Environment)[70] was used to predict the protonation state of the ligands and cofactor at pH 7.4. The dominant protonation state of the compound is selected for the following docking experiments. Open Babel[71] was used to generate random ligand conformations, ParamChem[72,73] was used to prepare the ligand topology and parameter files and the MMTSB tool set[58] was used to cluster the binding poses. The ligand is minimized with CHARMM in vacuum before the docking experiments. Clustering used the tool cluster.pl with a 1 Å cutoff radius for the K-means clustering based on ligand heavy atom RMSD. In preparing each receptor for docking experiments, all co-crystal structures from the same receptor were superimposed to each other based on the backbone root-mean-square deviation of atomic positions (RMSD) using PyMOL[74] to provide a common reference frame. The CHARMM C36 force fields[49] were used and docking was performed in CHARMM[48] with the CHARMM/OpenMM parallel simulated annealing feature.[7] The RMSD cutoff to identify native-like poses is set to be 2.5 Å for flexible docking and 2.0 Å for rigid docking, to be consistent with the evaluation criteria in the previous studies.[7,9,13]

The idea of flexible docking is to allow conformational changes of receptor side chains to occur so that the ligand can identify its native-like pose. Incorporation of protein flexibility has become more feasible due to advances in computational resources. In drug discovery, one often starts with one or a few co-crystal structures and uses flexible docking experiments to identify the native-like pose for ligands that do not yet have a co-crystal structure. To mimic this process, for each receptor-ligand holo complex in a cross-docking dataset with N crystal structures, we develop a unique definition of the flexible side chains for each receptor based on the distance between ligand heavy atoms and the corresponding target structure side chain atoms. We identified a receptor side chain as flexible if at least one pairwise interaction of the ligand heavy atoms and the side chain was within a 4Å cutoff.[7,13] All side chain atoms from the C-alpha carbon are considered flexible. The average number of flexible side chains in this study is 10. The distribution of flexible side chains and rotatable bonds within the ligands are shown in Figure 3.1.

For the cross-docking experiment against a given receptor with flexible side chains defined as noted above, $N-1$ ligands were docked to this structure with no changes in flexible side

chain selections. The cross-docking experiments were performed for all N receptor structures, each with a potentially different set of flexible side chains depending on the bound ligand in that receptor. Thus, we have a total of $N \times (N-1)$ cross-docking experiments for a dataset with N crystal structures. For completeness, we also report the N self-docking results as well. In the real application, we note that one could incorporate more structures with varied ligand interfaces to the receptor if they were available, thereby expanding the size of the flexible side chain region to accommodate the additional knowledge about the targeted receptor. If one does not have any knowledge of the bound state except a targeted binding pocket, one could perform some rigid docking trials to identify side chains that may interact with the ligands and choose the flexible region in this manner.

3.2.3 Benchmark Dataset

In high throughput screening, the more common case is that there exits a large number of different ligands docked to the same receptor. Thus, we use the 6 datasets shown in Table 3.1, which cover different receptor classes and binding environments. Each receptor contains several ligands. All co-crystal structures from the same receptor class share the same binding pocket. A brief description of the data sets and our rationale for choosing each is given in the following.

Table 3.1: Flexible Receptor Docking Benchmark Dataset

Receptor Name	Co-crystal structures	Cross-docking/re-docking experiments	Receptor class
PDE10A	44	1892 / 44	Kinase
T4 L99A	23	506 / 23	Lysozyme
T4 L99A/M102Q	21	420 / 21	Lysozyme
DHFR	11	110 / 11	Folate enzyme
Thrombin	14	182 / 14	Serine protease
Riboswitch	4	12 / 4	RNA

T4 L99A dataset. This dataset was chosen to provide a simple model where the binding pocket is small, buried and hydrophobic and is comprised of a total of 23 different

holo structures. It was chosen to test the performance of the flexible docking algorithm for the case of a small, buried and hydrophobic binding pocket.[75]

T4 L99A/M102Q dataset. This dataset is examined to provide a simple model where the binding pocket is small, buried and contains only one hydrophilic side chain.[75] It contains a total of 21 different holo structures. It has been included to test the performance of our flexible docking algorithm when the binding pocket that is small, buried and hydrophobic with only one hydrophilic side chain presented.

Riboswitch dataset. Many research groups have developed different docking algorithms for RNA-ligand docking.[76–85] The common feature in these studies is that the majority of the investigated targets are complex, including large and flexible ligands and water-mediated interactions. These are still challenging in the current protein-ligand docking methodologies and make it difficult to distinguish these effects from issues specific to RNA-ligand flexible docking. Therefore, we selected a rather simple RNA-ligand docking system developed by the Brenk group, which allows us to focus on the impact of side chain flexibility on docking accuracy (i.e., backbone RMSD less than 2 Å).[86] There are only 4 different holo structures in this dataset. Compared with the two T4 datasets, this dataset also serves the purpose to test the performance of our flexible docking algorithm for cases of a small, buried and hydrophilic binding pocket.

PDE10A dataset. This dataset contains 44 different holo structures of phosphodiesterase 10A (PDE10A), which is a kinase. There are either two nickel ions or one zinc ion and one magnesium ion included as part of the binding pockets. The nickel ions have not been parameterized in the CHARMM force field. Since all of the ions have a +2 charge and occupy similar positions in the binding pocket, we used the same zinc ion and magnesium ion in the cross-docking experiments. This dataset is built to test the performance of the flexible docking algorithm for cases where there are ions in the binding pocket.

DHFR dataset. This dataset contains 11 different holo structures of dihydrofolate reductase (DHFR), which is a folate enzyme.[75] It has a more open binding site displaying both polar and apolar binding regions. Nine of the holo structures have a common cofactor NDP (NADPH dihydro-nicotinamide-adenine-dinucleotide phosphate). One of the holo structures (PDBID: 1DR3) has a slightly different cofactor TAP (7-thionicotinamide-

adenine-dinucleotide phosphate). The other holo structure (PDBID: 2CD2) also has a slightly different cofactor NAP (NADP nicotinamide-adenine-dinucleotide phosphate). All of the cofactors occupy the same position near the binding pocket. To be consistent in the cross-docking experiments, we used the same cofactor NDP. The main purpose of including this set of structures was to increase the variability of our dataset and test our flexible docking algorithm when cofactors are present.

Thrombin dataset. This dataset contains 14 different holo structures of thrombin, which is a serine protease.[75] The ligands from this dataset are the largest. The binding pocket is also more open than the others. This dataset is built to test our flexible docking algorithm with large ligands.

3.2.4 Flexible Docking Scoring Function with Side Chain Entropic Contributions

Upon binding, the change in free energy can be represented by eq 3.2. In cases of differentiating the correct binding pose from incorrect ones, the same ligand is docked to the receptor multiple times and generates multiple docking poses. Since the ligand and protein always start from the same initial state, $G_{initial}$ will be the same for all trials. Thus, only G_{final} needs to be calculated for pose identification. The enthalpic contribution (H_{final}), including protein-ligand interactions, ligand internal energy and protein internal energy, have been well-established in the previous Flexible CDOCKER scoring function.[9,27] The entropic contribution (S_{final}) can be separated into contributions from solvation and conformational entropy. Since we consistently dock the same ligand to the same binding pocket in one measurement, we assume the solvation contribution is approximately the same for different docking poses, and we suggest that it can be neglected. Thus, the flexible scoring function can be simplified to eq 3.3:

$$\begin{aligned} \Delta G_{binding} &= G_{final} - G_{inital} \\ &= H_{final} - H_{initial} - T\left(S_{final} - S_{initial}\right) \end{aligned} \tag{3.2}$$

$$\Delta G_{binding} = E_{protein-ligand\ interaction} + E_{protein\ internal\ energy} + E_{ligand\ internal\ energy} - TS_{conformational\ entropy} \tag{3.3}$$

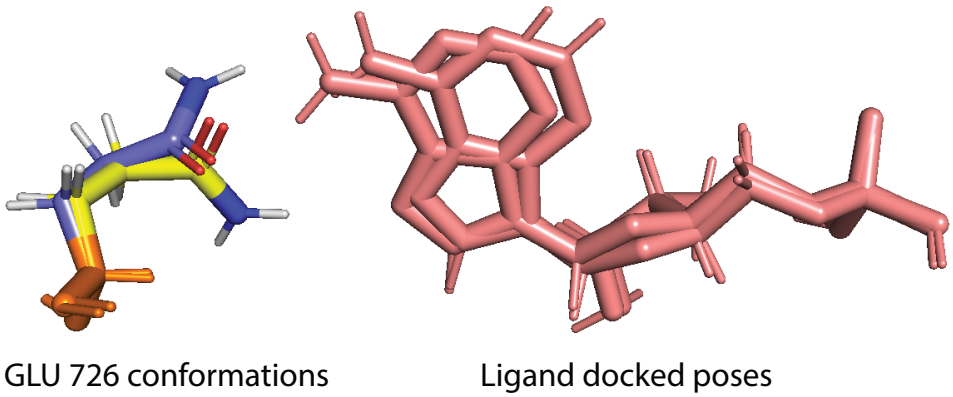

Figure 3.2: 2OUN flexible self-docking results using the flexible receptor docking algorithm. These two docking poses (pink) belong to the same cluster and are native-like docking poses. The corresponding flexible side chain GLU726 adopts two different conformations. The backbone atoms of the two GLU726 conformations are shown in orange, while the side chain atoms are shown in yellow and blue, respectively. The side chain amide groups adopt two different orientations.

For a given flexible docking measurement, the docking poses are clustered based on the ligand heavy atoms. Thus, in one ligand cluster, all ligand conformations are similar to each other. Therefore, the conformational entropy of the ligand within that cluster is assumed to be zero. However, as shown in Figure 3.2, the receptor side chains can adopt different conformations, which results in a non-zero side chain conformational entropy and a variation of enthalpy within one cluster. Here, we use the microscopic definitions to compute the side chain conformational entropy upon binding.

Protein side chain conformational states. The amino acid side chain conformational states can be classified by the dihedral angle of the rotamers, an idea originally used for computing side chain conformational entropy in protein unfolded states.[87] Only the dihedral angle χ, involving four heavy atoms, is considered as a rotamer. The IUPAC-IUB convention is used to define trans ($\pm 180^o$), gauche- ($g-, -60^o$) and gauche+ ($g+, +60^o$) conformations for a rotamer.[88] Gly, Ala and Pro are excluded. Because of the symmetry in the benzyl

group and phenyl group in Phe and Tyr respectively, these two amino acids have χ_2 rotamer conformations of 2 instead of 3. The maximum number of states that one amino acid can access is listed in Table 3.2.

Table 3.2: Number of Rotamer States for Each Amino Acid

Residue	Total number of dihedral angles, χ_i	Maximum number of states
Arg, Lys	4	81
Gln, Glu, Met	3	27
Asn, Asp, His, ILe, Leu, Trp	2	9
Phe, Tyr	2	6
Cys, Ser, Thr, Val	1	3

RNA side chain conformational states. The increasing number of RNA crystal structures enables a structure-based approach to the discovery of new RNA-binding ligands,[89,90] and a number of RNA-ligand docking software, such as AutoDock[77], DOCK[77,85] and RiboDock[81], are available. RNA can adopt different three dimensional structures that are critical for its function. Thus, it is important to consider receptor flexibility during docking. DOCK[85] rescores the docking poses with receptor side chains being flexible, and RiboDock[81] uses an ensemble of receptor structures to mimic the receptor flexibility. Here, we want to expand the application of our new flexible docking algorithm to RNA-ligand docking, which allows the receptor side chains and ligand configurations to explore their conformational space simultaneously.

We define the sugar and phosphate group in the nucleotide monophosphate as the backbone of RNA and the nucleobase is considered as the flexible side chain. Thus, only one dihedral angle χ is presented in all 4 bases. Instead of using the IUPAC-IUB convention, we rotate this dihedral angle and record the total energy for each nucleotide monophosphate. The conformational states determined by this protocol are summarized in Table 3.3.

Final flexible docking scoring function. For a specific side chain in a given ligand cluster of size N, the conformational entropy of this side chain is calculated using the microscopic definition of entropy (eq 3.4). The state of a side chain is a function of all dihedral

Table 3.3: Conformational State Determined by Dihedral Angle, χ

Nucleotide monophsphate	State 1	State 2
ADE, GUA	$-91 \sim 175^{\circ}$	$-180 \sim -91^{\circ}$ or $75 \sim 180^{\circ}$
URA, CYT	$-170 \sim 137^{\circ}$	$-180 \sim -170^{\circ}$ or $137 \sim 180^{\circ}$

angles in that side chain. Different side chains are considered independent of each other.

$$\begin{aligned} S_{side\ chain\ conformation} &= -k_B \sum_j p_j ln\ p_j \\ p_j &= \frac{number\ of\ side\ chains\ in\ state\ j}{N} \\ state\ j &= f\{\chi_1, \chi_2, ..., \chi_n\} \end{aligned} \tag{3.4}$$

As we mentioned above, differences in structural conformations result in a variation of enthalpy in one cluster. For the same cluster, the system can be treated as a classical and discrete canonical ensemble. Each docking pose in this cluster is one state. The probability of a docking pose in state j can be calculated based on its energy. Since the entropy for all ligands in a given cluster is a constant, we can simplify the partition function and calculate the ensemble average of the enthalpic contributions with the following equation (eq 3.5):

$$\begin{aligned} p_j &= \frac{e^{-\beta \Delta G_{binding,\ j}}}{\sum_i e^{-\beta \Delta G_{binding,\ i}}} = \frac{e^{-\beta(H_j - TS_j)}}{\sum_i e^{-\beta(H_i - TS_i)}} = \frac{e^{-\beta H_j} e^{\beta TS_j}}{\sum_i e^{-\beta H_i} e^{\beta TS_i}} = \frac{e^{-\beta H_j}}{\sum_i e^{-\beta H_i}} \\ \overline{H} &= \sum_j p_j H_j \end{aligned} \tag{3.5}$$

By using an ensemble average of the enthalpy, the minimum energy pose, which has the largest weight among the cluster members, will be chosen as the best individual (representative) of that cluster. Therefore, we reach the final equation (eq 3.6). The temperature is set to be $298K$ in all docking experiments in the current study.

$$\begin{aligned} \Delta G_{binding} = \overline{E}_{protein-ligand\ interaction} + \overline{E}_{protein\ internal\ energy} \\ + \overline{E}_{ligand\ internal\ energy} + k_B T \sum_{all\ side\ chains} \sum_j p_j ln\ p_j \end{aligned} \tag{3.6}$$

3.2.5 New Hybrid Searching Algorithm

To further augment sampling in the context of flexible receptor side chains, we propose a hybrid search algorithm for Flexible CDOCKER that combines molecular dynamics (MD) based simulated annealing[2,3,9,27] and a continuous genetic algorithm[2,3,11,13,69,91,92]. In genetic algorithms, the genome is the set of variables to optimize. A given set of values for these variables comprises a docking solution and is called an individual. In this study, these variables are the coordinates of the ligand and flexible side chains of the receptor. Following the ideas of hyperplane sampling in a discrete genetic space, where each individual (potential solution) is considered to be a hyperplane and the competition among different hyperplanes is reflected by the population[91] , we cluster the docking poses and each cluster is considered to be a hyper-surface partially specified by the coordinates of the ligands, where the size of the cluster reflects the competition among different hyper-surfaces. The clustering we perform identifies common basins in the docking energy landscape of the ligand by identifying the clusters of ligands possessing similar positions and configurations, determined by clusters based on the ligand heavy atom RMSD with a radius cutoff of 1 Å. We combine the MD based simulated annealing algorithm to optimize the results by local minimization and redistributing the population of different hyper-surfaces by crossing genes comprised of ligand positions and conformations and flexible side chain conformations. The overall workflow is shown in Figure 3.3.

Each docking measurement creates 500 individuals (these comprise the docking trials). The Open Babel functionality (obrotamer), which uses a genetic algorithm to perform a systematic search over all ligand rotatable bonds, is used to create a starting library of diverse conformers.[71] This library of randomly generated conformers is then centered at the binding pocket followed by a random translation (maximum ± 2 Å) and random rotation (maximum 360°) to generate the initial ligand coordinates. An energy cutoff is applied to filter out significant collisions between ligand atoms and protein atoms due to the random translation and rotation.[7] The receptor flexible side chains are initialized with the coordinates from the input conformation of the receptor. Then these individuals are optimized by a MD based simulated annealing algorithm. The docking poses (optimized individuals) are then

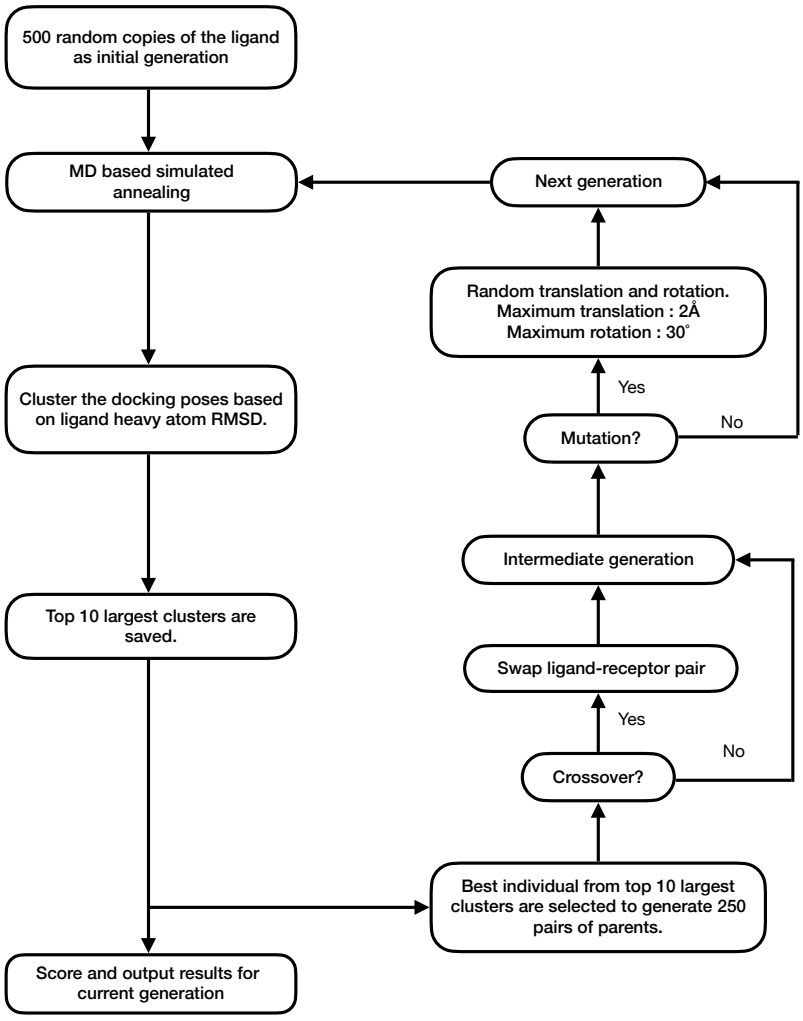

Figure 3.3: Flexible docking searching algorithm.

K-means clustered based on ligand heavy atom RMSD with a radius cutoff of 1 Å. The top 10 largest clusters are scored with the scoring function described above. Since these optimized individuals could include docking poses that are outside of the binding pocket (i.e., large RMSD with respect to the (unknown) binding pose). We adopt the idea of

"promising area" and "intensification" from the continuous genetic algorithm purposed by Chelouah.[92] The genetic algorithm and clustering method are performed to localize multiple local minima in the docking energy landscape (promising area), including (potentially) the global minima (native-like poses). The key concept of intensification is the concentrating of potential optimal solutions. To test this idea we examined the distribution of ligand RMSD with respect to the native pose for each of the ligands associated with the 6 receptors listed in Table 3.1 following the initial application of MD-based simulated annealing.

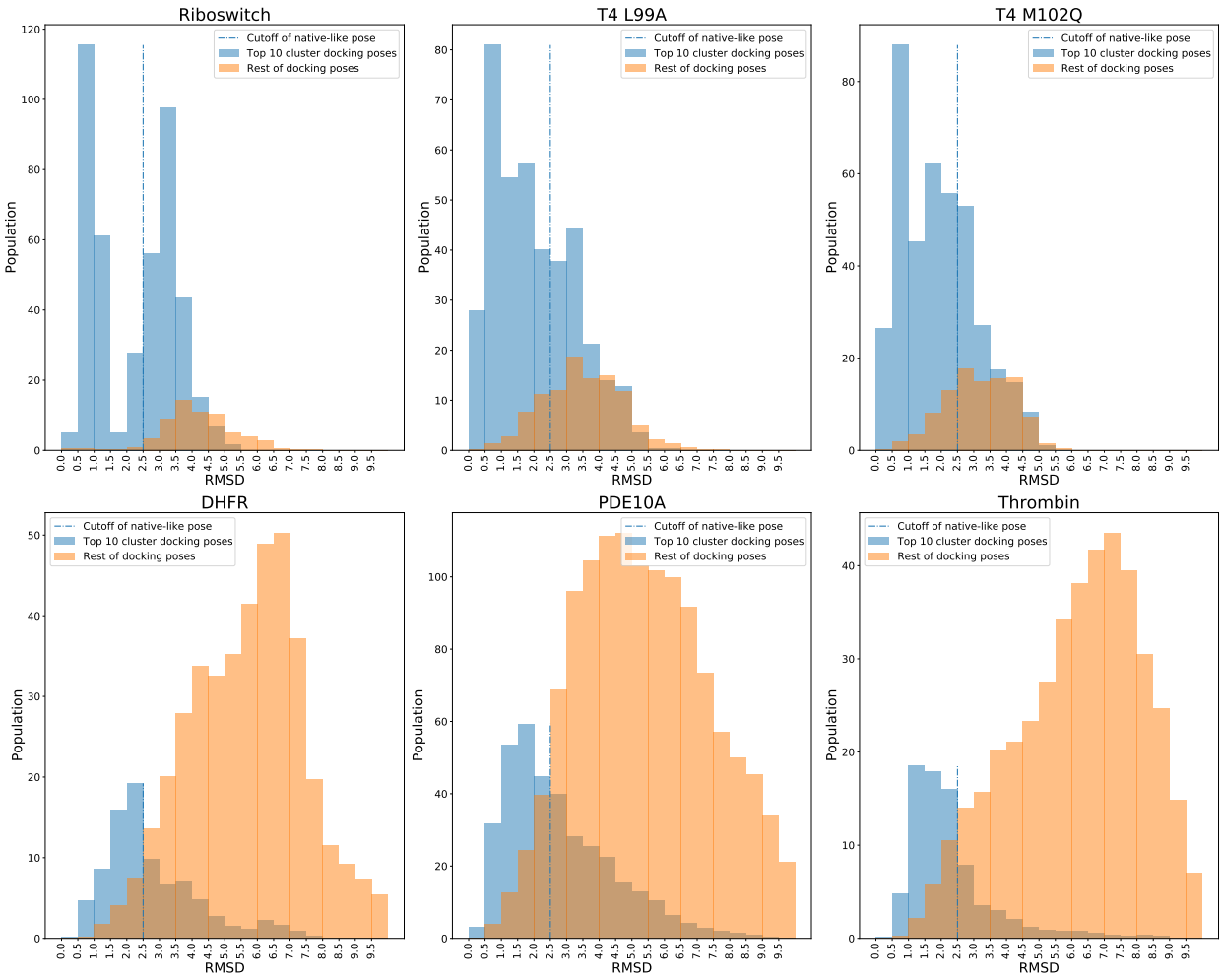

Figure 3.4: Average RMSD distribution of ligand docking poses in the initial generation. The RMSD values are binned with a 0.5 increment.

As shown in Figure 3.4, the less populated clusters are frequently away from the binding pocket. For simple systems (i.e., Riboswitch, T4 L99A and T4 L99A/M102Q) where the

binding pocket is small and buried, the sampling space is small and the majority of the docking poses are native-like and clustered into the top 10 largest clusters. For complex systems (i.e., DHFR, PDE10A and Thrombin) where the binding pocket is more open and the ligands are larger, the sampling space is large and the ligand has a greater probability of adopting an incorrect binding pose that is away from the binding pocket. Regardless, the top 10 largest clusters contain the majority of the native-like poses. There is a small percentage of the less populated clusters that also contain native-like poses (i.e., within 2.5 Å RMSD cutoff of the native binding pose). This is because the RMSD cutoff for clustering is set to be 1 Å so that not all of the native-like poses are grouped into one large cluster. However, it is safe to select best individuals from the top 10 largest clusters as the "promising area" to construct the next generation of individuals.

Intensification. The minimum energy pose for each cluster is considered as the best individual of a cluster. We select the best individuals from the top 10 clusters as the parents. We first select 250 pairs of the individuals randomly using a roulette-wheel selection method. The probability of selecting a given parent is based on the population of the clusters (eq 3.7).

$$P_{select,i} = \frac{N_i}{\sum_{j=1}^{10} N_j} \qquad (3.7)$$

$$N_i, \text{ Population of solution j}$$

The first intensification is done by performing a crossover that swaps the ligand-receptor pair to produce the intermediate generation. The probability for a pair of parents to undergo crossover is 0.5. The second intensification is mutation. The mutation operator in our hybrid searching algorithm is defined as a random translation (maximum 2 Å) and rotation (maximum 30°) of the ligand. The idea of the mutation operator is to find a lower energy state for the receptor-ligand pair. The probability of mutation for a given individual in the intermediate generation is a function of the difference between the total energy of that

individual and the largest total energy value among the 10 parents ($E_{cutoff,\ 1}$) (eq 3.8).

$$
\begin{aligned}
&E_{cutoff,\ 1} = max\{E_{parent}\} \\
&E = E_{crossover} - E_{cutoff} \\
&P_{(E)} = \begin{cases} 1 - 0.5 \times exp(E), & \text{if } E < 0. \\ 0.5 \times exp(-E), & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.8}
$$

Therefore, the more stable a given individual is, the less likely it is to undergo a mutation. The acceptance of this mutation is iterative and an energy cutoff of the total energy is applied to avoid collision with the receptor resulting from crossover and mutation. If the energy cutoff is reached for a given individual then another mutation will be applied to this individual until the system energy is lower than the cutoff value. This energy threshold is calculated by adding 500 $kcal/mol$ to $E_{cutoff,\ 1}$. This step allows us generate a new generation around the previously found "promising area" and the search around the best individuals from the previous generation.

Termination criteria. One disadvantage of using the genetic algorithm is the number of generations (time) needed for the search to converge.[91] Currently, Flexible CDOCKER requires 1 hr for 500 docking trials (one generation) on a GPU[7] and AutodockFR requires on average of 7.3 hr for one generation on a CPU.[13] AutodockFR considers the solutions within 2 kcal/mol of the lowest energy solution as the "promising area" and performs an iterative genetic search so that all solutions will result in this focused sampling space, which by default uses 50 rounds of genetic evolution.[13]

However, as we mentioned before, the majority of the native-like poses are clustered into a small number of populated clusters and are considered as "promising areas", and the following intensification step concentrates the individuals (potential solutions) in the next generation within this focused sampling space. Therefore, our searching algorithm requires fewer generations before the population of native-like poses reaches a plateau. This is more feasibly applied in practical applications, such as high-throughput virtual screening. Thus, we performed flexible docking experiments with 5 generations and recorded the population of native-like poses for each generation for all 6 different receptors. As shown in Figure 3.5,

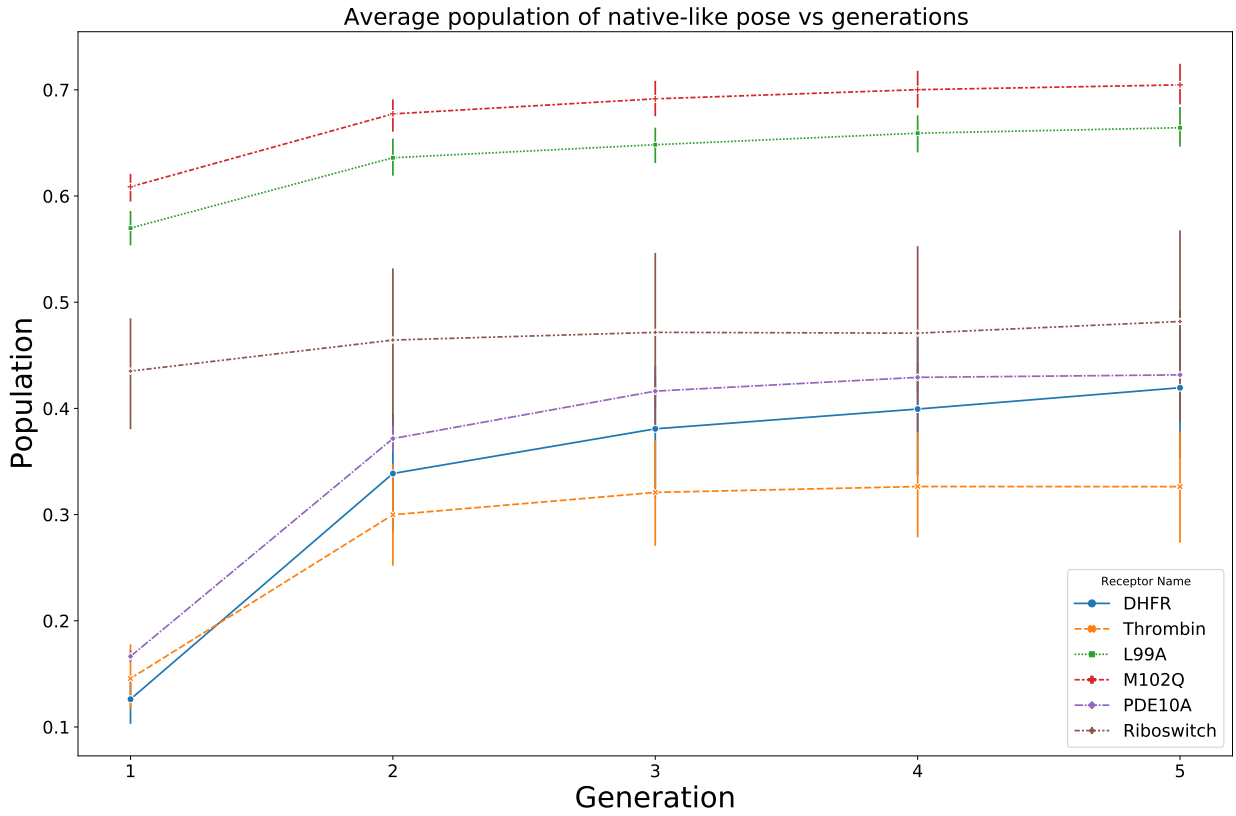

Figure 3.5: Average population of native-like poses vs generation. Population for each docking measurement is calculated by dividing number of native-like poses by 500 (the number of trials in a generation). Average population of native-like poses for all 6 datasets is plotted with their corresponding error bars constructed by computing the standard deviation.

for all 6 datasets, the average population of native-like poses reaches a plateau after the second generation using the move set and gene construction we employ here. Therefore, we use 2 generations in our searching algorithm in the following experiments, which provides a sufficient population of native-like poses within a reasonable timeframe.

3.3 Results

3.3.1 Flexible Docking vs Rigid Docking

We have demonstrated that our purposed flexible docking algorithm generates solutions of well-populated native-like poses in a highly competitive timeframe. The search of the

optimal orientation (i.e., bound conformation) and ranking it as top rank is the fundamental objective of docking.[67,93] Now, we consider the following two questions: (1) Compared with rigid docking, does flexible docking improve in cross-docking native-pose identification? (2) In representative realistic applications, can we identify limitations that suggest areas for future improvement? To assess these questions, we performed cross-docking calculations on the 6 receptor datasets listed in Table 3.1 with the proposed flexible docking algorithm. Rigid CDOCKER and AutoDock Vina are used for direct comparison.

Datasets containing T4 L99A, T4 L99A/M102Q and the Riboswitch. These three datasets provide simple binding environments: a hydrophobic environment, a hydrophobic environment with one hydrophilic side chain and a hydrophilic environment. All of the binding pockets are small and buried.[75,86] The ligands bound to those receptors are also small and rigid compared with ligands in other datasets as illustrated in Figure 3.6. The cumulative docking accuracy for flexible docking and rigid docking trials is illustrated in Figure 3.6 A-C. Top ranking accuracy of pose prediction is shown in Table 3.4.

Table 3.4: Top Rank Accuracy in Pose Prediction of Cross-docking/Re-docking

Receptor Name	Flexible CDOCKER	Rigid CDOCKER	AutoDock Vina
T4 L99A	66.21% / 82.61%	56.13% / 60.87%	49.01% / 47.83%
T4 L99A/M102Q	77.62% / 61.90%	54.29% / 80.95%	51.19% / 80.95%
Riboswitch	25.00% / 50.00%	41.67% / 50.00%	41.67% / 50.00%

As shown in Figure 3.6 and Table 3.4, flexible docking overall performs better than rigid docking. The top 10 ranking accuracy is above 98% for both T4 datasets. The largest improvement in top rank accuracy is for the T4 L99A/M102Q dataset, which is 23.33% higher than Rigid CDOCKER. This result suggests that flexible docking performs better when there exists differences in the binding environment that increases the specificity. The top ranking accuracy for both flexible docking and rigid docking is lower compared with the other two datasets for the Riboswitch. Two main differences between proteins and RNA in binding are[86]: (1) RNA molecules are highly charged and (2) RNA-ligand interactions are dominated by polar contacts. In the current docking setup, the docking scoring function uses a distance dielectric constant set to $3r$.[9,27,94] This reduces the electrostatic interactions,

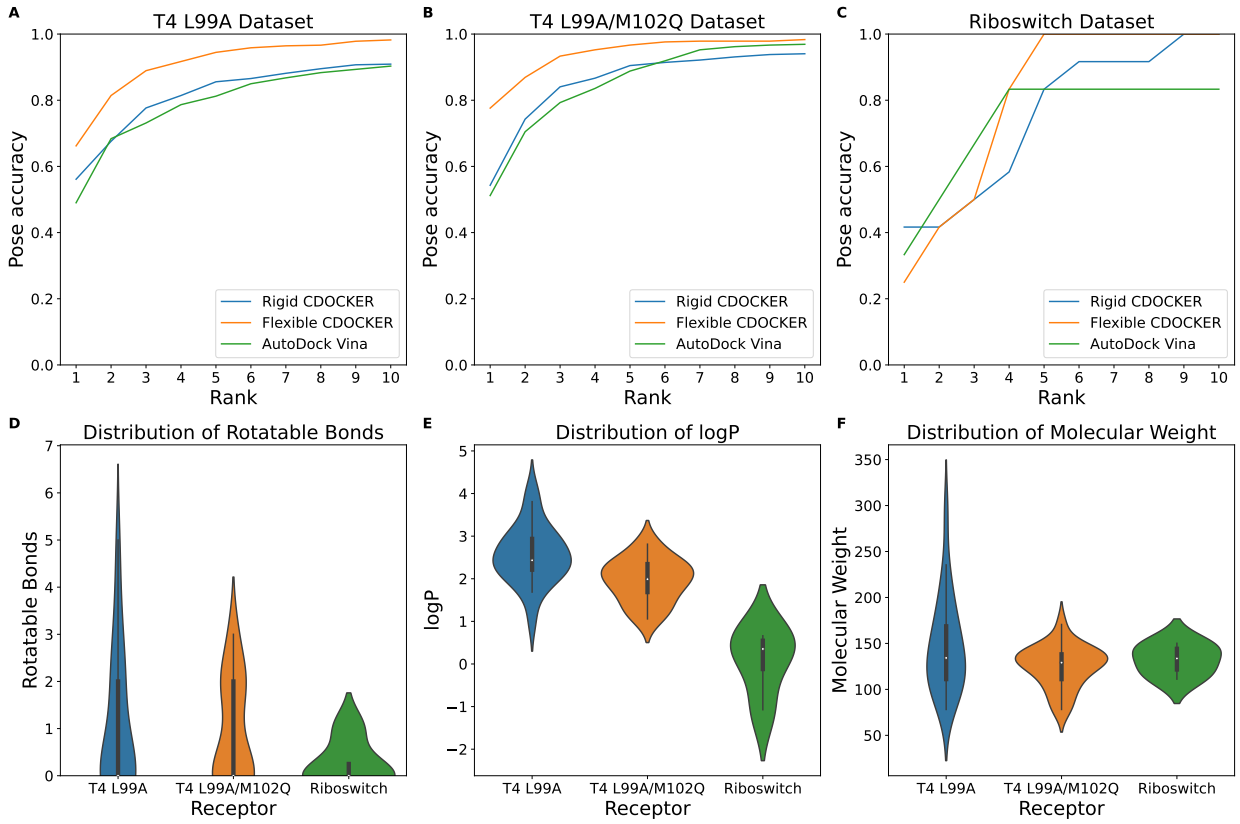

Figure 3.6: Cumulative docking accuracy for the (A) T4 L99A dataset, the (B) T4 L99A/M102Q dataset and the (C) Riboswitch dataset. Distribution of ligand properties: (D) rotatable bonds, (E) logP and (F) molecular weight. A rank of N means the correct docking pose is within the top N solutions. AutoDock Vina trials use an exhaustiveness of 20. Rigid CDOCKER uses 500 docking trials for each cross-docking experiment.

which are the main interactions in RNA-ligand recognition. Studies have shown that using optimized force-field parameters for RNA-ligand docking could improve accuracy.[86] This could be a potential solution to the relatively lower top rank accuracy in the Riboswitch dataset and will be further explored in more focused studies of RNA-based targets but are beyond the scope of our current study.

Datasets containing PDE10A, DHFR and Thrombin. We next move on to evaluate the flexible docking algorithm with three more complex systems: PDE10A dataset, DHFR dataset and thrombin dataset. All of these systems have a more open binding pocket. The PDE10A dataset has two ions within the binding site, while the DHFR dataset has a cofactor in the binding pocket. The ligands in the thrombin dataset are larger and more

flexible than ligands we tested in the other datasets as illustrated in Figure 3.7. The ions or the cofactors are implicitly represented by the grids, but are present within the context of our physics based scoring function used to compute the receptor grid. The cumulative docking accuracy of both flexible docking and rigid docking calculations is shown in Figure 3.7 A-C. Top ranking accuracy of pose prediction is listed in Table 3.5.

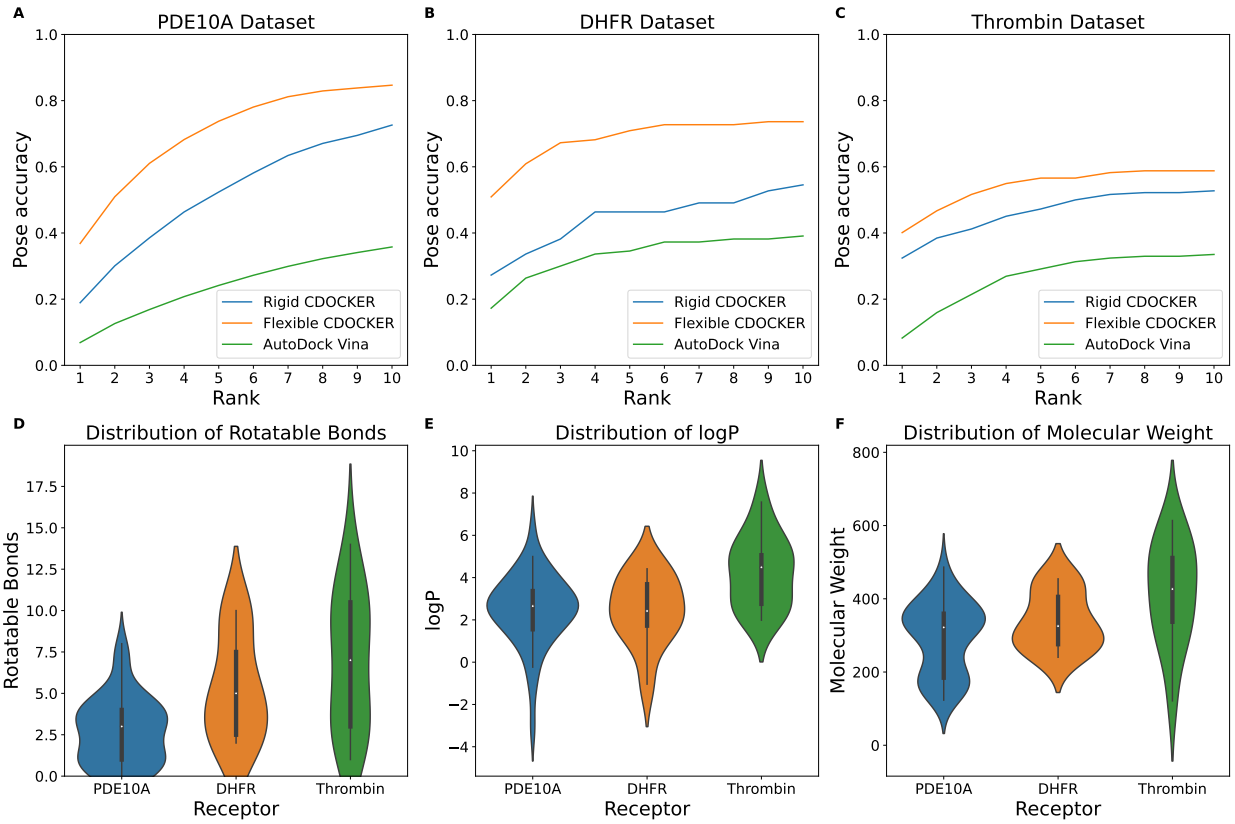

Figure 3.7: Cumulative docking accuracy for the (A) PDE10A dataset, the (B) DHFR dataset and the (C) Thrombin dataset. Distribution of ligand property: (D) rotatable bonds, (E) logP and (F) molecular weight. A Rank of N means the correct docking pose is within the top N solutions. AutoDock Vina trials use an exhaustiveness of 20. Rigid CDOCKER uses 500 docking trials for each cross-docking experiment.

We again see better performance for flexible docking compared to the rigid docking. The ranking results show that our proposed flexible docking algorithm works well for these complex systems. Having ions or cofactors does not appear affect the performance of the purposed flexible docking algorithm. The largest improvement in the top rank accuracy is observed in the DHFR dataset, which is 23.64% higher than rigid docking. The relatively

Table 3.5: Top Rank Accuracy in Pose Prediction of Cross-docking/Re-docking

Receptor Name	Flexible CDOCKER	Rigid CDOCKER	AutoDock Vina
PDE10A	36.84% / 50.00%	18.92% / 47.72%	6.87% / 31.82%
DHFR	50.91% / 54.54%	27.27% / 36.36%	17.27% / 54.54%
Thrombin	40.11% / 50.00%	32.42% / 50.50%	8.24% / 14.28%

low performance for the AutoDock Vina against the PDE10A could result from the fact that it does not support metal ions.

Impact of ligand initial placement on pose prediction accuracy. One general problem in docking is how to place the ligand in the vicinity of the binding site and what initial ligand internal conformation to choose. The initial conformation and position of a ligand might be in an unfavorable configuration relative to the binding site (i.e., incorrect orientation or conformation). Due to the size of the large ligands, instead of adopting the correct conformation and reorienting in the binding pocket, they often need to leave the binding pocket, reorient the conformation and re-enter the binding pocket. This is very unlikely because of the relatively abbreviated sampling schedules required to make high-throughput docking feasible, and is evident from the decrease in pose prediction accuracy for systems with more flexible and large ligands as shown in Figure 3.6 and Figure 3.7 for all three docking algorithms. To explore the impact of searching exhaustiveness of ligand conformational space in docking, we designed and performed another set of flexible and rigid cross-docking experiments for the DHFR and Thrombin datasets. In this test, half of the docking trials (250 docking poses) in the initial generation had the ligand start with an internal conformation that matched that of the native bound conformation, followed by the random rotation and translation in docking with Flexible CDOCKER and Rigid CDOCKER, while the remaining initial configurations were chose as described earlier using obrotomer. For AutoDock Vina, half of the initial ligand configurations matched the native ligand's conformation and the other half was randomly distributed.

As is clear from Figure 3.8, by having some of the ligands starting with the native internal conformation, we observed improved accuracy in pose prediction for all three docking methods. It is surprising that ligand internal conformation also affects the docking performance

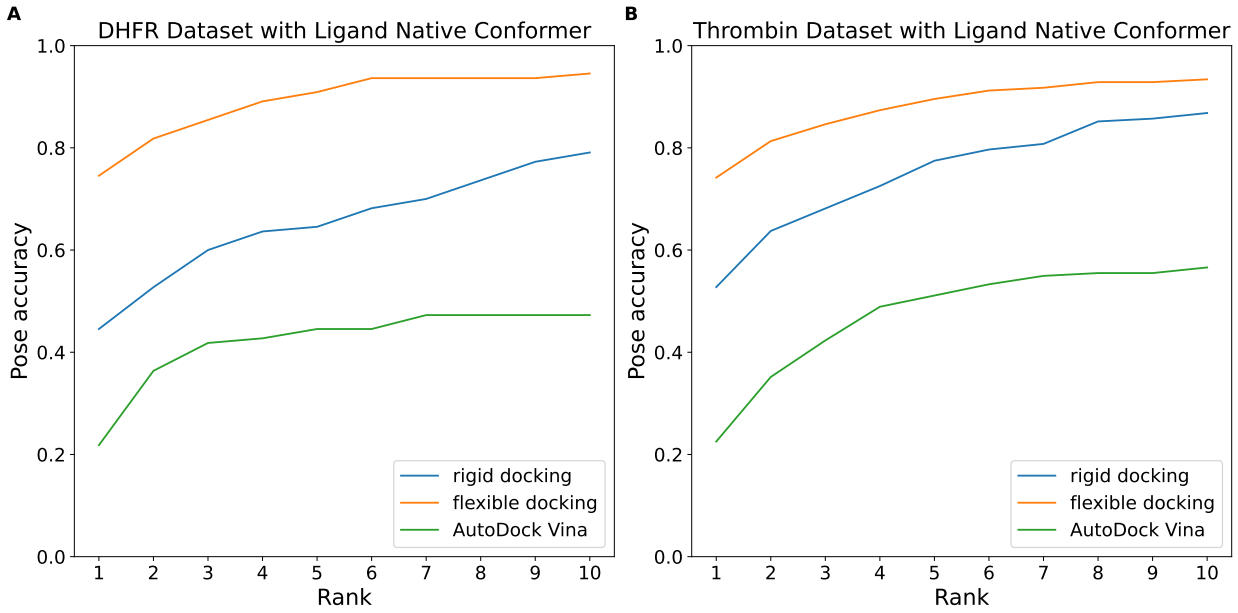

Figure 3.8: Cumulative docking accuracy for the (A) DHFR dataset and the (B) Thrombin dataset with some ligands starting with their native pose internal conformations. Top rank pose prediction accuracy of flexible docking is 74.54% and 74.18% using the proposed flexible receptor docking algorithm for each dataset, respectively.

of AutoDock Vina, however, the results are clear. In real-world applications, the correct conformation will not be known beforehand. This suggests that either better methods to choose a ligand's initial conformation and placement in the binding site, or the use of more extensive sampling may improve docking results for large flexible ligands. This is not a surprising result because the sampling space for highly flexible ligands is large and complex, and alternative approaches for ligand initial conformation picking and placement in the binding pocket are a topic of ongoing exploration.

3.3.2 Discriminating Binders from Non-binders

In a real application for lead compound discovery, one typically would face two questions: (1) For a given target with no knowledge of any inhibitors, how well does a docking method identify novel lead compounds within a large chemical space? If we have already identified several inhibitors for a specific target, one would expect that the derivatives of these known inhibitors could contain binders and non-binders for this target. (2) Then the question be-

comes how well does a docking method perform in discriminating binders from non-binders among the derivatives? To examine the effectiveness of our new sampling and scoring methods in distinguishing non-binders from binders, we perform flexible docking experiments against both ligands and non-binding decoys. The area under the curve (AUC) value of the receiver operating characteristic (ROC) curve is used to evaluate the performance in distinguishing the non-binders from binders.[95]

Scoring function. In order to compare different small molecules, we need to augment our scoring function to consider the system in the unbound state. Because these compounds are docked to the same protein target, the protein internal energy and entropy in the unbound state is a constant and can be neglected. The proposed scoring function is augmented by subtracting the ligand internal energy and conformational entropy in the unbound state (eq 3.9).

$$\begin{aligned}\Delta G_{binding} = {} & \overline{E}_{protein-ligand\ interaction} + \overline{E}_{protein\ internal\ energy\ at\ bound\ state} \\ & + \overline{E}_{ligand\ internal\ energy\ at\ bound\ state} + k_B T \sum_{all\ side\ chains} \sum_{j} p_j ln\ p_j \\ & - \overline{E}_{ligand\ internal\ energy\ at\ unbound\ state} + TS_{ligand}\end{aligned} \tag{3.9}$$

The ligand internal energy in the unbound state is calculated by generating 500 ligand random conformations using Open Babel. After minimizing each of these conformations, the ensemble average of the 500 energies is computed as the ligand internal energy in the unbound state. The ligand conformational entropy (S_{ligand}) is calculated based on the microscopic definition of entropy (eq 3.10). We assume that the rotatable bonds of the ligand are independent of each other and equally sampling all three states (i.e., trans, gauche- and gauche+). The scoring function in Rigid CDOCKER has the same modification to calculate solvation free energy, ligand internal energy and entropy at unbound state.

$$S_{ligand} = -k_B N_{rotors} ln \frac{1}{3} \tag{3.10}$$

The solvation free energy difference is computed using two different approaches: (1) implicitly represented in the proposed scoring function by the distance dielectric constant of

$3r$.[94] and (2) rescoring the docked pose using the FACTS implicit solvent model.[40] Because we perform clustering towards of the docked poses and collect the minimum energy pose (best individual) from each of the top 10 largest clusters as we described previously. These 10 docked poses (ligand and receptor) are rescored with the FACTS implicit solvent model with a short minimization to better estimate each enthalpy terms in eq 3.9 while maintaining the low computational cost. Because this is a short minimization (1000 steps), we assume the side chain conformational entropy remains the same for each cluster. The computational cost for FACTS implicit sovlent model rescoring is about 10% of the average runtime of the proposed docking algorithm.

Identifying novel inhibitors among a large chemical space. The Database of Useful Decoys-Enhanced (DUD-E) contains a large number of experimentally verified actives and decoys, and has been widely used in testing different docking methods.[95] These decoys are generated so that they are physico-chemically similar and topologically dissimilar to the known actives.[95,96] Compared with the original DUD dataset, the ligands in each target are clustered using Bemis-Murcho atomic frameworks to ensure chemotype diversity (i.e., filtering out actives with similar topology features).[95] Here, we perform flexible docking experiments against the 3 receptor targets in the DUD-E dataset, mineralocorticoid receptor (MCR), glucocorticoid receptor (GCR) and androgen receptor (ANDR) and use DOCK for direct comparison because these receptors were a focus of their earlier studies.[95] All three receptors have hydrophobic pockets with flexible binding site residues and are recommended by the Shoichet group for testing flexible receptor docking methods.[95]

Table 3.6: AUC Value for Docking Against MCR, GCR and ANDR.

Receptor Name	Flexible CDOCKER[a]	Flexible CDOCKER[b]	DOCK[c]
MCR	58.20	39.53	36.29
GCR	65.76	53.83	43.92
ANDR	55.60	47.73	51.06

[a] Solvation free energy is calculated using FACTS implicit solvent model. [b] Solvation free energy is calculated using a distance dielectric constant of $3r$. [c] AUC values as reported in the original DUD-E paper.[95]

As is shown in Table 3.6, Flexible CDOCKER has better performance than DOCK in

identifying binders. It is not a surprising result that using the FACTS implicit solvent model significantly improves the docking results. Different research groups have tried implicit solvent models for physics based scoring function and observed improved results.[95,97–99] On the other hand, in the original DUD-E paper, large variation of the AUC value was also observed based on their rigid receptor docking protocol when using different receptor structures.[95] This also supports the necessity of flexible receptor docking methods.

Discriminating binders from non-binders. After identifying known inhibitors for a specific target, one often constructs a compound library of the derivatives of these known ligands and anticipates that docking will rank binders among the top ranks. To examine the effectiveness of our new sampling and scoring methods for flexible receptor docking in distinguishing non-binders from binders, we selected the T4 L99A and T4 L99A/M102Q decoy sets that were collected and constructed by the Shoichet group and perform flexible docking experiments. These two receptor targets are well-defined and have been widely used for evaluating docking methods.[75] The T4 L99A decoy set contains 64 ligands and 66 experimentally validated non-binders.[75,97,99–104] The T4 L99A/M102Q decoy set contains 33 ligands and 25 experimentally validated non-binders.[75,97–99]

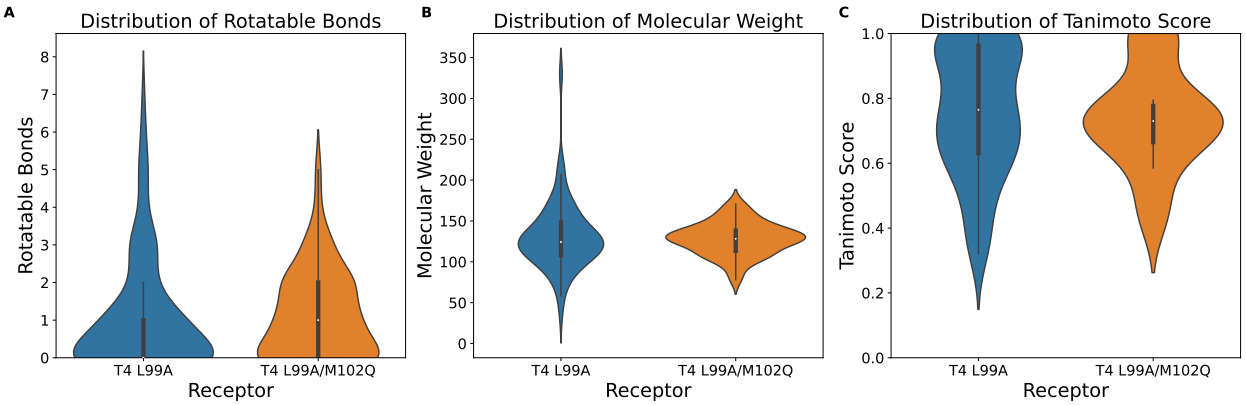

Figure 3.9: Properties of compounds in T4 L99A decoy set and T4 L99A/M102Q decoy set.

We compute the Tanimoto score between each pair of non-binder and binder and record the largest Tanimoto score for a given non-binder (i.e., the maximum similarity for a given non-binder among binders). As shown in Figure 3.9, the majority of the non-binders in these

two decoy sets are similar to the binders, which fits the purposes of this experiment. These compounds are also relatively small and rigid.

Table 3.7: Summary Average AUC Values for T4 L99A and T4 L99A/M102Q Decoy Sets

Method	T4 L99A decoy set	T4 L99A/M102Q decoy set
Flexible CDOCKER[a]	81.10 ± 5.62	67.72 ± 2.61
Rigid CDOCKER[a]	78.96 ± 6.67	66.94 ± 3.54
Flexible CDOCKER[b]	63.43 ± 3.55	48.77 ± 3.29
Rigid CDOCKER[b]	61.44 ± 5.73	48.41 ± 4.30
AutoDock Vina[c]	56.71 ± 0.02	53.33 ± 0.03
AutoDock Vina[d]	56.70 ± 0.01	53.32 ± 0.06

[a] Solvation free energy is calculated using FACTS implicit solvent model. [b] Solvation free energy is calculated using a distance dielectric constant of $3r$. [c] Exhaustiveness = 8. [d] Exhaustiveness = 20.

All receptor structures in these two sets as well as the corresponding unique flexible side chain selections are used for the docking experiment. AutoDock Vina and Rigid CDOCKER are used for direct comparison. Average AUC value of the ROC curves are shown in Table 3.7. One key reason for the relatively low AUC values observed for T4 L99A/M102Q in this table is the small number of compounds in the set, since a small change in the compound ordering will make a big difference in the ROC curve (i.e., moving the order of one non-binder will result in a change of 4% in false positive rate). Flexible CDOCKER with the FACTS implicit solvent model has the best performance which agrees with the results in the previous experiment. Both Flexible CDOCKER and Rigid CDOCKER have high accuracy in distinguishing binders from non-binders. However, Flexible CDOCKER has a smaller standard deviation across different receptor types. This also suggests that our proposed flexible receptor docking algorithm is better at modeling side chains in the binding pocket when different ligands bind.

3.4 Conclusions and Discussions

In prospective applications, such as virtual screening, it is more common that both ligand and receptor undergo conformational changes upon binding. In these cases, accurate prediction

of the binding pose is fundamental before one uses such methods to conduct any structure-function exploration. Many research groups have shown that flexible receptor docking is more accurate in finding native-like docking poses for a given ligand.[7,13,14] In the present work, we have provided further support for the importance of flexible receptor docking approaches, and presented a revised flexible docking algorithm, including a new physics-based scoring function incorporating side chain conformational entropy and an updated hybrid searching algorithm combining molecular dynamics (MD) based simulated annealing and a continuous genetic algorithm. The new physics-based scoring function provides a framework for the computation of the side chain conformational entropy, which allows us to explore and quantify the conformational variance when different ligands bind or a ligand binds with different poses. Overall, the cross-docking results we present show that the proposed flexible receptor docking algorithm provides greater accuracy in identifying native-like poses as top rank in protein-ligand docking and RNA-ligand docking compared with rigid docking. The largest improvement in top ranking accuracy is 23.64% for ligands binding to DHFR. We also show that the proposed flexible receptor docking algorithm with the FACTS implicit solvent model has the ability to identify novel compounds and distinguish binders from non-binders.

Table 3.8: Average Runtime for Different Flexible Receptor Docking Algorithm

Methods	Runtime
Glide	400 CPU hours and 50 GPU hours for 20 docking trials
AutoDockFR	365 hours with 10 flexible side chains
Flexible CDOCKER	100 minutes for 500 docking trials with 10 flexible side chains

Flexible receptor docking methods have, however, been less adopted because of the relatively large computational cost. As shown in Table 3.8, compared with AutoDockFR[13] and the Glide flexible receptor docking algorithm[14], our proposed flexible receptor docking method significantly reduces the computational cost. We realize that the proposed flexible receptor docking method is still expensive compared with rigid receptor docking methods. But, we suggest that the speed-ups we observe are sufficient such that they significantly broaden the scope of flexible receptor docking methods in high-throughput docking campaigns.

Chapter 4

TMPRSS2 Inhibitor Discovery Facilitated Through an in Silico and Biochemical Screening Platform

Amanda L. Peiffer, Julie M. Garlick, Yujin Wu, Jesse W. Wotring, Matthew B. Soellner, Jonathan Z. Sexton, Charles L. Brooks III and Anna K. Mapp. "TMPRSS2 Inhibitor Discovery Facilitated Through an in Silico and Biochemical Screening Platform". *Manuscript under review* DOI: 10.1101/2021.03.22.436465. I developed the docking method and performed high throughput virtual screening. The experimental validations were performed by Amanda L. Peiffer, Julie M. Garlick and Jesse W. Wotring. The detail of wet lab experimental setup is presented in Chapter APPENDIX.

4.1 Introduction

The emergence of COVID-19 in late 2019 and the rapid transmission of the disease around the globe has prompted an urgent need for effective treatments.[109] As with many coronaviruses, infection with nearly all SARS-CoV-2 variants requires host cell cooperation; the spike (S) protein protruding outside the viral coat requires priming by TMPRSS2, a human transmembrane serine protease, for viral entry via the receptor Angiotensin converting enzyme 2 (ACE2) (Figure 4.1A).[105,110–112] While many have focused on blocking the inter-

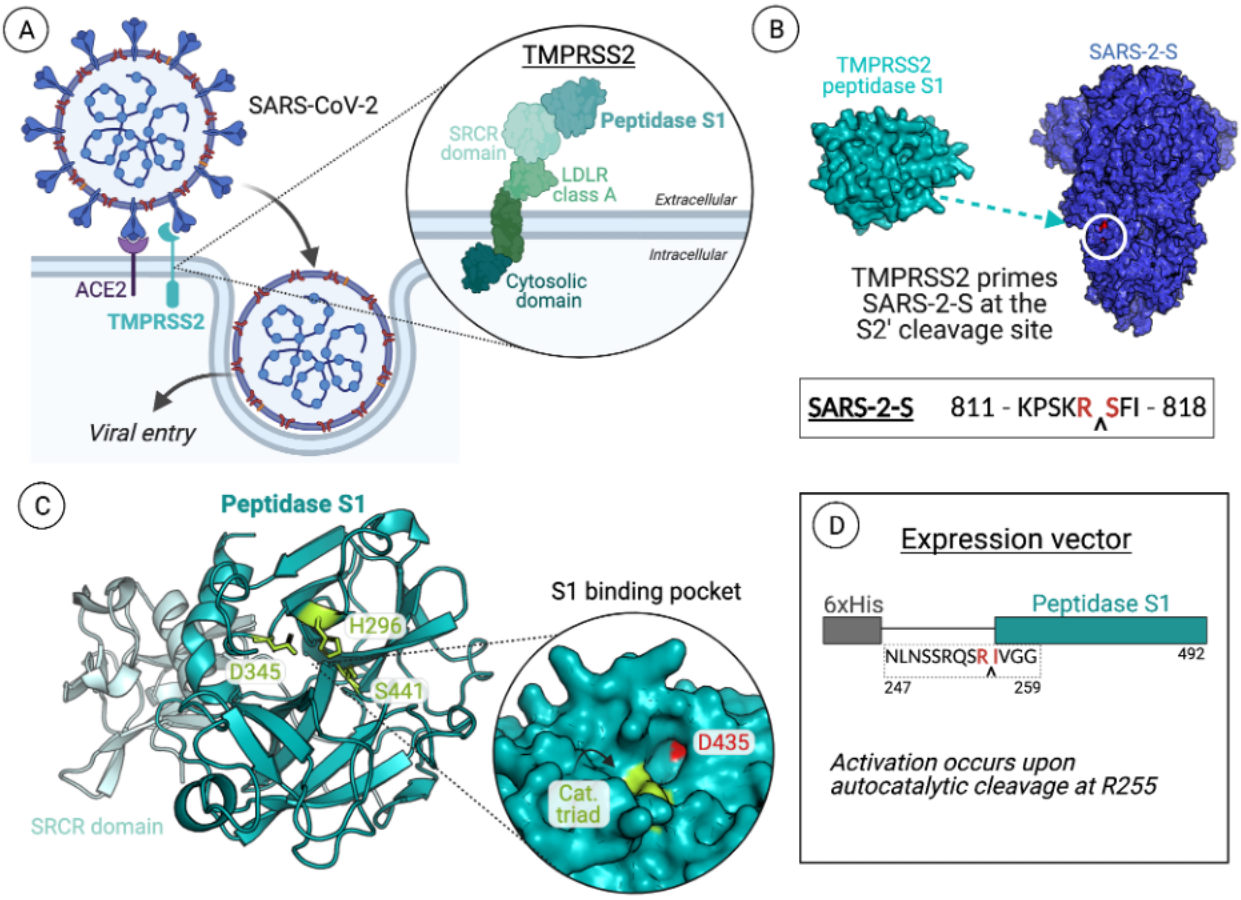

Figure 4.1: The role of TMPRSS2 in SARS-CoV-2 infection. **(A)** TMPRSS2 primes the viral S protein (SARS-2-S), which promotes membrane fusion and ultimately viral entry. TMPRSS2, part of the type II transmembrane serine protease family and hepsin/TMPRSS subfamily, is anchored at the cell membrane.[105] The protein is mostly extracellular, with a small intracellular cytosolic domain. The extracellular portion of the protein is composed of a LDLR class A domain, an SRCR domain, and finally the peptidase S1 domain required for catalytic activity.[106] **(B)** The peptidase S1 domain of TMPRSS2 cleaves SARS-2-S at the S2' cleavage site.[107,108] **(C)** Homology model of TMPRSS2 SRCR and peptidase S1 domains, generated using SWISS-Model. Peptidase S1 domains are chymotrypsin-like and contain the catalytic triad histidine, aspartic acid, and serine (active site residues for TMPRSS2 are H296, D345, S441).[106] **(D)** The DNA vector constructed for recombinant expression and purification for the protease domain of TMPRSS2.

actions between ACE2 and the S protein, ACE2 also plays an important role in healthy cell function by counterbalancing ACE to lower and maintain healthy blood pressure.[113] Alternatively, there is little known about the biological function of TMPRSS2, with data suggesting it is likely functionally redundant.[114,115] Along with SARS-CoV-2, TMPRSS2 has

been implicated in priming other pathogenic coronaviruses such as SARS-CoV and MERS, as well as influenza.[116–119] TMPRSS2$^{-/-}$ knockout mice had little phenotypic differences compared to wild-type animals, yet conferred resistance to viral infections, suggesting that the protein is not essential.[120] Inhibiting the expression of TMPRSS2 via Bromodomain and Extra-Terminal motif (BET) inhibitors leads to decreased infectivity of SARS-CoV-2 in human lung cells, further suggesting the viability of TMPRSS2 inhibition as an antiviral strategy.[121] Thus, TMPRSS2 is a desirable drug target for treating SARS-CoV-2 and future coronavirus infections.

To date, there are few established TMPRSS2 inhibitors, although recent studies have identified ketobezothiozole-based peptides that are effective at inhibiting TMPRSS2.[122] Camostat, a compound initially discovered as a Matriptase 2 inhibitor, also inhibits TMPRSS2 and has been clinically approved in some countries to treat chronic pancreatitis.[105,123] Nafamostat and gabexate have also been reported to inhibit TMPRSS2. Both camostat and nafamostat inhibit a wide range of serine proteases and are rapidly metabolized in mammals to structures with poorly defined activity.[124–126] It has been reported that each of these compounds all form a covalent bond with the active site serine of serine proteases via the central ester, also a site of metabolic breakdown.[35,36] Additionally, molecular modeling studies on TMPRSS2 supports covalent bond formation with both camostat and nafamostat, as well as the camostat metabolite FOY 251.[127] Thus, TMPRSS2 inhibitors with less reactive architectures are highly desirable.

As a strategy for rapid TMPRSS2 inhibitor discovery, we developed a combined in silico and biochemical workflow to identify approved drugs that inhibit TMPRSS2. Strategic development of a TMPRSS2 expression protocol, utilizing autocatalysis-based affinity tag removal, facilitated purification for biochemical assay development. This allowed existing TMPRSS2 inhibitors to be profiled and characterized as covalent. Our protocol for virtual screening against a TMPRSS2 homology model (Figure 4.1C) comprised the flexible receptor docking method[12] described in Chapter 3. Our approach curated a subset of promising TMPRSS2 ligands, which upon subsequent biochemical testing were identified as active inhibitors. In doing so, we identify new non-covalent hit compounds that are approved drugs and can be both repurposed for SARS-CoV-2 infections as well as derivatized to yield

improved TMPRSS2 inhibitors.

4.2 Methods

4.2.1 Construction and Refinement of TMPRSS2 Homology Model

Virtual screening methods have greatly improved over the past two decades, leading many drug discovery campaigns by filtering out thousands/millions of molecules before testing them in vitro. However, such studies require a structural model in which to dock compounds into the active site. Because no crystallographic or NMR-based models existed for TMPRSS2 at the time of this work, we developed a homology model for the active soluble domain starting from prediction using the SWISS-MODEL web-interface (Figure 4.1C).[128–132] This structure was built based on sequence homology of hepsin (PDB 5CE1). The structure showed high homology with the TMPRSS2 peptidase domain (34% sequence similarity with 70% sequence coverage) and also contained the bound ligand 2-[6-(1-hydroxycyclohexyl)pyridin-2-yl]-1H-indole-5-carboximidamide, which served as one of the templates for pharmacophore-based docking of putative ligands as described below. The SWISS-MODEL structure of TMPRSS2 was further "conditioned" through the application of molecular dynamics in an implicit solvent (GBMV) model to facilitate better packing and configurational relaxation.[133–135] Comparing our homology model to the reported crystallographic structure of TMPRSS2 that was solved after our studies,[136] we find that the predicted structure overlays well with the peptidase domain (RMSD of 2.6 Å using all atoms).

4.2.2 Ligand Representation

The fastdock protocol is a python-based workflow that integrates the align-it software[137] to search across our curated library of compounds for 3D pharmacophore matches to an inhibitor from a solved structure. The fastdock ligand templates are taken from the Hepsin structure used in the initial generation of the model (PDB 5CE1) as well as from a plasma kallikrein structure with the 1 nM inhibitor N-[(6-amino-2,4-dimethylpyridin-3-yl)methyl]-1-(4-[(1H-pyrazol-1-yl)methyl]phenylmethyl)-1H-pyrazole-4-carboxamide bound (PDB 6O1G;

43% sequence similarity and 51% sequence coverage). Scoring of the pharmacophore matches is based on a volumetric Tanimoto value of the target ligand pharmacophore map and the reference ligand map. Based on this initial selection of potential ligands for exploration, we harvested 1-10% of the top hits.

The MMTSB tool set[58] was used to cluster binding poses and prepare pdb files. Open Babel[71] was used to generate ligand random conformations. MOE (Molecular Operating Environment)[70] was used to predict the correct protonation state for the ligands at pH 7.4. ParamChem[72,73] was used to prepare the ligand topology and parameter files with the CGenFF force field. Clustering used the tool cluster.pl with a 1 Å cutoff radius for the K-means clustering. The CHARMM C36 force fields[49] were used and docking was performed in CHARMM[48] with the CHARMM/OpenMM parallel simulated annealing feature.[7]

4.2.3 General Docking Setup

Flexible CDOCKER with FACTS implicit solvent model described in Chapter 3 was used to dock and rank the top hits.[12] Flexible CDOCKER uses a physics-based scoring function and allows both ligand and protein side chains to explore their conformational space simultaneously. The following amino acid side chains are considered flexible : His-296, Tyr-337, Lys-342, Asp-435, Ser-436, Gln-438, Ser-441, Thr-459, Trp-461 and Cys-465.

Each docking measurement represents 500 genes (docking trials). The coordinates of the ligand-protein flexible side chains are used to assemble a gene (potential docking pose). Each ligand in the dataset is first aligned to the pharmacophore model with align-it. In the initial generation, half of the genes have the ligand starting with the aligned position. The rest of the genes are constructed by generating a random conformation of the ligand with Open Babel and centering at the binding pocket. A random translation (within a volume with a 2 Å edge length) and rotation (maximum 360°) are performed on ligands in each gene. An energy cutoff is applied to avoid potential collision between ligand atoms and protein atoms due to the random translation and rotation. The protein flexible side chains are initialized with the coordinates from the input homology model. Then these genes are optimized by an MD based simulated annealing algorithm. Detailed values for softness parameter Emax used in flexible receptor docking are discussed in Chapter 2.

The docking poses (optimized genes) are then K-means clustered based on ligand heavy atom RMSD with a radius cutoff of 1 Å. We then select the best individuals (minimum energy pose) from the top 10 largest clusters to construct the second generation. In Chapter 3, we show that using two generations is adequate and the average computer time for each docking measurement is around $30 \sim 45$ minutes. After the second generation, the docking poses are clustered and the best individuals from the top 15 largest clusters are saved. These docking poses are then rescored using the FACTS implicit solvent model.

4.3 Results

4.3.1 Virtual Screening Yields Preliminary Hits for in Vitro Assays

Extensive virtual screening was performed to obtain putative hits for follow-up testing via in vitro inhibition assays (Figure 4.2). A total of 134,109 molecules were collected from multiple databases, which were subjected to a hierarchical refinement of docking poses. In the first stage, rigid receptor docking was performed exploring two means of initially positioning the small molecules. One utilized a novel 3D pharmacophore fastdock framework, which operates by superposing pharmacophores onto compounds bound in experimentally solved structures in other bound serine protease structures; the other initiated from a random generation of molecular conformations and random positioning inside the pocket (Figure 4.2A).

From this initial stage, 4,307 Level 1 screening candidates were determined and subjected to GPU accelerated Flexible-CDOCKER methods described in Chapter 3. This approach utilizes flexible side chains for residues in or near the binding pocket while using a grid representation for the remaining receptor. Multiple copies of each set of side chains and initial ligand poses are created, which allows for parallel, multiple copy processing of multiple flexible ligands-flexible receptor trials simultaneously on GPUs. The flexible docking searching algorithm combines molecular dynamics (MD) based simulated annealing and a continuous genetic algorithm search protocol to enhance the sampling of differing receptor conformations.

We utilize a novel scoring methodology by performing conformational clustering of the

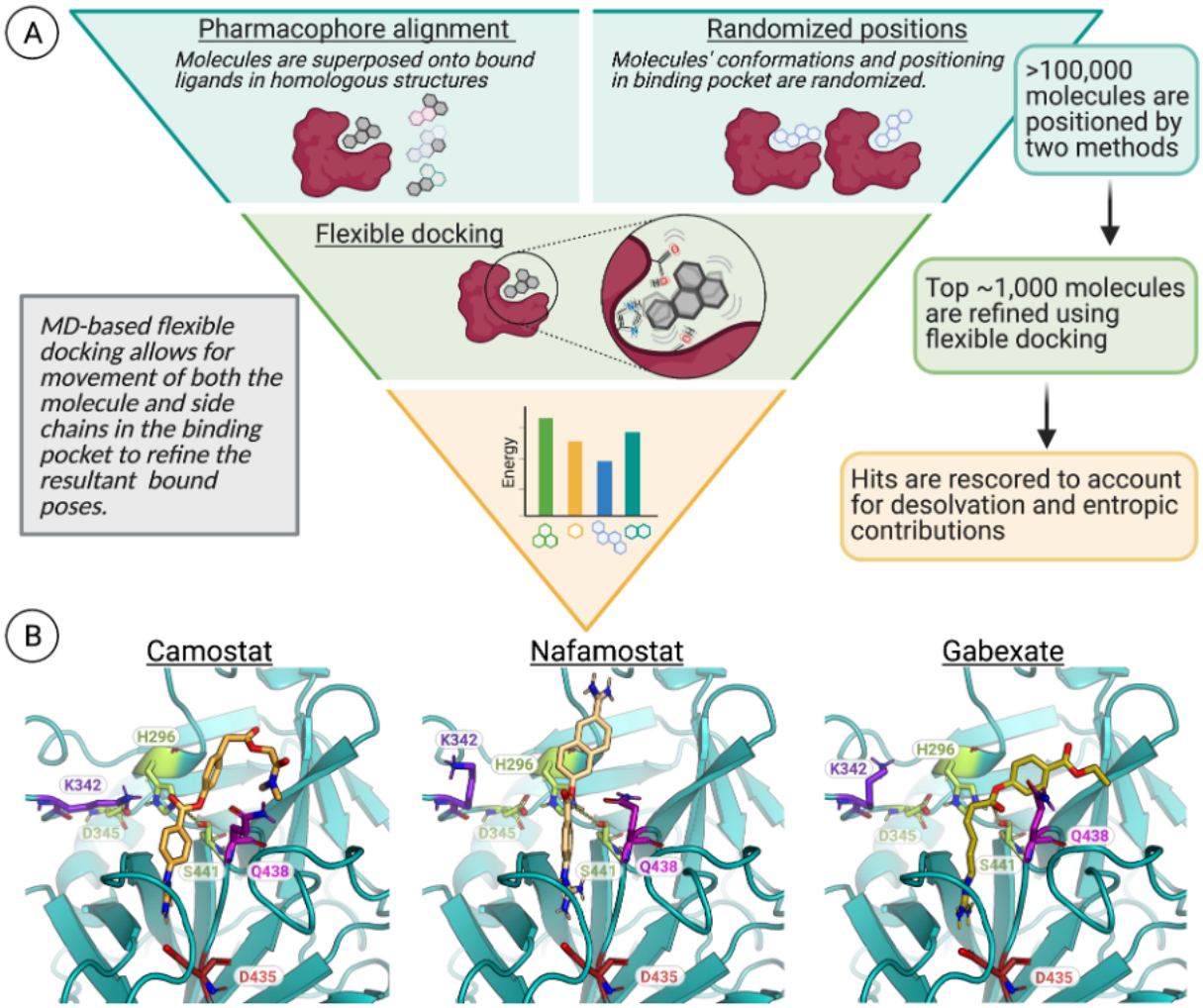

Figure 4.2: The hierarchical docking workflow overview. **(A)** The virtual screening protocol involves multiple refinements, including pharmacophore based screening and Flexible CDOCKER. **(B)** Docked poses of camostat, nafamostat, and gabexate in the TMPRSS2 active site with the proton transfer from Ser-441 to His-296 that occurs after substrate binding. Shown in red is D435, which resides at the bottom of the S1 binding pocket, and lime green corresponds to the catalytic triad (His-296, Asp-345, Ser-441). Dashed lines shown here indicate the distance between the catalytic serine oxygen and the carbonyl carbon of each inhibitor (camostat = 3.5 Å, nafamostat = 4.8 Å, gabexate = 3.5 Å).

flexible side chains and the ligand, which provides key contributions to the ligand scoring from the entropic variation of the side chains to accommodate various ligand poses. The ligands are rescored in the protein binding site a final time using an implicit solvent model that captures aspects of the desolvation costs not generally accessible in typical docking methods. The rescoring is accomplished by minimizing the docked poses from the flexible

side chains and flexible ligand in the context of the rigid protein, while also considering the total energy of the solvated docked and undocked systems.

The virtual screen was successful on many fronts. MD-based flexible docking identified residues near the active site that are conformationally dynamic to accommodate different ligands. Residues Gln-438 and Lys-342 in particular show the greatest conformational change upon ligand binding, suggesting that they participate in stabilizing bound compounds. The known inhibitors camostat, nafamostat, and gabexate all ranked in the top 5 compounds, and all adopted poses that demonstrate how the catalytic serine residue positions itself to ultimately react with the inhibitors while positioning the guanidinium functionality to form a salt-bridge interaction with the active site Asp-435. As the mechanism of covalent inhibition involves His-296 deprotonating Ser-441, we performed a subsequent docking experiment using the three molecules with these charge changes (Figure 4.2B). For all three molecules, the deprotonated serine positions itself into a more reactive state to attack the carbonyl carbon. For instance, the distance between the serine oxygen and camostat carbonyl carbon decreased from 4.9 Å to 3.5 Å, and the distance for gabexate decreased from 5.9 Å to 3.3 Å. While nafamostat appears to be further away from the reactive carbon (4.4 Å to 4.8 Å), the molecule flips so that the carbonyl is positioned for reactivity.

4.3.2 Identification of Noncovalent Inhibitors

Several clinically approved drugs emerged as top ranked compounds in the virtual screen, which we selected to test in our in vitro assay. Like the covalent inhibitors, pentamidine, propamidine, and debrisoquine all contain a guanidinium moiety and docked into the active site of TMPRSS2 with the positive charge pointing towards Asp-435 (Figure 4.3A). Biochemically, we found that all three molecules did in fact inhibit TMPRSS2 activity, with debrisoquine being the least potent (Figure 4.3B). The docked poses of pentamidine and propamidine show both compounds are positioned to block the active site residues, whereas debrisoquine does not fully span the catalytic triad, which likely contributes to the differences in potency (Figure 4.3A). Pentamidine and propamidine are of similar size to camostat and nafamostat, typical of small molecule inhibitors ($> 350MW$), while debrisoquine is quite small, at $175.2MW$, classifying it as a fragment rather than a small molecule. However, de-

brisoquine has the greatest ligand efficiency (LE) at $0.42 kcal/mol$ compared to pentamidine and propamidine, which are $0.33 kcal/mol$ and $0.31 kcal/mol$ respectively.

Debrisoquine, pentamidine and propamidine were tested for their ability to inhibit SARS-CoV-2 viral infection in Calu-3 cells, a human lung cell line (Figure 4.3C), and camostat was included as a control. Of the three molecules discovered from our screening process, debrisoquine showed the greatest inhibition of viral infection, with an IC_{50} value of $22 \pm 1.5 \mu M$ and no decrease in cell viability up to 500 μM (Figure 4.3D). Validation of TMPRSS2-dependent inhibition was tested in viral infectivity assays using the human intestinal epithelial cell line Caco-2. Caco-2 cells express significantly lower levels of TMPRSS2 than do Calu-3 cells.[138] Further, SARS-CoV-2 viral entry in Caco-2 cells has been shown to be far less dependent on TMPRSS2, with entry occurring via a cathepsin-mediated mechanism.[139] We found that debrisoquine was unable to inhibit infection of Caco-2, confirming the TMPRSS2 specific mechanism of action for debrisoquine. Thus, the high LE of $0.42 kcal/mol$ in our in vitro assays in conjunction with the reduced viral infection seen in the Calu-3 cellular model suggests that debrisoquine would be an excellent starting point for fragment expansion to increase potency for TMPRSS2 inhibition.

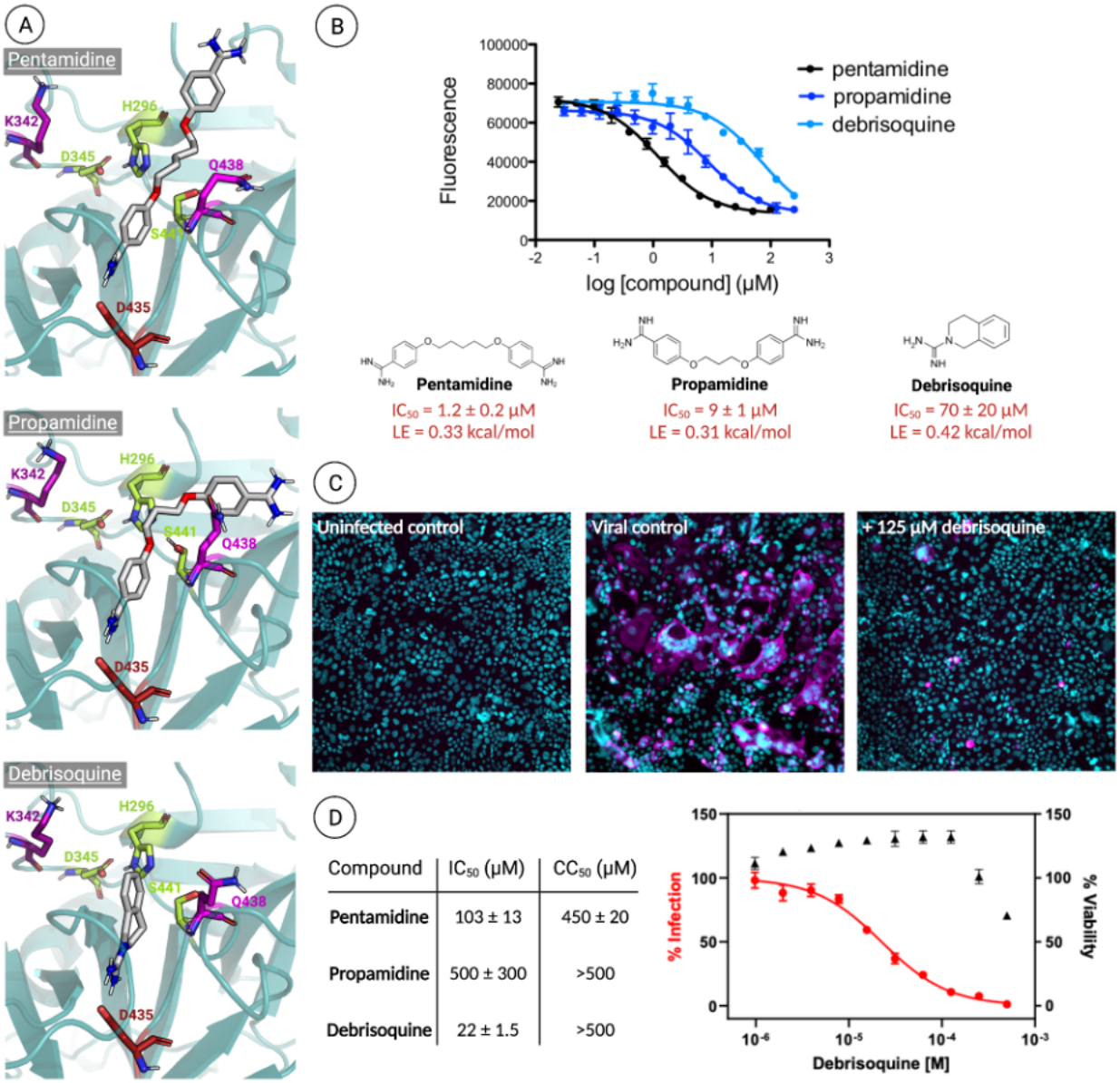

Figure 4.3: The hierarchical docking workflow screening result. **(A)** Docking results for the three drugs identified as hits both in the virtual screen and the in vitro assay. All three molecules fit into the active site (Asp-435 at the bottom of the pocket shown in red). Pentamidine and propamidine obstruct access to the catalytic triad (residues shown in lime green), whereas the fragment debrisoquine only partially reaches those residues. **(B)** Computational hits inhibit TMPRSS2 in biochemical activity assay. Top: inhibition assays to obtain IC_{50} curves. Data is the average of duplicate experiments conducted in technical triplicate. Calculated IC_{50} values and ligand efficiencies (LE) are shown below the chemical structures of the hits. **(C)** TMPRSS2 inhibitors have *in vitro* efficacy in the human lung cell line Calu-3. Representative images ($10X$ magnification) of viral and uninfected controls for the anti-SARS-CoV-2 bioassay. Nuclei are shown in cyan, viral nucleocapsid protein is shown in magenta. Right: representative image for the most efficacious (non-toxic) concentrations of debrisoquine. **(D)** Left: IC_{50} and CC_{50} values for the three inhibitors calculated from the infectivity assay. Right: the inhibition curve from the infectivity assay for debrisoquine.

4.4 Conclusions and Discussions

As a practical example of this methodology, we worked with a team of experimental colleagues to identify potential therapeutics for the host transmembrane serine protease TMPRSS2, a promising antiviral target that plays a direct role in SARS-CoV-2 infections. Because no crystallographic or NMR-based models existed for TMPRSS2 at the time of this work, we first developed a homology model as we described in the Methods section. After the crystal structure of TMPRSS2 became available, we compared the docked pose using the homology model and the crystal structure. As shown in Figure 4.4, the docked poses are very similar to each other. This further validates the binding pocket representation in our homology model.

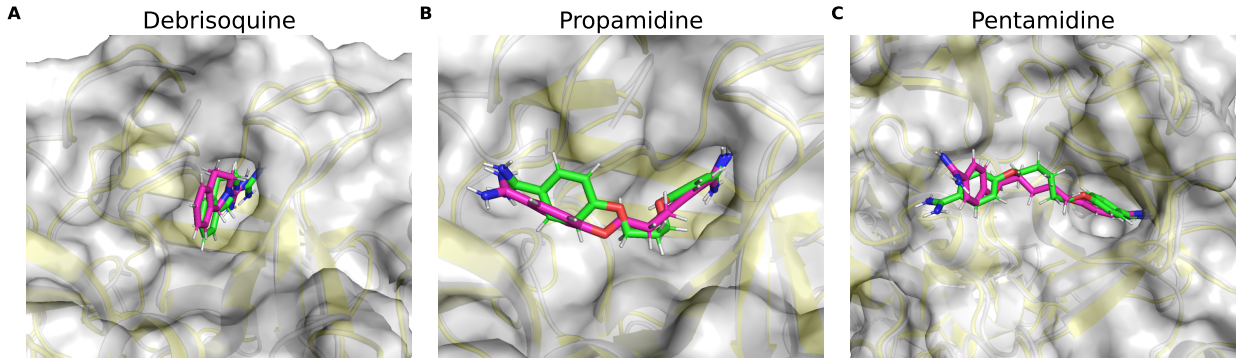

Figure 4.4: Docking poses comparison using homology model and crystal structure for (A) debrisoquine, (B) propamidine and (C) pentamidine. The crystal structure and the corresponding docked pose is shown in grey and green respectively. The homology model and the corresponding docked pose is shown in yellow and pink respectively.

We designed a hierarchical workflow that uses pharmacophore similarity to filter very large compound libraries followed by direct application of the flexible receptor docking and the implicit solvent scoring methodology presented here. A total of 4,308 candidates were identified and docked with the flexible docking method described above and led to the identification of new inhibitors. This hierarchical workflow takes the advantage of the flexible receptor docking method in high-throughput virtual screening for lead compound identification while reducing the overall computational cost. On the other hand, there are no specific changes of the docking protocols in the proposed workflow. Therefore, the proposed hierar-

chical workflow could be applied to other biological targets without any further modification.

Chapter 5

Covalent Docking in CDOCKER

Yujin Wu and Charles L. Brooks III. "Covalent Docking in CDOCKER" *Manuscript under review*

5.1 Introduction

Recently, a crystal structure of human TMPRSS2 in complex with Nafamostat was deposited in the PDB bank.[136] As we mentioned in Chapter 4, nafamostat forms a covalent bond with the receptor side chain Ser-441. We compared the docked pose from the homology model with the crystal structure. As shown in Figure 5.1, the warhead of the docked pose (i.e., the guanidinium functional group) overlaps well with the ligand in the crystal structure and inserts into the binding pocket. This motivated us to develop a covalent docking algorithm as a component of the CDOCKER docking methods.

Targeted covalent inhibitors (TCIs) have gained increased interest in drug discovery in the last two decades, with nearly 30% of currently marketed drugs known to be covalently bound to the therapeutic target.[37–39] TCIs are designed such that the initial, reversible association is followed by the formation of a covalent bond between the ligand and receptor, which strengthens the interactions and increases the potency.[38,39] Tethered docking methods have become an efficient means of structure based TCI design, and has been widely used in identifying lead compounds.[140–142] Generally speaking, docking involves two main components: searching and scoring.[2–4] In one element, the searching generates multiple docking poses of

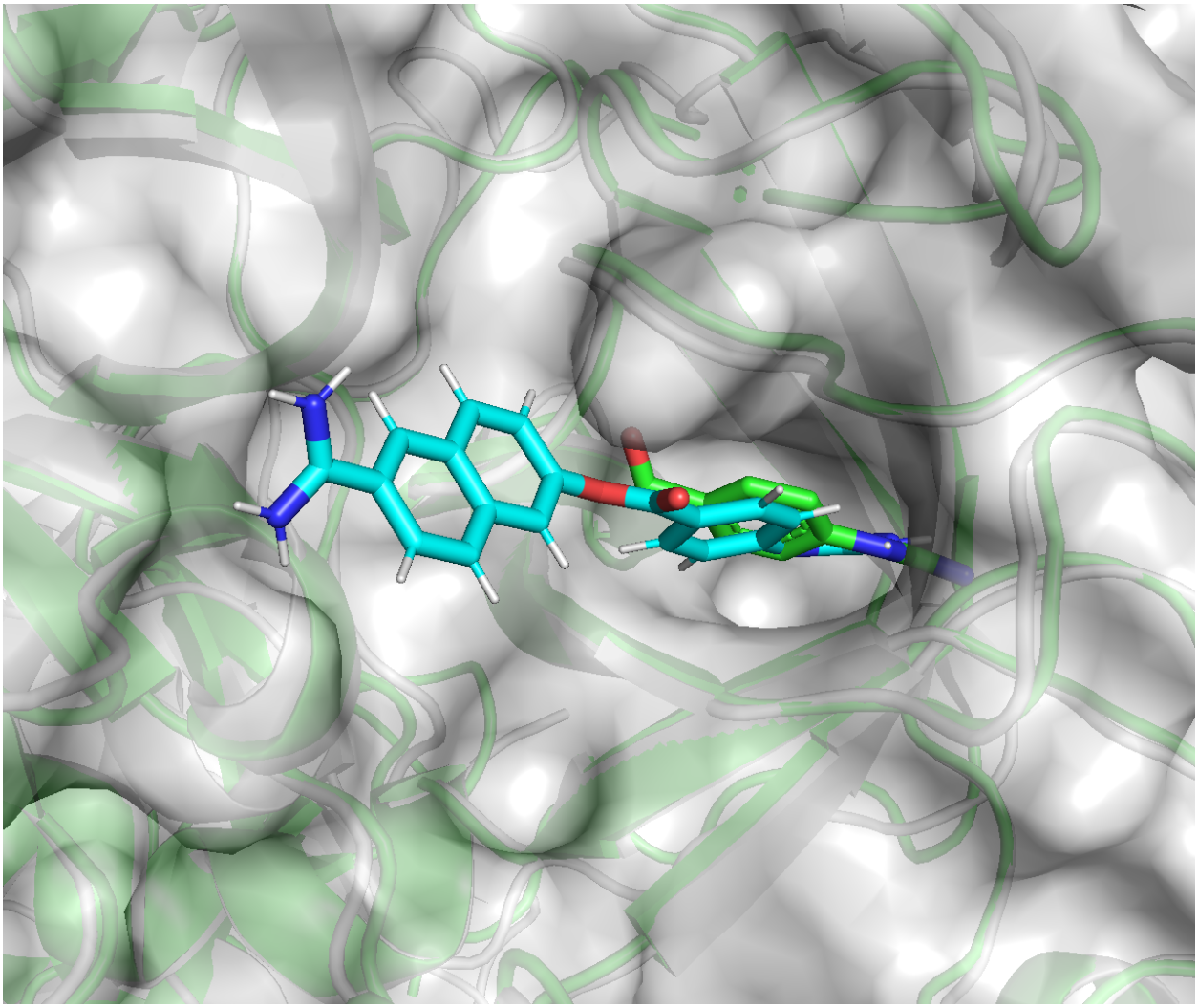

Figure 5.1: Compare nafamostat docked pose with crystal structure. The docked pose is shown in cyan. The crystal structure (receptor and ligand) is shown in green.

a ligand within the constraints of the receptor binding site. The application of a scoring function then ranks these poses and is expected to identify the correct binding pose through the assumption the correct binding pose is at the top rank. Today, multiple off-the-shelf protein-ligand covalent docking programs, either commercial or free, are available for use, such as DOCKovalent[143], GOLD[144–146], AutoDock4[147,148], CovDock[149,150], FITTED[151], ICM-Pro[152] and MOE.[153] In a recent pose prediction challenge to reproduce the binding mode of 207 cysteine-bound covalent complexes, ICM-Pro showed the best performance with the top ranking accuracy of 62%, followed by CovDock(59%), FITTED(56%), AutoDock4(55%),

and GOLD(53%).[154]

The covalent docking methods DOCKovalent, GOLD, AutoDock4, ICM-Pro and MOE handle the ligand in its bound state (i.e. covalent form).[143–146,148,152,153] These methods form a physical bond between the ligand reactive atom to the receptor reactive atom and before searching for the binding pose, which have two potential issues: (1) the sampling space is reduced and does not include the initial, reversible association for ligands from an unbound form, which is essential for TCIs, (2) and the ligand preparation or reaction type generally requires manual definition which can cause difficulties in high throughput screening.

CovDock and FITTED use the ligand in its non-covalent form.[149–151] CovDock has two different versions which are Lead Optimization mode (CovDock-LO)[149] and Virtual Screening mode (CovDock-VS).[150] The first has a better accuracy with higher computational cost and was used in a previous pose prediction challenge.[154] Typically, CovDock-LO requires $1 \sim 3$ hours to dock one compound.[150] The CovDock-VS mode is designed to address high throughput needs with a lower docking accuracy. Both methods will automatically identify the ligand warhead atoms and form a covalent bond during the docking simulations if the warhead is predicted to be in close proximity to the targeted residues.[149–151] On the other hand, FITTED does not allow customization of the warhead, which makes it unable to recognize certain covalent bond formation reactions (i.e., nucleophilic substitution, ring opening and disulfide bridge formation).[151,154] Thus, both docking methods have limitations in real applications for identification of TCIs.

Rigid CDOCKER is a grid based MD docking algorithm where the ligand-receptor interaction energy is precomputed and stored on a grid.[9] This grid-based representation of interactions has been applied widely in many docking protocols which provides computational efficiency while maintaining much of the accuracy of the full force field method. Rigid CDOCKER uses a physics-based scoring function (eq 5.1) and was originally designed for docking reversible inhibitors.[9] In the current work, we implement a covalent docking module with Rigid CDOCKER by introducing an customizable covalent bond grid potential in the scoring function to mimic the free energy change of bond formation between the ligand and the receptor as described below.

$$\Delta G_{binding} = E_{ligand\ internal\ energy} + E_{vdw} + E_{elec} \tag{5.1}$$

5.2 Methods

5.2.1 Benchmark Dataset

Two sets of protein-ligand complexes and the ZINC12 compound library[155,156] are used for optimizing and benchmarking the protein-ligand docking method described below.

Dataset used for optimizing the covalent CDOCKER scoring function and evaluating pose prediction accuracy. We employed the same dataset as the one used in the previous pose prediction challenge, which contains 207 complexes representing 54 protein targets.[154] This dataset contains 7 different chemical reaction types: addition to aldehyde, disulfide bond formation, addition to ketone, Michael addition, addition to nitrile, nucleophilic substitution and ring opening, that are common for TCIs.[38,154] We select this dataset to optimize the scoring function and compare the pose prediction accuracy of the proposed covalent docking algorithm with other covalent docking programs.

Retrospective virtual screening dataset. One major application of docking is to identify lead compounds for a given target. A successful in silico docking protocol can save a large amount of time and money in the drug discovery process. Thus, it is important to evaluate the virtual screening performance for a newly proposed docking algorithm. Many research groups constructed different retrospective virtual screening datasets in developing their covalent docking methods.[143,149,150,157] However, there are different issues with these datasets: (1) they do not include different covalent bond formation reactions of TCIs, (2) there is no additional process for filtering out decoys that are physico-chemically dissimilar with the actives for some of the datasets, and (3) the size of both the active sets and decoy sets are very small for some of the datasets. Therefore, we decided to construct our own retrospective virtual screening dataset.

We did not select the standard ZINC15 library because it has been filtered by ZINC12 clean filters.[156,158] This removes aldehydes and thiols in the ZINC15 standard library, which

Table 5.1: Summary of the ZINC12 Subsets of Different Electrophiles.

Electrophile	Reaction type	SMARTS expression	Number of compounds
Thiol	Disulfide Bond Formation	[#6][#16X2H]	10196
Epoxide	Ring opening	C1OC1	9975
Aldehyde[a]	Addition to aldehyde	[CX3H1](=O)[#6]	48841
Ketone[b]	Addition to ketone	[#6][CX3](=O)[#6]	595283
Nitrile	Addition to nitrile	[NX1]#[CX2]	745350
α,β-unsaturated carbonyl	Michael addition	[CX3](=O)[CX3]=[CX3]	986697

[a] Compounds with both the functional group aldehyde and the functional group α,β-unsaturated carbonyl are excluded for the ZINC12 aldehyde subset. [b] Compounds with both the functional group ketone and the functional group α,β-unsaturated carbonyl are excluded for the ZINC12 ketone subset.

is contradictory to the purpose of constructing virtual screening dataset of the reaction addition to aldehyde. Thus, the entire standard ZINC12 library containing 16 million compounds is used to construct subsets with different electrophiles that correspond to different covalent bond formation reactions. This classification is performed based on SMARTS regular expression (Table 5.1). If the same functional group appears more than once in a compound, then this compound is excluded in the corresponding electrophile subset. Nucleophilic substitution reactions could involve different functional groups and leaving groups, thus it is hard to maintain both specificity and generalization by using only one SMARTS regular expressions. Therefore, we decided to not include a benchmark set for the nucleophilic substitution reaction. Overall, this provides large compound libraries for each of the remaining reaction types and is used for our curation of the decoy sets.

For each of the remaining reaction types, we intended to construct $1 \sim 2$ benchmark sets with different target receptors. All experimentally validated binders for each target receptor are collected from BindingDB.[159] For the curation of active sets with a desired electrophile, the collected ligand libraries are filtered using the same SMARTS regular expressions (Table 5.1). We assume these filtered actives with the intended electrophile are TCIs (i.e., these inhibitors form a covalent bond upon binding.) We also perform an additional filtering to only include lead-like molecules ($250 \leq$ molecular weight ≤ 350; LogP ≤ 3.5;

number of rotatable bonds ≤ 7) for the chemical reaction type addition to ketone, addition to nitrile and Michael addition. This allows us to limit the docking library size and avoid potential docking issues from large and flexible compounds.[143]

To construct the decoy set for each target receptor that we choose, we selected physico-chemically similar compounds from the corresponding electrophile subsets. It is understood that some of the compounds in the decoy set might actually be true binders, and our choice of using high physico-chemcially similar compounds make the retrospective virtual screening test even more challenging. A summary of the retrospective virtual screening dataset is listed in Table 5.2.

Table 5.2: Summary of the Retrospective Virtual Screening Dataset.

Receptor name	Number of actives	Number of decoys	PDB	Reaction type
CATK	56	3050	2AUX	Addition to aldehyde
PAPAIN	21	2525	1CVZ	Ring opening
ALDH3A1	36	11678	4L1O	Addition to ketone
CASP3	79	3392	1RHJ	Addition to ketone
EGFR	151	10755	5UG8	Michael addition
JAK3	356	11773	5TTS	Michael addition
CATS	150	18312	1MS6	Addition to nitrile
CATK	215	23596	2F7D	Addition to nitrile

Unfortunately, we did not identify receptor targets with adequate actives (i.e., more than 10 experimentally validated inhibitors) for the chemical reaction disulfide bond formation mainly because only about 2% of TCIs undergo disulfide bond formation with the target receptor. Overall, we construct a retrospective virtual screening benchmark dataset modeling 5 common covalent bond formation reactions of TCIs. To our knowledge, this is the largest benchmark dataset for covalent docking methods.

5.2.2 Rigid CDOCKER Algorithm Overview

There are three main elements to the Rigid CDOCKER algorithm: the receptor and ligand representation, a searching algorithm, and the newly developed scoring function, which includes an additional energy term that approximates the free energy change of covalent bond

formation.

5.2.3 Receptor and Ligand Representation

All protein structure files were acquired from the Protein Data Bank (PDB). Both protein structures and ligand structures are manually examined and reverted to pre-reaction form. The receptor structure is represented implicitly by grids with a grid space of 0.5 Å. Ligand structure files are manually examined and reverted to pre-reaction form. MOE (Molecular Operating Environment)[70] was used to predict the protonation state of the ligands at pH 7.4. The dominant protonation state of the compound is selected for the following docking experiments. RDKit[44] was used to generate random ligand conformations using the EDKTG method[160,161], ParamChem[72,73] was used to prepare the ligand topology and parameter files and the MMTSB tool set[58] was used to cluster the binding poses. Clustering used the tool cluster.pl with K-means clustering. The CHARMM C36 force fields[49] were used and docking was performed in CHARMM[48] with the CHARMM/OpenMM parallel simulated annealing feature.[7] The RMSD cutoff to identify native-like poses is set to be 2 Å for rigid docking, to be consistent with the evaluation criteria in the previous studies.[7,9]

5.2.4 Docking Searching Algorithm

One general problem in docking is how to place the ligand in the vicinity of the binding site and what initial ligand internal conformation to chosen. In the original CDOCKER docking protocol, we typically use 500 docking trials for each ligand.[7] This might be redundant for a rigid compound, while at the same time insufficient for a highly flexible compound. It is endemic to docking methodology that a non-exhaustive searching of ligand conformational space and initial placement will result in a decrease in top ranking accuracy. Hence, this will also affect the performance of docking methods in virtual screening as observed by the Shoichet group in their retrospective virtual screening test when the ligand has many rotatable bonds.[143]

Therefore, after we generate N conformers for a given ligand, we perform 100 initial placements for each conformer (i.e., $100N$ docking trials in total). These initial starting poses are

then optimized by the molecular dynamics (MD) based simulated annealing algorithm[2,3,7,9,27] and scored with the scoring function described below. In the current study, the parameters for the van der Waals and electrostatic interactions are the same as discussed in Chapter 2.[7,27] The CHARMM scripts used for docking covalent inhibitors with Rigid CDOCKER can be acquired through GitHub.

5.2.5 Optimizing the Covalent Docking Scoring Function

The binding free energy can be written as eq 5.2. In cases of pose prediction, the same ligand is docked to the receptor multiple times and generates a distribution of docking poses. Since the initial state ($G_{initial}$) is the same for all docking trials (i.e., both ligand and receptor are presented separately in the solution). Thus, only G_{final} needs to be calculated for pose prediction. The enthalpic contribution (H_{final}) can be separated into ligand internal energy (E_{ligand}), van der Waals interaction (E_{vdw}), electrostatic interactions (E_{elec}) and free energy for the chemical reaction that forms the covalent bond ($E_{covalent}$). The energy terms E_{ligand}, E_{vdw} and E_{elec} have been well-established in the CDOCKER scoring function.[9,27] The entropic contribution (S_{final}) can be separated into contributions from solvation and conformational entropy. Since we consistently dock the same ligand to the same binding pocket in one measurement, the change in conformational entropy for the same ligand is a constant, and we assume the solvation contribution is approximately the same for different docking poses. Thus, we suggest that the entropic contribution and the solvation contribution can be neglected. Therefore, the scoring function for Rigid CDOCKER in covalent docking can be written as eq 5.3.

$$\begin{aligned} \Delta G_{binding} &= G_{final} - G_{inital} \\ &= H_{final} - H_{initial} - T\left(S_{final} - S_{initial}\right) \end{aligned} \tag{5.2}$$

$$\Delta G_{binding} = E_{vdw} + E_{elec} + E_{covalent} + E_{ligand} \tag{5.3}$$

Rigid CDOCKER is a grid based MD docking algorithm.[9] Here, we introduce a customizable grid potential (eq 5.4) to mimic the free energy change for the chemical reaction that forms the covalent bond. We adopt the idea of the *two-point attractor method*[148], where the

ligand is modeled as a free ligand, and this covalent bond grid potential is used to bring together the ligand reactive atom and the targeted receptor reactive atom. As shown in Figure 5.2, the parameters r_1, r_2 and E_{max} determine the width and well depth of the covalent bond grid potential. The variable r is the distance between the grid point and receptor reactive atom. When the ligand reactive atom is close to the receptor reactive atom (i.e., between r_1 and r_2), this potential acts like a covalent bond and lowers the binding free energy ($G_{binding}$).

$$E_{covalent} = \begin{cases} \frac{-4E_{max}}{(r_2-r_1)^2}(r-r_1)(r-r_2) & \text{if } r > r_1 \text{ and } r < r_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

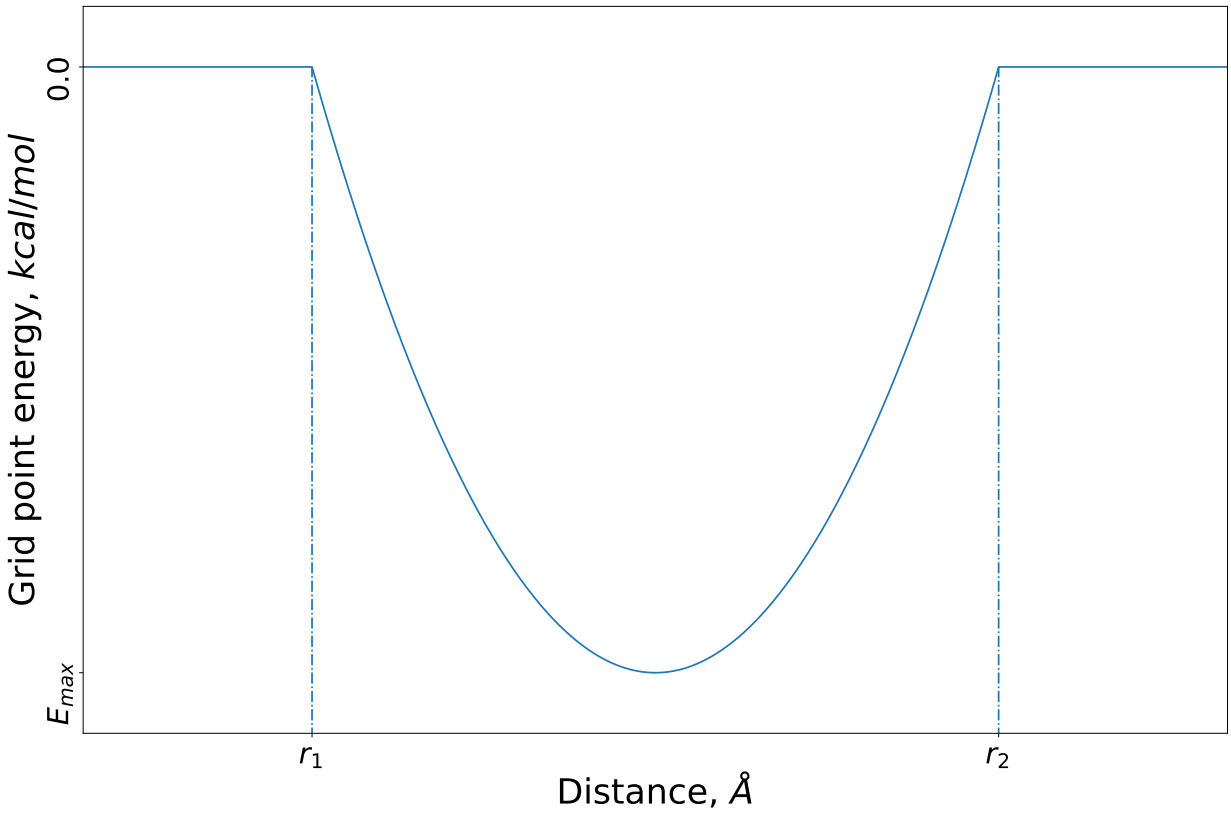

Figure 5.2: Covalent bond grid potential as a function of grid point distance.

For different covalent bond formation reactions, one would expect the covalent bond grid potential parameters should adopt different values. To identify the best parameters for different reaction types, we perform the docking experiments on the pose prediction

dataset using different covalent bond grid potentials. This dataset contains 207 different protein-ligand complexes and is used in previous comparative docking analysis of the covalent docking methods. The different parameter values used in this experiment are listed in Table 5.3. Thus, we have a total of 100 different covalent bond grid potentials (i.e., 100 different combinations of the parameters). Therefore, we perform 100 docking experiments with different covalent bond grid potential against each protein-ligand complex and record the corresponding docking result. This docking experiment is repeated for 3 times.

Table 5.3: Values of Parameter for the Covalent Bond Grid Potential.

Parameter	Value
r_1	$0, 0.5, 1, 1.5$ and 2
r_2	$3, 3.5, 4, 4.5$ and 5
E_{max}	$-2.5, -5, -7.5$ and -10

Because the number of receptor structures in each reaction type is relatively small (i.e., only 5 receptor structures in the chemical reaction disulfide bond formation) and the randomness of dividing a dataset into for training and testing components does not exist[162], we adopt the idea of leave-one-out cross validation and analyze the docking results with the following procedure:

1. For a reaction type with N receptor structures, we separate the dataset into training sets with $N - 1$ receptors and testing sets with one receptor (i.e., leave-one-out approach).

2. Calculate the cumulative ranking accuracy averaged across the 3 independent docking experiments for the training set.

3. Record the corresponding top ranking accuracy and area under curve (AUC) value of the cumulative ranking accuracy plot.

4. Select the covalent bond grid potential parameters with the best top ranking accuracy. If multiple sets of the grid potentials give the same highest top ranking accuracy result, we then select the one with the largest AUC value and record the corresponding average ranking result in the testing set.

5. Repeat step $2 \sim 4$ for all training and testing sets.

Thus, for each reaction type, we will have a set of grid parameters. We consider the most frequent set of grid parameter as the optimized grid parameters for this reaction type and compute the corresponding top ranking accuracy (Table 5.4). A complete docking result is listed in the supplementary files rank-result.tsv and can be acquired through GitHub.

Table 5.4: Covalent Bond Grid Potential for Different Reaction Types

Reaction Type	r_1, Å	r_2, Å	E_{max}, $kcal/mol$
Addition to aldehyde	1.5	4.5	-10
Disulfide bond formation	0	3.5	-10
Addition to ketone	0	5	-10
Michael addition	0	4.5	-10
Addition to nitrile	1.5	4	-10
Nucleophilic substitution	1	5	-7.5
Ring opening	1	5	-10

Table 5.5: Top Ranking Accuracy for Covalent Docking and Non-covalent Docking in Rigid CDOCKER

Reaction Type	Non-covalent docking	Covalent docking
Addition to aldehyde	18.06%	58.33%
Disulfide bond formation	40.00%	53.33%
Addition to ketone	8.33%	55.56%
Michael addition	23.05%	53.90%
Addition to nitrile	51.06%	66.67%
Nucleophilic substitution	23.53%	56.86%
Ring opening	20.83%	45.83%

As shown in Table 5.4, for each of the chemical reactions, we identified the best covalent bond grid potential, and the top ranking accuracy for each of the chemical reactions is listed in Table 5.5. The standard Rigid CDOCKER non-covalent docking methodology is used for direct comparison. We demonstrate that the additional covalent energy term ($E_{covalent}$) significantly improves the docking performance. The top ranking accuracy against the pose prediction dataset is comparable with other covalent docking methods used in the same

benchmark dataset.[154] The average docking runtime is about 15 minutes, which is comparable or faster than other covalent docking methods.[147–152] We notice that we have a relative lower pose prediction accuracy for the reaction type ring opening. On the other hand, in the pose prediction challenge for the reaction type ring opening, all of the other five covalent docking methods have a top rank accuracy below 25% (Table 5.6). This indicates there might be potential issues in dataset construction or modeling ring opening in general for tethered docking methods.

Table 5.6: Top Ranking Accuracy for the Chemical Reaction Ring Opening with Differenct Covalent Docking Method

Covalent Docking Methods	Top ranking accuracy
Covalent docking in CDOCKER	45.83%
AutoDock4[a]	25.00%
CovDock[a]	12.50%
GOLD[a]	25.00%
ICM-Pro[a]	12.50%
MOE[a]	25.00%

[a] Top ranking accuracy reported in the previous pose prediction challenge.[154]

5.2.6 Augmented Scoring Function for Virtual Screening

One of the most common applications of docking is to identify novel inhibitors for a given target. Covalent docking methods have been used to rank compounds with the same warhead chemistry and succeed in identifying covalent inhibitors.[143,163–165] The proposed scoring function estimates the total energy of the protein-ligand complex upon binding. Therefore, in order to compare different small molecules, we need to augment our scoring function to consider the system in the unbound state. Because these compounds are docked to the same protein target, the protein energy in the unbound state is a constant and cancels out in comparative studies. Thus, to complete the scoring function for this situation, we only need to include contributions to the ligand internal energy and conformational entropy of

the unbound state (eq 5.5).

$$\begin{aligned}\Delta G_{binding} &= G_{final} - G_{inital} \\ &= E_{Scoring\ function} - E_{ligand\ internal\ energy\ at\ unbound\ state} + TS_{ligand} \\ &= E_{vdw} + E_{elec} + E_{covalent} + E_{ligand\ internal\ energy\ in\ the\ bound\ state} \\ &- E_{ligand\ internal\ energy\ in\ the\ unbound\ state} + TS_{ligand}\end{aligned} \tag{5.5}$$

The top rank docked pose is used to calculate the total energy of the protein-ligand complex using the scoring function just proposed in (eq 5.5). The conformers for each of the ligands used for docking are minimized in vacuum. The ligand internal energy in the unbound state is then calculated by computing the ensemble average of the internal energy of these conformers. The ligand conformational entropy (S_{ligand}) is calculated based on the number of the rotatable bonds of the ligand using the microscopic definition of entropy (eq 5.6). We assume that the rotatable bonds of the ligand are independent of each other and all three states (i.e., trans, gauche- and gauche+) can be equally sampled. The temperature (T) is set to be room temperature (298 K).

$$S_{ligand} = -k_B N_{rotors} ln \frac{1}{3} \tag{5.6}$$

The solvation free energy difference is computed using two different approaches: (1) implicitly represented in the proposed scoring function by the distance dielectric constant of $3r$.[7,9,27] and (2) rescoring the system using the FACTS implicit solvent model.[40] In the rescoring approach with the FACTS implicit solvent model, the ligand internal energy in the unbound state is calculated by the same approach. The rescoring of the protein-ligand complex at the bound state is performed by minimizing the top rank docked pose with the FACTS implicit model. The coordinates of the protein atoms and the ligand reactive atoms are fixed for two reasons: (1) reduce the computational cost of the rescoring, and (2) the distance between the protein and ligand reactive atoms remains unchanged. Therefore, we do not need to re-estimate the covalent bond formation energy ($E_{covalent}$).

As shown in Figure 5.3, the dominant effects in the ranking orders is the van der Waals and electrostatic energy differences, which reflects the structure complimentary between the

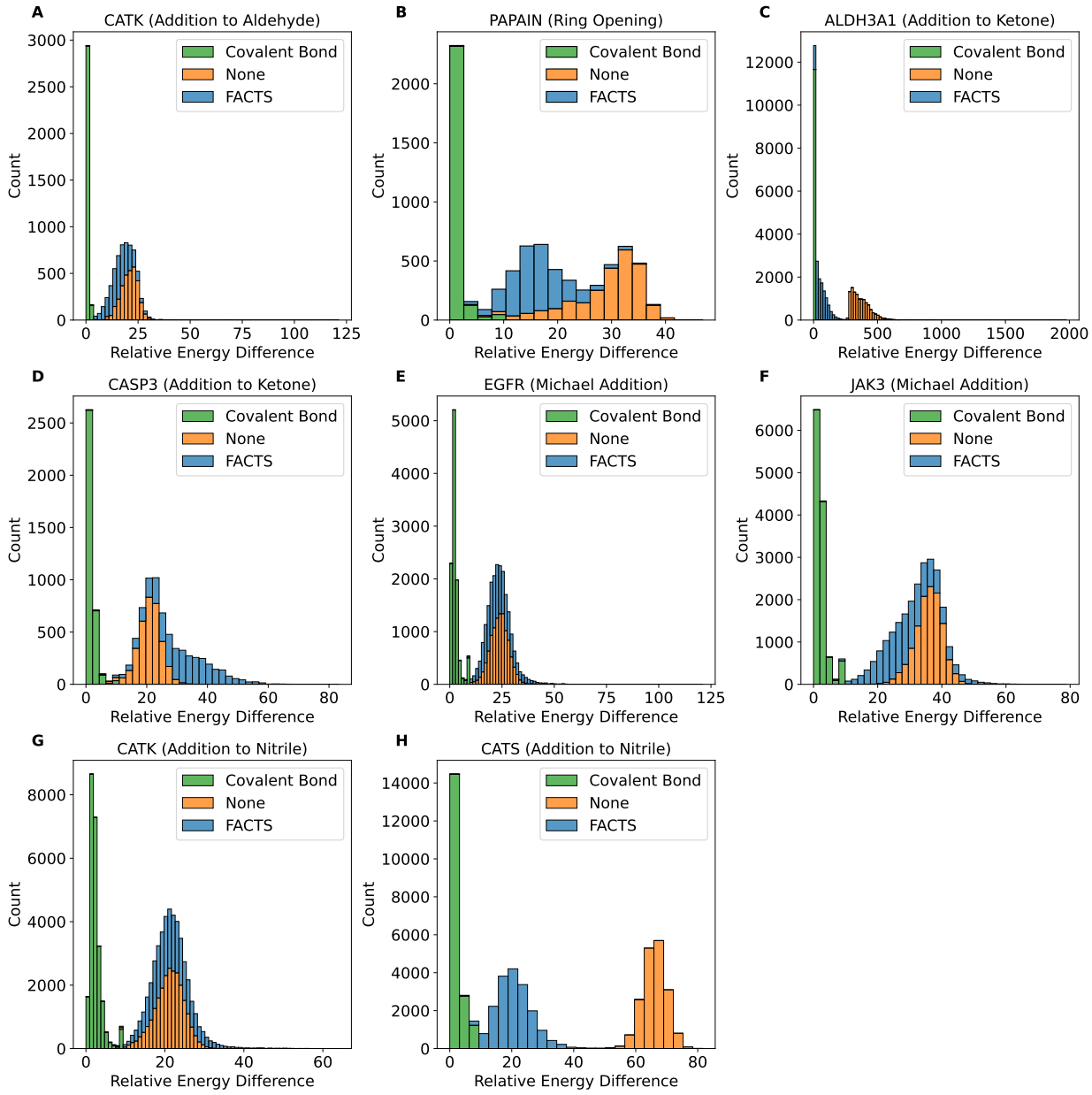

Figure 5.3: The distribution of the relative energy difference observed in the retrospective virtual screening for the receptor targets and the corresponding reaction type (A) CATK (addition to aldehyde), (B) PAPAIN (ring opening), (C) ALDH3A1 (addition to ketone), (D) CASP3 (addition to ketone), (E) EGFR (Michael addition), (F) JAK3 (Michael addition), (G) CATK (addition to nitrile) and (H) CATS (addition to nitrile). The energy differences plotted here are covalent bond formation energy $E_{covalent}$ (green), change of the system energy upon binding after subtracting the covalent bond formation energy (i.e., $\Delta G_{binding} - E_{covalent}$) in vacuum (orange) or using FACTS implicit solvent model (blue). The minimum energy for all comparison is set to be zero.

receptor and ligand. The difference of the covalent bond formation energy ($E_{covalent}$) is relatively small for different compounds. This reflects the likelihood of forming a covalent bond between the corresponding ligands and receptor and shifts the ranking orders for compounds with similar structural and chemical properties.

5.3 Results

5.3.1 Virtual Screening Performance with Generic Parameters

As mentioned above, we constructed a retrospective virtual screening dataset containing 7 receptor targets modeling 5 common TCI warheads. We perform covalent docking experiments with the docking methods just described against both ligands and non-binding decoys. We use the generic covalent bond grid potential parameters, which are the ones optimized in the pose prediction just described (Table 5.4). The area under the curve (AUC) value of the receiver operating characteristic (ROC) curve (Figure 5.4) and two enrichment factors (EF_1 and EF_{20}) are used to evaluate its performance in distinguishing the non-binders from binders (Table 5.7).

Table 5.7: Summary of the Retrospective Virtual Screening Performance.

Receptor	AUC[a]	AUC[b]	EF_1[a]	EF_1[b]	EF_{20}[a]	EF_{20}[b]	Reaction type
CATK	0.909	0.694	16.071	7.143	4.375	2.232	Addition to aldehyde
PAPAIN	0.623	0.731	4.762	0.0	2.143	2.143	Ring opening
ALDH3A1	0.8	0.791	2.778	2.778	2.917	3.056	Addition to ketone
CASP3	0.492	0.593	0.0	0.0	0.759	1.013	Addition to ketone
EGFR	0.67	0.63	11.258	0.662	2.285	1.556	Michael addition
JAK3	0.799	0.744	28.324	2.601	3.165	2.471	Michael addition
CATK	0.741	0.635	18.605	1.86	2.674	1.628	Addition to nitrile
CATS	0.704	0.609	22.667	3.333	2.533	1.9	Addition to nitrile

[a] Binding free energy is estimated after rescoring with the FACTS implicit solvent model.
[b] Binding free energy is estimated using the proposed scoring function with the distance dielectric constant of $3r$.

As demonstrated in Figure 5.4 and Table 5.7, the proposed covalent docking method with both solvation models has the ability to distinguish binders from non-binders in general.

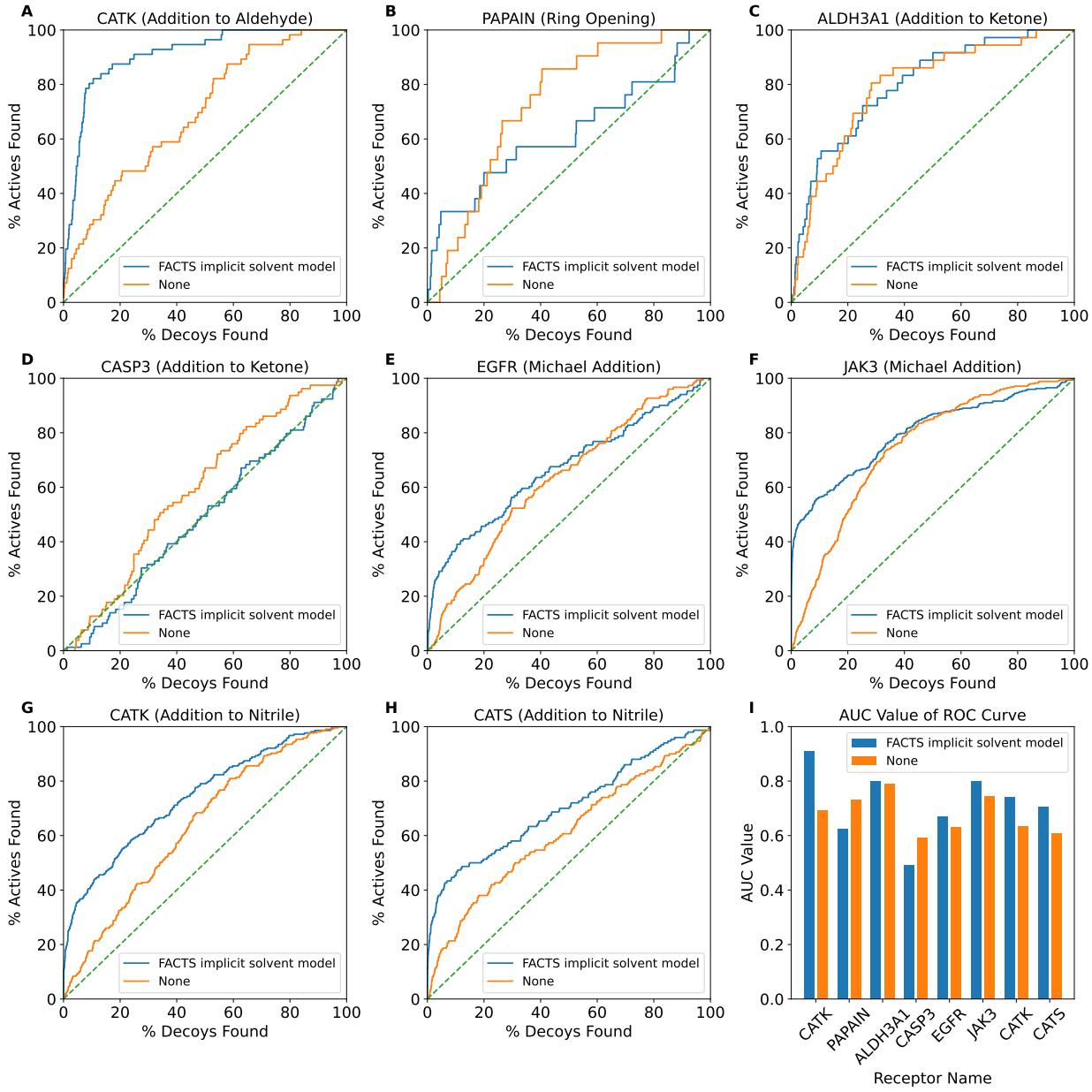

Figure 5.4: The receiver operating characteristic (ROC) curves of the receptor targets and the corresponding reaction type (A) CATK (addition to aldehyde), (B) PAPAIN (ring opening), (C) ALDH3A1 (addition to ketone), (D) CASP3 (addition to ketone), (E) EGFR (Michael addition), (F) JAK3 (Michael addition), (G) CATK (addition to nitrile) and (H) CATS (addition to nitrile). (I) Summary of the AUC value for each of the ROC curve.

The best performance is for the target receptor CATK with an AUC value of 0.909 and the warhead chemistry of the covalent inhibitors is addition to the aldehyde functionality. As we noted earlier, some of the compounds in the decoy set might be true binders and this could

skew our success measures. Thus, the reported values (AUC, EF_1 and EF_{20}) are actually a lower bound. Using the FACTS implicit solvent model shows improved performance, especially in the early enrichment of binders (i.e., EF_1 value). The largest improvement of the EF_1 value is against the receptor target JAK3 (28.324 vs. 2.601). The computational cost of rescoring with the FACTS implicit solvent model is about $5 \sim 10\%$ of the average runtime of the proposed docking algorithm.

For the warhead chemistry of addition to ketone, as shown in Figures 5.5A and 5.5B, the properties of the compounds (i.e., number of rotatable bonds and molecular weight) are similar to each other. However, the binding pocket for the receptor target ALDH3A1 is small and buried (Figure 5.5C) and the binding pocket for the receptor target CASP3 is large and open (Figure 5.5D). Thus, the searching space in the case of CASP3 is relatively larger and the surface binding pocket requires less structural complimentary. This may be why we observe difference virtual screening performance for the warhead chemistry of addition to ketone and consistent with the observation that structural complimentary is important in docking.

5.4 Conclusions and Discussions

Targeted covalent inhibitors (TCIs) are designed such that the covalent warheads can target rare, non-conserved residues of a particular target protein and lead to the development of highly selective inhibitors with high potency and extended duration of action. In this work, we introduced a customizable covalent bond grid potential in the original Rigid CDOCKER scoring function.

In the covalent docking in Rigid CDOCKER, the covalent bond grid potential acts as an attractor if the ligand reactive atom and the protein reactive atom are close to each other and provides an estimate of the free energy change upon covalent bond formation. This is clearly evident by our comparison of the rigid docking protocol with and without the covalent bond grid potential shown in Table 5.5. Different bond formation reactions (warhead chemistries) should be modeled differently. We optimized and provided a set of generic parameters for the covalent bond grid potential for different reaction types. Our

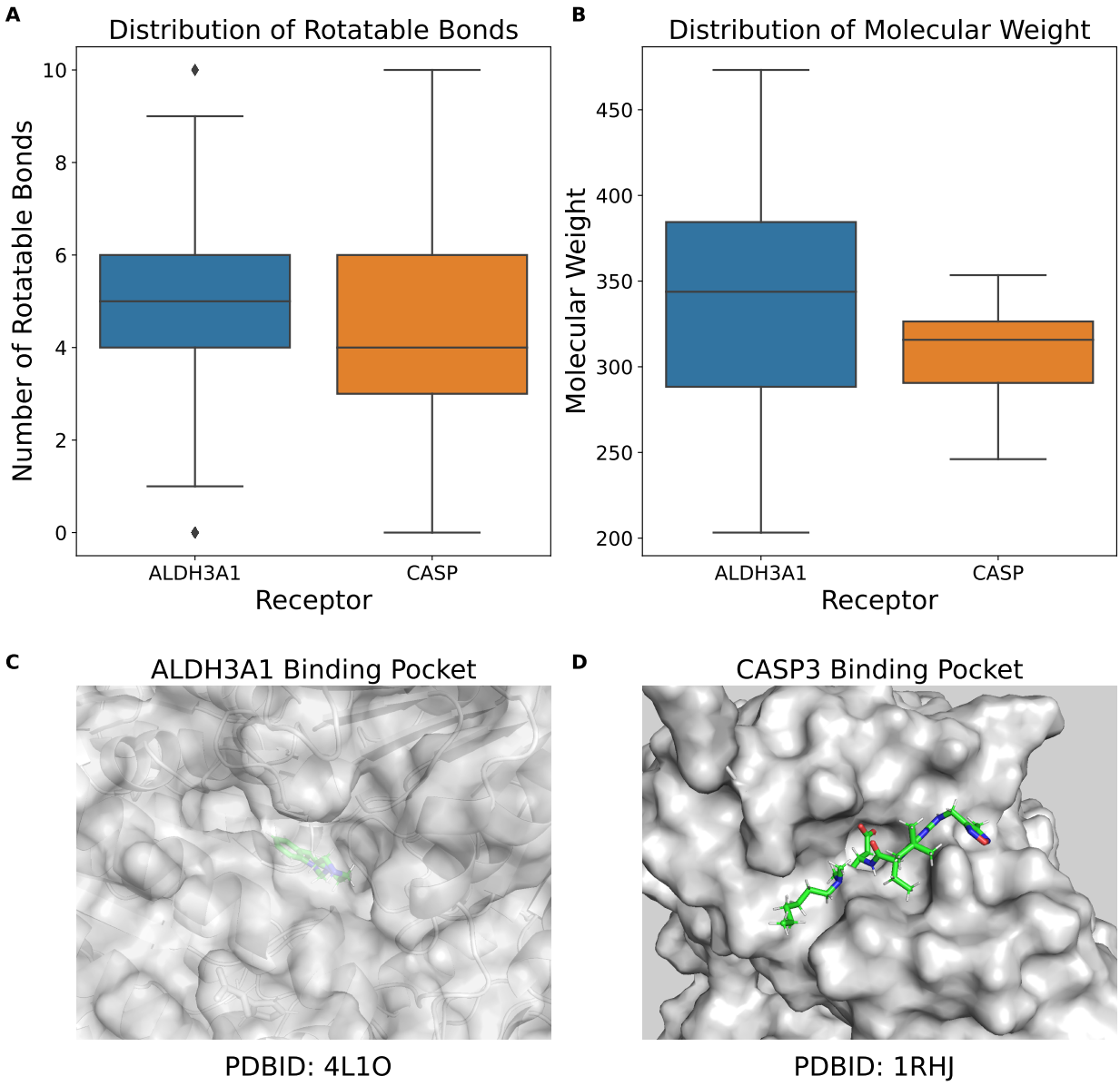

Figure 5.5: Retrospective virtual screening experiments for the warhead chemistry of addition to ketone. (A) ALDH3A1 binding pocket (PDBID: 4L1O), (B) CASP3 binding pocket (PDBID: 1RHJ), Distribution of the compound property: (C) rotatable bonds, (D) molecular weight.

covalent docking algorithm shows comparable pose prediction accuracy with other popular docking algorithms. The average docking runtime is about 15 minutes, which is comparable or faster than other covalent docking methods.

We also constructed a benchmark dataset for evaluating the ability to identify novel

TCIs for tethered docking methods. To our knowledge, this is the largest retrospective virtual screening benchmark set for evaluating tethered docking methods. We demonstrate that the proposed covalent docking algorithm has the ability to discriminate binders from non-binders with both solvent models. The covalent bond grid potential in this test uses the generic parameters. This suggests that the proposed covalent docking algorithm can be widely used for different targets for lead compound identification. Using the FACTS implicit solvent model with a small additional computational cost shows better performance, especially in the early enrichment of the active compounds.

In real applications, one could further optimize or adjust this covalent bond grid potential to achieve further improvements. Recently, we developed python package of the CDOCKER family as a workflow functionality in pyCHARMM (i.e., CHARMM through a python interface), which reduces the complexity in using CDOCKER for potential users unfamiliar with CHARMM or CDOCKER. This allows one to easily modify the grid parameters and integrate CDOCKER based docking workflows with other commonly used python packages. Finally, we suggest that the proposed docking algorithm with the FACTS implicit solvent model can effectively be applied in the real-world applications of identifying novel TCIs. The benchmark dataset and code examples are provided on Github https://github.com/wyujin/Covalent-Docking-in-CDOCKER.

Chapter 6

pyCHARMM CDOCKER – Automatic Docking in Python

Manuscript in preparation

6.1 Introduction

The binding of small molecule ligands to protein or nucleic acid targets is important to numerous biological processes. Accurate prediction of the binding modes between a ligand and a macromolecule is of fundamental importance in structure-based structure-function exploration. Docking predicts the orientation and conformation of a small molecule (ligand) in the binding site of the target protein and estimates its binding affinity. In Chapter 2, 3 and 5, we demonstrated the newly developed CDOCKER methodology has better or comparable performance in pose prediction accuracy and retrospective virtual screening experiments with less computational cost. In Chapter 4, we also demonstrated that the CDOCKER methodology can be applied to identify novel compounds in an application to TMPRSS2, a target associated with SARS-CoV-2 viral entry.

To perform a successful CDOCKER docking experiment, additional cheminformatics tools described in previous chapters are necessary. Python[166,167] is one of the most popular and standard scripting languages in cheminformatics. Today, multiple off-the-shelf Python language based cheminformatics toolkits are available for use, such as OpenMM[41,42], Pybel[43],

RDKit[44] and Scikit-learn.[45] Leveraging the popularity and utility of the Python language, we introduced the Python functionality of CHARMM (i.e., pyCHARMM). pyCHARMM provides direct access from Python to the Fortran subroutines and functions in the CHARMM API (application programming interface).

Here, we will focus on the development and performance of pyCHARMM CDOCKER (i.e., Python functionality of the CDOCKER family). This will further accelerate CDOCKER docking algorithms, simplify the integration between CDOCKER methodology and other cheminformatics tools, and provide the standard CDOCKER docking protocols for potential users who have little knowledge of docking or the CDOCKER family.

6.2 Methods

6.2.1 General Receptor and Ligand Representation

Open Babel[71] was used to generate random ligand conformations, ParamChem[72,73] was used to prepare the ligand topology and parameter files and the MMTSB tool set[58] was used to cluster the binding poses. The CHARMM C36 force fields[49] were used and docking was performed in pyCHARMM with the CHARMM/OpenMM parallel simulated annealing feature.[7]

CDOCKER is a grid-based docking method where the ligand-receptor interaction energy is precomputed and stored on a grid.[7,9,27] Thus, the receptor Protein Data Bank (PDB) file and protein structure (PSF) file are required for generating grids. The ligand PDB file and the corresponding topology and parameter file (*ligandrtf*) are also required to perform a docking experiment. We suggest using the default file names and values listed in Table 6.1 so that one does not need to specify the corresponding arguments in the Python script.

The other important component in grid generation is the center and the size of the grid box, which needs to be determined each time by the user. The center of the grid box is specified by the cartesian coordinates with the Python arguments *xcen, ycen* and *zcen*. The size of the grid box is specified by the Python argument *maxlen*.

Table 6.1: General Parameters for the pyCHARMM CDOCKER

Parameter	Default value	Meaning
receptorPDB	*protein.pdb*	receptor PDB file
receptorPSF	*protein.psf*	receptor PSF file
ligPDB	*ligand.pdb*	ligand PDB file
ligSeg	*LIGA*	segment ID in the ligand PDB file

6.2.2 Standard pyCHARMM CDOCKER Rigid Docking Experiment

The standard pyCHARMM CDOCKER rigid docking (i.e., pyCHARMM Rigid CDOCKER) requires a precomputed ligand conformer library. The path to this conformer library is specified by the argument *confDir* with a default value of *'conformer/'*. Thus, if one saves the precomputed ligand conformations under the folder *'conformer'*, then one does not need to specify this argument again in the Python script. For each conformer, pyCHARMM Rigid CDOCKER performs 100 docking trials by default. The following script demonstrates a standard and basic pyCHARMM CDOCKER rigid docking experiment.

```
## Import necessary pyCHARMM libraries
import pycharmm
import pycharmm.lib as lib
import pycharmm.read as read
import pycharmm.lingo as lingo
import pycharmm.settings as settings

## Import standard pyCHARMM Rigid CDOCKER function (Rigid_CDOCKER)
from pycharmm.cdocker import Rigid_CDOCKER

## Topology and parameter files
dataDir = "../toppar/"
settings.set_bomb_level(-1)
```

```
read.rtf(dataDir + 'top_all36_prot.rtf')
read.rtf(dataDir + 'top_all36_cgenff.rtf', append = True)
read.prm(dataDir + 'par_all36m_prot.prm', flex = True)
read.prm(dataDir + 'par_all36_cgenff.prm', append = True, flex = True)
settings.set_bomb_level(0)
lingo.charmm_script('stream "./ligandrtf"')

## Rigid CDOCKER standard docking protocol
sortedResult, dockResult = Rigid_CDOCKER(xcen = 0, ycen = 0,
                                         zcen = 0, maxlen = 10)
```

6.2.3 Standard pyCHARMM CDOCKER Covalent Docking Experiment

The major difference between the covalent docking and non-covalent docking in the pyCHARMM Rigid CDOCKER is that the user needs to specify the well-depth and cutoffs of the covalent bond grid potential (i.e., Figure 5.2 and Table 5.3). The corresponding arguments and default values are listed in Table 6.2.

Table 6.2: Parameters for the pyCHARMM Covalent Docking

Parameter	Default value	Meaning
rcta	*0*	customizable grid left cutoff (i.e., r_1)
rctb	*0*	customizable grid right cutoff (i.e., r_2)
hmax	*0*	customizable grid well-depth (i.e., E_{max})
flag_use_hbond	*False*	whether or not the customizable bond grid potential will be used in Rigid CDOCKER

The user also needs to specify the set of the receptor reactive atoms (donor/nucleophile) and the set of the ligand reactive atoms (acceptor/electrophile) in the receptor and ligand parameter files. The following script demonstrates a standard and basic pyCHARMM CDOCKER covalent docking experiment using the pyCHARMM function *Rigid_CDOCKER*.

```
## Rigid CDOCKER standard covalent docking protocol
```

```
clusterResult, dockResult = Rigid_CDOCKER(xcen = 0, ycen = 0, zcen = 0,
                            maxlen = 10, flag_use_hbond = True,
                            hmax = -10, rcta = 0, rctb = 5)
```

6.2.4 Standard pyCHARMM CDOCKER Flexible Docking Experiment

Flexible CDOCKER allows the ligand and receptor flexible side chains configurations to explore their conformational space simultaneously. Therefore, the user needs to specify which amino acids are considered flexible by the argument *flexchain* prior to performing the flexible docking experiment. By default, pyCHARMM CDOCKER flexible docking (i.e., pyCHARMM Flexible CDOCKER) utilizes Open Babel to generate 20 random ligand conformations and performs 25 docking trials (i.e., 500 docking trials per docking experiment by default). The flexible docking experiment uses 2 generations in the genetic algorithm as we discussed in Chapter 3 and can be changed by the argument *generation*. The following script demonstrates a standard and basic pyCHARMM CDOCKER flexible docking experiment.

```
## Import necessary pyCHARMM libraries
import pandas as pd
import pycharmm
import pycharmm.lib as lib
import pycharmm.read as read
import pycharmm.lingo as lingo
import pycharmm.settings as settings

## Import standard pyCHARMM Flexible CDOCKER function (Flexible_CDOCKER)
from pycharmm.cdocker import Flexible_CDOCKER

## Topology and parameter files
dataDir = "../toppar/"
settings.set_bomb_level(-1)
```

```
read.rtf(dataDir + 'top_all36_prot.rtf')
read.rtf(dataDir + 'top_all36_cgenff.rtf', append = True)
read.prm(dataDir + 'par_all36m_prot.prm', flex = True)
read.prm(dataDir + 'par_all36_cgenff.prm', append = True, flex = True)
settings.set_bomb_level(0)
lingo.charmm_script('stream "./ligandrtf"')

## Read in the receptor flexible side chain selection look up table
flexchain = pd.read_csv('flexchain.csv', sep = '\t', index_col = 0)

## Flexible CDOCKER standard docking protocol
clusterResult, dockResult = Flexible_CDOCKER(xcen = 0, ycen = 0, zcen = 0,
                            maxlen = 10, flexchain = flexchain)
```

6.2.5 pyCHARMM CDOCKER Output

pyCHARMM CDOCKER will create a new folder containing the docking results (i.e., docked poses and the corresponding docking score). Two tab-separated values (TSV) files, *dockResult.tsv* and *clusterResult.csv*, will always be saved. These two files include docking scores for all docked poses and cluster representatives respectively. Table 6.3 demonstrates the options and the corresponding explanations.

6.3 Results

6.3.1 Improved Performance in Pose Prediction Experiments

High pose prediction accuracy should be the fundamental of any docking application. We first compare the top ranking accuracy of the pyCHARMM CDOCKER with our previous study (Chapters 2 and 3).

The major difference is that we used a more extensive searching in the pyCHARMM Rigid CDOCKER (i.e., more docking trials) within the same computational cost (i.e., GPU and

Table 6.3: Parameters for the pyCHARMM CDOCKER Output Options

Parameter	Default Value	Meaning
saveDir	*'dockresult/'*	The folder contains the docking results.
flag_save_all	*Ture*	save all docked pose
flag_save_cluster	*Ture*	save cluster representative
flag_delete_grid	*True*	delete grids after docking
flag_save_top[a]	*True*	save top 10 lowest energy poses
flag_save_placement[b]	*True*	save initial placement
flag_save_crossover[b, c]	*True*	save the $(N-1)^{th}$ intermediate generation

[a] Only present in the *Rigid_CDOCKER*. [b] Only present in the *Flexible_CDOCKER*. [c] Only meaningful when the number of generations is greater than 1.

Table 6.4: Top Ranking Accuracy of Multiple Rigid Receptor-Flexible Ligand Docking Programs on the SB2012 Set.

Docking method	Top ranking accuracy
pyCHARMM Rigid CDOCKER[a]	65.3%
Rigid CDOCKER[b]	42.9%
Autodock Vina[b]	53.2%
DOCK v6.7[b]	55.3%

[a] Docking experiment is repeated for three times. [b] We used the top-ranking accuracy reported in our previous study for direct comparison.[7]

Table 6.5: Top Ranking Accuracy in Cross-Docking Experiment.

Receptor Name	pyCHARMM Flexible CDOCKER[a]	Flexible CDOCKER[b]
T4 L99A	71.34%	66.21%
T4 L99A/M102Q	79.52%	77.62%
Thrombin	41.21%	40.11%
DHFR	49.09%	50.91%
PDE10A	43.66%	36.84%

[a] Docking experiment is repeated for three times. [b] We used the top-ranking accuracy reported in our previous study for direct comparison.[12]

CPU runtime). Therefore, as shown in Table 6.4, pyCHARMM Rigid CDOCKER has the best performance. The largest improvement in the top ranking accuracy is 22.4% compared with Rigid CDOCKER. We also performed the same cross-docking experiments described

in Chapter 3 with pyCHARMM Flexible CDOCKER. As shown in Table 6.5, we observed an improved top ranking accuracy. Overall, the Python interface reduces the I/O process and allows more extensively sampling, resulting in improved pose prediction accuracy for the pyCHARMM CDOCKER.

6.3.2 Reduced Computational Cost in Flexible Docking

Flexible receptor docking methods have shown improved performance in pose prediction and retrospective virtual screening experiments.[12–14] However, it has been less adopted because of the relatively large computational cost. Because the Python interface reduces the I/O process, we want to investigate its affects on computational cost for flexible receptor docking (i.e., pyCHARMM Flexible CDOCKER). Therefore, we compared the computational cost with our previous study using the same docking systems.[12] We also plotted the average docking runtime in the cross-docking experiments as a function of total number of system rotatable bounds in Figure 6.1.

Table 6.6: Average Docking Runtime for with 10 Flexible Side Chains.

Docking method	Runtime
pyCHARMM Flexible CDOCKER	40 minutes for 500 docking trials
Flexible CDOCKER[12]	100 minutes for 500 docking trials
AutodockFR[13]	365 hours for 50 generations
Glide[14]	400 CPU hours and 50 GPU hours for 20 docking trials

As shown in Table 6.6, pyCHARMM Flexible CDOCKER further reduce the computational cost by at least 2-fold. As demonstrated in Figure 6.1, the average computational cost is proportional to the number of rotatable bonds, which is not a surprising results. This also provides an estimation of the computational cost for a given system.

6.4 Conclusions and Discussions

pyCHARMM CDOCKER provides a high-level Python interface to the widely used CHARMM CDOCKER functionality. One of the aims of pyCHARMM CDOCKER is to provide simple

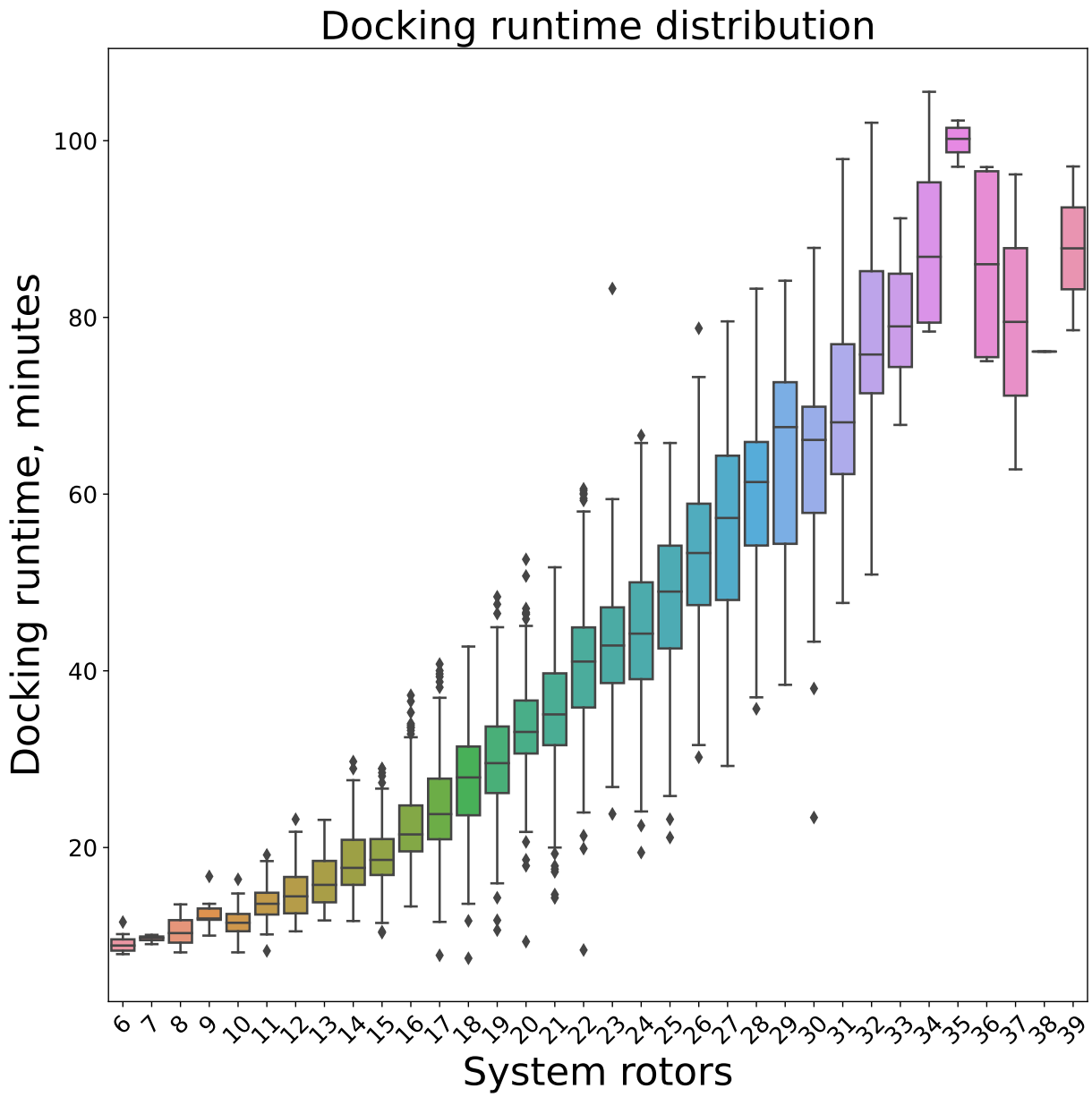

Figure 6.1: Average docking runtime as a function of total number of rotatable bonds in the system.

one-liner code to perform standard CDOCKER docking experiments. We demonstrate significantly improved pose prediction accuracy for rigid receptor docking. We also note that this feature significantly reduces the computational cost in the flexible receptor docking. With this acceleration, to our knowledge, pyCHARMM Flexible CDOCKER is the fastest flexible receptor docking method. The Python interface also allows pyCHARMM CDOCKER more

easily to integrate with other commonly used cheminformatics toolkits.

Chapter 7

Discussions and Conclusions

The purpose of docking is to accurately predict the binding poses of a ligand within the constraints of a receptor binding site and to correctly estimate the strength of binding.[2–4] The most common application of docking is to identify novel compounds among a large compound library, and has been a focus of development in the field for decades.[17,21,141] The purpose of this chapter is to summarize the advances in the CDOCKER and general docking methods and discuss the future work of docking methods. The corresponding CDOCKER examples and workshop can be found through Github https://github.com/wyujin/Covalent-Docking-in-CDOCKER and https://github.com/clbrooksiii/pyCHARMM-Workshop.

When ligands with different sizes bind to the same binding pocket, it is reasonable to assume the binding pocket undergoes conformational changes, and this has been captured experimentally.[31–34] However, flexible receptor docking methods have been less adopted because of the relatively large computational cost. In this thesis, I began with the methodology development of the flexible receptor docking method (Flexible CDOCKER) in Chapter 2 and 3. The implementation of parallel MD based simulated annealing running on a GPU is 20 times faster than the original simulated annealing running on a CPU when 500 docking trials are used.[7]. I then describe a novel hybrid searching algorithm that combines both MD based simulated annealing and genetic algorithm crossovers to address the enhanced sampling of the increased search space.[12] I proposed a new physics-based scoring function that includes both enthalpic and entropic contributions upon binding by considering the conformational variability of the flexible side chains within the ensemble of docked poses.[12] These advance

the field of physics-based scoring functions in flexible receptor docking and broaden the scope of flexible receptor docking methods in high-throughput docking campaigns.

With the implementation of the proposed searching algorithm and scoring function, Flexible CDOCKER demonstrates significant improvement in pose prediction and in distinguishing binders from non-binders. To our knowledge, Flexible CDOCKER is the fastest flexible receptor docking method and is sufficiently fast to be utilized in high-throughput docking screens in the context of hierarchical approaches. In Chapter 4, as a practical example, I worked with a team of experimental colleagues to identify potential therapeutics for the host transmembrane serine protease TMPRSS2, a promising antiviral target that plays a direct role in SARS-CoV-2.[22] We designed a hierarchical workflow that used pharmacophore similarity to filter compound library followed by the direct application of the Flexible CDOCKER and FACTS implicit solvent scoring methodology. A total of 134,109 molecules were collected from multiple databases and subjected to the hierarchical docking experiment, which led to the identification of three new inhibitors. This hierarchical workflow takes the advantage of the flexible receptor docking method in high-throughput virtual screening for lead compound identification while reducing the overall computational cost. On the other hand, there were no specific changes of the docking protocols in the proposed workflow. Therefore, the proposed hierarchical workflow could be applied to other biological targets without any further modifications.

During the TMPRSS2 project, we realized a set of known compounds form a covalent bond with the active site serine of serine proteases.[35,36] On the other hand, TCIs are considered to be an important component in the toolbox of drug discovery and about 30% of currently marketed drugs are TCIs.[37–39] This motivated me to develop a covalent docking protocol in Rigid CDOCKER by introducing a customizable covalent bond grid potential. This covalent bond grid potential acts as an attractor if the ligand reactive atom is close to the receptor reactive atom and provides an estimate of the free energy change upon covalent bond formation. Different bond formation reactions (warhead chemistries) are modelled differently. We optimized and provided a set of generic parameters for the common covalent bond formation reactions. We demonstrate higher pose prediction accuracy and the ability to identify novel compounds among a large chemical space. In the real applications related

to the TCIs, one typically would have additional information about the binding pockets (i.e., where the covalent bond forms). It is reasonable to augment the scoring function to perform a "biased" docking experiment. One of the aims of the customizable grids is to allow one to further optimize or adjust this covalent bond grid potential to achieve improvements.

With the implementation of new features as described above, CDOCKER shows comparable or better performance in the field of rigid receptor docking and flexible receptor docking.[7,12] However, CDOCKER is less adopted because of the relatively complicated scripting languages. This motivated me to develop pyCHARMM CDOCKER. The main purpose of pyCHARMM CDOCKER is to provide simple one-liner code to perform standard CDOCKER docking experiments. The realization of pyCHARMM CDOCKER also demonstrates improved pose prediction accuracy and lower computational cost. The Python interface also allows pyCHARMM CDOCKER more easily to integrate with other commonly used cheminformatics toolkits.

We have shown the advances in the CDOCKER docking methodologies and its potential in drug discovery. One remaining challenge is to screen ultra-large compound libraries within a reasonable timeframe. As the compound libraries have grown, the ability to screen hundreds of millions, and even billions, of compounds is important for emerging docking methods. Future development of the CDOCKER family could focus on incorporating FFT docking[7] to the hierarchical workflow.

APPENDIX

Supporting Information: TMPRSS2 Inhibitor Discovery Facilitated Through an in Silico and Biochemical Screening Platform

Methods

Below is the supporting information of the wet lab experiment setups of the TMPRSS2 protein expression and measurement of the IC_{50} of the novel inhibitors.

Vector design. Plasmids were constructed by Twist Bioscience by inserting the gene for protease domain of TMPRSS2, specifically amino acids 247-492, into the pET28a+ vector using the NdeI_XhoI restriction enzyme cut-sites.

Protein expression. The pET28a+ plasmids containing the TMPRSS2 genes were transformed into BL21(DE3) and plated on LB agar with kanamycin. The bacteria were grown in small 5 mL LB (+ kanamycin) cultures overnight at 37 $\circ C$. The 5 mL starters were used to inoculate 1 L LB (+ kanamycin) cultures, which were grown to $OD = 0.8$ at 37 $\circ C$ with shaking at 250 rpm. Expression was induced using 1 mM Isopropyl β-d-1-thiogalactopyranoside (IPTG), which we let grow for 5 hours. The cells were then spun down at 9,500 x g for 15 min. The pellets were collected, flash frozen and stored at -80 $\circ C$.

Chemical lysis and denaturing. Before lysing, the cell pellet was first fully thawed until it reached room temperature. Chemical lysis was performed by resuspending the pellet using B-PER reagent (Fisher, PI78243) with lysozyme (Fisher, 90082) and DNase I (Fisher, 90083) following manufacturer's protocols. The cells were then spun at 15,000 x g for 5

minutes, and the cell lysate was collected and saved. The insoluble portion, which contained inclusion bodies of TMPRSS2, was resuspended / washed using lysis buffer containing detergent (50 mM Tris HCl, 0.9% NaCl, 1% Triton X-100, pH 7.5) and then spun at 15,000 x g for 5 minutes. After removing the supernatant, the pellet was washed once more with a lysis buffer that did not contain detergent (50 mM Tris HCl, 0.9% NaCl, pH 7.5) and then spun at 15,000 x g for 5 minutes.

The pellet was resuspended and the inclusion bodies were denatured by adding 20 mL denaturing buffer (8 M urea, 10 mM Tris, 100 mM sodium phosphate, pH 8.0) plus reducing agent (1:1000 BME). Denaturing occurred at room temperature with rotation for at least 30 minutes. The concentration of protein was determined via nanodrop, and additional denaturing buffer was added to reduce the concentration to below 1 mg/mL. Denaturing occurred at room temperature on a rotator for at least 30 minutes before being spun down and decanted (20,000 x g, 15 minutes).

Batch binding. Ni-NTA agarose (Qiagen, 30210) was prepared by washing 3 times with binding buffer (8 M urea, 10 mM Tris, 100 mM sodium phosphate, pH 8.0). Denatured protein was added to Ni-NTA resin (750 μL) and incubated at 4 $\circ C$ on a rotator for 1.5 hours. Resin was pelleted by centrifugation at 2500 x g and flowthrough was removed. Resin was washed 3 times with wash buffer (8 M urea, 10 mM Tris, 100 mM sodium phosphate, 20 mM imidazole, pH 6.5), followed by addition of elution buffer (8 M urea, 10 mM Tris, 100 mM sodium phosphate, 500 mM Imidazole, pH 6.5). Eluting was performed on a rotator at 4 $\circ C$ for 30 minutes. The resin was again pelleted by centrifugation at 2500 x g, and the sample was collected.

Refolding. The denatured sample was diluted 1:100 into refolding buffer (50 mM Tris, 0.5 M arginine, 20 mM $CaCl_2$, 1 mM EDTA, 100 mM NaCl, 0.01% NP-40, 0.05 mM GSSG, 0.5 mM GSH, pH 7.5) at room temperature using a syringe pump (flow rate 1 mL/min) while allowing the solution to gently stir. The refolding protein was left at 4 $\circ C$ for 3 days with gentle stirring.

The sample was concentrated 10-fold using Amicon Stirred Cells with 10 kDa Ultrafugation disks (Millipore, UFC801024). Once concentrated, the sample was dialyzed overnight into assay buffer (50 mM Tris, 500 mM NaCl, 0.001% NP-40, pH 7.5) at 4 $\circ C$.

Protein gel and silver staining. LDS loading dye was added to protein samples and samples were boiled for 5 minutes at 95 $\circ C$. 10 μL of each sample was loaded onto a 4-20% mini-PROTEAN TGX gel (BioRad, 4561096) and run at 180V for 45 minutes. Total protein was visualized using a Pierce Silver Staining Kit (Thermo, PI24612) following manufacturer's protocols.

Western Blot. After running gel as described above, protein was transferred to PVDF membrane using a BioRaD TransferBox Turbo following the standard protocols. Membrane was blocked for 1 hour at room temperature using Super Block (Thermo Scientific, 37515). TMPRSS2 antibody (Novus biologicals, NBP1-20984) was added to membrane (1:1000 dilution in Super Block) and incubated overnight at 4 $\circ C$ with gentle shaking. After removal of primary antibody and three washes with TBST, HRP conjugated secondary antibody (abcam, ab6741, 1:20,000 in Super Block) was added to membrane and incubated at RT for 1 hour with shaking. After removal of secondary antibody with three washes with TBST, HRP substrate (Thermo Scientific, 34095) was added and after 1 minute Western blot was visualized using Chemiluminescence on an Azure Biosystems c600 imager.

Kinetic assays. Assays were conducted on a Molecular Devices Spectramax Spectrophotometer using 96-well plates (Fisher, 12-565-501). Protein was first plated, followed by addition of substrate, Boc-QAR-AMC (Bachem, 4017019.0005) at concentrations to give the indicated final concentration in a 100 μL volume. After addition of substrate, fluorescence was immediately read (Ex: 380, Em: 460 nm), taking measurements every 30 seconds for 20 minutes. Active protein was quantified by titrating in the known active site protease inhibitor FPR-CMK (Haematologic Technologies). To determine the K_M the initial fluorescence data, at less than 10% substrate conversion, was fit to a linear equation and the slope was determined, V_0. V_0 was plotted vs substrate concentration and the data was fit to the Michaelis Menten equation using GraphPad Prism.

Fluorescence endpoint assays for IC_{50} determination. Assays were conducted in 384 well black plates (Costar, 4514) using an Envision plate reader, ex. filter 350 nm and em. filter 450 nm. The compounds were first plated (10 μL, at various concentrations) followed by addition of TMPRSS2 protein (8 μL, 0.5 nM final concentration). After 30 minute incubation (unless otherwise specified) at room temperature, substrate (2 μL, 2.5 μM final

concentration) was added. At 30 minutes, corresponding to less than 20% substrate cleavage as measured by comparing fluorescence of the negative control to free AMC (Millipore, 257370), fluorescence was read. Wells containing no TMPRSS2 protein (substrate only) served as positive controls. Wells containing no inhibitors (TMPRSS2 and substrate only) served as negative controls. Fluorescence readout was plotted against the log of inhibitor concentration and fit to log(inhibitor) vs response - variable slope equation in GraphPad Prism. Fluorescence endpoint assays with trypsin were conducted utilizing 1 nM protein and 5 μM substrate.

Cells and Virus. Calu-3, Caco-2, and Vero E6 cells were maintained at 37 $\circ C$ and 5% CO_2 in Dulbecco's Modified Eagle Medium (DMEM, Thermo Fisher) supplemented with 10% heat-inactivated fetal bovine serum (Fisher Scientific, 10-437-028) and 1X pen-strep (Gibco, 15140122). Cells were tested for mycoplasma contamination before usage and the results were negative. SARS-CoV-2 B.1.1.7 strain was obtained from BEI resources and was propagated in Vero E6 cells. Viral titers were determined by TCID50 assays in Vero-E6 cells using the Reed and Muench method by microscopic scoring. Viral titer of the B.1.1.7 stock was 1x108 TCID50/mL. All experiments using SARS-CoV-2 virus were conducted at the University of Michigan under Biosafety Level 3 (BSL3) protocols in compliance with containment procedures in laboratories for use by the University of Michigan Institutional Biosafety Committee (IBC) and Environment, Health and Safety (EHS).

SARS-CoV-2 Infection Bioassay. 384 well Imaging plates (Perkin-Elmer, 6057300) were seeded with Calu-3 or Caco-2 cells at 10,000 cells per well in 50 μL/well of complete DMEM. Cells were allowed to attach and grow in the plates for 5 days (37 $\circ C$ and 5% CO_2), after which they were roughly 70% confluent. Then, media was discarded and replaced with 30 μL/well of complete DMEM. Compounds (debrisoquine, propamidine, pentamidine, and camostat) were solubilized in molecular grade biology water at 20 mM and then diluted with media when appropriate. Compounds were added in a 10-point, 2-fold dilution series ($N = 4$ replicates per condition). Debrisoquine, propamidine, and pentamidine were all added in 10 μL of media with a top concentration of 500 μM. Camostat, which was used as a control TMPRSS2 inhibitor, was also added in 10 μL media, but with a top concentration of 100 μM. Compounds were pre-incubated for 1 hour, then the plate was transferred to a BSL3 lab

for infection with SARS-CoV-2 B.1.1.7 strain. Virus was added at a multiplicity of infection (MOI) of 10 in 10 μL DMEM to yield a final volume per well of 50 μL. The plates were then allowed to incubate at 37 $\circ C$ and 5% CO_2 for 48 hours. After incubation with virus, cells were fixed with 4% paraformaldehyde for 30 minutes, then permeabilized with 0.3% Triton X-100 for 20 minutes, washed 2x with cold PBS, and blocked with antibody buffer (1.5% bovine serum albumin, 1% goat serum and 0.0025% Tween 20). The 384 well plates were then sealed, surface decontaminated, and transferred to a BSL2 space for overnight labeling with anti-nucleocapsid SARS-CoV-2 primary antibody at 4 $\circ C$ (Antibodies Online, Cat# ABIN6952432). Following primary antibody staining, cells were washed 2x with PBS and stained with an optimized fluorescent dye set containing 1:1000 Hoechst 33342 Pentahydrate (bis-Benzimide) for nuclear labeling and 1:1000 secondary antibody Alexa-647 (goat anti-mouse, Thermo Fisher, A21235) for viral labeling for 30 minutes at room temperature. Cells were washed 2x with PBS, then left in a final volume per well of 50 μL PBS in preparation for fluorescent imaging.

Bioassay Imaging and Analysis. Fluorescent stained plates were imaged using a Thermo Fisher CX5 high-content microscopes 10X magnification. Images were taken with LED excitation (386/23 nm, 650/13 nm). Exposure times were varied to achieve optimal signal-to-background ratio and images were collected at a single Z-plane as determined by image-based autofocus on the nuclear channel. A total of 9 fields per well were imaged accounting for roughly 80% of the total well area. Images were analyzed using the open-source image segmentation software Cell-Profiler 4.0.5. A pipeline was designed to identify both nuclei (Hoechst 33342 channel) and viral objects (Alexa Fluor 647 channel). Infected cells were identified using the relate objects module, where any nucleus contained within a viral object was defined as infected. Cell counts and infected cell counts were exported at the field level and joined with treatment metadata for further analysis. Field level data was joined to yield a total number of infected cells per well. Infection score was determined by normalizing to the average raw percent infected of the virally infected controls ($N = 20$). Similarly, percent viability was determined by normalization to the average cell count per well of the virally infected control. Concentration response curves for this data were fit using GraphPad Prism 9.0 (Graphpad Software, San Diego, CA, USA) using a semilog 4-

parameter variable slope model. Representative images of cells were generated by overlaying the nuclear and viral channels in ImageJ.

BIBLIOGRAPHY

(1) Morris, G. M.; Lim-Wilby, M. In *Molecular modeling of proteins*; Springer: 2008, pp 365–382.

(2) Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J. *Nat. Rev. Drug Discov.* **2004**, *3*, 935–949.

(3) Yuriev, E.; Agostino, M.; Ramsland, P. A. *J. Mol. Recognit.* **2011**, *24*, 149–164.

(4) Taylor, R. D.; Jewsbury, P. J.; Essex, J. W. *J. Comput. Aided. Mol. Des.* **2002**, *16*, 151–166.

(5) Bender, B. J.; Gahbauer, S.; Luttens, A.; Lyu, J.; Webb, C. M.; Stein, R. M.; Fink, E. A.; Balius, T. E.; Carlsson, J.; Irwin, J. J., et al. *Nature protocols* **2021**, *16*, 4799–4832.

(6) Koshland Jr, D. E. *Angewandte Chemie International Edition in English* **1995**, *33*, 2375–2378.

(7) Ding, X.; Wu, Y.; Wang, Y.; Vilseck, J. Z.; Brooks III., Charles L *J. Chem. Theory Comput.* **2020**.

(8) Pagadala, N. S.; Syed, K.; Tuszynski, J. *Biophysical reviews* **2017**, *9*, 91–102.

(9) Wu, G.; Robertson, D. H.; Brooks III., Charles L; Vieth, M. *J. Comput. Chem.* **2003**, *24*, 1549–1562.

(10) Goodsell, D. S.; Morris, G. M.; Olson, A. J. *J. Mol. Recognit.* **1996**, *9*, 1–5.

(11) Trott, O.; Olson, A. J. *J. Comput. Chem.* **2010**, *31*, 455–461.

(12) Wu, Y.; Brooks III, C. L. *Journal of Chemical Information and Modeling* **2021**, *61*, 5535–5549.

(13) Ravindranath, P. A.; Forli, S.; Goodsell, D. S.; Olson, A. J.; Sanner, M. F. *PLOS Comput. Biol* **2015**, *11*.

(14) Miller, E.; Murphy, R.; Sindhikara, D.; Borrelli, K.; Grisewood, M.; Ranalli, F.; Dixon, S.; Jerome, S.; Boyles, N.; Day, T., et al. *ChemRxiv* **2020**.

(15) Patel, H.; Ihlenfeldt, W.-D.; Judson, P. N.; Moroz, Y. S.; Pevzner, Y.; Peach, M. L.; Delannée, V.; Tarasova, N. I.; Nicklaus, M. C. *Scientific data* **2020**, *7*, 1–14.

(16) Grygorenko, O. O.; Radchenko, D. S.; Dziuba, I.; Chuprina, A.; Gubina, K. E.; Moroz, Y. S. *Iscience* **2020**, *23*, 101681.

(17) Basak, S. C. *Curr Comput Aided Drug Des* **2012**, *8*, 1–2.

(18) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K., et al. *Nature* **2019**, *566*, 224–229.

(19) Stein, R. M.; Kang, H. J.; McCorvy, J. D.; Glatfelter, G. C.; Jones, A. J.; Che, T.; Slocum, S.; Huang, X.-P.; Savych, O.; Moroz, Y. S., et al. *Nature* **2020**, *579*, 609–614.

(20) Gorgulla, C.; Boeszoermenyi, A.; Wang, Z.-F.; Fischer, P. D.; Coote, P. W.; Padmanabha Das, K. M.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A., et al. *Nature* **2020**, *580*, 663–668.

(21) Sabe, V. T.; Ntombela, T.; Jhamba, L. A.; Maguire, G. E.; Govender, T.; Naicker, T.; Kruger, H. G. *European Journal of Medicinal Chemistry* **2021**, *224*, 113705.

(22) Peiffer, A. L.; Garlick, J. M.; Wu, Y.; Soellner, M. B.; Brooks III, C. L.; Mapp, A. K. *bioRxiv* **2021**, DOI: 10.1101/2021.03.22.436465.

(23) Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. *J. Chem. Inf. Model.* **2018**, *59*, 895–913.

(24) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. *J. Chem. Inf. Model.* **2014**, *54*, 1700–1716.

(25) Li, Y.; Han, L.; Liu, Z.; Wang, R. *J. Chem. Inf. Model.* **2014**, *54*, 1717–1736.

(26) Roche, O.; Kiyama, R.; Brooks III., Charles L *J. Med. Chem.* **2001**, *44*, 3592–3598.

(27) Gagnon, J. K.; Law, S. M.; Brooks III., Charles L *J. Comput. Chem.* **2016**, *37*, 753–762.

(28) Allen, W. J.; Balius, T. E.; Mukherjee, S.; Brozell, S. R.; Moustakas, D. T.; Lang, P. T.; Case, D. A.; Kuntz, I. D.; Rizzo, R. C. *J. Comput. Chem.* **2015**, *36*, 1132–1156.

(29) 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., et al. *J. Med. Chem.* **2004**, *47*, 1739–1749.

(30) Yu, W.; Lakkaraju, S. K.; Raman, E. P.; MacKerell, A. D. *Journal of computer-aided molecular design* **2014**, *28*, 491–507.

(31) Zavodszky, M. I.; Kuhn, L. A. *Protein Sci.* **2005**, *14*, 1104–1114.

(32) Bowman, G. R.; Geissler, P. L. *J. Phys. Chem. B* **2014**, *118*, 6417–6423.

(33) Teague, S. J. *Nat. Rev. Drug Discov.* **2003**, *2*, 527–541.

(34) Kuhn, L. A. *Computational and Structural Approaches to Drug DiscoVery: Ligand-Protein Interactions* **2007**, 181–191.

(35) Spraggon, G.; Hornsby, M.; Shipway, A.; Tully, D. C.; Bursulaya, B.; Danahay, H.; Harris, J. L.; Lesley, S. A. *Protein Science* **2009**, *18*, 1081–1094.

(36) Sun, W.; Zhang, X.; Cummings, M. D.; Albarazanji, K.; Wu, J.; Wang, M.; Alexander, R.; Zhu, B.; Zhang, Y.; Leonard, J., et al. *Journal of Pharmacology and Experimental Therapeutics* **2020**, *375*, 510–521.

(37) Kumalo, H. M.; Bhakat, S.; Soliman, M. E. *Molecules* **2015**, *20*, 1984–2000.

(38) Baillie, T. A. *Angew. Chem. Int. Ed.* **2016**, *55*, 13408–13421.

(39) Scarpino, A.; Ferenczy, G. G.; Keserű, G. M. *Curr. Pharm. Des.* **2020**.

(40) Haberthür, U.; Caflisch, A. *J. Comput. Chem.* **2008**, *29*, 701–715.

(41) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D., et al. *PLOS Comput. Biol.* **2017**, *13*, e1005659.

(42) Eastman, P. et al. *J. Chem. Theory Comput.* **2012**, *9*, 461–469.

(43) O'Boyle, N. M.; Morley, C.; Hutchison, G. R. *Chemistry Central Journal* **2008**, *2*, 1–7.

(44) Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling., 2013.

(45) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. *the Journal of machine Learning research* **2011**, *12*, 2825–2830.

(46) Sousa, S. F.; Fernandes, P. A.; Ramos, M. J. *Proteins: Struct., Funct., Bioinf.* **2006**, *65*, 15–26.

(47) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. *J. Med. Chem.* **2004**, *47*, 1750–1759.

(48) Brooks, B. R.; Brooks III., Charles L; Mackerell Jr, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S., et al. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(49) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P; Vorobyov, I., et al. *J. Comput. Chem.* **2010**, *31*, 671–690.

(50) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. *J. Cheminformatics* **2011**, *3*, 33.

(51) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. *J. Chem. Theory Comput.* **2013**, *9*, 3878–3888.

(52) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. *J. Mol. Graphics Modell.* **2010**, *29*, 116–125.

(53) Hynninen, A.-P.; Crowley, M. F. *J. Comput. Chem.* **2014**, *35*, 406–413.

(54) Hartshorn, M. J.; Verdonk, M. L.; Chessari, G.; Brewerton, S. C.; Mooij, W. T.; Mortenson, P. N.; Murray, C. W. *J. Med. Chem.* **2007**, *50*, 726–741.

(55) Mukherjee, S.; Balius, T. E.; Rizzo, R. C. *J. Chem. Inf. Model.* **2010**, *50*, 1986–2000.

(56) Vanommeslaeghe, K; Hatcher, E; Acharya, C; Kundu, S; Zhong, S; Shim, J; Darian, E; Guvench, O; Lopes, P; Vorobyov, I; Mackerell, A. D. *J. Comput. Chem.* **2010**, *31*, 671–690.

(57) Gaillard, T. *J. Chem. Inf. Model.* **2018**, *58*, 1697–1706.

(58) Feig, M.; Karanicolas, J.; Brooks III., Charles L *J. Mol. Graph. Model.* **2004**, *22*, 377–395.

(59) Raveh, B.; London, N.; Zimmerman, L.; Schueler-Furman, O. *PLOS ONE* **2011**, *6*, e18934.

(60) Gilli, P.; Ferretti, V.; Gilli, G.; Borea, P. A. *J. Phys. Chem.* **1994**, *98*, 1515–1518.

(61) Caro, J. A.; Harpole, K. W.; Kasinath, V.; Lim, J.; Granja, J.; Valentine, K. G.; Sharp, K. A.; Wand, A. J. *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114*, 6563–6568.

(62) Verteramo, M. L.; Stenstrom, O.; Ignjatovic, M. M.; Caldararu, O.; Olsson, M. A.; Manzoni, F.; Leffler, H.; Oksanen, E.; Logan, D. T.; Nilsson, U. J., et al. *J. Am. Chem. Soc.* **2019**, *141*, 2012–2026.

(63) Amaral, M.; Kokh, D.; Bomke, J; Wegener, A; Buchstaller, H.; Eggenweiler, H.; Matias, P; Sirrenberg, C; Wade, R.; Frech, M *Nat. Commun* **2017**, *8*, 1–14.

(64) Deng, W.; Breneman, C.; Embrechts, M. J. *J. Chem. Inf. Comput. Sci* **2004**, *44*, 699–703.

(65) Ballester, P. J.; Mitchell, J. B. *Bioinformatics* **2010**, *26*, 1169–1175.

(66) Springer, C.; Adalsteinsson, H.; Young, M. M.; Kegelmeyer, P. W.; Roe, D. C. *J. Med. Chem.* **2005**, *48*, 6821–6831.

(67) Kinnings, S. L.; Liu, N.; Tonge, P. J.; Jackson, R. M.; Xie, L.; Bourne, P. E. *J. Chem. Inf. Model.* **2011**, *51*, 408–419.

(68) Wang, R.; Lai, L.; Wang, S. *J. Comput. Aided Mol. Des.* **2002**, *16*, 11–26.

(69) Morris, G. M.; Goodsell, D. S.; Halliday, R. S.; Huey, R.; Hart, W. E.; Belew, R. K.; Olson, A. J. *J. Comput. Chem.* **1998**, *19*, 1639–1662.

(70) Inc, C. C. G. Molecular operating environment (MOE)., 2016.

(71) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. *J. Cheminformatics* **2011**, *3*, 33.

(72) Vanommeslaeghe, K.; MacKerell Jr, A. D. *J. Chem. Inf. Model.* **2012**, *52*, 3144–3154.

(73) Vanommeslaeghe, K.; Raman, E. P.; MacKerell Jr, A. D. *J. Chem. Inf. Model.* **2012**, *52*, 3155–3168.

(74) DeLano, W. L. et al. *CCP4 Newsletter on protein crystallography* **2002**, *40*, 82–92.

(75) Graves, A. P.; Brenk, R.; Shoichet, B. K. *J. Med. Chem.* **2005**, *48*, 3714–3728.

(76) Lind, K. E.; Du, Z.; Fujinaga, K.; Peterlin, B. M.; James, T. L. *Chem. Biol.* **2002**, *9*, 185–193.

(77) Detering, C.; Varani, G. *J. Med. Chem.* **2004**, *47*, 4188–4201.

(78) Kang, X.; Shafer, R. H.; Kuntz, I. D. *Biopolymers* **2004**, *73*, 192–204.

(79) Moitessier, N.; Westhof, E.; Hanessian, S. *J. Med. Chem.* **2006**, *49*, 1023–1033.

(80) Park, S.-J.; Jung, Y. H.; Kim, Y.-G.; Park, H.-J. *Bioorg. Med. Chem.* **2008**, *16*, 4676–4684.

(81) Morley, S. D.; Afshar, M. *J. Comput. Aided Mol. Des.* **2004**, *18*, 189–208.

(82) Barbault, F.; Zhang, L.; Zhang, L.; Fan, B. T. *Chemometr. Intell. Lab. Syst.* **2006**, *82*, 269–275.

(83) Pfeffer, P.; Gohlke, H. *J. Chem. Inf. Model.* **2007**, *47*, 1868–1876.

(84) Guilbert, C.; James, T. L. *J. Chem. Inf. Model.* **2008**, *48*, 1257–1268.

(85) Lang, P. T.; Brozell, S. R.; Mukherjee, S.; Pettersen, E. F.; Meng, E. C.; Thomas, V.; Rizzo, R. C.; Case, D. A.; James, T. L.; Kuntz, I. D. *RNA* **2009**, *15*, 1219–1230.

(86) Daldrop, P.; Reyes, F. E.; Robinson, D. A.; Hammond, C. M.; Lilley, D. M.; Batey, R. T.; Brenk, R. *Chem. Biol.* **2011**, *18*, 324–335.

(87) Creamer, T. P. *Proteins* **2000**, *40*, 443–450.

(88) Hoffman-Ostenhof, O; Cohn, W.; Braunstein, A.; Karlson, P; Keil, B; Klyne, W; Liebecq, C; Slater, E.; Webb, E.; Whelan, W. *J. Mol. Biol.* **1970**, *52*, 1–17.

(89) Franceschi, F.; Duffy, E. M. *Biochem. Pharmacol.* **2006**, *71*, 1016–1025.

(90) Fulle, S.; Gohlke, H. *J. Mol. Recognit.* **2010**, *23*, 220–231.

(91) Whitley, D. *Stat. Comput.* **1994**, *4*, 65–85.

(92) Chelouah, R.; Siarry, P. *J. Heuristics* **2000**, *6*, 191–213.

(93) Korb, O.; Stutzle, T.; Exner, T. E. *J. Chem. Inf. Model.* **2009**, *49*, 84–96.

(94) Vieth, M.; Hirst, J. D.; Kolinski, A.; Brooks III., Charles L *J. Comput. Chem.* **1998**, *19*, 1612–1622.

(95) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. *J. Med. Chem.* **2012**, *55*, 6582–6594.

(96) Huang, N.; Shoichet, B. K.; Irwin, J. J. *J. Med. Chem.* **2006**, *49*, 6789–6801.

(97) Graves, A. P.; Shivakumar, D. M.; Boyce, S. E.; Jacobson, M. P.; Case, D. A.; Shoichet, B. K. *J. Mol. Biol.* **2008**, *377*, 914–934.

(98) Boyce, S. E.; Mobley, D. L.; Rocklin, G. J.; Graves, A. P.; Dill, K. A.; Shoichet, B. K. *J. Mol. Biol.* **2009**, *394*, 747–763.

(99) Lee, H.; Fischer, M.; Shoichet, B. K.; Liu, S.-Y. *J. Am. Chem. Soc.* **2016**, *138*, 12021–12024.

(100) Liu, L.; Marwitz, A. J.; Matthews, B. W.; Liu, S.-Y. *Angew. Chem. Int.* **2009**, *48*, 6817–6819.

(101) Su, A. I.; Lorber, D. M.; Weston, G. S.; Baase, W. A.; Matthews, B. W.; Shoichet, B. K. *Proteins* **2001**, *42*, 279–293.

(102) Morton, A.; Baase, W. A.; Matthews, B. W. *Biochemistry* **1995**, *34*, 8564–8575.

(103) Mobley, D. L.; Graves, A. P.; Chodera, J. D.; McReynolds, A. C.; Shoichet, B. K.; Dill, K. A. *J. Mol. Biol.* **2007**, *371*, 1118–1134.

(104) Merski, M.; Fischer, M.; Balius, T. E.; Eidam, O.; Shoichet, B. K. *Proc. Natl. Acad. Sci.* **2015**, *112*, 5039–5044.

(105) Hoffmann, M.; Kleine-Weber, H.; Schroeder, S.; Krüger, N.; Herrler, T.; Erichsen, S.; Schiergens, T. S.; Herrler, G.; Wu, N.-H.; Nitsche, A., et al. *cell* **2020**, *181*, 271–280.

(106) Paoloni-Giacobino, A.; Chen, H.; Peitsch, M. C.; Rossier, C.; Antonarakis, S. E. *Genomics* **1997**, *44*, 309–320.

(107) Xia, X. *Viruses* **2021**, *13*, 109.

(108) Hoffmann, M.; Kleine-Weber, H.; Pöhlmann, S. *Molecular cell* **2020**, *78*, 779–784.

(109) Zhu, N.; Zhang, D.; Wang, W.; Li, X.; Yang, B.; Song, J.; Zhao, X.; Huang, B.; Shi, W.; Lu, R., et al. *New England journal of medicine* **2020**.

(110) Letko, M.; Marzi, A.; Munster, V. *Nature microbiology* **2020**, *5*, 562–569.

(111) Shang, J.; Wan, Y.; Luo, C.; Ye, G.; Geng, Q.; Auerbach, A.; Li, F. *Proceedings of the National Academy of Sciences* **2020**, *117*, 11727–11734.

(112) V'kovski, P.; Kratzel, A.; Steiner, S.; Stalder, H.; Thiel, V. *Nature Reviews Microbiology* **2021**, *19*, 155–170.

(113) Turner, A. J. *The protective arm of the renin angiotensin system (RAS)* **2015**, 185.

(114) Kim, T. S.; Heinlein, C.; Hackman, R. C.; Nelson, P. S. *Molecular and cellular biology* **2006**, *26*, 965–975.

(115) Stopsack, K. H.; Mucci, L. A.; Antonarakis, E. S.; Nelson, P. S.; Kantoff, P. W. *Cancer discovery* **2020**, *10*, 779–782.

(116) Shulla, A.; Heald-Sargent, T.; Subramanya, G.; Zhao, J.; Perlman, S.; Gallagher, T. *Journal of virology* **2011**, *85*, 873–882.

(117) Matsuyama, S.; Nagata, N.; Shirato, K.; Kawase, M.; Takeda, M.; Taguchi, F. *Journal of virology* **2010**, *84*, 12658–12664.

(118) Shirato, K.; Kawase, M.; Matsuyama, S. *Journal of virology* **2013**, *87*, 12552–12561.

(119) Shen, L. W.; Mao, H. J.; Wu, Y. L.; Tanaka, Y.; Zhang, W. *Biochimie* **2017**, *142*, 1–10.

(120) Hatesuer, B.; Bertram, S.; Mehnert, N.; Bahgat, M. M.; Nelson, P. S.; Pöhlman, S.; Schughart, K. *PLoS pathogens* **2013**, *9*, e1003774.

(121) Qiao, Y.; Wang, X.-M.; Mannan, R.; Pitchiaya, S.; Zhang, Y.; Wotring, J. W.; Xiao, L.; Robinson, D. R.; Wu, Y.-M.; Tien, J. C.-Y., et al. *Proceedings of the National Academy of Sciences* **2021**, *118*.

(122) Mahoney, M.; Damalanka, V. C.; Tartell, M. A.; hee Chung, D.; Lourenço, A. L.; Pwee, D.; Bridwell, A. E. M.; Hoffmann, M.; Voss, J.; Karmakar, P., et al. *Proceedings of the National Academy of Sciences* **2021**, *118*.

(123) Breining, P.; Frølund, A. L.; Højen, J. F.; Gunst, J. D.; Staerke, N. B.; Saedder, E.; Cases-Thomas, M.; Little, P.; Nielsen, L. P.; Søgaard, O. S., et al. *Basic & clinical pharmacology & toxicology* **2021**, *128*, 204–212.

(124) Shrimp, J. H.; Kales, S. C.; Sanderson, P. E.; Simeonov, A.; Shen, M.; Hall, M. D. *ACS pharmacology & translational science* **2020**, *3*, 997–1007.

(125) Hofmann-Winkler, H.; Moerer, O.; Alt-Epping, S.; Bräuer, A.; Büttner, B.; Müller, M.; Fricke, T.; Grundmann, J.; Harnisch, L.-O.; Heise, D., et al. *Critical care explorations* **2020**, *2*.

(126) Wang, D.; Li, Z.; Liu, Y. *Journal of infection and public health* **2020**, *13*, 1405–1414.

(127) Hempel, T.; Raich, L.; Olsson, S.; Azouz, N. P.; Klingler, A. M.; Hoffmann, M.; Pöhlmann, S.; Rothenberg, M. E.; Noé, F. *Chemical Science* **2021**, *12*, 983–992.

(128) Waterhouse, A.; Bertoni, M.; Bienert, S.; Studer, G.; Tauriello, G.; Gumienny, R.; Heer, F. T.; de Beer, T. A. P.; Rempfer, C.; Bordoli, L., et al. *Nucleic acids research* **2018**, *46*, W296–W303.

(129) Bienert, S.; Waterhouse, A.; de Beer, T. A.; Tauriello, G.; Studer, G.; Bordoli, L.; Schwede, T. *Nucleic acids research* **2017**, *45*, D313–D319.

(130) Guex, N.; Peitsch, M. C.; Schwede, T. *Electrophoresis* **2009**, *30*, S162–S173.

(131) Studer, G.; Rempfer, C.; Waterhouse, A. M.; Gumienny, R.; Haas, J.; Schwede, T. *Bioinformatics* **2020**, *36*, 1765–1771.

(132) Bertoni, M.; Kiefer, F.; Biasini, M.; Bordoli, L.; Schwede, T. *Scientific reports* **2017**, *7*, 1–15.

(133) Onufriev, A.; Case, D. A.; Bashford, D. *Journal of computational chemistry* **2002**, *23*, 1297–1304.

(134) Lee, M. S.; Feig, M.; Salsbury Jr, F. R.; Brooks III, C. L. *Journal of computational chemistry* **2003**, *24*, 1348–1356.

(135) Gong, X.; Chiricotto, M.; Liu, X.; Nordquist, E.; Feig, M.; Brooks III, C. L.; Chen, J. *Journal of computational chemistry* **2020**, *41*, 830–838.

(136) Fraser, B. J.; Beldar, S.; Seitova, A.; Hutchinson, A.; Mannar, D.; Li, Y.; Kwon, D.; Tan, R.; Wilson, R. P.; Leopold, K., et al. *BioRxiv* **2021**.

(137) Taminau, J.; Thijs, G.; De Winter, H. *Journal of Molecular Graphics and Modelling* **2008**, *27*, 161–169.

(138) Koch, J.; Uckeley, Z. M.; Doldan, P.; Stanifer, M.; Boulant, S.; Lozach, P.-Y. *The EMBO journal* **2021**, *40*, e107821.

(139) Peacock, T. P.; Goldhill, D. H.; Zhou, J.; Baillon, L.; Frise, R.; Swann, O. C.; Kugathasan, R.; Penn, R.; Brown, J. C.; Sanchez-David, R. Y., et al. *Nature Microbiology* **2021**, *6*, 899–909.

(140) Wan, X.; Yang, T.; Cuesta, A.; Pang, X.; Balius, T. E.; Irwin, J. J.; Shoichet, B. K.; Taunton, J. *JACS* **2020**, *142*, 4960–4964.

(141) Chowdhury, S. R.; Kennedy, S.; Zhu, K.; Mishra, R.; Chuong, P.; Nguyen, A.-u.; Kathman, S. G.; Statsyuk, A. V. *Bioorg. Med. Chem.* **2019**, *29*, 36–39.

(142) Shraga, A.; Olshvang, E.; Davidzohn, N.; Khoshkenar, P.; Germain, N.; Shurrush, K.; Carvalho, S.; Avram, L.; Albeck, S.; Unger, T., et al. *Cell Chem. Biol.* **2019**, *26*, 98–108.

(143) London, N.; Miller, R. M.; Krishnan, S.; Uchida, K.; Irwin, J. J.; Eidam, O.; Gibold, L.; Cimermančič, P.; Bonnet, R.; Shoichet, B. K., et al. *Nat. Chem. Biol.* **2014**, *10*, 1066–1072.

(144) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. *J. Mol. Biol.* **1997**, *267*, 727–748.

(145) Jones, G.; Willett, P.; Glen, R. C. *J. Mol. Biol.* **1995**, *245*, 43–53.

(146) Verdonk, M. L.; Cole, J. C.; Hartshorn, M. J.; Murray, C. W.; Taylor, R. D. *Proteins: Struct., Funct., Bioinf.* **2003**, *52*, 609–623.

(147) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S.; Olson, A. J. *J. Comput. Chem.* **2009**, *30*, 2785–2791.

(148) Bianco, G.; Forli, S.; Goodsell, D. S.; Olson, A. J. *Protein Science* **2016**, *25*, 295–301.

(149) Zhu, K.; Borrelli, K. W.; Greenwood, J. R.; Day, T.; Abel, R.; Farid, R. S.; Harder, E. *J. Chem. Inf. Model.* **2014**, *54*, 1932–1940.

(150) Toledo Warshaviak, D.; Golan, G.; Borrelli, K. W.; Zhu, K.; Kalid, O. *J. Chem. Inf. Model.* **2014**, *54*, 1941–1950.

(151) Corbeil, C. R.; Englebienne, P.; Moitessier, N. *J. Chem. Inf. Model.* **2007**, *47*, 435–449.

(152) Abagyan, R.; Totrov, M.; Kuznetsov, D. *J. Comput. Chem.* **1994**, *15*, 488–506.

(153) Vilar, S.; Cozza, G.; Moro, S. *Current topics in medicinal chemistry* **2008**, *8*, 1555–1572.

(154) Scarpino, A.; Ferenczy, G. G.; Keserű, G. M. *J. Chem. Inf. Model.* **2018**, *58*, 1441–1458.

(155) Irwin, J. J.; Shoichet, B. K. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.

(156) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.

(157) Ouyang, X.; Zhou, S.; Su, C. T. T.; Ge, Z.; Li, R.; Kwoh, C. K. *J. Comput. Chem.* **2013**, *34*, 326–336.

(158) Sterling, T.; Irwin, J. J. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337.

(159) Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; Gilson, M. K. *Nucleic Acids Res.* **2007**, *35*, D198–D201.

(160) Riniker, S.; Landrum, G. A. *J. Chem. Inf. Model.* **2015**, *55*, 2562–2574.

(161) Wang, S.; Witek, J.; Landrum, G. A.; Riniker, S. *J. Chem. Inf. Model.* **2020**, *60*, 2044–2058.

(162) Wong, T.-T. *Pattern Recognit.* **2015**, *48*, 2839–2846.

(163) Luo, Y. L. *J. Chem. Inf. Model.* **2021**, *61*, 5307–5311.

(164) Li, A.; Sun, H.; Du, L.; Wu, X.; Cao, J.; You, Q.; Li, Y. *J. Mol. Model.* **2014**, *20*, 1–13.

(165) London, N.; Farelli, J. D.; Brown, S. D.; Liu, C.; Huang, H.; Korczynska, M.; Al-Obaidi, N. F.; Babbitt, P. C.; Almo, S. C.; Allen, K. N., et al. *Biochem.* **2015**, *54*, 528–537.

(166) vanRossum, G. *Department of Computer Science [CS]* **1995**.

(167) Van Rossum, G. et al. In *USENIX annual technical conference*, 2007; Vol. 41, pp 1–36.