Methodological Advances for Drug Discovery and Protein Engineering

by

Xinqiang Ding

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Bioinformatics)
in the University of Michigan
2018

Doctoral Committee:

Professor Charles L. Brooks III, Chair
Professor Heather A. Carlson
Assistant Professor Aaron T. Frank
Assistant Professor Peter L. Freddolino
Associate Professor Nina Lin

Xinqiang Ding

xqding@umich.edu

ORCID ID: 0000-0002-4598-8732

ACKNOWLEDGMENTS

First and foremost I would like to thank my thesis advisor Prof. Charles Brooks III for the guidance and help in all of my research projects. I have greatly benefited from his enthusiasm and expertise that consistently provide for me great advice. Much of the work presented in the dissertation would have never happened without his guidance. I also want to express my gratitude to my thesis committee members — Barry Grant, Georgios Skiniotis, Aaron Frank, Heather Carlson, Peter Freddolino, and Nina Lin — for their suggestions on the research projects. I would also like to thank Daniel Burns and Margit Burmeister. As co-directors of the bioinformatics program, they not only gave me the opportunity to have training in the program but also provided guidance on my coursework and research rotations. I have had a great time doing research in the Brooks lab and received enormous help from many Brooks lab members including Shanshan Cheng, Jessica Gagnon, Kira Armacost, Ryan Hayes, Jonah Vilseck, Kathleen Dyki, and David Braun. I also want to thank Julia Essen, the administrator for the bioinformatics program, for her help throughout my training. At the time of writing the dissertation, I am grateful to be financially supported by the Rackham Predoctoral Fellowship program. Finally, I want to thank my parents and my wife for their patience and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Designing and engineering molecules not only tests our understanding of nature but also plays an important role in improving both human health and industrial productivity. Two such examples are drug discovery, which aims to design new molecules to treat diseases, and protein engineering, which develops useful proteins for medical purposes or catalyzing industrial chemical reactions. Drug discovery and protein engineering are both time-consuming and financially expensive processes because they require multiple rounds of trial-and-error. One effective path to reducing these costs and accelerating these processes is through the development of computational methods that rationalize the course of design and engineering. Facilitated by methodological developments and the increasing availability of computational resources, computational strategies are becoming effective approaches to assist drug discovery and protein engineering tasks. In this dissertation, I describe the development of novel computational methodologies for drug discovery and protein engineering that exploit evolving accelerated computing architectures and the intersection between statistical approaches and statistical mechanics.

Protein-ligand docking and free energy calculations are widely employed computational methods in drug discovery. In the dissertation, I first describe the development of an accelerated version of the protein-ligand docking method, CDOCKER, by introducing two new features — fast Fourier transform based docking and parallel simulated annealing, both of which utilize the parallel computing power of graphical processing units (GPUs). These advances not only accelerate CDOCKER by more than an order of magnitude but also provide an approach to calculate an upper bound on the docking accuracy of current scoring

functions. In the second project that is directed toward a more rigorous assessment of a ligand's binding affinity for a receptor, I introduced two new methods for protein-ligand binding free energy calculations: the Gibbs sampler λ-dynamics (GSLD) methodology and Rao-Blackwell estimators (RBE) for improved analysis of the simulation results from GSLD. Compared with the original λ-dynamics approach, GSLD is more flexible, easier to implement, and retains the capacity to calculate free energies for multiple ligands in a single simulation. Compared with the empirical estimator used in λ-dynamics, RBE has the advantages of being an unbiased estimator that does not depend on ad hoc cutoff values as previously used in the empirical estimators associated with λ-dynamics. Additionally, RBE has smaller variance than the empirical estimators.

In the realm of protein engineering, I investigated the development and application of variational auto-encoder (VAE) models to infer protein stability, evolution, and fitness landscapes based on alignments of protein sequences. VAE models are probabilistic generative models that embed discrete sequences in a lower dimensional continuous latent space. Utilizing the multiple sequence alignment from a protein family as training data, VAE models learn a probability distribution of sequences for the protein family. The probability distribution may then by employed to predict protein stability changes upon mutation. The embedding of sequences in a low dimensional latent space not only provides an approach to visualize a protein family's sequence space, but also captures evolutionary relationships between sequences. Together with experimental fitness data, the embedding enables the visualization and expression of the fitness landscape in a low dimensional continuous space. Exploiting the rapidly increasing amount of protein sequence data resulting from advances in sequencing technology, we demonstrate that these features of the VAE models are of significance for studying protein properties and evolution as well as guiding protein engineering efforts.

CHAPTER 1

Introduction

Designing and engineering molecules that have specified properties for good use is one of the goals of natural sciences. It not only tests our understanding of nature but also plays an important role in improving both human health and industrial productivity[1]. Two such examples are drug discovery [2] and protein engineering [3], which will be the focus of this dissertation. Drug discovery aims to design molecules to treat or even cure diseases, which is essential to continuously improve human health. Protein engineering designs new proteins or modifies existing proteins to make useful proteins such as antibodies and protein drugs for medical purposes, and enzymes for catalyzing industrial chemical reactions.

Drug discovery and protein engineering are both time-consuming and financially expensive processes. For instance, developing a new drug requires on average one billion dollars and ten years of effort [2]. One of the reasons for the high cost is that these processes require multiple rounds of trial-and-error[2]. Therefore, one path to reducing the cost is to develop methods that can rationalize the course of designing and engineering processes. A particularly effective approach is developing computational methods that can make predictions and help guide the design and engineering processes [4, 5]. As an example, the computational methods — protein-ligand docking and free energy calculations — have been widely employed in assisting drug discovery processes [4, 6]. Specifically, the protein-ligand docking method is used to search a large library of small molecules to identify molecules that can potentially bind with a target protein [4]. The free energy calculation approach is for more rigorous evaluation of a ligand's binding affinity with target

proteins [6]. In the realm of protein engineering, computational methods for designing proteins and predicting protein property change upon mutation are also increasingly used for guiding protein engineering efforts [5]. With continuous computational methodological developments and the increasing availability of computing resources, computational approaches are becoming more and more effective in assisting both drug discovery and protein engineering [4, 5].

In this dissertation, I describe the development and implementation of novel computational methodologies for drug discovery and protein engineering that exploit both evolving accelerated computing architectures and the intersection between statistics and statistical mechanics. In chapter 2, I describe the development and implementation of two new features — fast Fourier (FFT) transform docking and parallel simulated annealing — added to the protein-ligand docking method, CHARMM DOCKER(CDOCKER) [7]. These advances not only accelerate CDOCKER by more than an order of magnitude but also provide an approach to calculate an upper bound on the docking accuracy that can be achieved with current functions used in scoring docked poses. In chapter 3, two new methods for calculating protein-ligand binding free energies — Gibbs sampler λ-dynamics (GSLD) and Rao-Blackwell estimators (RBE) — are described [8]. GSLD is a new sampling method that combines the Gibbs sampler in statistics and the λ-dynamics approach in computational chemistry. RBE is introduced to replace the empirical estimator used in the original λ-dynamics to better analyze the simulation results from GSLD. In chapter 4, I describe the development and application of variational auto-encoders (VAE) models to infer information regarding protein stability, evolution, and fitness landscapes using alignments of multiple protein sequences. These features of the VAE models are of significance for both studying protein properties and evolution and guiding protein engineering efforts.

The organization within chapter 2, 3, and 4 follows the same structure. At the beginning of each chapter, an introduction is given to provide an overview of the specific field, followed by a review of existing corresponding computational methods. After the review,

novel computational methods developed in the dissertation are described in detail. These novel methods are applied and compared with existing methods on different systems. Then each chapter is ended by conclusions or discussions on the novel computational methods.

CHAPTER 2

Protein-Ligand Docking

Ding, Xinqiang, Ryan L. Hayes, Jonah Z. Vilseck, Murchtricia K. Charles, and Charles L. Brooks III. "CDOCKER and λ-dynamics for prospective prediction in D3R Grand Challenge 2." *Journal of computer-aided molecular design* 32, no. 1 (2018): 89-102.

Ding, Xinqiang, Yanming Wang, Charles L. Brooks III "Accelerated CDOCKER with fast Fourier transform docking and parallel simulated annealing on graphical processing units." *in preparation*.

2.1 Introduction

Protein-ligand docking methods aim to predict how ligands bind with a target protein, i.e., binding poses of ligands and their binding affinities [9]. They are widely employed in drug discovery processes to virtually screen libraries of a large number of small molecules to search for hit compounds that might be able to strongly bind with target proteins [4]. Today multiple off-the-shelf protein-ligand docking programs, either commercial or free, are available for use [10], such as CDOCKER[7], Autodock[11], Autodock Vina[12], DOCK[13], and Glide[14, 15]. Most of protein-ligand docking programs consist of two essential components — a scoring function and a search algorithm [7]. The scoring function quantifies the fit between a ligand's binding pose and the target protein and is expected to be able to differentiate the correct binding pose from incorrect ones by the assumption that the correct binding pose has the best score. When used to predict binding affinities, the

scoring function is also expected to approximate the binding free energy between ligands and target proteins. The search algorithm is utilized to sample potential ligand binding poses and identify the binding pose with the best score. Because scoring functions used in protein-ligand docking programs are not convex functions and might have multiple local minimums, heuristic search algorithms such as genetic algorithms and simulated annealing are often utilized in protein-ligand docking programs[7, 12].

CDOCKER[7], a CHARMM[16] module for protein-ligand docking, is one of the protein-ligand docking programs that are widely used in both academia and industry for drug discovery. It uses the interaction energies between proteins and ligands calculated with the CHARMM force field for proteins and the CGenFF force field [17] for ligands as its scoring function. To search for the lowest energy poses of ligands, CDOCKER utilizes molecular dynamics (MD) based simulated annealing followed by energy minimization. In the MD based simulated annealing, MD is used to simulate the dynamics of protein-ligand interactions and the temperature of MD first increases to a high value and then slowly decreases. As the temperature of MD decreases, ligands are expected to adopt to low energy poses. Resulting ligand poses from simulated annealing are further optimized by energy minimization. As the MD-based simulated annealing is a heuristic search approach, it is not guaranteed that the ligand will converge to the lowest energy pose in each trial of MD-based simulated annealing. To increase the chance that the lowest energy pose of the ligand is identified, multiple trials of simulated annealing are needed. In each trial, the ligand is first initialized with a random conformation, a random orientation, and a random position within the binding pocket before going through the MD-based simulated annealing and energy minimization. After the energy minimization, the resulting poses, one from each trial, are ranked by their interaction energies with the protein and the pose with the lowest interaction energy is predicted to be the binding pose. In a typical application of CDOCKER, a large number of ligands need to be docked with a protein. Therefore, the docking procedure has to run fast enough to make the method practical. To accelerate the docking

procedure and help search for the lowest energy poses of ligands, CDOCKER utilizes a grid representation of the binding pocket and soft-core potentials[7, 9], respectively, which will be described in detail in the **Methods** section. In this chapter, two new features — fast Fourier transform (FFT) [18] docking and parallel MD simulated annealing — are added to CDOCKER to help quantify the accuracy of CDOCKER scoring function and to further accelerate the search algorithm in CDOCKER.

2.1.1 Fast Fourier transform docking.

The FFT approach for docking was first used in rigid protein-protein docking [19]. In this approach, proteins are represented as 3 dimensional grids such that the surface complementarity of two proteins can be formulated as the correlation function between two grids [19]. Calculating the correlation function between two grids can be greatly accelerated using the FFT algorithm[20]. Since its first use in protein-protein docking [19], the FFT approach has been extended and improved in several aspects. In addition to the original potential term representing protein shape complementarity [19], potential terms representing desolvation and electrostatic interactions were added into the scoring function [21, 22, 23] to more accurately model the physical interactions between proteins. Moreover, the FFT approach was further accelerated by using spherical polar Fourier correlations to speedup the rotational space search [24, 25, 26] and by utilizing the parallel computing power of graphics processor units (GPUs) [27, 28]. With these extensions and improvements, the FFT approach has been widely adopted in multiple protein-protein docking programs [25, 23, 29].

In contrast to FFT's wide application in protein-protein docking, its application in protein-ligand docking is largely unexplored [30]. One difficulty in adopting the FFT approach for protein-ligand docking is to represent the scoring function as a correlation function between grids, as the scoring function used in protein-ligand docking is often more complicated than that in protein-protein docking. In addition, the FFT approach assumes both protein and ligand are rigid bodies, whereas, in protein-ligand docking, at least the

ligand needs to be flexible. Therefore multiple FFTs are required to search the ligand's conformation space. This in turn requires a fast implementation of FFT. Otherwise running multiple FFTs will take too much time to be practical.

In this chapter, we investigated the use of the FFT approach for protein-ligand docking in the context of CDOCKER where the CHARMM force field [16, 17] was used as the scoring function. The interaction energy, including electrostatic and van der Waals energy, between proteins and ligands are represented as the sum of multiple correlation functions between multiple pairs of grids and the calculation of correlation functions is accelerated using FFTs. Moreover, calculating multiple FFTs is further accelerated using GPUs.

2.1.2 Parallel MD-based simulated annealing with GPUs.

One of the advances in using MD simulations to study both chemical and biological systems has been the utilization of GPUs in running MD [31, 32, 33, 34]. Compared with the traditional central processing units (CPUs), the parallel computing power of GPUs enables us to run MD simulations orders of magnitude faster and simulate longer timescale dynamics of chemical and biological systems, which makes MD suitable to study processes that are not accessible before [31, 32, 33, 34]. Although GPUs have been widely employed in running MD simulations of large chemical and biological systems, they are rarely used to accelerate protein-ligand docking methods. In this chapter, we investigated the utilization of GPU computing to accelerate CDOCKER for protein-ligand docking by running MD-based simulated annealing of multiple copies of ligands in parallel on GPUs.

2.2 Methodology

2.2.1 Grids and soft-core potentials in CDOCKER

In CDOCKER's docking protocol, most of the computational time is spent on calculating forces on ligand atoms and the ligand's interaction energy with the protein for a large number of ligand poses. To accelerate the force and energy calculation a grid representation of the binding pocket is used. Specifically, the binding pocket inside a protein is discretized into a 3 dimensional grid. Probe atoms are placed on each of the grid points and their interaction energies with the protein are saved in a lookup table. Then the force and the interaction energy of a ligand atom with the protein can be rapidly calculated by looking up values in the tables, instead of explicitly calculating its interaction with all of the protein atoms.

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

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

where E_{ij}^{*} is regular interaction energy; E_{max} is a parameter controlling the "softness" of the potential; a and b are determined using the condition that the energy and the force calculated using the new formula 2.1 have to be equal to that with the regular formula at the switch distance at which $|E_{ij}^{*}| = |E_{\text{max}}|/2$. Three sets of values for parameter E_{max} are used in this study and they are summarized in Table 2.1

Table 2.1: Soft-core potentials with different "softness"

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

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

2.2.2 Fast Fourier transform (FFT) docking

2.2.2.1 Representing non-bonded interaction energy between proteins and ligands as correlation functions between grids.

In order to use the FFT approach for protein-ligand docking, the interaction energy between proteins and ligands needs to be expressed as correlation functions between grids. Because CDOCKER uses the CHARMM force field [16, 17] as its scoring function, the interaction between proteins and ligands includes electrostatic and van der Waals interactions [7].

The electrostatic interaction energy between proteins and ligands is calculated as

$$U_{\text{elec}} = \sum_{i\in L}\sum_{j\in P}\frac{1}{4\pi\epsilon}\frac{q_i q_j}{|r_i - r_j|} = \sum_{i\in L} q_i \cdot \sum_{j\in P}\frac{1}{4\pi\epsilon}\frac{q_j}{|r_i - r_j|} = \sum_{i\in L} q_i \cdot V_{\text{elec}}(r_i), \qquad (2.2)$$

where L and P are collections of ligand atoms and protein atoms, respectively; q_i and q_j are atom partial charges; r_i and r_j are atom coordinates. $V_{\text{elec}}(r_i) = \sum_{j\in P}\frac{1}{4\pi\epsilon}\frac{q_j}{|r_i - r_j|}$ is the protein electrostatic potential at position r_i. As equation (2.2) shows, the electrostatic interaction energy between protein and ligand atoms can be calculated as inner-product between the ligand atoms' charge vector $q_L = (q_i)_{i\in L}$ and the protein electrostatic potential vector $V_{\text{elec}} = (V_{\text{elec}}(r_i))_{i\in L}$. However, the protein electrostatic potential vector V_{elec} still depends on positions of ligand atoms that are not known in advance. To get rid of this dependency, grid representations are used for both the protein electrostatic potential and the ligand atoms' charges (Fig. 2.1). Specifically, the binding pocket of a protein is discretized

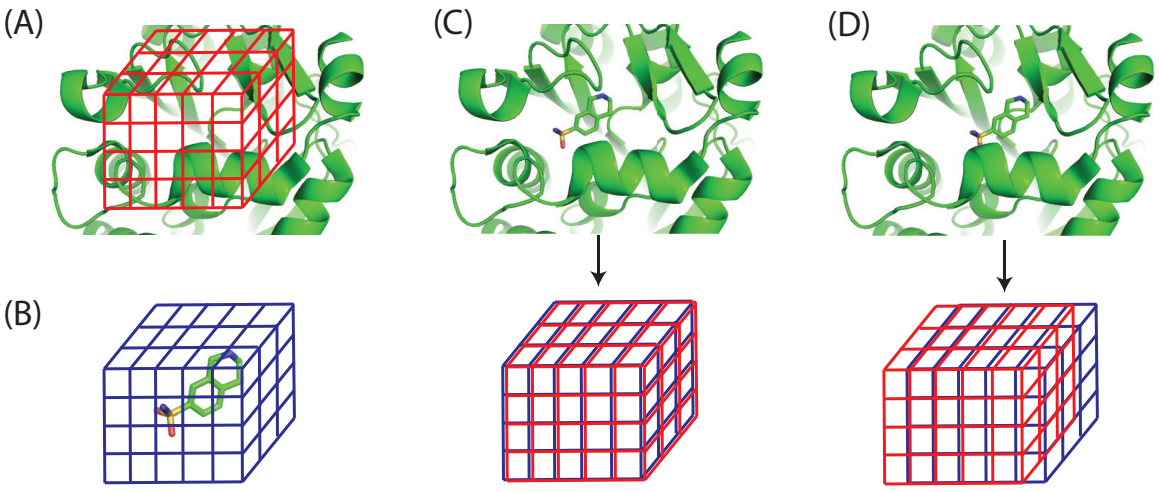

Figure 2.1: The electrostatic interaction energy between proteins and ligands can be calculated as a cross correlation function between the protein electrostatic potential grid and the ligand charge grid. **(A)** The bind pocket in the protein is discretized into a 3 dimensional grid with equal spacing distance. **(B)** Charges of ligand atoms are distributed onto a 3 dimensional grid which has the same spacing distance and the same number of grid points as the potential grid in (A). **(C,D)** As the ligand translates within the binding pocket by multiple units of the spacing distance, the electrostatic interaction energy can be approximated using a cross correlation between the protein potential grid and the ligand charge grid. (C) and (D) corresponds to the cases $U_{\text{elec}}(0,0,0)$ and $U_{\text{elec}}(0,1,0)$, respectively.

using a 3 dimensional grid and protein electrostatic potentials at all the grid points are calculated and saved in a lookup table (Fig. 2.1A). The protein electrostatic potential at the grid point (l,m,n) is represented as $V_{\text{elec}}^{\text{grid}}(l,m,n)$. Because the protein electrostatic potential is calculated only at the grid points, in order to calculate the electrostatic interaction energy between proteins and ligands, the partial charges of ligand atoms are distributed onto a 3 dimensional grid (Fig. 2.1B) in a trilinear manner (Fig. S1). The aggregated charge at the grid point (l,m,n) is represented as $Q^{\text{grid}}(l,m,n)$. Then the electrostatic interaction energy between protein atoms and ligand atoms can be approximated using the inner-product of protein electrostatic potential grid and ligand charge grid (Fig. 2.1C):

$$U_{\text{elec}} \approx \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \sum_{n=0}^{N_z-1} Q^{\text{grid}}(l,m,n) \cdot V_{\text{elec}}^{\text{grid}}(l,m,n), \tag{2.3}$$

where N_x, N_y, and N_z are numbers of grid points along X,Y,and Z direction, respectively.

Moreover, when the ligand is translated with the binding pocket by i, j, and k grid spacing units in the X, Y, and Z direction, respectively, the electrostatic potential energy between the protein and ligand can be similarly approximated using (Fig. 2.1D):

$$U_{\text{elec}}(i,j,k) \approx \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \sum_{n=0}^{N_z-1} Q^{\text{grid}}(l,m,n) \cdot V_{\text{elec}}^{\text{grid}}(l+i, m+j, n+k), \tag{2.4}$$

where $V_{\text{elec}}^{\text{grid}}$ is extended into a periodic grid, i.e., $V_{\text{elec}}^{\text{grid}}(l,m,n) = V_{\text{elec}}^{\text{grid}}(l \pmod{N_x}, m \pmod{N_y}, n \pmod{N_z})$. As shown in Eq. 2.4, as the ligand moves within the binding pocket by distances of multiple units of grid spacing in each direction, the electrostatic interaction energy between the protein and ligand can be approximated as a cross correlation function between the protein electrostatic potential grid $V_{\text{elec}}^{\text{grid}}$ and the ligand charge grid Q^{grid}. An advantage of using the grid representation, as in Eq. 2.4, over that in Eq. 2.2 is $V_{\text{elec}}^{\text{grid}}$ is independent of the ligand and can be calculated with only the protein. Similarly, grid Q^{grid} is independent of the protein and can be calculated with only the ligand.

The van der Waals interaction energy between proteins and ligands is calculated using the Lennard-Jones potential:

$$\begin{aligned} U_{\text{vdw}} &= \sum_{i \in L} \sum_{j \in P} \epsilon_{ij} \left[\left(\frac{r_{ij}^{\text{min}}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{ij}^{\text{min}}}{r_{ij}} \right)^{6} \right] \\ &= \sum_{i \in L} \sum_{j \in P} \sqrt{\epsilon_i \epsilon_j} \left[\left(\frac{(r_i^{\text{min}} + r_j^{\text{min}})/2}{|r_i - r_j|} \right)^{12} - 2 \left(\frac{(r_i^{\text{min}} + r_j^{\text{min}})/2}{|r_i - r_j|} \right)^{6} \right] \\ &= \sum_{i \in L} \sqrt{\epsilon_i} \cdot V_{\text{vdw}}(r_i, r_i^{\text{min}}), \end{aligned} \tag{2.5}$$

where r_i^{min}, r_j^{min}, ϵ_i, ϵ_j are parameters of the Lennard-Jones potential and are parts of the CHARMM force field;

$$V_{\text{vdw}}(r_i, r_i^{\text{min}}) = \sum_{j \in P} \sqrt{\epsilon_j} \left[\left(\frac{(r_i^{\text{min}} + r_j^{\text{min}})/2}{|r_i - r_j|} \right)^{12} - 2 \left(\frac{(r_i^{\text{min}} + r_j^{\text{min}})/2}{|r_i - r_j|} \right)^{6} \right]. \tag{2.6}$$

The Eq. 2.5 for van der Waals energy is similar to that in Eq. 2.2, except that $V_{\text{vdw}}(r_i, r_i^{\text{min}})$ depends on not only ligand coordinates r_i but also parameters r_i^{min}, whereas $V_{\text{elec}}(r_i)$ only depends on ligand coordinates r_i. Because of this difference, the approach used to represent the electrostatic energy between proteins and ligands as a cross correlation function between a pair of grids can not be applied to van der Waals interaction directly. In the CHARMM force field, parameters r^{min} of ligand atoms depend on their atom types and the total number of atom types is finite. Therefore, there are only a finite number of possible values for r^{min}. Taking advantage of this fact, we can group the terms in Eq. 2.5 based on the value of r^{min}:

$$U_{\text{vdw}} = \sum_{i \in L} \sqrt{\epsilon_i} \cdot V_{\text{vdw}}(r_i, r_i^{\text{min}}) = \sum_{r^{\text{min}} \in R^{\text{min}}} \sum_{i \in L_{r^{\text{min}}}} \sqrt{\epsilon_i} \cdot V_{\text{vdw}}^{r^{\text{min}}}(r_i) = \sum_{r^{\text{min}} \in R^{\text{min}}} U_{\text{vdw}}^{r^{\text{min}}}, \quad (2.7)$$

where R^{min} is the set of possible values of r^{min} for ligand atoms and $L_{r^{\text{min}}}$ is the set of ligand atoms that have the parameter of r^{min}. The individual van der Waals energy corresponding r^{min} is $U_{\text{vdw}}^{r^{\text{min}}} = \sum_{i \in L_{r^{\text{min}}}} \sqrt{\epsilon_i} \cdot V_{\text{vdw}}^{r^{\text{min}}}(r_i)$, which is similar to the Eq. 2.2 and can be calculated as a cross correlation function between grids using the same approach used for calculating the electrostatic energy. Therefore, the total van der Waals interaction energy between proteins and ligands can be approximated as the sum of multiple correlation functions between multiple pairs of grids.

2.2.2.2 Calculating cross correlation functions between grids using FFTs in parallel on GPUs.

Based on the convolution theorem [20], the cross correlation function for electrostatic energy in Eq. 2.4 can be calculated by applying a Fourier transform and an inverse Fourier transform successively on both sides of the equation:

$$U_{\text{elec}} = \mathcal{F}^{-1}\{\mathcal{F}\{Q^{\text{grid}}\}^* \cdot \mathcal{F}\{V_{\text{elec}}^{\text{grid}}\}\}. \quad (2.8)$$

The FFT algorithm is utilized to efficiently calculate both the Fourier transform and the inverse Fourier transform operations. In contrast to the naive algorithm which requires $\mathcal{O}((N_x N_y N_z)^2)$ number of operations to calculate the cross correlation function, the FFT algorithm only needs $\mathcal{O}((N_x N_y N_z)\log(N_x N_y N_z))$ number of operations. Similarly, the FFT algorithm can also be used to calculate the van der Waals interaction energy in Eq. 2.7 Although the FFT algorithm can significantly accelerate the calculation of cross correlation functions, one cross correlation function can only provide interaction energies between proteins and ligands as the ligand translates within the binding pocket with a fixed conformation and a fixed orientation. In other words, FFTs only accelerate the search of the ligand translational space. However, in protein-ligand docking where at least the ligand is flexible, the interaction energies need to be calculated for the ligand's different conformations and orientations, in addition to different positions. Therefore, multiple FFTs, each for one particular conformation and orientation of the ligand, are needed in protein-ligand docking. To accelerate this calculation, multiple FFTs are run on GPUs in batch mode to take advantage of the parallel computing power of GPUs[35].

2.2.3 Parallel MD-based simulated annealing with GPUs

As the protein-ligand interaction energy landscapes have local minimums and the MD-based simulated annealing is a heuristic search method, multiple trials of MD-based simulated annealing have to be employed to help search for the lowest energy pose. As the number of trials increases, the docking accuracy improves. In addition, in a typical application, CDOCKER needs to dock a large number of ligands with a protein. Therefore, accelerating multiple trials of MD-based simulated annealing can help CDOCKER to dock a large number of ligands in a limited time while maintaining docking accuracy. Because trials of MD-based simulated annealing are independent with each other, one way to accelerate the calculation is to run them in parallel with multiple processors. With previous implementation of CDOCKER, multiple trials of MD-based simulated annealing can already be run

in parallel with multiple CPUs. Here we introduce a new feature of CDOCKER to enable it to run multiple trials of MD-based simulated annealing simultaneously on GPUs which have been widely used to accelerate other MD simulations.

As there are already implementations of MD engines running on GPUs, instead of writing a new MD engine specifically for running multiple trials of MD-based simulation annealing on GPUs, we adopt the existing GPU-enabled MD engine in OpenMM[36]. To utilize the MD engine from OpenMM for our purpose, we make a customized system consisting of multiple copies of a ligand and one copy of the potential grids of the protein. Atoms in each copy of the ligand interacts with ligand atoms in the same copy and the potential grids, but do not interact with atoms in all other copies of ligands. Therefore, although the system includes multiple copies of the ligand, these copies of ligands are independent with each other and the dynamics of each copy of ligand is the same as if there is just one copy of ligands. Running one trial of MD-based simulated annealing with this customized system is equivalent to running multiple trails of simulated annealing for the ligand.

This approach of running multiple trials of MD-based simulated annealing on GPUs is also applicable to flexible CDOCKER [9], in which both ligand atoms and protein side chain atoms of the amino acids near the binding pocket are flexible. In this case, the customized OpenMM[36] system includes not only multiple copies of ligand atoms but also multiple copies of protein side chain atoms that are flexible. Similarly, each copy of flexible protein side chain atoms only interact with itself and the corresponding copy of ligand atoms and do not interact with other copies of either ligand atoms or flexible protein atoms.

2.2.4 Benchmark dataset

Two sets of protein-ligand complexes, the Astex diverse set[37] and the SB2012 set[38], are used as benchmark datasets to test protein-ligand docking methods in this study. The

Astex diverse set contains 85 diverse high-resolution protein-ligand complexes and has been widely used for benchmarking different protein-ligand docking methods[37]. In this study, 70 of the 85 protein-ligand complexes that do not include cofactors are used. Compared to the Astex diverse set, the SB2012 set[38] is a much larger set of protein-ligand complexes. It contains 1043 protein-ligand complexes, out of which the 1003 complexes that do not have cofactors and can be typed using CGenFF [39] are used in this study. The 1003 protein-ligand complexes from the SB2012 set include 69 out of 70 complexes from the Astex diverse set.

2.3 Results

2.3.1 Fast Fourier transform docking

2.3.1.1 Energy calculation acceleration with FFTs and GPUs

When a ligand has a fixed conformation and a fixed orientation, its interaction energy with a protein as the ligand translates on grid points can be represented as cross correlation functions between grids and both FFTs and GPUs are used to accelerate the calculation of these cross correlation functions. To see the extent to which FFTs and GPUs can accelerate the calculation, we applied the FFT approach to a test example utilizing the protein-ligand complex 1G9V(PDB ID). The ligand in 1G9V has dimensions of $5.8\text{Å} \times 14.5\text{Å} \times 8.5\text{Å}$ in the X, Y, and Z directions, respectively. With a grid spacing distance of 0.5Å, the ligand grid has $13 \times 30 \times 18$ points. The binding pocket is defined as a cubic box with a dimension of 29.5Å, and the protein potential grid with the same grid spacing distance as the ligand grid has 60 grid points in all three directions. Therefore, within the binding pocket, the ligand has $59,220 = 47 \times 30 \times 42$ possible positions. The interaction energy between the protein and the ligand for all possible positions of the ligand is calculated using three methods: the naive method which explicitly calculates the interaction energy for each position on a CPU,

Table 2.2: Wall time used by the three methods: the naive method looping through all positions on a CPU, FFTs (CPU), and FFTs (GPU) to calculate interaction energies between the protein and ligand in 1G9V for the ligand's 59,220 positions.

Methods	Naive(CPU[a])	FFTs(CPU[a])	FFTs(GPU[b])
Wall time (seconds)	31.20	0.28	0.002[c]

[a] The CPU used is the Intel Xeon Processor E5645 2.4GHz
[b] The GPU used is the NVIDIA GeForce GTX 1080
[c] Multiple FFTs run in parallel on GPUs in batch mode. The wall time is calculated as the wall time used to run one batch of FFTs divided by the batch size which is 100.

FFTs running on a CPU, and FFTs running on a GPU. The wall times used by the three methods are summarized in Table 2.2. Compared with the naive method, the FFT approach with CPUs accelerates the calculation by more than 100 times and running FFTs on GPUs in batch mode further accelerates the calculation by 140 fold. Overall, compared with the naive method, the speedup of using both FFTs and GPUs is about 15,000 fold.

2.3.1.2 The scoring function's accuracy in identifying ligand native orientations and positions.

With the acceleration of both FFTs and GPUs for calculating the interaction energy between ligands and proteins, it becomes feasible to systematically search ligand orientations and positions in a reasonable computation time. This, in turn, enables us to investigate the scoring function's accuracy in terms of identifying ligand native orientations and positions given the conformations of both the ligand and the protein. Using the Astex diverse set and the SB2012 set as test sets, we applied the FFT-based approach with GPUs to rigidly dock ligands onto proteins using the native conformations of ligands and proteins. To systematically search the orientation and translation space of ligands, 100,000 randomly sampled orientations of each ligand are used. For each orientation, the ligand's translational space is uniformly covered by a 3 dimensional grid with a grid spacing distance of 0.5Å. The docked pose of a ligand is chosen to be the lowest energy pose among the poses with all possible combinations of sampled orientations and translations.

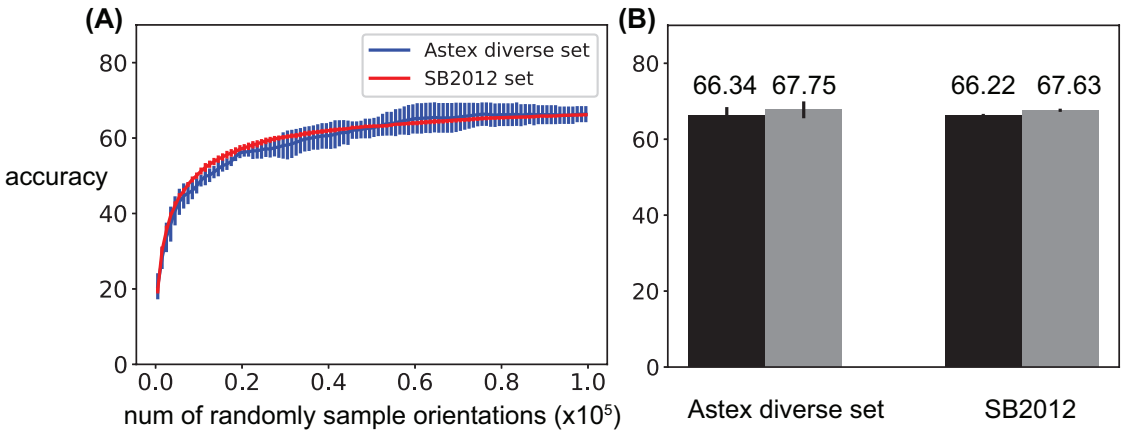

Figure 2.2: Docking accuracy of the FFT approach for docking rigid ligands onto rigid proteins with the native conformations of both ligands and proteins using the Astex diverse set and the SB2012 set. **(A)** Docking accuracy increases as the number of randomly sampled orientations increases. The error bars are estimated using 10 independent repeats. **(B)** Docking accuracy when 100,000 randomly sampled orientations are used (black) and when 100,000 randomly sampled orientations plus the native orientation are used (grey).

For both test sets, the docking accuracy first increases as the number of randomly sampled ligand orientations increases and reaches a plateau when 100,000 random orientations are used (Fig. 2.2A). This plateau occurs at a docking accuracy of about 66.34% and 66.22% for the Astex diverse set and the SB2012 set, respectively. (Fig. 2.2B). When the native orientation is included, in addition to the 100,000 random orientations, the docking accuracy increases to about 67.75% and 67.63% for the Astex diverse set and the SB2012 set, respectively. (Fig. 2.2B). It is notable that this small difference suggests that the use of 100,000 rotational samples is sufficiently dense to cover the rotational space. The docking accuracy at the plateau, which is around 68%, represents the accuracy of the CHARMM force field in identifying the native orientations and positions of ligands assuming the native conformations of ligands are given. This accuracy should be an upper bound of the accuracy of the CHARMM force field in identifying the native ligand poses, which includes the native conformations in addition to the native orientations and positions. Although the size of the SB2012 set is more than ten times larger than the Astex diverse set, the results on the two sets are quite similar. This suggests that the CHARMM force field does not over

fit a specific set of protein-ligand complexes. This should be the case since the CHARMM CGenFF force field together with the CHARMM protein force field representing the protein grid are transferable force fields. This contrasts the anticipated behavior of purely empirical scoring functions for docking [12, 11, 40, 41, 42, 43], which utilize data sets of known protein ligand complexes to optimize the parameters of their scoring function. This suggests that the scope of application of transferable force fields like those used in CDOCKER should be much broader than that of empirical scoring functions.

We note that the above FFT-based rigid ligand, rigid receptor docking approach could be generalized to permit ligand conformational space to be sampled. This would involve first sampling a suitable ensemble of ligand conformations [44] and then carrying out the rotational/translational sampling to identify the lowest energy conformation using GPU-accelerated FFTs. This protocol can readily be implemented using CHARMM scripting language [45]. However, we instead pursue in the following integration of ligand (and possibly receptor side chain) sampling into an MD simulated annealing scheme as employed in CDOCKER [7] and Flexible CDOCKER [9].

2.3.2 Parallel MD-based simulated annealing with GPUs.

2.3.2.1 Speedup of parallel MD-based simulated annealing with GPUs compared with the original CDOCKER with CPUs.

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

trials.

Table 2.3: Speedup of parallel MD-based simulated annealing with GPUs compared with the original CDOCKER with CPUs on the Astex diverse set.

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

[a] The accuracy when 100 trials are used. The ligand native conformation is used as the starting conformation.
[b] The wall time used when 100 trials are used.
[c] The wall time used when 500 trials are used.

2.3.2.2 Comparison with other protein-ligand docking programs.

The accelerated CDOCKER is compared with three other widely used protein-ligand docking programs including Autodock, Autodock Vina, and DOCK. The re-docking results on the Astex diverse set and the SB2012 set are shown in Table 2.4 and Table 2.5, respectively. With the acceleration achieved by the parallel MD-based simulated annealing with GPUs in CDOCKER, the average wall time required by CDOCKER for docking one protein-ligand complex is either faster than or on par with other programs. For CDOCKER, Autodock, and Autodock Vina, their docking accuracies depend on whether ligands' native or random conformations are used as starting conformations. Staring with ligands' native conformations makes the conformational search easier and the docking accuracies much higher than their docking accuracies which are corresponding to using ligands' random conformations as starting conformations. Because the DOCK program uses the "anchor and grow" search method[13], its accuracy does not depend on the starting conformations of ligands.

Based on the result from the Astex diverse set, when ligand random conformations are used as starting conformations, DOCK and Autodock Vina have similar and highest docking accuracy. Autodock has the lowest docking accuracy and CDOCKER is in between. Increasing the parameter that controls the searching exhaustiveness in Autodock

Table 2.4: Docking accuracy of multiple protein-ligand docking programs on the Astex diverse set.

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

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

Vina from 8 to 20 proportionally increases the running time, but it does not change its docking accuracy significantly. Compared with the results on the Astex diverse set (Table 2.4), the relative performance of the protein-ligand docking programs for the SB2012 set is the same in terms of docking accuracy (Table 2.5). However, for all the programs, the docking accuracies are significantly lower on the SB2012 set (Table 2.5) than that on the Astex diverse set. Although the Astex diverse set contains a diverse set of protein-ligand complexes, the number of protein-ligand complexes in the set is relatively small. Because the SB2012 dataset contains more than an order of magnitude more protein-ligand complexes, the performance on the SB2012 set should be a more objective measure of the protein-ligand docking programs' docking accuracies. The lower docking accuracies on the SB2012 set for all the tested protein-ligand docking programs can be attributed to either search algorithms or scoring functions or both. In the case of Autodock Vina, increasing the exhaustiveness from 8 to 20 only slightly improves its docking accuracy, which implies that the empirical scoring function used in Autodock Vina might over fit, to some extent, the protein-ligand complexes that are used to parameterize its scoring function. In

Table 2.5: Docking accuracy of multiple protein-ligand docking programs on the SB2012 set.

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

[a] Ligand native conformations are used as starting conformations.
[b] Ligand random conformations are used as starting conformations.
[c] 500 trials are used in CDOCKER.
[d] exhaustiveness = 8.
[e] exhaustiveness = 20.

the cases of both CDOCKER and DOCK, because their scoring functions are based on MD force fields that are more physically realistic, their lower performance on the SB2012 set are more likely because of search algorithms. The docking accuracies of both Autodock Vina and DOCK reported in this study are quite different from those reported in previous studies [12, 13, 46]. It is because of the fact, as shown in this study, that the docking accuracy of a protein-ligand docking program can vary significantly depending on ligand starting conformations and benchmark datasets.

2.4 Conclusion and Discussion

Two new features — fast Fourier transform (FFT) docking and parallel MD-based simulated annealing — are implemented and added to the protein-ligand docking program CDOCKER in CHARMM. The FFT docking not only utilizes the acceleration provided by FFTs but also employs the parallel computing power of GPUs. Overall, FFT docking with GPUs accelerates the search of ligand's positions and orientations by as much as 15,000 fold. With the significant speedup achieved by FFT docking with GPUs, it becomes practical to almost exhaustively search the translation and rotation space of ligands when docking rigid ligands into binding pockets. Although FFT docking alone can not solve

the protein-ligand problem in which ligands are flexible, the FFT docking can be used to quickly to calculate an upper bound of the docking accuracy that can be achieved by a scoring function. This in turn can provide insights into the problems of current scoring functions and help improve the scoring function. In addition, because FFT docking with GPUs can efficiently calculate protein ligand interaction energies for an almost exhaustive list of positions and orientations given a ligand conformation, FFT docking could also be used to explicitly calculate the partition function corresponding to ligands' translational and rotational space, which can be combined with existing scoring functions in protein-ligand docking to more accurately estimate protein-ligand binding affinities. A similar idea has been investigated by Nguyen et. al.[47] The parallel MD-based simulated annealing with GPUs enables CDOCKER to run about 20 times faster when 500 trials of simulated annealing are used. The speedup becomes even larger when more trials of simulated annealing are employed. With the acceleration, the speed of CDOCKER is on par with or faster than several other popular protein-ligand docking programs tested in this study.

CHAPTER 3

Free Energy Calculation

Ding, Xinqiang, Jonah Z. Vilseck, Ryan L. Hayes, and Charles L. Brooks III. "Gibbs sampler-based λ-dynamics and Rao-Blackwell estimator for alchemical free energy calculation." *Journal of chemical theory and computation* 13, no. 6 (2017): 2501-2510.

3.1 Background

Free energy calculation is fundamental for understanding many important biophysical processes, such as protein conformational changes, protein-protein interactions, and protein-ligand binding processes.[4, 48] Calculating protein-ligand binding free energy has important applications in drug discovery, especially in the lead compound generation and optimization stages.[6, 49, 50] These stages only require calculating protein-ligand relative binding free energy, which has been shown to be easier than calculating protein-ligand absolute binding free energy.[6, 49]

One widely used methodology for calculating protein-ligand relative binding free energy is the alchemical free energy approach.[6, 49, 50] This approach utilizes the thermodynamic cycle shown in Figure 3.1.[48] This thermodynamic cycle specifies that $\Delta\Delta G^{\text{binding}}_{L_0\to L_1} = \Delta G^{\text{binding}}_{L_1} - \Delta G^{\text{binding}}_{L_0} = \Delta G^{\text{bound}}_{L_0\to L_1} - \Delta G^{\text{unbound}}_{L_0\to L_1}$. In order to calculate the relative binding free energy between ligand L_0 and L_1 with receptor R, i.e, $\Delta\Delta G^{\text{binding}}_{L_0\to L_1}$, the alchemical free energy method calculates $\Delta G^{\text{unbound}}_{L_0\to L_1}$ and $\Delta G^{\text{bound}}_{L_0\to L_1}$ by employing alchemical trans-

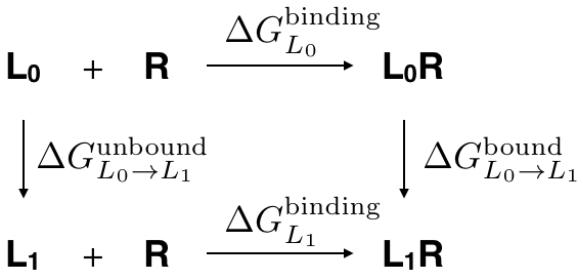

Figure 3.1: The thermodynamic cycle used for calculating a relative binding free energy between ligand L_0 and L_1 with a receptor R.

formations morphing ligand L_0 into ligand L_1 in both unbound and bound environments, respectively.

3.2 Previous methods

Several alchemical free energy calculation methods have been developed over the last several decades, such as free energy perturbation[51, 52], thermodynamic integration[48, 53], enveloping distribution sampling[54, 55] and λ-dynamics[56, 57, 58, 59, 60, 61, 62]. λ-dynamics is a generalized ensemble method in which the alchemical transformation variable λ is a continuous variable ranging from 0 to 1, with $\lambda = 0$, $0 < \lambda < 1$, and $\lambda = 1$ corresponding to the ligand being in L_0 state, intermediate hybrid states, and L_1 state, respectively. The potential energy corresponding to λ is

$$V(\lambda, \{x_i\}_{i=0}^{1}, X) = (1-\lambda)V_0(x_0, X) + \lambda\big(V_1(x_1, X) + G_1^b\big) + V_{\text{env}}(X), \tag{3.1}$$

where X, x_0 and x_1 are atomic coordinates associated with the environment, the ligand L_0 and the ligand L_1, respectively. $V_i(x_i, X)$ is the potential energy between ligand L_i and the environment and $V_{\text{env}}(X)$ is the potential energy of the environment. G_1^b is a biasing potential to ensure that the two physical states, corresponding to $\lambda = 0$ and $\lambda = 1$, are both sampled in the simulation. The biasing potential G_1^b is determined iteratively by running multiple short simulations[59, 60, 63, 64]. The dynamics of the system $(\lambda, \{x_i\}_{i=0}^{1}, X)$ is

generated from the extended Hamiltonian:

$$H(\lambda, \{x_i\}_{i=0}^{1}, X) = T_{x,X} + T_\lambda + V(\lambda, \{x_i\}_{i=0}^{1}, X) \tag{3.2}$$

where $T_{x,X}$ and T_λ are the kinetic energy associated with coordinates $(\{x\}_{i=0}^{1}, X)$ and λ, respectively. The free energy difference between ligand L_0 and L_1, with the biasing potential G_1^b, is

$$\Delta G = -\beta^{-1} \ln \frac{P(\lambda = 1)}{P(\lambda = 0)}, \tag{3.3}$$

where β is the inverse temperature; $P(\lambda = 0)$ and $P(\lambda = 1)$ are probability denisties of λ at points 0 and 1, respectively. In practice, this free energy difference ΔG is estimated using the following empirical estimator based on the trajectory of λ:

$$\Delta \hat{G} = -\beta^{-1} \ln \frac{P(\lambda > \lambda_{\text{cutoff}})}{P(\lambda < 1 - \lambda_{\text{cutoff}})}, \tag{3.4}$$

where λ_{cutoff} $(0 < \lambda_{\text{cutoff}} < 1)$ is a cutoff value which is chosen to be close to 1.[59]

Although the empirical estimator is straightforward to evaluate based on the λ trajectory, it is not necessarily optimal. One issue is that the empirical estimator is systematically biased as it uses $P(\lambda < 1 - \lambda_{\text{cutoff}})$ and $P(\lambda > \lambda_{\text{cutoff}})$ to approximate $P(\lambda = 0)$ and $P(\lambda = 1)$, respectively. Additionally, the bias depends on the cutoff value λ_{cutoff}, which is chosen empirically and is difficult to quantify as it may vary among different systems.

In the current work, we present a novel form of λ-dynamics called the Gibbs sampler based λ-dynamics (GSLD) with the Rao-Blackwell estimator (RBE). The Gibbs sampler framework for calculating free energy differences between two ligands was first suggested by Chodera and Shirts[65]. In their work, λ was treated as a discrete variable and MBAR[66] was used to estimate the free energy change. In this study, we show that GSLD and RBE can treat λ as either a discrete variable or a continuous variable when calculating free energy differences between two ligands. When λ is treated as a continuous variable,

GSLD and RBE can be generalized to simultaneously calculate free energies of multiple ligands in one simulation, as in the generalization of λ-dynamics[59]. We explore these new methods through applications to three model systems in this paper. This paper is organized as follows. In section 2, we describe GSLD and its generalization to multiple ligands. Then we introduce the RBE and show that the MBAR/UWHAM equations [66, 67, 68] can be derived from the RBE. In section 3, we give detailed setup information for the setup and simulation of the three systems with which we tested the methods. Our results for these three systems are presented in section 4. We conclude with a discussion of how the GSLD and RBE can be used for other applications.

3.3 Gibbs sampler based λ-dynamics

As a generalized ensemble method, GSLD samples from the joint distribution of λ and the atomic coordinates of the system using the Gibbs sampler. In this section, we first briefly introduce the Gibbs sampler. We then use the Gibbs sampler to formulate pairwise GSLD. We conclude by showing how the GSLD can be generalized to work for multiple ligands.

3.3.1 The Gibbs sampler.

The Gibbs sampler, which is widely used in both statistics and machine learning, is a Markov Chain Monte Carlo (MCMC) method for sampling from multivariate distributions[69, 70]. To sample (X, Y) from the joint distribution: $(X, Y) \sim P(X, Y)$, the Gibbs sampler generates a Markov chain of states $\{(X_t, Y_t), t = 0, 1, 2, ..., N\}$ using the following procedure:

- **Step 0:** initialize the starting state (X_0, Y_0).
- **Step t:** sample from the conditional distribution

– **Updating X**: given the state (X_{t-1}, Y_{t-1}) from step $t-1$, sample X_t from the conditional distribution of $X_t \sim P(X_t|Y_{t-1})$.

– **Updating Y**: given X_t from the above update step, sample Y_t from the conditional distribution of $Y_t \sim P(Y_t|X_t)$. The resulting sample (X_t, Y_t) is the state for step t.

Because the above procedure satisfies the detailed balance condition with respect to the joint distribution: $(X, Y) \sim P(X, Y)$, the sampled states $\{(X_t, Y_t), t = 0, 1, 2, ..., N\}$ converge to the joint distribution.[69, 70] The update steps require sampling from both conditional distributions: $X_t \sim P(X_t|Y_{t-1})$ and $Y_t \sim P(Y_t|X_t)$. If direct sampling from the conditional distribution is possible, independent samples can be directly drawn using numerical pseudo-random number generators. Otherwise, samples can be drawn using other Monte Carlo methods or Hamiltonian dynamics, as long as the method satisfies the detailed balance condition with respect to the corresponding conditional distribution.[70, 71] This property of the Gibbs sampler makes it quite flexible on choosing appropriate sampling methods based on the conditional distributions.

3.3.2 Pairwise GSLD.

Pairwise GSLD calculates the free energy difference between two ligands: ligand L_0 and ligand L_1. In pairwise GSLD, λ can be treated as either a continuous variable or a discrete variable.

Continuous λ. When λ is treated as a continuous variable, pairwise GSLD samples from the joint distribution of $(\lambda, \{x_i\}_{i=0}^{1}, X)$:

$$P(\lambda, x_0, x_1, X) = \frac{\exp(-\beta\Big[(1-\lambda)V_0(x_0, X) + \lambda\big(V_1(x_1, X) + G_1^b\big) + V_{\text{env}}(X)\Big])}{Z}, \quad (3.5)$$

where Z is the partition function of the generalized ensemble and G_1^b is a biasing potential. G_1^b is determined automatically in the current simulations using a Wang-Landau like

algorithm[72] which is described in 3.7.1. The Gibbs sampler for sampling from the above joint distribution is as follows:

- **Step 0:** initialize the starting state $(\lambda^0, \{x_i^0\}_{i=0}^1, X^0)$.

- **Step t:** sample from the conditional distributions:

  - **Updating** $(\{x_i\}_{i=0}^1, X)$: given the state $(\lambda^{t-1}, \{x_i^{t-1}\}_{i=0}^1, X^{t-1})$ from step $t-1$, sample $(\{x_i^t\}_{i=0}^1, X^t)$ from the conditional distribution: $P(\{x_i^t\}_{i=0}^1, X^t|\lambda^{t-1}) \propto \exp(-\beta\Big[(1-\lambda^{t-1})V_0(x_0^t, X^t) + \lambda^{t-1}\big(V_1(x_1^t, X^t) + G_1^b\big) + V_{\text{env}}(X^t)\Big])$, which is the canonical ensemble distribution at the inverse temperature β. A sample can be drawn from this distribution using molecular dynamics simulation.

  - **Updating** λ: given the atomic coordinates $(\{x_i^t\}_{i=0}^1, X^t)$ sampled from the above update step, sample λ^t directly from the conditional distribution $P(\lambda^t|\{x_i^t\}_{i=0}^1, X^t)$ using numerical pseudo-random number generator. The conditional distribution $P(\lambda^t|\{x_i^t\}_{i=0}^1, X^t)$ is:

$$
\begin{aligned}
&P(\lambda^t|\{x_i^t\}_{i=0}^1, X^t) \\
&= \frac{\exp(-\beta\Big[(1-\lambda^t)V_0(x_0^t, X^t) + \lambda^t\big(V_1(x_1^t, X^t) + G_1^b\big) + V_{\text{env}}(X^t)\Big])}{\int_0^1 \exp(-\beta\Big[(1-\lambda^t)V_0(x_0^t, X^t) + \lambda^t\big(V_1(x_1^t, X^t) + G_1^b\big) + V_{\text{env}}(X^t)\Big])d\lambda} \\
&= \frac{\beta \cdot (\Delta V_{0\to1}^t + G_1^b)\exp(-\lambda^t \cdot \beta \cdot [\Delta V_{0\to1}^t + G_1^b])}{1 - \exp(-\beta \cdot [\Delta V_{0\to1}^t + G_1^b])} \quad (0 \le \lambda^t \le 1),
\end{aligned}
\tag{3.6}
$$

    where $\Delta V_{0\to1}^t = V_1(x_1^t, X^t) - V_0(x_0^t, X^t)$. This is an exponential distribution of λ^t restricted on the interval of $[0, 1]$. Therefore, sampling λ^t directly from this distribution can be done using the inverse transformation method:

$$
\lambda^t = -\frac{1}{\beta \cdot (\Delta V_{0\to1}^t + G_1^b)} \ln\Big[1 - [\,1 - e^{\beta\cdot(\Delta V_{0\to1}^t + G_1^b)}\,] \cdot u\Big] \tag{3.7}
$$

    where u is a random sample from the uniform distribution on $[0, 1]$. The resulting sample $(\lambda^t, \{x_i^t\}_{i=0}^1, X^t)$ is the state for step t.

Discrete λ. When λ is a discrete variable specified by the set $\{l_1, l_2, ...l_M\}$, GSLD samples from the joint distribution

$$P(\lambda = l_j, x_0, x_1, X) \propto \text{exp}(-\beta\Big[V_0(x_0, X, 1-l_j) + V_1(x_1, X, l_j) + G_j^b + V_{\text{env}}(X)\Big]), \quad (3.8)$$

where G_j^b is the biasing potential added to the state corresponding to $\lambda = l_j$. Sampling from this distribution is done in the same way as the case where λ is continuous except that the conditional distribution $P(\lambda^t|\{x_i^t\}_{i=0}^1, X^t)$ becomes a multinomial distribution:

$$\begin{aligned} &P(\lambda^t = l_j|\{x_i^t\}_{i=0}^1, X^t) \\ &= \frac{\text{exp}(-\beta\Big[V_0(x_0^t, X^t, 1-l_j) + V_1(x_1^t, X^t, l_j) + G_j^b\Big])}{\sum_{k=1}^M \text{exp}(-\beta\Big[V_0(x_0^t, X^t, 1-l_k) + V_1(x_1^t, X^t, l_k) + G_k^b\Big])} \end{aligned} \quad (3.9)$$

from which samples can also be drawn directly using numerical methods. The biasing potentials G_j^b are determined similarly as the case when λ is continuous. We note that equation 3.9 is similar to the distribution calculated using the infinite swap limit in replica exchange methods.[73, 74, 75, 76]

The advantage of using λ as a discrete variable is that the pairwise GSLD still works when the potential energy $V_i(x_i, X, \lambda)$ is λ dependent, such as when a soft-core Lennard-Jones potential[77] is employed to facilitate sampling. When λ is continuous, using λ dependent $V_i(x_i, X, \lambda)$ will make the normalization constant of the conditional distribution $P(\lambda|\{x_i\}_{i=0}^1, X)$ not analytically integrable and prevent direct sampling from the conditional distribution $P(\lambda|\{x_i\}_{i=0}^1, X)$. However, as shown below, the advantage of using λ as a continuous variable is that the GSLD can be generalized for multiple ligands.

3.3.3 Generalizing GSLD for multiple ligands.

Like λ-dynamics, GSLD can be generalized to calculate the free energies for multiple ligands in one simulation. Assuming there are n ligands, the fraction of the ith ligand in the

hybrid state is represented by λ_i, for $i = 1, 2, ..., n$. The hybrid state is specified by the value of $(\lambda_1, \lambda_2, ..., \lambda_n)$ which satisfies the conditions $\sum_{i=1}^{n} \lambda_i = 1$ and $0 \leq \lambda_i \leq 1, i = 1, 2, ..., n$. The hybrid state's potential energy is defined as: $V(\{\lambda_i\}_{i=1}^n, \{x_i\}_{i=1}^n, X) = \sum_{i=1}^{n} \lambda_i(V_i(x_i, X) + G_i^b) + V_{\text{env}}(X)$, where x_i and X are atomic coordinates associated with the ith ligand and environment, respectively; G_i^b is the biasing potential added for the ith ligand and can be determined similarly as in the pairwise GSLD. Sampling from the generalized ensemble distribution: $P(\{\lambda_i\}_{i=1}^n, \{x_i\}_{i=1}^n, X) \propto \exp(-\beta \cdot V(\{\lambda_i\}_{i=1}^n, \{x_i\}_{i=1}^n, X))$ can be done using the following Gibbs sampler procedure:

- **Step 0**: initialize the starting state $(\{\lambda_i^0\}_{i=1}^n, \{x_i^0\}_{i=1}^n, X^0)$.

- **Step t:** sample from the conditional distributions.

  - **Updating** $(\{x_i\}_{i=1}^n, X)$: given the state $(\{\lambda_i^{t-1}\}_{i=1}^n, \{x_i^{t-1}\}_{i=1}^n, X^{t-1})$ from step $t-1$, sample $(\{x_i^t\}_{i=1}^n, X^t)$ from the conditional distribution $P(\{x_i^t\}_{i=1}^n, X^t | \{\lambda_i^{t-1}\}_{i=1}^n)$ using molecular dynamics simulation.

  - **Updating** $\{\lambda_i\}_{i=1}^n$: given the sample $(\{x_i^t\}_{i=1}^n, X^t)$ from the above update step, the conditional distribution of $\{\lambda_i^t\}_{i=1}^n$ in the set $S = \{(\lambda_1, ..., \lambda_n) \mid \sum_{i=1}^{n} \lambda_i = 1 \text{ and } \lambda_i \geq 0, i = 1, ...n\}$ is given by

$$P(\{\lambda_i^t\}_{i=1}^n | \{x_i^t\}_{i=1}^n, X^t) = \frac{\exp(-\beta \Big[\sum_{i=1}^{n} \lambda_i^t \, [\, V_i(x_i^t, X^t) + G_i^b \,] + V_{\text{env}}(X^t) \Big])}{Z}, \tag{3.10}$$

where

$$\begin{aligned} Z &= \int_S \exp(-\beta \Big[\sum_{i=1}^{n} \lambda_i^t \, [\, V_i(x_i^t, X^t) + G_i^b \,] + V_{\text{env}}(X^t) \Big]) dm_S(\lambda) \\ &= e^{-\beta V_{\text{env}}(X^t)} \sum_{i=1}^{n} \frac{e^{-\beta \Big[V_i(x_i^t, X^t) + G_i^b \Big]}}{\beta^{n-1} \prod_{j \neq i} (\Big[V_j(x_j^t, X^t) + G_j^b \Big] - \Big[V_i(x_i^t, X^t) + G_i^b \Big])}, \end{aligned} \tag{3.11}$$

and $dm_S(\lambda)$ is the infinitesimal volume element of the simplex S. Because $\sum_{i=1}^{n} \lambda_i^t =$

1, the conditional distribution $P(\{\lambda_i^t\}_{i=1}^n|\{x_i^t\}_{i=1}^n, X^t)$ has only $n-1$ degrees of freedom. Sampling from this conditional distribution is equivalent to sampling from the $n-1$ dimensional distribution:

$$\begin{aligned} &P(\{\lambda_i^t\}_{i=1}^{n-1}|\{x_i^t\}_{i=1}^n, X^t) \propto \\ &\exp(-\beta\Big[\sum_{i=1}^{n-1}\lambda_i\,[\,V_i(x_i^t, X^t) + G_i^b - V_n(x_n^t, X^t) - G_n^b\,]\Big]), \end{aligned} \tag{3.12}$$

where $0 \leq \sum_{i=1}^{n-1}\lambda_i^t \leq 1$, and $\lambda_i^t \geq 0$. The environment atom energy term, $V_{\text{env}}(X^t)$, does not appear in equation (3.12) because it is part of both the numerator and denominator of equation (3.10) and can be canceled out as a constant when $(\{x_i^t\}_{i=1}^n, X^t)$ is fixed. Sampling from this $n-1$ dimensional distribution $P(\{\lambda_i^t\}_{i=1}^{n-1}|\{x_i^t\}_{i=1}^n, X^t)$ is done using the rejection method. In the rejection method, each $\{\lambda_i^t\}_{i=1}^{n-1}$ is sampled independently from the distribution: $P(\lambda_i^t) \propto \exp(-\beta\lambda_i\Big[V_i(x_i^t, X^t) + G_i^b - V_n(x_n^t, X^t) - G_n^b\Big])$, where $0 \leq \lambda_i^t \leq 1$. If the sample $\{\lambda_i^t\}_{i=1}^{n-1}$ satisfies the condition $0 \leq \sum_{i=1}^{n-1}\lambda_i^t \leq 1$, it is accepted, otherwise the sample $\{\lambda_i^t\}_{i=1}^{n-1}$ is rejected. This procedure is repeated until a sample $\{\lambda_i^t\}_{i=1}^{n-1}$ is accepted. Set $\lambda_n^t = 1 - \sum_{j=1}^{n-1}\lambda_j^t$ and the resulting sample $(\{\lambda_i^t\}_{i=1}^n, \{x_i^t\}_{i=1}^n, X^t)$ is the state for step t.

3.4 Rao-Blackwell estimators

3.4.1 Rao-Blackwell estimators for λ-dynamics

Although the empirical estimator used in λ-dynamics can also be utilized in GSLD to estimate the free energy, it is not an optimal estimator and may contain a system dependent bias. RBE is introduced here to eliminate these potential issues. RBE is the estimator derived by applying the Rao-Blackwellization transformation to the empirical estimator. Rao-Blackwellization is a statistical method, inspired by the Rao-Blackwell theorem[78, 79], to transform a crude estimator into a better estimator that has smaller mean squared error for

estimating the quantity of interest[80]. Specifically, if $\delta(Z)$ is an estimator of an unknown parameter θ and $T(Z)$ is a sufficient statistics for the parameter θ, the Rao-Blackwellized estimator of the estimator $\delta(Z)$ is the conditional expected value $E(\delta(Z)|T(Z))$[78, 79].

For pairwise GSLD with continuous λ, the quantity of interest is the free energy $\Delta G = -\beta^{-1} \ln \big[P(\lambda = 1)/P(\lambda = 0)\big]$. The values of both $P(\lambda = 1)$ and $P(\lambda = 0)$ are viewed unknown parameters. To estimate ΔG, i.e., $P(\lambda = 1)$ and $P(\lambda = 0)$, the empirical estimator approximates $P(\lambda = 1)$ and $P(\lambda = 0)$ directly by calculating the fraction of λs which are close to 1 and 0, respectively, based on the λ trajectory. In contrast, the RBE ignores the λ trajectory and only uses the atomic coordinate trajectory. Because the coordinate $(\{x_i\}_{i=0}^1, X)$ is a sufficient statistics for the parameters $P(\lambda = 1)$ and $P(\lambda = 0)$, applying the Rao-Blackwellization yields the RBE estimators as $P(\lambda = 1) = \mathbb{E}_{\{\{x_i\}_{i=0}^1, X\}}\big[P(\lambda = 1|\{x_i\}_{i=0}^1, X)\big]$ and $P(\lambda = 0) = \mathbb{E}_{\{\{x_i\}_{i=0}^1, X\}}\big[P(\lambda = 0|\{x_i\}_{i=0}^1, X)\big]$. Therefore, RBE uses the following formula to estimate the free energy ΔG:

$$\begin{aligned}
\Delta G_{\text{RBE}} &= -\beta^{-1} \ln \frac{P(\lambda = 1)}{P(\lambda = 0)} \\
&= -\beta^{-1} \ln \frac{\mathbb{E}_{\{\{x_i\}_{i=0}^1, X\}}\big[P(\lambda = 1|\{x_i\}_{i=0}^1, X)\big]}{\mathbb{E}_{\{\{x_i\}_{i=0}^1, X\}}\big[P(\lambda = 0|\{x_i\}_{i=0}^1, X)\big]} \\
&= -\beta^{-1} \ln \frac{1/N \cdot \sum_{t=0}^N P(\lambda = 1|\{x_i^t\}_{i=0}^1, X^t)}{1/N \cdot \sum_{t=0}^N P(\lambda = 0|\{x_i^t\}_{i=0}^1, X^t)}
\end{aligned} \tag{3.13}$$

where

$$\begin{aligned}
P(\lambda = 1|\{x_i^t\}_{i=0}^1, X^t) &= \frac{\beta \cdot (\Delta V_{0\to1}^t + G_1^b) \cdot \exp(-\beta \cdot \big[\Delta V_{0\to1}^t + G_1^b\big])}{1 - \exp(-\beta \cdot \big[\Delta V_{0\to1}^t + G_1^b\big])} \\
P(\lambda = 0|\{x_i^t\}_{i=0}^1, X^t) &= \frac{\beta \cdot (\Delta V_{0\to1}^t + G_1^b)}{1 - \exp(-\beta \cdot \big[\Delta V_{0\to1}^t + G_1^b\big])},
\end{aligned} \tag{3.14}$$

and N is the number of samples.

For the generalized GSLD with multiple ligands, the RBE can be derived similarly. To estimate the free energy of the ith ligand given by $G(\lambda_i = 1, \lambda_{j\neq i} = 0) = -\beta^{-1} \ln P(\lambda_i =$

$1, \lambda_{j\neq i} = 0)$, the RBE uses the following formula:

$$\begin{aligned} G_{\text{RBE}}(\lambda_i = 1, \lambda_{j\neq i} = 0) &= -\beta^{-1} \ln P(\lambda_i = 1, \lambda_{j\neq i} = 0) \\ &= -\beta^{-1} \ln \mathbb{E}_{\{\{x_k\}_{k=1}^n, X\}} \Big[P(\lambda_i = 1, \lambda_{j\neq i} = 0 | \{\{x_k\}_{k=1}^n, X\}) \Big] \\ &= -\beta^{-1} \ln \Big[1/N \cdot \sum_{t=0}^{N} P(\lambda_i = 1, \lambda_{j\neq i} = 0 | \{\{x_k\}_{k=1}^n, X\}) \Big] \\ &= -\beta^{-1} \ln \Big[1/N \cdot \sum_{t=0}^{N} \frac{\exp(-\beta \big[V_i(x_i^t, X^t) + G_i^b \big])}{Z} \Big], \end{aligned} \tag{3.15}$$

where Z is given in equation 3.11 in **section 2.1.3**.

As shown in the above formulas, the RBE estimator ΔG_{RBE} does not depend on the empirical cutoff value of λ_{cutoff}. Based on the Rao-Blackwell theorem, ΔG_{RBE} is an unbiased estimator. In addition, if the samples from GSLD are independent, the mean squared error of RBE is guaranteed to be smaller than or equal to that of the empirical estimator. Although the samples from GSLD are usually not truly independent, the advantage of RBE can often be justified empirically.[81]

3.4.2 Derivation of the MBAR/UWHAM equations using RBE

Although RBE is originally introduced to estimate free energies based on sampling from GSLD, RBE can also be used when multiple equilibrium states are sampled independently. When RBE is applied to this case, it generates the MBAR/UWHAM equations[66, 67, 68], which are widely used in current alchemical free energy methods.

Let us assume there are M equilibrium states with potential energy function of $V_i, i = 1, 2, ..., M$. Each equilibrium state is sampled independently. The conformations sampled from state i are represented as $x_i^k, k = 1, 2, ..., n_i$, where n_i is the number of conformations from state i. The total number of conformations is $N = \sum_{j=1}^{M} n_j$. The free energy of state i is represented as G_i^*. We use $\lambda \in \{1, 2, ..., M\}$ as an index variable to represent the

M equilibrium states, with $\lambda = i$ corresponding to state i. To calculate the free energies for all the equilibrium states, all the conformations $\{x_i^k, i = 1, 2, ..., M, k = 1, 2, ..., n_i\}$ are pooled together and viewed as samples from the generalized ensemble $P(\lambda = i, x) \propto e^{-\beta[V_i(x)+G_i^b]}$, where G_i^b is the biasing energy added to state i to adjust the relative weight of state i to be proportional to n_i, i.e, G_i^b needs to satisfy the condition:

$$G_i = G_i^* + G_i^b = -\beta^{-1}\ln\frac{n_i}{N}, \tag{3.16}$$

where G_i is the free energy of state i with the biasing potential of G_i^b and G_i^* is the unbiased free energy of state i. We note that the biasing potentials G_i^b in equation 3.16 are unknown variables. They are introduced to make the equation 3.16 valid, which is the requirement for applying the RBE. These unknown biasing potentials G_i^b can be calculated after the values of G_i^* are solved. The RBE for this generalized ensemble is:

$$\begin{aligned} G_i &= -\beta^{-1}\ln P(\lambda = i) \\ &= -\beta^{-1}\ln\frac{1}{N}\sum_{j=1}^{M}\sum_{k=1}^{n_i} P(\lambda = i|x_j^k) \\ &= -\beta^{-1}\ln\frac{1}{N}\sum_{j=1}^{M}\sum_{k=1}^{n_j}\frac{e^{-\beta\left[V_i(x_j^k)+G_i^b\right]}}{\sum_{l=1}^{M} e^{-\beta\left[V_l(x_j^k)+G_l^b\right]}} \end{aligned} \tag{3.17}$$

Combining equation 3.16 with equation 3.17, we have:

$$G_i^* = -\beta^{-1}\ln\sum_{j=1}^{M}\sum_{k=1}^{n_j}\frac{e^{-\beta\left[V_i(x_j^k)\right]}}{\sum_{l=1}^{M} n_l \cdot e^{-\beta\left[V_l(x_j^k)-G_l^*\right]}} \tag{3.18}$$

which is the same as the MBAR/UWHAM equations[66, 67, 68]. Previously, the MBAR/UWHAM equations were derived as either a result of the maximum likelihood principle or an unbinned extension of the weighted histogram analysis method (WHAM).[66, 67, 68] Here we have shown that the MBAR/UWHAM equations can also be derived using RBE.

3.5 Applications of GSLD and RBE

3.5.1 System setups and computational details

To illustrate how GSLD works and the advantage of RBE over the empirical estimator typically used in λ-dynamics, we applied GSLD and RBE to three test cases: **(a)** calculation of the free energy difference between two states of a harmonic oscillator system, **(b)** calculation of the relative hydration free energies of three benzene derivatives, and **(c)** calculation of the binding free energy difference between benzene and p-xylene bound to the L99A mutant of the protein T4 lysozyme[82, 83]. The simulations in these calculations were run using CHARMM[45] compiled with OpenMM[34]. Each calculation was repeated 10 times. Error bars were calculated as the standard variation of the results from these 10 independent repeats.

3.5.1.1 Harmonic System.

The harmonic system consists of a one dimensional particle that switches between two states: state 0 and state 1. Each state has a harmonic potential energy. The purpose is to calculate the free energy difference of the particle when it changes from state 0 to state 1, i.e, $\Delta G = G_1 - G_0$. Specifically, state 0 has a potential energy given by $\frac{1}{2}k_0(x - x_0^e)^2$, and state 1 has a potential energy given by $\frac{1}{2}k_1(x - x_1^e)^2$. In order to prevent the particle from moving too far from the equilibrium position, a restraining potential is added for each state. This restraining potential is not scaled by λ. The resulting hybrid potential energy is:

$$\begin{aligned} V(\lambda, x_0, x_1) = (1-\lambda)\cdot\frac{1}{2}k_0(x_0 - x_0^e)^2 + \lambda\cdot\frac{1}{2}k_1(x_1 - x_1^e)^2 \\ + \frac{1}{2}k_{\text{env}}(|x_0| - x_{\text{env}}^e)^2\mathbb{1}\{|x_0| \geq x_{\text{env}}^e\} + \frac{1}{2}k_{\text{env}}(|x_1| - x_{\text{env}}^e)^2\mathbb{1}\{|x_1| \geq x_{\text{env}}^e\}, \end{aligned}$$

where $\mathbb{1}\{\text{condition}\}$ is equal to 1 if the condition is true, otherwise it is equal to 0. GSLD is used to sample from the joint distribution of $(\lambda, \{x_i\}_{i=0}^1)$: $P(\lambda, \{x_i\}_{i=0}^1) \propto \exp(-\beta \cdot$

$V(\lambda, \{x_i\}_{i=0}^1)$). Given the value of λ, sampling the coordinates ($\{x_i\}_{i=0}^1$) is accomplished by running Langevin dynamics for 1 ps with a step size of 1 fs, temperature of 300 K, and friction coefficient of 10 ps^{-1}. The total simulation time is 10 ns. The parameters used for x_0^e, x_1^e, x_{env}^e and k_{env} are -2.0 Å , 2.0 Å , 4.0 Å , and 2.5 kcal/mol $\cdot$ Å^{-2}, respectively. Two variations of the model system that correspond to setting different values for k_0 and k_1 are used: a symmetrical system with $k_0 = k_1 = 0.75$ kcal/mol $\cdot$ Å^{-2}, and an asymmetrical system with $k_0 = 0.75$ kcal/mol $\cdot$ Å^{-2} and $k_1 = 0.075$ kcal/mol $\cdot$ Å^{-2}.

3.5.1.2 Relative hydration free energies for three benzene derivatives.

Relative hydration free energies for three benzene derivatives: benzene, phenol, and benzaldehyde were calculated from the difference between alchemical free energy changes computed in vacuum and in water. The topology and parameter files for the hybrid ligand were generated using MATCH[84] and in-house developed scripts based on the CHARMM General Force Field (CGenFF)[39]. The simulation in water was done in a water box consisting of 800 TIP3P[85] water molecules with cubic periodic boundary conditions. The water box had a size of 30.0 Å $\times$ 30.0 Å $\times$ 30.0 Å. A nonbonded cutoff of 14 Å was used, and the van der Waals switching function and electrostatic force switching function [86] were used between 12 Å and 14 Å. Sampling from the conditional distribution $P(x, X|\lambda)$ was accomplished by running Langevin dynamics at 298.15 K for 0.2 ps. The time step size was 2 fs and the friction coefficient was 10 ps^{-1}. The length of all bonds involving hydrogen atoms was fixed during the simulation using the SHAKE algorithm[87]. The three relative hydration free energies were first calculated by three independent pairwise GSLDs. Then they were calculated simultaneously using the generalized GSLD for multiple ligands. For comparison, the three relative hydration free energies were also calculated using the FEP/MBAR method, in which 11 states corresponding to $\lambda = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$ were used.

3.5.1.3 Relative binding free energy between benzene and p-xylene with T4 lysozyme.

The L99A mutant of T4 lysozyme has been a model protein system for testing free energy calculation methods[88, 89, 90]. It has experimental binding free energy data for a series of benzene derivatives including benzene and p-xylene. [82, 83] The relative binding free energy between benzene and p-xylene was calculated using the difference between the alchemical free energy change in water and in the protein environment. The alchemical free energy change in water was calculated using pairwise GSLD with continuous λ. Calculating the alchemical free energy in the protein environment is challenging, even though the binding site of T4 lysozyme is a relatively simple non-polar pocket and the alchemical change from benzene to p-xylene is small. This challenge arises from the fact that T4 lysozyme has a conformational change for the side-chain dihedral angle χ (N-CA-CB-CG1) of residue Val111, which accompanies the alchemical transformation from benzene to p-xylene.[88] When T4 lysozyme binds with benzene (PDB ID: 181L), the dihedral angle stays in the *trans* conformation ($\chi \approx -180°$). When it binds with p-xylene (PDB ID: 187L), the dihedral angle changes into the *gauche* conformation ($\chi \approx -60°$). Failing to sample these two relevant conformations in a free energy calculation would cause a quasi-nonergodicity problem, i.e, the calculated free energy will depend on which conformation is used as the starting conformation.[88, 90] To address the problem, several methods have been developed. These methods include enhanced sampling methods such as the 2-dimensional replica exchange method (REM)[89] and the free energy perturbation/replica exchange with solute tempering (FEP/REST) method [90], and the potential of mean force (PMF) method, which was first introduced by Tobias and Brooks for addressing a similar problem in 1989[91] and rediscovered as the "confine-and-release" method by Mobley et al. in 2007.[88] Here we combined the PMF method with GSLD to calculate the alchemical free energy changes between benzene and p-xylene in the protein environment.

To make our computational protocol clear, we reformulated the PMF method[91, 88]

using conditional probability as shown in 3.7.2. The free energy change $\Delta G(\chi^*)$ was calculated using pairwise GSLD with a harmonic restraint potential on χ to keep it near χ^* during the pairwise GSLD simulation. The force constant of the harmonic restraint potential was 1195.3 kcal/mol$\cdot$radius^{-2}. In our calculations, we chose χ^* to be $-180°$ and $-60°$, although the final calculated result ΔG did not depend on the choice of χ^*. In the pairwise GSLD, λ was chosen to be a discrete variable specified by the set $\{l_1, l_2, ..., l_{16}\}$. $\lambda = l_1$ corresponds to the physical state that the ligand is benzene and $\lambda = l_{16}$ corresponds to the physical state that the ligand is p-xylene. When λ was changed from l_1 to l_{16}, the ligand was alchemically transformed from benzene into p-xylene. During the alchemical transformation, the partial charges on benzene atoms were turned off first. Then the benzene atoms were transformed into p-xylene atoms before the partial charges on p-xylene atoms were turned on. A soft-core Lennard-Jones potential was used during the transformation.[92] The formula used for both electrostatic potential and the soft-core Lennard-Jones potential is shown in Table S1. The potential energy scaling factors used for each state $\lambda = l_i$ are also shown in Table S1. The free energy $-\beta^{-1}\ln P(\chi^*|\lambda = l_1)$ and the free energy $-\beta^{-1}\ln P(\chi^*|\lambda = l_{16})$ were computed by calculating the potential of mean force (PMF) with respect to χ when T4 lysozyme binds with benzene ($\lambda = l_1$) and with p-xylene ($\lambda = l_{16}$), respectively. The simulations was run inside a TIP3P water box with a size of 79.0Å $\times$ 56.4Å $\times$ 55.4Å and rectangular periodic boundary conditions were used. The water box had 7112 water molecules in total. The CHARMM36 force field[93] was used for T4 lysozyme and the CHARMM General Force Field (CGenFF)[39] was used for the ligands. The nonbonded interaction options were the same as that used in the relative hydration free energy calculations.

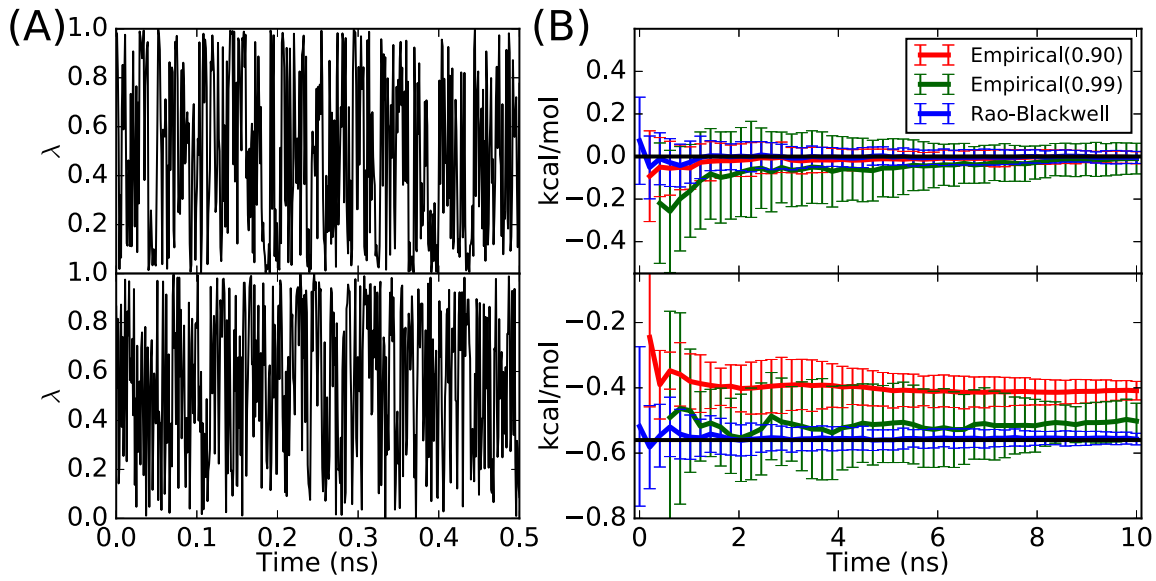

Figure 3.2: Results of pairwise GSLD and RBE on harmonic systems. (A) λ trajectories for the symmetrical harmonic system (top) and the asymmetrical harmonic system (bottom); (B) Free energy estimations for the symmetrical system (top) and the asymmetrical system (bottom) using the empirical estimators with a cutoff of 0.9 and 0.99 and the Rao-Blackwell estimator. The horizontal black line is the calculated free energy change using numerical integration.

3.5.2 Results

3.5.2.1 The harmonic system.

As shown in Figure 3.2(A), GSLD is able to sample the continuous λ well for both symmetrical and asymmetrical systems. Figure 3.2(B) shows the estimated free energy changes ΔG using the Rao-Blackwell estimator and two empirical estimators with cutoff values of 0.9 and 0.99. For the symmetrical system, the true value for the free energy changes is equal to 0 kcal/mol because of the symmetry. The RBE and the empirical estimator with cutoff of 0.9 converge to 0 kcal/mol within 2 ns, whereas the empirical estimator with cutoff of 0.99 needs 10 ns of simulation to converge to 0 kcal/mol. Moreover the RBE has the smallest variance among the three estimators. For the asymmetrical system, the empirical estimator with a cutoff of 0.9 converges to -0.41 ± 0.03 kcal/mol and the empirical estimator with a cutoff of 0.99 converges to -0.50 ± 0.06 kcal/mol, whereas the result from

numerical integration is -0.56 kcal/mol. This shows that the results of empirical estimators can be biased and the bias depends on the value of the cutoff. Increasing the cutoff value decreases the estimation bias, but it increases the estimation variance because a higher cutoff decreases the number of valid samples used by the empirical estimator. In contrast, the result of RBE converges to -0.56 ± 0.02 kcal/mol, which is closest to the true value and also has the smallest variance. The detailed numerical results can be found in the Table S2. Overall, the results suggest that, for this harmonic system, the GSLD is able to extensively sample the alchemical states and the RBE is better than the empirical estimator in terms of both bias and variance.

3.5.2.2 Relative hydration free energies for three benzene derivatives.

Results of pairwise GSLD simulations in vacuum and in water are shown in Figure S1 and Figure 3.3, respectively. The pairwise GSLD is able to sample the alchemical states very well for both the simulations in vacuum and the simulations in water. For the simulation in vacuum, the RBE outperforms empirical estimators in terms of both bias and variance, as in the harmonic system. For the simulation in water, the RBE has a similar variance to that of the empirical estimators, because samples from the simulation in water are more correlated than those from the simulations in vacuum. Nevertheless, the RBE is still better than the empirical estimators in terms of the bias. As shown in Figure 3.3 **(B)**, the empirical estimator depends on the cutoff. As the cutoff increases from 0.9 to 0.99, the empirical estimator results move towards to the RBE results. As an example, for the alchemical change from benzene to benzaldehyde, when the cutoff increases from 0.9 to 0.99, the empirical estimator result changes from 2.20 ± 0.08 kcal/mol to $2.60 \pm .08$ kcal/mol. The RBE result is 3.04 ± 0.09 kcal/mol, which is indistinguishable from the FEP/MBAR result 3.01 ± 0.02 kcal/mol. The detailed numerical values from pairwise GSLD and FEP/MBAR can be found in the Table S3 and S4.

The simulation results in vacuum and in water from generalized GSLD for multiple

ligands are shown in Figure S2 and Figure 3.4, respectively. The ternary plots[94] of $(\lambda_1, \lambda_2, \lambda_3)$ trajectories show that the generalized GSLD is able to explore the hybrid ligand configuration space of $(\lambda_1, \lambda_2, \lambda_3)$: the unit simplex $\{(\lambda_1, \lambda_2, \lambda_3) | \sum_{i=1}^{3} \lambda_i = 1, 0 \leq \lambda_i \leq 1 \text{ for } i = 1, 2, 3\}$, in both vacuum and water. In vacuum, the configuration space $(\lambda_1, \lambda_2, \lambda_3)$ is sampled rather uniformly, while in water, the configuration space is sampled mostly close to the physical states, i.e. the corners of the ternary plot in Figure 3.4. This difference is because the biasing potential energy used in this study is a linear biasing potential $\lambda_i G_i^b$. With the linear biasing potential, the biased free energy landscape over the configuration space $(\lambda_1, \lambda_2, \lambda_3)$ in vacuum is almost flat. In water, the corresponding biased free energy landscape is not flat due to the polarization energy of the solvent interacting with reactant and product states, and the biased free energies of the physical states is lower than the intermediate non-physical states, which explains why the sampled $(\lambda_1, \lambda_2, \lambda_3)$ are mostly around the physical states. Based on the trajectory from the generalized GSLD simulation, the calculated free energy using RBE and empirical estimators are shown in Figure S2 **(B)** and Figure 3.4 **(B)**. These results suggests again that, compared with the empirical estimators, the RBE is a better estimator as it has no bias and a smaller variance. The detailed numerical results from the generalized GSLD for multiple ligands is shown in the Table S5.

The calculated relative hydration free energies for the three benzene derivatives using pairwise GSLD, generalized GSLD for multiple ligands and FEP/MBAR methods are combined in Table 3.1. The results from all three methods agree well with each other. The total simulation time in water for calculating all three relative hydration free energies is 9 ns for pairwise GSLD, 3 ns for generalized GSLD for multiple ligands and 33 ns for FEP/MBAR methods, which suggests the efficacy of the generalized GSLD for multiple ligands.

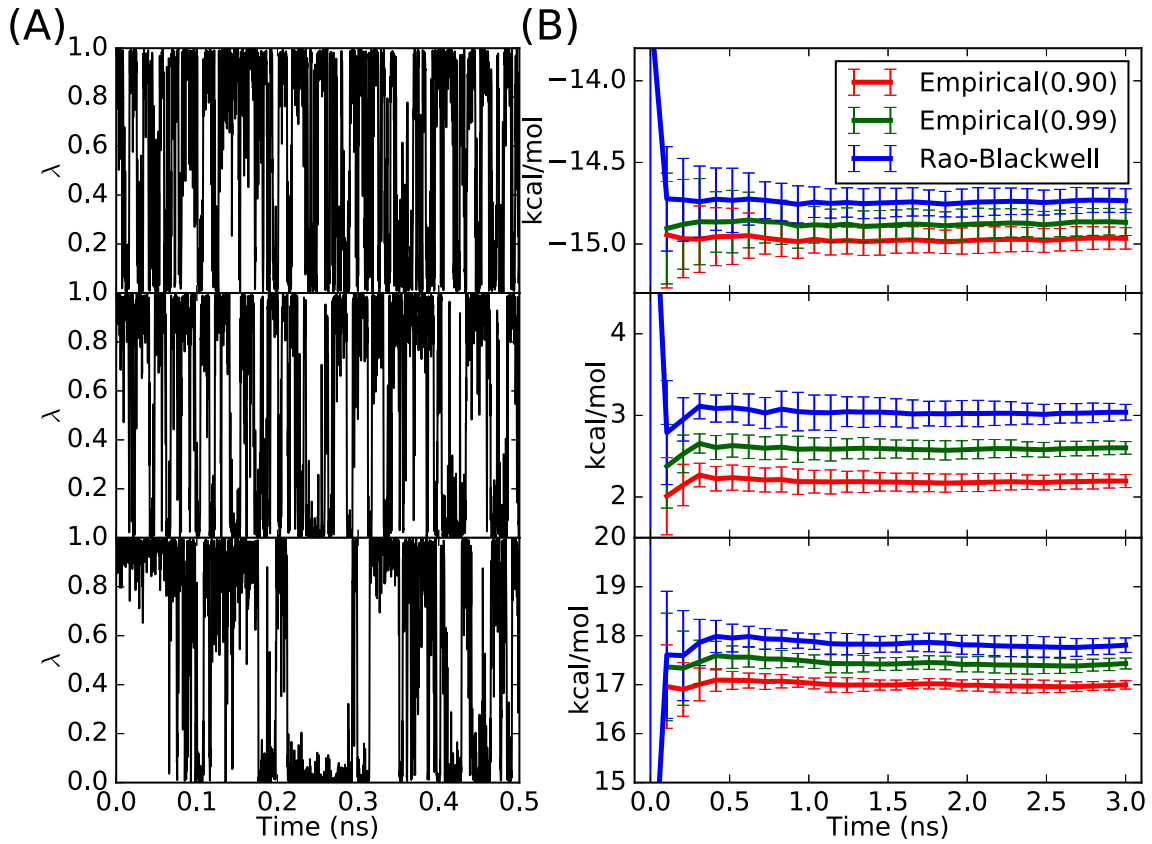

Figure 3.3: Results of pairwise GSLD and RBE for calculating solvation free energies. **(A)** λ trajectories from simulations in water using GSLD for alchemical changes benzene to phenol (top), benzene to benzaldehyde (middle), and phenol to benzaldehyde (bottom). **(B)** Estimated alchemical free energy changes in water using empirical estimators with different cutoff values and the Rao-Blackwell estimator for alchemical changes benzene to phenol (top), benzene to benzaldehyde (middle), and phenol to benzaldehyde (bottom).

3.5.2.3 Relative binding free energy of benzene and p-xylene with T4 lysozyme.

The λ trajectories from the simulation with T4 lysozyme using pairwise GSLD and the free energy estimations using RBE are shown in Figure 3.5. For both the case where χ is restricted to the *trans* conformation ($\chi^* = -180°$) and the case where χ is restricted to the *gauche* ($\chi^* = -60°$) conformation, the pairwise GSLD is able to sample the alchemical switching variable λ well and the RBE estimations converge in 10 ns of simulation. When χ is restricted to the *trans* conformation, the estimated free energy converges to -8.40 ± 0.46 kcal/mol. When χ is restricted in the *gauche* conformation, the estimated free energy

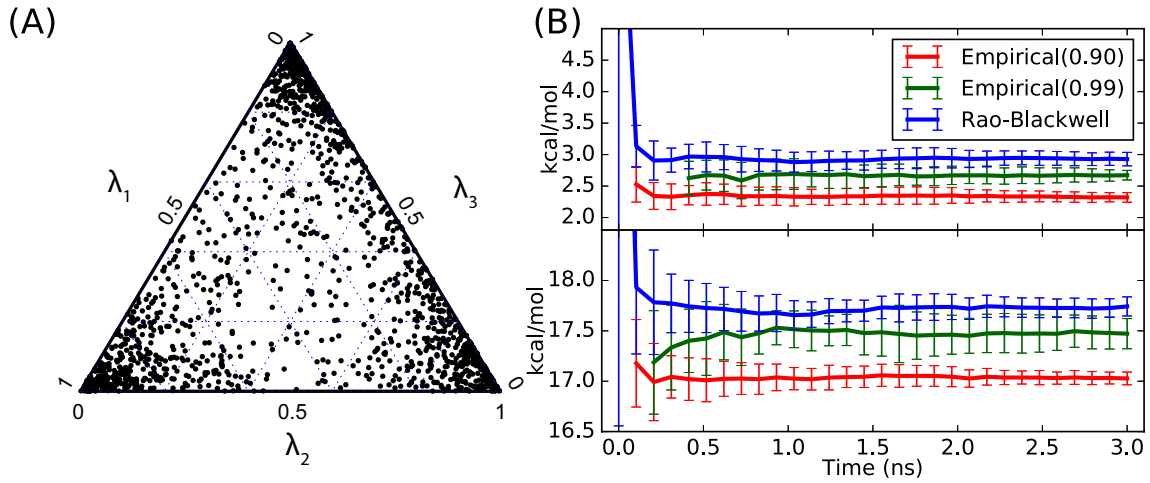

Figure 3.4: Results of generalized GSLD for multiple ligands and RBE for calculating solvation free energies. **(A)** Ternary plot of $(\lambda_1, \lambda_2, \lambda_3)$ sampled using GSLD for multiple ligands in water. **(B)** Estimated free energy changes in water for alchemical changes: benzene to benzaldehyde (top) and phenol to benzaldehyde (bottom) using empirical estimator with different cutoff values and RBE.

Table 3.1: Comparison of Relative Hydration Free Energies ($\Delta\Delta G$ in kcal/mol) for The Three Benzene Derivatives. The total simulation time in water for each method is shown in parenthesis.

substituents change	$\Delta\Delta G_{exp}$	Pairwise GSLD $\Delta\Delta G$(9ns)	GSLD for Multiple Ligands $\Delta\Delta G$(3ns)	FEP/MBAR $\Delta\Delta G$(33ns)
Benzene $\rightarrow$ Phenol	-5.77	-4.46 ± 0.08	-4.53 ± 0.15	-4.46 ± 0.03
Benzene $\rightarrow$ Benzaldehyde	-3.18	-3.11 ± 0.11	-3.22 ± 0.11	-3.13 ± 0.03
Phenol $\rightarrow$ Benzaldehyde	2.59	1.39 ± 0.17	1.31 ± 0.10	1.34 ± 0.14

converges to -10.60 ± 0.36 kcal/mol. These two free energy estimations are different by 2.20 kcal/mol because the dihedral angle χ is restricted to different conformations. Based on the PMF method, in order to get the free energy corresponding to the case where χ is not restricted, the restricting free energies ($-\beta^{-1} \ln P(\chi^*|\lambda = l_1)$ and $-\beta^{-1} \ln P(\chi^*|\lambda = l_{16})$) need to be considered and used to correct the free energy $\Delta G(\chi^*)$ using equation 3.21 in **Appendix B**. These corrections are shown in Table 3.2. After the corrections, the estimated free energy ΔG is -9.27 ± 0.50 kcal/mol when $\chi^* = -180°$ and -9.01 ± 0.40 kcal/mol when $\chi^* = -60°$. Therefore, after the corrections, the estimated free energy differences (ΔG) agree very well within statistical uncertainty. Based on the these corrected values, the relative binding free energies ($\Delta\Delta G$) are 0.27 ± 0.56 kcal/mol and 0.43 ± 0.46 kcal/mol

when $\chi^* = -180°$ and $\chi^* = -60°$, respectively. These results are close to the relative binding free energy from experiment, which is 0.52 ± 0.22 kcal/mol [82, 83].

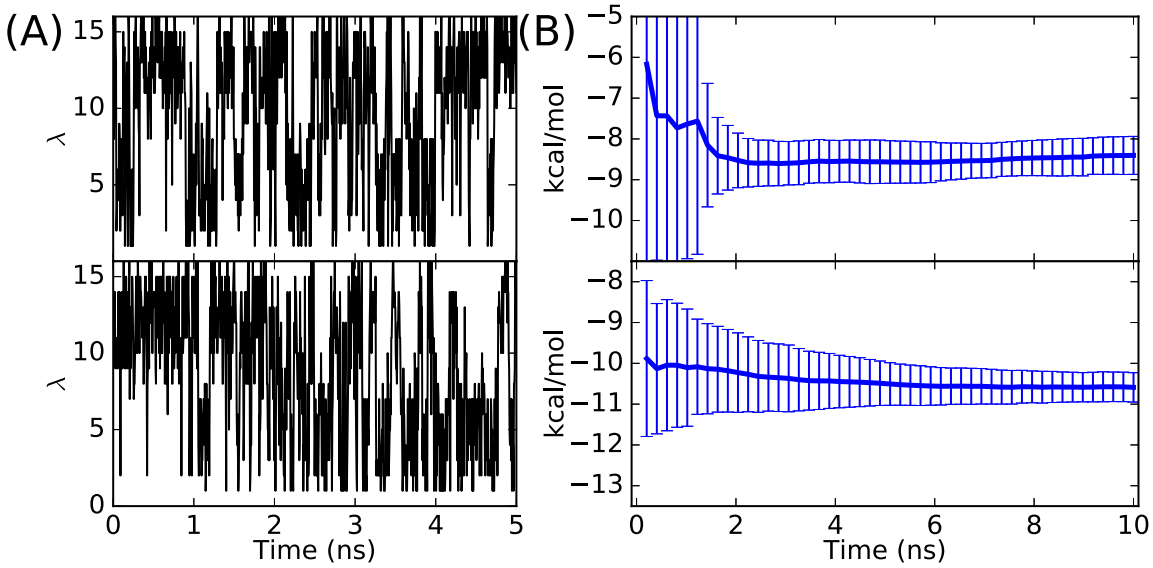

Figure 3.5: Results of GSLD and RBE for calculating relative binding free energy between benzene and p-xylene with T4 lysozyme. (A) λ trajectories for simulations with T4 lysozyme using pairwise GSLD for the $\chi^* = -180°$ (top) and $\chi^* = -60°$ (bottom); (B) Free energy estimation using RBE for $\chi^* = -180°$ (top) and $\chi^* = -60°$ (bottom).

Table 3.2: Alchemical Free Energy Changes (kcal/mol) Between Benzene and p-Xylene Binding with T4 Lysozyme Calculated Using Pairwise GSLD with Corrections from PMFs.

χ^*	$\Delta G(\chi^*)$	$-\beta^{-1}\ln P(\chi^*\|\lambda=0)$	$-\beta^{-1}\ln P(\chi^*\|\lambda=1)$	ΔG	$\Delta\Delta G$
trans ($\chi^* = -180°$)	-8.40 ± 0.46	-0.47 ± 0.01	0.4 ± 0.03	-9.27 ± 0.50	0.27 ± 0.56
gauche ($\chi^* = -60°$)	-10.60 ± 0.36	1.14 ± 0.03	-0.45 ± 0.01	-9.01 ± 0.40	0.43 ± 0.46

The alchemical free energy change ΔG in water is -9.44 ± 0.06 kcal/mol and the experimental relative binding free energy $\Delta\Delta G$ is 0.52 ± 0.22 kcal/mol.

3.6 Discussion

Although the GSLD and RBE are applied only for calculating relative hydration free energy and relative binding free energy in this study, they could also be used for other purposes. One of the applications would be for calculating the pK_a value of protein amino acids by combining with the constant pH molecular dynamics methods (CPHMD)[95, 63, 96], as

several CPHMD methods are based on λ-dynamics. Furthermore, the GSLD framework presented here is not limited to alchemical free energy calculations. The λ variable could be replaced by the pH values, which would correspond to pH generalized ensemble simulations. In these cases, we can also derive the corresponding RBE similarly.

In this study, we have presented the formalism for the Gibbs sampler based λ-dynamics (GSLD) and the Rao-Blackwell estimator (RBE) for alchemical free energy calculations. These methods were successfully demonstrated for three test cases of increasing complexity. The GSLD, a generalized ensemble sampling method, works for the case where λ is a discrete variable and for the case where λ is considered to be continuous. When λ is continuous, the GSLD can be generalized to calculate free energies for multiple ligands simultaneously in one simulation. The RBE not only eliminates the bias problem of the empirical estimator used in the original λ-dynamics, but also has smaller estimation variance than the empirical estimator. Moreover, we have also shown that the RBE can be used to derive the MBAR/UWHAM equations, which provides new understanding for the MBAR/UWHAM method.[66, 67, 68]

3.7 Auxiliary methods

3.7.1 A Wang-Landau like algorithm to automatically determine the biasing potential G_1^b used in pairwise GSLD when λ is continuous.

The purpose of the biasing potential G_1^b used in the pairwise GSLD when λ is continuous is to make the biased free energy landscape over the λ space flat, i.e. to make the simulation spend about equal time at all λ values between 0 and 1. In current study, a linear biasing potential λG_1^b is utilized, because with the linear biasing potential the biased free energy landscape over λ space is quite flat, i.e. the energy barrier between the two physical states

$\lambda = 0$ and $\lambda = 1$ is small enough that the λ is well sampled across the interval $[0, 1]$. If the linear biasing potential energy cannot make the biased free energy landscape over the λ space flat enough, a quadratic form of biasing potential can be utilized as in Hayes et al.'s flattening method[64]. The biasing potential G_1^b is determined automatically using the following Wang-Landau like algorithm:

- Set the initial biasing potential $G_1^b = 0$ kcal/mol, the decay parameter α such that $0 < \alpha < 1$ ($\alpha = 0.998$ in this study), the biasing potential increment Δ in each step ($\Delta = 2.0$ kcal/mol in this study) and the number of steps R ($R = 3000$ in this study). Initialize the starting state $(\lambda^0, \{x_i^0\}_{i=0}^1, X^0)$.

- For t = 1 to R :

  Sample $(\{x_i^t\}_{i=0}^1, X^t)$ from the conditional distribution: $P(\{x_i^t\}_{i=0}^1, X^t|\lambda^{t-1})$ by running molecular dynamics simulations and then sample λ^t from the conditional distribution $P(\lambda^t|\{x_i^t\}_{i=0}^1, X^t)$. Set $G_1^b(t) = G_1^b(t-1) + (\lambda^t - 0.5) * \Delta(t)$ and $\Delta(t) = \alpha * \Delta(t-1)$.

- The final value of G_1^b from the above step is fixed and used as the biasing potential in following simulations.

3.7.2 Reformulation of the PMF method using conditional probability.

The PMF method requires prior knowledge of which slow degree of freedom is affecting the free energy calculation. In the context of T4 lysozyme, the slow degree of freedom is the side-chain dihedral angle N-CA-CB-CG1 (χ) of residue Val111. The joint distribution of $(\chi, \lambda) : P(\chi, \lambda)$ is of most interest, as it encapsulates all the relevant information required to calculate the free energy $\Delta G = -\beta^{-1}\ln(P(\lambda = l_{16})/P(\lambda = l_1))$. Based on the chain

rule of conditional probability, we have the following equations:

$$P(\chi = \chi^*, \lambda = l_{16}) = P(\chi = \chi^*|\lambda = l_{16})P(\lambda = l_{16}) = P(\lambda = l_{16}|\chi = \chi^*)P(\chi = \chi^*)$$
$$P(\chi = \chi^*, \lambda = l_1) = P(\chi = \chi^*|\lambda = l_1)P(\lambda = l_1) = P(\lambda = l_1|\chi = \chi^*)P(\chi = \chi^*) \tag{3.19}$$

Combining the above two equation gives us:

$$\frac{P(\lambda = l_{16})}{P(\lambda = l_1)} = \frac{P(\lambda = l_{16}|\chi = \chi^*)}{P(\lambda = l_1|\chi = \chi^*)} \cdot \frac{P(\chi = \chi^*|\lambda = l_1)}{P(\chi = \chi^*|\lambda = l_{16})}. \tag{3.20}$$

Therefore, we can calculate the free energy ΔG as

$$\begin{aligned}\Delta G &= -\beta^{-1}\ln\frac{P(\lambda = l_{16})}{P(\lambda = l_1)} \\ &= -\beta^{-1}\ln\frac{P(\lambda = l_{16}|\chi = \chi^*)}{P(\lambda = l_1|\chi = \chi^*)} - \beta^{-1}\ln\frac{P(\chi = \chi^*|\lambda = l_1)}{P(\chi = \chi^*|\lambda = l_{16})} \\ &= \Delta G(\chi = \chi^*) + \big[-\beta^{-1}\ln P(\chi = \chi^*|\lambda = l_1)\big] - \big[-\beta^{-1}\ln P(\chi = \chi^*|\lambda = l_{16})\big],\end{aligned} \tag{3.21}$$

where $\Delta G(\chi = \chi^*)$ is alchemical free energy change when χ is fixed at the value χ^*; $-\beta^{-1}\ln P(\chi = \chi^*|\lambda = l_1)$ is the free energy required to restrict the dihedral angle χ at the value χ^* when T4 lysozyme binds with benzene, i.e, $\lambda = l_1$; $-\beta^{-1}\ln P(\chi = \chi^*|\lambda = l_{16})$ is the corresponding free energy required when T4 lysozyme binds with p-xylene, i.e, $\lambda = l_{16}$. The above equation holds regardless of the value of χ^*.

CHAPTER 4

Protein Engineering

Ding, Xinqiang, Zhengting Zou, and Charles L. Brooks III. "Learning protein stability, evolution and fitness landscapes with variational auto-encoder models." *submitted.*

4.1 Introduction

With the advance of nucleic acid sequencing technology, a large amount of protein sequence data has been accumulated in protein sequence databases such as UniProt [97] and Pfam[98]. For many protein families, many thousands of sequences from different species are available [98]. These naturally occurring diverse protein sequences, belonging to the same protein family but functioning in a diverse set of environments, are the result of mutation and selection occurring in protein evolution. The selection in evolution favors sequences which have high fitness and filters out sequences that do not fold correctly or have low fitness. Therefore, it is expected that the distribution of a protein family's sequences observed in present species carries information about the protein family's properties, such as structure[99, 100, 101, 102, 103], stability [104, 100, 105, 106, 105, 107], evolution [100], and fitness [108, 109, 110]. With large numbers of protein sequences becoming available, several methods have been developed to learn these protein properties using the sequence data [99, 104, 108, 111, 112, 113, 99].

Of particular interest in this paper are methods that are based on learning probabilistic generative models of a protein family's sequence distribution [114]. Biologically, a

protein family is a collection of proteins that share the same evolutionary origin [98]. Protein sequences belonging to the same protein family can vary among species, as observed in multiple sequence alignments (MSAs) of protein families in the Pfam database [98]. From a probabilistic point of view, a protein family represented by sequences containing L amino acids corresponds to a distribution in the protein sequence space: $\{P(S = (s_1, s_2, ..., s_L)) \mid s_j \in \{0, 1, 2, ..., 20\}, j = 1, 2, ..., L\}$, where s_j corresponds to the amino acid type at the jth position of the protein and the amino acid types are labelled using numbers from 0 to 20 with 0 representing a gap. Such a probabilistic generative model assigns a proper probability $P(S = (s_1, s_2, ..., s_L))$ for each protein sequence with L amino acids. Moreover, new sequences can be sampled from the model based on the protein family's sequence distribution. Building a probabilistic generative model of a protein family's sequence distribution is useful in several aspects. For example, for a given protein sequence $S = (s_1, s_2, ..., s_L)$, the probability $P(S = (s_1, s_2, ..., s_L))$ assigned by the model measures how likely the sequence belongs to the protein family, which is useful for searching protein homologies [115]. In addition, new sequences sampled from the model can be used as candidates for protein engineering. Furthermore, the probability function $P(S = (s_1, s_2, ..., s_L))$ of sequences may contain information about dependency between protein positions, which can be utilized to infer protein residue contact maps and epistasis effects between protein positions [99, 113, 112, 100, 108, 111, 113].

Two example methods based on probabilistic generative models are sequence profiles [116, 115] and direct coupling analysis (DCA).[99, 117, 113, 118, 112, 111, 119, 114, 108]. Sequence profiles, widely used for searching homologous sequences, make a strong assumption that amino acid types at different protein positions are independent, i.e., $P(S = (s_1, s_2, ..., s_L)) = \prod_{j=1}^{L} P_j(s_j)$ [116, 115]. Ignoring dependency between positions greatly reduces the number of parameters necessary to model sequence profiles, which makes it feasible to learn a profile even with a limited number of sequences. In contrast, DCA approaches model sequence distributions by taking pairwise dependency between protein

positions into account [99]. Although the number of parameters in DCA is much larger than that in sequence profiles, multiple studies have shown that, for many protein families, sequences available in current databases are sufficient to train DCA models that are useful to predict protein residue contact maps [99, 117, 113, 118, 112, 111, 119, 114] and protein stability change upon mutation [112, 104] .

Although sequence profiles and DCA have proved to be effective at detecting homologous sequences and predicting protein residue contact maps, respectively, they are limited by their inherent assumptions about dependency between protein positions. Sequence profiles do not model any dependency between protein positions and DCA ignores dependency of more than two positions. However, dependency of more than two positions has been observed in real proteins and plays important role in shaping evolutionary trajectories [120, 121, 122]. To overcome these limitations, we propose using variational auto-encoder models [123] for modeling protein family sequence distributions. As a probabilistic generative model, compared with sequence profiles and DCA, variational auto-encoder models do not employ inherent assumptions about dependency between protein positions and can potentially model dependency among any number of positions. In the work presented here, with examples of both natural protein families and simulated sequences, it is shown that variational auto-encoder models are useful for predicting protein stability change upon mutation, capturing evolutionary relationships between sequences, and delineating protein fitness landscapes. Our findings suggest that, with an increasing amount of protein sequence data, variational auto-encoder models will be useful tools for both the study and engineering of proteins.

4.2 Previous Methods

4.2.1 Sequence profiles

Given a protein family's multiple sequence alignment, sequence profiles [115] model its sequence distribution by assuming protein positions are independent, i.e.,

$$P(S = (s_1, s_2, ..., s_L)) = \prod_{j=1}^{L} P_j(s_j), \tag{4.1}$$

where $s_i \in \{0, 1, 2, ..., 20\}$; s_j represents the amino acid type (labelled using numbers from 0 to 20) at the jth position of the protein; $P_j(k)$ represents the probability that the amino acid type at the jth position is k. Therefore, a profile model of a protein family with L amino acids contains $21 \times L$ parameters which are $P_j(k), j = 1, ..., L, k = 0, ..., 20$. These parameters are estimated using the protein family's multiple sequence alignment:

$$P_j(k) = \frac{\sum_{n=1}^{N} w^n * I(s_j^n = k)}{\sum_{n=1}^{N} w^n}, \tag{4.2}$$

where N is the total number of sequences in the MSA; w^n is the weight of the nth sequence; s_j^n is the amino acid type at the jth position in the nth sequence of the MSA; $I(s_j^n = k)$ is equal to 1, if $s_j^n = k$ and 0, otherwise. With the estimated parameters, the profile assigns a probability for any given sequence S with L amino acids based on Eqn. [4.1]. The free energy of the sequence is calculated as $\Delta G_{\text{Profile}}(S) = -\log P(S)$.

4.2.2 Direct coupling analysis

The direct coupling analysis (DCA) method [99, 117, 113, 118, 112, 111, 119] models the probability of each sequence as

$$P(S = (s_1, s_2, ..., s_L)) = \frac{1}{Z} \exp(-\left[\sum_{i=1}^{L-1} \sum_{j=i+1}^{L} J_{ij}(s_i, s_j) + \sum_{i=1}^{L} b_i(s_i)\right]), \tag{4.3}$$

where the partition function Z is

$$Z = \sum_{s_1,s_2,...,s_L} \exp(-\left[\sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}(s_i,s_j) + \sum_{i=1}^{L} b_i(s_i)\right]). \tag{4.4}$$

The parameters in DCA include the bias term $b_i(\cdot)$ for the ith position and the interaction term $J_{ij}(\cdot,\cdot)$ between the ith and the jth position of the protein. Learning these parameters by maximizing likelihood of the model on training data involves calculating the partition function Z, which is computationally expensive. Therefore, the pseudo-likelihood maximization method [117] is used to learn these parameters. Similarly as in sequence profiles, the free energy of a sequence is calculated as

$$\Delta G_{\text{DCA}}(S) = -\log P(S) = \sum_{i=1}^{L-1}\sum_{j=i+1}^{L} J_{ij}(s_i,s_j) + \sum_{i=1}^{L} b_i(s_i) + \log Z. \tag{4.5}$$

Although the partition function Z is not known, we can still calculate the difference of ΔG_{DCA} between two sequences ($\Delta\Delta G_{\text{DCA}}$), because the partition function Z is a constant and does not depend on sequences.

4.2.3 Gaussian process regression

The Gaussian process (GP) regression method [124] is used to fit the fitness (T_{50}) landscape for chimeric cytochrome P450 sequences. To train a GP regression model, a kernel function needs to be chosen to specify the covariance between sequences [124]. When the latent space representation Z is used as the feature vector of sequences, the radial basis function (RBF) kernel [124] is used:

$$K(Z^1, Z^2) = \sigma^2 \exp(-\frac{1}{2}||Z^1 - Z^2||^2), \tag{4.6}$$

where Z^1, Z^2 are latent space representations of two protein sequences and $||\cdot||$ is Euclidean distance in the latent space. When the binary matrix representations X, as in Fig. 1, is used

as the feature vector, the linear kernel is used in GP regression:

$$K(X^1, X^2) = \sigma^2 \sum_{i=1}^{21} \sum_{j=1}^{L} X_{ij}^1 \cdot X_{ij}^2, \tag{4.7}$$

where X^1, X^2 are two $21 \times L$ binary matrices of two protein sequences. The linear kernel function of two sequences is proportional to the sequence identity of the two sequences. The parameter σ^2 in both RBF and linear kernels is estimated by maximizing the likelihood of the GP model on T_{50} training data.

4.3 Variational Auto-Encoder

4.3.1 Learning variational auto-encoder (VAE) models of a protein family's sequence distribution

In VAE models, a protein sequence $S = (s_1, s_2, ..., s_L)$ is represented as a binary $21 \times L$ matrix X for which $X_{ij} = 1$ if $s_j = i$, and $X_{ij} = 0$ otherwise (Fig. 4.1). In addition to the variables X representing sequences, VAE models also include latent space variables Z that can be viewed as a "code" for X. VAE models define the joint distribution of X and Z as $p_\theta(X, Z) = p_\theta(Z)p_\theta(X|Z)$, where θ represents parameters of the joint distribution. The joint distribution $p_\theta(X, Z) = p_\theta(Z)p_\theta(X|Z)$ implies a probabilistic generative process for (X, Z): the latent variables Z are sampled from a prior distribution $p_\theta(Z)$ first and then the sequence variables X are sampled from the conditional distribution $p_\theta(X|Z)$ given Z. The conditional distribution $p_\theta(X|Z)$ acts as a "decoder" that converts "codes" Z into protein sequences X. Although protein sequences X are discrete random variables, the latent space variables Z are modeled as continuous random variables. The prior distribution of Z, $p_\theta(Z)$, is chosen to be an independent multivariable normal distribution with mean of zero. The conditional distribution $p_\theta(X|Z)$ is parameterized using an artificial neuron network with one hidden layer. Given observed sequence data for variables X, learning the

parameters θ that parameterize the generative process is challenging and has been an intensive research topic in machine learning [123]. One reason for the difficulty is that when the conditional distribution $p_\theta(X|Z)$ is complex, such as parameterized by a neuron network, the posterior distribution $p_\theta(Z|X)$ becomes analytically intractable and it is difficult to even draw samples from it efficiently [123]. In this study, given a protein family's multiple sequence alignment, the reparameterization trick, first proposed in VAE models [123], is used to learn the parameters θ that include weight and bias parameters in the decoder neuron network. To remedy the difficulty with the posterior distribution $p_\theta(Z|X)$, in VAE models, a reparameterized "encoder" $q_\phi(Z|X)$ is introduced to approximate the posterior distribution $p_\theta(Z|X)$. In this paper, the encoder $q_\phi(Z|X)$ is also parameterized using an artificial neuron network with one hidden layer (Fig. 4.1).

4.4 Processing sequences in multiple sequence alignments

Before being used as training data for learning VAE models, sequences in multiple sequence alignments are processed to remove positions at which too many sequences have gaps, and sequences with too many gaps. The processing procedure is as the following: (i) positions at which the query sequence has gaps are removed; (ii) sequences with the number of gaps larger than 20% of the total length of the query sequence are removed; (iii) positions at which larger than 20% of sequences have gaps are removed again; (iv) duplicated sequences are removed.

4.5 Variational auto-encoder

VAE models with the reparameterization trick, introduced for learning Bayesian graphical models[123], have been successfully applied for several machine learning problems, such as image and natural language processing [123, 125, 126]. VAE models also have been applied to discover continuous representations of organic molecules [127]. In this paper, a

VAE model similar with that in [123] is employed.

4.5.1 Model setup

The prior distribution of Z, $p_\theta(Z)$, is a m dimensional Gaussian distribution with mean at the origin and variance being the identity matrix. The decoder model $p_\theta(X|Z)$ is parameterized using a fully connected artificial neuron network with one hidden layer as $H = \tanh(W_1 Z + b_1)$ and $p_\theta(X|Z) = \text{softmax}(W_2 H + b_2)$, where the parameters θ include the weights $\{W_1, W_2\}$ and the biases $\{b_1, b_2\}$. The encoder model $q_\phi(Z|X)$ is chosen to be a m dimensional Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where Σ is a diagonal matrix with diagonal elements of $\sigma^2 = (\sigma_1^2, \sigma_2^2, ..., \sigma_m^2)$. The mean μ and the variance σ^2 are parameterized using an artificial neuron network with one hidden layer as $H = \tanh(W_3 X + b_3)$, $\mu = W_4 H + b_4$, $\log \sigma^2 = W_5 H + b_5$. The parameters ϕ for the encoder model $q_\phi(Z|X)$ include weights $\{W_3, W_4, W_5\}$ and biases $\{b_3, b_4, b_5\}$.

4.5.2 Model training

The weights of sequences in a protein multiple sequence alignment are calculated using the position-based sequence weights.[128] Given weighted protein sequences, VAE models learn the parameters of both encoder and decoder models simultaneously by optimizing the evidence lower bound objective function (ELBO) [123] which is defined as

$$\text{ELBO}(\theta, \phi) = \sum_Z q_\phi(Z|X) \log p_\theta(X|Z) + \sum_Z q_\phi(Z|X) \log \frac{p_\theta(Z)}{q_\phi(Z|X)}.$$

To reduce overfitting, a regularization term of $\gamma \cdot \sum_{i=1}^{5} \parallel W_i \parallel_F^2$ is added to the objective ELBO(θ, ϕ), where γ is called the weight decay factor and $\parallel W_i \parallel_F$ is the Frobenius norm of weight matrix W_i. The gradient of ELBO plus the regularization term with respect to the model parameters is calculated using the backpropagation algorithm [129] and the parameters are optimized using the Adam optimizer [130]. The weight decay factor γ is

selected from the set of values $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$ using 5-fold cross validation (using 10-fold cross validation in the case of cytochrome P450s). In the cross validation, models trained with different weight decay factors are evaluated based on the marginal probability assigned by the model on the held-out sequences (based on the Pearson correlation coefficient in the case of cytochrome P450s).

4.5.3 Calculating the marginal probability of a sequence X, $p_\theta(X)$

Given a sequence X, the marginal probability, $p_\theta(X)$, is equal to the integral $\int p_\theta(X, Z)\mathrm{d}Z$ which is calculated using importance sampling:

$$\begin{aligned} p_\theta(X) &= \int p_\theta(X, Z)\,\mathrm{d}Z = \int q_\phi(Z|X)\frac{p_\theta(X, Z)}{q_\phi(Z|X)}\,\mathrm{d}Z \\ &= \mathop{\mathbb{E}}_{Z\sim q_\phi(Z|X)}\left[\frac{p_\theta(X, Z)}{q_\phi(Z|X)}\right] = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{p_\theta(X, Z^i)}{q_\phi(Z^i|X)}\right], \end{aligned}$$

where Z^i are independent samples from the distribution $q_\phi(Z|X)$, and N is number of samples. In this study, $N = 1 \times 10^6$.

4.6 Simulating multiple sequence alignments

A random phylogenetic tree with 10,000 leaf nodes was generated using the *populate* function of the master Tree class from ETE Toolkit [131]. The random branch range is chosen to be from 0 to 0.3. The LG evolutionary model [132] was used to simulate the sequence evolution on the generated phylogenetic tree. Sequences from leaf nodes were combined into a multiple sequence alignment.

4.7 A predefined protein fitness function

In the predefined protein fitness function, the parameters, $B_i(\cdot)$, represent the contribution of individual positions and their values are specified by sampling from the normal distribution $\mathcal{N}(0, 2^2)$. The parameters, $J_{ij}(\cdot, \cdot)$, corresponds to the epistatic effect between position i and position j. These are chosen with a probability of 0.95 that there is no epistatic effects between positions i and j, i.e., $J_{ij}(\cdot, \cdot) = 0$. Otherwise, the values of $J_{ij}(\cdot, \cdot)$ are chosen from the normal distribution $\mathcal{N}(0, 1)$. Similarly, the parameters, $J_{ijk}(\cdot, \cdot, \cdot)$, stand for epistasis among positions i, j, and k. These are chosen with a probability of 0.998 that there is no epistatic effect between positions, i.e., $J_{ijk}(\cdot, \cdot, \cdot) = 0$. Otherwise, their values are chosen from the normal distribution $\mathcal{N}(0, 0.5^2)$.

4.8 Results and Discussion

4.8.1 Predicting protein stability change upon mutations

With a protein family's multiple sequence alignment as training data, VAE models learn the joint distribution of latent space variables Z and sequence variables X: $p_\theta(X, Z)$. After learning a VAE model, a marginal probability $p_\theta(X)$ can be calculated for each sequence X with L amino acids as $p_\theta(X) = \int p_\theta(X, Z)\mathrm{d}Z$. The marginal probability of a sequence X, $p_\theta(X)$, measures how likely it is that the given sequence X belongs to the protein family, i.e., how similar the given sequence is to the sequences from the protein family's MSA. Because the protein family's MSA are results of selection in protein evolution, sequences with higher probability belonging to the protein family's MSA are expected to have better adaptation under selection pressures. Selection pressures for protein evolution may include stability, enzyme activity, drug resistance, or other properties. It can also be a mixture of different selection pressures. Although different protein families might be under different sets of selection pressures in evolution, a common selection pressure shared by many

structured protein families is protein stability.

To test if a protein sequence's probability assigned by VAE models correlates with the sequence's stability, we applied VAE models for three protein families: fibronectin type III domain (Pfam accession id: PF00041), staphylococcal nuclease (PF00565), and phage lysozyme (PF00959). These three protein families were selected because there are both experimental data on stability change upon mutation [134] and a large number of sequences in the Pfam database[98] for the three protein families. After processing, the number of unique sequences in their MSAs is 46498, 7649, and 3560 for fibronectin type III domain, staphylococcal nuclease, and phage lysozyme, respectively. A VAE model was trained with these unique sequences for each family and was used to calculate marginal probabilities of sequences that have experimental folding free energies. To be comparable with experimental folding free energies, probabilities of sequences, $p_\theta(X)$, are transformed into unitless "free energies" by $\Delta G_{\text{VAE}}(X) = -\log p_\theta(X)$.

For protein stability change upon single site mutations, the predicted results using VAE models are compared with experimental results for the three protein families (Fig. 4.2A). The Pearson's correlation coefficients between the experimental and predicted results for fibronectin type III domain, saphylococcal nuclease, and phage lysozyme are 0.81, 0.52, and 0.43, respectively. The corresponding Spearman's rank correlation coefficients are 0.85, 0.50, and 0.42, respectively. Protein families with more unique sequences in their MSAs used as training data tend to have higher correlation coefficients. For example, fibronectin type III domain, with the largest number of unique sequences in its MSA, has the highest correlation coefficient among the three protein families. Therefore, the limited number of unique sequences in their MSAs might be one of the reasons why staphylococcal nuclease and phage lysozyme have more modest correlation coefficients. The stability change upon single site mutations is also predicted using sequence profiles and DCA. The results from both methods are compared with those from the VAE models in terms of Spearman's rank correlation coefficients (Fig. 4.2A). The performance of the VAE models

is comparable with that of sequence profiles and DCA. VAE models are slightly better than the other two methods for the fibronectin type III domain and staphylococcal nuclease, which have a relatively larger number of sequences (Fig. 4.2A).

The effects of double and triple site mutations on phage lysozyme's stability are also predicted using all three methods. The predicted results are compared with experimental results [133] in Fig. 4.2B. Because sequence profiles assume that protein positions are independent and ignore epistasis between positions, its prediction on the effects of multiple mutations on stability is much poorer than the other two models, both of which take the dependency between positions into account.

In summary, VAE models are useful for predicting protein stability change as a result of mutation. For predicting the effect of single site mutations on protein stability, the VAE model's performance is comparable with sequence profiles and DCA and becomes better than the other two methods when a large number of sequences are available. Like DCA, VAE models also capture the pairwise dependency between positions, which enables DCA and VAE models to outperform the sequence profile method in predicting the effect of double and triple site mutations on protein stability. Moreover, VAE models should also be able to capture dependency among more than two protein positions, which is not modeled in DCA.

4.8.2 VAE latent space representation captures phylogenetic relationships between sequences

After training with a protein family's MSA, the VAE encoder, $q_\phi(Z|X)$, can be used to embed sequences in a low dimensional continuous latent space, Z. Embedding sequences in a low dimensional continuous space can be useful for several reasons. The low (2 or 3) dimensionality makes it easier to visualize sequence distributions and sequence relationships. The continuity of the space enables us to apply operations to the family of sequences, such as interpolation and extrapolation, that are best suited to continuous variables.

For visualization purposes, the latent space used in this section is 2-dimensional. For the three protein families: fibronectin type III domain, staphylococcal nuclease, and phage lysozyme, the latent space embedding of all the sequences from their MSAs is shown in Fig. S1(A-C). These embedding results show that, in the latent space, sequences are not distributed randomly. Their distributions have a star structure with multiple spikes, each of which points from the center towards the outside along a specific direction. The star structure resembles phylogenetic tree structures that represent phylogenetic relationships between sequences. To test if the latent space representation can capture phylogenetic relationships between sequences like phylogenetic trees, we applied VAE models on a simulated protein family MSA. The simulated MSA is generated by neutrally evolving a random protein sequence with 100 amino acids on a simulated phylogenetic tree [131] with 10,000 leaf nodes and combining sequences from all the leaf nodes (Fig. 4.3A). Thus, the phylogenetic relationships between sequences in this simulated MSA are known based on the phylogenetic tree used for simulation.

As with the three protein families shown above, the latent space representation of the simulated sequences has a similar star structure with multiple separate spikes (Fig. 4.3B), even though the sequence evolves neutrally in the simulation. As a negative control, a VAE model is also trained on an MSA consisting of random sequences sampled from the equilibrium distribution of the LG evolutionary model [132]. The star structure is not observed in the latent space representation of these random sequences (Fig. S1D), which strongly supports the idea that the star structure is derived from the evolutionary relationships encoded in the tree structure used in the simulation. To compare the latent space star structure with the phylogenetic tree, sequences are grouped together if they share the same ancestor at a reference evolutionary time point based on the phylogenetic tree. Sequences in the same group have the same color in their latent space representation (Fig. 4.3B). Sequences with the same color, i.e., sharing the same ancestor at the chosen time point, are observed to have their latent space representations in the same spike or multiple adjacent

spikes (Fig. 4.3B). The multiple adjacent spikes occupied by the same group of sequences represent more fine-grained phylogenetic relationships between sequences and these more fine-grained phylogenetic relationships can be recovered by changing the reference time point used to group the sequences (Fig. 4.3C). Therefore, the spatial organization of the latent space representation of the sequences captures features of the phylogenetic relationship between sequences.

Another similarity between the star structure in the latent space and the phylogenetic tree is that the phylogenetic tree originates from the root node and spikes in the star structure originate from the origin of the latent space (Fig. 4.3B). This similarity is supported by the observation that the latent space representation of the root node sequence is near the origin of the latent space (Fig. 4.3D). Furthermore, to see how a sequence's latent space representation moves in the latent space as the sequence evolves, both leaf node sequences and their corresponding ancestral sequences are projected into the latent space. For a leaf node sequence and its corresponding ancestral sequences, the primary moving direction is calculated as the first component direction using principal component analysis (Fig. 4.3D). It is shown that a sequence's distance from the origin along the moving direction in the latent space is highly correlated with the sequence's evolutionary distance from the root node sequence (Fig. 4.3D and E). This correlation suggests that as sequences evolve from the root node towards leaf nodes in the phylogenetic tree, their latent space representations move from the origin of the latent space towards the outside along specific directions (Fig. 4.3D). This pattern holds for most of the leaf node sequences and their corresponding ancestral sequences (Fig. 4.3F).

The comparison between the phylogenetic tree structure and the latent space representation of sequences demonstrates that the VAE latent space representation can capture similar phylogenetic relationships between sequences as does the phylogenetic tree. Phylogenetically close sequences are clustered spatially together as spikes in the latent space. In addition, as a sequence evolves, its latent space representation moves from the origin

towards the outside along a spike. These phylogenetic relationships captured in the VAE's latent space representation make the VAE a potentially useful tool for studying protein evolution. Compared with traditional phylogenetic trees, VAE models does not require choosing a specific evolutionary model. Moreover, VAE models can work with a much larger number of sequences (hundreds of thousands of sequences or more) than a phylogenetic tree, because it does not require the tree structure search or pairwise sequence comparison. One disadvantage of the VAE model is that it may not be able to capture as many details of the evolutionary relationships as does the phylogenetic tree. Therefore, a mixture model of both phylogenetic trees and VAE models might have the best of both approaches for studying protein evolution.

4.8.3 Navigating the protein fitness landscape in the VAE latent space

A protein's fitness landscape is a map from the protein's sequence to the protein's fitness, such as the protein's stability and activity, among a host of other properties. Knowing a protein's fitness landscape can greatly assist in studying and engineering proteins with altered properties. A protein's fitness landscape can also be viewed as a fitness function in a high dimensional discrete space of sequences. Because of the high dimensionality and discreteness of this sequence space and the effects of epistasis between different protein positions, it has been difficult for protein researchers to characterize protein fitness landscapes. As only a relatively small number of sequences can be synthesized and have experimentally measured fitness values, a common problem facing researchers is, given the fitness values for a collection of sequences from a protein family, how does one predict the fitness value of a new sequence from the same protein family, or design a new sequence which will have a desired fitness value.

Here we propose a semi-supervised learning framework utilizing the VAE latent space representation to learn protein fitness landscapes using both protein sequence data and experimental fitness data (Fig. 4.4D). Although fitness values are usually known for only

a small subset of sequences from a protein family, we often have access to a large number of homologous sequences from the same protein family. These sequences represent functional proteins from species living in different environments. The distribution of these sequences is shaped by evolutionary selection. Therefore, we expect that the distribution of these sequences contains information about the relationship between sequence and fitness. To utilize this information, with a large number of sequences from a protein family, we can model the distribution of sequences by learning a VAE model for the protein family. The resulting VAE model provides us with a sequence encoder and a sequence decoder. With the sequence encoder, sequences are first embedded into a low dimensional continuous latent space. Then the fitness landscape is estimated in the latent space with experimental fitness data. With an estimated fitness landscape in the latent space, we can predict the fitness value of a new sequence using its latent space representation. In addition, we can also design new sequences with desired fitness values by choosing points in the latent space based on the fitness landscape and converting these points into sequences using the VAE decoder (Fig. 4.4D). To test this framework, we applied it to two protein families: a simulated protein family with a predefined fitness function and the cytochrome P450s [135, 136, 137].

4.8.4 A simulated protein family with a predefined fitness function

An ideal case to test the above framework would be a protein family whose fitness function is known. For natural protein families, fitness values are known for only a small number of sequences. Therefore, we first applied the framework to a simulated protein family for which a fitness function is predefined as:

$$\text{Fitness}(s_1, s_2, ..., s_L) = \sum_{i=1}^{L} B_i(s_i) + \sum_{1 \le i < j \le L} J_{ij}(s_i, s_j) + \sum_{1 \le i < j < k \le L} J_{ijk}(s_i, s_j, s_k),$$

where s_i is the amino acid type at position i. This fitness function not only includes the effect of amino acid types at individual positions ($B_i(s_i)$), but also includes the effects of second order ($J_{ij}(s_i, s_j)$) and third order ($J_{ijk}(s_i, s_j, s_k)$) epistasis. The parameters of the fitness function, B_i, J_{ij}, and J_{ijk}, are specified using the procedure described in **Methods**.

The setup used for simulating sequences is the same as in Fig. 4.3 except that as the sequence evolves along the path from the root node to a randomly chosen leaf node A, its fitness value has to increase monotonically based on the predefined fitness function (Fig. 4.4A). Mutations that decrease the fitness value are rejected. The simulated MSA is used to train a VAE model with a two dimensional latent space and the sequences corresponding to nodes on the path under selection are projected into the latent space using the VAE encoder (Fig. 4.4B). Similar to the pattern observed in Fig. 4.3D, these sequences align along a preferred direction in the latent space (Fig. 4.4B) because of their ancestral relationship. As the sequence evolves from the root node to the leaf node A, its latent space representation moves away from the origin along a direction which is obtained using the lowest frequency eigenvector from a principal component analysis of the latent space representation of these sequences. Because the fitness value increases monotonically as the sequence evolves along the path, the sequences' fitness values correlate with their positions in latent space along the principal component eigenvector direction (orange points in Fig. 4.4C). This correlation can be viewed as the fitness landscape along the eigenvector direction, but it is observed only at a finite number of discrete points. Does this correlation hold continuously along this direction? To answer this question, 300 points, uniformly distributed along the eigenvector direction, were converted into protein sequences using the VAE decoder and their fitness values are calculated with the predefined fitness function (blue points in Fig. 4.4C). For these decoded sequences, fitness values also correlate with their positions along the eigenvector direction in the latent space (Fig. 4.4C). Because the correlation holds continuously, it is useful to not only predict fitness of sequences whose latent space representation lies along this eigenvector, but also to design sequences

that have fitness values in between by interpolating sequences through their latent space representation.

4.8.5 Cytochrome P450

The cytochrome P450 protein family was chosen to test our framework because there are both experimental fitness data and a large number of sequences available for the protein family. The Arnold group made a library of 6561 chimeric cytochrome P450 sequences by recombining three cytochrome P450s (CYP102A1, CYP102A2, CYP102A3) at seven crossover locations [135] (Fig. S2) and measured T_{50} values (the temperature at which 50% of the protein is inactivated irreversibly after 10 minutes) for 278 sequences [135, 136, 137]. In addition to these experimental T_{50} fitness data, the cytochrome P450 family (PF00067) has more than 28K unique homologous sequences in its MSA from the Pfam database [98].

For visualization purposes, we first trained a VAE model with a two dimensional latent space. Embedding the 28K sequences from its MSA (Fig. S3A) shows that the latent space representation of these sequences has a similar star structure as observed in Fig. 4.3B. Comparing the latent space representation of sequences from the MSA (Fig. S3A) with that of chimeric sequences (Fig. S3B), we can see that the 6561 chimeric sequences, made by all possible recombinations of three proteins at seven crossover locations, only occupy a small fraction of latent space available for the protein family. This suggests that most of the sequence space of cytochrome P450 is not covered by these chimeric sequences. Therefore, the two dimensional latent space representation, though simple, is useful to estimate how much sequence space has been covered by a set of sequences. In addition, it can also potentially guide designing sequences from the unexplored sequence space by converting points in the unexplored latent space region into sequences using the VAE decoder.

Embedding the sequences which have T_{50} data into the two dimensional latent space and coloring the sequences based on their fitness values provide a way to visualize the fitness landscape (Fig. S3C). As the fitness landscape is not necessarily linear, Gaussian

processes are used to fit a continuous fitness surface using the two dimensional latent space representation as features and using the radial basis function (RBF) kernel with Euclidean distance. The 278 sequences with T_{50} experimental data are randomly separated into a training set of 222 sequences and a testing set of 56 sequences. Based on 10-fold cross validation on the training set, just using the two dimensional latent space representation of sequences which have 466 amino acids, the Gaussian process model can predict the T_{50} values for the training set with a Pearson correlation coefficient of 0.80 ± 0.06 and a MAD (mean absolute deviation) of 3.2 ± 0.4°C (Fig. S3D). For the testing set, the Pearson correlation coefficient is 0.84 and the MAD is 2.9°C.

As the method is not restricted to two dimensional latent spaces, VAE models with latent spaces of different dimensionality combined with Gaussian processes may also used to predict the T_{50} experimental data. Based on 10-fold cross validation Pearson correlation coefficients, the VAE model with a 30 dimensional latent space works the best with a Pearson correlation coefficient of 0.93 ± 0.02 and a MAD of 1.9 ± 0.2°C on the training set (Fig. 4.4E). On the testing set, the Pearson correlation coefficient is 0.93 and the MAD is 2.0°C.

We note that Gaussian processes have been used before to learn the T_{50} fitness landscape of cytochrome P450 either employing sequences as features with a structure based kernel function [136] or using embedding representations [138]. Compared with previous methods [136, 138], one difference of our method lies in the embedding method. The embedding method used in this study is the VAE encoder learned by modeling the sequence distribution of the protein family. Therefore, it utilizes information specific to the protein family. In contrast, the embedding method proposed in [138] is a generic *doc2vec* embedding method, which is learned by pooling sequences from many protein families together and viewing all protein sequences equally. Another difference with our method is that points in the embedding space, i.e., the latent space, can be converted into sequences using the VAE decoder. Therefore, the transformation between sequence space and embedding

space is a two-way transformation, instead of one way as in [138]. This enables our approach to be used to propose new sequences based on the fitness landscape in the latent space.

4.9 Conclusion

Using both simulated and experimental data, we have demonstrated that VAE models, trained only with MSAs of protein families, can predict protein stability change upon mutation and learn phylogenetic relationships between sequences. Unlike the sequence profile method and the DCA method, VAE models can potentially model amino acid dependency among any number of protein positions. Compared with phylogenetic trees, to learn phylogenetic relationships between sequences, VAE models do not assume a predefined evolutionary model and can work with a much larger number of sequences. When experimental data on protein fitness is available for a subset of sequences, VAE models can also help learn fitness landscapes with the low dimensional continuous latent space representation of sequences. With an estimated fitness landscape in the latent space and the two-way transformation between the latent space and the sequence space, the VAE models can not only predict fitness values of sequences, but also help design new candidate sequences with desired fitness for experimental synthesis and validation. With the advance of sequencing technology, the amount of protein sequence data that are available to train VAE models increases rapidly. Moreover, recent deep mutational scanning experiments are generating large-scale data sets of the relationship between protein sequences and function [139]. With this increasing amount of both protein sequence and fitness data, the VAE model will be a useful tool to learn information about protein stability, evolution, and fitness landscapes and provide insights into the engineering of proteins with modified properties.

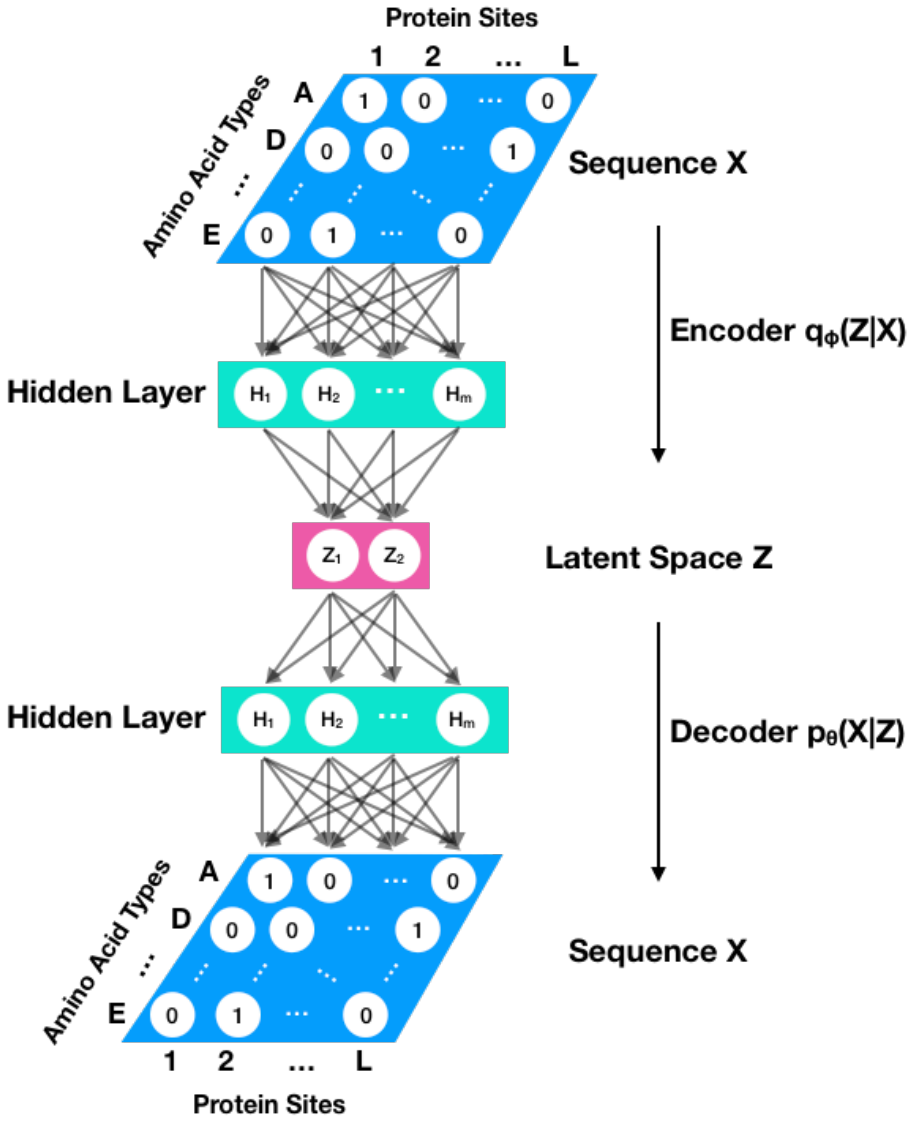

Figure 4.1: **Encoder and decoder models used in the variational auto-encoder.** Both encoder and decoder models used in this paper are fully connected artificial neuron networks with one hidden layer H. The encoder model transforms each protein sequence X into a distribution $q_\phi(Z|X)$ of Z in the latent space; the decoder model transforms each point in the latent space Z into a distribution $p_\theta(X|Z)$ of X in the protein sequence space. In both models, protein sequences from a multiple sequence alignment with L amino acids are represented as a $21 \times L$ matrix whose entries are either 0 or 1 based on a one-hot coding scheme. Gaps in sequences are modeled as an extra amino acid type.

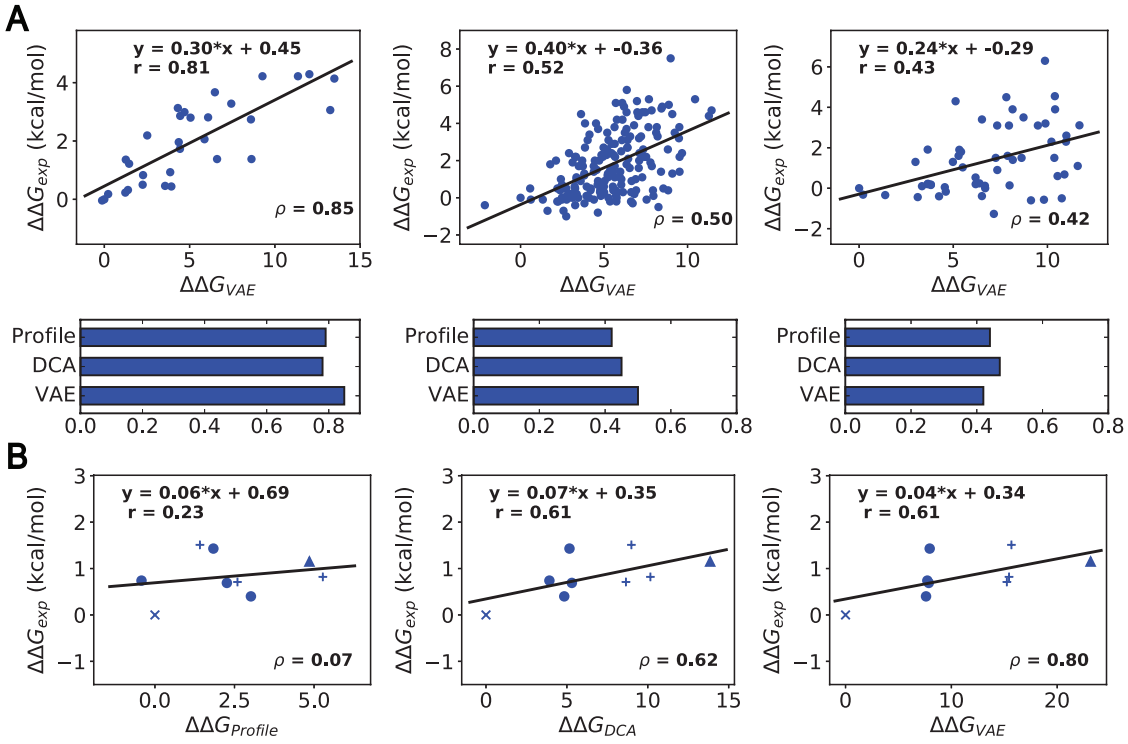

Figure 4.2: **Predicting protein stability change upon mutations. (A) (Top)** Correlation between experimental results and predicted results using VAE models on protein stability change upon single mutations for fibronectin type III domain (left), staphylococcal nuclease (middle), and phage lysozyme (right). $\Delta\Delta G_{\text{exp}}$ is experimental protein folding free energy change upon single mutations compared with the wild type protein. $\Delta\Delta G_{\text{VAE}}$ is predicted protein stability change upon single mutations using VAE. $\Delta\Delta G_{\text{VAE}}$ is calculated as the change of negative log-likelihood of sequences when single mutations are introduced. Therefore, $\Delta\Delta G_{\text{VAE}}$ is an unitless quantity. Each point corresponds to a mutant sequence with one mutation compared with the wild type sequence. r and ρ are Pearson's correlation coefficients and Spearman's rank correlation coefficients, respectively. **(Bottom)** In addition to VAE models, protein stability change upon single mutations are also predicted using sequence profiles and DCA. Spearman's rank correlation coefficients between experimental results and predicted results using the three methods are compared for the same three protein families. **(B)** Correlation between experimental results and predicted results on protein stability change upon single ($\circ$), double (+) and triple ($\triangle$) mutations for phage lysozyme using profiles (left), DCA (middle), and VAE (right) models. The estimated measurement error in $\Delta\Delta G_{\text{exp}}$ is ± 0.2 kcal/mol [133]. We note that the correlations shown here are results on testing sets because the experimental folding free energy changes are not used in training the VAE model.

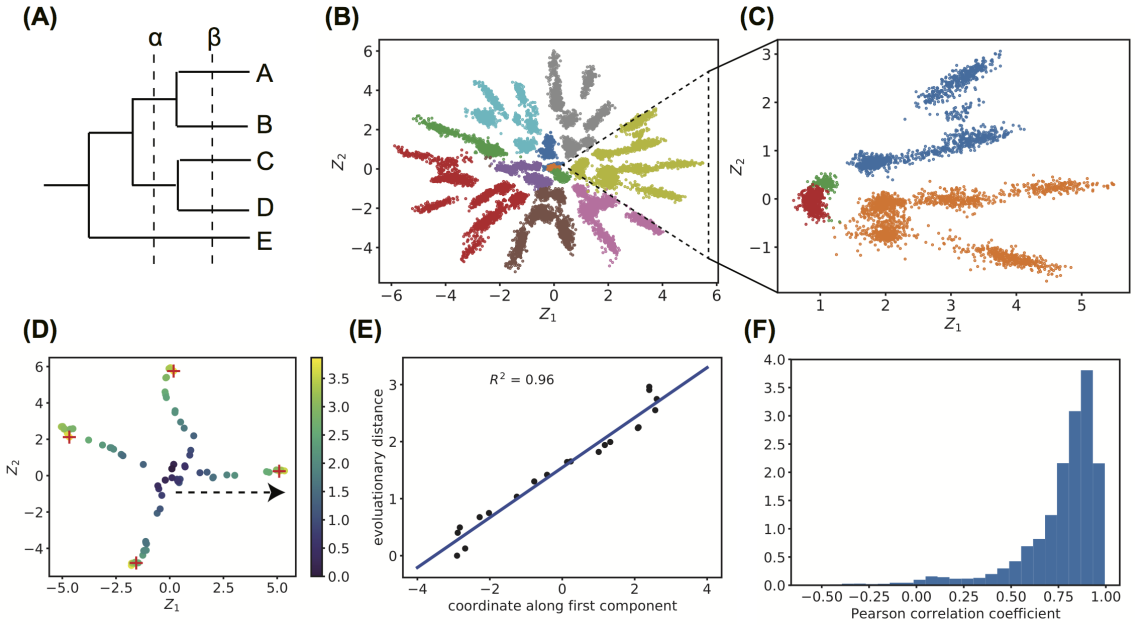

Figure 4.3: **VAE latent space representation of sequences captures phylogenetic relationships between sequences.** **(A)** A schematic representation of the phylogenetic tree used for simulating evolution of a random protein sequence with 100 amino acids. The actual tree used has 10,000 leaf nodes. Dash lines, α and β, represent two reference evolutionary time points on which sequences of leaf nodes are grouped. Sequences of leaf nodes are in the same group if they share the same ancestor at the reference time point, either α or β. **(B)** VAE latent space representation of sequences of all leaf nodes. The sequence of each leaf node is projected into the 2-dimensional latent space onto the point $\mathbb{E}_{q_\phi(Z|X)}Z$, where $Z = (Z_1, Z_2)$ based on the VAE encoder $q_\phi(Z|X)$. Sequences are separated into groups at the reference time point α, which has an evolutionary distance of 0.5 from the root node. Sequences in the same group have the same color. **(C)** Sequences from the yellow colored group in **(B)** are regrouped and recolored based on the reference time point β, which has an evolutionary distance of 0.92 from the node. **(D)** VAE latent space representation of four representative leaf node sequences, labelled as plus signs, and their ancestral sequences, labelled as dots. Sequences are colored based on their evolutionary distances from the root node. The sequence of the root node sits around the origin in the latent space. As the sequence evolves from the root node to a leaf node, its latent space representation moves from the origin towards the surroundings along a direction. The moving direction, labelled as a dashed arrow line for the right most leaf node, is calculated as the first component direction using the principal component analysis. **(E)** For the leaf node sequence at the rightmost of **(D)** and its corresponding ancestry sequences, their coordinates along the moving direction correlates with their evolutionary distances from the root node. **(F)** The distribution of Pearson's correlation coefficients of all leaf node sequences, as calculated in **(E)**.

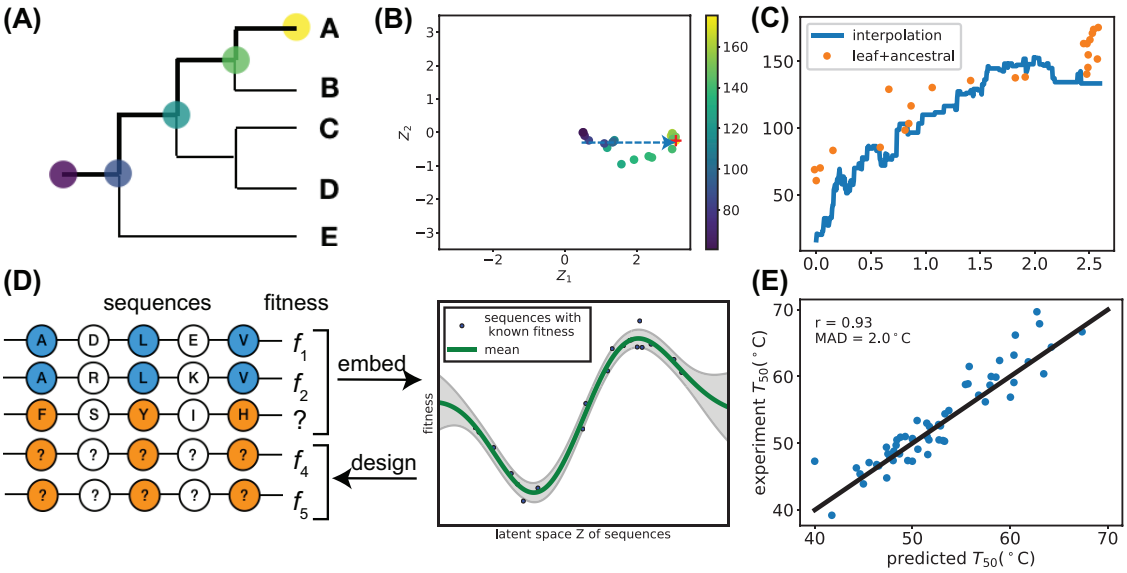

Figure 4.4: **Navigating the protein fitness landscape in the VAE latent space.** **(A)** A schematic representation of the phylogenetic tree used for simulating evolution of a random protein sequence with 100 amino acids. The simulation setup is the same as that in Fig. 4.3A except that a selection pressure with a predefined fitness function is applied through the path (bold) from the root node to a leaf node A. Therefore, fitness of sequences increases monotonically along the path. **(B)** Latent space representation of sequences corresponding to the nodes along the bold path in **(A)**. Color represents fitness values of sequences. Red plus sign represents the position of the leaf node sequence. Dashed arrow line represents the primary moving direction, which is used in **(C)**. **(C)** (Orange) Fitness of sequences from both the leaf node and the ancestral nodes along the path under selection. (Blue) Fitness of interpolated sequences which are calculated by decoding points along the primary moving direction in the latent space into sequences. **(D)** The proposed framework on how VAE latent space representation of sequences can be combined with other methods, such as Gaussian processes in this study, to predict fitness of a new sequence and to design a new sequence with specified fitness values. **(E)** Correlation between predicted T_{50} and experimental T_{50} for P450 chimera sequences in testing set.

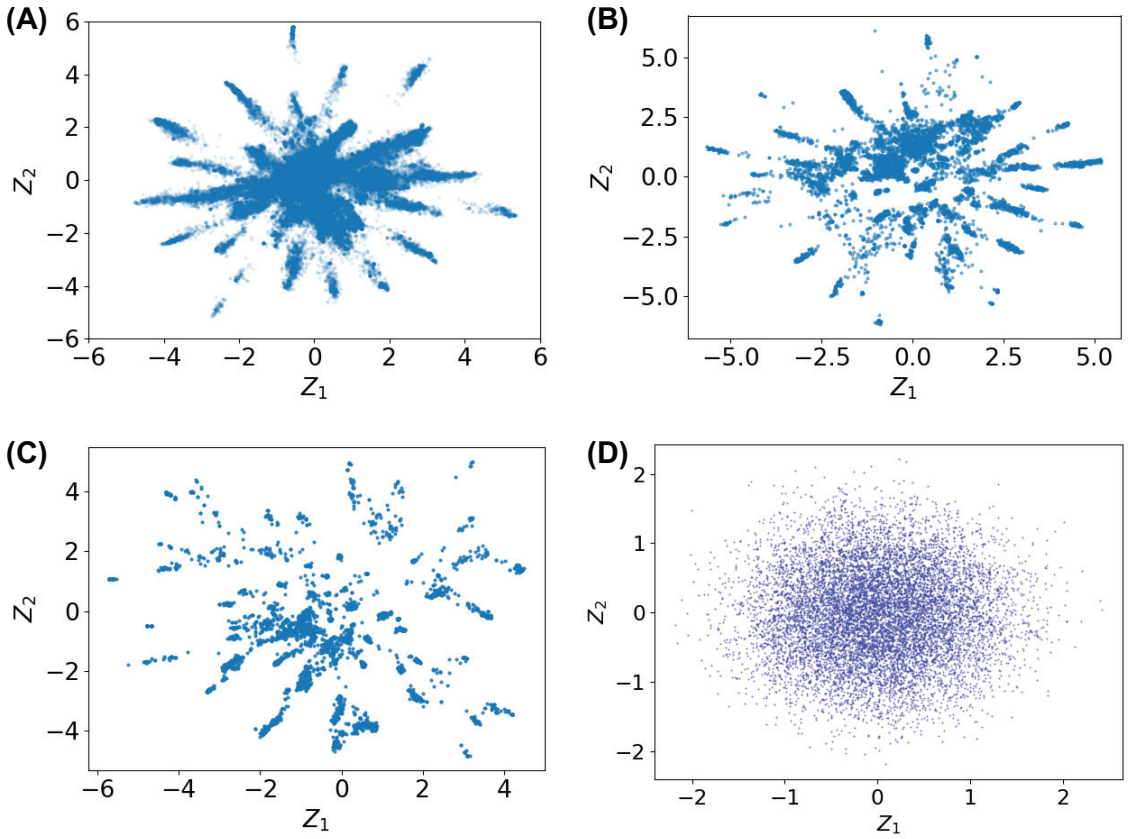

Figure 4.5: Two dimensional latent space representations of sequences from multiple sequence alignments for protein families: fibronectin type III domain **(A)**, staphylococcal nuclease **(B)**, and phage lysozyme **(C)**. A two dimensional latent space representation of random sequences with 100 amino acids sampled from the equilibrium distributions of the LG evolutionary model **(D)**.

Figure 4.6: Sequences of the three parent cytochrome P450s (CYP102A1, CYP102A2, CYP102A3). The chimeric sequences are made by recombining the three proteins at the seven cross over locations marked by arrows [135].

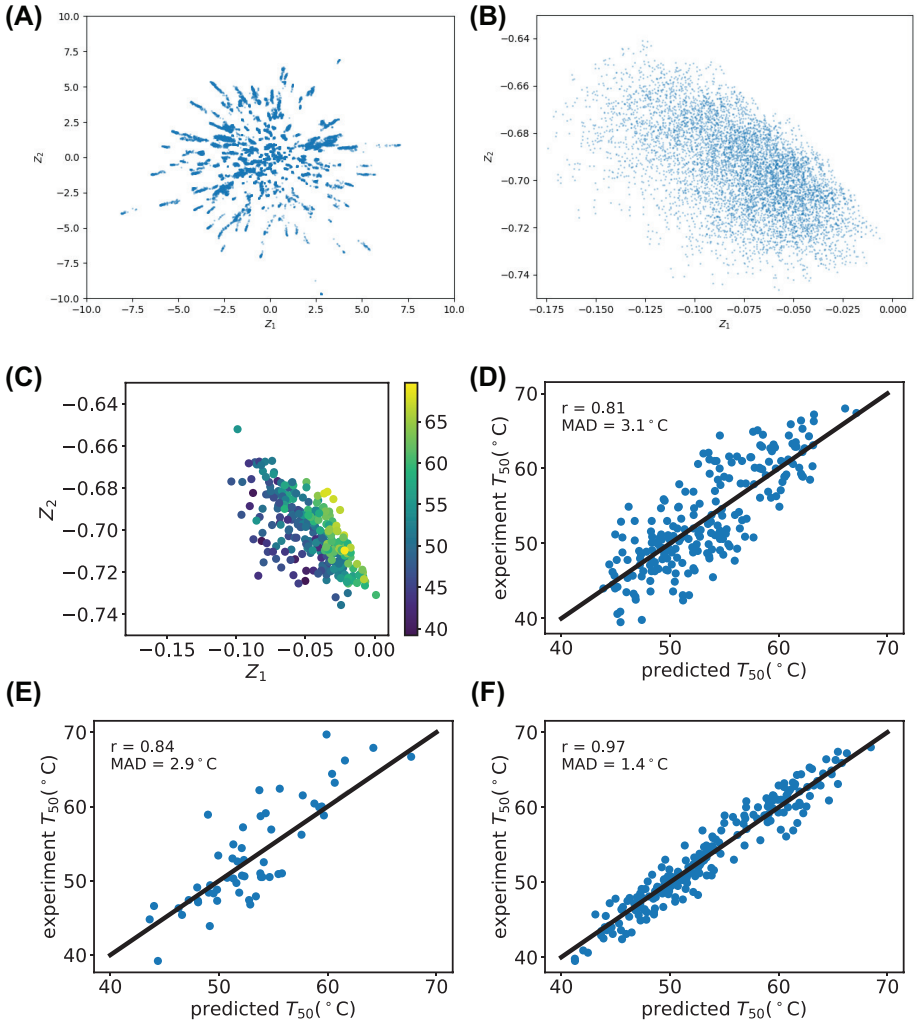

Figure 4.7: Latent space representations of sequences for cytochrome P450 family and its fitness landscape. **(A)** A two dimensional latent space representation of sequences for cytochrome P450 family (PF00067). **(B)** The two dimensional latent space representation of 6561 chimeric cytochrome P450 sequences made by combining the three cytochromes P450 (CYP102A1, CYP102A2, CYP102A3) at seven crossover locations. **(C)** The two dimensional latent space representation of 278 chimeric cytochrome P450 sequences whose T_{50} values are measured experimentally by the Arnold group. Each point represents a chimeric cytochrome P450 sequence. Points are colored by their experimental T_{50} values. **(D)** The Gaussian process's performance at predicting T_{50} on the training set of 222 chimeric cytochrome P450 sequences using the two dimensional latent space representation (Z_1, Z_2) as features and using the radial basis function kernel with Euclidean distance in latent space Z. **(E)** The performance of the Gaussian process model from **(D)** at predicting T_{50} on the testing set of 56 chimeric cytochrome P450 sequences. **(F)** The Gaussian process's performance at predicting T_{50} on the training set of 222 chimeric cytochrome P450 sequences using the 30 dimensional latent space representation $(Z_1, ..., Z_{30})$ as features and using the radial basis function kernel with Euclidean distance in latent space Z.

CHAPTER 5

Discussion and Conclusions

In the previous chapters three methodological advances have been developed and presented for both drug discovery and protein engineering. Although these advances do solve several problems in drug discovery and protein engineering, they are still far from perfect and can be further improved. Moreover, in addition to the applications shown in the dissertation, these advances also have potential to be used in other applications. Therefore, the first purpose of this chapter is to share with the reader how these advances can be further improved and used in other applications.

The significant speedup achieved by the FFT docking on GPUs for searching ligands' translational and rotational space can be used to rank ligand docking conformations in protein-ligand docking. In most current protein-ligand docking programs, ligand conformations are ranked by their interaction energies with proteins, which does not take the entropic effects into account explicitly[13, 14, 12, 7, 9]. With the FFT docking, a Boltzmann weighted energy can be calculated by calculating interaction energies of a ligand conformation with proteins for all of its positions and orientations and taking a Boltzmann average of these energies. This explicit way of incorporating the entropic effects of ligand translations and rotations into ranking ligand docking conformations has the potential to improve the ranking accuracy of the initial scoring function. The current implementation of parallel MD simulated annealing running on a GPU is only 20 times faster than the original simulated annealing running on a CPU when 500 trials of simulated anneal-

ing are used. One of the reasons that the speedup is only 20 is that the implementation is using OpenMM[36] and a large fraction of computing time is spent on constructing the OpenMM context. Therefore, implementing the parallel MD simulated annealing directly using CUDA might be able to accelerate the calculation much further beyond a speedup of 20 times.

The Gibbs sampler λ-dynamics (GSLD) approach for free energy calculation is only applied to small test systems in the dissertation[8]. Like λ-dynamics, GSLD can also be extended to calculate free energies for a large number of states simultaneously. One particular suitable application of this extension will be the use of GSLD for constant pH molecular dynamics, which can provide information on how pH affects the dynamics of a biological system. In the dissertation, the Rao-Blackwell estimator is shown to not only address the biasing problem of the empirical estimator but also provide a new understanding of the multistate Bennett acceptance ratio (MBAR) equations[66]. This new way of understanding the MBAR equations has inspired us to develop a fast solver for solving large scale MBAR equations and a manuscript describing this new fast solver is in preparation.

The variational auto-encoder (VAE) approach [123] for learning protein stability, evolution, and fitness landscape information from protein sequences is still in an early stage and several questions are worth further investigation. For instance, considering that the VAE approach can work with a large number of sequences, an effective way to combine it with tools from evolutionary biology to help build phylogenetic trees of a large number of sequences can provide a very useful tool for evolutionary biologists. One limitation of the proposed VAE approach is that its input needs to be protein sequences from a multiple sequence alignment (MSA) and obtaining MSAs is much more difficult than collecting protein sequences. Therefore, a very useful extension of the current VAE approach is to make it work with unaligned protein sequences, which can significantly increase the amount of data available to train the model. Moreover, working with unaligned protein sequences could also enable the VAE method to model the sequence distribution of multiple protein

families, which might be able to provide insights into the evolutionary relationship between protein families in addition to the evolutionary relationship between sequences in the same protein family. Another straightforward application of the VAE approach is for analyzing RNA sequences, because, similar to protein sequences, RNA sequences can also be clustered into different families and combined into sequence alignments [140].

These methodological advances are made by combining and adapting theories, methods, and tools from multiple disciplines including statistical mechanics, statistics, machine learning, and computer science. In studying theories and methods from different disciplines and developing new interdisciplinary methods, I have learned several lessons about how different research fields are closely connected and how new methods developed in one field can have a big impact in another field. Therefore, it is the second purpose of this chapter to share some of these lessons.

Free energy has been a quantity of great interest to calculate in statistical mechanics and computational chemistry. At the same time, free energy is also a quantity of interest in statistics, especially in Bayesian statistics. In Bayesian statistics, calculating free energies is usually known as calculating normalization constants or calculating evidence, which is necessary when deciding which model/assumption is more convincing given observed data. Therefore, researchers from both statistical mechanics and Bayesian statistics have been developing similar or equivalent methods for calculating free energies. For example, the free energy perturbation (FEP) method using the Zwanzig equation developed in statistical mechanics is equivalent to the importance sampling method developed in statistics, although they were developed independently in the two fields in 1950s [51, 141]. In 1976, the Bennett acceptance ratio (BAR) method was developed in statistical mechanics and it is a much better method than the FEP method for calculating free energies [142]. However, the BAR method was not widely known or used in statistics until 20 years later in 1998 when the BAR method was introduced and analyzed in statistics[143]. This shows that, despite the fact that researchers from both fields are trying to solving the same problem,

sometimes researchers from one field are not aware of progress made in the other field and it takes a long time for a new idea proposed in one field to be borrowed and have an impact on another field. In addition to the free energy methods mentioned above, another research area shared by both statistical mechanics and Bayesian statistics is development of enhanced sampling methods. Making connections between enhanced sampling methods that have been developed in the two disciplines could provide insights and help develop better sampling methods that can benefit both fields.

The last lesson I want to share is that, during the last 10 years or so, there have been great advances in machine learning, especially in the area of unsupervised learning[123, 144, 145]. Combined with the availability of a large amount of data, these advances have been making large impacts on traditional machine learning areas such as image processing, recommendation systems, and natural language processing[145]. Due to the advances of both imaging and sequencing technology in biology, the amount of data in biology and medicine, especially sequence data, has been rapidly increasing [98, 97, 140]. Most of these data in biology is unlabelled. Therefore, unsupervised learning methods are required to learn valuable information from the large amount of unlabelled data. The application of the new advances in unsupervised machine learning methods in biology and medicine is still in a starting stage and should be a research area worth much further investigation.

BIBLIOGRAPHY

[1] Drexler, K. E., "Molecular engineering: An approach to the development of general capabilities for molecular manipulation," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 78, No. 9, 1981, pp. 5275–5278.

[2] Paul, S. M., Mytelka, D. S., Dunwiddie, C. T., Persinger, C. C., Munos, B. H., Lindborg, S. R., and Schacht, A. L., "How to improve R&D productivity: the pharmaceutical industry's grand challenge," *Nat. Rev. Drug Discov.*, Vol. 9, No. 3, 2010, pp. 203.

[3] Jäckel, C., Kast, P., and Hilvert, D., "Protein design by directed evolution," *Annu. Rev. Biophys.*, Vol. 37, 2008, pp. 153–173.

[4] Jorgensen, W. L., "The Many Roles of Computation in Drug Discovery," *Science*, Vol. 303, No. 5665, March 2004, pp. 1813–1818.

[5] Looger, L. L., Dwyer, M. A., Smith, J. J., and Hellinga, H. W., "Computational design of receptor and sensor proteins with novel functions," *Nature*, Vol. 423, No. 6936, 2003, pp. 185.

[6] Chodera, J. D., Mobley, D. L., Shirts, M. R., Dixon, R. W., Branson, K., and Pande, V. S., "Alchemical free energy methods for drug discovery: progress and challenges," *Curr. Opin. Struct. Biol.*, Vol. 21, No. 2, April 2011, pp. 150–160.

[7] Wu, G., Robertson, D. H., Brooks III, C. L., and Vieth, M., "Detailed analysis of grid-based molecular docking: A case study of CDOCKER-A CHARMm-based MD docking algorithm," *J. Comput. Chem.*, Vol. 24, No. 13, 2003, pp. 1549–1562.

[8] Ding, X., Vilseck, J. Z., Hayes, R. L., and Brooks III, C. L., "Gibbs Sampler-Based λ-Dynamics and Rao–Blackwell Estimator for Alchemical Free Energy Calculation," *J. Chem. Theory Comput.*, Vol. 13, No. 6, 2017, pp. 2501–2510.

[9] Gagnon, J. K., Law, S. M., and Brooks III, C. L., "Flexible CDOCKER: Development and application of a pseudo-explicit structure-based docking method within CHARMM," *J. Comput. Chem.*, Vol. 37, No. 8, 2016, pp. 753–762.

[10] Sousa, S. F., Fernandes, P. A., and Ramos, M. J., "Protein–ligand docking: current status and future challenges," *Proteins: Struct., Funct., Bioinf.*, Vol. 65, No. 1, 2006, pp. 15–26.

[11] Goodsell, D. S., Morris, G. M., and Olson, A. J., "Automated docking of flexible ligands: applications of AutoDock," *J. Mol. Recognit.*, Vol. 9, No. 1, 1996, pp. 1–5.

[12] Trott, O. and Olson, A. J., "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *J. Comput. Chem.*, Vol. 31, No. 2, 2010, pp. 455–461.

[13] Allen, W. J., Balius, T. E., Mukherjee, S., Brozell, S. R., Moustakas, D. T., Lang, P. T., Case, D. A., Kuntz, I. D., and Rizzo, R. C., "DOCK 6: impact of new features and current docking performance," *J. Comput. Chem.*, Vol. 36, No. 15, 2015, pp. 1132–1156.

[14] Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., et al., "Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy," *J. Med. Chem.*, Vol. 47, No. 7, 2004, pp. 1739–1749.

[15] Halgren, T. A., Murphy, R. B., Friesner, R. A., Beard, H. S., Frye, L. L., Pollard, W. T., and Banks, J. L., "Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening," *J. Med. Chem.*, Vol. 47, No. 7, 2004, pp. 1750–1759.

[16] Brooks, B. R., Brooks III, C. L., Mackerell Jr, A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., et al., "CHARMM: the biomolecular simulation program," *J. Comput. Chem.*, Vol. 30, No. 10, 2009, pp. 1545–1614.

[17] Vanommeslaeghe, K., Hatcher, E., Acharya, C., Kundu, S., Zhong, S., Shim, J., Darian, E., Guvench, O., Lopes, P., Vorobyov, I., et al., "CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields," *J. Comput. Chem.*, Vol. 31, No. 4, 2010, pp. 671–690.

[18] Brigham, E. O. and Brigham, E. O., *The fast Fourier transform and its applications*, Vol. 448, prentice Hall Englewood Cliffs, NJ, 1988.

[19] Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., and Vakser, I. A., "Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 89, No. 6, 1992, pp. 2195–2199.

[20] Bracewell, R. N. and Bracewell, R. N., *The Fourier transform and its applications*, Vol. 31999, McGraw-Hill New York, 1986.

[21] Gabb, H. A., Jackson, R. M., and Sternberg, M. J., "Modelling protein docking using shape complementarity, electrostatics and biochemical information1," *J. Mol. Biol.*, Vol. 272, No. 1, 1997, pp. 106–120.

[22] Mandell, J. G., Roberts, V. A., Pique, M. E., Kotlovyi, V., Mitchell, J. C., Nelson, E., Tsigelny, I., and Ten Eyck, L. F., "Protein docking using continuum electrostatics and geometric fit," *Protein Eng. Des. Sel.*, Vol. 14, No. 2, 2001, pp. 105–113.

[23] Chen, R., Li, L., and Weng, Z., "ZDOCK: an initial-stage protein-docking algorithm," *Proteins: Struct., Funct., Bioinf.*, Vol. 52, No. 1, 2003, pp. 80–87.

[24] Ritchie, D. W., Kozakov, D., and Vajda, S., "Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational FFT generating functions," *Bioinformatics*, Vol. 24, No. 17, 2008, pp. 1865–1873.

[25] Garzon, J. I., Lopéz-Blanco, J. R., Pons, C., Kovacs, J., Abagyan, R., Fernandez-Recio, J., and Chacon, P., "FRODOCK: a new approach for fast rotational protein–protein docking," *Bioinformatics*, Vol. 25, No. 19, 2009, pp. 2544–2551.

[26] Padhorny, D., Kazennov, A., Zerbe, B. S., Porter, K. A., Xia, B., Mottarella, S. E., Kholodov, Y., Ritchie, D. W., Vajda, S., and Kozakov, D., "Protein–protein docking by fast generalized Fourier transforms on 5D rotational manifolds," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 113, No. 30, 2016, pp. E4286–E4293.

[27] Ritchie, D. W. and Venkatraman, V., "Ultra-fast FFT protein docking on graphics processors," *Bioinformatics*, Vol. 26, No. 19, 2010, pp. 2398–2405.

[28] Sukhwani, B. and Herbordt, M. C., "GPU acceleration of a production molecular docking code," *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, ACM, 2009, pp. 19–27.

[29] Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S., "PIPER: an FFT-based protein docking program with pairwise potentials," *Proteins: Struct., Funct., Bioinf.*, Vol. 65, No. 2, 2006, pp. 392–406.

[30] Padhorny, D., Hall, D. R., Mirzaei, H., Mamonov, A. B., Moghadasi, M., Alekseenko, A., Beglov, D., and Kozakov, D., "Protein–ligand docking using FFT based sampling: D3R case study," *J. Comput. Aided Mol. Des.*, Vol. 32, No. 1, 2018, pp. 225–230.

[31] Salomon-Ferrer, R., Gtz, A. W., Poole, D., Le Grand, S., and Walker, R. C., "Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent particle mesh Ewald," *J. Chem. Theory Comput.*, Vol. 9, No. 9, 2013, pp. 3878–3888.

[32] Stone, J. E., Hardy, D. J., Ufimtsev, I. S., and Schulten, K., "GPU-accelerated molecular modeling coming of age," *J. Mol. Graph. Model.*, Vol. 29, No. 2, 2010, pp. 116–125.

[33] Hynninen, A.-P. and Crowley, M. F., "New faster CHARMM molecular dynamics engine," *J. Comput. Chem.*, Vol. 35, No. 5, 2014, pp. 406–413.

[34] Eastman, P., Friedrichs, M. S., Chodera, J. D., Radmer, R. J., Bruns, C. M., Ku, J. P., Beauchamp, K. A., Lane, T. J., Wang, L.-P., Shukla, D., Tye, T., Houston, M., Stich, T., Klein, C., Shirts, M. R., and Pande, V. S., "OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation," *J. Chem. Theory Comput.*, Vol. 9, No. 1, Nov. 2012, pp. 461–469.

[35] Nvidia, C., "CUFFT library," 2010.

[36] Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., et al., "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," *PLOS Comput. Biol.*, Vol. 13, No. 7, 2017, pp. e1005659.

[37] Hartshorn, M. J., Verdonk, M. L., Chessari, G., Brewerton, S. C., Mooij, W. T., Mortenson, P. N., and Murray, C. W., "Diverse, high-quality test set for the validation of protein- ligand docking performance," *J. Med. Chem.*, Vol. 50, No. 4, 2007, pp. 726–741.

[38] Mukherjee, S., Balius, T. E., and Rizzo, R. C., "Docking validation resources: protein family and ligand flexibility experiments," *J. Chem. Inf. Model.*, Vol. 50, No. 11, 2010, pp. 1986–2000.

[39] Vanommeslaeghe, K., Hatcher, E., Acharya, C., Kundu, S., Zhong, S., Shim, J., Darian, E., Guvench, O., Lopes, P., Vorobyov, I., and Mackerell, A. D., "CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields," *J. Comput. Chem.*, Vol. 31, No. 4, March 2010, pp. 671–690.

[40] Verdonk, M. L., Cole, J. C., Hartshorn, M. J., Murray, C. W., and Taylor, R. D., "Improved protein–ligand docking using GOLD," *Proteins: Struct., Funct., Bioinf.*, Vol. 52, No. 4, 2003, pp. 609–623.

[41] Wang, R., Lai, L., and Wang, S., "Further development and validation of empirical scoring functions for structure-based binding affinity prediction," *J. Comput. Aided Mol. Des.*, Vol. 16, No. 1, 2002, pp. 11–26.

[42] Eldridge, M. D., Murray, C. W., Auton, T. R., Paolini, G. V., and Mee, R. P., "Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes," *J. Comput. Aided Mol. Des.*, Vol. 11, No. 5, 1997, pp. 425–445.

[43] Tao, P. and Lai, L., "Protein ligand docking based on empirical method for binding affinity estimation," *J. Comput. Aided Mol. Des.*, Vol. 15, No. 5, 2001, pp. 429–446.

[44] Vieth, M., Hirst, J. D., and Brooks, C. L., "Do active site conformations of small ligands correspond to low free-energy solution structures?" *J. Comput. Aided Mol. Des.*, Vol. 12, No. 6, 1998, pp. 563–572.

[45] Brooks, B. R., Brooks, III, C. L., Mackerell, A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., Caflisch, A., Caves, L., Cui, Q., Dinner, A. R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R. W., Post, C. B., Pu, J. Z., Schaefer, M., Tidor, B., Venable, R. M., Woodcock, H. L., Wu, X., Yang, W., York, D. M., and Karplus, M., "CHARMM: The biomolecular simulation program," *J. Comput. Chem.*, Vol. 30, No. 10, July 2009, pp. 1545–1614.

[46] Gaillard, T., "Evaluation of AutoDock and AutoDock Vina on the CASF-2013 benchmark," *J. Chem. Inf. Model.*, Vol. 58, No. 8, 2018, pp. 1697–1706.

[47] Nguyen, T. H., Zhou, H.-X., and Minh, D. D., "Using the fast fourier transform in binding free energy calculations," *J. Comput. Chem.*, Vol. 39, No. 11, 2018, pp. 621–636.

[48] Bash, P. A., Singh, U. C., Langridge, R., and Kollman, P. A., "Free Energy Calculations by Computer Simulation," *Science*, Vol. 236, No. 4801, June 1987, pp. 564–8.

[49] Shirts, M. R., Mobley, D. L., and Chodera, J. D., "Chapter 4 Alchemical Free Energy Calculations: Ready for Prime Time?" Elsevier, 2007, pp. 41–59.

[50] Wang, L., Wu, Y., Deng, Y., Kim, B., Pierce, L., Krilov, G., Lupyan, D., Robinson, S., Dahlgren, M. K., Greenwood, J., Romero, D. L., Masse, C., Knight, J. L., Steinbrecher, T., Beuming, T., Damm, W., Harder, E., Sherman, W., Brewer, M., Wester, R., Murcko, M., Frye, L., Farid, R., Lin, T., Mobley, D. L., Jorgensen, W. L., Berne, B. J., Friesner, R. A., and Abel, R., "Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field," *J. Phys. Chem. B*, Vol. 137, No. 7, Feb. 2015, pp. 2695–2703.

[51] Zwanzig, R. W., "High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases," *J. Chem. Phys.*, Vol. 22, No. 8, Aug. 1954, pp. 1420–1426.

[52] Zwanzig, R. W., "High-Temperature Equation of State by a Perturbation Method. II. Polar Gases," *J. Chem. Phys.*, Vol. 23, No. 10, Oct. 1955, pp. 1915–1922.

[53] Straatsma, T. P. and Berendsen, H. J. C., "Free energy of ionic hydration: Analysis of a thermodynamic integration technique to evaluate free energy differences by molecular dynamics simulations," *J. Chem. Phys.*, Vol. 89, No. 9, Aug. 1998, pp. 5876–5886.

[54] Christ, C. D. and van Gunsteren, W. F., "Enveloping distribution sampling: A method to calculate free energy differences from a single simulation," *J. Chem. Phys.*, Vol. 126, No. 18, May 2007, pp. 184110.

[55] Riniker, S., Christ, C. D., Hansen, N., Mark, A. E., Nair, P. C., and van Gunsteren, W. F., "Comparison of enveloping distribution sampling and thermodynamic integration to calculate binding free energies of phenylethanolamine N-methyltransferase inhibitors," *J. Chem. Phys.*, Vol. 135, No. 2, July 2011, pp. 024105.

[56] Kong, X. and Brooks, III, C. L., "λ-dynamics: A new approach to free energy calculations," *J. Chem. Phys.*, Vol. 105, No. 6, June 1998, pp. 2414–2423.

[57] Guo, Z., Brooks, III, C. L., and Kong, X., "Efficient and flexible algorithm for free energy calculations using the λ-dynamics approach," *J. Phys. Chem. B*, Vol. 102, No. 11, 1998, pp. 2032–2036.

[58] Guo, Z. and Brooks, III, C. L., "Rapid screening of binding affinities: application of the λ-dynamics method to a trypsin-inhibitor system," *J. Phys. Chem. B*, Vol. 120, No. 8, 1998, pp. 1920–1921.

[59] Knight, J. L. and Brooks, III, C. L., "λ-Dynamics free energy simulation methods," *J. Comput. Chem.*, Vol. 30, No. 11, Aug. 2009, pp. 1692–1700.

[60] Knight, J. L. and Brooks, III, C. L., "Multisite λ Dynamics for Simulated Structure–Activity Relationship Studies," *J. Chem. Theory Comput.*, Vol. 7, No. 9, Aug. 2011, pp. 2728–2739.

[61] Armacost, K. A., Goh, G. B., and Brooks, III, C. L., "Biasing Potential Replica Exchange Multisite λ-Dynamics for Efficient Free Energy Calculations," *J. Chem. Theory Comput.*, Vol. 11, No. 3, Feb. 2015, pp. 1267–1277.

[62] Zheng, L., Chen, M., and Yang, W., "Random walk in orthogonal space to achieve efficient free-energy simulation of complex systems." *Proc. Natl. Acad. Sci. U. S. A.*, Vol. 105, No. 51, Dec. 2008, pp. 20227–20232.

[63] Goh, G. B., Hulbert, B. S., Zhou, H., and Brooks, III, C. L., "Constant pH molecular dynamics of proteins in explicit solvent with proton tautomerism," *Proteins: Struct., Funct., Bioinf.*, Vol. 82, No. 7, July 2014, pp. 1319–1331.

[64] Hayes, R. L., Armacost, K. A., Vilseck, J. Z., and Brooks, C. L., "Adaptive Landscape Flattening Accelerates Sampling of Alchemical Space in Multisite Dynamics," *J. Phys. Chem. B*, 2017, PMID: 28112940, doi:10.1021/acs.jpcb.6b09656.

[65] Chodera, J. D. and Shirts, M. R., "Replica exchange and expanded ensemble simulations as gibbs sampling: Simple improvements for enhanced mixing," *J. Chem. Phys.*, Vol. 135, No. 19, 2011, pp. 194110.

[66] Shirts, M. R. and Chodera, J. D., "Statistically optimal analysis of samples from multiple equilibrium states," *J. Chem. Phys.*, Vol. 129, No. 12, 2008, pp. 124105.

[67] Tan, Z., Gallicchio, E., Lapelosa, M., and Levy, R. M., "Theory of binless multi-state free energy estimation with applications to protein-ligand binding," *J. Chem. Phys.*, Vol. 136, No. 14, April 2012, pp. 144102.

[68] Zhang, B. W., Xia, J., Tan, Z., and Levy, R. M., "A stochastic solution to the unbinned WHAM equations," *J. Phys. Chem. Lett.*, Vol. 6, No. 19, 2015, pp. 3834–3840.

[69] Geman, S. and Geman, D., "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-6, No. 6, 1984, pp. 721–741.

[70] Smith, A. F. and Roberts, G. O., "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods," *J. R. Stat. Soc. B.*, 1993, pp. 3–23.

[71] Ritter, C. and Tanner, M. A., "Facilitating the Gibbs sampler: the Gibbs stopper and the griddy-Gibbs sampler," *J. Am. Stat. Assoc.*, Vol. 87, No. 419, 1992, pp. 861–868.

[72] Wang, F. and Landau, D. P., "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States," *Phys. Rev. Lett.*, Vol. 86, No. 10, March 2001, pp. 2050–2053.

[73] Zhang, B. W., Dai, W., Gallicchio, E., He, P., Xia, J., Tan, Z., and Levy, R. M., "Simulating replica exchange: Markov state models, proposal schemes, and the infinite swapping limit," *J. Phys. Chem. B*, Vol. 120, No. 33, 2016, pp. 8289–8301.

[74] Plattner, N., Doll, J., Dupuis, P., Wang, H., Liu, Y., and Gubernatis, J., "An infinite swapping approach to the rare-event sampling problem," *J. Chem. Phys.*, Vol. 135, No. 13, 2011, pp. 134111.

[75] Dupuis, P., Liu, Y., Plattner, N., and Doll, J. D., "On the infinite swapping limit for parallel tempering," *Multiscale Model. Simul.*, Vol. 10, No. 3, 2012, pp. 986–1022.

[76] Plattner, N., Doll, J., and Meuwly, M., "Overcoming the rare event sampling problem in biological systems with infinite swapping," *J. Chem. Theory Comput.*, Vol. 9, No. 9, 2013, pp. 4215–4224.

[77] Beutler, T. C., Mark, A. E., van Schaik, R. C., Gerber, P. R., and van Gunsteren, W. F., "Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations," *Chem. Phys. Lett.*, Vol. 222, No. 6, June 1994, pp. 529–539.

[78] Rao, C. R., "Information and accuracy attainable in the estimation of statistical parameters," *Bull. Calcutta Math. Soc.*, Vol. 37, No. 3, 1945, pp. 81–91.

[79] Blackwell, D., "Conditional expectation and unbiased sequential estimation," *Ann. Math. Stat.*, 1947, pp. 105–110.

[80] Gelfand, A. E. and Smith, A. F., "Sampling-based approaches to calculating marginal densities," *J. Am. Stat. Assoc.*, Vol. 85, No. 410, 1990, pp. 398–409.

[81] Pearl, J., "Evidential reasoning using stochastic simulation of causal models," *Artificial Intelligence*, Vol. 32, No. 2, May 1987, pp. 245–257.

[82] Morton, A. and Matthews, B. W., "Specificity of ligand binding in a buried nonpolar cavity of T4 lysozyme: linkage of dynamics and structural plasticity," *Biochemistry*, Vol. 34, No. 27, 1995, pp. 8576–8588.

[83] Morton, A., Baase, W. A., and Matthews, B. W., "Energetic origins of specificity of ligand binding in an interior nonpolar cavity of T4 lysozyme," *Biochemistry*, Vol. 34, No. 27, 1995, pp. 8564–8575.

[84] Yesselman, J. D., Price, D. J., Knight, J. L., and Brooks, III, C. L., "MATCH: An atom-typing toolset for molecular mechanics force fields," *J. Comput. Chem.*, Vol. 33, No. 2, Jan. 2012, pp. 189–202.

[85] Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., and Klein, M. L., "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.*, Vol. 79, No. 2, 1983, pp. 926–935.

[86] Steinbach, P. J. and Brooks, B. R., "New spherical-cutoff methods for long-range forces in macromolecular simulation," *J. Comput. Chem.*, Vol. 15, No. 7, 1994, pp. 667–683.

[87] Van Gunsteren, W. F. and Berendsen, H. J. C., "Algorithms for macromolecular dynamics and constraint dynamics," *Mol. Phys.*, Vol. 34, No. 5, Aug. 2006, pp. 1311–1327.

[88] Mobley, D. L., Chodera, J. D., and Dill, K. A., "The Confine-and-Release Method: Obtaining Correct Binding Free Energies in the Presence of Protein Conformational Change." *J. Chem. Theory Comput.*, Vol. 3, No. 4, 2007, pp. 1231–1235.

[89] Jiang, W. and Roux, B., "Free Energy Perturbation Hamiltonian Replica-Exchange Molecular Dynamics (FEP/H-REMD) for Absolute Ligand Binding Free Energy Calculations," *J. Chem. Theory Comput.*, Vol. 6, No. 9, July 2010, pp. 2559–2565.

[90] Wang, L., Berne, B. J., and Friesner, R. A., "On achieving high accuracy and reliability in the calculation of relative protein-ligand binding affinities." *Proc. Natl. Acad. Sci. U. S. A.*, Vol. 109, No. 6, Feb. 2012, pp. 1937–1942.

[91] Tobias, D. J., Brooks, III, C. L., and Fleischman, S. H., "Conformational flexibility in free energy simulations," *Chem. Phys. Lett.*, Vol. 156, No. 2-3, March 1989, pp. 256–260.

[92] Shirts, M. R. and Pande, V. S., "Solvation free energies of amino acid side chain analogs for common molecular mechanics water models," *J. Chem. Phys.*, Vol. 122, No. 13, April 2005, pp. 134508.

[93] Best, R. B., Zhu, X., Shim, J., Lopes, P. E. M., Mittal, J., Feig, M., and MacKerell Jr., A. D., "Optimization of the Additive CHARMM All-Atom Protein Force Field Targeting Improved Sampling of the Backbone ϕ, ψ and Side-Chain χ 1and χ 2Dihedral Angles," *J. Chem. Theory Comput.*, Vol. 8, No. 9, Sept. 2012, pp. 3257–3273.

[94] Harper, M., Weinstein, B., Simon, C., chebee7i, Swanson-Hysell, N., The Gitter, B., Maximiliano, G., and Guido, Z., "python-ternary: Ternary Plots in Python," 2015.

[95] Goh, G. B., Knight, J. L., and Brooks, III, C. L., "Constant pH Molecular Dynamics Simulations of Nucleic Acids in Explicit Solvent," *J. Chem. Theory Comput.*, Vol. 8, No. 1, Dec. 2011, pp. 36–46.

[96] Wallace, J. A. and Shen, J. K., "Continuous Constant pH Molecular Dynamics in Explicit Solvent with pH-Based Replica Exchange," *J. Chem. Theory Comput.*, Vol. 7, No. 8, July 2011, pp. 2617–2629.

[97] Consortium, U. et al., "UniProt: the universal protein knowledgebase," *Nucleic Acids Res.*, Vol. 46, No. 5, 2018, pp. 2699.

[98] Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., et al., "The Pfam protein families database: towards a more sustainable future," *Nucleic Acids Res.*, Vol. 44, No. D1, 2015, pp. D279–D285.

[99] Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T., "Identification of direct residue contacts in protein–protein interaction by message passing," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 106, No. 1, 2009, pp. 67–72.

[100] Onuchic, J. N. and Morcos, F., "Protein Sequence Coevolution, Energy Landscapes and their Connections to Protein Structure, Folding and Function," *Biophys. J.*, Vol. 114, No. 3, 2018, pp. 389a.

[101] Ortiz, A. R., Kolinski, A., Rotkiewicz, P., Ilkowski, B., and Skolnick, J., "Ab initio folding of proteins using restraints derived from evolutionary information," .

[102] Skolnick, J., Koliński, A., Brooks III, C. L., Godzik, A., Rey, A., et al., "A method for prediction of protein structure from sequence," *Curr. Biol.*, Vol. 3, 1993.

[103] Roy, A., Kucukural, A., and Zhang, Y., "I-TASSER: a unified platform for automated protein structure and function prediction," *Nat. Protoc.*, Vol. 5, No. 4, 2010, pp. 725.

[104] Bueno, C. A., Potoyan, D. A., Cheng, R. R., and Wolynes, P. G., "Prediction of Changes in Protein Folding Stability Upon Single Residue Mutations," *Biophys. J.*, Vol. 114, No. 3, 2018, pp. 199a.

[105] Wheeler, L. C., Lim, S. A., Marqusee, S., and Harms, M. J., "The thermostability and specificity of ancient proteins," *Curr. Opin. Struct. Biol.*, Vol. 38, 2016, pp. 37–43.

[106] Lim, S. A., Hart, K. M., Harms, M. J., and Marqusee, S., "Evolutionary trend toward kinetic stability in the folding trajectory of RNases H," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 113, No. 46, 2016, pp. 13045–13050.

[107] Hart, K. M., Harms, M. J., Schmidt, B. H., Elya, C., Thornton, J. W., and Marqusee, S., "Thermodynamic system drift in protein evolution," *PLOS Biol.*, Vol. 12, No. 11, 2014, pp. e1001994.

[108] Levy, R. M., Haldane, A., and Flynn, W. F., "Potts Hamiltonian models of protein co-variation, free energy landscapes, and evolutionary fitness," *Curr. Opin. Struct. Biol.*, Vol. 43, 2017, pp. 55–62.

[109] Flynn, W. F., Haldane, A., Torbett, B. E., and Levy, R. M., "Inference of epistatic effects leading to entrenchment and drug resistance in HIV-1 protease," *Mol. Biol. Evol.*, Vol. 34, No. 6, 2017, pp. 1291–1306.

[110] Figliuzzi, M., Jacquier, H., Schug, A., Tenaillon, O., and Weigt, M., "Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1," *Mol. Biol. Evol.*, Vol. 33, No. 1, 2015, pp. 268–280.

[111] Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C., "Protein 3D structure computed from evolutionary sequence variation," *PloS one*, Vol. 6, No. 12, 2011, pp. e28766.

[112] Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., Sander, C., and Marks, D. S., "Mutation effects predicted from sequence co-variation," *Nat. Biotechnol.*, Vol. 35, No. 2, 2017, pp. 128.

[113] Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M., "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 108, No. 49, 2011, pp. E1293–E1301.

[114] Balakrishnan, S., Kamisetty, H., Carbonell, J. G., Lee, S.-I., and Langmead, C. J., "Learning generative models for protein fold families," *Proteins: Struct., Funct., Bioinf.*, Vol. 79, No. 4, 2011, pp. 1061–1078.

[115] Söding, J., Biegert, A., and Lupas, A. N., "The HHpred interactive server for protein homology detection and structure prediction," *Nucleic Acids Res.*, Vol. 33, No. suppl_2, 2005, pp. W244–W248.

[116] Thompson, J. D., Higgins, D. G., and Gibson, T. J., "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Res.*, Vol. 22, No. 22, 1994, pp. 4673–4680.

[117] Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E., "Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models," *Phys. Rev. E*, Vol. 87, No. 1, 2013, pp. 012707.

[118] Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., and Weigt, M., "Inverse statistical physics of protein sequences: A key issues review," *Rep. Prog. Phys.*, Vol. 81, No. 3, 2018, pp. 032601.

[119] Ovchinnikov, S., Kamisetty, H., and Baker, D., "Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information," *Elife*, Vol. 3, 2014.

[120] Sailer, Z. R. and Harms, M. J., "High-order epistasis shapes evolutionary trajectories," *PLOS Comput. Biol.*, Vol. 13, No. 5, 2017, pp. e1005541.

[121] Sailer, Z. R. and Harms, M. J., "Molecular ensembles make evolution unpredictable," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 114, No. 45, 2017, pp. 11938–11943.

[122] Sailer, Z. R. and Harms, M. J., "Detecting high-order epistasis in nonlinear genotype-phenotype maps," *Genetics*, 2017, pp. genetics–116.

[123] Kingma, D. P. and Welling, M., "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[124] Rasmussen, C. E., "Gaussian processes in machine learning," *Advanced lectures on machine learning*, Springer, 2004, pp. 63–71.

[125] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S., "Generating sentences from a continuous space," *arXiv preprint arXiv:1511.06349*, 2015.

[126] Ravanbakhsh, S., Lanusse, F., Mandelbaum, R., Schneider, J. G., and Poczos, B., "Enabling Dark Energy Science with Deep Generative Models of Galaxy Images." *AAAI*, 2017, pp. 1488–1494.

[127] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A., "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Cent. Sci.*, 2018.

[128] Henikoff, S. and Henikoff, J. G., "Position-based sequence weights," *J. Mol. Biol.*, Vol. 243, No. 4, 1994, pp. 574–578.

[129] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning internal representations by error propagation," Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[130] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[131] Huerta-Cepas, J., Serra, F., and Bork, P., "ETE 3: reconstruction, analysis, and visualization of phylogenomic data," *Mol. Biol. Evol.*, Vol. 33, No. 6, 2016, pp. 1635–1638.

[132] Le, S. Q. and Gascuel, O., "An improved general amino acid replacement matrix," *Mol. Biol. Evol.*, Vol. 25, No. 7, 2008, pp. 1307–1320.

[133] Hurley, J. H., Baase, W. A., and Matthews, B. W., "Design and structural analysis of alternative hydrophobic core packing arrangements in bacteriophage T4 lysozyme," *J. Mol. Biol.*, Vol. 224, No. 4, 1992, pp. 1143–1159.

[134] Gromiha, M. M., An, J., Kono, H., Oobatake, M., Uedaira, H., Prabakaran, P., and Sarai, A., "ProTherm, version 2.0: thermodynamic database for proteins and mutants," *Nucleic Acids Res.*, Vol. 28, No. 1, 2000, pp. 283–285.

[135] Otey, C. R., Landwehr, M., Endelman, J. B., Hiraga, K., Bloom, J. D., and Arnold, F. H., "Structure-guided recombination creates an artificial family of cytochromes P450," *PLOS Biol.*, Vol. 4, No. 5, 2006, pp. e112.

[136] Romero, P. A., Krause, A., and Arnold, F. H., "Navigating the protein fitness landscape with Gaussian processes," *Proc. Natl. Acad. Sci. U.S.A*, Vol. 110, No. 3, 2013, pp. E193–E201.

[137] Li, Y., Drummond, D. A., Sawayama, A. M., Snow, C. D., Bloom, J. D., and Arnold, F. H., "A diverse family of thermostable cytochrome P450s created by recombination of stabilizing fragments," *Nat. Biotechnol.*, Vol. 25, No. 9, 2007, pp. 1051.

[138] Yang, K. K., Wu, Z., Bedbrook, C. N., and Arnold, F. H., "Learned protein embeddings for machine learning," *Bioinformatics*, Vol. 1, 2018, pp. 7.

[139] Fowler, D. M. and Fields, S., "Deep mutational scanning: a new style of protein science," *Nat. Methods*, Vol. 11, No. 8, 2014, pp. 801.

[140] Kalvari, I., Argasinska, J., Quinones-Olvera, N., Nawrocki, E. P., Rivas, E., Eddy, S. R., Bateman, A., Finn, R. D., and Petrov, A. I., "Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families," *Nucleic Acids Res.*, Vol. 46, No. D1, 2017, pp. D335–D342.

[141] Kahn, H. and Harris, T. E., "Estimation of particle transmission by random sampling," *National Bureau of Standards applied mathematics series*, Vol. 12, 1951, pp. 27–30.

[142] Bennett, C. H., "Efficient estimation of free energy differences from Monte Carlo data," *J. Comput. Phys.*, Vol. 22, No. 2, Oct. 1976, pp. 245–268.

[143] Gelman, A. and Meng, X.-L., "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Stat. Sci.*, 1998, pp. 163–185.

[144] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[145] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *nature*, Vol. 521, No. 7553, 2015, pp. 436.