2016

Data-Driven Rational Drug Design

Yun Ding
Louisiana State University and Agricultural and Mechanical College

DATA-DRIVEN RATIONAL DRUG DESIGN

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

Department of Physics and Astronomy

by
Yun Ding
BS, Wuhan University, 2007
December 2016

ACKNOWLEDGMENTS

I'd like to thank my parents, Shihuang Ding and Xiuying Ye, for their unrelenting support through my life, allowing the only child of the family to go abroad and pursue his interest. They taught me to lead a good life style, to be perseverant upon difficult tasks, and always to keep a curious mind; these carried me through the last 5 years of my PhD.

To my advisors, Mark Jarrell and Michal Brylinski, I owe not only my scientific maturity but also my sense of duty to apply it well. To my committee members, Prof. Waldrop and Prof. Matthews, thank you very much for keeping me on track.

Several collaborators have been indispensable in the completion of my several projects. Ye Fang is the main contributor to GeauxDock code base. David M. Koppelman has given a lot of invaluable advices on the parallelization and benchmarking. Many brilliant ideas were sparked from discussions with Wei P. Feinstein, Jagannathan Ramanujam and Juana Moreno.

To my friends and colleagues in the Department of Physics and Department of Biology, thank you for always being there when I am in a great need of help, no matter in research or in my personal life.

The text of Chapter 2 is a reprint of the materials as it appears in:
Ding, Y. et al., 2016. Assessing the similarity of ligand binding conformationswith the Contact Mode Score. Computational Biology and Chemistry, 64, pp.403413.

It appears here with permission from the authors. The letter of permission from the publisher is in Appendices A. The dissertation author is the primary author of this paper.

The text of Chapter 3 is a reprint of the materials as it appears in:
Ding, Y. et al., 2015. GeauxDock: A novel approach for mixed-resolution ligand docking using a descriptor-based force field. Journal of Computational Chemistry, 36(27), pp.20132026.

It appears here with permission from the authors. The letter of permission from the publisher is in Appendices B. The dissertation author is the primary author of this paper.

The text of Chapter 3 is a reprint of the materials as it appears in:
Fang, Y. et al., 2016. GeauxDock: Accelerating Structure-Based Virtual Screening with Heterogeneous Computing. Plos One, 11(7), p.e0158898.

It appears here with permission from the authors. According to the publisher, there is no need to request permission for any kind of re-use (`http://blogs.plos.org/everyone/authors/qa/`). The dissertation author is the second author of this paper, but equally contributed to the work. The contribution from the dissertation author is identified at the start of each section in the chapter.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Vast amount of experimental data in structural biology has been generated, collected and accumulated in the last few decades. This rich dataset is an invaluable mine of knowledge, from which deep insights can be obtained and practical applications can be developed. To achieve that goal, we must be able to manage such "Big Data" in science and investigate them expertly. Molecular docking is a field that can prominently make use of the large structural biology dataset. As an important component of rational drug design, molecular docking is used to perform large-scale screening of putative associations between small organic molecules and their pharmacologically relevant protein targets. Given a small molecule (ligand), a molecular docking program simulates its interaction with the target protein, and reports the probable conformation of the protein-ligand complex, and the relative binding affinity compared against other candidate ligands.

This dissertation collects my contributions in several aspects of molecular docking. My early contribution focused on developing a novel metric to quantify the structural similarity between two protein-ligand complexes. Benchmarks show that my metric addressed several issues associated with the conventional metric. Furthermore, I extended the functionality of this metric to cross different systems, effectively utilizing the data at the proteome level. After developing the novel metric, I formulated a scoring function that can extract the biological information of the complex, integrate it with the physics components, and finally enhance the performance. Through collaboration, I implemented my model into an ultra-fast, adaptive program, which can take advantage of a range of modern parallel architectures and handle the demanding data processing tasks in large scale molecular docking applications.

Chapter 1
INTRODUCTION

This thesis discusses how to improve the rational drug design by taking advantage of the abundant data, often referred as the Big Data, in structural biology.

1.1 Data-intensive scientific exploration

Big Data has become the buzzword of the day [1,2]. The term was firstly introduced in the commercial world [3], referred as the data sets and analytical techniques in applications that are so large (from terabytes to exabytes) and complex (from sensor to social media data) that they require advanced and unique data data storage, management, analysis and visualization technologies [4]. Information technology companies are the first few Big Data players. Google invented the MapReduce technology to be able to analyze billions of web sites across the world [5]. E-commerce platforms such as Amazon collect user search and interaction logs on a 24/7 basis in order to better understand the customer needs and identify new business opportunities [6,7]. Even in more conventional industries, Big Data can also play a pivotal role [8]. A typical example is that supply chain management applies large scale predictive analytics to enhance the efficiency by estimating past and future levels of integration of business processes among functions or companies [2].

Apart from the commercial world, Big Data is emerging as a new, fourth paradigm for scientific exploration [9]. Science used to be categorized as either experimental or theoretical. Then, for many problems the theoretical models grew too complicated to solve analytically, and simulation as the third paradigm came forth. Today, researchers are experiencing a moving from data paucity to a data plethora that unprecedented amount of data are being generated by simulations and experiments. This ready availability of diverse data is shifting scientific approaches from the traditional, hypothesis-driven scientific method to methods more exploration-based [10,11]. Since the techniques and methodologies to conduct such exploration are so different that it is worth distinguishing such approach from

computational science as a new, fourth paradigm for scientific exploration, data-intensive scientific exploration(DISE) [9].

DISE consists of three basic and consecutive activities: data capture, curation and analysis [1].

Data in DISE comes in a variety of scales and shape, covering large international experiments; cross-laboratory, single-laboratory, individual observations [9]; and even potentially individual lives [12]. Large international experiments, such as the Australia Square Kilometre Array of radio telescope project [13], CERN's Large Hadron Collider [14], and Pan-STARRS [15] array of celestial telescopes are capable of generating several petabytes of data per day. Being shared by many different research teams, these large data detectors usually conform to a very standard way to codify the raw data. Data from other disciplines can be much more heterogeneous. In ecological science, data are generated by a wide variety of groups using a wide range of sampling or simulation methodologies and data standards [16]. Similarly in bioinformatics, data can range from metabolic pathways to the behavior of a cell to the structure of proteins [17, 18]. To facilitate the data sharing and promote the collaboration between different research group, considerable amount of work was devoted into developing programs to translate data between different forms [19, 20]

Data curation is as much important as data capture and analysis. Without proper assembling, organizing and maintaining of the data, scientists can not guarantee data quality, or data reusing and preservation over time [1]. Notwithstanding that the first data-driven scientific discovery can date back to 400 years ago, when Johannes Kepler took Brache's catalog of systematic astronomical observations and discovered the laws of planetary motion, through the 20th century, data sit "under-analyzed in databases all over the world" [21]. Most of the data on which scientific theories were based was often buried in individual scientific notebooks, which were likely to be thrown out when a scientist retires, or at best be held in libraries until it is discarded. Long-term data provenance as well as community access to distributed data were the main challenges [22]. Recognizing the

growing importance of data curation and communication for scientific research, the National Science Board of the National Science Foundation published "Long-Live Digital Collections: Enabling Research and Education in the 21st Century" [23]. This report highlighted the urgency to build cyberinfrastructures to facilitate DISE. National Center for Atmospheric Research (NCAR) [24] is one of the first few sites for the modeling, collective and curation of Earth science data. The San Diego Supercomputer Center [25], though normally associated with supplying computational power to scientific computation, recognized the need to add data to its mission, and set aside 400 terabytes of disk space for both public and private databases in 2009. There are also commercial incentives to address the infrastructure needs in science. Arend Sidow, a computational biologist at Stanford University, co-founded a company called DNAnexus as a platform for sharing and management of genomic data and tools to accelerate genomics study [26].

Being able to collect and maintain massive amount of scientific data only lays the foundation of DISE. In order to have insights into those vexing questions that previously would have been infeasible to address, it is crucial to be able to analyze the data expertly [27, 28].

At the core of analytics in DISE is machine learning (ML), an important subject of artificial intelligence which aims to design algorithms that allow computers to evolve behaviors based on empirical data [1, 29, 30]. It is interesting to note that ML has been applied to a wide variety of scientific fields [31], including astronomy [32], particle physics [33], bioinformatics [34], social science [27], medical health *et al* [11, 35]. While each field has its own version of scientific process, the cycle of observing, creating hypotheses, testing and iteratively building up comprehensive testable models or theories is shared cross disciplines. For each stage of this abstracted scientific process, there are relevant developments in ML that will lead to semiautomatic supportive tools [31].

At the outset of DISE, exploration of high volume of data can be greatly facilitated by various ML techniques when the data can hardly be handled by human perception. Pattern

recognition has been adopted by particle physics for decades for significant events detection [33]. A recent example is event analysis for Cherenkov detectors used in neutrino oscillation experiments [36]. Another promising approach is dimensionality reduction: reducing data from many original dimensions (e.g., thousands of gene expression measurements) to just a few dimensions in a new space, for the purpose of easier visualization on computer display [37]. Clustering of coexpressed genes is often used in expression bioinformatics to investigate direct or indirect coregulation [38].

ML techniques discussed above provide researcher with better tools to perceive the raw data and formulate hypotheses. In addition, ML can assist studying and validating the hypotheses using a handful of methods developed by the community to learn good models from the data. Specifically, when the observed data can be labeled by a scientist as continuous numbers, or categorical values as "positive" and "negative", supervised ML models can be learned to predict the labels. It should be noted that label prediction can not ultimately replace the more traditional component of the scientific method [39]. While supervised ML models try to generalize regularities from the data to make predictions, science aims to employ those regularities to construct a unified means of understanding them *a priori*. In that case, models that include the learning of causal mechanisms are more suitable (e.g., Bayesian networks, graphical models, or nonlinear regression [40]). This type of ML models are named Generative models. The other type of ML models, a.k.a Discriminative models, strive only to capture the ability to make predictions, while making no attempt to explicitly capture the true underlying physics of the phenomena. However, discriminative models can also provide insights into the nature [31]. Many recent successful applications of Discriminative models gives valuable information, including such aspects as which input dimensions are most useful, which examples are most likely to be outliers, and what new observations might be most worthwhile to gather [41, 42].

DISE relies on cloud computing and parallel computing to manage and speed up the processing of the unprecedented quantity of data. Cloud computing centers are built with

the basic idea that "algorithm and calculation should be brought to the data instead of the other way around". Because the size of data is much larger than the program, it is too expensive and inefficient to download the data and analyze locally. In that regard, cloud computing centers usually consist of two components, a data center serving as the "Big Data parking lots" where data resides, in addition to a computing center serving as the "processing workhorse" where data are analyzed. Large cloud computing centers create economies of scale in facility design and construction, equipment acquisition and maintenance, mitigating the possible technical barriers for individual laboratories [43].

Parallel computing in DISE takes place usually in two forms. The first form is the so-called "embarrassing parallelism", where many independent data processing tasks are simply partitioned over multiple computing nodes while no communication is required in between. Such type of data analysis appears repeatedly in large-scale scientific analyses. For example, consider the task of matching one gene sequence against millions of template gene sequences [44], or searching for anomalies in scans of brain images [45]. While "embarrassing parallelism" seeks to speed up the processing at the task level using multiple nodes, parallelism can also be exploited at the hardware level of each individual node and/or across the nodes, which I call "serious parallelism". This paradigm of parallelism involves more data sharing and message exchanging using programming models such as MPI [46], OpenMP [47] and CUDA [48]. Conducted at a very low level of abstraction, these programming models often requires the codes be broken into components that run on specific processor. In many ways, the state of "serious parallelism" nowadays is similar to the early days of computing, when programs were written in assembly languages for a specific architecture and had to be rewritten to run on a different machine. In the case of "serious parallelism" in DISE, one huge advantages of coding in low-level languages is ultra-fast processing speed, but that is at the cost of low reusability and great difficulty for the domain scientists who prefer high-level languages such as Matlab and Python.

In *Chapter 4, GeauxDock computing*, I will discuss how to strike a balance between code-reusing and platform-crossing.

1.2 Rational drug design

Drug discovery is a process of identifying a small group of bioactive compounds from a vast collection of candidates. Specifically, drugs are those compounds that can bind and modulate the function of a target protein implicated in a disease state. A drug molecule must prossess a certain geometry and physicochemical properties in order to have a sufficiently high binding affinity toward a given macromolecular target. As a result, the number of bioactive compounds is very small compared to a vast collection of candidate compounds. The ZINC database of commercially available small molecule entities consists of 17,900,742 drug-like compounds collected from 243 vendors as of January 2016 [49]. Considering molecules yet to be synthesized, the chemical universe comprises an estimated novemdecillion (10^{60}) of small organic compounds [50]. In the early stage of drug discovery, this large number of candidates need to be downsized to hundreds or thousands of the most promising compounds. Experimental high-throughput screening is a conventional approach used by the pharmaceutical industry to identify bioactive molecules, however, it suffers from high costs and relatively low hit rates [51]. For instance, a recent study by the Tufts Center for the Study of Drug Development estimates that the development of a new prescription medicine typically continues for longer than a decade with the total costs of over 2.5 billion US dollars [52].

Rational drug design employs computational modeling to reduce the overall costs, improve the efficiency and speed up the drug development time [53, 54]. Instead of experimental high-throughput screening, candidate small molecules are virtually screened (VS) by computational modeling before sending to experiment tests [55]. Current VS techniques fall into two main categories: ligand-based similarity search and structure-based molecular docking [56]. Although the experimentally solved structures of target proteins are not required in the ligand-based approach, an initial set of already developed drug molecules

must be known. This information, however, is often missing, particularly for novel protein targets. On the other hand, the advances in X-ray crystallography and nuclear magnetic resonance constantly accumulate structures of biological molecules at atomic-level, which foster structure-based drug discovery projects [57, 58].

1.3 Molecular docking

Structure-based molecular docking mainly utilizes the vast quantity of data in structural biology, where various methodologies in DISE can be applied to improve the state-of-the-art.

Data in structural biology is captured in a distributed way through each individual laboratory at research universities or institutes. X-ray crystallography [59] is currently the mostly used technique for structure determination of biological macromolecules at atomic resolution. A second method is nuclear magnetic resonance (NMR) [60]. NMR provides data that are in many ways complementary to those obtained from X-ray crystallography and thus widen our view of protein molecules. After the structure of the macromolecule is determined, the information will be recorded in text formats, and then uploaded to data centers for archiving. There are several mirror centers over the world, all under the name of "Protein Data Bank"(PDB), including RCSB PDB and BMRB (Biological Magnetic Resonance Data Bank) in the USA, PDBe in Europe, and PDBj in Japan. As the "Big Data parking lots", these PDB sites serve well. As the day of writing (October 2016), a total of 123,273 entries about experimentally-determined structures of proteins, nucleic acids, and complex assemblies are archived and publicly accessible at RCSB PDB, which occupies over 555 GBbytes of disk usage [61]. Nevertheless, the functionality of "processing workhorse" is missing in these data centers, which means that over 555 GBbytes of files need to be transferred to perform any comprehensive analysis on the data. This missing functionality is currently under the intensive development [62].

Building a molecular docking model can be tackled as training a supervised machine learning model [63–66]. Specifically, the training dataset consists of a set of training ex-

amples, and each example is a pair of an input object (known as the feature vector), and a desired output value (known as the target value). A supervised learning algorithm examines the training data and yield a fitted function, which can be used for mapping new examples. An optimal fitted function will determine the labels as accurate as possible for even unseen instances [67]. In molecular docking, supervised learning algorithms take in a large number of candidate ligands and their various conformations are evaluated by the supervised learning algorithm, then report the label for each candidate ligand. The feature vectors are the quantification of different nature (geometrical, physicochemical interactions, pharmacophore) of protein-ligand complexes [63], while the target values are the predicted conformation of the ligands and the corresponding binding affinity. In chapter 2 and 3, I will discuss how to improve the supervised ML model for molecular docking by deploying a more advanced target value as well as formulating a better feature vector.

Rational drug design often involves running molecular docking programs over a large number of candidate ligands against the target protein. In this situation, assessing each protein-ligand pair can be viewed as one independent task, and "embarrassing parallelism" is very suitable. However, in order to achieve higher accuracy, more sophisticated molecular docking models are emerging, which has a rising demand on the processing speed for each docking task [68, 69]. Therefore, conventional "embarrassing parallelism" needs to be complemented with "serious parallelism" to meet the demand. In chapter 4, I will discuss about how to achieve this goal in molecular docking using modern heterogeneous computing platforms.

1.4 Guide to the chapters

This research work aims to improve the current methodologies in rational drug design by (1) investigating into the two main aspects in supervised learning: target value and the feature vector, and (2) implementing the methods on parallel computing architects to power such data-intensive scientific calculations.

In *Chapter 2 Contact mode score*, we superseded the conventional target value, RMSD, with our newly developed Contact Mode Score (CMS), which addresses several issues with RMSD. CMS mitigates the dependence on the ligand size, and can be applied to evaluate flexible docking methods simulating receptor conformational changes upon ligand docking. We further developed eXtended Contact Mode Score, or XCMS, which capitalizes on the conservation of ligand binding across structurally similar pockets occupied by chemically similar ligands. For instance, it can be used to systematically evaluate complex structures constructed by virtual screening, where a retrospective assessment cannot be performed because the experimental structures of the majority of complexes are unavailable.

In *Chapter 3 GeauxDock engine*, we feature engineered a hybrid model incorporating evolutionary related information as well as physics attributes of the target complex. Benchmark calculations demonstrate that the model has a strong capacity to recognize native-like binding modes. In addition, the model reveals that evolutionary information can be effectively compensated by the increased contribution from physics-based attributes, which successively help maintain the accuracy of the model at the low level of evolutionary information. Therefore, our model is well suited for proteome-scale applications utilizing increasingly growing data in structural biology.

In *Chapter 4 GeauxDock computing*, we implemented the GeauxDock model on heterogeneous computing platforms using a modular code framework, which supports modern multi-core CPU, as well as Xeon Phi and GPU accelerators with significant speedup compared with serial codes. In addtion to the evaluation of the computational performance, we examined the energy consumption and hardware costs, and found that heterogeneous computing platforms offer considerable advantages over traditional CPU-based systems.

Chapter 2
CONTACT MODE SCORE

2.1 Introduction

Management guru and author Peter Drucker famously observed, "if you can't measure it, you can't improve it". Drucker means that you can't know whether or not you are successful unless success is defined and tracked. Only with a clearly established metric, one can quantify progress and adjust the process to produce the desired outcome.

In an molecular docking simulation, the success is often measured using the root-mean-square deviation (RMSD) [70]. Typically, predictions within an RMSD of 2 Å are considered successful, whereas values higher than 3 Å indicate that the docking failures. A standard RMSD function that quantifies the difference between two poses of the same molecule is computed as follows:

$$RMSD(A,B) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||a_i - b_i||^2} \tag{2.1}$$

where molecule poses $A = a_1, a_2, ..., a_n$ and $b_1, b_2, ..., b_n$ are defined by sets of Cartesian coordinates a_i and b_i of individual heavy (non-hydrogen) atoms. This formulation shows that the RMSD is calculated based on a predefined one-to-one correspondence between atoms in poses A and B. Although equivalent atoms can be found by matching atom indices, the presence of symmetric functional groups may result in inflated RMSD values [71]. Several modified RMSD calculation methods were developed to handle symmetric molecules [71, 72]. These techniques re-index atoms dynamically instead of using the predefined order of atoms.

Further, a strong dependence of the RMSD on the number of atoms complicates the assessment of molecules with different sizes [73, 74]. On the other hand, the development and optimization of scoring functions for molecular docking often involves tuning force

field parameters against diverse datasets of protein-ligand complexes. For example, weight factors can be adjusted to maximize the capability to recognize near native conformations amongst a large set of docking decoys [75–77]. An imprecise classification of near native and decoy conformations, e.g. by using a fixed RMSD threshold, may lead to suboptimal weight factors. Even though the number of ligand atoms can be taken into account by calculating the statistical significance of RMSD values [73, 74], statistical testing is rarely employed in the development and optimization of docking algorithms and scoring functions.

Another issue is that ligand RMSD does not account for the protein environment [78]. Depending on the ligand size and complexity, low RMSD values can be obtained even if key interactions with the protein are absent. Conversely, a substantial deviation from the experimental structure of a moiety that is irrelevant to binding (e.g., a solvent-exposed group) can notably increase the RMSD even when crucial binding features are recovered by docking calculations [79]. To address this problem, the relative displacement error (RDE) [80] was developed. The RDE down-weights large deviations, therefore, it is less sensitive to a small number of misplaced atoms compared to the RMSD. Nevertheless, similar to RMSD, the RDE takes no account of the protein environment.

Although conventional docking methods employ a single, static structure of the receptor, more recent approaches incorporate protein flexibility by docking against protein ensembles or using rotamer libraries for binding residue side chains [69, 81, 82]. The traditional ligand RMSD cannot be used to assess the accuracy of fully flexible molecular docking, where not only ligands, but also receptors change their internal conformations. For that reason, an alternative measure based on real space R-factors was proposed to compare electron density rather than to calculate the RMSD from Cartesian coordinates [79]. Moreover, predicted binding modes can be visually inspected in order to identify key protein-ligand interactions recovered by docking calculations [78]. However, the lack of automation makes this approach inapplicable to large datasets of docked ligand conformations.

The calculation of RMSD is straightforward and has a low computational complexity, therefore, it is still frequently used as the assessment measure, particularly across large datasets of protein-ligand complexes. Nevertheless, new techniques are highly desired to evaluate not only purely geometrical features, but also biological aspects of binding. On that account, we developed the Contact Mode Score (CMS), which effectively quantifies the similarity of ligand binding conformations. CMS compares the sets of interatomic contacts formed by a ligand and its receptor rather than ligand Cartesian coordinates. Such an approach also allows for the protein environment to be included in the assessment. Further, we developed the eXtended Contact Mode Score (XCMS), which provides a convenient template-based method to compare those protein-ligand complexes composed of different proteins and non-identical ligands. In contrast to the RMSD, CMS and XCMS are less dependent on the ligand size and have a well-defined statistical significance.

2.2 Materials and methods

2.2.1 Experimental Datasets

Three datasets of protein-ligand complexes are used in this study. The first dataset was compiled from the eFindSite library [83] by clustering template proteins at 40% sequence identity using PISCES [84], and then selecting representative chains that non-covalently bind small organic molecules at distinct locations. This procedure produced a set of 14,059 non-redundant structures of protein-ligand complexes, referred to as the eFindSite dataset, which was used to develop a mixed-resolution model of complex structures. In addition, we used the Astex/CCDC dataset [85] comprising the high-quality experimental structures of 201 pharmacologically relevant proteins co-crystalized with drug molecules. The dependence of CMS and RMSD on the number of ligand atoms was examined against the Astex/CCDC dataset. Finally, the XCMS was developed and tested on the BioLiP database [86]. BioLiP provides a comprehensive collection of protein-ligand complex structures curated specifically for studies focusing on biologically relevant interactions

and template-based modeling approaches. From the entire database comprising 94,887 ligands bound to 71,359 proteins, we randomly selected 2,200 protein-ligand complexes as query structures. In XCMS benchmarking, we searched the complete BioLiP database for non-identical templates for each query structure. A complex was used as the template if the Pocket Similarity score (PS-score) against the query pocket is <0.9, the fingerprint Tanimoto coefficient (1D-TC) against the query ligand is >0.5, and the number of ligand heavy atoms is greater than 6. Using these criteria produced a dataset of 802,058 query-template pairs to benchmark the XCMS. The PS-score measures the structural similarity of two ligand binding sites; it ranges from 0 to 1 with higher values indicating higher similarity [87]. 1D-TC employs 1024-bit molecular fingerprints to quantify the chemical similarity of two small molecules. The calculations of 1D-TC were conducted with OpenBabel [88], which supports fingerprint indexing to accelerate searches against large databases.

2.2.2 Simulated Datasets

In addition to experimental datasets, three sets of computer-generated structures were compiled for benchmarking purposes. The first simulated dataset is based on Astex/C-CDC [85] and it was prepared to assess the dependence of RMSD and CMS on the number of ligand heavy atoms. A series of systematic perturbations were applied to co-crystalized ligands, each comprising random translations and rotations about the x, y and z-axis of up to 0.02 and 5 deg, respectively. After each round of perturbation, RMSD and CMS were computed against the native conformation of a ligand. The second simulated dataset contains Metropolis Monte Carlo (MMC) trajectories constructed by GeauxDock [77] for Astex/C-CDC complexes. GeauxDock employs a mixed-resolution representation of protein-ligand complexes and a hybrid scoring function comprising physics-, evolution-based energy terms and statistical potentials. By lowering the unitless pseudo energy calculated by the scoring function, GeauxDock effectively finds the near native structures of protein-ligand complexes by exploring low-energy configurations according to a dimensionless scoring function. Here, binding ligands were initialized at random conformations and GeauxDock simulation en-

gine [77] was used to generate docking trajectories through 800 MMC cycles. The CMS was calculated for each accepted conformation against the ligand bound in the crystal complex structure.

The last simulated dataset was built on BioLiP [86] to benchmark RMSD,CMS and XCMS using predicted and random ligand conformations. First, query ligands were randomized within receptor binding pockets to produce a set of 2,200 random conformations of query ligands. Subsequently, each randomized ligand was re-docked to the protein with AutoDock Vina [72]. The docking box was set to an optimal size based on the radius of gyration of the ligand [89] and the binding pocket center was set to the geometric center of the compound bound in the experimental complex. This procedure produced 2,200 docked conformations of query ligands. For each simulated conformation, RMSD and CMS were calculated against the experimental structure, whereas the XCMS was calculated using a template. Similar to the experimental BioLiP dataset, we included only those templates having more than 6 heavy atoms, a PS-score of <0.9, and a 1D-TC of >0.5. For the template-based assessment with XCMS, suitable templates were identified for a subset of 695 targets.

2.2.3 Molecular Representation

Fast computation without compromising molecular details is achieved by describing protein-ligand complex structures at a mixed-resolution. A heavy-atom representation is used for ligands with the following chemical types according to SYBYL [90]: carbon sp ($C.1$), carbon sp2 ($C.2$), carbon sp3 ($C.3$), aromatic carbon ($C.ar$), carbocation in guadinium groups ($C.cat$), nitrogen sp ($N.1$), nitrogen sp2 ($N.2$), nitrogen sp3 ($N.3$), positively charged nitrogen sp3 ($N.4$), amide nitrogen ($N.am$), aromatic nitrogen ($N.ar$), trigonal planar nitrogen ($N.pl3$), oxygen sp2 ($O.2$), oxygen sp3 ($O.3$), oxygen in carboxylate and phosphate groups ($O.co2$), phosphorous sp3 ($P.3$), sulfur sp2 ($S.2$), sulfur sp3 ($S.3$), sulfoxide sulfur ($S.O$), sulfone sulfur ($S.O2$), and halogens (Br, Cl, F, I). Proteins are represented at the coarse-grained level. Proteins are represented at the coarsed-grained level.

In CMS, two effective backbone points per residue are placed at the position of its Cα atom (CA) and the geometrical center of the peptide plane (PP). Small side chains of Ala, Asn, Asp, Cys, Ile, Leu, Pro, Ser, Thr and Val are reduced to one pseudo atom located at the geometric center, Leu, Pro, Ser, Thr and Val are reduced to one pseudo atom located at the geometric center, whereas longer side chains of Arg, Gln, Glu, His, Lys, Met, Phe, Trp and Tyr are described by two effective points corresponding to the middle of a virtual Cβ-Cγ bond and the geometric center of the remaining side-chain atoms [91]. It is noteworthy that this model is already implemented in a molecular docking program, GeauxDock [77]. In XCMS, two effective points per residue are used at the positions of its Cβ-Cγ atoms (CA and CB, respectively), except for glycine that has only the CA atom.

2.2.4 Intermolecular Contacts

Contacts between ligand heavy atoms and protein effective points in the mixed-resolution model are calculated using type-dependent distance thresholds. These threshold values were optimized against the exact interatomic contacts extracted from high-resolution complex structures in the eFindSite dataset, defined as pairs of heavy atoms within a distance of 4.5 Å. This cutoff is commonly used to determine the first hydration shell for proteins; when solvent molecules are present within this shell, proteins atoms have less freedom to interact with ligand atoms [92]. For each unique combination of a ligand atom type l and an amino acid effective point type p, we found an optional distance, D_{lp}^{cnt}, that reproduces high-resolution iteratomic contacts by maximizing the Matthews correlation coefficient (MCC) [93]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FN)(TN + FN)}} \tag{2.2}$$

Here, TP is the number of true positives, i.e. interatomic contacts that are correctly reproduced in the mixed-resolution model. TN is the number of true negatives, i.e. heavy atom pairs farther away than 4.5 Å from each other in high-resolution structures and also above

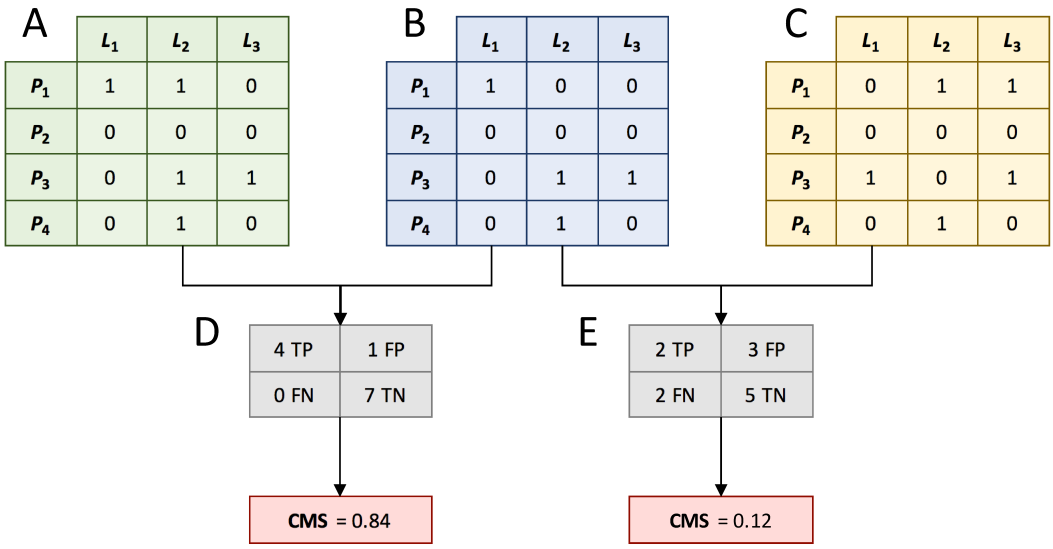

Figure 2.1: Calculation of the Contact Mode Score (CMS). First, intermolecular contacts calculated between ligand atoms L and protein effective points P are stored in binary matrices (1 - contact, 0 - no contact). Contact matrices for two arbitrary ligand conformations are shown in A and C, whereas B is a contact matrix constructed for the reference conformation. Next, a confusion table is computed for a pair of contact matrices; tables D and E are calculated for pairs A-B and C-B, respectively. Finally, CMS is calculated as the Matthews correlation coefficient for a given confusion table.

the corresponding type-dependent distance threshold for ligand atoms and protein effective points in the mixed-resolution model. FP and FN are the numbers of false positives and false negatives, respectively, i.e. those contacts that are over- and underestimated by using the mixed-resolution description. Note that ligand atoms in our model are treated equally when counting interatomic contacts. Although some methods prioritize certain parts of the ligand to better capture important aspects of binding [78], these approaches largely depend on manual inspection and thus cannot be automated.

2.2.5 Contact Mode Score

Essentially, the CMS quantifies the overlap of interatomic contacts in protein-ligand complex structures. Figure 2.1 illustrates a procedure to calculate the CMS for three conformations of a simplified system, in which the ligand has 3 heavy atoms ($L_1 - L_3$)

and the protein has 4 effective points ($P_1 - P_4$). The first step is to construct the Global Contact Matrix (GCM) encoding the interaction pattern for a particular ligand binding conformation (Figure 2.1A - 2.1C). Here, the distance between each ligand atom L of type l and each protein point P of type p is compared with the D_{lp}^{cnt} threshold to determine whether L and P are in contact. The corresponding entry in the GCM matrix is set to 1 if the distance is below D_{lp}^{cnt}, otherwise it is set to 0. Next, a confusion matrix is generated for a pair of GCMs, where one GCM represents a query (Figures 2.1A and 2.1C) and the other is the reference (Figure 2.1B). Confusion matrices consist of the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP are interatomic contacts that are present in both conformations and TN are pairs of ligand atoms and protein effective points not in contact in both conformations. FP and FN are over- and under-predicted contacts in the query conformation. Finally, Eq. 2.2 is used to calculate the CMS whose values range from -1 to 1, with greater values indicating a higher similarity between two conformations. Since relative distances between interacting points are used in CMS calculations, the resulting similarity score is independent of the absolute coordinate frames of query and reference structures. Furthermore, CMS correctly handles any degrees of freedom associated with the molecular flexibility, therefore, it can be applied to evaluate complex structures generated by ensemble docking and flexible receptor docking protocols.

2.2.6 eXtended Contact Mode Score

CMS requires a predefined one-to-one atomic correspondence, therefore, it can be used to measure the similarity of different conformations of the same protein-ligand pair. In order to compare non-identical complexes formed by different proteins and ligands, we developed the eXtended Contact Mode Score. In XCMS, equivalent atoms in two different ligand molecules are identified with the kcombu program [94]. Kcombu implements a fast and accurate build-up algorithm to perform chemical structure alignments and reports the similarity between ligands in terms of the topological Tanimoto coefficient (2D-TC).

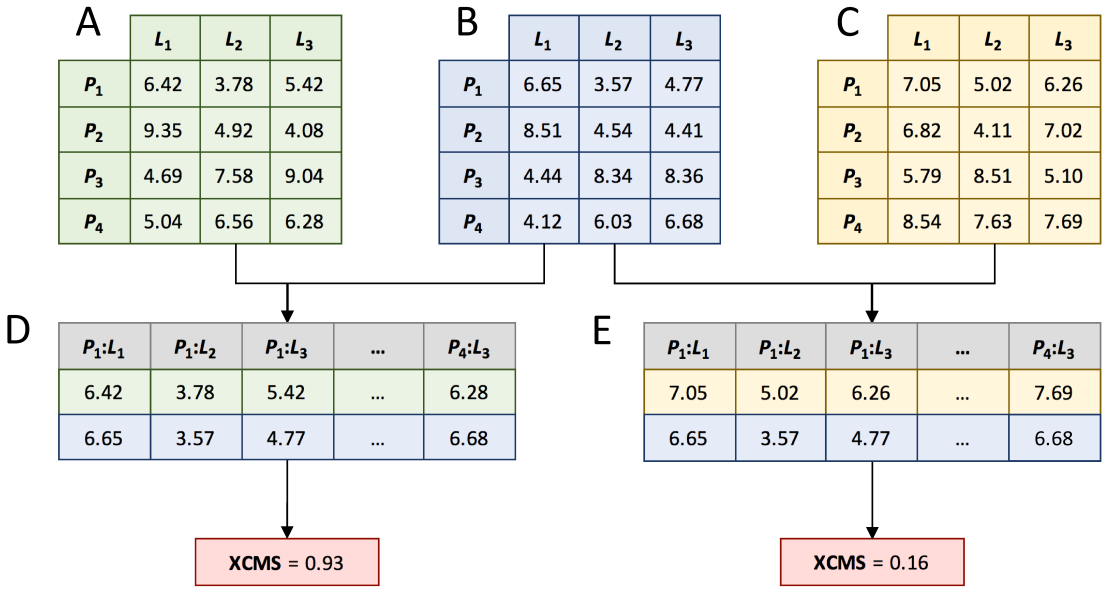

Figure 2.2: Calculation of the eXtended Contact Mode Score (XCMS). First, Cartesian distances calculated between ligand atoms L and protein effective points P are stored in distance matrices. Matrices for two arbitrary ligand conformations are shown in A and C, whereas B is a distance matrix for the reference conformation (distances are given in Å). Next, two matrices are converted to distance vectors whose elements correspond to pairs of protein effective points and ligand atoms ($P : L$). Finally, XCMS is computed as Spearmans rank correlation coefficient for a given set of vectors.

Further, the local structure alignment algorithm APoc [87] is employed to match ligand-binding pockets in a given pair of proteins in order to find equivalent residues. APoc uses the geometrical and physicochemical features of binding sites and provides a PS-score value, which measures the local similarity of ligand binding sites. Since equivalent residues reported by APoc for two proteins may have different types, we use a Cα-Cβ coarse-grained model in XCMS. Moreover, XCMS employs Local Contact Matrices ($LCMs$) because alignments generated by APoc are local, covering only ligand binding sites.

XCMS calculations are illustrated in Figure 2.2. Three non-identical complexes are shown in Figures 2.2A-2.2C. L_1-L_3 represent ligand heavy atoms matched by kcombu, so that an atom L_1 in the first complex is equivalent to L_1 atoms in the second and third complexes and so on. Protein residues are classified as ligand binding if any ligand atom

is found within a distance of 7 Å from any protein atom. This distance was selected to ensure that a sufficient number of binding residues are used for local alignments with APoc. Protein residues matched by APoc are stored in the LCM as rows arranged according to the pocket alignment. LCM entries are the distances between ligand atoms L and protein effective points P corresponding to the CA and CB atoms of binding residues. Next, LCMs are unrolled into 1D vectors maintaining the order of $P : L$ pairs (Figures 2.2D and 2.2E). The XCMS is then calculated as non-parametric Spearmans rank correlation coefficient between two vectors [95].

Similar to the CMS, XCMS ranges from -1 to 1 with higher values indicating a higher similarity between two conformations. However, in contrast to the CMS calculated from a 4×4 confusion matrix, XCMS depends on the length of distance vectors. Therefore, XCMS values are assigned a statistical significance under a null hypothesis that XCMS is zero for a pair of randomly generated LCMs; the alternative hypothesis is that two LCMs are significantly similar. The one-sided p-value is computed using the scipy package [96] based on the Fisher transformation method [97]. Given a positive XCMS, lower p-values indicate a higher statistical significance of the conformational similarity of protein-ligand complexes.

2.3 Results and discussion

2.3.1 Mixed-Resolution Contacts

Many all-atom models define interatomic contacts using a distance threshold of 4.5 Å corresponding to the second solvation shell [92]. In the mixed-resolution model used to calculate the CMS, type-dependent distance thresholds are optimized against the eFindSite dataset of protein-ligand complexes to reproduce all-atom contacts. Figure 2.3A shows the distribution of 720 (24 types of ligand atoms 30 types of protein effective points) contact distances, D_{lp}^{cnt}. The majority of contact distances fall within a range of 4-6 Å. Those effective points comprising more protein atoms, e.g. the side chains of Trp-2, Arg-2 and

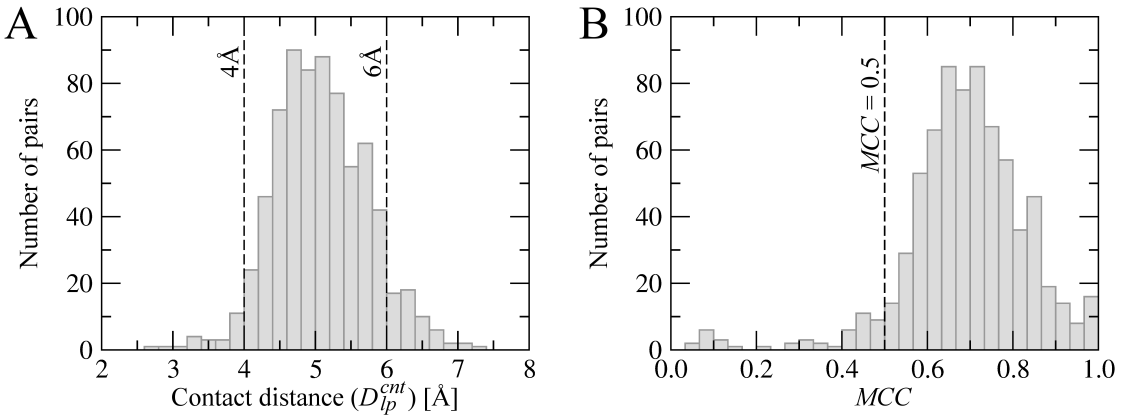

Figure 2.3: Parameterization of mixed-resolution intermolecular contacts. The distribution of (A) contact distance thresholds D_{lp}^{cnt} and (B) the Matthews correlation coefficient (MCC) values calculated vs. exact interatomic contacts across the eFindSite dataset.

Tyr-2, typically have larger D_{lp}^{cnt} thresholds than small amino acids, such as Ala-1, Ser-1 and Cys-1, as well as Cβ-Cγ virtual bonds and backbone CA and PP groups. In general, optimized distance thresholds in the mixed-resolution model reliably reproduce the exact interatomic contacts. As shown in Figure 2.3B, MCC values for most interacting pairs are larger than 0.5 with an average MCC of 0.7. Such accuracy in calculating intermolecular contacts in the mixed-resolution model is sufficient to develop a contact-based similarity measure.

2.3.2 Ligand Size Dependence of RMSD and CMS

The dependence of RMSD and CMS values on the ligand size was evaluated in a perturbation experiment. Table 2.1 shows the average RMSD and CMS after the first round of perturbation for Astex/CCDC complexes grouped based on the number of ligand heavy atoms. Both CMS and RMSD show some dependence on the ligand size because small ligands yield lower RMSD and higher CMS values compared with larger molecules. In Figure 2.4A, we plot similarity (CMS, light gray circles) and dissimilarity (RMSD, dark gray squares) values against the ligand size. The dependence of the (dis)similarity on the ligand size is assessed by the Pearson correlation coefficient (PCC) [98]. The PCC is 0.850

Table 2.1: Dependence of RMSD and CMS on the ligand size. Ligand conformations from the Astex/CCDC dataset were subjected to one round of perturbation comprising a set of forward translations and clockwise rotations. The mean values of RMSD and CMS are reported for each size range.

Ligand size	RMSD[Å]	CMS
6-17	0.527	0.879
18-28	0.851	0.793
29-39	0.961	0.757
40-50	1.334	0.666
51-62	1.541	0.625

for the RMSD and 0.780 for the CMS. In addition, we estimate the Mutual Information (MI) between the RMSD and CMS, and the ligand size. It has been shown that the MI can quantify the strength of a statistical association without bias for relationships of a specific form with higher MI value indicating a stronger association [99]. The MI against the ligand size is 0.714 for the RMSD and 0.512 for the CMS. Overall, the absolute values of PCC and MI are lower for CMS, indicating that it is less dependent on the ligand size than RMSD.

Next, we performed five rounds of perturbation of ligands in the Astex/CCDC dataset. Table 2.2 reports 25, 50 and 75 percentiles of RMSD and CMS as well as the quartile coefficient of dispersion (QCD) [100] after each perturbation round. The percentile values are also plotted in Figure 2.4B for the CMS and Figure 2.4C for the RMSD. Higher QCD values indicate larger fluctuations of a given measure. Although the QCD for the CMS increases with the number of perturbation rounds, it is systematically smaller than that for the RMSD demonstrating that the CMS is more stable.

2.3.3 Examples of CMS Calculations

The CMS is a convenient measure not only to assess docking accuracy, but also to analyze docking trajectories and the quality of scoring functions. On that account, we generated MMC trajectories for the Astex/CCDC dataset using GeauxDock [77] and calculated CMS values against the experimental structure for the accepted configurations. Two examples are shown in Figure 2.5, aspartyl proteinase penicillopepsin complexed with a pepstatin analogue (PDB-ID: 1apt, chain A, Figure 2.5A and 2.5B) [101] and and urokinase-type

Table 2.2: Changes in RMSD and CMS in the perturbation experiment. Ligand conformations from the Astex/CCDC dataset were subjected to multiple rounds of perturbation, each comprising a set of forward translations and clockwise rotations. 25, 50, and 75 percentiles as well as the quartile coefficient of dispersion (QCD) calculated across the dataset are reported for each perturbation round.

Perturbation round	CMS				RMSD			
	25%	50%	75%	QCD	25%	50%	75%	QCD
1	0.738	0.793	0.837	0.063	0.708	0.839	1.001	0.171
2	0.605	0.673	0.734	0.096	1.175	1.368	1.613	0.157
3	0.503	0.561	0.638	0.118	1.599	1.876	2.237	0.166
4	0.415	0.485	0.564	0.152	2.018	2.387	2.808	0.164
5	0.357	0.426	0.493	0.161	2.489	2.872	3.394	0.154

plasminogen activator complexed with an inhibitor (PDB-ID: 1c5x, chain B, Figure 2.5C and 2.5D) [102]. Figure 2.5A and 2.5C show that at the beginning of docking simulations, pseudo-energies are high and CMS values are low suggesting that initial ligands are far away from experimental binding poses. Blue lines in both plots show that MMC simulations in GeauxDock are driven by the pseudo-energy to reach low-energy states. Encouragingly, the CMS increases as the pseudo-energy gradually decreases indicating that ligands are moving toward native-like conformations. This correlation between the pseudo-energy and the native-likeness is a desired characteristic of a scoring function, which is shown as scatter plots in Figure 2.5B and 2.5D. It is noteworthy that our previous benchmarks of GeauxDock demonstrated that the pseudo-energy and CMS are correlated for about three-quarters of Astex/CCDC complexes [77].

Three representative snapshots selected from each docking trajectory are shown in Figure 2.6. These binding poses shown in blue were generated at the beginning (Figures 2.6A and 2.6D), in the middle (Figures 2.6B and 2.6E), and at the end (Figures 2.6C and 2.6F) of GeauxDock simulations. The corresponding CMS values calculated against experimental complex structures shown in orange are 0.286, 0.366 and 0.601 for penicillopepsin, and 0.424, 0.583 and 0.771 for plasminogen activator, respectively. It is clear that high CMS values correspond to docking conformations that are close to experimental structures, thus the CMS is a good indicator of the native-likeness.

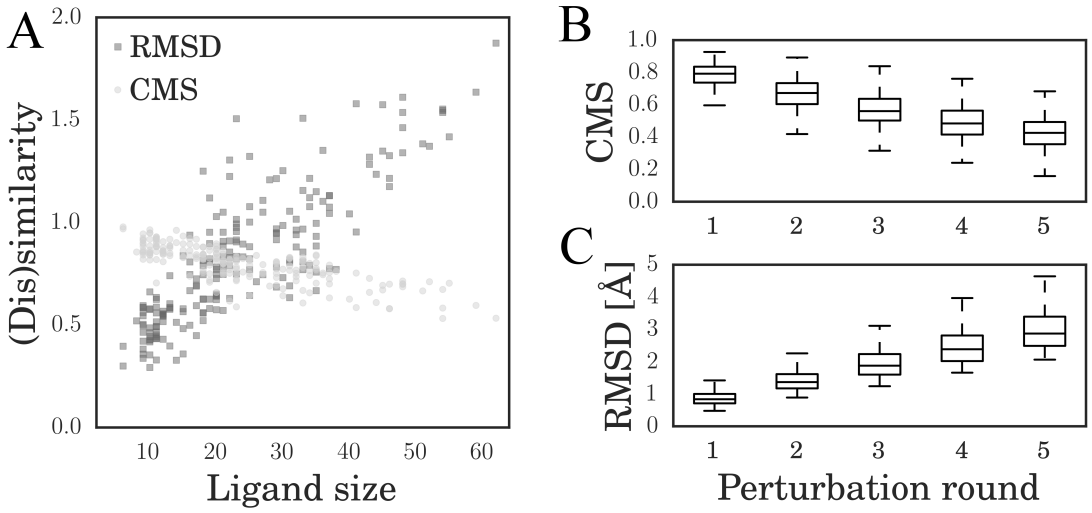

Figure 2.4: Comparison of RMSD and CMS in the perturbation experiment. (A) Scatter plot of RMSD (dark gray squares) and CMS (light gray circles) vs. the number of ligand atoms after a single perturbation round. Boxplots of (B) CMS and (C) RMSD calculated for ligand conformations generated through multiple perturbation rounds. Boxes end at the 25 and 75 percentiles, a horizontal line in a box is the 50 percentile (median).

2.3.4 Algorithm Complexity of CMS and RMSD

We compare the time to calculate CMS and RMSD using the Astex/CCDC dataset. Specifically, for each complex, CMS and RMSD values for 8 variational conformations were calculated against the experimental structure, resulting in 1 632 (2048) individual calculations. Using one thread on a 2.6 GHz Sandy Bridge Xeon 64-bit processor, the wall time to finish RMSD (CMS) calculations is 17 s (5 389 s), thus computing RMSD is about 317 times faster than CMS. The reason for a longer wall time required to calculate CMS is that it considers a protein environment and iterates over all pairs of ligand atoms and protein points, whereas the RMSD iterates only over ligand atoms. From the perspective of algorithm complexity, the CMS calculation is $O(P \times L)$ and the RMSD calculation is $O(L)$, where P and L are the total number of protein points and ligand atoms, respectively. Although both RMSD and CMS calculations are based on Euclidean distances,

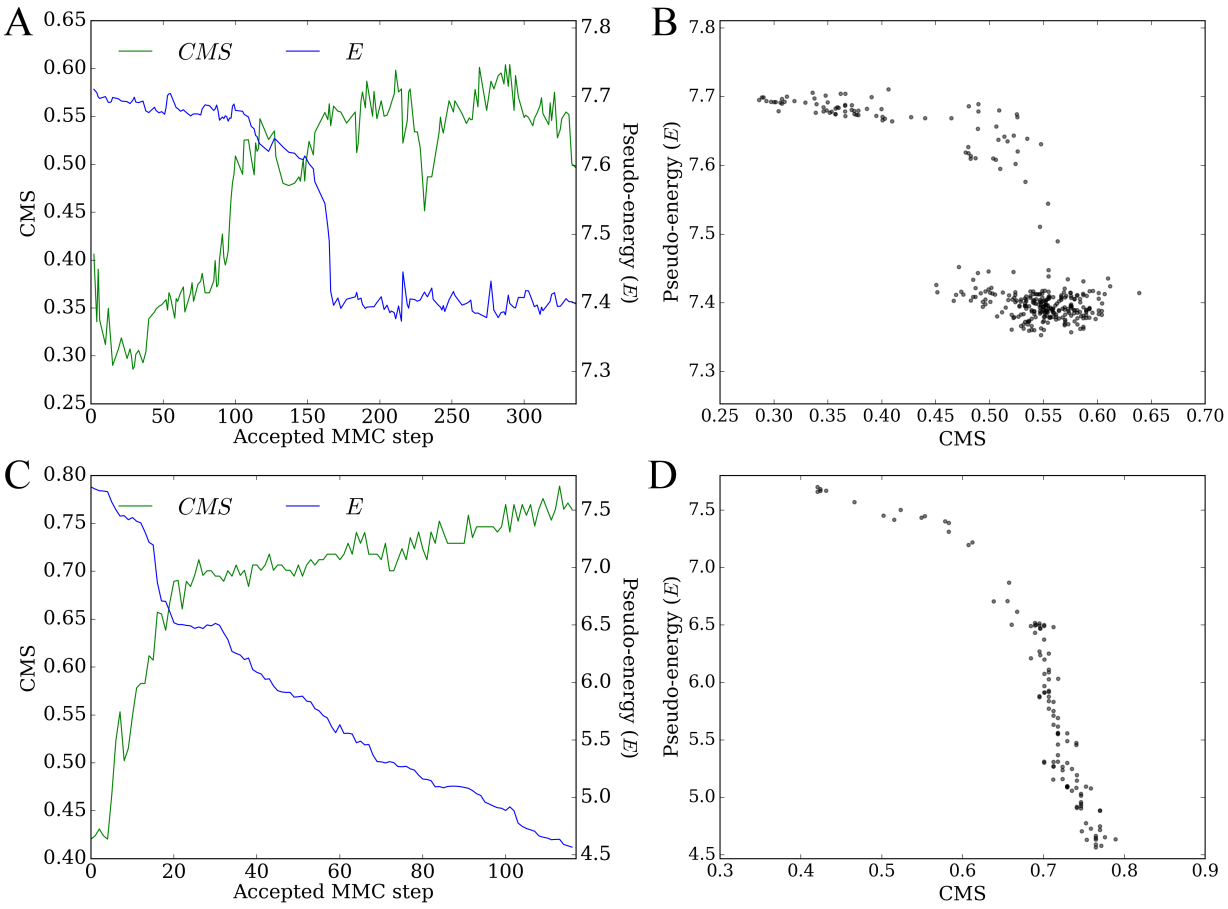

Figure 2.5: Analysis of docking trajectories with the CMS. Docking simulations were conducted using GeauxDock for (A, B) penicillopepsin/pepstatin analogue (PDB-ID: 1apt, chain A) and (C, D) plasminogen activator/inhibitor (PDB-ID: 1c5x, chain B). (A, C) Metropolis Monte Carlo trajectories for CMS (green) and pseudo-energy (E, blue). (B, D) Scatter plots of CMS vs. the pseudo-energy; each dot represents an accepted protein-ligand conformation.

CMS requires a longer computing time due to the relatively large number of 838 effective points per protein on average.

2.3.5 Dependence of XCMS on the Ligand and Pocket Similarity

XCMS was developed as an extension of the CMS to measure the similarity of ligand binding conformations among complexes formed by different proteins and ligands. In order to establish when a similar ligand binding conformation can be expected, we investigate the dependence of XCMS on the pocket and ligand similarity in experimental complex

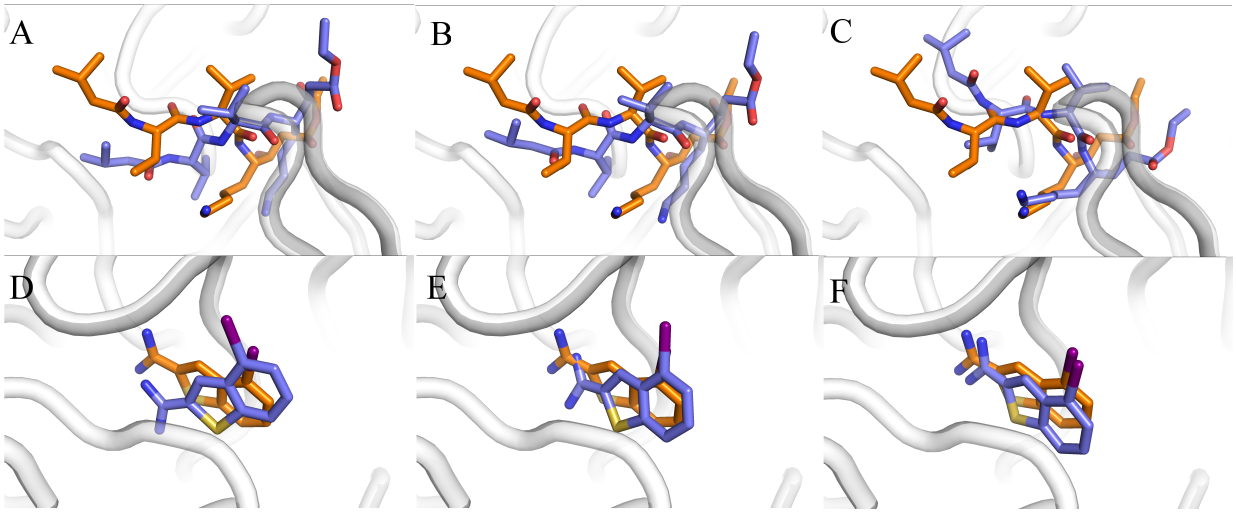

Figure 2.6: Examples of docking poses from GeauxDock simulations. (A-C) penicillopepsin/pepstatin analogue (PDB-ID: 1apt, chain A) and (D-F) plasminogen activator/inhibitor (PDB-ID: 1c5x, chain B). Three docking poses are shown in blue for each system, (A, D) initial, (B, E) intermediate, and (C, F) final conformations. The corresponding experimental complex structures are colored in orange.

structures. Specifically, XCMS, PS-score and 2D-TC values were calculated for all query-template pairs across the BioLiP database. Heat maps in Figure 7 were constructed by dividing query-template pairs into 400 groups based on 2D-TC and PS-score values and then averaging XCMS and p-value within each group. Note that those pairs having a PS-score between the query and the template of >0.9 were excluded in order to examine only non-identical systems. As expected, Figure 2.7A demonstrates that the conformational similarity of protein-ligand complexes captured by XCMS increases as their pockets and binding ligands become more similar. Figure 2.7B shows the statistical significance of query-template XCMS as a function of PS-score and 2D-TC. The significance of XCMS increases with the increasing similarity of ligands and binding pockets in query and template structures. A clear boundary in Figures 2.7A and 2.7B at a PS-score of 0.4 corresponds to a threshold separating statistically similar and dissimilar binding pockets in proteins [87]. Overall, these results corroborate previous studies reporting the conservation of ligand binding across structurally similar pockets occupied by chemically similar ligands [76, 103–

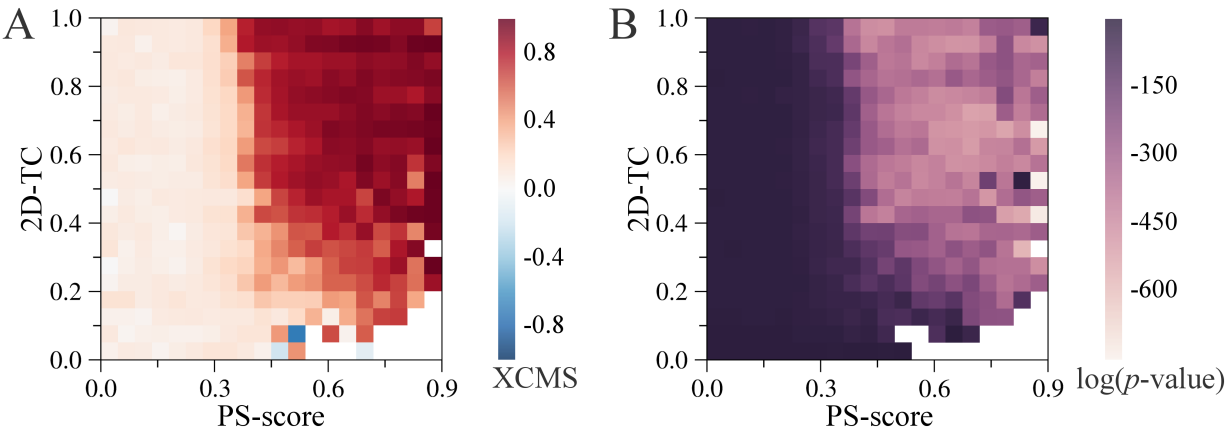

Figure 2.7: XCMS and its statistical significance for the BioLiP dataset. Query-template pairs are grouped based on the similarity between their ligands (measured by the 2D Tanimoto coefficient) and pockets (measured by PS-score). Heat maps of (A) the arithmetic mean values of XCMS and (B) the geometric mean of the p-value for positive XCMS.

105]. It is important to note that both pocket similarity and ligand similarity should be taken into account when selecting a template to calculate XCMS. In practice, we first rank templates by the product of 2D-TC and PS-score and then take the top-ranked structure to assess the target conformation using XCMS.

2.3.6 Large-Scale Benchmarking of Molecular Docking

Molecular docking with AutoDock Vina was performed for a subset of 2,200 query complexes selected the BioLiP dataset. In Figure 2.8, we first use this simulated dataset to investigate the relationship between RMSD, CMS and XCMS. Here, the strength of association is measured with the maximal information coefficient (MIC) [106]. The MIC belongs to the maximal information-based nonparametric exploration class of statistics and quantifies linear and non-linear associations by applying mutual information to continuous random variables. Figure 2.8A shows the correlation between CMS and RMSD, both of which are calculated against the experimental structures of query complexes; the MIC between the CMS and RMSD is as high as 0.91. Figure 2.8B shows the correlation between CMS and XCMS, where the XMCS is calculated using template structures. Encouragingly, these two contact-based measures are also highly correlated with a MIC of 0.88. Both MIC

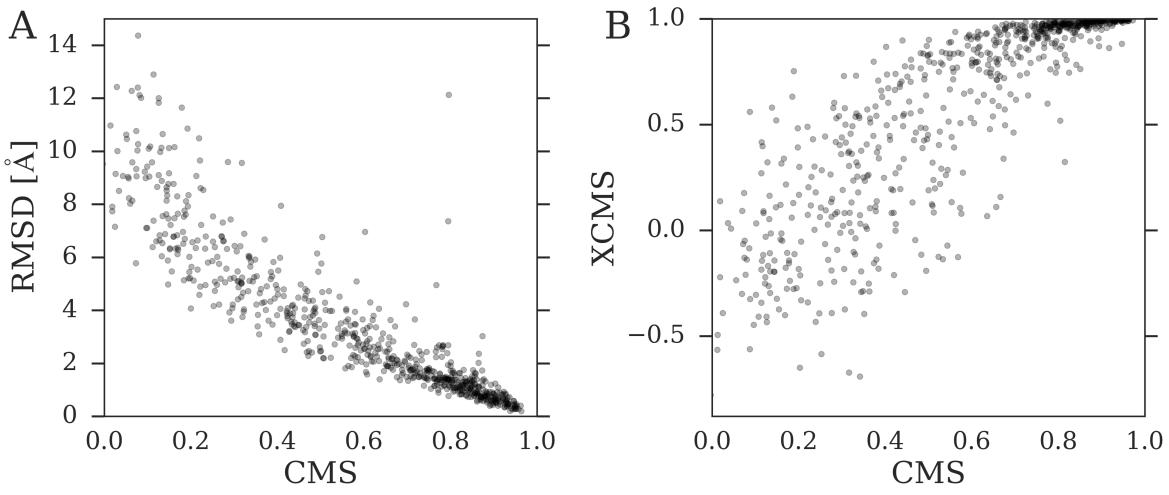

Figure 2.8: Correlation between RMSD, CMS, and XCMS. Docking conformations generated for the BioLiP dataset by AutoDock Vina are used to calculate RMSD and CMS against experimental binding poses. XCMS was computed against a holo template selected from the BioLiP database based on the highest value of the product of PS-score and the 2D Tanimoto coefficient. Scatter plots of (A) CMS vs. RMSD and (B) CMS vs. XCMS.

values are statistically significant at p-value of $< 1.28\times10^{-6}$ [106] demonstrating a strong association between RMSD, CMS and XCMS.

Next, we use the RMSD, CMS and XCMS to evaluate the accuracy of molecular docking for the BioLiP dataset. In Figure 2.9 and Table 2.3, docking poses generated by AutoDock Vina are compared to random ligand conformations generated within receptor binding pockets. Regardless of the evaluation metric, Vina constructed native-like conformations for a significant number of complexes, whereas the vast majority of random conformations are far away from experimental structures. For instance, the median (50% quartile) RMSD, CMS, and XCMS for Vina is 2.89 Å, 0.574, and 0.694, respectively, compared to 7.60 Å, 0.152, and 0.198 for random conformations. Overall, these results demonstrate that when suitable templates can be identified in the BioLiP database, a retrospective assessment with RMSD and CMS against experimental structures can be replaced with a template-based evaluation using the XCMS.

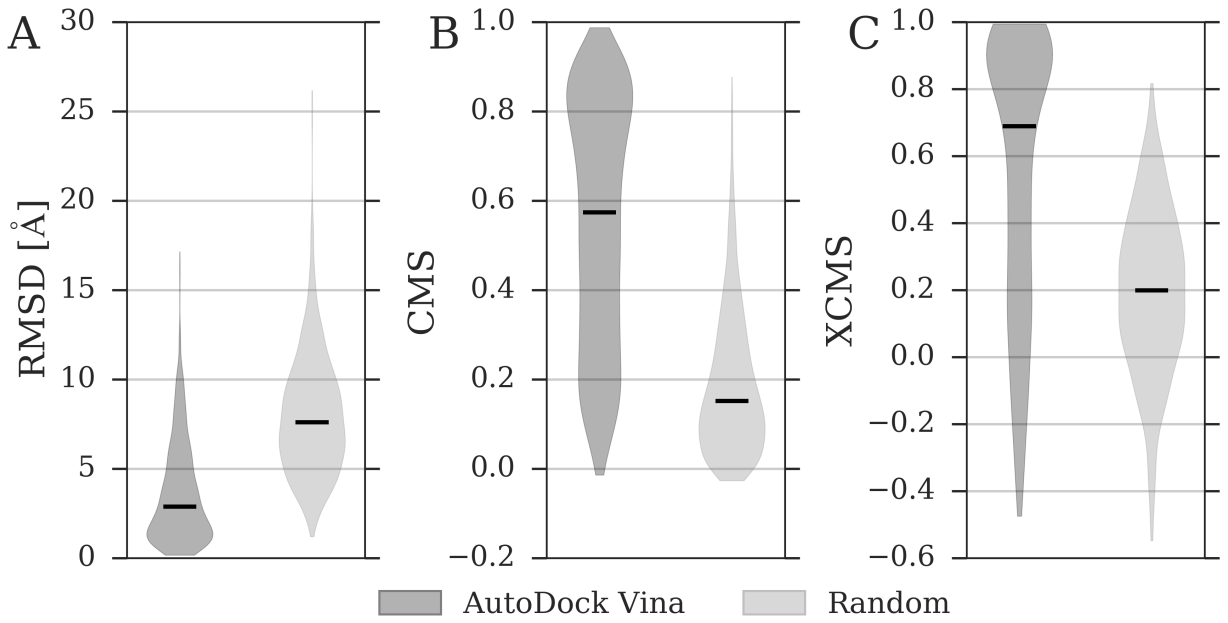

Figure 2.9: Assessment of docked and randomized ligand conformations across the BioLiP dataset. The similarity to experimental binding poses is assessed with (A) RMSD, (B) CMS, and (C) XCMS. RMSD and CMS were calculated against experimental complex structures. XCMS was calculated against a holo template selected from the BioLiP database based on the highest value of the product of PS-score and the 2D Tanimoto coefficient. Dark gray violins correspond to ligands docked by AutoDock Vina, whereas light gray violins are calculated for randomized ligand conformations. Black horizontal lines are median values.

2.3.7 Examples of XCMS Calculations

Finally, we discuss two representative examples illustrating how XCMS can be used to evaluate docking conformations, mitogen-activated protein kinase 14 (MAPK14, PDB-ID: 2yiw, ligand: YIW, chain: A) [107] and ribose-5-phosphate isomerase (RpiA, PDB-ID: 1o8b, ligand: ABF, chain A) [108]. Both query ligands (YIW and ABF) were docked into their target binding pockets by AutoDock Vina [72] starting from random conformations. We first calculated the RMSD and CMS against native complexes to evaluate the docking accuracy. Table 2.4 shows that docking simulations were successful in both cases and the predicted conformations are highly similar to experimental structures; for instance, the RMSD is 0.42 Å and the CMS is 0.94 for MAPK14. Next, we evaluate docking con-

Table 2.3: Assessment of docked and randomized ligand conformations across the BioLiP dataset. RMSD and CMS were calculated against experimental complex structures. XCMS was calculated against a holo template selected from the BioLiP database based on the highest value of the product of PS-score and 2D Tanimoto coefficient. Mean values as well as 25%, 50% and 75% quartiles are reported.

Statistic	AutoDock Vina			Random		
	RMSD[Å]	CMS	XCMS	RMSD[Å]	CMS	XCMS
mean	3.66	0.548	0.545	8.03	0.191	0.194
25%	1.4	0.308	0.203	5.49	0.07	0.036
50%	2.89	0.574	0.694	7.6	0.152	0.198
75%	5.29	0.798	0.912	10.02	0.279	0.366

formations with the XCMS. Proto-oncogene tyrosine-protein kinase Src (c-Src, PDB-ID: 3f3u, ligand: 1AW, chain A) [109] was selected from the BioLiP database as a template for MAPK14, whereas central glycolytic gene regulator (CggR, PDB-ID: 3bxh, ligand: F6P, chain A) [110] was selected as a template for RpiA. XCMS values calculated against template complexes reported in Table 2.4 demonstrate that the template-based assessment is consistent with the direct evaluation using CMS and RMSD; for instance, the XCMS is 0.96 with a highly significant p-value of close to 0 for MAPK14.

Table 2.4 also includes various similarity scores for query-template pairs as well as their functional classification. MAPK14 and c-Src belong to the same class of transferase enzymes transferring phosphorus-containing groups (Enzyme Commission, EC number 2.7.-.-) and have globally similar structures with a Template Modeling score (TM-score) of 0.76. TM-score is a length-independent measure of the structural similarity between proteins [111]; it ranges from 0 to 1, with values 0.4 and higher indicating a statistically significant similarity. In contrast, RpiA and CggR have unrelated structures with a TM-score of 0.27. RpiA is an enzyme, ribose-5-phosphate isomerase (EC number 5.3.1.6), whereas non-enzyme CggR belongs to the SorC/DeoR family of prokaryotic transcriptional regulators. In both cases, template-bound ligands are similar to query ligands with a 2D-TC of 0.41 for MAPK14/c-Src and 0.88 for RpiA/CggR. In order to visually compare ligand binding conformations, global and local structure alignments constructed for MAPK14/c-Src and

Table 2.4: Assessment of ligand binding poses docked by AutoDock Vina. Two case studies are presented, MAPK14 complexed with triazolopyridine inhibitor (PDB-ID: 2yiw, ligand YIW, chain A) and ribose-5-phosphate isomerase complexed with the inhibitor arabinose-5-phosphate (PDB-ID: 1o8b, ligand ABF, chain A).

Metric/info	Case study	
	2YIW_YIW_A	1O8B_ABF_A
Calculated against experimental complex structure		
RMSD[Å]	0.42	1.58
CMS	0.94	0.77
Template-based assessment		
Template	3F3U_1AW_A	3BXH_F6P_A
TM-scorea	0.76	0.27
PS-scoreb	0.7	0.46
p-value of PS-score	6.28E-09	4.90E-05
2D-TCc	0.41	0.88
Query EC#	2.7.11.24	5.3.1.6
Template EC#	2.7.10.2	Non-enzyme
XCMS	0.96	0.76
p-value of XCMS	0	1.56E-63

RpiA/CggR are shown in Figure 10. Ligands bound to MAPK14 and c-Src adopt a similar conformation when protein structures are superposed according to the global alignment by Fr-TM-align [112] (Figure 2.10A) and the local alignment by Apoc [87] (Figure 2.10B). Since the global structure alignment between RpiA and CggR is random, it cannot be used to provide equivalent residues for XCMS calculations (Figure 2.10C). Nonetheless, APoc constructed a statistically significant local alignment of binding pockets in RpiA and CggR with a PS-score of 0.46 and the corresponding p-value of 4.9×10^{-5}. When protein structures are superposed according to the local alignment, binding ligands in RpiA and CggR adopt a similar conformation (Figure 2.10D). These examples demonstrate that although XCMS calculations do not require globally similar templates, the chemical similarity of bound ligands as well as the similarity of binding sites in query and template structures should be high enough to ensure a meaningful template-based assessment.

2.4 Conclusions

The Contact Mode Score, or CMS, was developed in this study to quantify the conformational similarity of protein-ligand complexes based on intermolecular contacts. Its major advantages over the traditional root-mean-square deviation include less dependency on the ligand size and taking into account the protein environment. Consequently, the CMS can be used to measure the ligand binding similarity across diverse protein-ligand datasets as well as to evaluate flexible docking methods simulating receptor conformational changes upon ligand binding. In order to effectively compare binding poses of non-identical ligands bound to different proteins, we further developed the eXtended Contact Mode Score, or XCMS. The XCMS capitalizes on the conservation of ligand binding across structurally similar pockets occupied by chemically similar ligands. For instance, it can be used to systematically evaluate complex structures constructed by virtual screening, where a retrospective assessment cannot be performed because the experimental structures of the majority of complexes are unavailable. CMS and XCMS are freely available at `http://brylinski.cct.lsu.edu/content/contact-mode-score` and `http://geaux-computational-bio.github.io/contact-mode-score/`.

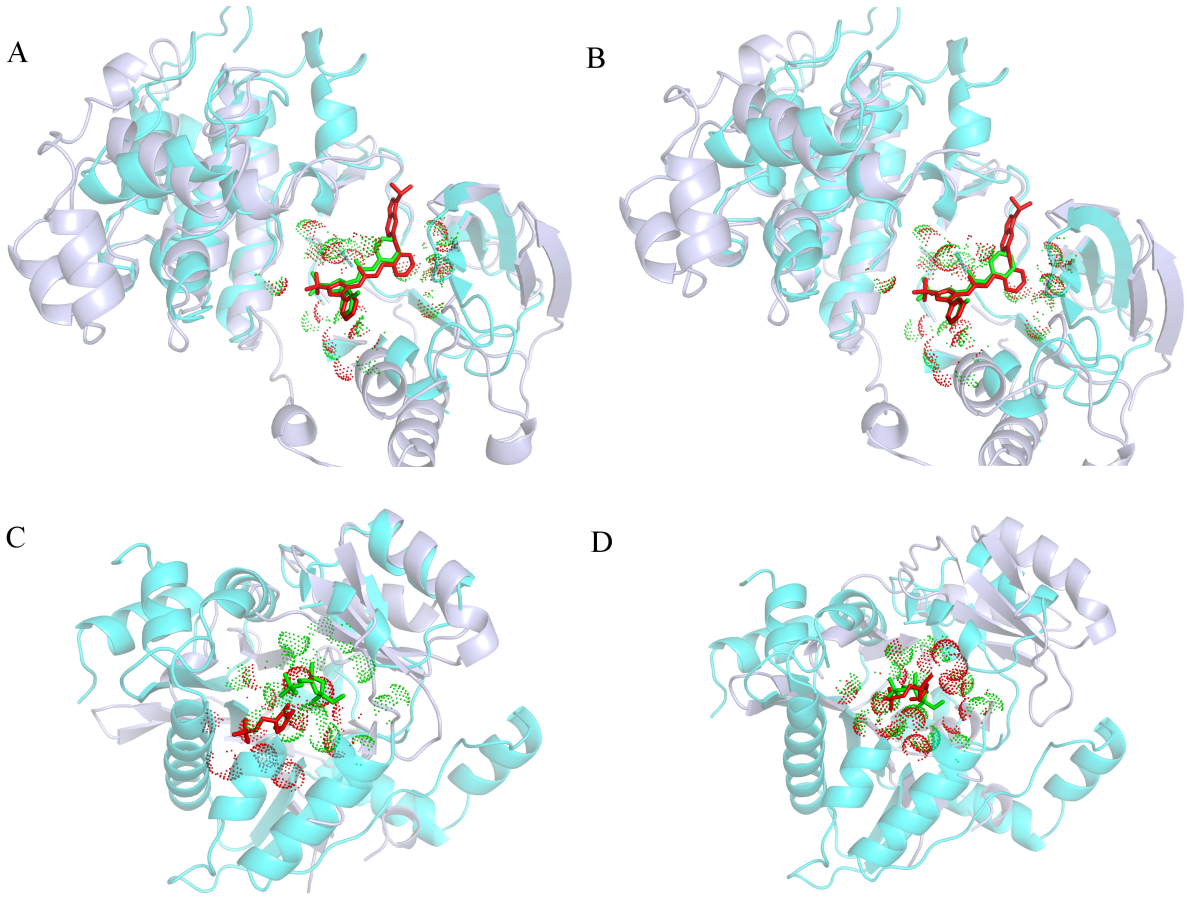

Figure 2.10: Examples of the superposition of query and template structures. The query protein is ice blue with its binding residues marked by red dots and the bound ligand shown as red sticks. The template protein is cyan with its binding residues marked by green dots and the bound ligand shown as green sticks. (A, B) The superposition of MAPK14 (PDB-ID: 2yiw, chain A) and c-Src (PDB-ID: 3f3u, chain A). (C, D) The superposition of ribose-5-phosphate isomerase (PDB-ID: 1o8b, chain A) and central glycolytic gene regulator (PDB-ID: 3bxh, chain A). For each pair, two superpositions are shown, (A, C) the global structure alignment by Fr-TM-align and (B, D) the local pocket alignment by APoc.

Chapter 3
GEAUXDOCK ENGINE

3.1 Introduction

Computational identification of potential leads against a specific protein target is of paramount importance to modern drug design. As of April 2015, the ZINC database of commercially available small molecule entities for drug discovery contains 17,900,742 drug-like compounds collected from the catalogs of 236 vendors [49]. At the outset of drug development, this vast number of compounds must be downsized to typically hundreds to thousands of the most promising candidate molecules. High-throughput screening (HTS) often adopted by the pharmaceutical industry is a conventional approach for lead identification, however, it suffers from high costs and low hit rates. Conversely, computational methods such as virtual screening (VS) provide faster and cheaper alternatives to HTS with many successful examples described in the literature [113–115]. Current VS techniques fall into two main categories: ligand-based similarity searching and structure-based molecular docking [56]. Although the experimentally solved structures of target proteins are not required in the ligand-based approach, an initial set of already developed compounds must be known. However, this information is often unavailable, particularly for novel molecular targets. In contrast, the advances in X-ray crystallography and nuclear magnetic resonance result in the accumulation of atomic-level structures of biological molecules fostering docking-based drug discovery projects [57, 58].

A typical molecular docking program incorporates two important components, the prediction of the binding mode of a drug candidate within the target pocket and the estimation of binding affinity from molecular interactions. Most currently available docking approaches implement effective algorithms to predict near-native binding modes [116–119], however, noticeable differences still exist when compared with the experimental data. For instance, a recent study evaluated seven popular docking programs on a dataset of 1300

complexes showing a wide range of the average root-mean-square-deviation (RMSD) values from 2.7 Å to 4.5 Å [120]. In addition to binding mode prediction, a scoring function is another pivotal component of molecular docking that guides the exploration of the conformational space and estimates the binding affinity for putative binding modes. Many scoring functions developed to date [121–126] can be broadly categorized into three classes, force field-based, empirical, and knowledge-based [127–129]. Recently, Liu and Wang proposed a new type of scoring function called descriptor-based or machine learning-based to capture the new trend in this field [130]. Methods using descriptor-based scoring functions encode the properties of ligands and proteins as well as protein-ligand interactions into sets of descriptors followed by applying machine learning to compute protein-ligand binding scores [130]. Notwithstanding the progress in the development of scoring functions for ligand docking, several comparative studies reported that no single algorithm systematically outperforms other methods across all protein targets [116, 131, 132].

In general, high-resolution protein structures are required for satisfactory results from molecular docking regardless of which scoring function is used [133]. Additionally, the prediction success rate drops from the ligand-bound to ligand-free conformational state of a target protein [134]. This is due to the fact that many proteins undergo structural changes in functionally relevant regions on ligand binding [135]. It has been demonstrated that even minor changes affect the docking accuracy; for example, the mean protein rearrangement greater than 1.5 Å may cause a loss of 90% of the initial docking accuracy [136]. Although high-resolution structures are usually preferred in docking simulations, these may not be available in the near future for many pharmacologically important drug targets such as membrane spanning G-protein coupled receptors and ion channels [137]. Conversely, Skolnick et al. pointed out that high-resolution structures may actually conceal the inherent structural plasticity of ligand binding regions [138]. For instance, the structural variation of a highly conserved ATP-binding site is about 2.4 Å, as measured over a subset of inhibitor-bound crystal structures of protein kinases [139]. To address this issue, a recently

developed ligand homology modeling (LHM) approach [] integrates structural information extracted from evolutionarily related proteins into the modeling of protein-ligand interactions to improve the tolerance to distortions in target binding sites. LHM was one of the first approaches to successfully incorporate evolutionary information in ligand docking and VS [76]. $Q-Dock^{LHM}$ further exploited the ideas of LHM by implementing a descriptor-based scoring function. Nevertheless, an open questions is how evolutionary descriptors supplement physics-based components in a force field that combines these two classes of scoring terms.

In this study, we describe the development and benchmarking of GeauxDock, a new approach for ligand molecular docking. GeauxDock uses a descriptor-based scoring function integrating evolutionary constraints with statistical potentials and physics-based energy terms. Moreover, it features a mixed-resolution molecular representation of protein-ligand complex structures at the level of ligand heavy atoms and pro- tein effective points. A Monte Carlo protocol is used to efficiently sample the conformational space with the flexibility of ligand and receptor molecules modeled using an ensemble- based approach. The scoring function in GeauxDock was parameterized on a large dataset of protein-ligand complexes and further optimized to produce a correlation between the total pseudoenergy and the native-likeness of binding poses. Finally, we carry out an analysis of the contribution of various scoring terms to the identification of final docking conformations. We demonstrate that although evolutionary constraints generally improve the docking accuracy, the scarcity of this information can be effectively compensated by increasing the contribution from physics-based energy components.

3.2 Materials and methods

3.2.1 Datasets

Two datasets of protein-ligand complexes are used in this study. The first set was compiled from the *e*FinedSite library [83] by clustering template proteins at 40% sequence

identity using PISCES [84] and then selecting representative chains that noncovalently bind small organic molecules at distinct locations. This procedure resulted in 14,059 nonredundant structures of protein-ligand complexes, referred to as the *e*FindSite protein Data Bank (PDB) [140] dataset, that are used to derive potentials and parameters for the docking force field. The second dataset comprises 201 high-quality crystal structures taken from the Astex/CCDC collection of pharmacologically relevant drug targets complexed with ligand molecules [85]. As our force field includes potentials calculated from evolutionarily related binding pockets, we selected those proteins for which *e*FindSite predicted the binding site within a distance of 8 Å from the geometric center of a ligand in the experimental complex structure. *e*FindSite is a threading/structure-based method that detects conserved binding sites accross sets of homologous proteins [83]. For each target, we ran *e*FindSite at two different thresholds for the maximum target-template sequence identity, 80 and 40%. The first protocal uses both close and remote homologs to detect functional sites, whereas the second uses only those templates that are evolutionarily weakly related to the target. The Astex/CCDC dataset is used for the force field optimization and benchmarking.

3.2.2 Molecular representation of complex structures

GeauxDock uses the same molecular representation as in the section 2.2.3.

3.2.3 Force field for molecular docking

Protein-ligand complexes are stabilized by a variety of molecular interactions. Here, we developed a new descriptor-based force field for the modeling of protein-ligand interactions that combines classical physics-based potentials with statistical and knowledge-based scoring terms. Specifically, we include the following nine energy terms: (i) electrostatic and (ii) van der Waals interactions, (iii) hydrogen bonds, (iv) hydrophobic interactions, (v) generic and (vi) pocket-specific contact potentials, (vii) a pseudopharmacophore potential, and position restraints on (viii) family conserved anchor substructures, and (ix) the binding site center.

Electronic and van der Waals interactions (i, ii). Because of the mixed-resolution model, we use soft electrostatic, P_{ele}^{soft}, and soft Lennard-Jones, P_{vdW}^{soft}, potentials [141]. Electrostatic interactions are described by:

$$P_{ele}^{soft}(l,p) = q_l q_p g(r_{lp}) \tag{3.1}$$

Let r_{lp} be the distance between the l^{th} ligand atom and the p^{th} protein effective point with the corresponding partial charges q_l and q_p. Then $g(r_{lp}) = 1/R_{lp}$ for $R_{lp} \geq 1$, and $g(r_{lp}) = k + aR_{lp}^2 + bR_{lp}^3$ for $R_{lp} < 1$, where $R_{lp} = sr_{lp}, a = 4 - 3k$ and $b = 2k - 3$. k is an adjustable parameter that controls the value of the electrostatic potential at zero separation and it is set to 2.0, and s is a scaling factor set to 0.5. Partial charges on ligand atoms are calculated using the Mulliken population analysis [142] implemented in Open Babel [88], whereas those on proten effective points are assigned by adding partial charges from the constituent atoms according to the Assisted Model Building with Energy Refinement (AMBER) ff03ua force field [143].

The electrostatic interaction score, E_{ele}^{soft}, is a sum of P_{cle}^{soft} values taken over $L \times P$ pairs of ligand atoms and protein effective points normalized by the total number of ligand atoms, L:

$$E_{ele}^{soft} = \frac{1}{L} \sum_{l}^{L} \sum_{p}^{P} P_{ele}^{soft}(l,p) \tag{3.2}$$

Van der Waals interactions are modeled using the following form of a soft Lennard-Jones potential:

$$P_{vdW}^{soft}(l,p) = \frac{(2\varepsilon_{lp} r_{lp}^{*9}/r_{lp}^{9}) - (3\varepsilon_{lp} r_{lp}^{*6}/r_{lp}^{6})}{(2\varepsilon_{lp} r_{lp}^{*9}/r_{lp}^{9})\alpha(1 + \beta r_{lp}^{2}) + 1} \tag{3.3}$$

where r_{lp}^{*} depends on both a ligand atom type and the amino acid effective point and it is defined as $r_{lp}^{*} = \kappa D_{lp}^{cnt}$. ε is the depth of the potential well, and r_{lp} is the distance between the l^{th} ligand atom and the p^{th} protein point. The parameter α controls the value of the

function at $r_{lp} = 0$, and the parameter β controls the rate at which the function approaches the maximum value at zero separation.

Type-dependent parameter ε are derived from the *e*FindSite/PDB dataset as follows:

$$\varepsilon_{lp} = ln(1 + \frac{n_{lp}}{n_{lp}^0}) \tag{3.4}$$

where n_{lp} is the observed number of contacts between a given pair of a ligand atom type and the amino acid effective point, and n_{lp}^0 is an expected number of contacts assuming no specificity. The latter is defined as $n_{lp}^0 = N\chi_l\chi_p$, with the total number of N protein-ligand contacts, and χ_l and χ_p corresponding to the mole fractions of ligand atoms of type l and protein points of type p, respectively.

Parameters α, β and κ are optimized empirically on the *e*FindSite/PDB dataset by minimizing the following $Z - score$ function:

$$Z_{vdW} = \sum_{lp}^{D} \frac{P_{vdW}^{nat}(l,p) - \left\langle P_{vdW}^{dec}(l,p) \right\rangle}{\delta} \tag{3.5}$$

where the summation runs over D pairs of ligand atoms and protein points that are in contact according to the mixed-resolution models of dataset complexes, P_{vdW}^{nat} is the value of the soft Lennard-Johns potential, $P^{soft}vdW$, for a given pair of the l^{th} ligand atom and the p^{th} protein point. $< P_{vdW}^{dec}(l,p) >$ is the value of P_{vdW}^{soft} averaged over a set of 10 "decoy" distances r_{lp} randomly generated around the interaction threshold D_{lp}^{cnt}, and δ is the corresponding standard deviation. The optimal values of $\alpha = 0.88$, $\beta = 0.74$, and $\kappa = 0.70$ were found using the evolutionary search strategy [144].

For a given protein-ligand complex, the van der Waals interaction score, E_{vdW}^{soft}, is calculated by summing P_{vdW}^{soft} values over all ligand atoms and protein effective points, and

then normalizing the sum by the total number of ligand atoms L:

$$E_{vdW}^{soft} = \frac{1}{L}\sum_{l}^{L}\sum_{p}^{P} P_{vdW}^{soft}(l,p) \tag{3.6}$$

Hydrogen bonds (iii). The hydrogen bond potential, P_{HB}, only applies to those atom pairs that can form hydrogen bonds and it is modeled using single Gaussian restraints:

$$P_{HB}(l,p) = -exp\left\{-0.5\left(\frac{r_{lp}^{HB}-\mu_{lp}^{HB}}{\sigma_{lp}^{HB}}\right)^2\right\}\frac{1}{\sqrt{2\pi\sigma_{lp}^{HB}}} \tag{3.7}$$

where r_{lp}^{HB} is the distance between the l^{th} ligand atom and the p^{th} protein effective point, and μ_{lp}^{HB} is the average hydrogen bond length between ligand atoms of the same type as l and protein points of the same type as p across the eFindSite/PDB dataset, with the corresponding standard deviation σ_{lp}^{HB}.

For a given protein-ligand complex, its hydrogen bond score is calculated by summing P_{HB} over those pairs of ligand atoms and protein effective points that can form hydrogen bonds, and then averaging by the total number of ligand atoms L:

$$E_{HB} = \frac{1}{L}\sum_{l}^{L}\sum_{p}^{P}\begin{cases} P_{HB}(l,p), \textit{if (l,p) can form a hydrogen bond} \\ 0, else \end{cases} \tag{3.8}$$

Hydrophobic interactions (iv). Hydrophobic interactions between ligand atoms and protein effective points are modeled using a spatial hydrophobicity distribution and softened Gaussian restraints. First, we calculate an empirical hydrophobicity, $P_{HP}(l)$, at the position of a ligand atom l resulting from the surrounding P protein side chains within a distance of r_{max} usinga simple sigmoid function [145]:

$$P_{HP}(l) = \sum_{p}^{P}\begin{cases} \widetilde{H}_p\left[1-\frac{1}{2}(7k_{lp}^2-9k_{lp}^4+5k_{lp}^6-k_{lp}^8)\right], \textit{if } r_{lp} \le r_{max} \\ 0, \ else \end{cases} \tag{3.9}$$

where r_{lp} is the distance between the l^{th} ligand atom and the p^{th} protein effective point, r_{max} has a fixed value of 9 Å [145], and $k_{lp} = r_{lp}/r_{max}$. $\hat{H}_p$ is the hydrophobicity parameter for the p^{th} protein effective point accross to a scale derived for amino acids in globular proteins from crystallographic data [146].

Next, we calculate a natural logarithm of the common Gaussian restraint with the average hydrophobicity μ_l^{HP} and the corresponding standard deviation σ_l^{HP}:

$$P_{HP}^{rst}(l) = \frac{1}{2}\left(\frac{P_{HP}(l) - \mu_l^{HP}}{\sigma_l^{HP}}\right)^2 - ln\left(\frac{1}{\sigma_l^{HP}\sqrt{2\pi}}\right) \tag{3.10}$$

Ligand type-dependent parameters μ_l^{HP} and σ_l^{HP} are derived from the eFindSite/PDB dataset by calculating the average empirical hydrophobicity, $P_{HP}(l)$, and the corresponding standard deviation for different ligand atom types.

The hydrophotic interaction score, E_{HP} is taken as the average P_{HP}^{rest} calculated over all ligand atoms, L:

$$E_{HP} = \frac{1}{L}\sum_{l}^{L} P_{HP}^{rest}(l) \tag{3.11}$$

Generic and pocket-specific contact potentials (v, vi). The molecular docking force field implemented in GeauxDock also includes generic and pocket-specific contact potentials. The generic potential, P_{CP}, between the i^{th} ligand atom and the p^{th} protein effective point is calculated as follows:

$$P_{CP}(l,p) = S(r_{lp})(-ln\frac{n_{lp}}{n_{lp}^0}) \tag{3.12}$$

where n_{lp} is the observed number of contacts between ligand atoms of a similar type as l and protein effective points of a similar type as p across the eFindSite/PDB dataset, and n_{lp}^0 is a reference number of contacts assuming no specificity [explained in eq. 3.4]. $S(r_{lp})$ is smoothing function defined as:

$$S(r_{lp}) = \frac{1}{1 + exp[(6 - \frac{r_{lp}}{2})(r_{lp} - D_{lp}^{cnt})]} \tag{3.13}$$

where r_{lp} is the distance between l and p, and D_{lp}^{cnt} is the contact threshold that depends on the types of both l and p.

The generic contact score, E_{cp}, is calculated by summing P_{CP} values over all pairs of ligand atoms and protein effective points, and then averaging over the total number of ligand atoms, L:

$$E_{CP} = \frac{1}{L} \sum_{l}^{L} \sum_{p}^{P} P_{CP}(l, p) \tag{3.14}$$

In addition to the generic potential P_{CP} derived from the eFindSite/PDB dataset, we calculate P_{CP}^{PS}, a pocket-specific (PS) contact potential [75]. The PS version uses the same formalism as the generic potential, however, rather than derived from the eFindSite/PDB, the numbers of contacts n_{lp} and n_{lp}^{0} are calculated using a set of evolutionarily related complex structures identified for a given target protein by eThresd [147] and eFindSite [83]. Similar to E_{CP}, the pocket-specific contact score, E_{CP}^{PS}, is calculated as:

$$E_{CP}^{PS} = \frac{1}{L} \sum_{l}^{L} \sum_{p}^{P} P_{CP}^{PS}(l, p) \tag{3.15}$$

Family conserved anchor substructures and pseudopharmacophore potential (vii, viii). Ligands extracted from evolutionarily related complex structures are also used to impose a series of harmonic restraints of family conserved anchor substructures, which were shown to be highly effective in ligand docking [148], and to construct a new pseudopharmacophore model. The former performs the chemical matching of a target ligand against all template ligands using kcombu [94] to identify the maximum common substructures (MCS). Subsequently, atomic equivalences within MCS provided by kcombu are used to calculate a weighted average for RMSD values obtained against a set of A template ligands, with weights corresponding to the target-template chemical similarity measured by the Tanimoto coefficient [149]. A position restraint, P_{MCS}, imposed on the ath anchor substructure, which

is essentially an MCS detected by kcombu, is calculated as:

$$P_{MCS}(a) = \sqrt{\frac{1}{E}\sum_{e}^{E}(r_a^e)^2} \tag{3.16}$$

where the summation runs over E pairs of equivalent atoms in the target and template ligands sharing the ath anchor substructure, and r_a^e is the atomic distance for the eth pair.

Typically, multiple templates and the corresponding anchor substructures are detected for a given protein-ligand target, therefore, the final score taking into account family conserved anchor substructures, E_{MCS}, is calculated as the natural logarithm of the weighted average of individual P_{MCS} values:

$$E_{MCS} = ln(\frac{1}{A}\sum_{a}^{A} TC_a P_{MCS}(a)) \tag{3.17}$$

where TC_a is the Tanimoto coefficient corresponding to the chemical similarity between the ath template and the target molecule, and A is the total number of templates used to extract the anchor substructures.

The second energy term in this group uses a pseudopharmacophore potential. Specifically, it applies a Kernel Density Estimation (KDE) method to the positions of heavy atoms of template ligands bound to the identified homologs to estimate a probability density function. We use a standard normal density function to describe the likelihood of an atom of the docking ligand to be at a certain position within the binding site; the following is the scaled form of the kernel, K_h:

$$K_h(l, e) = \frac{1}{(2\pi h)^{3/2}} exp(-\frac{(x_l - x_e)^2 + (y_l - y_e)^2 + (z_l - z_e)^2}{2h^2}) \tag{3.18}$$

where h is the bandwidth with a value of 0.5, l is a target ligand atom, and e is a template ligand atom (l and e are of the same type). KDE provides a convenient way of data smoothing, where inference about the population are made based on a finite data sample [150, 151].

The pseudopharmacophore potential on the lth ligand atom is then calculated as:

$$P_{PHR}(l) = \frac{1}{E}\sum_{e}^{E}\begin{cases} K_h(l,e), \text{if type(e)=type(l)} \\ 0, else \end{cases} \tag{3.19}$$

Where E is the total number of template ligands.

The pseudopharmacophore score for a given configuration of a ligand within the binding site of the target protein is calculated as the average P_{PHR} over all ligand atoms, L:

$$E_{PHR} = \frac{1}{L}\sum_{l}^{L} P_{PHR}(l) \tag{3.20}$$

Distance restraint (ix). Finally, to limit the search space to the vicinity of a biding site, the following distance constraint is imposed:

$$E_{DST} = r_{cen} \tag{3.21}$$

where r_{cen} is the distance between the ligand geometric center and the binding pocket center predicted by eFindSite [83].

3.2.4 Ensemble docking

The flexibility of ligands and proteins in molecular docking is implemented using an ensemble-based approach. This commonly used technique first precalculates an ensemble of low-energy conformations and then performs a rigid-body docking for each conformer [152, 153]. For ligands, we used a procedure described previously [148] to generate nonredundant ensembles comprising up to 50 low-energy conformations whose pairwise RMSD is $> 1\mathring{A}$. Protein ensembles were constructed using Modeller [154] based on the experimental structure of each target (self-modeling). We used only the coordinates of $C\alpha$ atoms belonging to nonbinding residues to fully explore the flexibility of ligand binding regions. For each target proteins, 10 models were generated by Modeller through three

rounds of optimization using a variable target function method and molecular dynamics refinement with the objective function set to 10^6.

3.2.5 Monte Carlo sampling

We use the Metropolis Monte Carlo (MMC) method [75, 155] to sample the conformational space of protein-ligand interactions. This space consists of multiple subspaces representing unique combinations of protein and ligand conformations from the precalculated ensembles. In the current implementation, each subspace is sampled independently using the MMC method and the collected trajectories are merged at the end of simulations. In each single MMC step, the position and orientation of a ligand are randomly perturbed by translational and rotational steps about the x, y and z-axis of up to $0.02\mathring{A}$ and 5 deg, respectively. We found that this protocal allows a ligand to freely explore the conformation space without compromising the precision. Furthermore, the temperature factor is chosen such that the average acceptance ratio is about 0.5. Note that in GeauxDock, both the perturbation scale and the temperature factor can be modified to achieve a better performance for particular systems. As MMC simulations search for the global minimum energy state of a system, a scoring function implemented in GeauxDock is optimized to assign low pseudoenergy values to near-native conformations. Consequently, native-like binding modes are frequently visited during the conformational sampling providing a sufficient resolution of biologically relevant states.

3.2.6 Force field optimization

The total pseudoenergy score for a given configuration of a ligand binding site of its protein target is calculated as a linear combination of the individual energy terms:

$$E = w_1 E_{ele}^{soft} + w_2 E_{vdW}^{soft} + w_3 E_{HB} + w_4 E_{HP} + w_5 E_{CP} + w_6 E_{CP}^{PS} + w_7 E_{MCS} + w_8 E_{PHR} + w_9 E_{DST} \tag{3.22}$$

The energy weight factors, $w_1 - w_9$, are optimized on a large and nonredundant set of protein-ligand conformations constructed for the Astex/CCDC dataset [156]. Specifically,

for each complex, we first generated 10^5 configurations through a series MMC simulations including only the Lennard-Johns potential (i) to prevent steric clashes and the distance constraint (ix) to confine the sampling to the vicinity of a binding pocket. Next, we calculated pairwise CMS values for all conformations to create a $10^5 \times 10^5$ CMS matrix. To remove redundancy, this matrix was subjected to clustering using CLUTO [157] resulting in 5000 groups; a cluster centroid was selected to represent each group. The final dataset comprises 102,000 nonredundant protein-ligand configurations constructed for 204 complexes.

Subsequently, we compiled two subsets for the force field optimization, a group of 36,022 native-like conformations whose CMS values calculated against the experimental complex structures are ≥ 0.6, and a set of 847,849 decoys with the CMS of ≤ 0.4. The native-like recognition capability of the scoring function was enhanced by finding the set of weights $w_1 - w_9$ [see eq. 3.22] that maximize the energy gap between native-like and decoy conformations measured by the Z-score:

$$Z - score = \frac{< E_d > - < E_n >}{\sigma_n^2 + \sigma_d^2} \tag{3.23}$$

where $< E_d >$ and $< E_n >$ are the mean energy values calculated for native-like and decoy conformations, respectively, with the corresponding standard deviations σ_n and σ_d.

We used the evolutionary search algorithm [144] emulating the principles of natural evolution to identify the optimal set of energy weight factors that maximize the Z-score. To avoid any bias, the optimization was performed 10 times starting from different initial random sets of weights; the final weight factors were taken as the consensus of the 10 optimization rounds.

3.2.7 Other scoring functions

Two state-of-the-art algorithms, DrugScoreX (DSX) [122] and Ligand-Protein Contacts (LPC) [158], were selected for comparative benchmarks of GeauxDock. DSX is a knowledge-

based scoring function that features a distance-dependent pair potential, a torsion angle potential, and a novel solvent accessible surface-dependent potential [122]. LPC uses a scoring function that evaluates the geometric and chemical complementarity between a ligand and its receptor protein [159]. Both programs were used with their default set of parameters.

3.3 Results and discussion

3.3.1 Ensembles for pseudoflexible docking

It is well known that both proteins and ligands often undergo structural changes on complex formation [135, 160–162]; for instance, an analysis of 27 flexible ligands shows the RMSD variation from 0.19 to 2.96 Å [162] calculated between single-crystal and protein-bound states. A larger structural diversity is expected as the size of ligand molecules increases; for instance, the conformational range for two ubiquitous compounds, nicotinamide adenine dinucleotide and vlavin adenine dinucleotide was calculated as 3.58 Å ± 0.08 and 3.49 Å ± 0.13, respectively, when bound to proteins [161]. On that account, an accurate representation of biomolecules in simulations requires sampling multiple conformational states [163]. We use an ensemble docking technique to handle this issue in a discrete manner. Specifically, conformers are selected from a precomputed pool of configurations covering a large conformational space that includes biologically relevant molecules. In that regard, we investigate the coverage of Astex/CCDC ligands by calculating RMSD values using conformational ensembles and the corresponding experimental structures. The results in terms of maximum, minimum, and medium RMSD values are presented in 3.1. Figure 3.1A shows that the median RMSD for 81% of the flexible ligands is within the reference range of 0.19 to 2.96 Å [162] suggesting that the ligand flexibility is well represented across the generate docking ensembles. Furthermore, the average plasticity of ligand-binding regions in proteins expressed as the mean RMSD was estimated as 2.6 Å with a standard deviation of 1.0 Å [164]. Protein ensembles constructed in this study fall within this range

with the median binding site RMSD calculated over 204 ensembles of 2.61 Å, as shown in Figure 3.1B. Collectively, these results demonstrate that conformational ensembles for pseudoflexible docking provide a sufficient coverage of biologically relevant structures of both ligands and their protein targets.

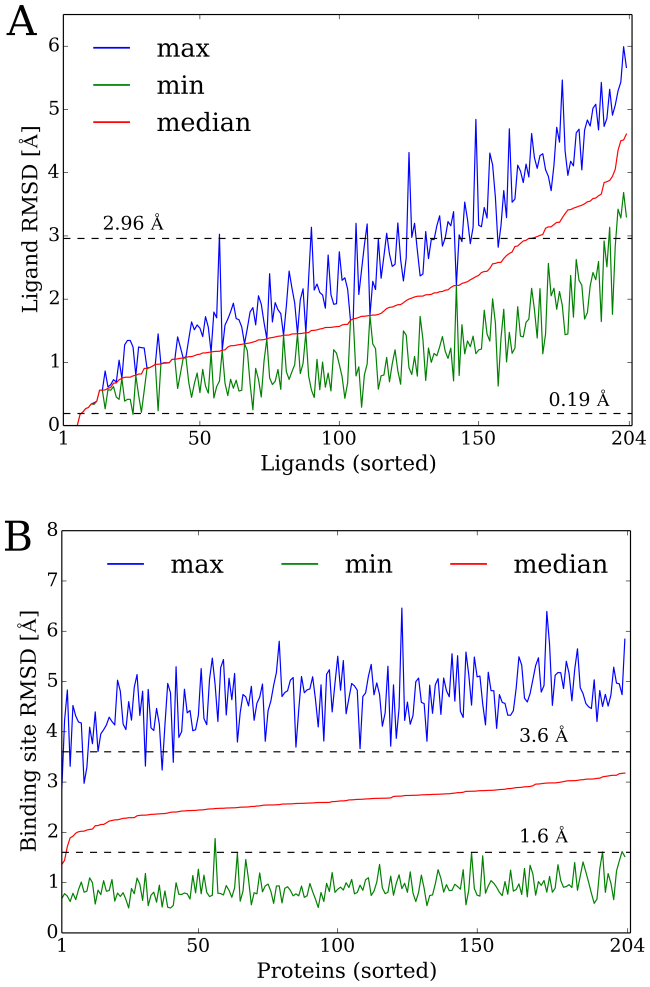

Figure 3.1: Structural characteristics of protein and ligand ensembles for pseudoflexible docking. All-atom RMSD values are calculated using the native conformation for (A) ligands and (B) protein binding sites. Dashed lines point out the estimated ranges of the molecular plasticity. Blue, green, and red lines correspond to the maximum, minimum, and median RMSD within each ensemble; molecules are sorted on the x-axis by their median values.

3.3.2 Force field parameterization

Force fields for molecular modeling typically require a careful parameterization to reproduce experimental data. We derived the parameters for GeauxDock from the *e*FindSite/PDB

dataset, a representative and nonredundant collection of protein-ligand complex structures. Selected force field potentials parameterized against eFindSite/PDB are presented in Figure 3.2. Figure 3.2A shows the soft Lennard-Jones potential used to model va der Waals interactions between effective points on Phe and Arg side chains, and selected ligand atoms. The corresponding parameters ε that defines the depth of the potential well are reported in Table 3.1 and Table 3.2. For instance, aromatic interactions between Phe-2 and C.ar, and a salt bridge between Arg-2 and O.co2 have deeper potential wells with $\varepsilon = 1.95$ and $\varepsilon = 1.54$, respectively, compared to those less favorable, for example, Phe-2 and N.3 ($\varepsilon = 1.07$), and Arg-2 and N.am ($\varepsilon = 0.43$). Furthermore, the softened potential, which is weaker at short distance than the translational form, is more appropriate to model interactions involving effective points representing clouds of atoms rather than the hard spheres of individual particles.

Table 3.1: Force field parameters (first part) for van der Waals interactions and the generic contact potential for selected ligand atom types and protein effective points.

Protein point	Parameter	Ligand atom type						
		C.3	C.ar	C.cat	N.3	N.am	N.ar	O.2
C_{alpha}	ϵ	0.59	0.65	0.99	0.53	0.44	0.86	1.08
	P_{cp}	0.21	0.09	-0.47	0.37	0.59	-0.33	-0.69
PP	ϵ	0.66	0.79	0.87	0.64	0.56	0.61	0.83
	P_{cp}	0.1	-0.09	-0.2	0.13	0.32	0.22	-0.23
Phe-1	ϵ	0.63	0.77	1.3	0.64	0.3	0.63	0.75
	P_{cp}	0.12	-0.1	-0.96	0.1	1.22	-0.1	-0.13
Phe-2	ϵ	1.77	1.95	1.58	1.07	1.34	1.95	1.46
	P_{cp}	-1.55	-1.79	-1.33	-0.69	-1.1	-1.72	-1.21
Arg-1	ϵ	0.19	0.21	0.12	0.47	0.18	0.26	0.39
	P_{cp}	1.6	1.4	2.06	0.53	1.61	1.17	0.85
Arg-2	ϵ	0.63	0.55	0.27	0.7	0.43	0.62	1
	P_{cp}	0.09	0.29	1.07	-0.01	0.63	0.14	-0.58

We also use a soft version of the electrostatic potential, where its values do not extend to infinity when the interaction distance r approaches zero. As shown in Figure 3.2B, the electrostatic potential creates a repulsion at close distances between those groups whose partial charges have the same sign, whereas positively and negatively charged particles attract each other. The strength of the interactions depends on the partial charges on

Table 3.2: Force field parameters (second part) for van der Waals interactions and the generic contact potential for selected ligand atom types and protein effective points.

Protein point	Parameter	Ligand atom type					
		O.3	O.co2	P.3	S.3	S.O2	Cl
C_{alpha}	ϵ	0.86	0.95	0.05	0.62	0.82	1.03
	P_{cp}	-0.26	-0.47	3.01	0.15	-0.19	-0.63
PP	ϵ	0.58	1.02	0.05	0.84	0.8	1.05
	P_{cp}	0.14	-0.55	-0.69	-0.25	-0.14	-0.57
Phe-1	ϵ	0.43	0.55	0.26	0.79	0.3	1.07
	P_{cp}	0.56	0.37	4.01	-0.18	0.87	-0.67
Phe-2	ϵ	1.29	1.6	1.17	1.66	0.82	2.19
	P_{cp}	-1	-1.27	-0.86	-1.48	-0.26	-2.11
Arg-1	ϵ	0.31	0.52	0.43	0.25	0.05	0.25
	P_{cp}	1.18	0.36	0.63	1.1	3	1.29
Arg-2	ϵ	0.82	1.54	0	1.08	0.85	0.56
	P_{cp}	-0.31	-1.31	6.52	-0.8	0.04	0.29

individual groups. Table 3.3 lists net charges assigned to protein effective points by collapsing AMBER partial charges of the constituent atoms. A point charge on the PP has a fixed value of -0.246, which balances positively charged Cα atoms of individual amino acids. Side chains of small hydrophobic residues are slightly positively charged, for example, $q_p = 0.047$ for Ile-1, in contrast to small polar amino acids that carry small negative charges on their side chains, for example, $q_p = -0.046$ for Ser-1. A small negatively charged Asp has the unit charge assigned to its side chain effective point, whereas larger charged residues have almost unit charge values; for example, the parameter q_p is -0.792, 0.901, and 0.927 for Glu-2, Arg-2, and Lys-2, respectively. Partial charges on ligand heavy atoms are calculated for individual compounds using the Mulliken population analysis [142], which is widely used parameterization method in molecular docking.

Hydrogen bonds are modeled for hydrogen donor-acceptor pairs using single Gaussian restraints. Table 3.4 lists force field parameters for hydrogen bonds and Figure 3.2C shows the parameterized potential for selected pairs. Mean values for the interaction distance, μ_{lp}^{HB}, derived from the *e*FindSite/PDB dataset, give the optimal type-dependent bond lengths, whereas σ_{lp}^{HB} parameters that describe the deviation from average interaction

Table 3.3: Partial charges on Cα and side chain (SC) effective points of amino acids.

Amino acid	Effective point		
	Cα	SC-1	SC-2
Gly	0.246	-	-
Ala	0.215	0.031	-
Asn	0.217	0.029	-
Asp	0.246	-1	-
Cys	0.088	0.158	-
Ile	0.199	0.047	-
Leu	0.204	0.042	-
Pro	0.112	0.119	-
Ser	0.292	-0.046	-
Thr	0.268	-0.022	-
Val	0.201	0.045	-
Arg	0.237	0.107	0.901
Glu	0.246	-0.208	-0.792
Gln	0.21	0.01	0.026
His	0.219	0.172	-0.145
Lys	0.227	0.092	0.927
Met	0.137	0.127	-0.018
Phe	0.214	0.049	-0.017
Trp	0.248	0.066	-0.068
Tyr	0.245	0.02	-0.02

distances across the dataset, control the interaction strength. For instance, μ_{lp}^{HB} for Thr-1 and N.3 (3.59 Å) is slightly smaller than that for Tyr-2 and N.am (3.88 Å). Moreover, the corresponding σ_{lp}^{HB} are 0.95 and 0.78, respectively, thus, the strength of hydrogen bonded pair of Tyr-2 and N.am at the optimal distance is greater than a hydrogen bond between Thr-1 and N.3.

In our model, protein residues create a polar/hydrophobic local environment favoring certain types of ligand atoms. These hydrophobic interactions are parameterized using statistical collected for *e*FindSite/PDB protein-ligand complexes and a standard hydrophobicity scale for amino acids. The derived force field parameters reported in Table 3.5 are in good agreement with physicochemical properties of ligand atom types. For example, aromatic barbon atoms ($\mu_l^{HP} = 0.11$) and halogens ($\mu_l^{HP} = 0.24$) tend toward nonpolar residues, whereas amine nitrogen ($\mu_l^{HP} = -0.27$) and carboxylate oxygen ($\mu_l^{HP} = -0.34$)

Table 3.4: Force field parameters for hydrogen bonds, $\mu_{lp}^{HB} \pm \sigma_{lp}^{HB}$, for selected ligand types and protein effective points.

Ligand atom type	Protein effective point				
	His-2	Ser-2	Thr-1	Tyr-2	PP
N.2	3.38 0.71	3.83 0.83	3.91 0.99	3.64 0.77	3.91 0.88
N.3	3.67 0.71	3.80 0.88	3.59 0.95	3.79 0.89	3.89 0.92
N.am	3.80 0.75	3.82 0.83	3.79 0.82	3.88 0.78	3.62 0.82
O.2	3.58 0.78	3.64 0.87	3.62 0.86	3.75 0.92	3.69 0.84
O.3	3.64 0.83	3.68 0.86	3.72 0.85	3.74 0.85	3.85 0.84
O.co2	3.50 0.76	3.45 0.87	3.64 0.92	3.46 0.86	3.75 0.86

Table 3.5: Force field parameters for hydrophobic interactions, $\mu_{lp}^{HB} \pm \sigma_{lp}^{HB}$, assigned to selected ligand types.

Ligand atom type	$\mu_{lp}^{HB} \pm \sigma_{lp}^{HB}$
C.3	-0.03 ± 0.43
C.ar	0.11 ± 0.46
C.cat	-0.26 ± 0.43
N.3	-0.27 ± 0.44
N.am	-0.10 ± 0.38
N.ar	0.03 ± 0.47
O.2	-0.21 ± 0.50
O.3	-0.28 ± 0.46
O.co2	-0.34 ± 0.46
P.3	-0.50 ± 0.41
S.3	-0.14 ± 0.45
S.O2	-0.10 ± 0.40
Cl	0.24 ± 0.52

atoms prefer a polar microenvironment. Hydrophobicity restraints P_{HP}^{rest} for selected ligand atom types are shown in Figure 3.2D as a function of the environment created by surrounding amino acids. The extremes of -1.0 and 1.0 describe a strongly polar and nonpolar character, respectively. The position of the function minimum determines the optimal environment for a particular atom type described by P_{HP}, thus, Cl and C.ar are on the positive side, and N.am and O.3 are on the negative side of the protein hydrophobicity range.

Statistical potentials are commonly used components of molecular docking force fields [165–167]. In this study, the parameters for pairwise interactions between ligand heavy atoms and protein effective points were derived from the *e*FindSite/PDB dataset. The

log-odds potential expresses the likelihood of individual contacts, where the interactions averaged over the entire dataset are taken as a reference state. Figure 3.2E shows the extreme values for the contact potential between aromatic carbon C.ar and all types of protein effective points. Clearly, aromatic moieties on the side chain effective points of Phe-2, Trp-2, and Tyr-2, as well as the hydrophobic parts of Cys-1, Ile-1, Met-2, Leu-1, and Val-1 make contacts with C.ar more often than by a random change. In contrast, the polar and charged groups of Glu-1, Lys-1, Arg-1, Glu-2, and Lys-2 are statistically unlikely to interact with ligand aromatic carbon atoms. Moreover, backbone effective points Cα and PP have no distinct preferences toward interacting with C.ar.

In the GeauxDock force field, we use a smoothing function that is less sensitive to small changes in ligand coordinates at the contact distance thresholds than the commonly used step function. This is shown in Figure 3.2F for selected interactions between ligand heavy atoms and protein effective points. For instance, salt bridges between Arg-2 and O.co2, and Asp-1 and N.3 contribute half of their negative interaction energy at $D_{lp}^{cnt} = 5.76$ Å and $D_{lp}^{cnt} = 5.36$ Å, respectively. Similarly, the positive energy contribution from destabilizing interactions between Ala-1 and O.3, and Glu-2 and C.ar reach half of their values at the corresponding contact thresholds. In addition to the generic contact potential derived from the *e*FindSite/PDB dataset, we calculate its pocket-specific variant using evolutionarily related complexes identified by sequence profile-based protein threading. These potentials are specific toward a particular family of proteins, however, they contain significantly less parameters compared with the genetic potential because of much smaller sample sizes (the number of template complexes). For example, out of 720 pairwise parameters derived from the *e*FindSite/PDB dataset for P_{CP}, the average number of P_{CP}^{PS} parameters calculated across the Astex/CCDC target pockets is only 110 ± 67. Nonetheless, the latter have been demonstrated to be more accurate than the generic potential in the scoring and ranking of ligand binding modes [75].

Different from traditional pharmacophore-based models that use known bio-active to

calculate sets of steric and physicochemical features necessary for molecular recognition [168], the pseudopharmacophore potential in GeauxDock is derived from evolutionarily ligand-bound templates. Specifically, it estimates a probability for each ligand heavy atom type to be at a certain position within the binding site. For instance, Figure 3.2G shows a one-dimensional probability density for C.ar, N.am, O.co2, and O.3 along the x-coordinate with the pocket centered at x=y=z=0 Å (the full potential is the product of probabilities at x, y, and z coordinates). In this example, amine nitrogen and hydroxyl oxygen atoms are most likely to be found at x=-1.4Å and x=-2.5Å, respectively. Carboxyl oxygen atoms have a bimodal distribution typical for symmetric moieties with two equivalent peaks at $x = 0.4\mathring{A}$ and $x = 2.5\mathring{A}$, whereas aromatic aromatic carbon atoms have a relatively borad probability of occurrence at $-0.6 < x < 2.5\mathring{A}$. Favoring ligand heavy atoms at their high probability positions predicts binding modes consistent with hte conserved evolutionary profiles observed across set of homologous proteins.

3.3.3 Force field optimization

Force field weights were optimized on a large dataset of protein-ligand configurations generated for Astex/CCDC complexes using the evolutionary search algorithm. The objective was to maximize the Z-score corresponding to the energy gap between native-like and decoy conformations. Figure 3.3A shows the trajectory of Z-score in one complete optimization process. The simulation converges within 400 generations, indicating an efficient update of weight factors. We performed the total of 10 simulations seeded with random initial weight factors; each calculation resulted in the same set of weight factors ($w_1 = 18.97, w_2 = 0.78, w_3 = 2.05, w_4 = 0.53, w_5 = 0.01, w_6 = 0.53, w_7 = 0.88, w_8 = 110.82$, and $w_9 = 46.4$), suggesting that the optimized values are stable and robust. Figures 3.3B and 3.3C show the distribution of energy values with different sets of weights. In Figure 3.3B, random weight factors do not provide a clear separation between native-like (green dots) and decoy (red dots) conformations whose median energy score is -1.67 and -1.03, respectively. In contrast, Figure 3.3C shows that the optimized weight factors yield better

energy-based partitioning of native-like and decoy conformations; here, native-like (decoy) binding modes have a median energy of -0.16(0.58). This analysis suggests that the total pseudoenergy calculated using the optimized set of weights has a great potential to effectively drive molecular docking toward correct ligand binding modes.

3.3.4 Recognition of native-like conformations

A strong capacity to identify native-like binding modes among a vast number of generated configurations plays a pivotal role in successful ligand docking simulations. Therefore, in Figure 3.4, we conduct a comparative Receiver Operating Characteristics (ROC) analysis of GeauxDock and two other scoring functions, DSX [122] and LPC [158]. Here, we use a precompiled dataset of protein-ligand configurations comprising 36,022 native-like binding poses and 847,849 decoys generated for Astex/CCDC complexes to uniformly cover the conformational space. In general, all docking algorithms are capable of identifying correct conformations across the training MMC trajectories generated for the Astex/CCDC dataset better than a purely random guess (dashed line). The area under the curve (AUC) for the unoptimized GeauxDock force field (all weight factors set to 1.0) is 0.654 in contrast to 0.851 for the optimized set of weights. For comparison, DSX_pair, DSX_pair_sas and LPC yield the AUC of 0.847, 0.858, and 0.765, respectively. Despite a slightly lower AUC, GeauxDock gives 5% higher true positive rate than DSX_pair_sas at relatively small false positive rates of 0.1-0.2. The results for DSX consistent with the original benchmarking calculation [132] suggest that our dataset is of high quality and the CMS indeed provides an effective evaluation metric.

Next, we performed full docking calculations using GeauxDock. The major difference from the previous analysis is that these validation simulations start from a random ligand conformation and use the complete, optimized force field to guide the conformational sampling. MMC trajectories generated for the Astex/CCDC dataset are analyzed in Figure 3.5. First, for each benchmarking complex, we calculated the Z-score between native-like and decoy conformations extracted from the docking trajectories. As shown in Figure 3.5A,

90% of the cases have positive Z-score values indicating that ligand binding modes close to native are systematically assigned a lower energy than those farther away from the experimental conformation. The median Z-score across Astex/CCDC complexes is 1.0, which is in accord with the training results reported in Figure 3.3. To further evaluate the quality of the GeauxDock force field, we calculated the Pearson correlation coefficient (PCC) between the total pseudoenergy score and CMS. Figure 3.5B shows that in the majority of the cases, the total pseudoenergy score and CMS are negatively correlated, that is, the energy increases with the decreasing similarity to the experimental binding mode. According to the scale provided by Salkind [168], a very strong ($-1.0 \leq PCC < -0.8$), strong ($-0.8 \leq PCC < -0.6$), moderate ($-0.6 \leq PCC < -0.4$), weak ($-0.4 \leq PCC < -0.2$), and very weak or no correlation ($-0.2 \leq PCC < 0.0$) between energy and CMS was obtained for 3.43%, 15.20%, 28.43%, 25.49% and 14.22% of the Astex/CCDC complexes, respectively; only 13.24% of the cases give the undesired positive correlation. Altogether, these results demonstrate that the scoring function in GeauxDock is correctly optimized to drive MMC simulations toward experimentally determined ligand binding modes.

3.3.5 Case studies

We select a couple of examples to demonstrate the accuracy of GeauxDock in finding near-native ligand binding modes, cathepsin K complexed with a peptidomimetic inhibitor (PDB-ID: 1bgo, chain A) [169], and actinidin complexed with an inhibitor E-64 (PDB-ID: 1aec, chain A) [170]. Both compounds were docked into the active sites of their target protein using GeauxDock starting from a random initial conformation. The results are shown in Figure 3.6 (panels A-C for cathepsin K and D-F for actinidin). First, we plot the values of CMS calculated against inhibitors bound in the crystal complex structures, and the total pseudoenergy and a low CMS for initial configurations indicate that ligands are far away from their native states (Figure 3.6A and 3.6D). During MMC simulations, a gradually decreasing energy E guides the conformational sampling to the vicinity of the experimental binding modes of inhibitors as indicated by high CMS values at the end of

simulations. Figure 3.6B and 3.6E demonstrate that in both cases, the optimized force field yield a negative correlation between CMS and E, where each dot represents one MMC snapshot. Next, we select three representative conformations from those scatter plots for each inhibitor, non-native (red), intermediate (orange), and near-native (green), whose CMS are 0.38, 0.49, and 0.90 for cathepsin K, and 0.38, 0.49 and 0.86 for actinidin, respectively. The corresponding molecular representations are shown in Figure 3.6C and 3.6F using the same color scheme. In both cases, low-energy binding modes (green) significantly overlap with bound inhibitors in the experimental structures of cathepsin K and actinidin complexes (ice blue sticks), whereas non-native and intermediate conformations are characterized by notably higher pseudoenergy values.

3.3.6 Evolution- and physics-based components

A descriptor-based force field in GeauxDock combines evolution- and physics-based scoring terms. The former are derived from evolutionary related complex structures at two different sequence similarity thresholds, 80% to allow force field parameters to be calculated from close homologs, and 40% to use only those templates that are weakly related to their targets. Therefore, we can analyze how the level of homology affects the accuracy of molecular docking. Using the Astex/CCDC dataset, the results are reported in Table 3.6 as the area under the ROC curve. As expected, the AUC significantly increases when close homologs are included in forcefield optimization and the docking conformations are evaluated by evolution-based components alone. In contrast, the performance of physics-based scoring terms remains, to a large extent, unaffected by the maximum target-template sequence identify, because these features are calculated from physical interactions that are more universal [171]. Interestingly, the performance of GeauxDock using a complete force field at a homology threshold of 80% is only slightly better than that at 40%, suggesting that the descriptor-based scoring function is able to adapt to the supplied amount of evolutionary information to maintain its accuracy.

To further investigate this intriguing observation, we calculated the relative contri-

Table 3.6: Performance of GeauxDock on the Astex/CCDC dataset assessed by the area under the curve (AUC). The force field is optimized at the homology thresholds of 40% and 80% and the performance of the complete scoring function is compared to physics- and evolution-based components.

Scoring function	AUC	
	40% homology	80% homology
Complete	0.831	0.848
Evolution-based	0.699	0.745
Physics -based	0.801	0.814

bution of both classes of scoring terms to the total pseudoenergy at the two homology thresholds. Figure 3.7 shows that the contribution from evolution-based components to the total score is about 5% higher at 80% homology compared with 40%. Considering only a slightly better performance of GeauxDock using close homologs, this analysis suggests that the scarcity of evolutionary information can be effectively compensated by the increased contribution from physics-based scoring terms. This unique feature of GeauxDock is particularly important in its large-scale applications at the proteome level, such as in inverse VS [172, 173] and rational drug repositioning [174–176], where the availability of only weakly homologous complex structures for the majority of drug targets will not compromise the accuracy of molecular docking.

A well-balanced contribution of physics- and evolution-based energy terms to the total pseudoenergy also suggests that these two classes of scoring features effectively complement each other. Nevertheless, AUC values reported in Table 3.6 indicate that a linear combination of individual energy terms perhaps does not fully exploit their predictive power; for instance, adding the evolution-based component improves the AUC of physics-based terms by about 3%. This may be caused by the feature intercorrelation, which is known to limit the performance improvements despite adding more descriptors [177]. A possible solution is to combine individual energy terms using a nonlinear model, under the assumption that noncovalent interactions often depend on one another in a nonlinear manner [178]. We will explore the feasibility of a machine learning-based force field in ligand molecular docking in the near future.

3.4 Conclusions

In this study, we describe the development of GeauxDock, a molecular docking approach featuring a novel descriptor-based scoring function and a mixed-resolution description of protein-ligand complexes. The scoring function was parameterized on a large dataset of crystal structures and further optimized using sets of computer-generated native-like and decoy conformations. Encouragingly, benchmarking calculations demonstrate that GeauxDock has a strong capacity to recognize native-like binding modes with the area under ROC of 0.85. The descriptor-based scoring function implemented in GeauxDock incorporates two distinct classes of energy terms, physics- and evolution-based. As the latter are derived from evolutionary related complex structures, their strength depends on the level of homology between the target and template systems. Interestingly, weak evolutionary constraints are effectively compensated by the increased contribution from physics-based terms, which, in turn, help maintain the accuracy of the GeauxDock scoring function at the lower levels of protein sequence similarity. Therefore, this new ligand docking approach is well suited for proteome-scale applications taking advantage of the increasingly growing protein sequence and structural data. GeauxDock is available at `http://www.institute.loni.org/lasigma/package/dock/`.

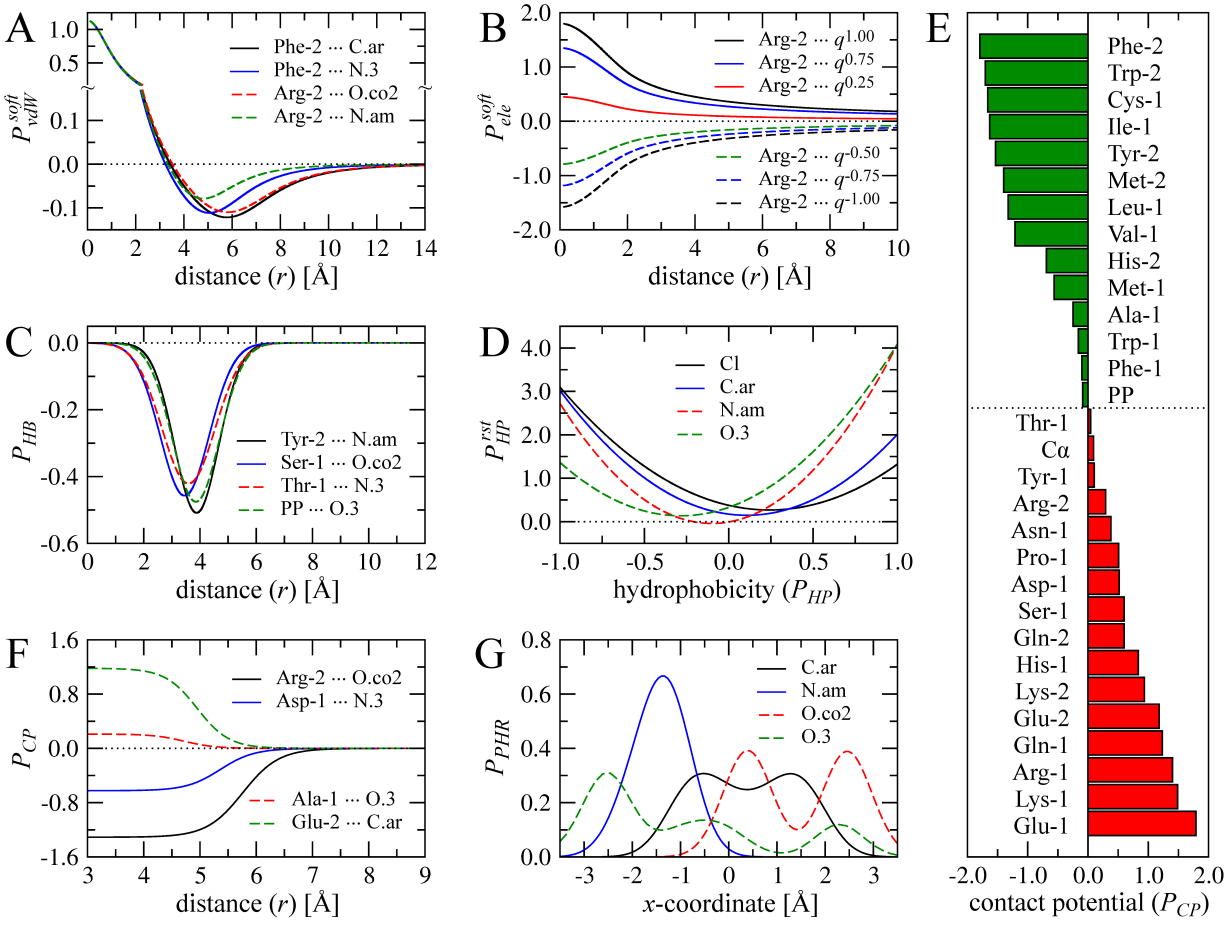

Figure 3.2: Examples of selected force field potentials. (A) Type-dependent soft Lennard-Jones potential, (B) soft electrostatic potential between protein effec- tive points and various charges on ligand atoms q, (C) hydrogen bond restraints, (D) restraints for hydrophobic interactions between different ligand atoms as a function of local hydrophobicity, (E) extreme values for the log-odds potential between aromatic carbon C.ar and protein effective points, (F) generic contact potential including a smoothing function, and (G) probability density for different ligand atoms estimated by KDE along the x-axis.

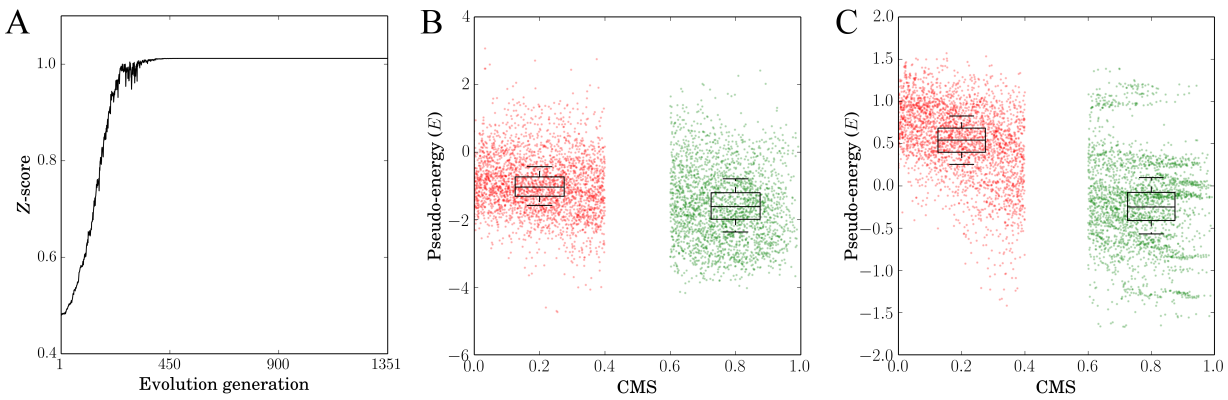

Figure 3.3: Force field optimization using the evolutionary algorithm. (A) The trajectory of Z-score in the course of the optimization procedure. The distribution of pseudoenergy values for native-like (green) and decoy (red) conformations for the (B) unoptimized and (C) optimized force field. Boxes in B and C end at the quartiles Q_1 and Q_3, a horizontal blue line in a box is the median, and whiskers show the 1.5 interquartile range.

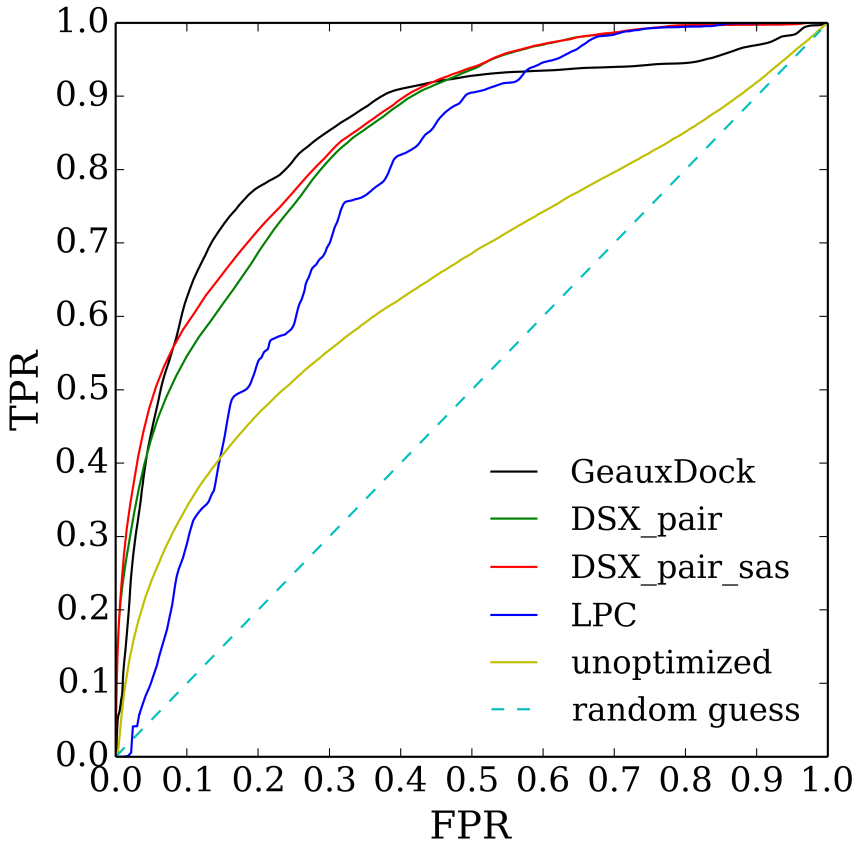

Figure 3.4: Recognition of native-like conformations across docking trajectories. A ROC plot for GeauxDock with an optimized force field is compared with those obtained using the unoptimized force field as well as other scoring functions, DSX and LPC. TPR true positive rate, FPR false posi- tive rate.

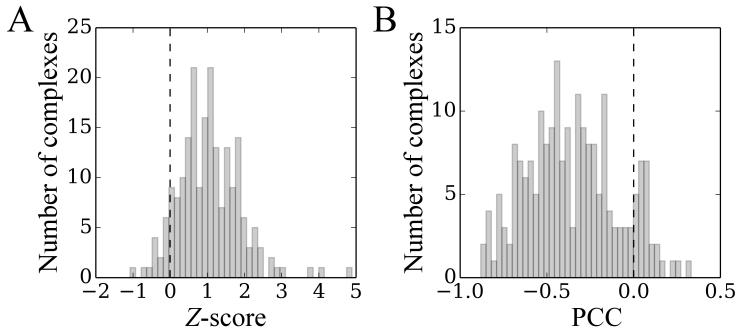

Figure 3.5: Quality assessment for the optimized force field implemented in GeauxDock. Histograms of (A) Z-score and (B) the PCC calculated from the Monte Carlo trajectories collected for the Astex/CCDC dataset.

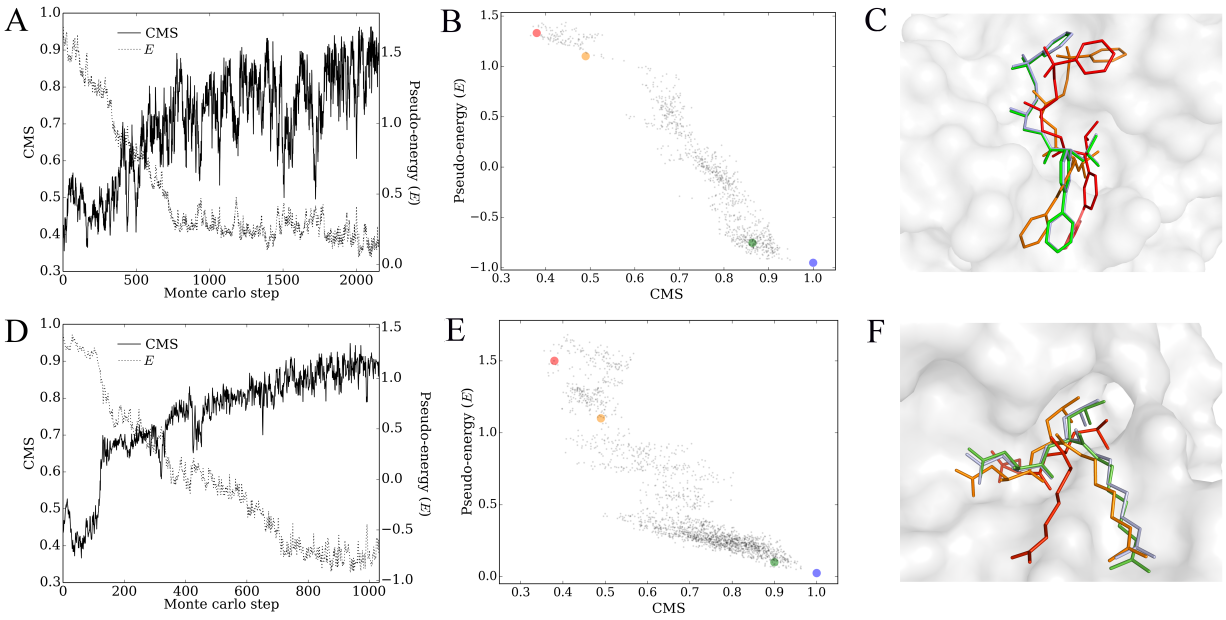

Figure 3.6: Docking results for (AC) cathepsin K and (DF) actinidin from GeauxDock. (A, D) Monte Carlo trajectories for the Contact Mode Score (CMS) and the pseudoenergy, (B, E) scatter plots of the CMS versus pseudoenergy, (C, F) representative conformations taken from docking trajectories. In B, C, E, and F selected non-native, intermediate, and near-native conformations are colored in red, orange, and green, respectively, whereas the experimental binding poses are shown in ice blue.

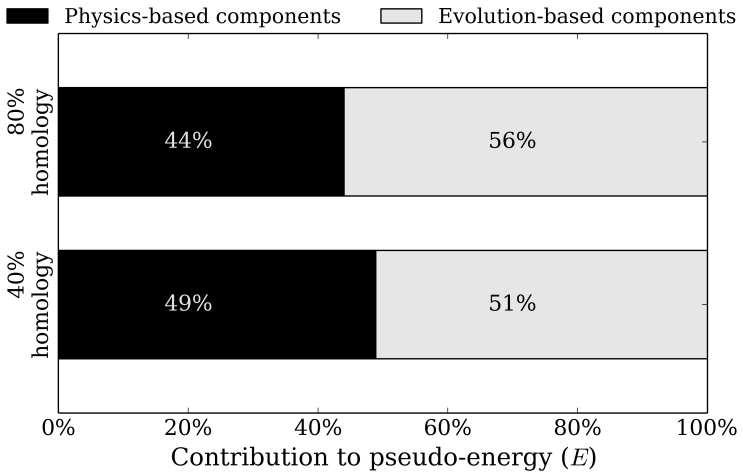

Figure 3.7: Balance of various energy terms in the optimized force field. The contribution from physics- and evolution-based components is calculated at the thresholds of 80 and 40% for the maximum target-template sequence identity.

Chapter 4
GEAUXDOCK COMPUTING

4.1 Introduction

[1]A number of computer programs have been developed to date for molecular docking simulation [179]. In general, using large compound databases increases the chances of finding bioactives, however, large-scale virtual screening typically requires a long computing time. In addition to the database size, computing time also increases with the increasing accu- racy of the modeling of drug-protein interactions. Although sophisticated models outperform simple approaches, these algorithms often have high demands for computational resources. For example, docking accuracy can be improved by incorporating the plasticity of biomolecules, e.g. using pre-generated ensembles of target protein structures [68]. Since ensemble- based docking requires conducting docking simulation for each target conformation, the computational complexity increases linearly with the number of conformers. Another approach to improve ligand docking incorporates the configurational entropy. This property can be approximated by clustering ligand binding poses generated by a docking program to calculate the conformational similarity between each pair of ligand modes, leading to O(n^2) complexity, where n is the total number of binding poses. Mining Minima provides a more accurate way to calculate entropy by integrating potential energies as a function of coordinates, however, at a significantly increased computational cos [69]. Finally, the simulation time can also affect docking accuracy for those docking programs relying on stochastic methods to sample the free energy landscape, where longer simulations are more likely to reach the global minimum [180].

Undeniably, achieving a good balance between docking accuracy and the computation time represents a major challenge in structure-based virtual screening. To address this problem, par- allel computing is often used to accelerate docking simulations. Parallel

[1]This section in the original published article is mainly accomplished by this dissertation author

architectures fall into two broad categories: 1) small groups of tightly coupled processors sharing a common memory space, and 2) large, scalable systems that do not share a common memory. Both models often coexist in a high-performance computing (HPC) environment; for instance, many HPC sys- tems use the distributed memory model to scale up to thousands of multi-processor nodes, each employing the shared memory model. A common programming practice for shared memory systems is to inform the compiler of parts of the serial code to be executed in parallel by including extra hints, e.g. using OpenMP pragmas [47]. In contrast, distributed memory systems require manually implemented message-passing procedures, e.g. using Message Pass- ing Interface (MPI) protocols [46]. Parallel programming used to be a small niche until the tra- ditional single-core Central Processing Unit (CPU) hit the "instruction level parallelism wall and the "clock speed wall" [181] a decade ago. Although CPU vendors managed to bypass these limitations by integrating more computing cores into a CPU, contemporary multi-core CPUs are not the ultimate solution due to the power [182] and energy [183] problems. A new trend in processor design to replace a handful of heavyweight cores with a massive amount of light-weight computing units upthrust parallel programming to the mainstream.

In contrast to traditional CPUarchitectures designed to minimize the execution latency of serial codes, highly simplified cores of modern accelerators are generally optimized for high- throughput computations, therefore, their performance on latency-sensitive applications is often poor. Consequently, these computing units are usually attached to conventional CPU- based systems as heterogeneous devices equipped with their own memory. Two major accelera- tor architectures currently available, NVIDIA Graphics Processing Unit (GPU) and Intel Xeon Phi, share some common features, but also have unique characteristics. With respect to hard- ware, both accelerators as well as contemporary multi-core CPUs share a two-level parallel design principle. The outer, coarse-grained level defines a computation cluster whose individ- ual processing elements provide the inner, fine-grained level of parallelism. With regard to software, each coarse-grained cluster handles its own

programming context known as a thread on CPU and Xeon Phi, and a thread block defined by the GPU Compute Unified Device Archi- tecture (CUDA) [48] paradigm. On CPU and Xeon Phi, the inner level exposes data parallel- ism, viz. Single Instruction, Multiple Data (SIMD) operations. NVIDIA GPU uses CUDA threads inheriting a similar principle of vector processing. For instance, a bundle of 32 conse- cutive CUDA threads, denoted as a warp, are scheduled together. Consequently, CUDA threads may go predication when a small, conditionally protected piece of code is encountered, forcing the execution of all instructions.

When different CUDA threads take different paths in multiple-path branches, more cycles are consumed leading to a lower device utilization. Although SIMD instructions on CPU and Xeon Phi have similar characteristics, the number of vector elements is about one-quarter to one-half of that on GPU and the code generation heuristic can vary significantly, therefore, an irregular code may perform dramatically differently on these platforms. Another major differ- ence between CPU and Xeon Phi, and GPU is that the former implement hardware multi- threading at the outer level, whereas multi-threading on GPU is at the inner level demanding more data parallelism. Compared to CPU, contemporary Xeon Phi delivers roughly equal amount of raw compute power per core in terms of the number of data operations per cycle. However, because of a larger number of computing cores on the co-processor, it offers certain advantages over CPU in processing regular, highly parallel workloads. On the other hand, CPU cores typically perform better for irregular workloads. In addition to core characteristics, com- puting performance is also affected by memory operations. Different from the automatic mem- ory management as cache on CPU and Xeon Phi, GPU exposes to programmers its fast on- chip memory, known as the CUDA shared memory.

A common programming practice for GPU is to exploit the parallelism using low-level Application Programming Interfaces (APIs), such as CUDA and OpenCL [184]. GPU program- ming typically comprises several stages, 1) identify parallel workloads, 2) copy

data from the host to the device, 3) map workloads to computing cores, 4) determine a suitable memory access for CUDA threads, 5) synchronize the execution between GPU and CPU, and 6) copy data back to the host. Despite a significant effort directed to help automate these steps, high- level GPU programming languages are still not versatile enough to fully unleash the power of GPU for complex applications. In contrast, Xeon Phi is designed to provide massive parallelism at considerably reduced programming efforts. Intel compilers can generate Xeon Phi acceler- ated binaries in a similar way to compiling traditional CPU codes [185], therefore, programming Xeon Phi in the native mode is fairly comparable to coding for multiple-core CPUs. Similar to GPU, Xeon Phi also offers an offload mode, where only selected portions of the code marked by compiler pragmas are executed on the accelerator. OpenMP can be used in both native and offload modes alleviating the need for low-level implementations.

In order to address computational challenges in structure-based virtual screening, several docking programs offer HPC capabilities. For instance, AutoDock Vina [72] supports multi- threading on CPU using the Boost::thread library yielding significant speedups on multi-core processors compared to a serial version. Moreover, a CUDA implementation of MolDock accelerates both the evolution search algorithm and its two-element scoring functions on GPU [186], whereas PLANTS employs a systematic grid search with an accelerated scoring function on GPUusing a high-level shading language [187]. A few projects take the heterogeneous con- cept one step further by developing a hybrid docking framework that can be executed on differ- ent computer architectures. For example, non-bonded interactions in molecular dynamics kernels were parallelized for both GPU (using CUDA) and CPU (using OpenMP), and further extended to fully utilize distributed platforms through MPI protocols [188]. The docking engine BUDE [189] employs the OpenCL language to maintain a parallel implementation of the genetic search algorithm for CPU, Xeon Phi and GPU. Nonetheless, to the best of our knowledge, an efficient multiple-backend implementa-

tion of the docking kernel based on Metropolis Monte Carlo (MMC) has not been reported yet.

Recently, we developed GeauxDock, a new molecular docking package to model drug-pro- tein complexes using a mixed-resolution molecular representation and the MMC search engine [77]. GeauxDock uses non-hydrogen atoms for ligands, whereas proteins are described at the coarse-grained, sub-residual level. Such a mixed-resolution description not only helps tolerate structural deformations in the target binding sites caused by using protein models as docking targets, but also speeds up calculations by decreasing the number of interaction points on mac- romolecules. The descriptor-based force field implemented in GeauxDock includes nine energy terms carefully optimized to drive docking simulations toward native-like conformations using a multi-replicaMMCsampling. Furthermore, GeauxDock employs an ensemble-based approach to effectively model the flexibility of ligands and proteins. Although GeauxDock sim- ulations typically converge in less than 1,000MMCcycles on standard datasets, its large-scale virtual screening applications remain computationally challenging due to a large number of candidate molecules to be evaluated. On that account, the present study describes our efforts porting GeauxDock to multi-core CPUs and massively parallel accelerators, Xeon Phi and GPU. Computational models and performance patterns are analyzed in detail for different architectures. We also discuss various code characteristics as well as general and platform-specific optimization techniques used to turn GeauxDock into an ultra-fast docking tool for large-scale drug virtual screening.

4.2 Materials and methods

4.2.1 Virtual Screening Workflow

[2]GeauxDock is designed for virtual screening applications, where a given protein target is screened against a large library of small organic compounds. A docking simulation of a

[2]The dissertation author contributed heavily to all the subsections in the Materials and Methods in the original published work, but did not write the text

single ligand is an independent computational task. Figure 4.1 shows four stages of virtual screening using GeauxDock. The procedure starts with reading the input data and creating a pool of tasks (Figure 4.1A). Protein and ligand files provide the initial coordinates of the target protein and library compounds. The parameter file specifies various parameters, such as coefficients to calculate energy terms, weight factors to linearly combine individual energy components, as well as the length of rotation and translation vectors to perturb ligand conformations duringMMCsimu- lations. Other files contain data to calculate a pseudo-pharmacophore using the Kernel Density Estimation (KDE), restraints on family-conserved anchor substructures using the Maximum Common Substructure (MCS), and a pocket-specific potential (PSP). The KDE component of the scoring function describes the likelihood of target ligand atoms to be at certain positions with respect to template-bound ligand atoms, whereas the MCS term imposes root-mean- square deviation (RMSD) restraints according to a chemical matching between the target ligand and template-bound ligands collected from the PDB [77,83]. Further, PSP is a contact-based statistical potential derived from weakly homologous holo-templates identified by threading rather than all protein-ligand complexes present in the PDB [77, 190]. Once the required input data are read and pre-processed, a computing device is initialized and the data is copied to the accelerator (Figure 4.1B). Subsequently, docking calculations are performed for individual tasks (Figure 4.1C) and finally, the output files are generated on the host (Figure 4.1D). Preliminary testing of this workflow reveals that the redundant loading and parsing of the same target protein when docking different ligands consumes up to 90% of the total I/O time (Table 4.1 and 4.2). As a consequence of these excessive I/O operations, the execution ofMMCkernels on GPUmakes for only 52% of the total simulation time. Furthermore, the repetitive GPU memory allocation and de-allocation performed for each task takes almost as much time as running the MMC kernel. Although the code for Xeon Phi is expected to have similar issues, the compiler pragmas are placed inside theMMCkernel code, thus the entire offload proce- dure combines data transfer and core calculations. The memory

management for the code off- load is not required in the CPU implementation. To address the problem of the excessive I/O operations particularly for GPU-based platforms, the four-step workflow for GeauxDock is arranged into two parts. The front-end consists of data loading, pre-processing and creating a pool of tasks (Figure 4.1A), whereas the back-end fetches tasks, initializes a computing device, exe- cutes the docking kernel, and periodically saves the output data (Figure 4.1B-4.1D). With this design, the memory allocation and de-allocation on GPU occur only once at the beginning and the end of the back-end process, respectively.

Table 4.1: Time in ms required to complete various stages of a docking simulation by GeauxDock for the 1a07 complex (first part).

Computing platform	Loading data	
	Protein	Other
CPU	214	21
Xeon Phi	214	21
GPU	216	21

4.2.2 Code Implementation

Docking simulations with GeauxDock can be conducted on three platforms, multi-core CPU, GPU and Xeon Phi. Therefore, the source code is modularized for an easy maintenance across different architectures (Figure 4.2). All three platforms share a common code for front-end computations, whereas back-end codes have two versions, one for CPU and Xeon Phi, and one for GPU. The C++ kernel employing OpenMP and Intel SIMD pragmas is shared between CPU and Xeon Phi. Using the "-Doffload" flag enables additional pragmas protected by the "#ifdef offlow" macro, which which instruct the compiler to generate

Table 4.2: Time in ms required to complete various stages of a docking simulation by GeauxDock for the 1a07 complex (second part).

Computing platform	Initialization		Simulation	Output generation	
	Device MemAlloc	Copy data to device	Docking kernel	Copy data from device	Device MemFree
CPU	-	-	4,848	-	-
Xeon Phi	3,135 (initialization + simulation + output)				
GPU	2,063	0.72	2,740	8	182

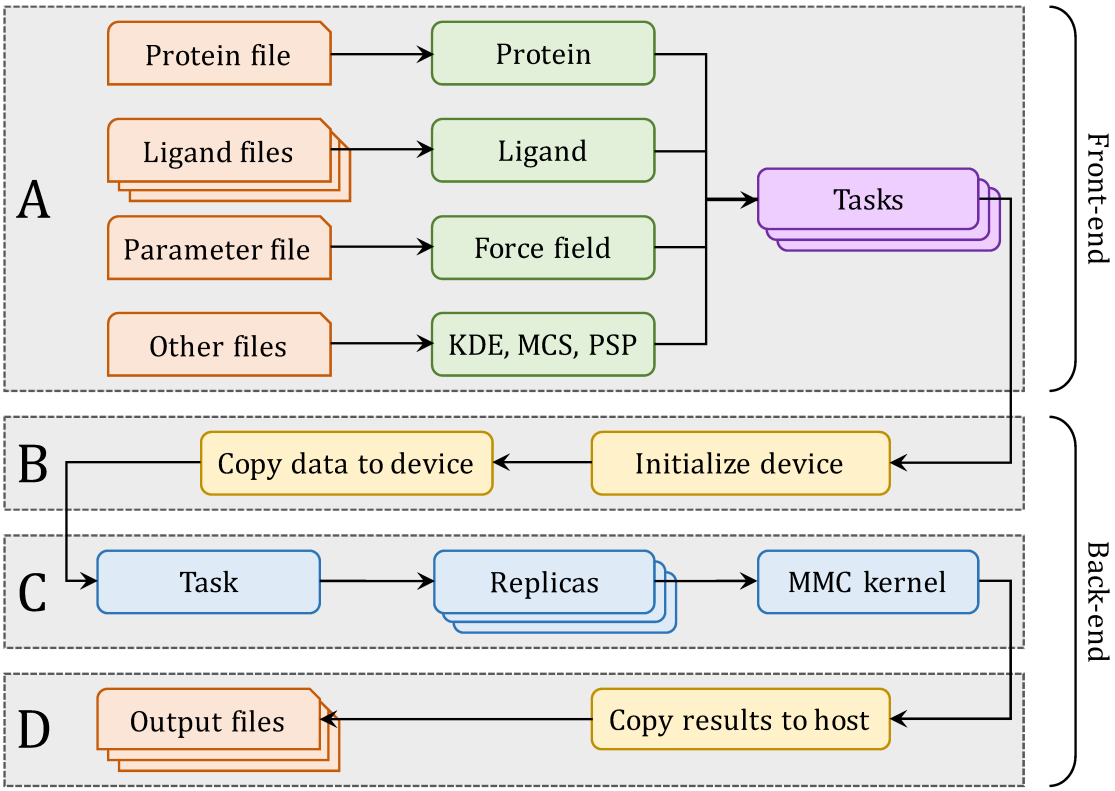

Figure 4.1: Workflow of virtual screening using GeauxDock. (A) The front-end reads input data and creates a pool of docking tasks. The back-end carries out three consecutive operations: (B) device initialization and data transfer, (C) docking calculations for individual tasks, and (D) saving output data.

object files for Xeon Phi instead of CPU. In contrast, the GPU version comprises a C++ launcher and a docking kernel implemented in CUDA. This design allows for maintaining a single front-end code and two versions of the back-end code. Compiling the source codes (Figure 4.2A) generates architecture-specific object files (Figure 4.2B), which are linked to create different versions of the binary (Figure 4.2C).

4.2.3 Parallelization Levels

GeauxDock features an enormous task-level parallelism, where different library compounds docked against the target protein correspond to individual tasks. In addition, the docking kernel exploits coarse- and fine-grained parallelism. Docking calculations for a single task involve multiple protein and ligand conformations, where each unique combination

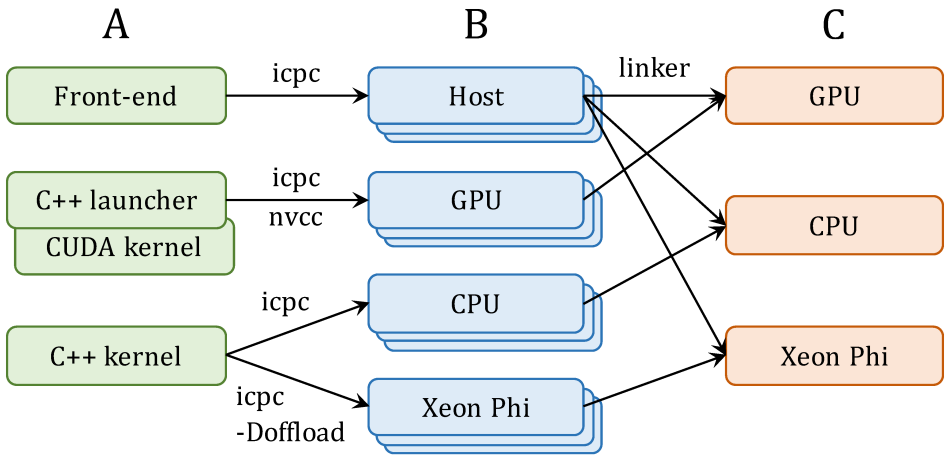

Figure 4.2: Implementation of GeauxDock. (A) The code repository is divided into three modules, a common front-end module for the CPU host and two back-end modules, one for GPU and one for CPU and Xeon Phi. (B) Compiling the source codes produces a series of architecture-specific object files. (C) Linking object files creates three binary versions for GPU, CPU and Xeon Phi.

of protein-ligand conformations is regarded as a replica of the system. Although replicas can be subjected to MMC simulations at different temperatures, only one temperature is currently used. For a given docking task, the corresponding ensembles of independent replicas are suitable for coarse-grained parallel computing. Moreover, a fine-grained parallelization takes place at the level of pairwise interactions between data points within each replica. These interactions are computed as three matrices, $protein_{ColumnVector} \times ligand_{RowVector}$ (PRT), $KDE_{ColumnVector} \times ligand_{RowVector}$ (KDE), and $MCS_{Matrix} \times ligand_{ColumnVector}$ (MCS). Here, a fairly large number of computations are subjected to fine-grained parallelization; the analysis of input data reveals up to 10^4 data points for a single replica, which is sufficient to saturate computing resources available on modern CPUs and accelerators.

Back-end calculations start when a task is fetched from the task pool. Figure 4.3 and Table 4.3 explain mapping between the docking algorithm and computing resources. First, replicas within each task are mapped to coarse-grained resources, GPU streaming multiprocessors (SMs) as well as CPU and Xeon Phi cores (Figure 4.3A and Table 4.3Name,

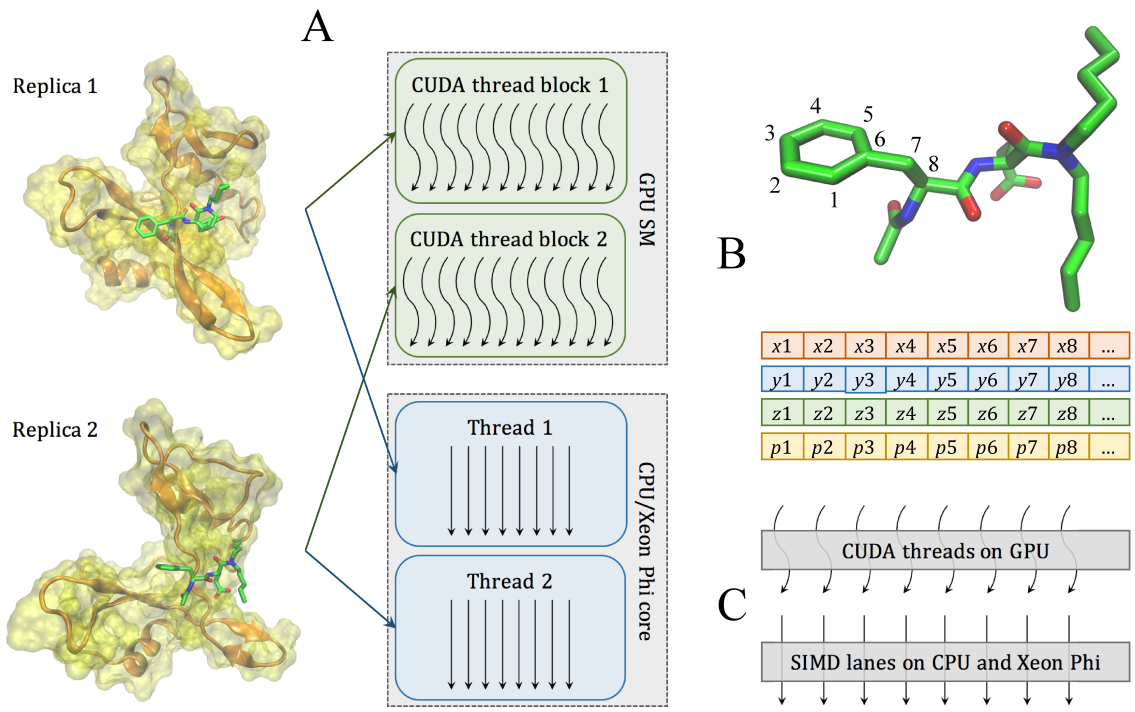

Figure 4.3: Two levels of parallelism in the docking kernel. (A) At the coarse-grained level, individual replicas are assigned to different CUDA thread blocks on GPU streaming multiprocessors (SMs) and different threads on CPU/Xeon Phi cores. (B) At the fine-gained level, data points for each replica are organized as Structure of Arrays containing Cartesian coordinates x, y, z, and parameters p associated with atoms, such as type, charge, and etc. Parameters for neighboring atoms are placed closely in memory to ensure the best execution efficiency. (C) Data points at the fine-gained level are accessed in parallel by CUDA threads on GPU and SIMD lanes on CPU and Xeon Phi.

Coarse-grained parallelism). When multiple GPUs are available, replicas within a given task are evenly assigned to the attached GPU cards. Second, interaction-level calculations (Figure 4.3B) are mapped to fine-grained resources, where computing 2D matrices utilizes SIMD lanes on CPU and Xeon Phi, and CUDA threads on GPU (Figure 4.3 Table 4.3e, Fine-grained parallelism). S1 Code illustrates loop operations on PRT, KDE, and MCS matrices involving a number of summation reductions. For instance, five energy terms calculated using the PRT matrix ($E_{ele}^{soft}, E_{vdW}^{soft}, E_{HB}, E_{CP}$ and E_{CP}^{PS}) are directly reduced from 2D array to a scalar value. Another type of reduction is hierarchical, where a 2D array $a[i][j]$ is first reduced to a 1D array $b[i]$ along the j-dimension, and then to a scalar

Table 4.3: Algorithm mapping to hardware and software models of coarse- and fine-grained parallelism in GeauxDock.

Algorithm mapping	Platform	Hardware model	Software model
Coarse-grained parallelism			
11-550 replicas	CPU	4-10 cores with 2-way multi-threading	8-20 threads
	Xeon Phi	60 cores with 4-way multi-threading	240 threads
	GPU	16 streaming multiprocessors	CUDA thread blocks
Fine-grained parallelism			
∼10,000 pairwise interactions	CPU	two 256-bit AVX SIMD instructions per cycle	8 SIMD lanes (SP)
	Xeon Phi	one 512-bit SIMD instruction per cycle	16 SIMD lanes (SP)
	GPU	192 scalar processors with multi-threading	CUDA threads

value along the i-dimension. This technique is applied to selected data across all three matrices, e.g. E_{HP} in the PRT matrix, E_{KDE} in the KDE matrix, and E_{MCS} in the MCS matrix. In order to implement hierarchical reductions on GPU, we made adjacent GPU threads efficiently exchange data by scheduling the i-dimension as the outer loop, and the j-dimension as the inner loop. Specifically, the outer (inner) loop iterates over $ligand_{RowVector}$ ($protein_{ColumnVector}$) for the PRT matrix, $ligand_{RowVector}$ ($KDE_{ColumnVector}$) for the KDE matrix, and rows of MCS_{Matrix} (columns of MCS_{Matrix}) for the MCS matrix.

2D CUDA thread blocks are responsible for calculations on GPU (Figure 4.3A, green rounded boxes). The shape and size of CUDA thread blocks are flexible and can be tuned for the optimal performance. Given that the CUDA warp size is fixed at 32, the x-dimension of the CUDA thread block is best defined as a multiple of 32. Also, the maximum number of 1,024 threads per CUDA thread block restricts the y-dimension, for example, the size of the y-dimension cannot be greater than 32 when x-dimension is 32, because 32 $\times$ 32 = 1024. However, the shapes of 2D interaction matrices do not always perfectly match those of CUDA thread blocks. For instance, the x-dimension is always greater than the y-dimension in PRT and KDE matrices, whereas a typical MCS matrix has the y-dimension

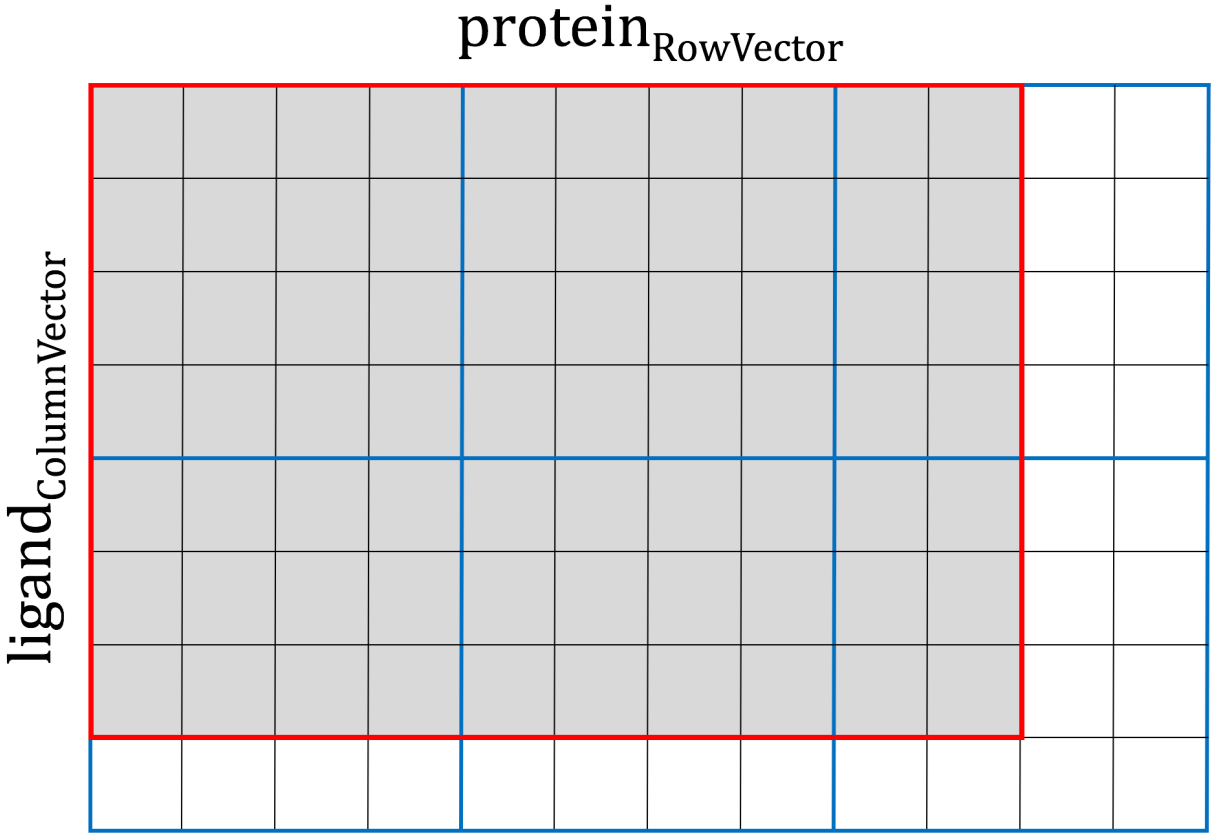

Figure 4.4: Example of parallel calculations for a data matrix. A small, 96-element matrix $ligand_{ColumnVector}$ $protein_{RowVector}$ is outlined in red, whereas the 4 × 4 CUDA thread block iterating over the matrix is outlined in blue. Here, at least 6 cycles are required to process the data matrix utilizing a total of 70 parallel threads (gray cells), while the remaining 26 threads are idle (white cells). An optimal shape of CUDA thread blocks can be constructed dynamically to improve the computational performance by reducing the number of cycles required to traverse the data matrix.

greater than the x-dimension. Therefore, boundary conditions require a careful design of CUDA thread blocks to leave a certain number of idle threads for the thread management. This procedure is illustrated in Figure 4.4, where processing a small, 70-element data matrix (outlined in red) requires at least six cycles of a 4 × 4 CUDA thread block (each cycle is outlined in blue). With this setup, 70 parallel threads are fully utilized (gray cells), leaving 26 threads idle (white cells). Overall, the number of CUDA threads is fixed at the compiling time, but the optimal shape of the thread block is defined at the runtime, when the input data become available. Here, the objective is to find the best combination of

x- and y-dimensions consuming the least amount of computing cycles to traverse the data matrix, where a computing cycle is defined as follows:

$$cycle = (ceiling(data\_size\_x/cuda\_threads\_x)) \times (ceiling(data\_size\_y/cuda\_threads\_y)) \tag{4.1}$$

In practice, only a handful of configurations are valid; we enumerate and evaluate these configurations to find the optimal solution. As an example, using Tesla K20Xm GPU with 1,024 threads per thread block, a typical configuration for PRT, KDE, MCS matrices is 128×8, 128×8, and 32×32, respectively. Different from the GPU version, the back-end for CPU implemented in C++ with OpenMP pragmas assigns processor threads to carry out computations for individual replicas (Figure 4.3A, blue rounded boxes). In order to avoid thread migration and ensure the best cache locality, the environment variable "OMP_PROC_BIND" is set to "true". In addition, inner loops in data computations iterating over $protein_{ColumnVector}$ (PRT matrix), $KDE_{ColumnVector}$ (KDE matrix), and columns of MCS_{Matrix} (MCS matrix) are marked with vector pragmas to assist Intel compiler in generating an efficient, vectorized code. Note that the same CPU code can be used on Xeon Phi since almost all performance tuning techniques for CPU apply to this accelerator as well. The major difference is that the code for Xeon Phi is required to be offloaded to the accelerator, which is conceptually similar to GPU programming. The offload is accomplished using compiler pragmas, i.e. "#pragma offload target (mic) in (data_in) out (data_out)". However, the present pragma-based Xeon Phi programming model was designed to offload a block of code to only one device. The current implementation of GeauxDock works only with a single Xeon Phi card. Although replicas could be distributed manually across multiple accelerators, one should keep in mind that at least 240 replicas are required to effectively utilize Xeon Phi. Since docking tasks have no more than 550 replicas, splitting the workload among multiple Xeon Phi cards would inadvertently decrease the overall performance. In addition, any code modification targeting the Xeon Phi platform would

complicate the code maintenance. In fact, workload sharing at the task level represents a more practical and scalable approach, which will be implemented in the future release of GeauxDock.

4.2.4 Data Structure

A docking task contains complex data, including read-only protein and ligand conformations, MMC simulation parameters, MCS, KDE and PSP force field parameters, as well as the dynamic configuration and output data from individual replicas. GeauxDock employs the Structure of Arrays (SoA) to store the data ensuring the best data locality. For example, the SoA for the ligand conformation shown as S2 Code A contains elements x[L], y[L], z[L], t[L], and c[L], representing x, y, z coordinates, the type, and electric charge for all ligand atoms, respectively. L defines the maximum number of ligand atoms and it is set at the compiling time. Figure 4.3B shows that the data associated with neighboring atoms are stored in consecutive memory addresses in order to maximize the efficiency of memory operations required for the fine-grained parallelization. With this design, CUDA threads on GPU and SIMD lanes on CPU and Xeon Phi access these data in a stride-1 pattern as illustrated in Figure 4.3C. Data structures for protein conformations, MMC simulation parameters, and PSP, KDE and MCS force field parameters are created in a similar fashion. These data constitute the first-level SoA providing read-only information, and are used as building blocks to construct the multiple-replica simulation context.

To systematically assemble replicas from these raw data, we created a data structure called "ReplicaInfo", whose purpose is to assemble a replica from the raw data using indirect references to various arrays. The concept of "ReplicaInfo" is presented in Fig 5, where two example replicas, (L_1, P_1, T_1) and (L_1, P_3, T_2), are created using indexes to the same ligand conformation (L_1), but different protein conformations (P_1 and P_3) and simulation temperatures (T_1 and T_2). ReplicaInfo was designed to yield a high computational efficiency of data exchange between replicas during parallel tempering MMC simulations [191], which requires swapping only a few indexes rather than the associated large data arrays. Further,

the ReplicaInfo structure is used to store the temporary simulation status, including energy values and ligand orientations with respect to the target protein pocket. Simulation logs are saved in the "Simlog" data structure, whose entry can also be found in ReplicaInfo. We note that the ReplicaInfo can be modified during MMC simulations, while the associated data are read-only.

```
struct Ligandconf {
  float x[L]; // x-coordinate
  float y[L]; // y-coordinate
  float z[L]; // z-coordinate
  int t[L]; // atom type
  float c[L]; // electric charge
  int center_xyz[3]; // geometric center
  int atom_number; // number of atoms
};
```

Figure 4.5: S2 Code A: Data structure for a ligand conformation (first-level Structure of Arrays)

```
struct Complex {
  Ligandconf ligandconf [MAX_LIG_CONFORMATION_NUM];
  Proteinconf proteinconf [MAX_PRT_CONFORMATION_NUM];
  Temperature temperature[MAX_TEMP_NUM];
  KDE kde;
  MCS mcs;
  PSP psp;
  ReplicaInfo replica_info[MAX_REPLICA_NUM];
  ComplexSize complexsize;
};
```

Figure 4.6: S2 Code B: Data structure for a ligand-protein complex (second-level Structure of Arrays

In addition to the first-level SoA, we designed the second-level SoA called the "Complex". S2 Code B provides the outermost container for the computation data. The elements of Complex are various data structures, including protein and ligand conformations, MMC simulation parameters, MCS, KDE and PSP force field parameters, ReplicaInfo, and the data size. Essentially, a single instance of Complex SoA and Simlog hold all data associated with a computation task. Because the memory for Complex and Simlog is allocated only

once, when either the CPU/Xeon Phi or GPU version of GeauxDock is initiated, it must be large enough to hold data for any docking tasks from the task pool. Docking calculations for the CCDC/Astex dataset require about 5 MB of memory for each Complex, whereas the entire Simlog would allocate about 1.5 GB of memory. In practice, only about 100 MB of Simlog data need to be transferred to the host and saved on disk.

4.2.5 Data Rearrangement

Irregular code patterns caused by dynamic data may significantly affect the performance. The docking kernel code contains conditional branches and indirect memory references, for example, calculating a branch path depends on the distance between a ligand atom and a protein point, which is changing in the course of MMC simulations. Although it is difficult to speed up the code containing these dependencies, we improved the code regularity for certain cases. For instance, incrementally sorting KDE data elements by the atomic type t helps improve the regularity of the conditional code `if (lig->t[index] == kde->t[index])` in a loop iterating over hundreds of KDE data points. Another example is the indirect memory reference, such as `d = array[ligand->t[index]][protein->t[index]]`. Here, sorting ligand and protein objects by t greatly improves the locality of accessing array elements. Altogether, data rearrangement enhances the performance of GeauxDock by 9.6%, 12.2% and 8.2%, on CPU, Xeon Phi and GPU, respectively.

4.2.6 Strength Reduction

In order to further speed up calculations within the docking scoring function, the strength reduction technique is applied to reduce its computation complexity. Original mathematical formulas for various energy terms in the MMC kernel are divided into pre-processing and computation groups. The pre-processing combined with data transformation is conducted within the front-end of GeauxDock. An example is shown as S3 Code, where the indirect memory reference prtconf.r[index] is removed from the original kernel (S3 Code A) and included in the pre-processing stage (S3 Code B), leading not only to a better memory locality, but also to fewer instructions in the optimized kernel. Another

technique used to accelerate computations within the docking kernel is the reduction of the arithmetic intensity. For instance, S4 Code A shows a part of the original kernel computing the soft van der Waals potential, which includes 6 loads, 9 multiplications, 3 division and 5 power functions. To speed up the MMC kernel, some calculations are either moved to the pre-processing step or executed between certain blocks of the code and then reused when calculating the potential. As the result, the optimized code shown as S4 Code B has only 2 loads, 6 multiplications, 3 divisions and no power functions.

```
Data structure:

struct ProteinConf {
  int r[P];
  int seq3[P];
  ...
}

Pre-processing:

none

Docking kernel:

int r = prtconf.r[index];
int seq3 = prtconf.seq3[r];
```

Figure 4.7: S3 Code A: Example of a data structure and the corresponding computation before strength reduction

4.2.7 Architecture Specific Optimization

The power of accelerators can be fully utilized only when time is primarily spent on computations rather than data communication. GeauxDock is implemented based on this principle by moving compute-intensive MMC simulations to Xeon Phi and GPU. S5 Code shows the MMC conformational sampling in ligand docking. First, a new configuration of a ligand is generated by randomly perturbing the present configuration. Next, the energy of the new configuration is calculated and compared to the energy of the old configuration using the Metropolis algorithm [75, 155]; the new configuration is accepted with a certain

```
Data structure:

struct ProteinConf {
  int seq3r[P];
  ...
}

Pre-processing:

seq3r[i] = prtconf.seq3[prt.r[i]]

Docking kernel:

int seq3 = prtconf.seq3r[index];
```

Figure 4.8: S3 Code B: Data structure and computation after strength reduction improving memory locality

```
Pre-processing:
none

Docking kernel (for computing the soft van der Waals potential):

float r1 = par.vdw[i][j][0];
float e1 = par.vdw[i][j][1];
float p1 = (2.0 x e1 x pow(par.lj[2] x r1, 9)) / (pow(dst, 9));
float p2 = (3.0 x e1 x pow (par.lj[2] x r1, 6)) / (pow(dst, 6));
float p4 = p1 x par.lj[0] x (1.0 + par.lj[1] x pow(dst, 2)) + 1.0f;
evdw += (p1  p2) / p4;
```

Figure 4.9: S4 Code A: Part of the docking kernel before the strength reduction

probability to be used in the next iteration, otherwise it is rejected. Even though some components of the docking kernel, such as evaluating the Metropolis criterion, are less suitable for the parallelization on GPU and Xeon Phi, this approach yields a better overall performance than offloading parts of the docking kernel. For instance, offloading only energy calculations could potentially generate an excessive communication between the host and the accelerator. In that case, advanced optimization techniques such as the asynchronous kernel execution and data copying between multiple tasks would have to be applied for a better performance. However, because extra communication is avoided in the

```
Pre-processing:

float tmp = par.lj[2] x par.vdw[i][j][0];
float e1 = par.vdw[i][j][1];
par.p1a[i][j] = 2.0f x e1 x powf(tmp, 9.0f);
par.p2a[i][j] = 3.0f x e1 x powf(tmp, 6.0f);
par_lj0 = enepara_lj[0];
par_lj1 = enepara_lj[1];

Docking kernel (reused by various code blocks):

float dst_pow2 = dst x dst;
float dst_pow4 = dst_pow2 x dst_pow2

Docking kernel (for computing the soft van der Waals potential)

float p1 = par.p1a[i][j] / (dst_pow4 x dst_pow4 x dst);
float p2 = par.p2a[i][j] / (dst_pow4 x dst_pow2);
float p4 = p1 x par_lj0 x (1.0f + par_lj1 x dst_pow2) + 1.0f;
evdw += (p1  p2) / p4;
```

Figure 4.10: S4 Code B: Part of the docking kernel after the reduction of the arithmetic intensity

MMC kernel, the code requires no further optimization of data transfer.

For GPU, the memory is carefully managed within the GeauxDock code with heavily reused variables, such as interaction distances, placed in registers. Moreover, the shared memory is used for those frequently reused data, such as ligand coordinates and energy parameters, which may have an irregular access pattern. Large arrays with the stride-1 parallel access pattern are defined as SoA, sorted for improved regularity, and saved in the global memory. Importantly, level 1 data cache on Tesla K20Xm GPU does not buffer the global memory traffic by default. The docking kernel has a good reuse pattern for PRT and KDE matrices, therefore, inserting _ldg intrinsic enables the level 1 data cache mechanisms to enhance memory operations. This technique improves the GPU performance by 4% for PRT and KDE matrices. In contrast, the cache optimization cannot be applied to computations for the MCS matrix, which have no global data reuse at all.

```
cycle_max := number of Monte Carlo cycles
C_old := old configuration
E_old := pseudo-energy of configuration C_old

  for Monte Carlo cycle c = 1 to cyclemax do
    C_new := new configuration, generated by randomly perturbing C_old
      E_new := pseudo-energy of configuration Cnew
      E_diff := E_new  E_old
      probability = exp {E_diff / temperature}
      r := random number from 0 to 1
      if ((E_diff < 0) or (r < probability)) then
          S_old := S_new
          E_old := E_new
      end if
  end for
```

Figure 4.11: S5 Code: Conformational sampling with the Metropolis Monte Carlo algorithm

Since the docking kernel invokes reduction operations, partial results in each CUDA thread need to be added to a scalar value. Here, a simple implementation stores temporary data in the shared memory, where the amount of the required memory scales linearly with the number of CUDA threads. In the early version of GeauxDock, the capacity of the shared memory limited the maximum number of CUDA threads per thread block to 768. Since using more CUDA threads per block generally delivers a better performance on Tesla K20Xm GPU, the current docking kernel uses __shfl and __shfl_xor intrinsic instructions for reduction operations. This technique enables a direct data exchange between CUDA threads without consuming the shared memory. Not only is the new reduction code 3× faster, but it also allows to use 1,024 CUDA threads per block improving the overall performance by 40%. Finally, many elementary functions, exp, log, sin, cos, etc., are frequently used in the docking kernel. The CUDA math library offers accelerated versions of these math functions [48], which are enabled by the "-use_fast_math" compiler flag. This tuning yields a 30% performance boost, however, the fast math intrinsic for GPU is not guaranteed to be fully compatible with the IEEE floating point standard. Nonetheless, a careful comparison of the results against the CPU code shows that the error rate is smaller than 0.0001%.

4.2.8 Performance Evaluation

The performance of MMC kernels in GeauxDock is evaluated on several computing platforms using diverse input data. We conducted benchmarking calculations using four Linux computers listed in Table 4.4, including a mainstream PC desktop, a PC desktop with the latest consumer grade GPU, a heterogeneous HPC cluster node with both GPU and Xeon Phi accelerators, and an HPC cluster node with two GPU cards. We set the optimization level to "-O3" with the following additional flags for the Intel compiler: "-fno-fnalias -ansi-alias -fargument-noalias" (to safely remove pointer aliases), "-ipo" (to enable inter-procedural optimization), "-vec-threshold0" (to enable vectorization whenever possible), and "-fma" (to enable the fused-multiplication-add code generation). Architectural events listed in Table 4.5 were recorded by hardware counters using the Performance Application Programming Interface (PAPI) library version 5.4.0 [192]. In addition, we implemented timers directly in the code in order to measure the execution time of an arbitrary segment of the code. We noticed that time measurements have minor fluctuations of 5%, therefore, all timings are reported as the average over 8 independent runs.

Table 4.4: Hardware and software specification of four computing platforms used to evaluate the performance of GeauxDock

Platform	Processor	Accelerator	Compiler
D1 (desktop)	1 × Intel Core i7-2600 4c, 8t, 3.4GHz, Turbo	-	Intel 14.0.2
D2 (desktop)	1 × Intel Xeon E5-2620 6c, 12t, 2.0GHz, Turbo	1 × GeForce GTX 980	GCC 4.4.7
C1 (HPC cluster)	2 × Intel Xeon E5-2680 v2 10c, 10t, 2.8GHz, Turbo	1 × Tesla K20Xm1 × Intel Xeon Phi 7120P	Intel 14.0.2 CUDA 6.5
C2 (HPC cluster)	2 × Intel Xeon E5-2670 8c, 8t, 2.6GHz, Turbo	2 × Tesla K20Xm	Intel 14.0.2 CUDA 5.5

Table 4.5: PAPI preset events used to assess the code performance.

PAPI event	Description
PAPI_LI_DCM	Number of level 1 data cache misses
PAPI_BR_MSP	Number of branch mispredictions
PAPI_TOT_INS	Total number of instructions
PAPI_TOT_CYCLES	Total number of CPU cycles

4.2.9 Benchmarking Dataset

Benchmarking calculations are carried out for a single target protein, the pp60(c-src) SH2 domain complexed with ace-malonyl Tyr-Glu-(N,N-dipentyl amine) (PDB-ID: 1a07) [193] and a set of 204 drug compounds selected from the CCDC/Astex dataset [156]. 1a07 represents a typical docking target with 344 protein effective points and an ensemble of 11 protein conformations. Depending on the number of rotatable bonds, up to 50 conformations are generated for ligands, thus the ensemble-based docking employs up to 550 replicas (11×50) of individual systems. In addition to this default protocol, we test the code scalability using a varying number of replicas at multiple temperatures. Other parameters affecting the computational complexity are the number of non-hydrogen ligand atoms and the number of points to compute the evolution-based components of the GeauxDock force field, KDE and MCS. Although both KDE and MCS scoring terms are used to calculate various restraints derived from homology rather than physical interactions, these points are iterable from the computing point of view. Therefore, KDE and MCS interacting points are equivalent to ligand atoms and protein effective points in the physics-based components of the GeauxDock force field.

4.3 Results and discussion

4.3.1 Dataset and Simulation Characteristics

[3]The distributions of the number of replicas, ligand atoms, as well as KDE and MCS points are shown in Figure 4.12. GeauxDock employs multiple replicas to account for the flexibility of protein-ligand complexes, where each replica contains a unique combination of protein and ligand conformations. The highest peak in Figure 4.12A at around 550 replicas corresponds to highly flexible compounds with multiple rotatable bonds, whereas the smaller peak at around 11 replicas represents those rigid complexes having only a single

[3]The dissertation author helped with preparing the data for benchmarking and case study, but did not write the text in the original published work

conformer. Given that the hydrogen atoms are omitted when counting atoms, the range between 6 and 62 heavy atoms presented in Figure 4.12B agrees well with the qualifying range for drug molecules according to the extended version of Lipinskis rule-of-five [194]. Because KDE points and rows in MCS_{Matrix} are calculated using template-bound ligands detected by the eFindSite algorithm [83, 195] their distributions (Figure 4.12C and 4.12D, respectively) depend on the number and size of ligands extracted from holo-templates.

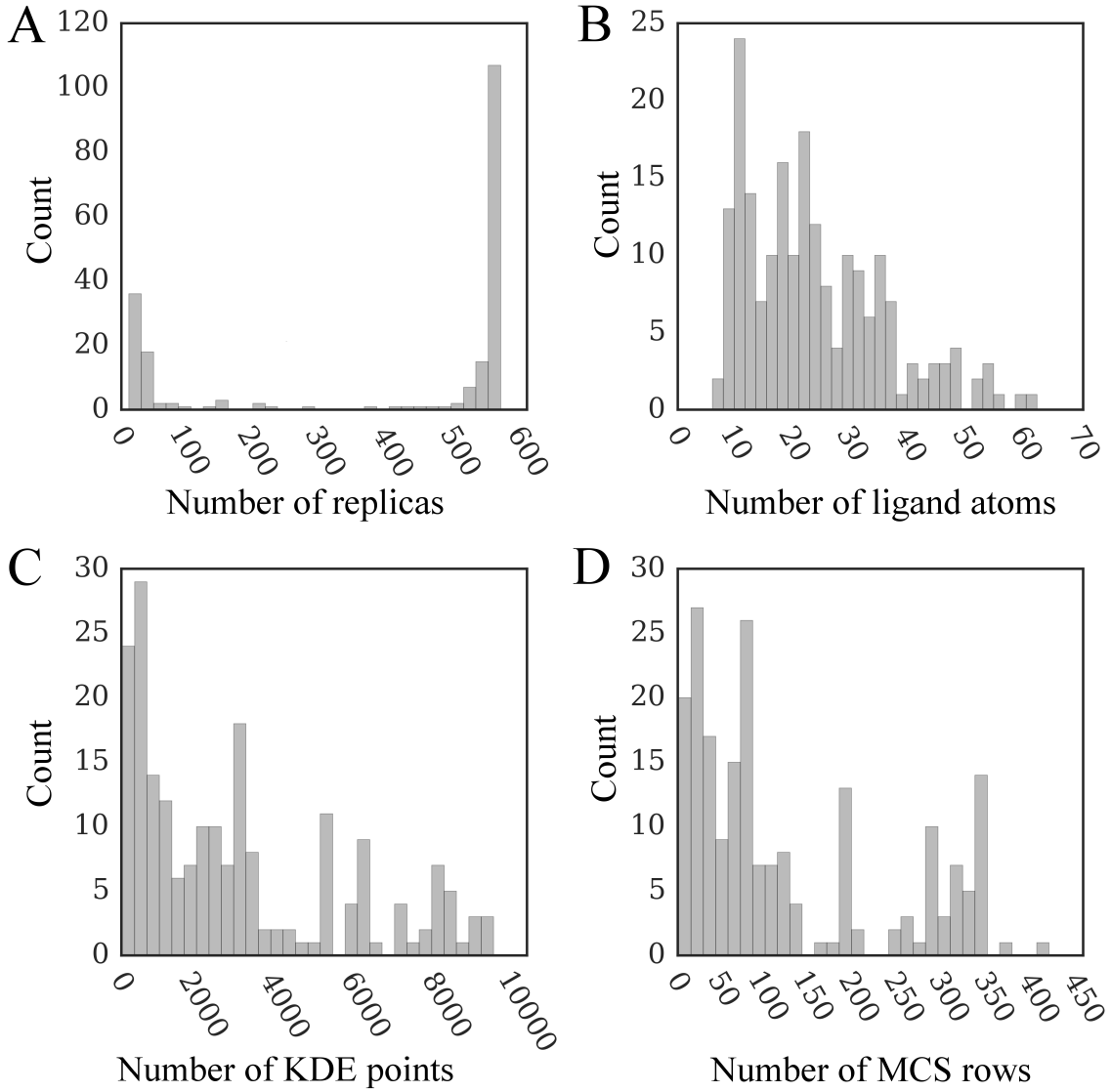

Figure 4.12: Distribution of various parameters affecting docking time. The number of (A) replicas, (B) ligand non-hydrogen atoms, (C) KDE points, and (D) rows in the MCS matrix are shown for the dataset of 204 CCDC/Astex compounds. KDE (Kernel Density Estimation) and MCS (Maximum Common Substructure) points are used to calculate evolution-based components of the docking force field.

Another important simulation parameter is the number of MMC cycles. We found that 1,000 MMC cycles is sufficient for production runs to converge. Since these calculations require 4.8 to 61 minutes on various platforms, the average wall time for the docking kernel is 1.4 seconds on the fastest machine (platform D2, Table 4.4) and 18 seconds on the slowest computer (platform D1, Table 4.4). Because the number of replicas (up to 550) is multiplied by the number of temperatures (up to 240) in our benchmarks, and several versions of the docking code needed to be tested, the time required to complete simulations could be hundreds times longer than that for production runs. Therefore, shorter simulations with 100 MMC cycles are used for benchmarking purposes.

4.3.2 Performance with an Ample Coarse-Grained Parallelism

The execution time for docking kernels includes not only computations but also time required for the data transfer to and from accelerator devices. Moreover, the kernel performance can be affected by the ensemble size (the number of replicas), because those docking systems containing rigid ligands provide insufficient coarse-grained parallelism to fully utilize computing resources. On that account, we first need to determine the ideal performance as well as a performance penalty caused by the meager coarse-grained parallelism. To address this problem, we conducted a series of simulations providing a sufficient number of replicas to deliver an ample coarse-grained parallelism. Specifically, we used 400 replicas for a dual CPU with 20 cores and 20 threads, 2,400 replicas for Xeon Phi with 60 cores and 240 threads, and 280 replicas for GPU with 14 streaming multiprocessors and 14 CUDA thread blocks.

The performance of docking kernels on CPU is assessed using the C1 computing system (Table 4.4). We first evaluate the serial performance by enabling only 1 thread on a single processor core. Using the total number of CPU cycles according to the PAPI event PAPI_TOT_CYCLES (Table 4.5) and the computing time measured by either the PAPI timer or our timer, the average dynamic CPU clock rate is 3.58 GHz $\pm$ 0.02. Figure 4.13 shows several characteristics assessing the overall computational performance of the

docking code. Computing PRT and KDE matrices are the major components of the docking kernel (Figure 4.16A and 4.16D). Since the maximum reuse distances [196] for these data (300 and 9000, respectively) are small enough to fit L1 data cache, the cache efficiency in GeauxDock is very high. Indeed, in most cases, the number number of L1 data cache misses per 103 instructions is less than 7 (Figure 4.13A), which is lower compared to a broad distribution of 5-30 misses reported for thoroughly tuned SPEC CPU2006 benchmark kernels [197] tested on the same CPU microarchitecture. Applying an additional loop tiling transformation [196] to further reduce the reuse distance does not improve the performance. Similarly, the number of branch mis-predictions per 103 instructions for the SPEC CPU2006 kernels is between 1 and 10 [197], therefore, the docking code is superior with no more than 2 branch mis-predictions (Figure 4.13B). Moreover, GeauxDock achieves an average instruction throughput rate of about 2, which is notably higher than 1.43 instructions per cycle reported for the most efficient SPEC CPU2006 kernel [197]. This comparison with the SPEC CPU2006 benchmark suite demonstrates that the serial, CPU version of the docking kernel in GeauxDock is indeed highly optimized.

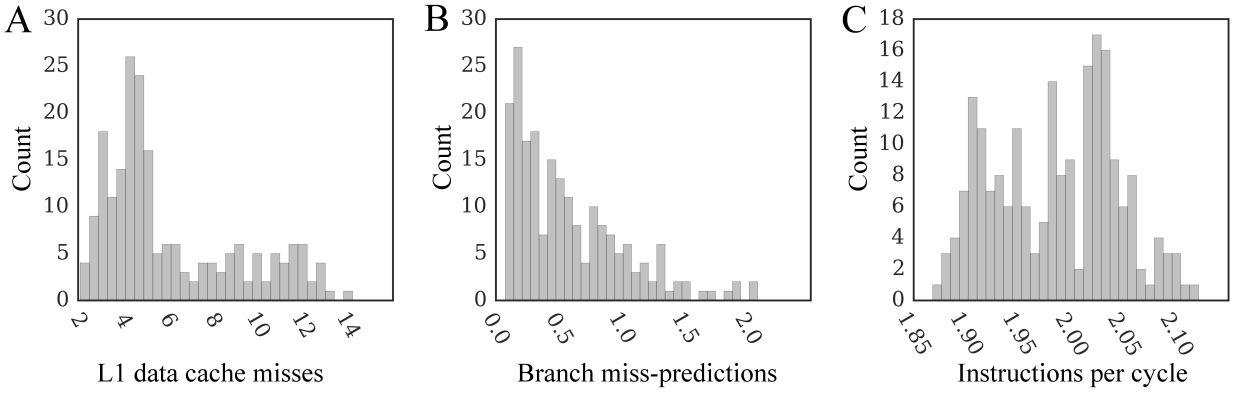

Figure 4.13: Performance characteristics for a single-threaded docking kernel on CPU. The number of (A) level 1 data cache misses per 10^3 instructions, (B) branch miss-predictions per 103 instructions, and (C) instructions per cycle.

Next, using the optimized serial CPU code as a baseline, we measure the performance of the parallel versions of GeauxDock on a dual multi-core CPU, Xeon Phi and GPU using the C1 computing system (Table 4.4). Enabling 20 threads on a dual CPU triggers the

dynamic frequency scaling and decreases the average CPU clock rate to 3.07 GHz ± 0.11. Figure 4.14A shows that the average speedup of multi-threaded GeauxDock over its serial version is 17.22×0.06, which actually corresponds to the maximum theoretical speedup accounting for the lower clock rate (20 × 3.07 GHz / 3.58 GHz). Compared to the serial code, the parallel docking kernel runs from 22× to 56× faster on Xeon Phi 7120P (Figure 4.14B) and 10× to 38× faster on Tesla K20Xm GPU (Figure 4.14C). One should bear in mind that the simulation time depends on not only the data size, but also the relative amount of PRT, KDE and MCS computations. Further, the irregular portions of the docking code are handled differently by various devices because of their architectural characteristics causing variations across the dataset. As we mentioned in the introduction section when discussing hardware design, the simpler computing units of Xeon Phi and GPU are more susceptible to dynamic branches than sophisticated CPU cores.

4.3.3 Performance of Docking Kernel on Real Data

Next, we test the parallel performance of each platform against realistic workloads. Figure 4.14D and 4.14F show that multi-threaded CPU and GPU versions of the docking kernel generally maintain their high performance on real data. In contrast, the performance of Xeon Phi is significantly affected by the lack of an ample coarse-grained parallelism (Figure 4.14E). Although the co-processor is twice as fast as a dual CPU in 71.1% of the cases (a speedup of 17× and more), Xeon Phi performs about twice as slow as a dual CPU for the remaining docking systems. This double peak pattern matches the bimodal distribution of the number of replicas shown in Figure 4.12A, demonstrating that the computational throughput of Xeon Phi is significantly affected by those workloads providing insufficient coarse-grained parallelism.

To further investigate the effect of the number of replicas on the parallel performance, we compiled a separate testing dataset comprising a single conformation of the target protein 1a07 and a rigid ligand adamantanone (PDB-ID: 5cpp) [198]. This docking system is replicated n times at different temperatures to strictly control the number of replicas in

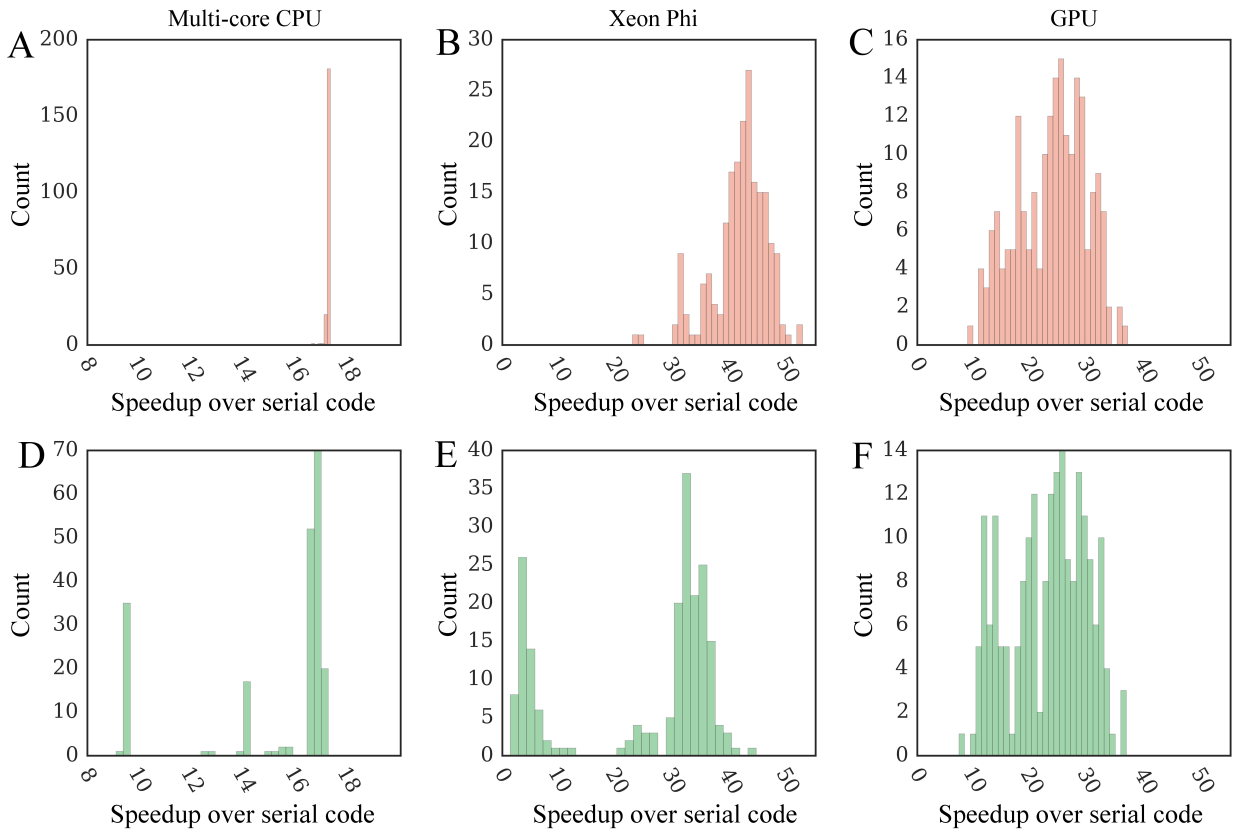

Figure 4.14: Distribution of speedups of parallel GeauxDock over the serial CPU version. Benchmarking calculations are conducted for the dataset of 204 CCDC/Astex compounds using (A-C, red) modified input data providing an ample coarse-grained parallelism and (D-F, green) unmodified input data. Three kernel implementations are tested for (A, D) multi-core CPU, (B, E) Xeon Phi, and (C, F) GPU.

docking simulations. The docking time for multi-core CPU, Xeon Phi and GPU kernels are presented in Figure 4.15. Figure 4.15A and 4.15C show sets of horizontally parallel lines with even vertical distances, whose width corresponds to the number of CPU cores and GPU streaming multiprocessors, respectively. Here, replicas are processed in parallel by independent computing units with the execution time equal to the number of replicas divided by the core count. The width of horizontal lines for Xeon Phi shown in Figure 4.15B is 240 because of the hardware multi-threading (60 cores $\times$ 4 threads per core). Clearly, it is beneficial to place 4 threads on a single core in order to fully utilize the hardware. Moreover, the kernel time for the first few data points at the beginning of each horizontal

line is somewhat shorter demonstrating that the co-processor performance is affected by the global resource contention.

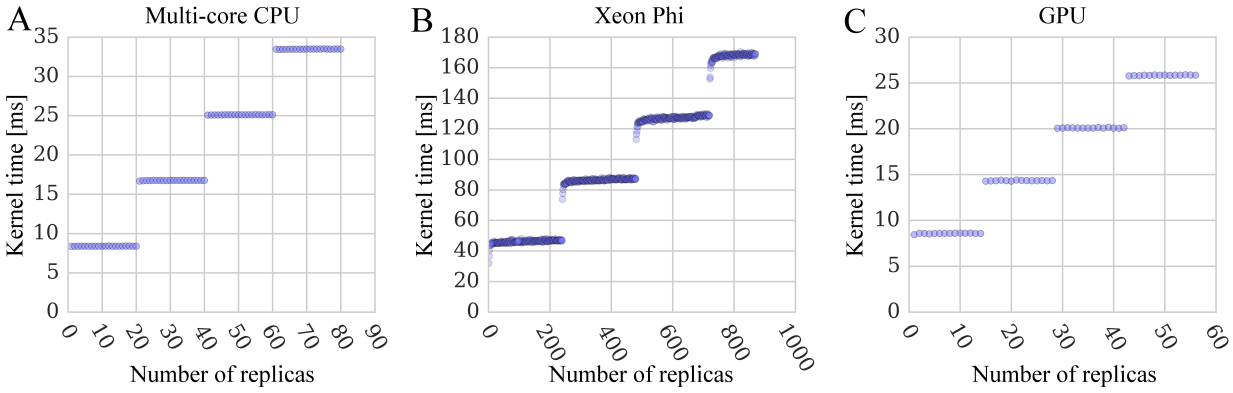

Figure 4.15: Performance scaling of docking kernels with different numbers of system replicas. Benchmarking calculations are performed using (A) multi-core CPU, (B) Xeon Phi, and (C) GPU. The width of horizontal lines is 20 replicas for a dual 10-core CPU, 240 for a 60-core Xeon Phi with 4-way multi-threading, and 14 for a 14-multiprocessor GPU.

4.3.4 A Reliable Model for the Docking Performance

To further understand the performance characteristics, we analyze various components of the docking kernel including the time spent on computing PRT, KDE, and MCS interaction matrices. KDE and MCS data are used to calculate evolution-based components of the docking force field, whereas the PRT matrix is used to calculate physics-based potentials. The time spent on computing the remaining operations is measured using a modified kernel, in which PRT, KDE, and MCS calculations are disabled. Figure 4.16 shows time contributions from these four components. Computing PRT contributes to 64.4%, 60.4%, and 32.1% of the total execution time on CPU, Xeon Phi, and GPU, respectively (Figure 4.16A-C). The percentage of the kernel time for KDE is 33.9% on CPU, 28.2% on Xeon Phi, and 46.3% on GPU (Figure 4.16D-F), whereas for MCS, it is 2.7% on CPU, 5.1% on Xeon Phi, and 10.4% on GPU (Figure 4.16G-I). The remaining operations make up about 10% of the total kernel time on Xeon Phi and GPU. In contrast, these computations require almost no time on CPU because the sophisticated processor cores handle sequential workloads (e.g. updating ligand coordinates, generating random numbers, calculating

Metropolis acceptance criterion, etc.) as efficiently as highly parallel workloads. Further, the CPU code has no data transfer between the host and the accelerator, which is required only for Xeon Phi and GPU.

Next, we analyze the correlation between the computing time and the static data size. In addition to the original docking code, we examine the performance impact of dynamic branches by forcing the calculation of all operations; this modified implementation is referred to as a "regulated" code. Figure 4.17 shows the correlation between the execution time and the data size for the original program in blue and the regulated code in red. Figure 4.17A-F demonstrate that the time required to calculate the PRT (KDE) matrix strongly correlates with its size; the coefficient of determination, R2, for the original code shown in blue is 0.996 (0.938) for CPU, 0.996 (0.987) for Xeon Phi, and 0.952 (0.981) for GPU. This correlation is somewhat weaker for the MCS matrix with the R2 of 0.957, 0.720 and 0.793 for CPU, Xeon Phi and GPU, respectively. Forcing the execution of the entire code by eliminating dynamic branches has two major effects on the kernel performance. First, it improves the correlation between the computing time and the data size, for instance, the R2 for the KDE matrix shown in red in Figure 4.17D-F is 0.999 for CPU and Xeon Phi, and 0.983 for GPU. Second, the regulated code is slower, however, the relative increase of the execution time is clearly architecture-dependent. In general, CPU skips executing most of the instructions downstream of branches because their conditional outcome can be accurately predicted, which yields a better performance (Figure 4.17A and 4.17D). The performance of GPU (Figure 4.17C and 4.17F) is unaffected by branches indicating that this accelerator always performs the predicated execution. Interestingly, the branch behavior of Xeon Phi falls between CPU and GPU. For the PRT matrix (Fig 4.17B), Xeon Phi performs the predicated execution similar to GPU, whereas the branch prediction clearly helps reduce the execution time on Xeon Phi for the KDE matrix when the KDE elements are sorted (Figure 4.17E). Nonetheless, the performance improvement for Xeon Phi is not as large as that for CPU because its computing cores are simpler and the wider SIMD

vectors are generally less suitable for irregular data.

The original code improves the performance of computing PRT and KDE, however, it negatively impacts the calculation of the MCS. This effect can be attributed to the irregularity and shape of the MCS data structure containing a dense $ligand_{ColumnVector}$, but a sparse MCS_{Matrix}. Note that since $protein_{ColumnVector}$ (Figure 4.17A-C) and $KDE_{ColumnVector}$ (Figure 4.17D-F) data structures are 1D arrays, there is a branch pattern between different elements, which can be further improved by data sorting. This pattern is lost in the sparse $MCS_{Matrix} \times ligand_{ColumnVector}$ causing a significant branch prediction penalty and longer execution times for CPU and Xeon Phi (Figure 4.17G and 4.17H). On the GPU platform, we analyzed two versions of the generated Streaming ASSembly (SASS) code. The original SASS code always performs predicated execution, while the regulated SASS code uses non-predicated instructions without testing branch conditions. For that reason, the regulated docking code performs better for the irregular MCS data.

As mentioned above, the correlation between the computing time and the size of the MCS matrix also tends to be weaker than that for PRT and KDE matrices. For instance, the R2 for the original (regulated) code shown in blue (red) in Figure 4.17G-I is 0.957 (0.946) for CPU, 0.720 (0.744) for Xeon Phi, and 0.793 (0.749) for GPU. This effect can be explained by the fact that the MCS data matrix is limited by the number of ligand atoms, which is between 6 and 62 for the CCDC/Astex dataset (Figure 4.12B). Consequently, the MCS matrix is not wide enough to efficiently utilize vector lanes on CPU (8 elements) and on Xeon Phi (16 elements) as well as the x-dimension of 2D CUDA thread blocks on GPU (32 elements); see Table 4.3. Consider a ratio of the data size and the number of cycles:

$$ratio = data\_size\_x/cycles \tag{4.2}$$

with the number of cycles required to traverse the x-dimension of the MCS matrix given

by:

$$cycles = ceiling(data\_size\_x/vector\_width\_x) \tag{4.3}$$

For PRT and KDE matrices, whose data size is much larger than the vector width, the ratio in Equation 4.2 is close to the vector width yielding a strong linear correlation between the computing time and data size. In contrast, performance fluctuations caused by idle cycles created by the underutilized vector lanes (Equation 4.3) slightly decrease the correlation for the MCS matrix.

Encouragingly, the time required to compute various interaction matrices scales linearly with the static data size. Therefore, we developed the following general linear regression model to estimate the wall clock time for the docking kernel:

$$time = w_1 PL + w_2 KL + w_3 ML + c \tag{4.4}$$

where, PL, KL, and ML are the sizes of PRT, KDE, and MCS matrices, respectively. The fitted weights and the intercept ($w_1/w_2/w_3/c$) are 7.493e-5/6.213e-6/5.121e-7/-0.025 for CPU, 2.343e-5/2.230e-6/5.937e-6/0.042 for Xeon Phi, and 4.798e-6/4.691e-6/1.783e-6/0.222 for GPU. Figure 4.18 shows that this model allows us to accurately predict the docking time from input data with the R2 of 0.974 on CPU (Figure 4.18A), 0.994 on Xeon Phi (Figure 4.18B), and 0.980 on GPU (Figure 4.18C). For those docking cases providing insufficient coarse-grained parallelism, we can further combine this linear regression with the performance model for the coarse-grained scaling (Figure 4.15). Specifically, the linear model predicts the average computing time for individual replicas assuming a sufficient coarse-grained parallelism. Since this value corresponds to the height of the first horizontal bar in Figure 4.15, we can estimate the execution time of a real task using the number of replicas and the repeating pattern of the regulated code.

4.3.5 Comparative Benchmarks of Platforms

Finally, we perform comparative benchmarks of all computing platforms listed in Table 4.4 using the 1a07 target protein and the dataset of 204 CCDC/Astex ligands. In these simulations, we use the original GeauxDock code and the real data with respect to the number of protein and ligand conformations. Timing reports include the total execution time of the docking kernel for 204 tasks and the simulation wall time averaged over 8 independent docking runs for each task. GeauxDock is specifically designed for virtual screening applications, therefore, it reads the target protein input data only once for a given set of docking ligands. Indeed, GeauxDock spends from 95.4% (GeForce GTX 980) to 99.7% (Xeon E5-2680 v2) of the total time executing docking kernels, while loading and pre-processing input data take only about 10 seconds on average (Table 4.6 and 4.7). The reference time required to complete docking calculations for the entire dataset is 61.31 minutes using a multi-threaded CPU version running on Core i7-2600 multi-core CPU (platform D1, Table 4.4). Figure 4.20 shows that high-performance servers and hardware accelerators yield significant speedups over a mainstream PC desktop. GeForce GTX 980 is the fastest computing device in our tests, which achieves a 12.6× speedup and dramatically reduces the wall time to only 4.84 minutes. Xeon Phi gives a 6.8× speedup corresponding to the wall time of 9.00 minutes, whereas the performance of a single Tesla K20Xm card with 11.14 minutes of wall time is about 23% worse than Xeon Phi. It is noteworthy that we obtained almost a perfect scaling on multiple GPU cards; using a pair of K20Xm GPUs increases the performance by 98%, compared with a single K20Xm GPU. A dual Xeon E5-2680 CPU needs 16.99 minutes to complete docking calculations, which is about 3.6 faster than the baseline i7-2600 CPU running at a higher clock rate.

One should keep in mind that not only the theoretical peak performance, but also the cost and the energy consumption vary greatly for the testing platforms, particularly between consumer and server grade hardware (Table 4.6 and 4.7). For instance, a single Core i7 2600 is 12 less expensive and requires 59% less energy than a dual Xeon E5-

Table 4.6: Benchmarking data for docking simulations conducted for the CCDC/Astex dataset using various computing devices (first part).

Computing device	Total wall time (kernel time) [min]	Theoretical peak performance [GFLOPS]
1 x Core i7-2600 (platform D1)	61.31 (61.15)	224
2 x Xeon E5-2680 v2 (platform C1)	16.99 (16.86)	992
1 x Xeon Phi 71200P (platform C1)	9.00 (8.79)	2553
1 x Tesla K20Xm (platform C1)	11.14 (11.01)	3936
2 x Tesla K20Xm (platform C2)	5.61 (5.46)	7872
1 x GeForce GTX 980 (platform D2)	4.84 (4.62)	4980

2680 CPU, whereas GeForce GTX 980 is more than 5 lower priced and requires 27% less energy than Tesla K20Xm. For that reason, in addition to evaluating a pure computational performance, we analyze the performance with respect to the energy consumption and hardware cost. GeForce GTX 980 systematically outperforms other computing platforms, for example, it gives a benefit of 6.5× per dollar and 7.3× per watt compared to the reference D1 platform (Figure 4.20). This remarkable performance results from mapping massively parallel computations and data structure to the GPU architecture. According to vendor specifications, GeForce GTX 980 has a higher core utilization and better energy efficiency than the previous generation Tesla K20Xm. Its streaming multiprocessors have two-thirds of the number of scalar processors of Tesla K20Xm, yet the number of registers is the same. Moreover, the size of the shared memory on GeForce GTX 980 is twice as large as that on Tesla K20Xm. Therefore, extra efforts were devoted to tune the CUDA docking kernel in order to take advantage of the abundant resources on GeForce GTX 980. The performance per dollar of K20Xm GPU is comparable to a server grade Xeon E5-2680 CPU and Xeon Phi 7120P, but it is 2× lower than a consumer grade Core i7 processor. Due to advances in the semiconductor technology constantly improving the energy efficiency, the performance per watt of a server grade hardware (Xeon E5 CPU, Xeon Phi and K20Xm)

Table 4.7: Benchmarking data for docking simulations conducted for the CCDC/Astex dataset using various computing devices (second part).

Computing device	Power consumption [watt]	Price [US dollar]
1 x Core i7-2600 (platform D1)	95	283
2 x Xeon E5-2680 v2 (platform C1)	230	3440
1 x Xeon Phi 71200P (platform C1)	300	4129
1 x Tesla K20Xm (platform C1)	225	3000
2 x Tesla K20Xm (platform C2)	450	6000
1 x GeForce GTX 980 (platform D2)	165	550

is about twice as high as that for an inexpensive, yet two years older Core i7 processor.

4.3.6 Case Study

To demonstrate how GeauxDock samples the conformational space when searching for native conformations, in Figure 4.21, we present docking trajectories for several representative examples. In addition to the target complex 1a07 used in the profiling and benchmarking of parallel GeauxDock, we performed docking simulations of glutathione to glutathione S-transferase (PDB-ID: 1aqw) [199], and a non-peptidyl, active site-directed inhibitor LY178550 to human α-thrombin (PDB-ID: 1d4p) [200]. Docking ligands were initialized at random orientations within target binding pockets to mimic a real application, where the native conformations are unknown. Solid lines in Figure 4.21A show the trajectories of the pseudo-energy E_1, E_2 and E_3 for 1a07, 1aqw and 1d4p, respectively. In all cases, the MMC sampling reached low-energy states with the fastest convergence for E_3. On the other hand, pseudo-energy variations for E_1 and E_2 are smaller compared to E_3, suggesting that the underlying energy surfaces for 1aqw and 1d4p are smoother.

In general, the convergence of molecular docking simulations is complicated by the fact that a large fraction of the search space may be sterically forbidden [180] and sophisticated scoring functions are often too sensitive to conformational changes in the binding

regions [201]. To further investigate docking trajectories, we calculated the RMSD for each accepted MMC step during the docking process of 1a07. Encouragingly, the dashed black line in Figure 4.21A shows that the RMSD decreases with the decreasing pseudo-energy owing to the fact that both quantities are strongly correlated (Figure 4.21B). Altogether, these results demonstrate that the scoring function in GeauxDock effectively drives docking simulations toward native-like conformations.

4.3.7 Comparison with Other Docking Software

Finally, in order to compare the docking accuracy of GeauxDock to the state-of-the-art, we performed benchmarking calculations of GeauxDock and AutoDock Vina [72] against the PDBbind dataset [202]. Here, we selected a set of 158 proteins whose length is below 600 residues. We ran both programs with the default parameters using randomized starting conformations of the docking ligands. The docking box for Vina was set to an optimal size based on the radius of gyration of query compounds, which was demonstrated to maximize docking accuracy [89]. First, we carried out a classical self-docking experiment, where the ligand is re-docked to the experimental protein structure co-crystalized with that compound. The geometric center of a ligand bound in the experimental complex structure was used as the binding pocket center for both programs. Docking accuracy is assessed by the RMSD calculated over ligand heavy atoms. Figure 4.22 (Self-docking) shows that the median ligand RMSD across the PDBbind dataset is 2.03 Åfor Vina and 2.43 Åfor GeauxDock. A p-value of 0.52 calculated by the Mann-Whitney U test demonstrates that the performance difference between Vina and GeauxDock in self-docking is statistically insignificant.

GeauxDock was designed to work with not only experimental structures, but also computer-generated models. Therefore, in addition to the self-docking experiment, we used both programs to dock ligands to the homology models of target proteins. Specifically, we constructed protein models for the PDBbind dataset using templates detected by HHsuite [203], whose sequence identity to the target is ¡70%. Moreover, in the model-

docking experiment, we employed binding sites identified by eFindSite [83, 195], so that ligand docking is performed solely with the predicted structural data. This dataset is clearly more challenging than that used in self-docking because of structural imperfections in the modeled target sites; the average heavy-atom RMSD calculated over binding site residues is 2.51Å±1.62. In addition, binding site locations are predicted with an average distance of 2.48Å±1.57 from the experimental pocket center. As expected, Figure 4.22 (Model-docking) shows that the median RMSD values for ligands docked by both programs tend to be higher than those obtained in the former experiment. Compared to self-docking, the median ligand RMSD for Vina increased by 4.30Åto 6.33 Å. However, the median RMSD for GeauxDock is 4.77 Å, thus, it has increased only by 2.34 Å, a value that roughly corresponds to the structural distortions of target binding sites. Further, the p-value between both docking programs reported by the Mann-Whitney U test is now 0.00025 clearly demonstrating that GeauxDock significantly outperforms Vina in ligand docking against protein models.

4.4 Conclusions

[4]In this communication, we discuss the optimization of a molecular docking code, GeauxDock. GeauxDock features a novel scoring function and Monte Carlo-based conformational space sampling and it is designed for large-scale virtual screening applications using heterogeneous computer architectures. Because of its modular code framework, GeauxDock supports modern multi-core CPU, as well as Xeon Phi and GPU accelerators. Considerable efforts were devoted to minimize the data communication leading to at least 95% of the time spent on executing MMC kernels. Further, we applied various tuning techniques to significantly accelerate the docking kernel based on the performance characteristics obtained by a meticulous code profiling using diverse input data. For instance, a systematic optimization of the serial CPU code brought about not only

[4]The dissertation author helped with making the conclusions, but did not write the text in the original published work

a 6.5× speedup on a single computing core, but also a perfect scaling with the number of cores on modern shared-memory platforms equipped with multiple sockets of multi-core CPUs. Docking benchmarks conducted on many-core accelerators show that using Xeon Phi 7120P yields 1.9× performance improvement over a dual-socket Xeon E5 CPU, whereas the fastest GPU, GeForce GTX 980, achieves a 3.5× speedup over a dual CPU. It is important to note that in addition to hardware capabilities, a thorough code tuning for accelerator devices plays an important role in increasing the computational performance. For example, an early version of the GeauxDock code running on Tesla K20Xm was about 30% slower than a dual-socket Xeon E5 CPU, but after employing GPU intrinsic instructions, we were able to make K20Xm 53% faster. In addition to the evaluation of a purely computational performance, we examined the energy consumption and hardware costs. In conclusion, heterogeneous computing platforms, especially those equipped with the latest GPU cards, offer significant advantages over traditional CPU-based systems. Using parallel codes optimized for modern heterogeneous HPC architectures can significantly accelerate structure-based virtual screening applications. GeauxDock is open-sourced and publicly available from our website at `www.brylinski.org/geauxdock` and `http://www.institute.loni.org/lasigma/package/dock/`.

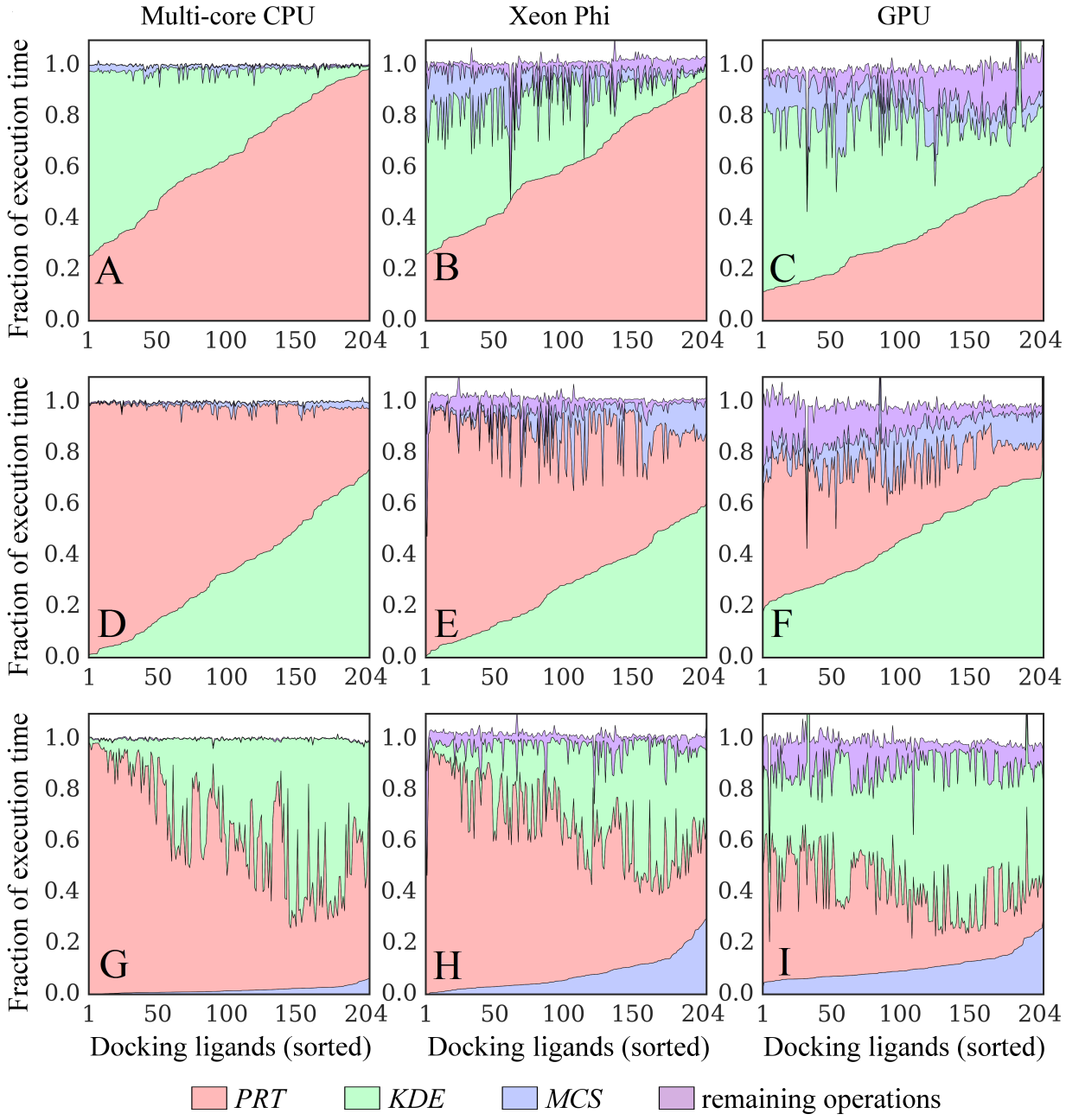

Figure 4.16: Time breakdowns for docking kernels running on different platforms. Kernel implementations for (A, D, G) multi-core CPU, (B, E, H) Xeon Phi, and (C, F, I) GPU are tested. Three major operations compute the following interaction matrices: $protein_{ColumnVector} \times ligand_{RowVector}$ (PRT, green), $KDE_{ColumnVector}\ ligand_{RowVector}$ (KDE, red), and $MCS_{Matrix}\ ligand_{ColumnVector}$ (MCS, blue). Purple areas correspond to the remaining operations. KDE (Kernel Density Estimation) and MCS (Maximum Common Substructure) points are used to calculate evolution-based components of the docking force field, whereas the PRT matrix is used to calculate the majority of physics-based potentials. Results collected for the dataset of 204 CCDC/Astex compounds are sorted on the x-axis with respect to increasing time of computing (A, B, C) PRT, (D, E, F) KDE, and (G, H, I) MCS matrices.

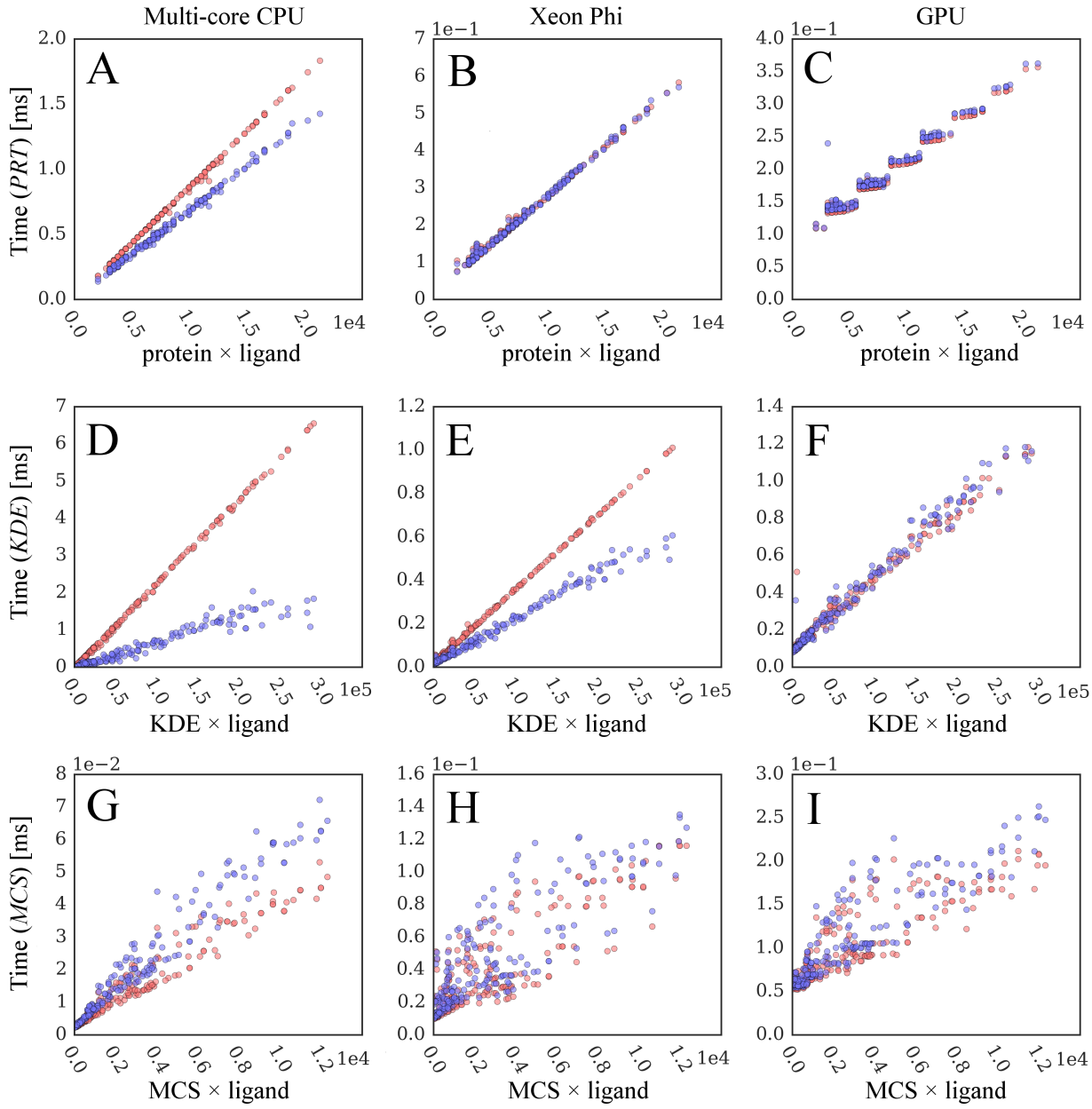

Figure 4.17: Correlation between computing time and static data size. Blue points are collected from original GeauxDock, whereas red points correspond to a modified docking code, where dynamic branches are turned off forcing the execution of all instructions. Three major operations compute (A-C) $protein_{ColumnVector}$ $ligand_{RowVector}$ (PRT), (D-F) $KDE_{ColumnVector}$ × $ligand_{RowVector}$ (KDE), and (G-I) MCS_{Matrix} $ligand_{ColumnVector}$ (MCS) matrices. Three kernel implementations are tested for (A, D, G) multi-core CPU, (B, E, H) Xeon Phi, and (C, F, I) GPU.

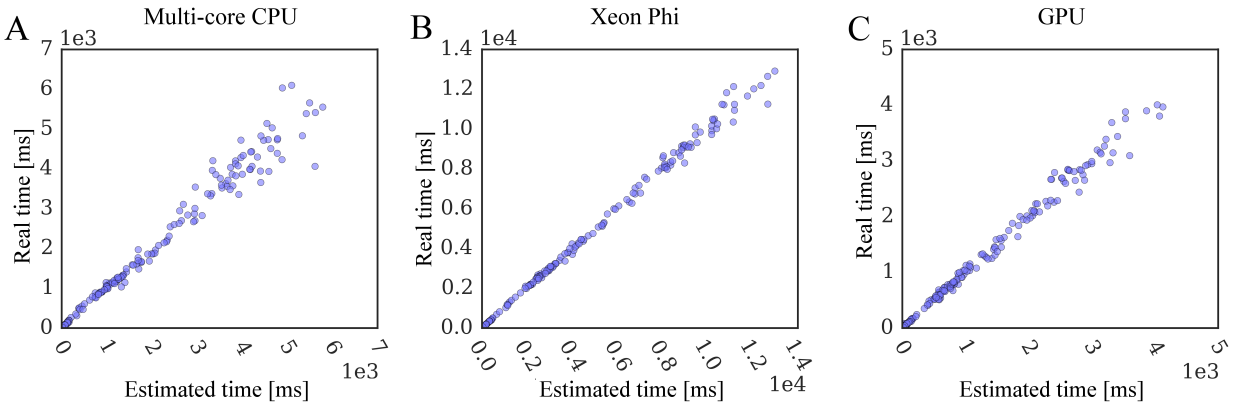

Figure 4.18: Correlation between the estimated and real docking time. Simulation time is estimated from static data size using a general linear regression model for (A) multi-core CPU, (B) Xeon Phi, and (C) GPU.

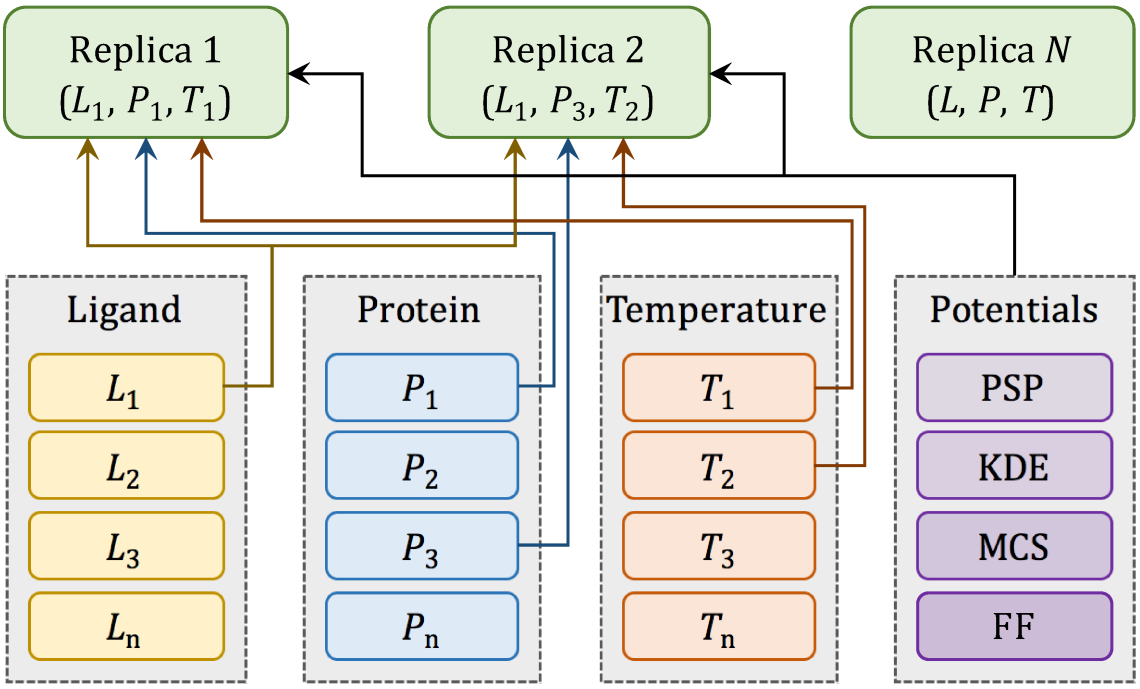

Figure 4.19: Data indexing for multi-replica Monte Carlo simulations. Individual replicas are multi-dimensional objects comprising different combinations of ligand (L) and protein (P) conformations, and temperatures (T), as well as the same set of PSP, KDE, MCS potentials and force field (FF) parameters. All these data are read-only, labeled with tags, and accessible through indexes as depicted by arrows.

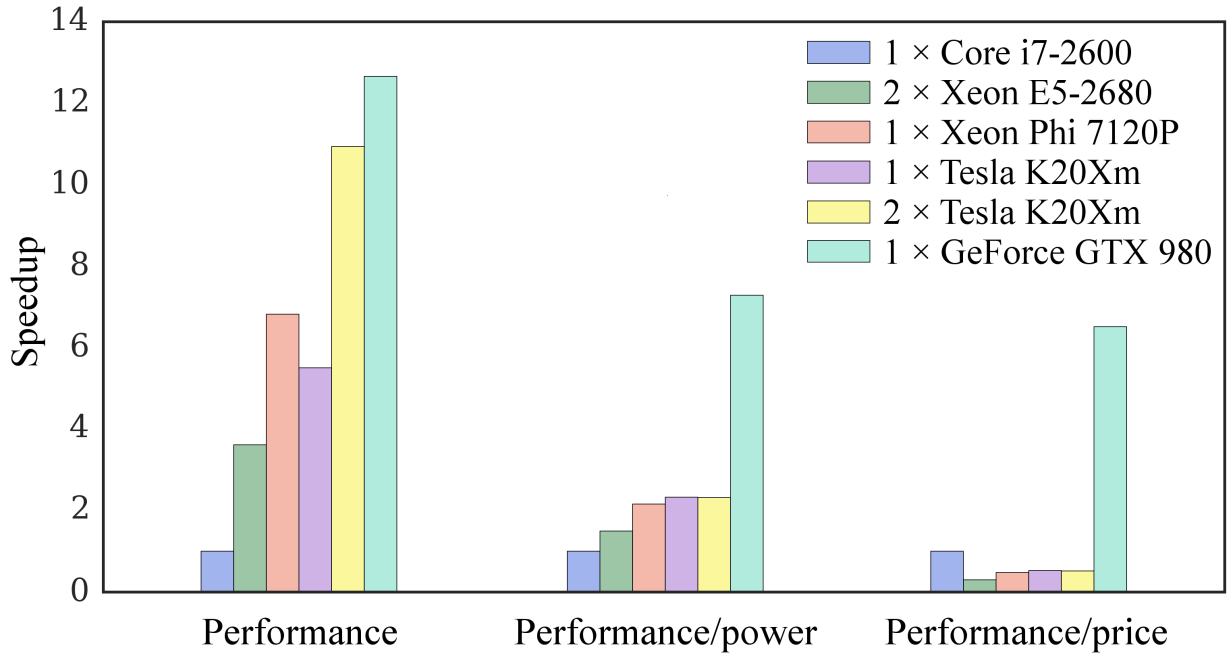

Figure 4.20: Benchmarks of GeauxDock against the CCDC/Astex dataset. Three measures are included, a pure computational performance, the performance divided by the energy consumption, and the performance divided by the hardware cost. Measurements for different platforms are normalized by the performance of Core i7-2600 CPU.

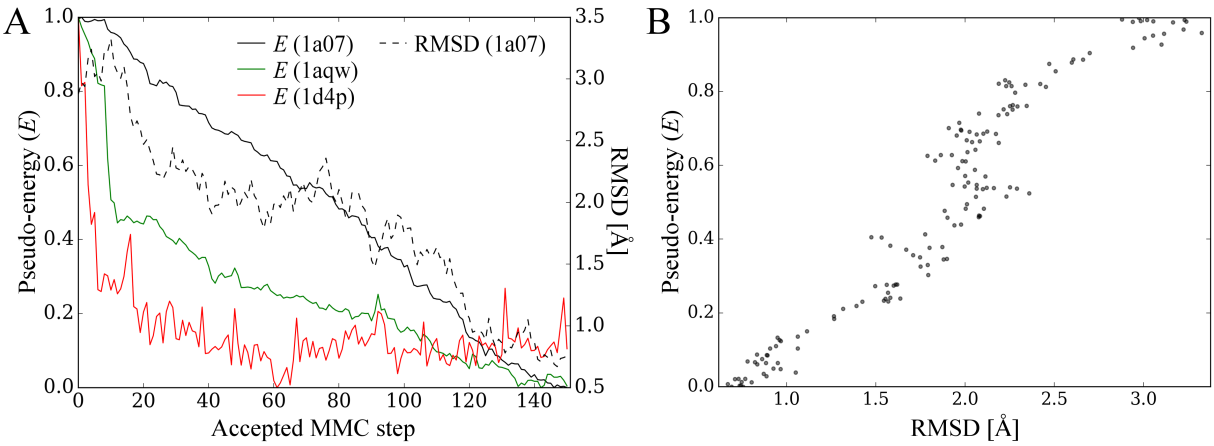

Figure 4.21: Examples of docking calculations using GeauxDock. Three cases are presented, a peptide ligand and C-src tyrosine kinase (PDB-ID: 1a07, black), glutathione and glutathione S-transferase (PDB-ID: 1aqw, green), as well as LY178550 and human -thrombin (PDB-ID: 1d4p, red). (A) Solid lines show the pseudo-energy plotted as a function of the accepted Metropolis Monte Carlo (MMC) step; a trajectory of the RMSD is plotted for 1a07 (dashed black line). (B) Scatter plot of the RMSD and pseudo-energy for 1a07.

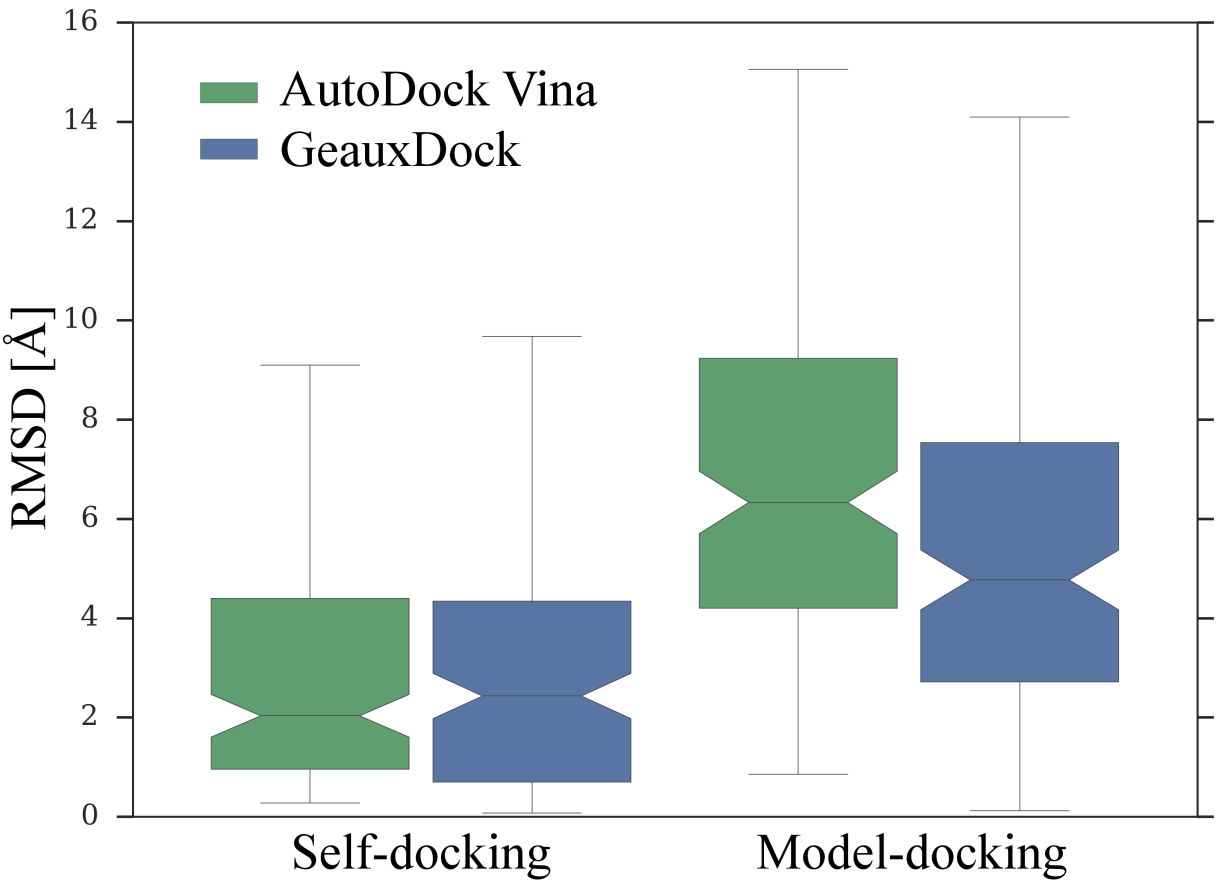

Figure 4.22: Docking accuracy of AutoDock Vina and GeauxDock on the PDBbind dataset. The performance is assessed by ligand heavy-atom RMSD calculated against experimental binding poses. A horizontal line inside each box is the median, boxes end at the first and the last quartile, and the whiskers span the distribution range of 10-90%. Two boxes on the left correspond to the self-docking experiment, whereas two boxes on the right are calculated for docking benchmarks against homology models.

Chapter 5
SUMMARY

Molecular docking is a promising technique that can make use of the vast amount of experimental data in structural biology for the purpose of rational drug design. This thesis summons up my original research work on several aspects in this field.

I developed the Contact Mode Score, a metric that quantifies the conformational similarity of protein-ligand complexes based on intermolecular contacts. Compared with the traditional root-mean-square deviation, its advantages include mitigating the dependence on the ligand size and taking into account the protein environment. I further developed the eXtended Contact Mode Score that capitalizes on the conservation of ligand binding across structurally similar pocket occupied by chemically similar ligands. It can be applied to evaluate predicted structures from molecular docking, where a retrospective assessment cannot be performed because the experimental structures of the majority of complexes are unavailable. The eXtended Contact Mode Score sets a typical example of using the readily available template data in structural biology.

I developed GeauxDock docking engine, a molecular docking approach featuring a novel descriptor-based scoring function and a mixed-resolution description of protein-ligand complexes. Benchmarks demonstrate that GeauxDock is capable of recognizing native-like binding modes with the area under ROC of 0.85. The scoring function of GeauxDock incorporates two distinct types of energy terms, physics- and evolution-based. The latter are derived from evolutionary related complex structures, and their strength depends on the level of homology between the target and template systems. In that regard, this new approach is able to take advantage of the increasingly accumulating protein structural data.

Through collaboration, the GeauxDock docking engine was implemented onto modern parallel computing architectures. The program supports multi-core CPU, as well as Xeon Phi and GPU accelerators. High parallel efficiency has been achieved that 95% of the

computing time is spent on executing the Monte Carlo kernels. In addition to the evaluation of a purely computational performance, we also examined the energy consumption and hardware costs. In conclusion, heterogeneous computing platforms, especially the ones equipped with the latest GPU cards, provide significant advantages over traditional CPU-based systems in processing large scale molecular docking simulations.

REFERENCES

[1] C. P. Chen and C.-Y. Zhang. *Data-intensive applications, challenges, techniques and technologies: A survey on Big Data.* Information Sciences **275**, 314 (2014).

[2] M. A. Waller and S. E. Fawcett. *Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management.* Journal of Business Logistics **34**, 77 (2013).

[3] P. Simon. *Too Big to Ignore: The Business Case for Big Data*, volume 72. John Wiley & Sons (2013).

[4] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, *et al. BigBench.* In *Proceedings of the 2013 international conference on Management of data - SIGMOD '13*, volume 36, p. 1197. ACM Press, New York, New York, USA (2013).

[5] J. Dean and S. Ghemawat. *MapReduce: simplified data processing on large clusters.* Communications of the ACM **51**, 107 (2008).

[6] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. *The hadoop distributed file system.* In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1–10. IEEE (2010).

[7] J. Lin and A. Kolcz. *Large-scale machine learning at twitter.* In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 793–804. ACM (2012).

[8] F. Provost and T. Fawcett. *Data Science and its Relationship to Big Data and Data-Driven Decision Making.* Big Data **1**, 51 (2013).

[9] T. Hey, S. Tansley, K. M. Tolle, *et al. The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA (2009).

[10] D. Gannon and D. A. Reed. *Parallelism and the cloud.* (2009).

[11] F. F. Costa. *Big data in biomedicine.* Drug Discovery Today **19**, 433 (2014).

[12] J. Gemmell, G. Bell, and R. Lueder. *MyLifeBits: a personal database for everything.* Communications of the ACM **49**, 88 (2006).

[13] *Australia Square Kilometre Array of radio telescope project.* `http://www.ska.gov.au/Pages/default.aspx`.

[14] *The Large Hadron Collider (LHC).* `http://home.cern/topics/large-hadron-collider`.

[15] *Panoramic Survey Telescope Rapid Response System.* `http://pan-starrs.ifa.hawaii.edu/public/`.

[16] J. R. Hunt, D. D. Baldocchi, and C. van Ingen. *Redefining ecological science using data.* (2009).

[17] Y. Katoh and M. Katoh. *Hedgehog signaling pathway and gastrointestinal stem cell signaling network (review).* International journal of molecular medicine **18**, 1019 (2006).

[18] C. H. Schilling, S. Schuster, B. O. Palsson, and R. Heinrich. *Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era.* Biotechnology progress **15**, 296 (1999).

[19] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, *et al. Open Babel: An open chemical toolbox.* Journal of cheminformatics **3**, 1 (2011).

[20] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, *et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics.* Bioinformatics **25**, 1422 (2009).

[21] V. Marx. *Biology: The big challenges of big data.* Nature **498**, 255 (2013).

[22] J. Gray. *Jim Gray on eScience: A transformed scientific method.* The fourth paradigm: Data-intensive scientific discovery **1** (2009).

[23] D. Simberloff, B. Barish, K. Droegemeier, D. Etter, N. Fedoroff, *et al. Long-lived digital data collections: enabling research and education in the 21st century.* National Science Foundation (2005).

[24] `https://ncar.ucar.edu/`.

[25] `http://www.sdsc.edu/`.

[26] `https://www.dnanexus.com/company`.

[27] J. Grimmer. *We are all social scientists now: how big data, machine learning, and causal inference work together.* PS: Political Science & Politics **48**, 80 (2015).

[28] B. L. Monroe, J. Pan, M. E. Roberts, M. Sen, and B. Sinclair. *No! Formal theory, causal inference, and big data are not contradictory trends in political science.* PS: Political Science & Politics **48**, 71 (2015).

[29] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. *Machine learning: An artificial intelligence approach.* Springer Science & Business Media (2013).

[30] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. *Machine learning: a review of classification and combining techniques.* Artificial Intelligence Review **26**, 159 (2006).

[31] E. Mjolsness and D. DeCoste. *Machine Learning for Science: State of the Art and Future Prospects.* Science **293**, 2051 (2001).

[32] Ž. Ivezić, A. J. Connolly, J. T. VanderPlas, and A. Gray. *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data.* Princeton University Press (2014).

[33] S. Whiteson and D. Whiteson. *Machine learning for event selection in high energy physics.* Engineering Applications of Artificial Intelligence **22**, 1203 (2009).

[34] P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach.* MIT press (2001).

[35] J. Robertson, D. DeHart, K. M. Tolle, and D. Heckerman. *Healthcare delivery in developing countries: challenges and potential solutions.* (2009).

[36] M. Shiozawa. *Reconstruction algorithms in the Super-Kamiokande large water Cherenkov detector.* Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **433**, 240 (1999).

[37] L. W. Hahn, M. D. Ritchie, and J. H. Moore. *Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions.* Bioinformatics **19**, 376 (2003).

[38] E. Mjolsness, R. Castano, T. Mann, and B. Wold. *From coexpression to coregulation: an approach to inferring transcriptional regulation among gene classes from large-scale expression data* (2000).

[39] P. GinsParG. *Text in a data-centric world.* (2009).

[40] P. W. Holland. *Statistics and causal inference.* Journal of the American statistical Association **81**, 945 (1986).

[41] M. C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, *et al. Learning to recognize volcanoes on Venus.* Machine Learning **30**, 165 (1998).

[42] M. P. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, *et al. Knowledge-based analysis of microarray gene expression data by using support vector machines.* Proceedings of the National Academy of Sciences **97**, 262 (2000).

[43] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, *et al. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing.* IEEE Transactions on Parallel and Distributed Systems **22**, 931 (2011).

[44] E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P. Nolan. *Computational solutions to large-scale data management and analysis.* Nature Reviews Genetics **11**, 647 (2010).

[45] P. A. Miranda, A. X. Falcão, and J. K. Udupa. *Cloud bank: A multiple clouds model and its use in MR brain image segmentation.* In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 506–509. IEEE (2009).

[46] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. *A high-performance, portable implementation of the MPI message passing interface standard.* Parallel computing **22**, 789 (1996).

[47] L. Dagum and R. Menon. *OpenMP: an industry standard API for shared-memory programming.* IEEE computational science and engineering **5**, 46 (1998).

[48] C. Nvidia. *Compute unified device architecture programming guide* (2007).

[49] J. J. Irwin and B. K. Shoichet. *ZINC-a free database of commercially available compounds for virtual screening.* Journal of chemical information and modeling **45**, 177 (2005).

[50] J.-L. Reymond and M. Awale. *Exploring chemical space for drug discovery using the chemical universe database.* ACS chemical neuroscience **3**, 649 (2012).

[51] P. Ripphausen, B. Nisius, L. Peltason, and J. Bajorath. *Quo vadis, virtual screening? A comprehensive survey of prospective applications.* Journal of medicinal chemistry **53**, 8461 (2010).

[52] P. D. Karp, B. Berger, D. Kovats, T. Lengauer, M. Linial, *et al. ISCB Ebola Award for Important Future Research on the Computational Biology of Ebola Virus.* PLoS Comput Biol **11**, e1004087 (2015).

[53] K. M. Merz Jr, D. Ringe, and C. H. Reynolds. *Drug design: structure-and ligand-based approaches.* Cambridge University Press (2010).

[54] G. Schneider and U. Fechner. *Computer-based de novo design of drug-like molecules.* Nature Reviews Drug Discovery **4**, 649 (2005).

[55] D. B. Kitchen, H. Decornez, J. R. Furr, and J. Bajorath. *Docking and scoring in virtual screening for drug discovery: methods and applications.* Nature reviews Drug discovery **3**, 935 (2004).

[56] A. N. Jain. *Virtual screening in lead discovery and optimization.* Current opinion in drug discovery & development **7**, 396 (2004).

[57] S. Ghosh, A. Nie, J. An, and Z. Huang. *Structure-based virtual screening of chemical libraries for drug discovery.* Current opinion in chemical biology **10**, 194 (2006).

[58] B. O. Villoutreix, R. Eudes, and M. A. Miteva. *Structure-based virtual ligand screening: recent success stories.* Combinatorial chemistry & high throughput screening **12**, 1000 (2009).

[59] M. Smyth and J. Martin. *X Ray Crystallography.* Journal of Clinical Pathology: Molecular Pathology **53**, 8 (2000).

[60] K. Wüthrich. *Protein structure determination in solution by NMR spectroscopy.* Journal of Biological Chemistry **265**, 22059 (1990).

[61] `http://www.wwpdb.org/download/downloads`.

[62] `http://www.rcsb.org/pdb/news.do?year=2016&article=57f3dffee6119df81075df89`.

[63] M. A. Khamis, W. Gomaa, and W. F. Ahmed. *Machine learning in computational docking*. Artificial Intelligence in Medicine **63**, 135 (2015).

[64] M. Hamzeh-Mivehroud, B. Sokouti, and S. Dastmalchi. *Molecular Docking at a Glance*. In *Methods and Algorithms for Molecular Docking-Based Drug Design and Discovery*, pp. 1–38. IGI Global (2016).

[65] G. R. Marshall. *Computer-aided drug design*. Annual review of pharmacology and toxicology **27**, 193 (1987).

[66] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press (2012).

[67] G. M. James. *Variance and bias for general loss functions*. Machine Learning **51**, 115 (2003).

[68] M. Totrov and R. Abagyan. *Flexible Ligand Docking To Multiple Receptor Conformations: a Practical Alternative*. Current Opinion in Structural Biology **18**, 178 (2008).

[69] M. a. Lill. *Efficient Incorporation of Protein Flexibility and Dynamics into Molecular Docking Simulations*. Biochemistry **50**, 6157 (2011).

[70] W. Kabsch. *A discussion of the solution for the best rotation to relate two sets of vectors*. Acta Crystallographica Section A **34**, 827 (1978).

[71] W. J. Allen and R. C. Rizzo. *Implementation of the Hungarian Algorithm to Account for Ligand Symmetry and Similarity in Structure-Based Design*. Journal of Chemical Information and Modeling **54**, 518 (2014).

[72] O. Trott and A. Olson. *AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading*. Journal of computational chemistry **31**, 455 (2010).

[73] B. Reva, A. Finkelstein, and J. Skolnick. *What is the probability of a chance prediction of a protein structure with an rmsd of 6 Å?* Folding and Design **3**, 141 (1998).

[74] A. Stark, S. Sunyaev, and R. B. Russell. *A Model for Statistical Significance of Local Similarities in Structure.* Journal of Molecular Biology **326**, 1307 (2003).

[75] M. Brylinski and J. Skolnick. *What is the relationship between the global structures of apo and holo proteins?* Proteins: Structure, Function, and Bioinformatics **70**, 363 (2007).

[76] M. Brylinski and J. Skolnick. *FINDSITELHM: A Threading-Based Approach to Ligand Homology Modeling.* PLoS Computational Biology **5**, e1000405 (2009).

[77] Y. Ding, Y. Fang, W. P. Feinstein, J. Ramanujam, D. M. Koppelman, *et al. Geaux-Dock: A novel approach for mixed-resolution ligand docking using a descriptor-based force field.* Journal of Computational Chemistry **36**, 2013 (2015).

[78] R. T. Kroemer, A. Vulpetti, J. J. McDonald, D. C. Rohrer, J.-Y. Trosset, *et al. Assessment of Docking Poses: Interactions-Based Accuracy Classification (IBAC) versus Crystal Structure Deviations.* Journal of Chemical Information and Computer Sciences **44**, 871 (2004).

[79] D. Yusuf, A. M. Davis, G. J. Kleywegt, and S. Schmitt. *An alternative method for the evaluation of docking performance: RSR vs RMSD.* Journal of Chemical Information and Modeling **48**, 1411 (2008).

[80] R. a. Abagyan and M. M. Totrov. *Contact area difference (CAD): a robust measure to evaluate accuracy of protein models.* Journal of Molecular Biology **268**, 678 (1997).

[81] C.-e. a. Chang, W. Chen, and M. K. Gilson. *Ligand configurational entropy and protein binding.* Proceedings of the National Academy of Sciences **104**, 1534 (2007).

[82] J. Meiler and D. Baker. *ROSETTALIGAND: Protein-small molecule docking with full side-chain flexibility.* Proteins: Structure, Function, and Bioinformatics **65**, 538 (2006).

[83] M. Brylinski and W. P. Feinstein. *eFindSite: Improved prediction of ligand binding sites in protein models using meta-threading, machine learning and auxiliary ligands.* Journal of Computer-Aided Molecular Design **27**, 551 (2013).

[84] G. Wang and R. L. Dunbrack. *PISCES: a protein sequence culling server.* Bioinformatics **19**, 1589 (2003).

[85] J. W. M. Nissink, C. Murray, M. Hartshorn, M. L. Verdonk, J. C. Cole, *et al. A new test set for validating predictions of protein-ligand interaction.* Proteins: Structure, Function, and Bioinformatics **49**, 457 (2002).

[86] J. Yang, A. Roy, and Y. Zhang. *BioLiP: A semi-manually curated database for biologically relevant ligand-protein interactions.* Nucleic Acids Research **41**, 1096 (2013).

[87] M. Gao and J. Skolnick. *APoc: Large-scale identification of similar protein pockets.* Bioinformatics **29**, 597 (2013).

[88] N. M. O'Boyle, R. Guha, E. L. Willighagen, S. E. Adams, J. Alvarsson, *et al. Open Data, Open Source and Open Standards in chemistry: The Blue Obelisk five years on.* J. Cheminformatics **3**, 37 (2011).

[89] W. P. Feinstein and M. Brylinski. *Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets.* Journal of Cheminformatics **7**, 18 (2015).

[90] M. Clark, R. D. Cramer, and N. Van Opdenbosch. *Validation of the general purpose tripos 5.2 force field.* Journal of Computational Chemistry **10**, 982 (1989).

[91] M. Zacharias. *Protein-protein docking with a reduced protein model accounting for side-chain flexibility.* Protein Science **12**, 1271 (2003).

[92] D. A. Beck, D. O. Alonso, and V. Daggett. *A microscopic view of peptide and protein solvation.* Biophysical Chemistry **100**, 221 (2002).

[93] B. Matthews. *Comparison of the predicted and observed secondary structure of T4 phage lysozyme.* Biochimica et Biophysica Acta (BBA) - Protein Structure **405**, 442 (1975).

[94] T. Kawabata. *Build-up algorithm for atomic correspondence between chemical structures.* Journal of chemical information and modeling **51**, 1775 (2011).

[95] E. C. Fieller, H. O. Hartley, and E. S. Pearson. *Tests for Rank Correlation Coefficients. I.* Biometrika **44**, 470 (1957).

[96] T. E. Oliphant. *Python for Scientific Computing.* Computing in Science & Engineering **9**, 10 (2007).

[97] R. Fisher and R. Fisher. *Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population.* Biometrika **10**, 507 (1915).

[98] K. Pearson. *Note on Regression and Inheritance in the Case of Two Parents.* Proceedings of the Royal Society of London **58**, 240 (1895).

[99] J. B. Kinney and G. S. Atwal. *Equitability, mutual information, and the maximal information coefficient.* Proceedings of the National Academy of Sciences **111**, 3354 (2014).

[100] D. G. Bonett. *Confidence interval for a coefficient of quartile variation.* Computational Statistics & Data Analysis **50**, 2953 (2006).

[101] M. N. G. James, A. R. Sielecki, J. Moult, V. Hruby, and D. Rich. *Crystallographic Analysis of a Pepstatin Analogue Binding to the Aspartyl Proteinase Penicillopepsin*

at 1.8 Angstroms Resolution. In *Peptides: Structure and Function, Proceedings of the of the Eighth American Peptide Symposium*, pp. 521–530 (1983).

[102] B. A. Katz, R. Mackman, C. Luong, K. Radika, A. Martelli, *et al. Structural basis for selectivity of a small molecule, S1-binding, submicromolar inhibitor of urokinase-type plasminogen activator.* Chemistry & Biology **7**, 299 (2000).

[103] J. Skolnick and M. Gao. *Interplay of physics and evolution in the likely origin of protein biochemical function.* Proceedings of the National Academy of Sciences **110**, 9344 (2013).

[104] J.-I. Ito, Y. Tabei, K. Shimizu, K. Tsuda, and K. Tomii. *PoSSuM: a database of similar protein-ligand binding and putative pockets.* Nucleic Acids Research **40**, D541 (2012).

[105] J.-I. Ito, Y. Tabei, K. Shimizu, K. Tomii, and K. Tsuda. *PDB-scale analysis of known and putative ligand-binding sites with structural sketches.* Proteins: Structure, Function, and Bioinformatics **80**, 747 (2012).

[106] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, *et al. Detecting Novel Associations in Large Data Sets.* Science **334**, 1518 (2011).

[107] D. S. Millan, M. E. Bunnage, J. L. Burrows, K. J. Butcher, P. G. Dodd, *et al. Design and Synthesis of Inhaled p38 Inhibitors for the Treatment of Chronic Obstructive Pulmonary Disease.* Journal of Medicinal Chemistry **54**, 7797 (2011).

[108] R.-g. Zhang, C. Andersson, A. Savchenko, T. Skarina, E. Evdokimova, *et al. Structure of Escherichia coli Ribose-5-Phosphate Isomerase.* Structure **11**, 31 (2003).

[109] J. R. Simard, S. Klüter, C. Grütter, M. Getlik, M. Rabiller, *et al. A new screening assay for allosteric inhibitors of cSrc.* Nature Chemical Biology **5**, 394 (2009).

[110] P. ezáčová, M. Kožíšek, S. F. Moy, I. Sieglová, A. Joachimiak, *et al. Crystal structures of the effector-binding domain of repressor Central glycolytic gene Regulator from Bacillus subtilis reveal ligand-induced structural changes upon binding of several glycolytic intermediates.* Molecular Microbiology **69**, 895 (2008).

[111] Y. Zhang and J. Skolnick. *Scoring function for automated assessment of protein structure template quality.* Proteins **57**, 702 (2004).

[112] S. Pandit and J. Skolnick. *Fr-TM-align: a new protein structural alignment method based on fragment alignments and the TM-score.* BMC Bioinformatics **9**, 531 (2008).

[113] N. J. Liverton, M. K. Holloway, J. A. McCauley, M. T. Rudd, J. W. Butcher, *et al. Molecular modeling based approach to potent P2-P4 macrocyclic inhibitors of hepatitis C NS3/4A protease.* Journal of the American Chemical Society **130**, 4607 (2008).

[114] D. J. Hazuda, N. J. Anthony, R. P. Gomez, S. M. Jolly, J. S. Wai, *et al. A naphthyridine carboxamide provides evidence for discordant resistance between mechanistically identical inhibitors of HIV-1 integrase.* Proceedings of the National Academy of Sciences of the United States of America **101**, 11233 (2004).

[115] W. L. Jorgensen. *The Many Roles of Computation in Drug Discovery.* Science **303**, 1813 (2004).

[116] G. L. Warren, C. W. Andrews, A.-M. Capelli, B. Clarke, J. LaLonde, *et al. A critical assessment of docking programs and scoring functions.* Journal of medicinal chemistry **49**, 5912 (2006).

[117] E. Kellenberger, J. Rodrigo, P. Muller, and D. Rognan. *Comparative evaluation of eight docking tools for docking and virtual screening accuracy.* Proteins: Structure, Function, and Bioinformatics **57**, 225 (2004).

[118] N. Paul and D. Rognan. *ConsDock: A new program for the consensus analysis of proteinligand interactions.* Proteins: Structure, Function, and Bioinformatics **47**, 521 (2002).

[119] J. S. Dixon. *Evaluation of the CASP2 docking section.* Proteins: Structure, Function, and Bioinformatics **29**, 198 (1997).

[120] D. Plewczynski, M. Łaźniewski, R. Augustyniak, and K. Ginalski. *Can we trust docking results? Evaluation of seven commonly used programs on PDBbind database.* Journal of Computational Chemistry **32**, 742 (2011).

[121] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, *et al. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function.* Journal of computational chemistry **19**, 1639 (1998).

[122] G. Neudert and G. Klebe. *DSX : A Knowledge-Based Scoring Function for the Assessment of ProteinLigand Complexes.* Journal of Chemical Information and Modeling **51**, 2731 (2011).

[123] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, *et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy.* Journal of medicinal chemistry **47**, 1739 (2004).

[124] I. Muegge. *PMF scoring revisited.* Journal of medicinal chemistry **49**, 5895 (2006).

[125] H. Gohlke, M. Hendlich, and G. Klebe. *Knowledge-based scoring function to predict protein-ligand interactions.* Journal of molecular biology **295**, 337 (2000).

[126] W. Mooij and M. L. Verdonk. *General and targeted statistical potentials for proteinligand interactions.* Proteins: Structure, Function, and Bioinformatics **61**, 272 (2005).

[127] D. B. Kitchen, H. Decornez, J. R. Furr, and J. Bajorath. *Docking and scoring in virtual screening for drug discovery: methods and applications.* Nature Reviews Drug Discovery **3**, 935 (2004).

[128] T. Cheng, Q. Li, Z. Zhou, Y. Wang, and S. H. Bryant. *Structure-Based Virtual Screening for Drug Discovery: a Problem-Centric Review.* The AAPS Journal **14**, 133 (2012).

[129] S.-Y. Huang, S. Z. Grinter, and X. Zou. *Scoring functions and their evaluation methods for proteinligand docking: recent advances and future directions.* Physical Chemistry Chemical Physics **12**, 12899 (2010).

[130] J. Liu and R. Wang. *Classification of Current Scoring Functions.* Journal of Chemical Information and Modeling **55**, 475 (2015).

[131] P. Ferrara, H. Gohlke, D. J. Price, G. Klebe, and C. L. Brooks. *Assessing scoring functions for protein-ligand interactions.* Journal of medicinal chemistry **47**, 3032 (2004).

[132] T. Cheng, X. Li, Y. Li, Z. Liu, and R. Wang. *Comparative Assessment of Scoring Functions on a Diverse Test Set.* Journal of Chemical Information and Modeling **49**, 1079 (2009).

[133] C. Bissantz, P. Bernard, M. Hibert, and D. Rognan. *Proteinbased virtual screening of chemical databases. II. Are homology models of gprotein coupled receptors suitable targets?* Proteins: Structure, Function, and Bioinformatics **50**, 5 (2003).

[134] S. L. McGovern and B. K. Shoichet. *Information decay in molecular docking screens against holo, apo, and modeled conformations of enzymes.* Journal of medicinal chemistry **46**, 2895 (2003).

[135] S. Karthikeyan, Q. Zhou, A. L. Osterman, and H. Zhang. *Ligand Binding-Induced Conformational Changes in Riboflavin Kinase: Structural Basis for the Ordered Mechanism , .* Biochemistry **42**, 12532 (2003).

[136] J. A. Erickson, M. Jalaie, D. H. Robertson, R. A. Lewis, and M. Vieth. *Lessons in Molecular Recognition: The Effects of Ligand and Protein Flexibility on Molecular Docking Accuracy.* Journal of Medicinal Chemistry **47**, 45 (2004).

[137] S. Renfrey and J. Featherstone. *Structural proteomics.* Nature Reviews Drug Discovery **1**, 175 (2002).

[138] J. Skolnick, H. Zhou, and M. Gao. *Are predicted protein structures of any value for binding site prediction and virtual ligand screening?* Current Opinion in Structural Biology **23**, 191 (2013).

[139] M. Brylinski and J. Skolnick. *Comprehensive structural and functional characterization of the human kinome by protein structure modeling and ligand virtual screening.* Journal of chemical information and modeling **50**, 1839 (2010).

[140] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, *et al. The protein data bank.* Nucleic acids research **28**, 235 (2000).

[141] C. Venkatachalam, X. Jiang, T. Oldfield, and M. Waldman. *LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites.* Journal of Molecular Graphics and Modelling **21**, 289 (2003).

[142] R. S. Mulliken. *Electronic population analysis on LCAOMO molecular wave functions. I.* The Journal of Chemical Physics **23**, 1833 (1955).

[143] L. Yang, C.-h. Tan, M.-J. Hsieh, J. Wang, Y. Duan, *et al. New-Generation Amber United-Atom Force Field.* The Journal of Physical Chemistry B **110**, 13166 (2006).

[144] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc. (1981).

[145] M. Levitt. *A simplified representation of protein conformations for rapid simulation of protein folding*. Journal of molecular biology **104**, 59 (1976).

[146] S. Miyazawa and R. L. Jernigan. *Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation*. Macromolecules **18**, 534 (1985).

[147] M. Brylinski and D. Lingam. *eThread: a highly optimized machine learning-based approach to meta-threading and the modeling of protein tertiary structures*. PloS one **7**, e50200 (2012).

[148] M. Brylinski. *Nonlinear Scoring Functions for Similarity-Based Ligand Docking and Binding Affinity Prediction*. Journal of Chemical Information and Modeling **53**, 3097 (2013).

[149] D. J. Rogers and T. T. Tanimoto. *A Computer Program for Classifying Plants*. Science **132**, 1115 LP (1960).

[150] E. Parzen. *On estimation of a probability density function and mode*. The annals of mathematical statistics pp. 1065–1076 (1962).

[151] M. Rosenblatt. *Remarks on some nonparametric estimates of a density function*. The Annals of Mathematical Statistics **27**, 832 (1956).

[152] I. Muegge and M. Rarey. *Small molecule docking and scoring*. Reviews in computational chemistry **17**, 1 (2001).

[153] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin. *A geometric approach to macromolecule-ligand interactions*. Journal of molecular biology **161**, 269 (1982).

[154] N. Eswar, B. Webb, M. A. Marti-Renom, M. Madhusudhan, D. Eramian, *et al. Comparative Protein Structure Modeling Using Modeller*. In *Current Protocols in Bioinformatics*, pp. 5.6.1–5.6.30. John Wiley & Sons, Inc., Hoboken, NJ, USA (2006).

[155] Z. Zhang and O. F. Lange. *Replica Exchange Improves Sampling in Low-Resolution Docking Stage of RosettaDock*. PLoS ONE **8**, e72096 (2013).

[156] M. J. Hartshorn, M. L. Verdonk, G. Chessari, S. C. Brewerton, W. T. M. Mooij, *et al. Diverse, High-Quality Test Set for the Validation of ProteinLigand Docking Performance*. Journal of Medicinal Chemistry **50**, 726 (2007).

[157] M. Steinbach, G. Karypis, and V. Kumar. *A comparison of document clustering techniques*. In *KDD workshop on text mining*, volume 400, pp. 525–526. Boston (2000).

[158] V. Sobolev, A. Sorokine, J. Prilusky, E. Abola, and M. Edelman. *Automated analysis of interatomic contacts in proteins*. Bioinformatics **15**, 327 (1999).

[159] V. Sobolev, T. M. Moallem, R. C. Wade, G. Vriend, and M. Edelman. *CASP2 molecular docking predictions with the LIGIN software*. Proteins Structure Function and Genetics **29**, 210 (1997).

[160] R. Najmanovich, J. Kuttner, V. Sobolev, and M. Edelman. *Sidechain flexibility in proteins upon ligand binding*. Proteins: Structure, Function, and Bioinformatics **39**, 261 (2000).

[161] G. R. Stockwell and J. M. Thornton. *Conformational Diversity of Ligands Bound to Proteins*. Journal of Molecular Biology **356**, 928 (2006).

[162] M. C. Nicklaus, S. Wang, J. S. Driscoll, and G. W. Milne. *Conformational changes of small molecules binding to proteins*. Bioorganic & Medicinal Chemistry **3**, 411 (1995).

[163] R. M. Knegtel, I. D. Kuntz, and C. Oshiro. *Molecular docking to ensembles of protein structures.* Journal of Molecular Biology **266**, 424 (1997).

[164] M. Brylinski, S. Y. Lee, H. Zhou, and J. Skolnick. *The utility of geometrical and chemical restraint information extracted from predicted ligand-binding sites in protein structure refinement.* Journal of structural biology **173**, 558 (2011).

[165] H. Fan, D. Schneidman-Duhovny, J. J. Irwin, G. Dong, B. K. Shoichet, *et al. Statistical potential for modeling and ranking of proteinligand interactions.* Journal of chemical information and modeling **51**, 3078 (2011).

[166] I. Muegge and Y. C. Martin. *A general and fast scoring function for protein-ligand interactions: a simplified potential approach.* Journal of medicinal chemistry **42**, 791 (1999).

[167] H. Gohlke and G. Klebe. *Statistical potentials and scoring functions applied to proteinligand binding.* Current opinion in structural biology **11**, 231 (2001).

[168] N. J. J. Salkind. *Encyclopedia of measurement and statistics.* Sage Publications (2006).

[169] R. L. DesJarlais, D. S. Yamashita, H.-j. Oh, I. N. Uzinskas, K. F. Erhard, *et al. Use of X-ray Co-crystal Structures and Molecular Modeling To Design Potent and Selective Non-peptide Inhibitors of Cathepsin K.* Journal of the American Chemical Society **120**, 9114 (1998).

[170] K. I. Varughese, Y. Su, D. Cromwell, S. Hasnain, and Nguyen Huu Xuong. *Crystal structure of an actinidin-E-64 complex.* Biochemistry **31**, 5172 (1992).

[171] S. Yin, L. Biedermannova, J. Vondrasek, and N. V. Dokholyan. *MedusaScore: An Accurate Force Field-Based Scoring Function for Virtual Drug Screening.* Journal of Chemical Information and Modeling **48**, 1656 (2008).

[172] Y. Chen and D. Zhi. *Ligand-protein inverse docking and its potential use in the computer search of protein targets of a small molecule.* Proteins: Structure, Function, and Genetics **43**, 217 (2001).

[173] G. Lauro, M. Masullo, S. Piacente, R. Riccio, and G. Bifulco. *Inverse Virtual Screening allows the discovery of the biological activity of natural compounds.* Bioorganic & Medicinal Chemistry **20**, 3596 (2012).

[174] D.-L. Ma, D. S.-H. Chan, and C.-H. Leung. *Drug repositioning by structure-based virtual screening.* Chemical Society Reviews **42**, 2130 (2013).

[175] S. L. Kinnings, N. Liu, N. Buchmeier, P. J. Tonge, L. Xie, *et al. Drug Discovery Using Chemical Systems Biology: Repositioning the Safe Medicine Comtan to Treat Multi-Drug and Extensively Drug Resistant Tuberculosis.* PLoS Computational Biology **5**, e1000423 (2009).

[176] Y. Y. L. J. An and S. J. M. Jones. *A large-scale computational approach to drug repositioning.* Genome Informatics **17**, 239 (2006).

[177] I. Guyon and A. Elisseeff. *An introduction to variable and feature selection.* The Journal of Machine Learning Research **3**, 1157 (2003).

[178] S. L. Kinnings, N. Liu, P. J. Tonge, R. M. Jackson, L. Xie, *et al. A machine learning-based method to improve docking scoring functions and its application to drug repurposing.* Journal of chemical information and modeling **51**, 408 (2011).

[179] D. M. Krüger, G. Jessen, and H. Gohlke. *How Good Are State-Of-The-Art Docking Tools in Predicting Ligand Binding Modes in Protein-Protein Interfaces?* Journal of Chemical Information and Modeling **52**, 2807 (2012).

[180] H. Merlitz and W. Wenzel. *Comparison of stochastic optimization methods for receptorligand docking.* Chemical Physics Letters **362**, 271 (2002).

[181] H. Sutter. *The free lunch is over: A fundamental turn toward concurrency in software.* Dr. Dobbs journal **30**, 202 (2005).

[182] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. *Dark silicon and the end of multicore scaling.* In *Proceeding of the 38th annual international symposium on Computer architecture - ISCA '11*, p. nil (2011).

[183] W. chun Feng, X. Feng, and R. Ge. *Green Supercomputing Comes of Age.* IT Professional **10**, 17 (2008).

[184] J. E. Stone, D. Gohara, and G. Shi. *OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems.* Computing in Science & Engineering **12**, 66 (2010).

[185] J. Jeffers and J. Reinders. *Intel Xeon Phi coprocessor high-performance programming.* Newnes (2013).

[186] M. Simonsen, M. H. Christensen, R. Thomsen, and C. N. S. Pedersen. *GPU-Accelerated High-Accuracy Molecular Docking Using Guided Differential Evolution*, pp. 349–367. Natural Computing Series. Springer Science + Business Media (2013).

[187] O. Korb, T. Stutzle, and T. E. Exner. *Accelerating Molecular Docking Calculations Using Graphics Processing Units.* Journal of Chemical Information and Modeling **51**, 865 (2011).

[188] G. D. Guerrero, H. E. Perez-Snchez, J. M. Cecilia, and J. M. Garcia. *Parallelization of Virtual Screening in Drug Discovery on Massively Parallel Architectures.* In *2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, p. nil (2012).

[189] S. McIntosh-Smith, J. Price, R. B. Sessions, and A. A. Ibarra. *High Performance in Silico Virtual Drug Screening on Many-Core Processors*. International Journal of High Performance Computing Applications **29**, 119 (2014).

[190] M. Brylinski and J. Skolnick. *Q-Dock: Low-resolution flexible ligand docking with pocket-specific threading restraints*. Journal of Computational Chemistry **29**, 1574 (2008).

[191] D. J. Earl and M. W. Deem. *Parallel Tempering: Theory, Applications, and New Perspectives*. Physical Chemistry Chemical Physics **7**, 3910 (2005).

[192] P. J. Mucci, S. Browne, C. Deane, and G. Ho. *PAPI: A portable interface to hardware performance counters*. In *Proceedings of the department of defense HPCMP users group conference*, pp. 7–10 (1999).

[193] P. S. Charifson, L. M. Shewchuk, W. Rocque, C. W. Hummel, S. R. Jordan, *et al. Peptide Ligands of Pp60 C-Src Sh2 Domains: A Thermodynamic and Structural Study*. Biochemistry **36**, 6283 (1997).

[194] A. K. Ghose, V. N. Viswanadhan, and J. J. Wendoloski. *A Knowledge-Based Approach in Designing Combinatorial Or Medicinal Chemistry Libraries for Drug Discovery. 1. A Qualitative and Quantitative Characterization of Known Drug Databases*. J. Comb. Chem. **1**, 55 (1999).

[195] W. P. Feinstein and M. Brylinski. *Efindsite: Enhanced Fingerprint-Based Virtual Screening Against Predicted Ligand Binding Sites in Protein Models*. Molecular Informatics **33**, 135 (2014).

[196] K. Beyls and E. DHollander. *Reuse distance as a metric for cache behavior*. In *Proceedings of the IASTED Conference on Parallel and Distributed Computing and systems*, volume 14, pp. 350–360 (2001).

[197] E. Blem, J. Menon, and K. Sankaralingam. *A detailed analysis of contemporary arm and x86 architectures*. UW-Madison Technical Report (2013).

[198] R. Raag and T. L. Poulos. *The Structural Basis for Substrate-Induced Changes in Redox Potential and Spin Equilibrium in Cytochrome P-450cam*. Biochemistry **28**, 917 (1989).

[199] L. Prade, R. Huber, T. H. Manoharan, W. E. Fahl, and W. Reuter. *Structures of Class Pi Glutathione S-Transferase From Human Placenta in Complex With Substrate, Transition-State Analogue and Inhibitor*. Structure **5**, 1287 (1997).

[200] N. Y. Chirgadze, D. J. Sall, V. J. Klimkowski, D. K. Clawson, S. L. Briggs, *et al. The Crystal Structure of Human α-thrombin Complexed With Ly178550, a Nonpeptidyl, Active Site-Directed Inhibitor*. Protein Science **6**, 1412 (1997).

[201] J. Gabel, J. Desaphy, and D. Rognan. *Beware of Machine Learning-Based Scoring Functions-On the Danger of Developing Black Boxes*. Journal of Chemical Information and Modeling **54**, 2807 (2014).

[202] Y. Li, Z. Liu, J. Li, L. Han, J. Liu, *et al. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set*. Journal of Chemical Information and Modeling **54**, 1700 (2014).

[203] M. Remmert, A. Biegert, A. Hauser, and J. Söding. *Hhblits: Lightning-Fast Iterative Protein Sequence Searching By Hmm-Hmm Alignment*. Nature Methods **9**, 173 (2011).

Appendix A
LETTER OF PERMISSION A

The next few pages is a copy of the letter of permission to use the published article titled "Assessing the similarity of ligand binding conformations with the Contact Mode Score" in Chapter 2.

ELSEVIER LICENSE
TERMS AND CONDITIONS

Oct 07, 2016

This Agreement between Yun Ding ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	3963910342532
License date	Oct 07, 2016
Licensed Content Publisher	Elsevier
Licensed Content Publication	Computational Biology and Chemistry
Licensed Content Title	Assessing the similarity of ligand binding conformations with the Contact Mode Score
Licensed Content Author	Yun Ding,Ye Fang,Juana Moreno,J. Ramanujam,Mark Jarrell,Michal Brylinski
Licensed Content Date	October 2016
Licensed Content Volume Number	64
Licensed Content Issue Number	n/a
Licensed Content Pages	11
Start Page	403
End Page	413
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	Data-driven rational drug discovery
Expected completion date	Nov 2016
Estimated size (number of pages)	100
Elsevier VAT number	GB 494 6272 12
Requestor Location	Yun Ding 4539 Alvin Dark Ave, Apt 4 BATON ROUGE, LA 70820 United States Attn: Yun Ding
Total	0.00 USD
Terms and Conditions	

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

GENERAL TERMS

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier at permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance, please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

9. Warranties: Publisher makes no representations or warranties with respect to the licensed material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing: This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).
13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.
14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:
15. **Translation**: This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article.
16. **Posting licensed content on any Website**: The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at http://www.sciencedirect.com/science/journal/xxxxx or the Elsevier homepage for books at http://www.elsevier.com; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.
Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at http://www.elsevier.com . All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.
17. **For journal authors:** the following clauses are applicable in addition to the above:
Preprints:
A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).
Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).
If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available

version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.
Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications.
Authors can share their accepted author manuscript:

- immediately
  - via their non-commercial person homepage or blog
  - by updating a preprint in arXiv or RePEc with the accepted manuscript
  - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group
  - directly by providing copies to their students or to research collaborators for their personal use
  - for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- after the embargo period
  - via non-commercial hosting platforms such as their institutional repository
  - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.
Policies for sharing publishing journal articles differ for subscription and gold open access articles:
Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.
Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.
If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.
Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a CrossMark logo, the end user license, and a DOI link to the formal publication on ScienceDirect.
Please refer to Elsevier's posting policy for further information.
18. **For book authors** the following clauses are applicable in addition to the above:
Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.

19. **Thesis/Dissertation**: If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author's choice of Creative Commons user license. See our open access license policy for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated.

The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect.

If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

CC BY: The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at http://creativecommons.org/licenses/by/4.0.

CC BY NC SA: The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. Further, any new works must be made available on the same conditions. The full details of the license are available at http://creativecommons.org/licenses/by-nc-sa/4.0.

CC BY NC ND: The CC BY-NC-ND license allows users to copy and distribute the Article, provided this is not done for commercial purposes and further does not permit distribution of the Article if it is changed or edited in any way, and provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, and that the licensor is not represented as endorsing the use made of the work. The full details of the license are available at http://creativecommons.org/licenses/by-nc-nd/4.0.

Any commercial reuse of Open Access articles published with a CC BY NC SA or CC BY NC ND license requires permission from Elsevier and will be subject to a fee.

Commercial reuse includes:

- Associating advertising with the full text of the Article
- Charging fees for document delivery or access
- Article aggregation
- Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.

20. **Other Conditions**:

v1.8

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix B
LETTER OF PERMISSION B

The next few pages is a copy of the letter of permission to use the published article titled "GeauxDock: A novel approach for mixedresolution ligand docking using a descriptorbased force field" in this Chapter 3.

JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS

Sep 02, 2016

This Agreement between Yun Ding ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	3940850956961
License date	Sep 02, 2016
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Journal of Computational Chemistry
Licensed Content Title	GeauxDock: A novel approach for mixed-resolution ligand docking using a descriptor-based force field
Licensed Content Author	Yun Ding,Ye Fang,Wei P. Feinstein,Jagannathan Ramanujam,David M. Koppelman,Juana Moreno,Michal Brylinski,Mark Jarrell
Licensed Content Date	Aug 6, 2015
Licensed Content Pages	14
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Electronic
Portion	Full article
Will you be translating?	No
Title of your thesis / dissertation	Data Driven Rational Drug Discovery
Expected completion date	Dec 2016
Expected size (number of pages)	100
Requestor Location	Yun Ding Department of Physics and Astronomy, LSU 202 Nicholson Hall, Tower Dr Baton Rouge LA 70803 BATON ROUGE, LA 70820 United States Attn: Yun Ding
Publisher Tax ID	EU826007151
Billing Type	Invoice
Billing Address	Yun Ding Department of Physics and Astronomy, LSU 202 Nicholson Hall, Tower Dr Baton Rouge LA 70803 BATON ROUGE, LA 70820 United States Attn: Yun Ding
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at http://myaccount.copyright.com).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.

- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order,** is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner.**For STM Signatory Publishers clearing permission under the terms of the STM Permissions Guidelines only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts,** You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding

("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.

- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.

- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.

- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.

- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.

- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.

- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.

- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC)License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.
Further details can be found on Wiley Online Library
http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

v1.10 Last updated September 2015

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

VITA

Yun Ding was born in Huangshi city of Hubei province, People's Republic of China. He attended Wuhan University at Wuhan and received a Bachelor of Science degree in Physics in July 2011. Afterwards, he began his graduate study at Louisiana State University in 2011 under supervision of Mark Jarrell and Michal Brylinski. He is a candidate to receive the degree of Doctor of Philosophy at December 2016 commencement.