May 2020

Using Novel Approaches for Navigating Complex Energy Landscapes: Ion Channel Conductance using Hyperdynamics and Human-Guided Global Optimization of Lennard-Jones Clusters

Wenxing Zhang
Clemson University, wxzhang1994@gmail.com

Using Novel Approaches for Navigating Complex Energy Landscapes: Ion Channel Conductance using Hyperdynamics and Human-Guided Global Optimization of Lennard-Jones Clusters

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Chemistry

---

by
Wenxing Zhang
May 2020

Accepted by:
Dr. Steven Stuart, Committee Chair
Dr. Brain Dominy
Dr. Leah Casabianca
Dr. Andrew Robb

Abstract

Molecular dynamics (MD) is a widely used tool to study molecular systems on atomic level. However, the timescale of a traditional MD simulation is typically limited to nanoseconds. Thus many interesting processes that occur on microseconds or larger timescale can't be studied. Hyperdynamics provides a way to extend the timescale of MD simulation. In hyperdynamics, MD is performed on a biased potential then corrected to get true dynamics provided certain conditions are met. Here, we tried to study potassium channel conductance using the hyperdynamics method with a bias potential constructed based on the potential of mean force of ion translocation through the selective filter of a potassium ion channel. However, when MD was performed on this biased potential, no ion translocation events were observed. Although some new insights were gained into the rate-limiting steps for ion mobility in this system from these negative results, no further studies are planned with this project.

The second project is based on the assumption that hybrid human–computational algorithm is more efficient than purely computational algorithm itself. Such ideas have already been studied by many "crowd-sourcing" games, such as Foldit [1] for the protein structure prediction problem, and QuantumMoves [2] for quantum physics. Here, the same idea is applied to cluster structure optimization. A virtual reality android cellphone app was developed to study global optimization of Lennard-Jones clusters

with both computational algorithm and hybrid human–computational algorithm. Using linear mixed model analysis, we found statistically significant differences between the expected runtime of both methods, at least for cluster of certain sizes. Further analysis of the data showing human intelligence weakened the strong dependence of the efficiency of the computationl method on cluster sizes. We hypothesis that this is due to that humans are able to make large moves that allows the alogrithm to cover a large region in the potential energy surface faster. Further studies with more cluster sizes are needed to draw a more complete conclusion. Human intelligence can potentially be integrated into more advanced optimization technique and applied to more complicated optimization problems in the future. Patterns analysis of human behaviors during the optmization process can be conducted to gain insights of mechanisms and strategies of optimization process.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Stuart, for his expert guidance through my PhD study.

Additional thanks to my committee members, Dr. Dominy, Dr. Casabianca and Dr. Robb, for their valuable advices.

Special thanks to Ayobamidele and Jocelyn, for their help with the data collection for the VR project.

Finally, I would thank my family and friends for their support.

Table of Contents

Title Page . . . . . **i**

Abstract . . . . . **ii**

Acknowledgments . . . . . **iv**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

I Ion Channel Conductance with Hyperdynamics **1**

1 Introduction **2**
1.1 Molecular Dynamics 2
1.2 The CHARMM Force Field 5
1.3 Hyperdynamics 7
1.4 Potential of Mean Force 10

2 Ion channel conductance with hyperdynamics **12**
2.1 Introduction 12
2.2 Methods 18
2.3 Results and Discussion 20
2.4 Conclusions 30

II Human-Guided Global Optimization of Lennard-Jones Clusters **32**

3 Introduction **33**
3.1 Lennard-Jones Cluster 33
3.2 Optimization Methods 36
3.3 Virtual Reality 38
3.4 Hypothesis Testing 39

4 Human-Guided Global Optimization of Lennard-Jones Clusters . 43
4.1 Introduction 43
4.2 Methods 47
4.3 Results and Discussion 58
4.4 Conclusions 80

5 Future Work 82

Appendices 84
A Informed Consent Form 85

Bibliography 87

List of Tables

2.1 Values for parameters in equation 2.1 27

3.1 Four outcomes of a hypothesis test. 40

4.1 n_{opt} values for cluster of different sizes 63
4.2 Participant records. (Data for participant 14 was discarded due to the device overheating during the experiment.) 69
4.3 Algorithm runtimes (number of timesteps) for different cluster sizes. . 70
4.4 γ values for different cluster sizes. 77
4.5 Average runtimes (number of timesteps) vs median runtimes (number of timesteps) for different cluster sizes. 77

List of Figures

1.1 The CHARMM36 potential energy function. 7

2.1 Kv channel 14
2.2 Molecular representation of the atomic model of the Kv1.2 chimera channel embedded in an explicit POPC membrane bathed by a 0.15M KCl aqueous salt solution. 21
2.3 System total energy during the equilibration. 22
2.4 System volume during the equilibration. 23
2.5 System temperature during the equilibration. 24
2.6 RMSD of protein backbone atoms during the equilibration. 25
2.7 RMSD of each residue at the end of equilibration. 26
2.8 2D PMF of ion translocation 28
2.9 The biased potential and the 2D PMF of ion translocation on the biased energy surface 29

3.1 The Lennard-Jones potential 34
3.2 Global minima of LJ clusters 35

4.1 VR app interface. 54
4.2 VR app interface - control pannel. 55
4.3 Tool tips for the Daydream controller. 57
4.4 VR app interface — color change of the cluster. 59
4.5 VR app interface — the end of a simulation. 60
4.6 Acceptance ratio with different α for different cluster sizes. 62
4.7 The effect of parameter n on the efficiency of the algorithm. 64
4.8 n_{opt} as a function of cluster size. 65
4.9 Comparison between the data collected by c++ implementation and unity VR implementation of Algorithm 1. 66
4.10 Median finish time (elapsed time) for different cluster sizes with purely computational method. 68
4.11 The plot of algorithm runtimes (number of timesteps) vs cluster sizes. The error bars represent the standard deviation. 71
4.12 The effect of the human involvement changes with the cluster size. . . 74
4.13 The performance data for purely computational method. 75
4.14 The performance data for hybrid method. 76

4.15 γ as a function of cluster size (n). 78

Part I

Ion Channel Conductance with Hyperdynamics

Chapter 1

Introduction

1.1 Molecular Dynamics

Molecular dynamics (MD) is a simulation technique that enables the study of microscopics interaction between atoms and molecules to help us understand the underlying mechanisms for interesting macroscopic phenomena such as protein folding. Compared to the other major family of classical simulation techniques, Monte Carlo (MC), MD has the advantage of being able to determine both the thermodynamic and dynamic properties of the simulated system. It does so by producing the trajectories of atoms and molecules using Newtonian mechanics.

Newton's second law states that

$$\overrightarrow{F}(\overrightarrow{x}) = m\overrightarrow{a}, \tag{1.1}$$

where $\overrightarrow{F}$ is the force acting on an object, m is the mass of the object and $\overrightarrow{a}$ is the acceleration the object gains, which can also be written as the second derivative of position $\overrightarrow{x}$ with respect to time t, $\frac{d^2\overrightarrow{x}}{dt^2}$.

The force acting on an object can be calculated using its potential energy funtion($V(\overrightarrow{x})$) by the following equation

$$\overrightarrow{F}(\overrightarrow{x}) = -\overrightarrow{\nabla} V(\overrightarrow{x}). \tag{1.2}$$

Combining those two equations, we get the following equation of motion

$$m\frac{d^2\overrightarrow{x}}{dt^2} = -\overrightarrow{\nabla} V(\overrightarrow{x}). \tag{1.3}$$

To model the physical movement of a 3-dimensional N-particle assembly, we need to solve $3N$ coupled 2^{nd}-order differential equations. When N gets large, the anaytical solution is very hard if not impossible to determine. The numerical solution, however, is straightforward.

There are many algorithms to solve the problem numerically. Velocity Verlet [3] is one of the most popular numerical integrators. It is time reversible, symplectic and has a global error of order two. The algorithm takes the initial positions and velocities as the input and calculates the initial accelerations of the particles. It then repeatly performs the following steps:

1. $\overrightarrow{x}(t + \Delta t) = \overrightarrow{v}(t) + \frac{1}{2}\overrightarrow{a}(t)\Delta t$
2. calculate $\overrightarrow{a}$(t + Δt) using equation 1.3 and updated positions
3. $\overrightarrow{v}(t + \Delta t) = \overrightarrow{v}(t) + \frac{1}{2}(\overrightarrow{a}(t) + \overrightarrow{a}(t + \Delta t))\Delta t$

The timestep Δt used by the integrator is a significant parameter in MD simulations. A large timestep is desirable, in that we can model events on a larger time scale with a reasonable running time. However, the larger the timestep we use, the greater the error in the trajectory becomes. In practice, the timestep is

usually chosen to be roughly one tenth of the timescale of the fastest motion in the system. For certain systems where the event we are interested in occurs infrequently and quickly, a variable timestep can be used to obtain longer simulation without sacrificing accuracy of the result.

The trajectory data generated by MD ($\{x(t)\}, \{v(t)\}$) is useful for both kinetic and thermodynamic study. It can be treated as a time series and allows correlations and rates of the transition between states to be determined. We can also discard the time evolution information and treat each data point in the trajectory as a Boltzmann sampling result of the equilibrium system to determine the thermodynamic properties, assuming our system is ergodic. The ergodic hypothesis states that a system spends its time in all accessible microstates with a probability proportional to $e^{-\beta E}$, where E is the energy of the microstate, over a long period of time. Thus, the time average of a ergodic system is equivalent to its ensemble average. A thermodynamic property M can be calculted by the following equation:

$$< M >= \frac{1}{T} \sum_{i=0}^{N} M(t_i) \Delta t_i, \tag{1.4}$$

where Δt is the time step size used by the integrator, N is the number of iterations the integrator performs and $T = \sum_{i=0}^{N} \Delta t_i$ is the total simulation time of the MD simulation.

Basic MD simulations conserve the total energy of the system. Sometimes, to match the existing experimental conditions, we would like the system to main a constant temperature (T) and/or pressure (P). A thermodstat or a barostat is then needed. There are generally three types of methods to control the T and P: Ad hoc methods (velocity rescaling, berendsen [4]...) are the fastest and simplest. They control T and P by directly adjusting the system velocities and volume to or toward

a precomputed desired value. Such methods do not preserve the correct distribution of the velocities and volume. The most representative of stochastic methods is the Langevin thermostat and barostat, which model the surroundings as Brownian solvent and control the T and P using random forces [5]. The resulting Langevin dynamics extend basic MD with the following equation of motion:

$$m\frac{d^2\overrightarrow{x}}{dt^2} = -\overrightarrow{\nabla}V(\overrightarrow{x}) - m\gamma\frac{d\overrightarrow{x}}{dt} + \overrightarrow{R}, \tag{1.5}$$

where γ is the friction coefficient characterizing the Brownian solvent and $\overrightarrow{R}$ is the random Brownian force. A typical extended-system method is the Nose-Hoover thermostat [6, 7], which couples an large external heat bath to the system and controls the system T with heat transfer between the system and the heat bath.

1.2 The CHARMM Force Field

The quality of a computational simulation largely depends on the quality of the potential energy function used. There are many existing force fields (the collections of the potential energy function and all the parameters in it) based on different experimental data and quantum chemistry calculations at varying levels of theory. The most widely used ones for biomolecular systems include CHARMM [8, 9], AMBER [10] and OPLS [11].

In this study, we used the CHARMM36 force field [9]. The potential energy

function includes the following components:

$$V_{\text{bond}} = \sum_{\text{bonds}} k_b(b - b_0)^2 \tag{1.6}$$

$$V_{\text{angle}} = \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 \tag{1.7}$$

$$V_{\text{Urey-Bradley}} = \sum_{\text{Urey-Bradley}} k_u(u - u_0)^2 \tag{1.8}$$

$$V_{\text{dihedral}} = \sum_{\text{dihedrals}} k_\phi[1 + \cos(n\chi - \delta)] \tag{1.9}$$

$$V_{\text{improper}} = \sum_{\text{impropers}} k_\psi(\psi - \psi_0)^2 \tag{1.10}$$

$$V_{\text{vdW}} = \sum_{\text{nonbonded}} \epsilon[(\frac{R_{min_{ij}}}{r_{ij}})^{12} - (\frac{R_{min_{ij}}}{r_{ij}})^6] \tag{1.11}$$

$$V_{\text{electrostatic}} = \sum_{\text{nonbonded}} \frac{q_i q_j}{\epsilon r_{ij}} \tag{1.12}$$

$$V_{\text{CAMP}} = f(\chi_1, \chi_2) \tag{1.13}$$

The first five terms account for the internal interactions between bonded atoms. More specifically, V_{bond} describes the bond stretching between 2 bonded atoms. V_{angle} and $V_{\text{Urey-Bradley}}$ describe the bond bending between 3 bonded atoms. V_{dihedral} and V_{improper} describe dihedral rotation and out of plane bending between 4 bonded atoms respectively. The external interaction between nonbonded atoms are represented by the next two terms. V_{vdW} describes the distance-dependent van der Waals attraction and repulsion between a pair of nonbonded atoms. $V_{\text{electrostatic}}$ describes the electrostatic interaction between two charged nonbonded atoms. The last term, V_{CAMP} accounts for the correlation of the central two dihedral angles χ_1 and χ_2 in a dipeptide allowing more realistic protein backbone conformations. Figure 1.1 gives a graphic illustration of these terms.

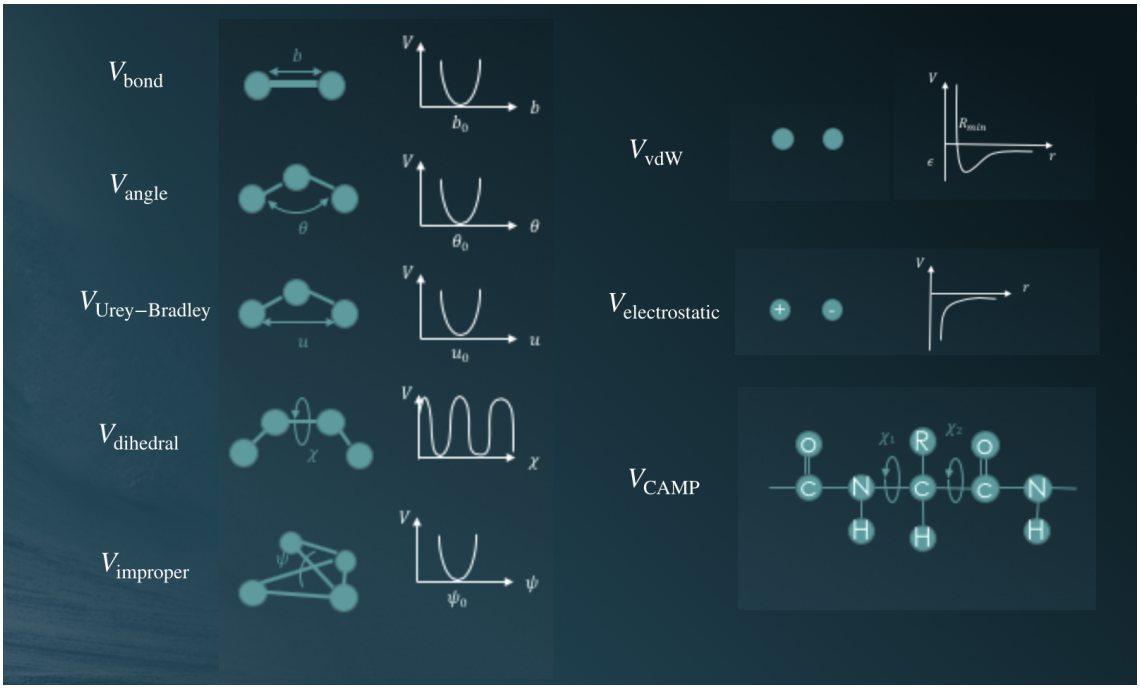


Figure 1.1: The CHARMM36 potential energy function.

1.3 Hyperdynamics

As mentioned in section 1.1, the timestep, Δt, is usually chosen to be small enough to cover the fastest the motion in the system to ensure numerical stability. The vibration of covalent bonds with hydrogen is the fastest motion in most biological and chemical systems and happens on timescales of approximately 10-20 femtoseconds. Thus a timestep of 1 fs is necessary to maintain integrator accuracy. As a result, with feasible computational resources, MD simulations rarely exceeds 10^9 steps or a few micoseconds of physical time [12].

However, many events of interest occur on a much larger timescale. For example, protein folding happens on timescales of microseconds to milliseconds [13]. Dissociation of a weak acid in water takes a few milliseconds. Vapor-deposited film growth takes place over a timescale of seconds. To study such events, at least one time

series with desired time length needs to be obtained. Furthmore, for the observation to be statistically meaningful, multiple events need to be sampled.

Many efforts have been put forth in addressing this limitation. The SHAKE [14] (for Verlet algorithm)/RATTLE [15] (for velocity Verlet) methods can be applied to constrain bond lengths and thus remove the fastest motion, bond vibration of covalent bonds involving hydrogen, in the system allowing a larger timestep to be used.

Transition State Theory (TST) [16,17] takes a different direction in bypassing the problem. For many molecular processes, the dynamic bottleneck is the rare event of crossing a high energy barrier, relative to the thermal energy, between two potential basins. Instead of sampling a single trajectory long enough for this rare event to happen, which is very likely to exceed the pratical MD timescale, TST treats the transition rate as an equilibrium property and calculates it using a two-step procedure [18]. The system is first moved from the reactant state reversibly to the transition state surface to determine probability for reaching the transition state. Then many short trajectories are initated at the transition state surface and the probability of actually passing over the transition state surface is determined by the fraction of the trajectories that directly go to the product state. Combining these, the TST rate can be obtained. Mathematically, the TST rate for eascaping a state A can be expressed with the following formula:

$$k_{A\rightarrow}^{TST} = \langle |v_A| \delta_A(\overrightarrow{r}) \rangle_A, \tag{1.14}$$

where v_A is the velocity normal to the transition surfaces that accounts for the probability of the actual crossing and $\delta_A(r)$ is the Dirac delta function that accounts for the probability of the system being at the transition surface [19]. Even though theoreti-

cally sound, in practice, TST is not easily applied to many systems because the states of the system are often unknown and the transition surface between those states is hard to characterize.

Hyperdynamics [19–22] is a TST-based method to extend the time scale of MD but without having the knowledge of system states or transition surfaces ahead of time. It is built on TST's basic assumption that the TST rate is an equilibrium property of the system. It also incorporates the idea of importance sampling by introducing a bias potential ($\Delta V_b(\overrightarrow{r})$) to the original system. The bias potential needs to be zero on the dividing surface and nonnegative elsewhere. With some manipulations of equation 1.14, the TST rate on the biased potential can be expressed as the following:

$$k^{TST}_{A_b\rightarrow} = k^{TST}_{A\rightarrow}\langle e^{\beta\Delta V_b(\overrightarrow{r})}\rangle_{A_b}, \tag{1.15}$$

where A_b represents the biased state A. Since the bias potential is nonnegative, term $\langle e^{\beta\Delta V_b(\overrightarrow{r})}\rangle_{A_b}$ must be larger than or equal to 1, thus $k^{TST}_{A_b\rightarrow} \geq k^{TST}_{A\rightarrow}$. We get faster dynamics with the biased potential. Furthermore, if we compute the relative rates of escaping from state A to its adjacent states B and C (assuming the bias potenital does not remove those basins), we get an important property of hyperdynamics:

$$\frac{k^{TST}_{A_b\rightarrow B}}{k^{TST}_{A_b\rightarrow C}} = \frac{k^{TST}_{A\rightarrow B}}{k^{TST}_{A\rightarrow C}}, \tag{1.16}$$

which states that with the biased potential, the probability of the system evolving from one state to another is the same as that for the unbiased system. So not only do we get acclerated dynamics, we also get correct ordering of state-to-state dynamics. The average boost factor is defined by the ratio of the total time the system has

envolved and the total time of the MD simulation:

$$\frac{t}{t_{MD}} = \langle e^{\beta \Delta V_b(\vec{r})} \rangle = \frac{1}{N} \sum_{i=1}^{N} e^{\beta \Delta V_b[\vec{r}(t_i)]}, \tag{1.17}$$

where N is the total number of MD steps and t_i is the time at the ith MD step. The overall computational speedup for hyperdynamics is the average boost factor offset by the extra cost of evaluating the bias potential.

1.4 Potential of Mean Force

The potential of mean force (PMF) [23], the free energy profile along a specific reaction coordinate, can be used to study various complex biological processes such as ion permeation through ion channels and enzyme catalysis.

There are a few methods that can be used to calculate the PMF. Popular and widely used ones include thermodynmaic integration [24], free energy perturbation [25], force constraint [26] and umbrella sampling [27] with weighted histogram analysis method (WHAM) [28]. The last one is used in this study and explained in detail in the following paragraphs.

Based on statistical mechanics, the free energy in the canonical ensemble along a chosen reaction coordinate $\xi = \xi(\mathbf{q})$ (assuming a geometrical reaction coordinate)/PMF can be determined by the following equation:

$$A(\xi) = -k_B T \ln P(\xi) + C, \tag{1.18}$$

where C is a constant and $P(\xi)$ is the probability distribution. $P(\xi_i)d\xi$ at a particular value of the reaction coordinate ξ_i can be approximated by the fraction of data points generated by MD simulations in which the system has $\xi \in [\xi_i, \xi_i + d\xi]$. A issue of

practical importance, is that due to high energy barriers in the energy landscape and finite simulation time, the system is likely to be stuck in some basins leaving the rest of the configuration space poorly sampled or completely unsampled. This will cause great statistical errors for the probability estimation and in turn, affect the PMF calculation. Umbrella sampling addresses the issue by adding a harmonic biasing funciton

$$V_i(\xi) = \frac{1}{2}k(\xi - \xi_i)^2 \tag{1.19}$$

to the original energy surface. The system is restrained to sample ξ_i and its neighborhood allowing a statistically significant estimation of the biased probability at that region. The original PMF can be produced with the biased probability through

$$A_i(\xi) = -k_B T \ln P'(\xi_i) - V_i(\xi) + C_i, \tag{1.20}$$

where C_i is a constant depending on V_i. To ensure sufficient sampling along the whole reaction coordinate, a series of simulations each with a harmonic potential added at different ξ_i values can be performed. The results from each simulation ($\{A_i(\xi)\}$) are then combined by WHAM, which adjusts the C_i to minimize the difference between the individual distributions in their overlapping regions, to give the final PMF ($A(\xi)$).

Chapter 2

Ion channel conductance with hyperdynamics

2.1 Introduction

Voltage-gated potassium (Kv) ion channels play an essential role in the generation and propagation of electrical signals in the nervous system. As the name suggests, they open/close in response to changes in the transmembrane potential. Upon activation, these channels allow rapid and selective passive flow of potassium ions from the intracellular space to extracellular space to repolarize the action potential. Potassium ions flow at a rate of approximately 10^8 ions per second in a concentration gradient of 140mM to 5mM, from intracellular to extracellular space, respectively.

Mutations in Kv channels have been found to be responsible for many diseases such as episodic ataxia with myokymia syndrome, and long QT syndrome [29]. Such mutations may either cause failure in producing functional channels (ex. Arg174Cys, Glu261Lys in KCNQ1) or alter the channel kinetics so as to reduce conductance (ex.

Leu272Phe, Ala300Thr in KCNQ1), change selectivity (ex. Asn629Asp in HERG), shift the voltage-dependence of activation to a more positive or negative potential (ex. Arg243His, Trp258Arg in KCNQ1), or slow or accelerate activation or deactivation (ex. Arg243Cys, Arg243His in KCNQ1) [29].

The structure of Kv channels (Fig. 2.1) has 4 identical α subunits arranged symmetrically around a central pore inside the membrane. There are also intracellular β subunits that co-assemble with the α subunits to modulate the activity of the channel and stabilize the multimeric complex. Each α subunit is composed of six membrane-spanning hydrophobic α-helical sequences. The peripheral four helices from each subunit form the voltage sensor domain while the inner helices form the pore domain. The narrowest part of the pore (selective filter, SF) is close to the extracellular side and is responsible for the selectivity of the channel. It's composed of five amino acids (TVGYG) and is highly conserved among potassium channels. Those five amino acids form 5 binding sites (S0-S4), which can be occupied either by a potassium ion or a water molecule. Below the SF in the pore domain is the wide diffuse cavity which helps overcome the dielectric barrier caused by the membrane [30].

Because of the biological importance of potassium channels and the availability of high resolution crystal structures (PDB ID: 2A79, 2R9R) [31, 32], many computational studies have been performed to study the ion-binding sites and permeation pathways [33–35], ion conductance [34–38], selectivity [39–41] and channel gating [34, 42–44] on the atomistic level. Among all simulation methods, molecular dynamics (MD), in which one propagates the classical equations of motion forward in time, has been widely used because of its ability to study the dynamical properties of a system. However, since accurate integration requires time steps short enough ($\sim$fs) to resolve atomic vibrations, the timescale of a traditional all-atom MD simulation is typically limited to nanoseconds. As mentioned before, a typical ion permeation pro-

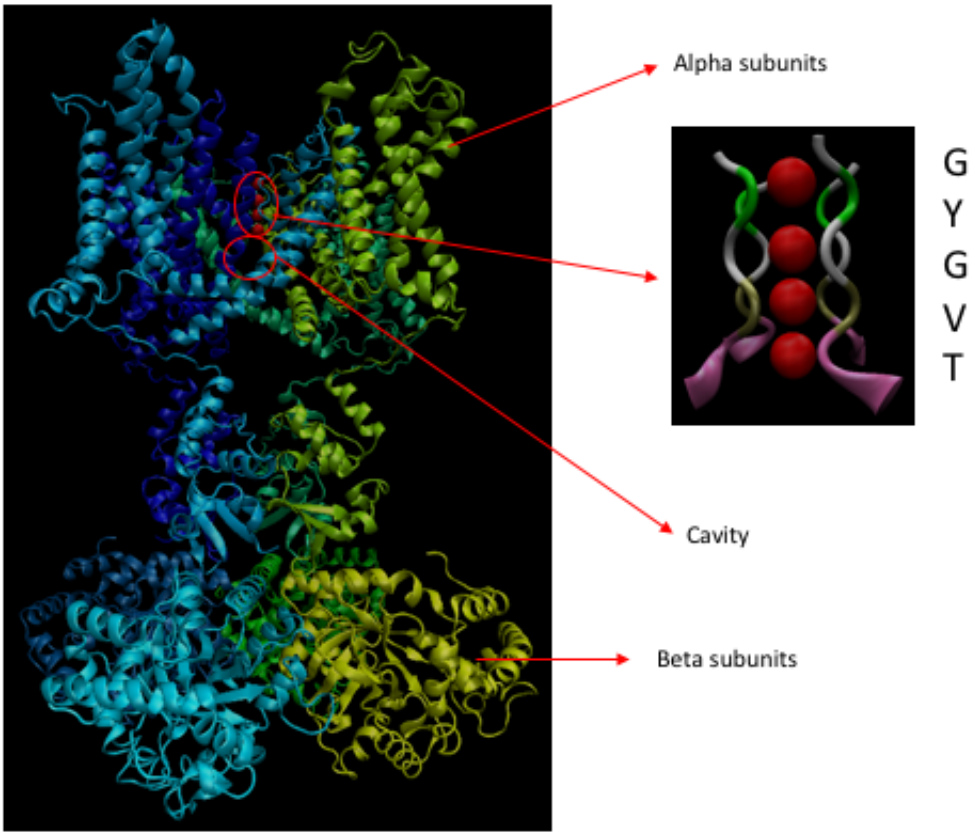


Figure 2.1: Kv channel (PDB ID: 2R9R)

cess under physiological conditions occurs on a timescale of nanoseconds. Thus, direct measurement of ion conductance, which would involve at least tens of ion permeation events to generate meaningful statistics, is quite challenging. This does not even consider gating events, which occur on a much longer timescale than ion conduction.

To study ion conductance with MD simulation, therefore, unphysically high voltages have been applied [37]. Though ions passing across the membrane can be observed this way, the dynamics might be distorted due to the unphysiological conditions used in these MD simulations [37, 38]. In another approach to address the timescale problem, a study has been done with a special machine designed to run millisecond-timescale MD simulations [35]. In that study, potassium ion concentration (0.6 M) at which saturation of conductance is reached was used instead of physiological concentration (0.15 M) to maximize ion permeation. Though enough ion permeations were observed to allow a measurement of the current, it was much lower than experimental data and the I–V curve vas not linear. Thus, more studies need to be done to fully understand ion conductance in Kv channels and a more general method to overcome the MD timescale limitation is desired.

Hyperdynamics is a powerful method to extend the time scale in MD simulations [19–22]. In the hyperdynamics approach, the potential energy surface of a system is modified by a bias potential, designed to raise the energy in regions other than at the dividing surface, and molecular dynamics are performed on this biased energy surface. Because the wells in the biased energy surface are not as deep as those in the unbiased one, when the system gets trapped, it escapes that state at an accelerated rate. And if the potential is not modified at the dividing surface, and if transition state theory (TST) is valid for the system and the biased potential, the system evolves from state to state in a sequence representative of the exact dynamics [19].

In practice, the difficulty of implementing hyperdynamics lies in building bias potentials that strictly satisfy the requirements (i.e. vanish at any transition state or dividing surface, and generate kinetics which should obey TST) while providing substantial acceleration of the dynamics. Several different types of bias potentials have been proposed since hyperdynamics was first introduced. The first was Voter's original Hessian-based method where the bias potential is positive for regions where the lowest eigenvalue of the Hessian is positive and zero elsewhere [19]. Another is the "bond-boost" method where the bias potential is determined by the deviation of the bond lengths of a specified set of atoms from their equilibrium values, and turned off if the distortion of any bond exceeds a predefined threshold [45]. In cases where suitable reaction coordinates are known, "collective variable-driven hyperdynamics" methods can be used, in which the bias potential depends only on a global collective variable calculated from local distortions of a set of local properties and is turned off if any local property is involved in a transition somewhere in the system [46]. When the dividing surfaces are understood, it is possible to use a "ridge-based" method, where the bias potential is a constant value if the system is far from the ridge and the total biased potential is set to the energy of the transition state if the system is near the ridge [47]. The difficulties in using many of these bias potentials are that they often require on some prior knowledge about the system, and thus are not suitable for general systems, and their effectiveness often decreases rapidly with dimensionality.

Generally speaking, then, hyperdynamics is not suitable for accelerating dynamics in complex biological systems. First, TST is often not a good approximation for reactions in solution. Second, the number of degrees of freedom required to model biological systems is typically very large compared to that needed to model solid state systems, because complex biological processes involve various types of nonbonded interactions, with a wide range of energy barriers. Building a proper bias potential for

a large system with a complex energy surface is quite challenging if possible at all.

However, the potassium ion channel is quite special. The pore structure is similar among different channels in the family and the selective filter structure is exactly conserved. The ion binding sites and permeation pathway have been extensively studied. And Hodgkin and Keynes's knock-on model [48] is widely accepted as the mechanism of ion permeation across the potassium channel with both experimental and computational supporting evidence. The rate-limiting step in the conduction process is usually assumed to be potassium ions passing through the SF in single file, which is further thought to be well described by a low-dimensional free energy profile or potential of mean force (PMF). Based on this prior knowledge, the idea comes that we can build a bias potential based on a a 2D PMF and perform MD on the resulting biased energy surface. If the bias potential is good enough and provided those widely held views about potassium channels match the true channel behavior, it should be possible to use this approach to speed up the ion permeation process and measure the channel conductance quantitatively. It should be emphasized that our method does not meet all of the conditions required by hyperdynamics, thus the dynamics we get might be distorted. However, since some features about the true energy surface are included in the bias potential, it is expected that the method should give a better approximation to the true dynamics than simply applying a constant strong electrical field.

2.2 Methods

2.2.1 System building and MD simulation

The membrane protein system was built mainly using the CHARMM-GUI tools [49]. The protein structure of the Kv1.2-Kv1.1 chimera (Kvchim, PDB ID: 2R9R) was obtained from the OPM (orientations of protein in membranes) [50] database. Only the pore and voltage sensor domains (resid id: 148-417) were kept. The protein was embedded in a 1-palmitoyl-2-oleoyl-sn-glycero-3-phosphocholine (POPC) bilayer. The whole system was solvated in 0.15 M KCl. Additional K^+ ions were used to keep system neutral. The final system has a volume of $113.243 \times 113.243 \times 86.701 \text{Å}^3$ large with a total of 100,335 atoms. Three potassium ions were initially placed in the SF, separated by single water molecules.

Molecular dynamics simulations were performed using NAMD2.10 [51]. The system first underwent energy minimization for 10 ps. Harmonic potentials with spring constants of 10 $\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^{-2}$ were first applied to the protein and lipid. The constraints were gradually released during the roughly 30 ns equilibration. A timestep of 1.0 fs was used in the beginning several stages of the equilibration and then was switched to 2.0 fs until the end of the equilibration. The temperature was held at 303.15 K using a Langevin thermostat with a damping coefficient of 1.0 ps^{-1}. The pressure was maintained at 1 atm using a Langevin piston barosat with an oscillation period of 50 fs and a damping time constant of 25 fs. Electrostatic interactions between charged atoms were calculated using the particle mesh Ewald method [52]. Van der Waals interactions were truncated at 12 Å with a switching function applied from 10 Å. RATTLE was used to constraint the length of all bonds involving a hydrogen atom. The root mean standard deviation (RMSD) from the initial structure was calculated using CHARMMc39 to confirm the system's equilibrium state.

2.2.2 Umbrella sampling and 2D PMF calculation

We followed the method of Fowler [53] which is a slightly simplified version of the method of Berneche and Roux [33] and make the following definition: The pore axis is parallel to the Z axis with the origin at the center of mass of the backbone atoms of the residues TVGY in the selective filter. Ions in the selective filter are labeled 1 to 3 in successive order starting from the outermost ion (i.e. the one closest to the extracellular end). The configuration of the selective filter is described by a point in the (Z_{12}, Z_3) space with Z_{12} corresponding to the center of mass of ions 1 and 2 along the pore axis and Z_3 corresponding to position of ion 3 along the pore axis.

For the umbrella sampling PMF calculations, a total of 182 independent simulations of 600 ps with a biasing harmonic potential centered on Z_{12} and Z_3 (varying successively from $-5.0\,\text{Å}$ to $-11.0\,\text{Å}$ and $4.0\,\text{Å}$ to $-2.5\,\text{Å}$, respectively, every $0.5\,\text{Å}$) were generated with a force constant of 20 $\text{kcal} \cdot \text{mol}^{-1} \cdot \text{Å}^{-2}$ using NAMD2.10. The first 200 ps of each simulation was considered as equilibration and was thus discarded. Any frames with two adjacent ions with no intervening water were discarded.

The umbrella sampling simulations were unbiased and merged using the weighted histogram analysis method (WHAM) [54] to calculate the two dimensional potential of mean force (PMF).

2.2.3 Hyperdynamics method

Based on the 2D PMF generated from umbrella sampling, three Gaussians were constructed in a way that raised the lowest-energy basins while affecting the transition state as little as possible. The three Gaussians were then used as the bias added to the system during molecular dynamics simulation. This was done using the

Tcl scripting interface provided by NAMD. With a user defined Tcl script, external forces were calculated from the bias and applied to involved atoms. Additionally, a constant electric field of 2.306 kcal/(mol $\cdot$ e), corresponding to a voltage of 100 mV (upper limit of voltages used in most experimental studies of voltage potassium channel conduction properties) across the simulation cell, was also applied.

2.3 Results and Discussion

MD simulations of the pore and voltage sensor domains were carried out. Fig. 2.2 shows a molecular representation of the simulation system.

Properties of the system (total energy, volume, temperature and protein backbone RMSD) were measured during the simulation to monitor equilibration. As shown in Figs. 2.3-2.6, the value of the each of these properties becomes stable, with fluctuations, by the end of the equilibration. Thus it's reasonable to assume the system reaches equilibrium. The RMSD of the backbone protein is relatively high. To ensure the protein structure is not distorted, the RMSD of each residue was measured at the end of the equilibration, as shown in Fig 2.7. The large RMSD is mainly due to residues with residue id around 200, which forms a highly flexible loop structure [31]. The RMSD of the remaining residues remains small. So we assume the protein structure is physiological.

After equilibration, a 10 ns MD simulation was performed with constant external electric field corresponding to 100 mV across the simulation box and no ion-crossing events were observed.

In order to build a proper bias potential, an equilibrium PMF was first calculated using umbrella sampling. Fig. 2.8 shows the resulting 2D PMF. As discussed in section 2.2.2, a two-coordinate collective variable (Z_{12}, Z_3) was used to follow the

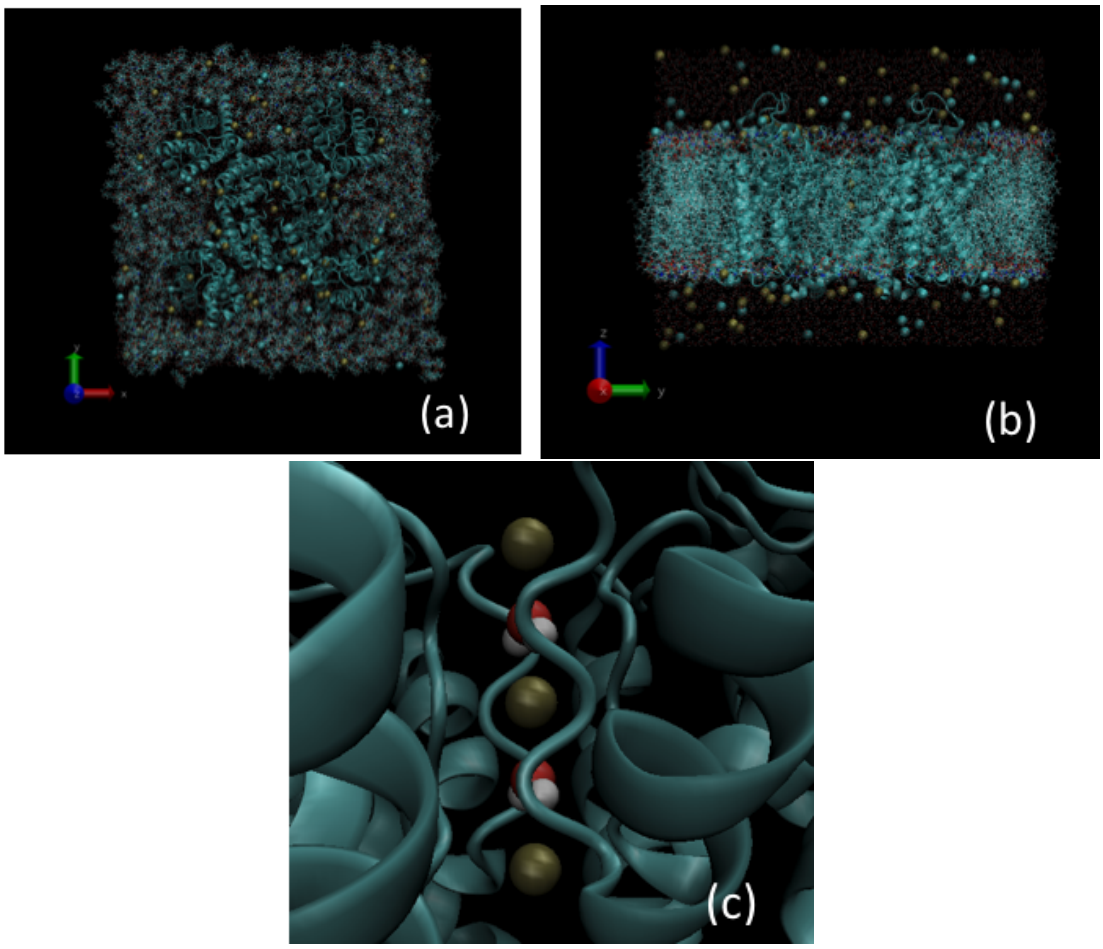


Figure 2.2: Molecular representation of the atomic model of the Kv1.2 chimera channel embedded in an explicit POPC membrane bathed by a 0.15M KCl aqueous salt solution. (a) Top view. (b) Side view. (c) Initial configuration of the selective filter.

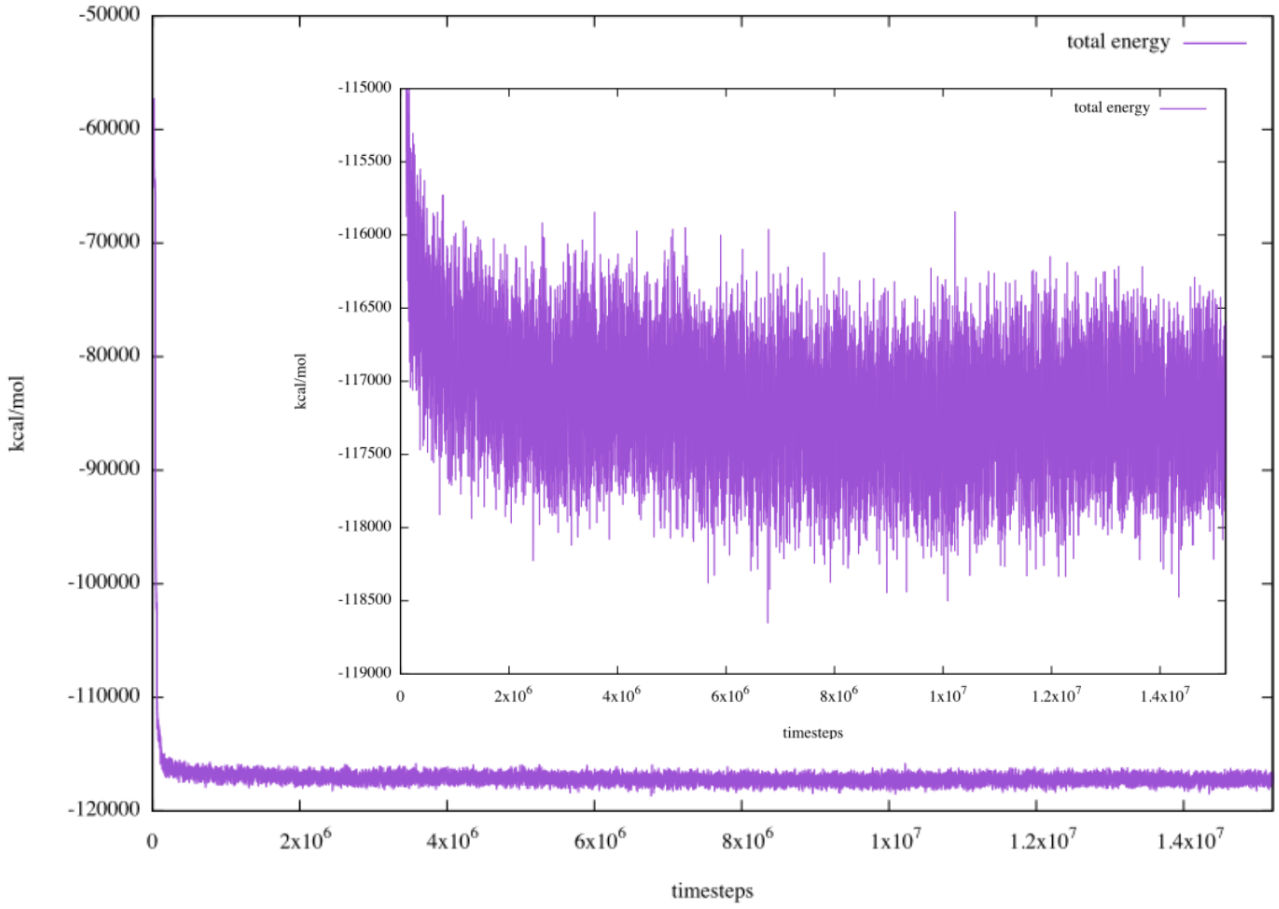


Figure 2.3: System total energy during the equilibration.

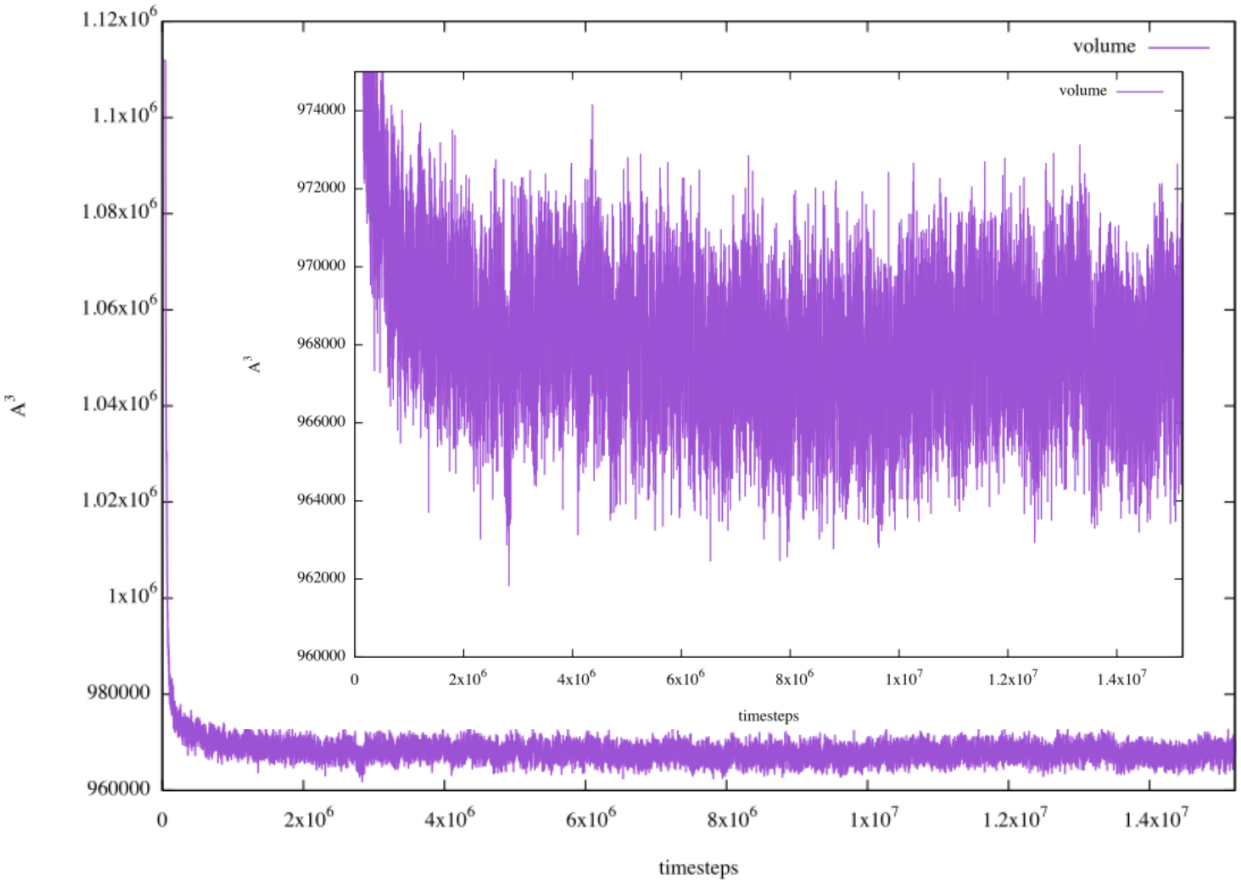


Figure 2.4: System volume during the equilibration.

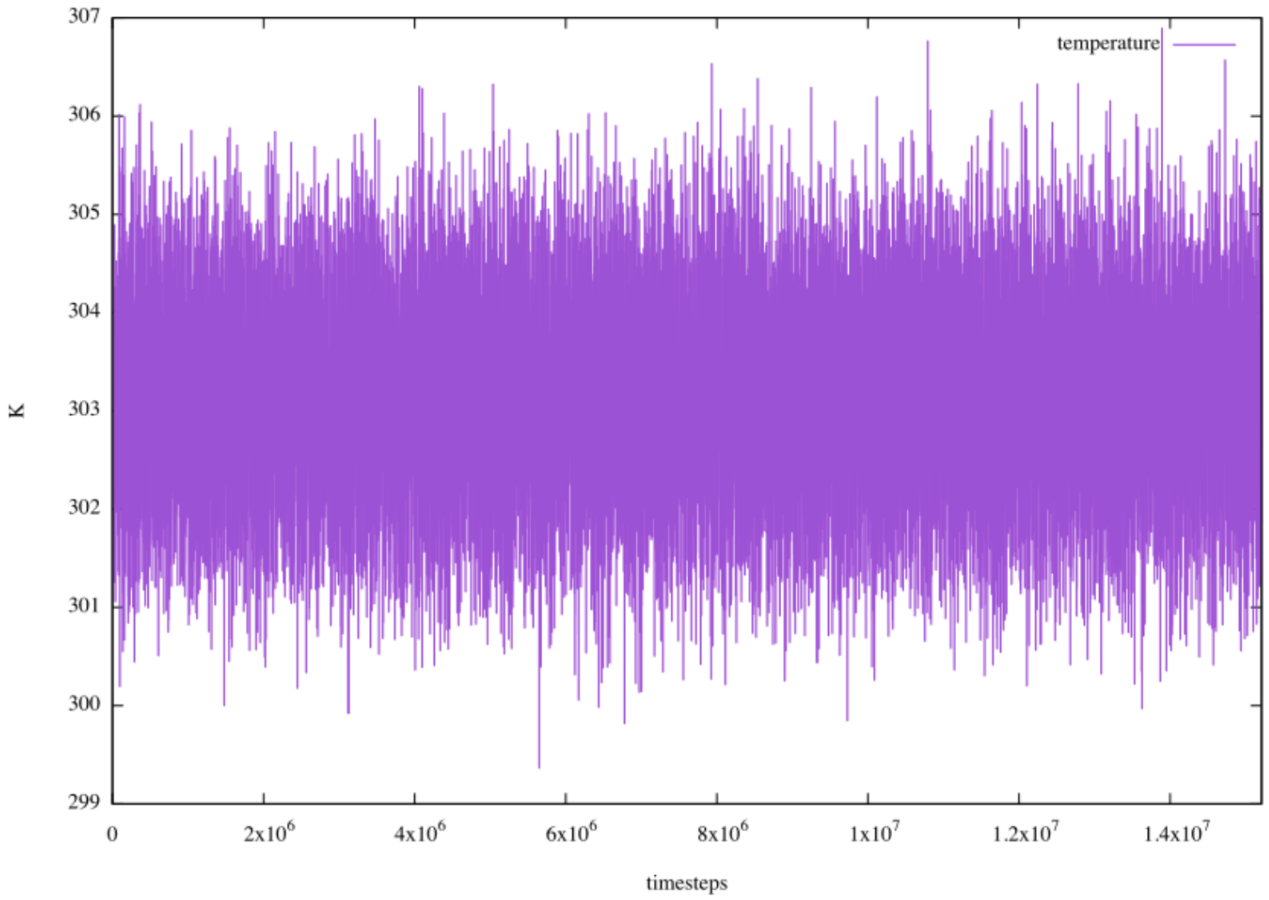


Figure 2.5: System temperature during the equilibration.

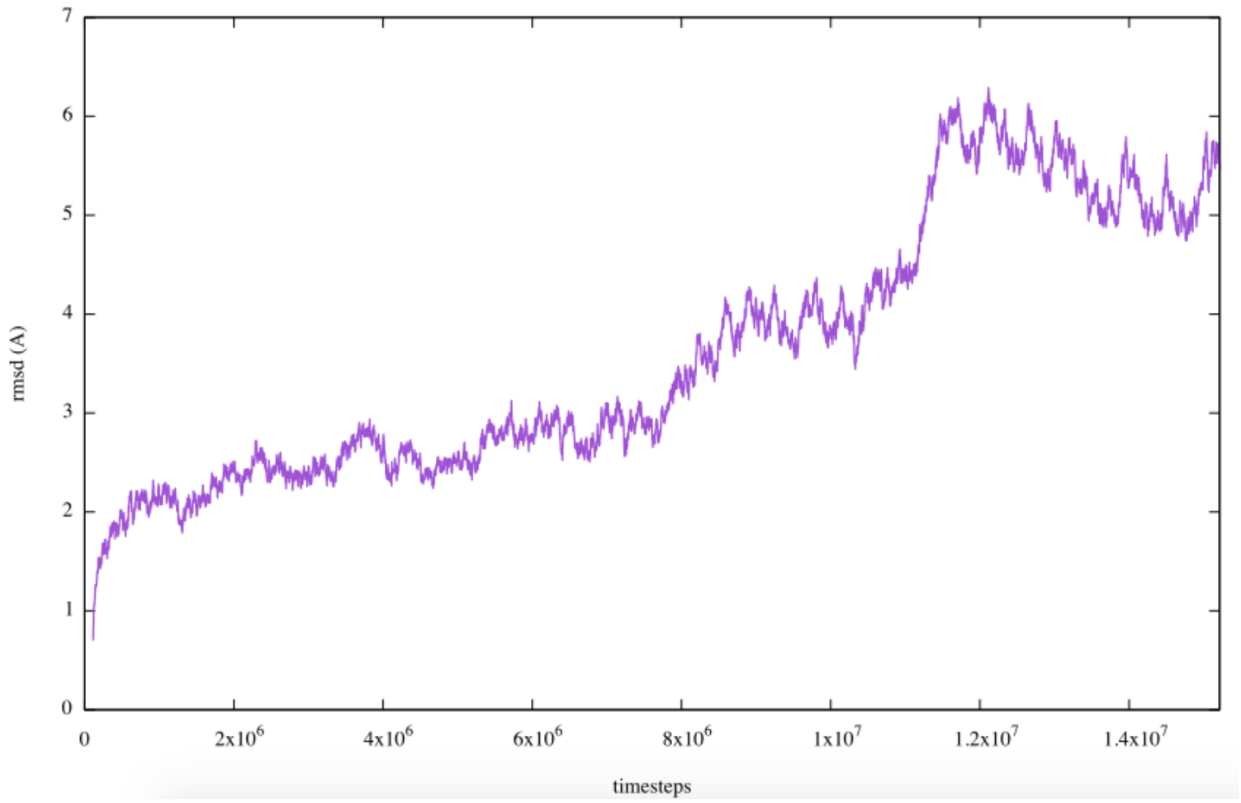


Figure 2.6: RMSD of protein backbone atoms during the equilibration. The reference structure used is the structure at the beginning of the equilibration.

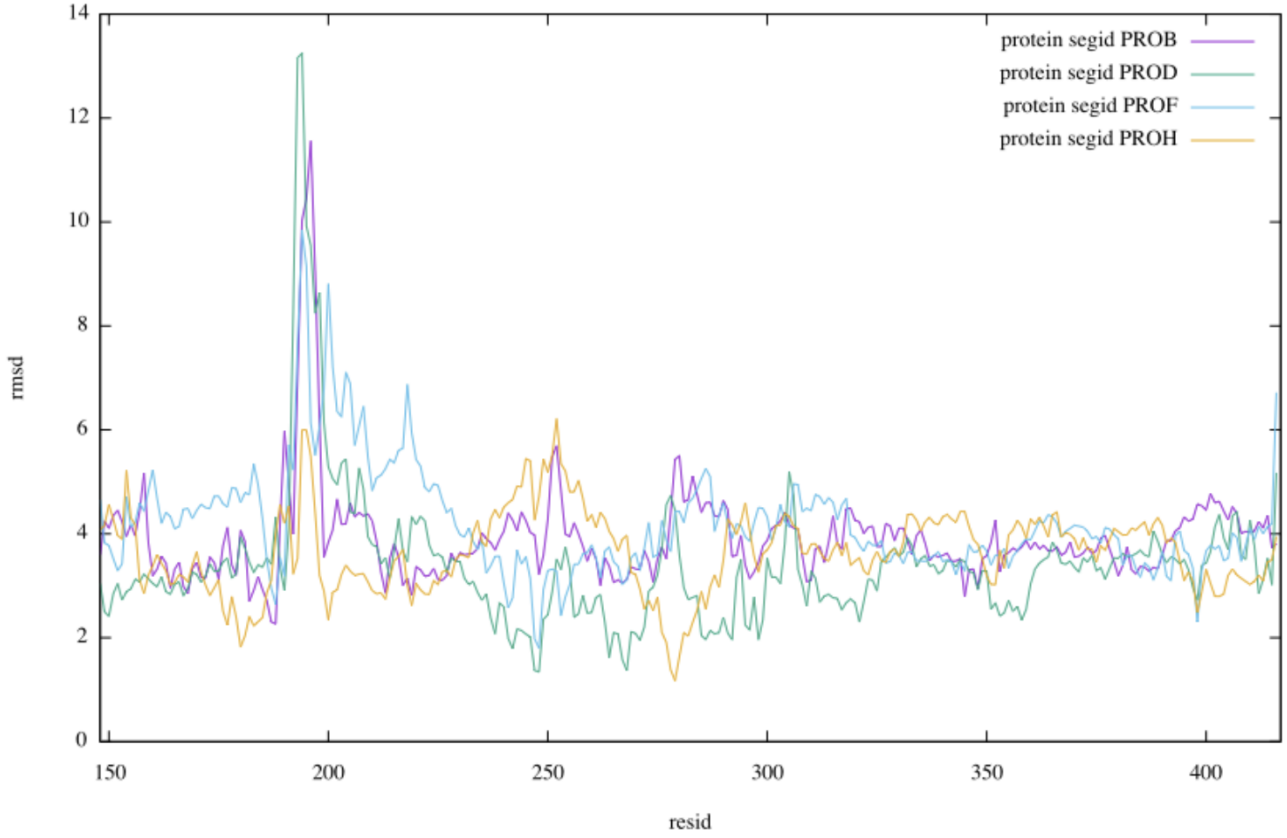


Figure 2.7: RMSD of each residue at the end of equilibration. The reference structure used is the structure at the beginning of the equilibration.

Gaussian	A (kcal/mol)	μ_x (Å)	μ_y (Å)	σ_x (Å)	σ_y (Å)
1	4.0	3.85	-5.45	0.4	0.8
2	6.0	3.6	-6.5	0.5	2.0
3	3.5	0	-8	0.8	0.4

Table 2.1: Values for parameters in equation 2.1

motion of the three potassium ions in the SF. Snapshots corresponding to several important configurations are provided (Fig 2.8(b)). The roughly estimated pathway for conduction process, 1–2–3–4, is consistent with the well accepted "knock-on" mechanism [48]. An ion in the cavity approaches the intracellular end of the SF and pushes two ions in the SF to move to the extracellular end.

Based on the 2D PMF, a bias potential composed of three Gaussians was proposed (Fig 2.9(a)):

$$\begin{aligned} f(Z_{12}, Z_3) = \quad & A_1 e^{(-\frac{(Z_{12}-\mu_{x_1})^2}{2\sigma_{x_1}^2} - \frac{(Z_3-\mu_{y_1})^2}{2\sigma_{y_1}^2})} + A_2 e^{(-\frac{(Z_{12}-\mu_{x_2})^2}{2\sigma_{x_2}^2} - \frac{(Z_3-\mu_{y_2})^2}{2\sigma_{y_2}^2})} \\ & + A_3 e^{(-\frac{(Z_{12}-\mu_{x_3})^2}{2\sigma_{x_3}^2} - \frac{(Z_3-\mu_{y_3})^2}{2\sigma_{y_3}^2})} \end{aligned} \tag{2.1}$$

where A, μ and σ values for three Gaussians are shown in Table 2.1.

Gaussians were optimized by overlapping the potential and PMF and examining the resulting PMF+potential graph by eye so that transition states were modified as little as possible while energy barriers were still reduced. Then umbrella sampling MD simulations were performed and an equilibrium 2D PMF were calculated on the biased potential to evaluate the quality of the bias potential.

As shown in Fig 2.9(b), with the bias potential, large energy barriers are removed and shape of the original energy surface is to some extent preserved. In particular, the barrier for the $3 \rightarrow 4$ transition is reduced from ≈ 5 kcal/mol to ≈ 2 kcal/mol.

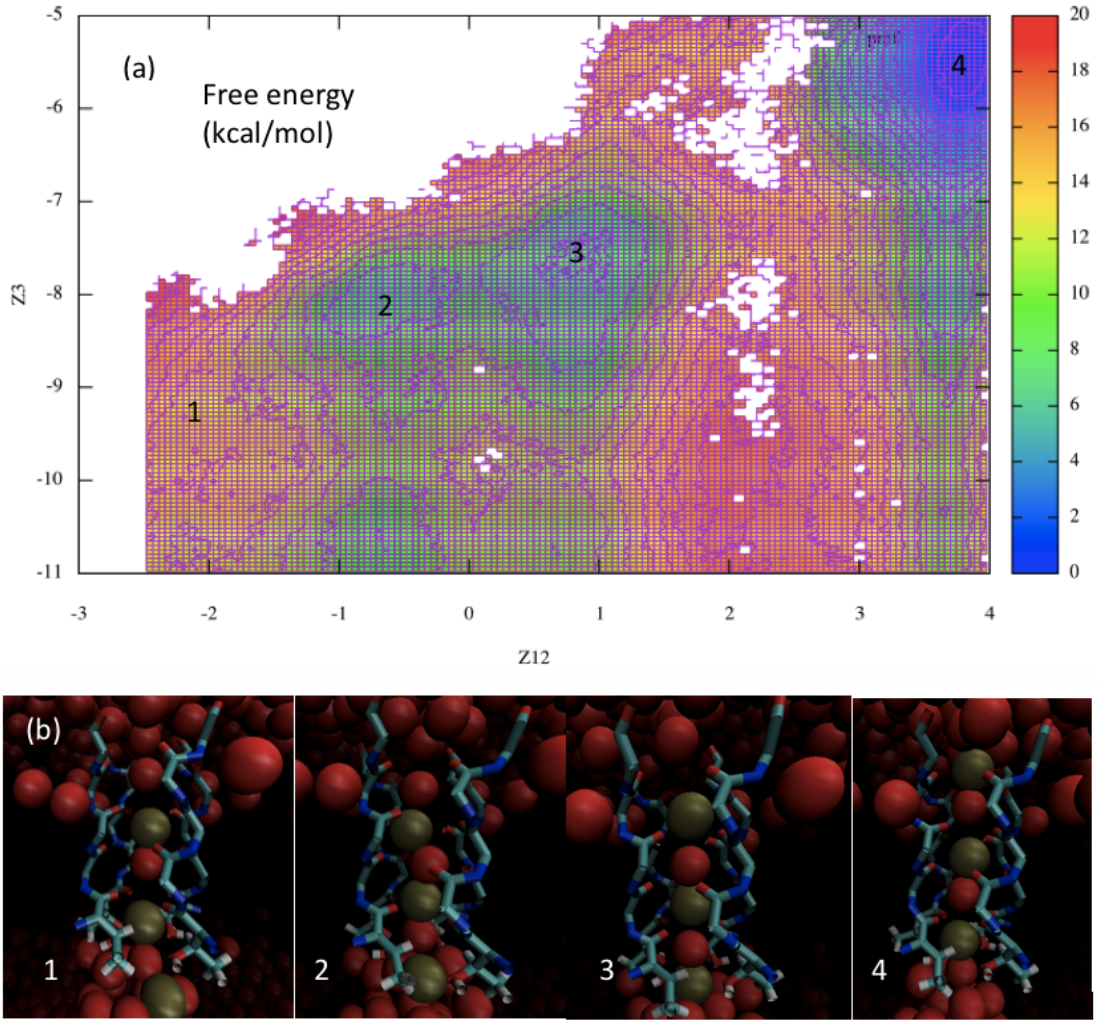


Figure 2.8: (a) 2D PMF of potassium ion translocation through SF of Kvchim channel calculated from umbrella sampling MD simulation. Z_{12} is the center of mass of the two potassium ions closest to the periplasm and Z_3 is the position of the third potassium ion. (b) Snapshots of the SF configurations corresponding to the points labeled in (a).

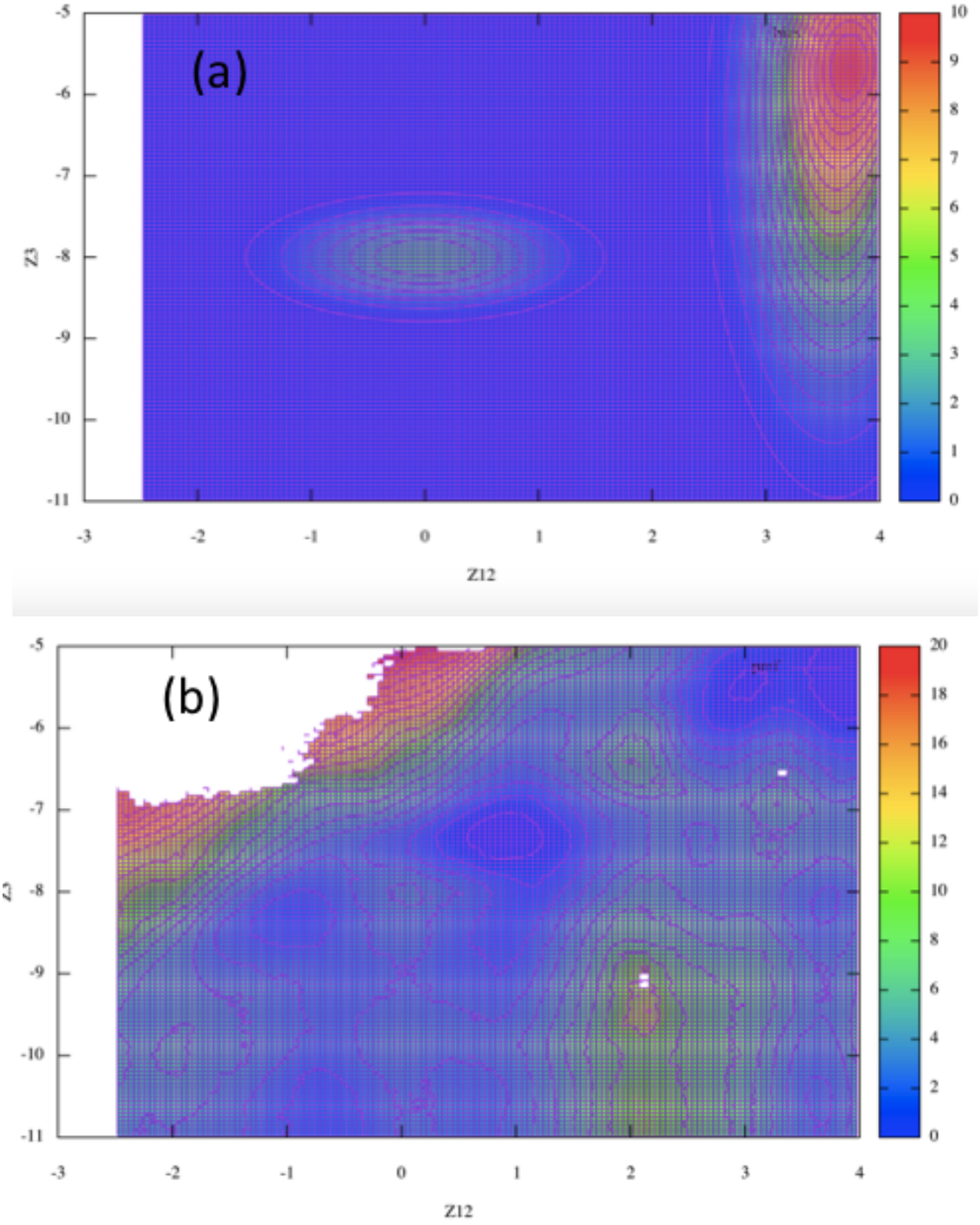


Figure 2.9: (a) bias potential for hyperdynamics (kcal/mol) (b) 2D PMF of potassium ions translocation through SF of Kvchim channel calculated from umbrella sampling MD simulation on biased energy surface (kcal/mol).

Hyperdynamics on the proposed biased energy surface with constant electric field corresponding to 100 mV across the simulation box were carried out. However, after 10 ns (MD time) simulation, still no ion crossing event was observed. It was also observed that no ions entered the cavity, from which a "knock-on" event could be initiated.

2.4 Conclusions

In this project, we applied the hyperdynamics method to extend the timescale of traditional MD simulations in an attempt to study the conductance properties of a potassium ion channel. A bias potential was built based on a 2D equilibrium PMF of potassium ions translocation through the SF of the channel. However, despite the acceleration, no ion crossing process has been observed.

An important observation during the simulation is that no ion entered the cavity. This may indicate that there is a substantial energy or entropy barrier for ions to enter the cavity, thus limiting the ion conductance rate. It is generally believed that ions passing through the SF is the rate limiting step. Thus, in the current study, the bias potential was applied only to the SF. However, it is possible that the barrier for ions entering the cavity is substantial enough that, after reducing the barriers in the SF, population of the cavity is the new rate-limiting step in the biased simulations, preventing ion crossing events on the ns timescale.

Previous studies have used either a bulk potassium ion concentration at which saturation of conductance is reached or a non-physiological electric field to maximize the current through the channel. It is worth noting that both of those methods would increase the rate of transitions into the cavity, as well as through the SF. Under high potassium concentration, for example, the cavity is typically filled with

several potassium ions. The importance of the barriers to ions entering the cavity has thus not been noticed before this work, and is worthy of further investigation.

In terms of future work, here are some ideas to explore. 1. Since in vivo, the movement of potassium ions is driven by the concentration difference between the intracellular environment and the extracellular environment, performing MD simulations under asymmetric ionic concentrations could potentially solve the problem mentioned above, i.e. few ions entered the cavity during the simulation. 2. One problem with applying hyperdynamics simulation to a complicated membrane protein in solution phase is that the actual potential surface is very complicated, making it very difficult to construct a bias potential that has no transition state modification and large speed up. In this work, we assumed the 2D PMF constructed using the methods mentioned above is a true representation of the real energy surface and built our bias potential based on the PMF. This assumption may be a potential cause of the failure of the hyperdynamic simulations. Instead, other accelerated dynamics simulation methods that are not dependent on the knowledge of the energy surfaces, for example, the parallel replica method, can be tried to study ion conductance of the potassium ion channel.

Part II

Human-Guided Global Optimization of Lennard-Jones Clusters

Chapter 3

Introduction

3.1 Lennard-Jones Cluster

A Lennard-Jones(LJ) cluster is a group of atoms in which the pair interaction between any two atoms is modeled by the LJ potential

$$V = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{3.1}$$

where ϵ and σ are atom type specific parameters, and r_{ij} is the distance between particle i and j. A graph representation is shown in Fig 3.1.

The LJ system has served as a testing ground for global optimization algorithm development due to its relatively simple mathematical form. It's also a widely used model for noble gas clusters. And its minimum energy geometry could provide some guidelines for the structure optimization of metal clusters such as nickel and gold [55]. For those reasons, the LJ clusters has attracted much attention and been studied intensely over the past years. Figure 3.2 shows the global minimum structure for some LJ clusters.

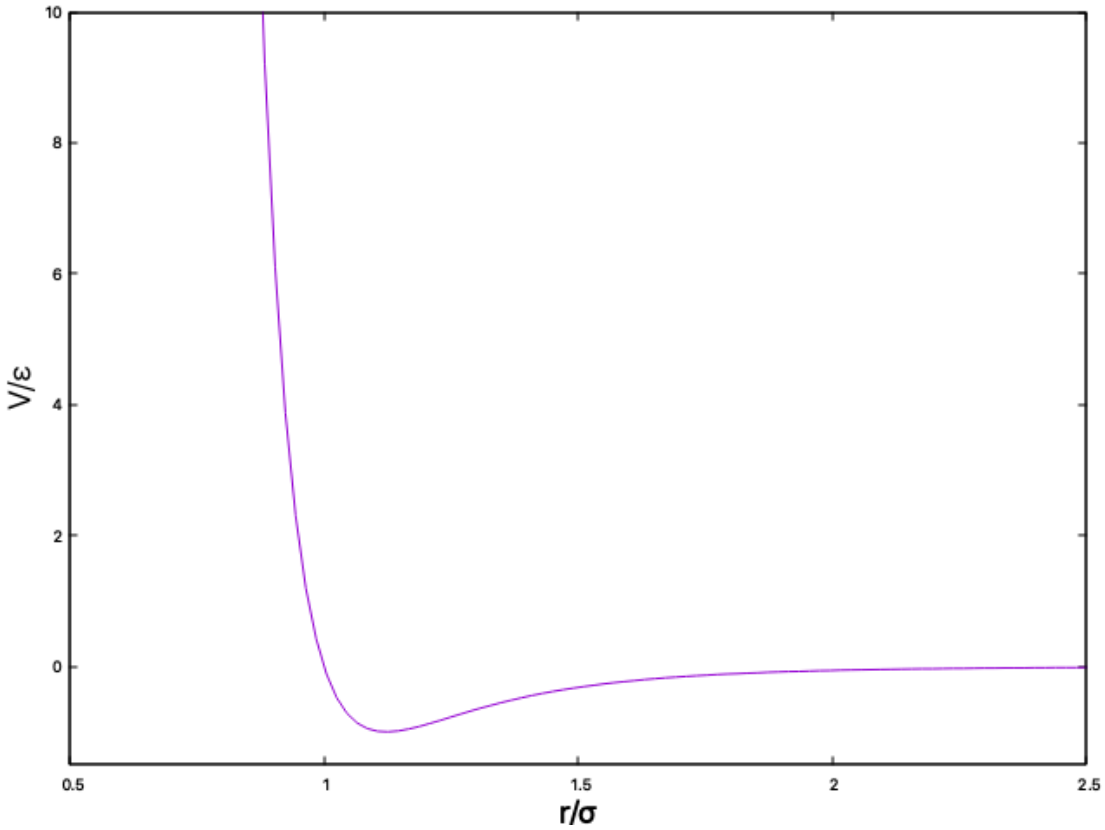


Figure 3.1: The Lennard-Jones potential for a pair of neural atoms.

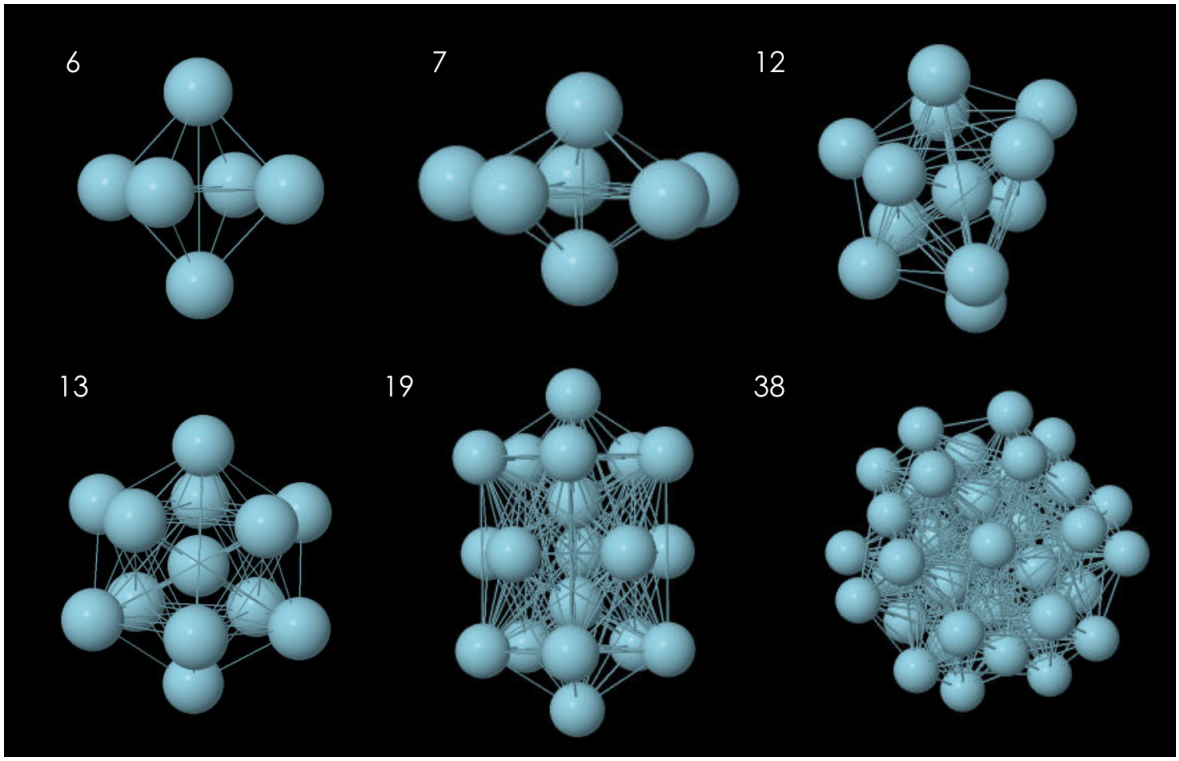


Figure 3.2: Energy structure of LJ clusters of size 6, 7, 12, 13, 19, 38.

.

3.2 Optimization Methods

Mathematically, the LJ cluster problem is

$$\underset{\mathbf{x}\in\mathbf{R}^n}{\operatorname{argmin}} \quad V_{LJ}(\mathbf{x}) = 4\epsilon \sum_{i=1}^{n} \sum_{j>i}^{n} \left[\left(\frac{\sigma}{\|\mathbf{x_i} - \mathbf{x_j}\|} \right)^{12} - \left(\frac{\sigma}{\|\mathbf{x_i} - \mathbf{x_j}\|} \right)^{6} \right]. \tag{3.2}$$

It's a continuous nonlinear optimzation problem with a nonconvex objective function. The problem is NP-hard, as proven by Wille and Vennik in 1985 using polynomial-time reductions [56]. It was shown by them that the traveling salesman problem is a special case of the heterogeneous LJ cluster optmization problem. Since the traveling salesman problem is a well known NP-hard problem, so is the heterogeneous LJ cluster problem. In computational complexity theory, the complexity class P contains all problems that are solvable in polynomial time with respect to the input size. The complexity class NP contains all problems that are solvable in polynomial time by nonderministic algorithms. A nondeterministic algorithm, is an algorithm that's able to track all possible paths simultaneously at any branching points, and return a solution as long as some paths are able to confirm it. Such alogrithms are purely theoretical. Equivalently, we can think of those problems in the NP class as: given a valid solution of the problem, such solution can be verifiable in polynomial time. An NP-hard problem is then defined as a problem such that if there is a polynomial-time alogrithm for the problem, all problems in NP class can be solvable in polynomial time (NP = P). An NP-hard problem is also an NP-complete problem if itself belongs to the NP class.

This section gives a overview of some major optimization methods in the context of this problem.

Deterministic global optimization methods are a group of methods that guar-

antee the global minimum to be found in finite time. To acheive this, the whole conformational space needs to be searched completely which is impossible to finish in finite time for continuous variables if doing naively. Branch and bound methods divides the whole search space into small subspaces. For each subspace, upper/lower bounds are estimated and compared to the existing solution to decide whether a better solution is likely to be found in the given region. The ones that are promising are subdivided further while the others are skipped without further search. Despite clever tricks, deterministic methods are still limited to small systems. With dimensionality increasing, they quickly become infeasible.

Simulated annealing [57] is one of the earliest methods used for global optimization problems. The system is equilibriated at some high temperature then gradually cooled down and eventually ends up in a local minimum of the potential energy surface. This minimum however, is not guaranteed to be the global minimum. To increase the chance of getting the correct result (the true global minimum), multiple simulations are usually attempted and the local minimum with the lowest energy is selected as the final result. For the LJ cluster problem, the simulated annealing methods haven't had much success except for the cluster with 24 atoms due to the problem that the free-energy global minimum can change at a temperature that the thermal energy is not enough for the system to escape its current local minimum [55].

Genetic algorithms [58] adopt the concept of the evolutionary theory. It starts with a population of cluster conformations, choosing randomly or/and with a priori knowledge. Each conformation is associated with a fitness value measuring its quality as global minimum structure candidate. The fitter ones are more likely to be selected and produce the next generation population of structure candidates through crossover and mutation. Crossover mixes the features of two conformations and produce one or two new conformations. Mutation changes part of a conformation to generate a new

one. The new generation is maintained the same size as the old one. The process is repeated for a certain number of generation or until a satisfying result is obtained. In practice, genetic algorithms work quite well in finding the global minimum structure of LJ clusters. However, no existing theories so far can explain such success. There are some hypotheses, such as the building block hypothesis [59], trying to provide a theoretical explanation for their success but lacking consensus among researchers.

Hypersurface deformation approaches [60–63] apply transformations to smooth the potenital energy surface thus reduce the number of minima as well as large energy barriers between minima. The global minimum of this smoothed surface is easier to find and its position can then be mapped back to the original surface to get the real result. However, depending on the smoothing transformation used, the global minimum on the deformed surface might differ from the original one dramatically. A local search procedure can be used during the reverse mapping step to address this issue. Serveral methods have been proposed using the hypersurface deformation technique, such as the distance scaling method [63] and the stochastic tunneling method [64]. Basin hopping [65] is one of the most successful methods to solve the LJ problem. It performs a "staircase" deformation where energy of each point on the original potential energy surface is mapped to the minimum energy of the basin it belongs to. This transformation broadens the transitions between local minima resulting in a significant probability of occupation at the global minimum provided the thermodynamic energy is high enough to overcome the free energy barriers [66,67].

3.3 Virtual Reality

Virtual reality (VR) is a computer-generated environment that provides its users an experience through artificial sensory stimulations making them immersed in

the virtual enviroment with little awareness of the real world.

A VR system is mainly comprised of displays that immerse the user in a simulated enviroment. In addition to visual stimuli, the system includes an array of sensors that collect information about the use's behaviors and the surroundings and relays this to the software. Collectively, the sofware combines sensor data to generate a immersive simulated environment that the user can interact with in real time [68].

VR is most commonly used in video games and immersive cinema. It has also been used for education, medical/astronaut/driver training, architectural design and many other fields.

3.4 Hypothesis Testing

Hypothesis testing is a framework used in statistical analysis to determine the validity of a claim regarding a population parameter, such as its mean or variance, using sample measurements. It's generally conducted through the following four steps:

1. State the null hypothesis (H_0) and the alternative hypothesis (H_1).

   The null hypothesis, H_0, states what value of the parameter of a population we assume to be true. The alternative hypothesis, H_1, states the population parameter takes a value: (i) not equal to, (ii) greater than, or (iii) less than the value asserted by H_0. Based on H_1, the test is labeled with one of two categories: a one-tailed test when H_1 is stated as an inequality, and a two-tailed test when H_1 is stated as a negation of equality, as in case (i).

2. Set the significance level (α) of the test.

   There are four outcomes of a hypothesis test, as shown in Table 3.1. The significance level α is the maximum acceptable probability of committing a

Truth value of hypothesis	Decision	
	Do not reject	Reject
True	Correct	Type I error
False	Type II error	Correct

Table 3.1: Four outcomes of a hypothesis test.

type I error, in which a true hypothesis is mistakenly rejected. Common values used in studies are 5% and 1%.

3. Compute the value of the test statistic from the selected sample.

   The test statistic is a mathematical formula that its sample distribution under H_0 is known. For example, if we want to test whether two population means are the same under the assumptions that the two populations are independent from each other and variance of both are unkown and different from each other, the test statistic is the following formula

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{\sqrt{S_1^2/n_1 + S_2^2/n_2}} \tag{3.3}$$

$$\nu = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{(S_1^2/n_1)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}}, \tag{3.4}$$

   which follows a t-distribution with degree of freedom ν under the null hypothesis.

4. Decide whether or not the hypothesis should be rejected.

   Based on the sample test statistic value, the p-value can be determined. A p-value is the probability of obtatining a value of the test statistic as extreme as the result computed, assuming H_0 is true. H_0 is retained if the p-value is larger than α and rejected otherwise.

Alternatively, we can compute the critical values, i.e. the test statistic values whose p-value is equal to α. Test statistics more extreme than this value form a rejection region. H_0 is rejected if the sample test statistic value falls inside the rejection region and retrained otherwise.

Often, we have data collected from some experiments designed to study a phenomenon that we are interested in. We propose a statistical model to predict the outcome from predictors and assess how the proposed model fits our data using hypothesis testing. Linear models are the most widely used, which use weighted sum of the predictors to predict the outcome. The general form of a linear model is shown in the following equation:

$$Y_i = b_0 + b_1 x_{1,i} + b_2 x_{2,i} + ... + \epsilon_i, \tag{3.5}$$

where Y_i is the outcome, $x_{j,i}$ are the j predictors and ϵ_i is the error, all for the ith measurement. The fitness of the model is assessed by a test statistic, which is defined as the ratio of the variance that can be explained by the model (systematic variance) over the variance that can not (unsystematic variance):

$$\text{test statistic} = \frac{\text{systematic variance}}{\text{unsystematic variance}}. \tag{3.6}$$

Depending on our null hypothesis, test statistics take different form. If the null hypothesis is that there's no relationship between the outcome and predictors, the F statistic is used. If the null hypothesis is that the j predictors significantly predict the outcome, the t statistic is used.

If we include both fixed and random effects in a linear model, the model is then called a linear mixed model. The definition of fixed factors and random factors

varies among different sources [69] and whether a factor is chosen to be fixed or random is often problem/context dependent. In general, a factor is considered fixed if data is collected from all levels of interest of the factor. A random factor, on the contrary, only has a small random sample from some normal distribution of all possible treatments in an experiment.

The introduction of random effects to a linear model provides certain benefits. In a traditional linear model, we have to assume fixed intercept (b_0) and slopes (b_j) among different groups and the observational units are independent of each other. However, the real world data is often complex, includes missing data and has hierarchical structures in nature. As a result, such assumptions are often violated. A linear mixed model allows us to model the multilevel relationships and the non-independence in such data. In the case of missing data, a linear mixed model allows the parameters to be assessed from available data so that the whole data case needs not be deleted.

Chapter 4

Human-Guided Global Optimization of Lennard-Jones Clusters

4.1 Introduction

Global optimization is the process of finding a function's extremum (or minimum/maximum since maximization and minimization can be turned into each other by a simple overall sign change of the function) on the entire domain of it. It plays an important role in various areas of chemistry including cluster structure optimization [2, 70, 71], molecular distance geometry [72], molecular docking [73], protein folding [74–76], parameterization of force fields [77, 78] or semi-empirical methods, quantum optimal control theory [79], etc.

Global optimization methods can generally be classified into two large categories: deterministic global optimization methods, which guarantee the solution found is the true global minimum, and stochastic global optimization methods, for which

no rigorous global optimality can be guaranteed. Though deterministic methods are highly valuable, for a non-convex objective function they typically need to perform a complete search over all points in the domain of the function. The search space grows exponentially with dimensionality and even in a one-dimensional case, visiting the entire search space already needs an infinite number of function evaluations for continuous variables in the absence of further simplifying assumptions. Thus deterministic global minimization is often too expensive to be practical. For larger systems, stochastic methods using heuristic strategies to search the search space in a more or less intelligent way are often used. Such methods (e.g. simulated annealing, genetic methods, basin-hopping) can find minima with function values not too far above the global minimum much faster than deterministic methods can find the true global minimum.

Nonetheless, global optimization problems are still quite challenging, especially for large complex systems. Employing clever tricks that reduce the search space or lead the search to promising regions is a common technique, but these tricks are not trivial to find. And the most efficient methods are system/problem-dependent due to additional heuristic elements especially tailored for the system/problem making generalization of global optimization method for routine usage quite difficult.

Human minds are capable of intuitively forming simple, low-dimensional heuristic strategies when trying to solve complex high-dimensional problems. By combining human intelligence with traditional computational stochastic-heuristic global optimization algorithms, a hybrid method might be able to improve the sampling of the search space and guide the search to promising regions more efficiently thus finding the global minimum faster or minima with function value closer to true global minimum. A hybrid algorithm will be able to make use of the strengths of the computational algorithm for fine-scale optimization steps (i.e. near a local minimum, when

the function is locally convex) and the complementary strengths of human intuition for coarser-scale optimization steps (i.e. in deciding when to abandon a local basin, and in determining which distant region of the domain to explore next).

The idea of exploiting human intelligence to solve scientific problems by general publics have already been successfully applied by several "crowd-sourcing" games, including Foldit [1] for the protein structure prediction problem, CrowdPhase [80] for the phasing problem in x-ray crystallography, EteRNA [81] for the RNA design problem, Quantum Moves [82] for the optimization problem in quantum physics and Phylo [83] for the multiple sequence alignment problem.

Compared to traditional computational methods, these citizen science applications have the advantage of allowing massive computaional resources (computer and human) to run simultaneously in parallel, allowing the solution space to be explored in a much faster way and new solutions to be discovered. And the natural diversity of human brains allows the search space to be searched in a less biased way collectively by the citizens, finding solutions that are otherwise difficult to be discovered by traditional computational methods. Phylo [83], for example, completely relies on the advantage of such massive quantity of computing power (computer and human). The application is designed intentionally to encapsulate the scientific detials into a casual game to allow more participants to be involved, and thus more computing power to be collected.

In addition to the large quantity of computing power, other applications also explore humans' skill at visual problem solving. Foldit [1] replaces the stochastic element of the Rosetta's search algorithm, the random perturbation of protein stuctures, with human decision making. Quantum Moves [82] uses players' solutions as seeds instead of computer generated seeds for the following multistarting local optmization alogritm. Both cases show results that, for certian problems, the algorithms

exploiting human intelligence are able to give superior solutions compared to the purely computationl ones [1, 82]. These applications studied a complex optimization problem and replaced the stochastic phase of a high-level optimization paradigm. We asked whether human intelligence can benefit fundamental optimization algorithms also, as they are the foundations of more complex algorithms. And further, can we add the human intelligence as an additional element to the existing algorithm rather than replacing certain elements of the exiting algorithm and acheive the same or better result? We believe this kind of integration can be more easily expanded to other applications.

In this work, we explore how human intelligence can be integrated with traditional optimization algorithms to help finding the minimum energy conformation of homogeneous Lennard-Jones (LJ) clusters, which are formed by identical atoms interacting with each other under the following potential:

$$V_{LJ} = 4\epsilon \sum_{i=1}^{n} \sum_{j>i}^{n} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{4.1}$$

where ϵ and σ are atom type specific parameters, and r_{ij} is the distance between particle i and j.

Even though the mathematical form of the potential is rather simple, the problem is notorious hard to solve and serves as a standard benchmark for global optimization algorithms. The number of local minima grows exponentially with cluster size n. To find the one with the lowest energy, successful alogrithms need to move among different local areas, identify promising local areas and avoid revisting the same area as effectively as possible. We believe that humans are capable of moving atoms in a way that helps the overall cluster to hop from one local minima to another on the potential energy surface. Moreover, based on previous knowledge of different

cluster configurations and associated energies, humans can form a rough intuitive idea of whether the current local search area is likely to contain the global minimum. In addition, human memory can be used to decide whether the current local search is exploring an new local minima or revisiting a old one. Most importantly, the ability of human minds can be easily combined with current state of art global optimization algorithms rather than replacing them to make potentially more effective ones.

Here we test the idea using a simple global optimization algorithm containing a series of a short period of Metropolis Monte Carlo simulation followed by a steepest descent optimization. We demonstrate that by incorporating human intelligence, such algorithm can solve the optimization problem more effectively.

4.2 Methods

4.2.1 Global Optimization Algorithm

In this study, we chose a Metropolis Monte Carlo method coupled with steepest descent method (as shown in Algorithm 1; all values are in LJ units) as the computational global optimization algorithm for its simplicity.

In such a method, the steepest descent method brings the LJ cluster to one of its local minima, at which point the energy is compared to the lowest energy for the cluster, based on prior literature results. If the energy doesn't match, the Metropolis Monte Carlo method allows the cluster to escape the current local basin and explore the surrounding area. Then the steepest descent method again brings the cluster to one of its local minima (ideally different from previous one) and compares the energy with the lowest energy known to date. The process repeats until the "global minimum" (at which the energy matches the lowest energy known to date) is found.

Algorithm 1 Global Optimization

Input: N: number of particles; T: temperature of the cluster system; $E_{min;global}$: lowest energy found in the literature for Lennard Jones cluster of size N

Output: The optimum solution

(i) Initialization:

for $i = 1$ to N **do**

Randomly initialize the position ($\mathbf{x}_i$) of the i^{th} particle in a $10 \times 10 \times 10$ box with the following constraints:

• the distance between the i^{th} particle and any existing particle j (d_{ij}) is larger than 1.0 ($\forall j < i$, $d_{ij} > 1.0$)

• the distance between the i^{th} particle and at least one existing particle is smaller than 2.0 ($\exists j < i$, $d_{ij} < 2.0$)

end for

Calculate the cluster potential energy ($f(\mathbf{x}) = 4 \sum_{i=1}^{N} \sum_{j>i}^{N} (\frac{1}{d_{ij}^{12}} - \frac{1}{d_{ij}^{6}})$)

Steepest Descent phase:

(ii) $\beta = 0.001$

(iii) $\Delta\mathbf{x} = \beta \bigtriangledown f(\mathbf{x})$

if $||\Delta\mathbf{x}|| > 2.0$ **then**

$\beta = 0.5\beta$

Go to step (iii)

else

$\mathbf{x}' = \mathbf{x} - \Delta\mathbf{x}$

end if

if $f(\mathbf{x}') > f(\mathbf{x})$ **then**

$\beta = 0.5\beta$

Go to step (iii)

else

if $|f(\mathbf{x}) - f(\mathbf{x}'))| < 1e^{-12}$ **then**

$\mathbf{x} = \mathbf{x}'$

if $|f(x) - E_{min;global}| < 5e^{-7}$ **then**

Terminate

else

Go to step (iv)

end if

else

$\mathbf{x} = \mathbf{x}'$

$\beta = 1.1\beta$

Go to step (iii)

end if

end if

Metropolis Monte Carlo phase:
(iv) $\alpha = 0.04$
for $t = 1$ to $\lfloor 80.964 \times 1.09453^N \rfloor$ **do**
  for $i = 1$ to N **do**
    Randomly choose a direction and move the i^{th} particle along that direction with distance α
  end for
  Accept the new particle positions with the probability $\min(1, e^{-\frac{f(\mathbf{x}')-f(\mathbf{x})}{T}})$
end for
Go to step (ii)

Steepest descent is a widely used first-order iterative method to find the local minimum of a function. The method involving taking steps proportional to the negative gradient of the function at current point. Most commonly, a locally optimal step size found by a line search is used at each iteration. Such a line search is complex and can be time-consuming, thus is avoided in our algorithm. Instead, a fixed step size is used to determine the step size at each iteration. Careful considerations about the value of the step size are needed. If the step size is too small, convergence to the local minimum is too slow. On the other hand, if the step size is too large, the algorithm might fail to converge or even diverge. And as the algorithm moves to different regions of the energy surface, range of proper step size values might vary. The strategy we used here is to start with a very small step size to enable convergence. Then the step size is increased at each iteration by a small amount to speed up the converging process. However, if at any iteration, any particle is moving too far or the step results in a higher energy, the step size is reduced by half and this iteration is redone. Since steepest descent is used as part of the global optimization algorithm and the local minimum energy needs to be compared to the reference global minimum energy, it should stop at a position that's as close to the actual local minimum as possible. Considering the precision of double-precision floating point numbers, the stop limit for steepest descent was chosen to be an energy difference between two

iterations that's smaller than 10^{-12} (LJ units).

Metropolis Monte Carlo method is a Markov chain Monte Carlo method for generating a set of configurations of the system from a desired statistical mechanical distribution. At each step of the algorithm, a random move from the current state i to a new state j is tried. If the move results in an energy decrease, the move is accepted. If the move is uphill in energy, the move is accepted with a probability defined by the ratio of probabilities of state i and j:

$$\frac{P_j}{P_i} = e^{-\frac{v_j - v_i}{kT}}. \tag{4.2}$$

v_i, v_j are energy at state i and j, k is the Boltzmann constant and T is the system temperature. If the move is accepted, the system is now in the new state j. If the move is rejected, the system remains in state i.

To actually implement the method in our system, several details need to be further explained. First of all, we consider move algorithm. The random move can be done by changing position of one particle at a time, randomly choosing several particles and changing their positions, or changing positions of all particles simultaneously. The choice shouldn't affect the results. The move algorithm chosen here is to simultaneously change all particle positions.

Second of all, we consider the move size. At each iteration, the particles try a move of size α along a random direction. A large move will be more likely to be rejected, thus causing the algorithm to be less efficient. A small move, on the other hand, is more likely to be accepted, but can only explore a much smaller nearby area, causing the algorithm to be inefficient too.

Here, we calculated the acceptance ratio (r) at various α for clusters of size 5, 10, 15, 20, 25 and 30. Clusters were first optimized by a steepest descent simulation

then underwent a Metropolis Monte Carlo simulation of 10000 steps where the number of steps that were accepted (n_{acc}) is recorded and the acceptance is ratio is calculated:

$$r = \frac{n_{\text{acc}}}{10000}. \tag{4.3}$$

The value of α was chosen to be the largest value at which acceptance ratios for all clusters are greater than 20%. The results are presented in section 4.3.

Lastly, it is necessary to choose the number of Metropolis Monte Carlo iterations (n) between two steepest descent simulations. If the Metropolis Monte Carlo algorithm is only run for a short time, it's very unlikely the cluster can escape the current local basin and the next steepest decent will bring the cluster to the same local minimum. On the other hand, if Metropolis Monte Carlo algorithm is run for too long, the cluster is likely to have explored several local basins by the end of the Metropolis Monte Carlo simulation. Since in our current global optimization algorithm, only the local basin at which the cluster is at the end of the Metropolis Monte Carlo simulation that matters, sampling multiple energy basins in one Metropolis Monte Carlo phase decreases the performance of the algorithm. Also, the cluster is more likely to evaporate or dissociate with longer Metropolis Monte Carlo simulations (since the equilibrium phase for LJ atoms in an infinitely large box is a gas). The steepest descent algorithm might then fail to converge to a local minimum. And even if it does, it would require a lot of steps to do so. This too makes the performance of the global optimization algorithm undesirable.

Note that the length of the Metropolis Monte Carlo simulation depends on the system dimension. Generally, larger the cluster size, longer the simulation. Here, for each of the clusters of size 10, 15, 18, 20, 22, 25, 28, the algorithm was run with different n. The one that needs smallest total number of steps to find the "global

minimum" are chosen to be the n_{opt} for the cluster. Then the n_{opt} as a function of cluster size is fitted using those data. These results are summarized in section 4.3.

4.2.2 Hybrid Optimization Algorithm

The hybrid optimization algorithm integrates human input by allowing a person to move any particle in the cluster to any position in the space at any time during the automatically global optimization process introduced in previous section. And once the algorithm detects a human input, it will immediately perform a steepest descent simulation on the new configuration and continue the global optimization process from there. Formally, the hybrid algorithm is described in Algorithm 2. The Initialization, Steepest Descent and Metropolis Monte Carlo steps are the same as in Algorithm 1, thus those details are omitted here.

4.2.3 Virtual Reality (VR) App

To implement the hybrid optimization algorithm, a VR application has been developed with Google Daydream and Unity. To use the app, a Daydream-ready smartphone and a Daydream View (a headset and a controller) are needed. The smartphone needs to be placed in the front compartment of the headset and viewed in VR through the headset's two lenses.

Fig. 4.1 shows the view that the user can see upon starting the application. A cluster of 10 particles whose positions are randomly assigned is displayed in the center of the view. On the top left corner, information about the cluster energy and optimzation process are shown. Such information can be hidden by deselecting the "Show simulation information" option on the control panel which can be brought out clicking the "Menu" icon on the top right (Fig. 4.2). The control panel also allows

Algorithm 2 Hybrid Optimization

Input: N: number of particles; T: temperature of the cluster system; $E_{min;global}$: lowest energy found in the literature for Lennard Jones cluster of size N; human inputs(optional)

Output: The optimum solution

(i) Initialization

(ii) Steepest Descent phase:

while Steepest Descent is not converged **do**

  if Human input detected **then**

    Update particle positions

    Go to step (ii)

  end if

  Run one step of Steepest Descent

end while

if Global optimum is reached **then**

  Terminate

end if

(iii) Metropolis Monte Carlo phase:

for $t = 1$ to $\lfloor 80.964 \times 1.09453^N \rfloor$ **do**

  if Human input detected **then**

    Update particle positions

    Go to step (ii)

  end if

  Run one step of Metropolis Monte Carlo

end for

Go to step (ii)

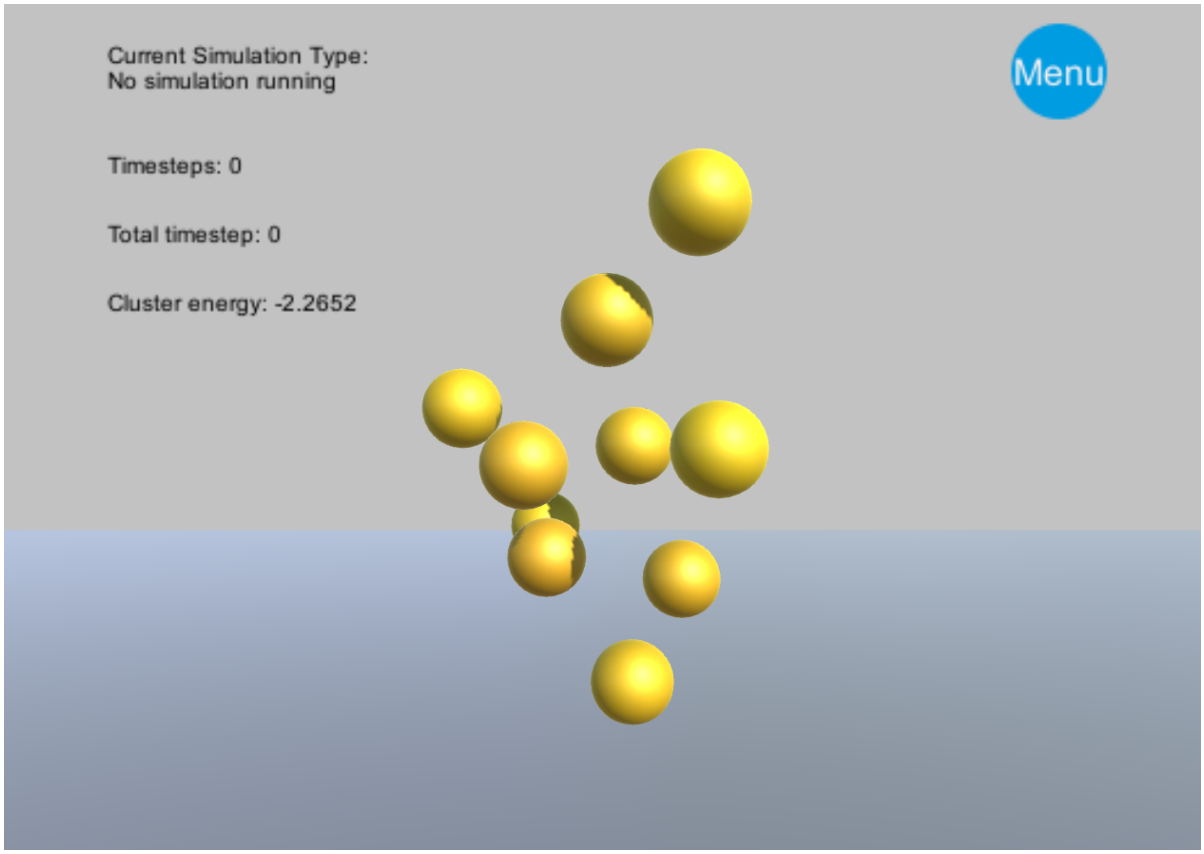


Figure 4.1: VR app interface.

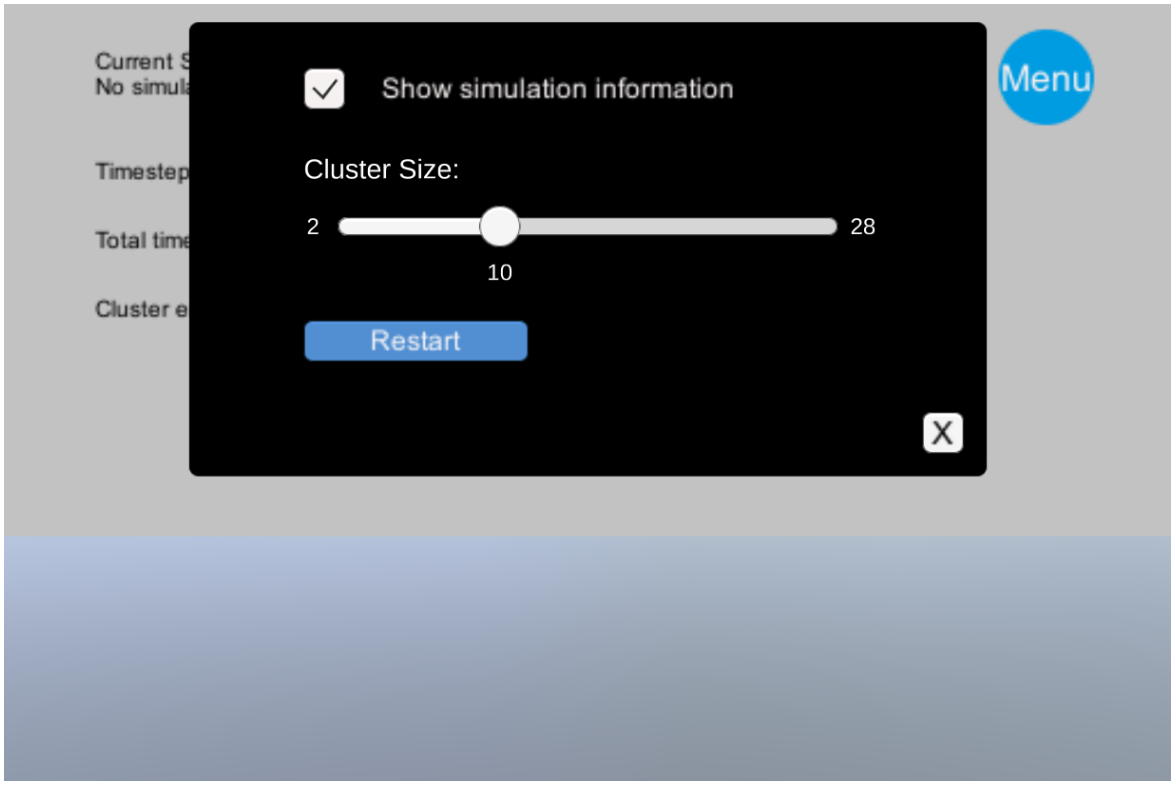


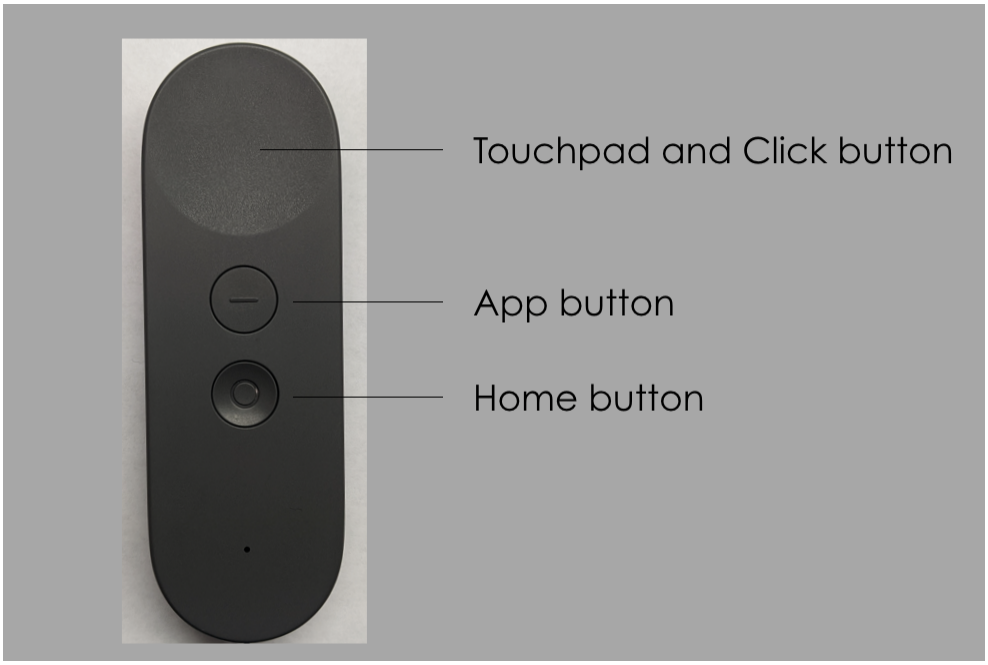
Figure 4.2: VR app interface - control pannel.

the user to choose the number of particles in the cluster by sliding the "Cluster Size" slide. Once the "Restart" button has been clicked, the current cluster configuration is discarded and a new one is initiated.

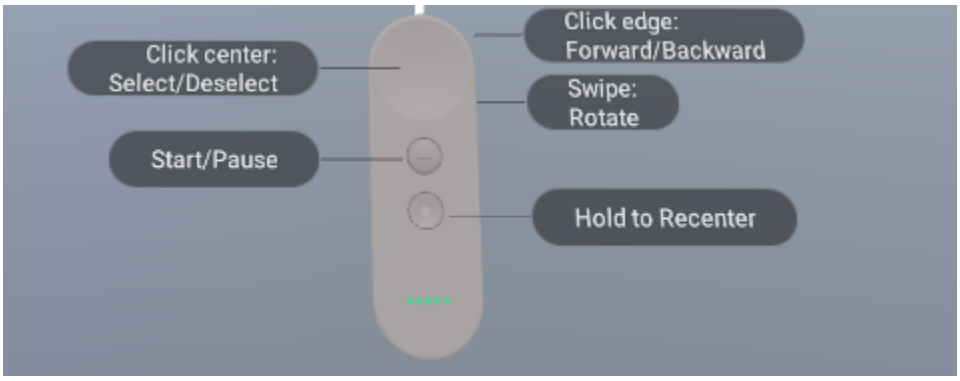
In our app, the user interacts with the VR environment mainly through the controller. A controller visualization is shown in VR corresponding to the actual controller the user holds in his/her hand. A laser is shown with the controller to allow the user to track easily where it is pointing at. Tool tips are also shown around the controller to remind the user of the allowable operations (Fig. 4.3).

At the begining, there's no simulation running. An optimization can be started by clicking the "App" button on the controller. If a simulation is currently running, clicking the same button causes it to pause. Users can move their viewpoint toward the cluster/away from the cluster by clicking the top/bottom edge of the touchpad until they reach a comfortable position. They can also swipe on the touchpad at any time to rotate the cluster to get whole picture of the cluster geometry. Typically, when the user starts the application, the cluster is shown in the center of view and controller is shown at lower right with the laser pointing to the front. In case the view shown is off the standard, holding the "Home" button will restore the standard view.

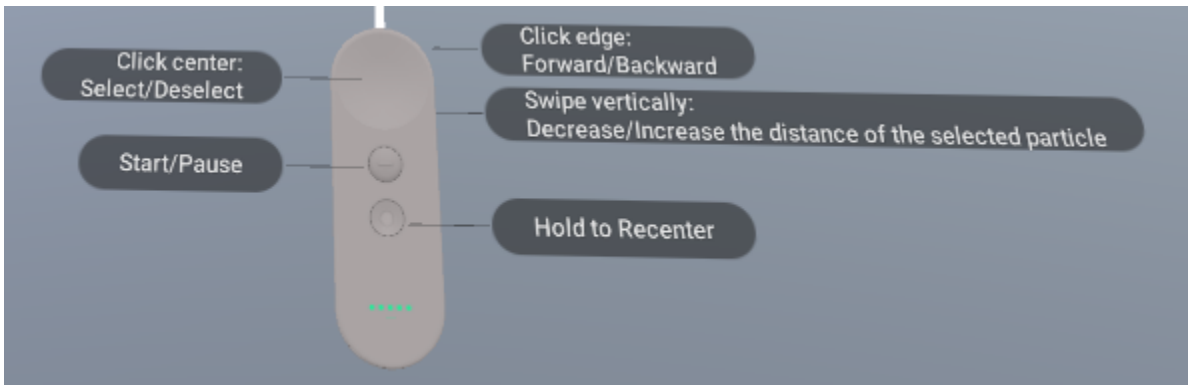
The optimization algorithm takes user inputs by allowing users to select any particle at any time and move the selected particle to any position in the space. The user selects a particle by pointing the controller toward the particle and clicking the center of the touchpad as the reticle shows on the particle. Once a particle is selected, the cluster is no longer rotatable until the selected particle is released. The user moves the selected particle up/down/left/right by moving the controller correspondingly and further/closer to the user by swiping vertically on the touchpad. Once the particle is at a desirable position, the user can release it by clicking the center of the touchpad



(a) The Daydream controller



(b) Normal situation



(c) During particle selection and movement

Figure 4.3: Tool tips for the Daydream controller.

again.

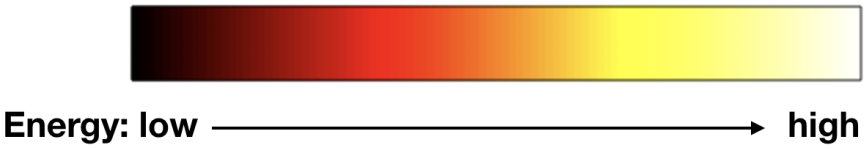
To assist humans in solving the problem, the particles are colored based on their individual potential energy (Fig. 4.4b). The black body color map (Fig. 4.4a) is used here, where as the energy goes from low to high, the color changes from black to red to yellow then finally to white. This allow the user to identify a relative high energy particle during the optimization process and help the optimization by placing it to a more favorable position.

Once the cluster is in its global optimum configuration, the "cluster energy" on the top left corner is highlighted red with "Global optimum" appended to it (Fig. 4.5) to inform the user the problem has been solved.

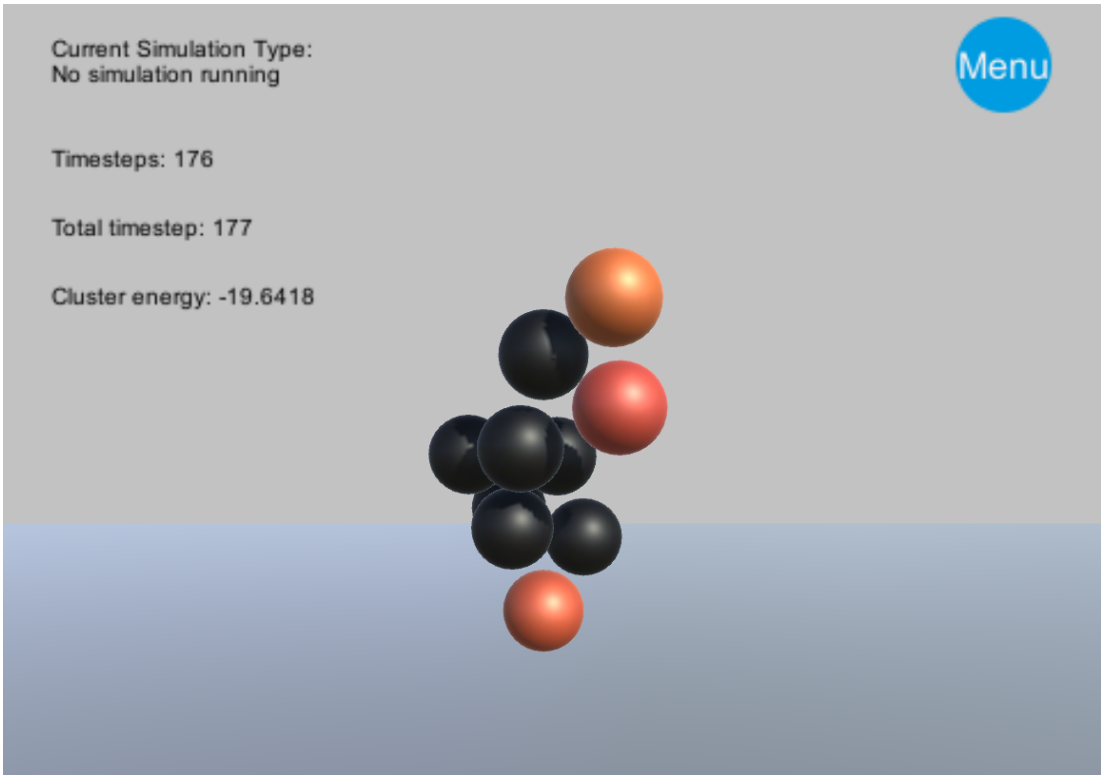
To allow further analysis of the perfomance of the hybrid optimization method, the app automatically records the lowest energy the algorithm has found at each timestep and saves them to the phone device while it's running.

4.3 Results and Discussion

Temperature is an important parameter for the Metropolis Monte Carlo phase of our optimization algorithm. Since we are interested in learning the optimal structure for LJ clusters, we hope to maintain our particle system in a liquid phase. If the temperature is so high that the equilibrium system is in the gas phase, clusters will not be sampled often, since interactions between LJ particles will be very small compared to kT. Even though we initialize the system in a way that LJ particles are in a cluster form, this initial cluster can easily be dissociated by the subsequent Metropolis Monte Carlo simulations. On the other hand, if the temperature is so low that the system is in a solid phase, we don't have this cluster forming problem. However, as shown in equation 4.2, the probability of accepting a Metropolis Monte



(a) The black body color map.



(b) An app snapshot

Figure 4.4: VR app interface — color change of the cluster.

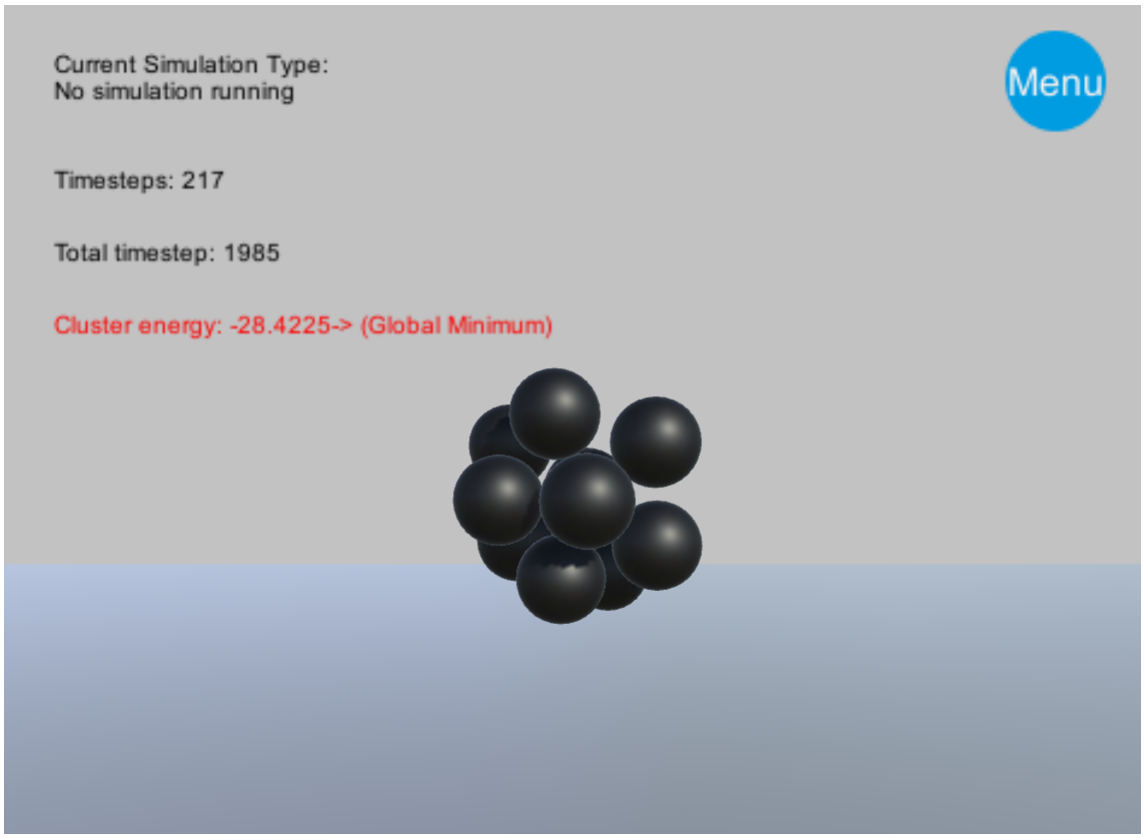


Figure 4.5: VR app interface — the end of a simulation.

Carlo move when the move results in an energy uphill depends on the temperature. Low temperature will cause the Metropolis Monte Carlo moves to be more likely to be rejected, thus making the algorithm less efficient. The LJ systems in this study have densities in the range of 10^{-3} to 10^{-2}, in reduced LJ units, thus the system temperature was chosen to be 0.7 to maintain the system in liquid phase, based on a LJ phase diagram.

As discussed in section 4.2.1, for the purpose of efficiency, we want the α to be as large as possible while maintaining a good acceptance ratio. Fig. 4.6 shows the acceptance ratio with different α for different cluster sizes. As shown in the graph, with α of 0.04, the acceptance ratio for all clusters of size 5, 10, 15, 20, 25 and 30 are larger than 20% which is high enough to be acceptable. With α larger than 0.04, the acceptance ratios for the cluster of size 30 decreases to around 0, which means that in Metropolis Monte Carlo simulations, the tentative moves are rejected most of the time, thus the system is almost always stuck in the initial local minimum. This is obviously very undesirable. Even though we might have chosen α differently for different cluster sizes, such choice will make determination of another important parameter, n_{opt}, more complicated since it is dependent on the choice of α. And because the overall efficiency of the optimization algorithm does not depend solely on α, this extra work might not make a huge difference on the performance of the algorithm. Thus an α of 0.04 was chosen for all cluster sizes.

With α determined, n is another important parameter that has a large effect on the performance of the algorithm. Fig. 4.7 shows total number of steps needed for a cluster of size 20 to reach the "global minimum" with different n. As shown in the figure, as n increases, the performance of the algorithm first improves, then gets worse, which agrees with our expectation as explained in section 4.2.1. The n_{opt} is chosen to be the n value corresponding to the lowest point in the graph, 500. Similar

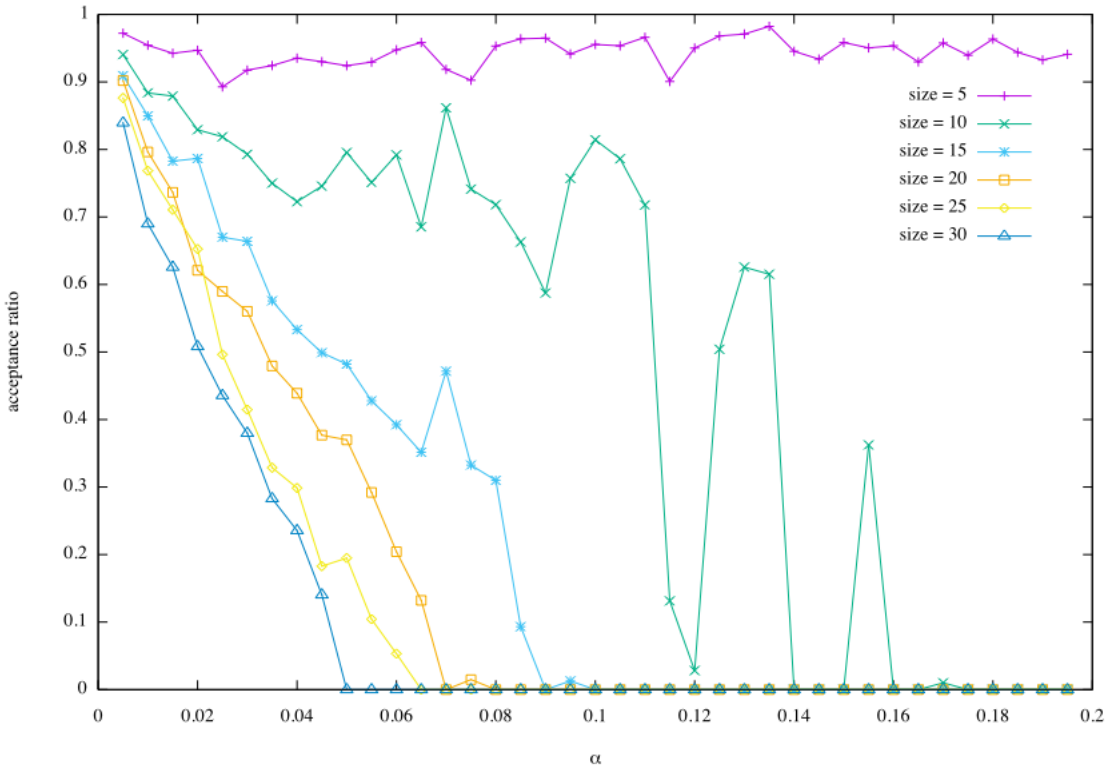


Figure 4.6: Acceptance ratio with different α for different cluster sizes.

Size	5	8	10	12	15	18	20	22	25	28
n_{opt}	100	150	200	200	300	450	500	650	750	100

Table 4.1: n_{opt} values for cluster of different sizes

graphs were obtained for cluster of sizes in the range of 5 to 28 and the corresponding n_{opt} values are shown in Table 4.1.

The value of n_{opt} varies with different cluster sizes. Thus it's impractical to choose a constant n value and expect it to work well for all cluster sizes. Instead, using a function of cluster size to represent n_{opt} is a more reasonable choice. The data were fit with a power function, which provided a reasonable description of the data. Fig. 4.8 shows the relationship between n_{opt} and cluster size. As shown in the figure, the power function $n_{\text{opt}} = 80.964 \times 1.09453^N$ fit the data adequately to describe the trend.

The performance data for the purely computational method is collected through a C++ implementation of Algorithm 1. Compared with the actual app, the C++ implementation excludes the graphic representation of the cluster and VR components. Further more, it runs on the desktop which has larger computational capacity compared to the phone. Together, the C++ implementation allows a faster data collection. To evaluate the quality of the data collected by the C++ implementation, 100 runs of both C++ implementation and unity vr implementation with cluster size 10 were conducted. Fig. 4.9 gives the comparison between the average data collected by both implementations.

In the graph, the two lines appears almost identical. Thus we claim that it's reasonable to collect the performance data for the purely computational method through the C++ implementation and use the collected data to establish a benchmark for the hybrid method.

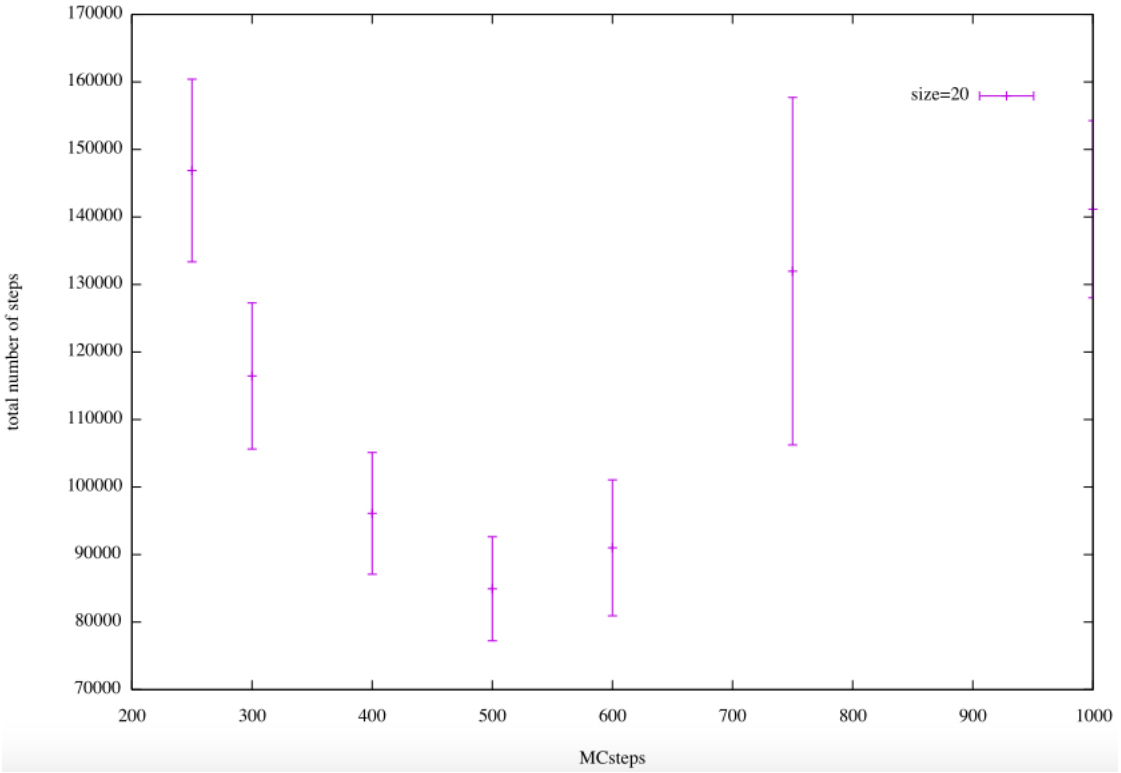


Figure 4.7: Total number of steps needed for a cluster of size 20 to reach the "global minimum" with different n. Each data point in the graph represent an average of 100 simulations. The error bars represent the standard error of the mean.

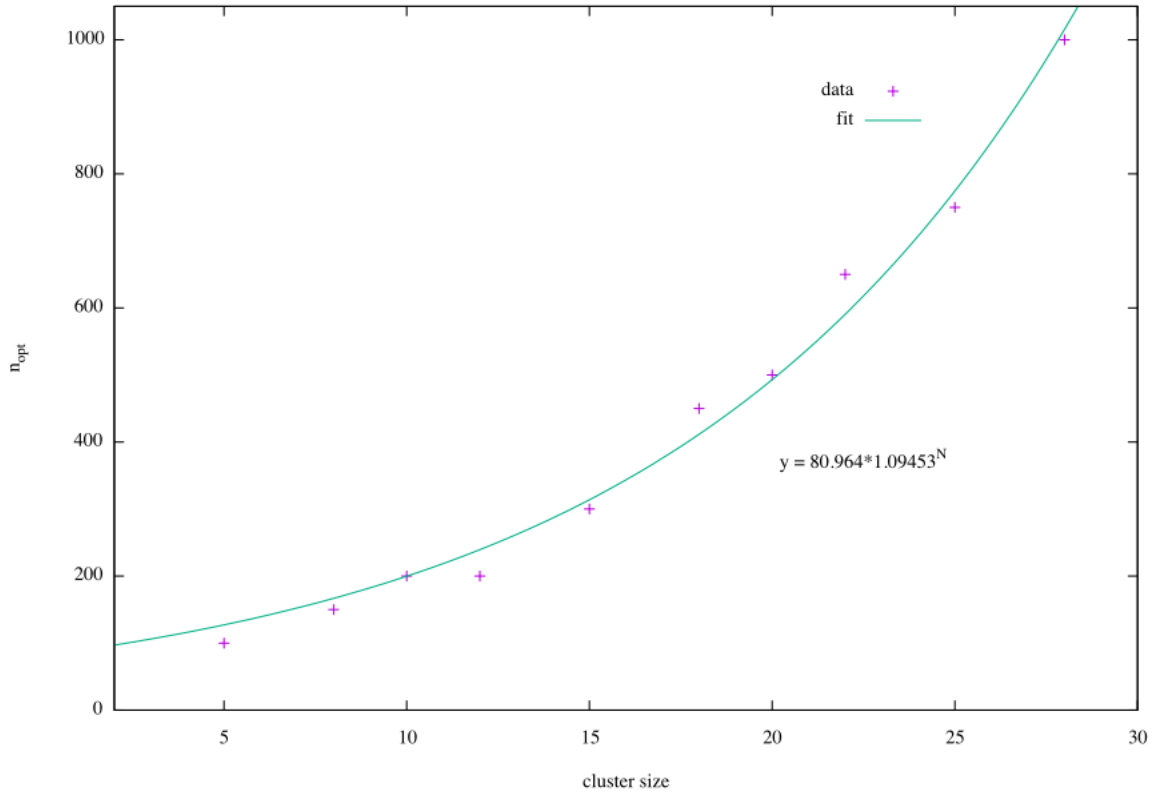


Figure 4.8: n_{opt} as a function of cluster size.

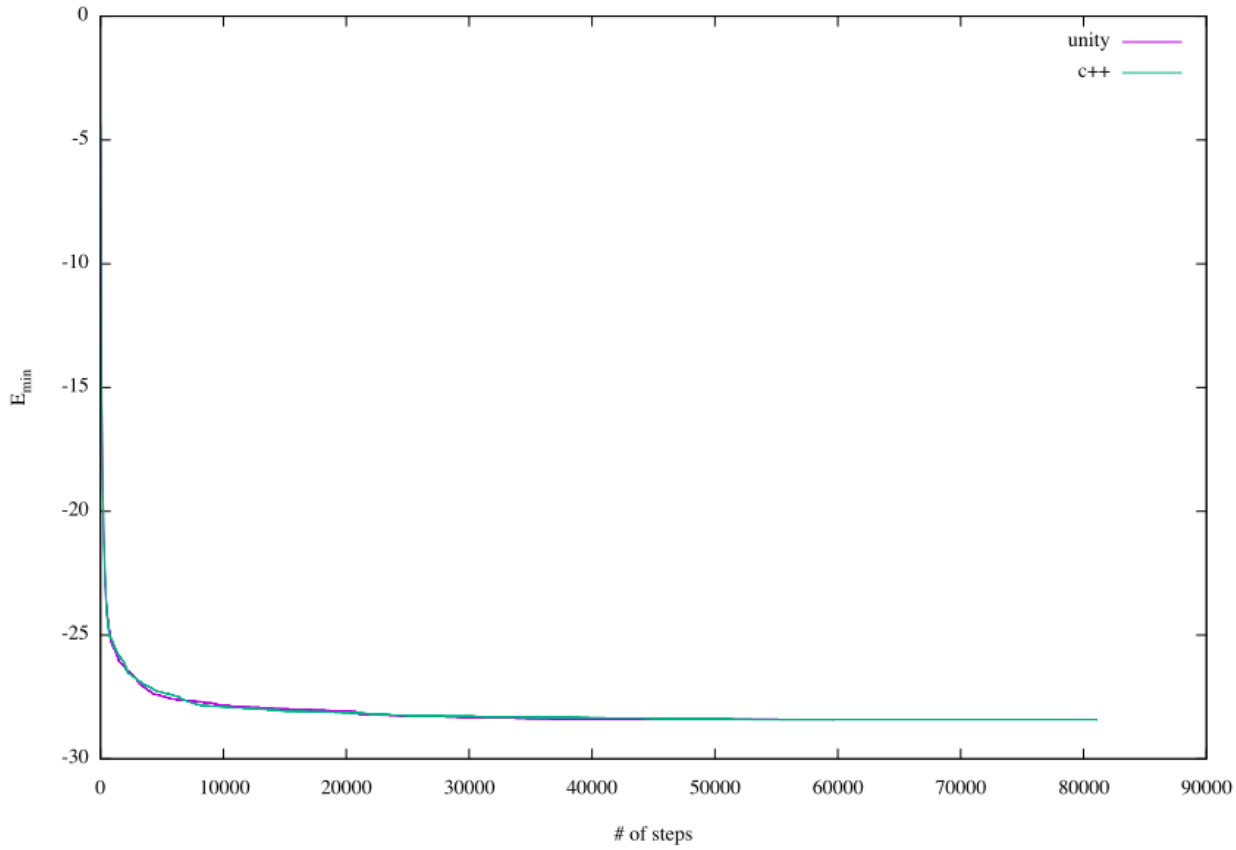


Figure 4.9: Comparison between the data collected by c++ implementation and unity VR implementation of Algorithm 1. Each data points represents an average of 100 runs with cluster size 10.

To collect the performance data for the hybrid method, we asked the volunteers to use the VR app and try to help the optmization process. The study is approved by Institutional Review Board (IRB) (Number : IRB2019-290). The informed consent form is attached in Section A. Due to practical considerations, the cluster sizes used for evaluating the hybrid method are restricted to the ones that can finished within 10 minutes half of the time using the purely computational method. So, for cluster size 2 to 28, with purely computational method, we first find the median finish time in the unit of number of timesteps. To find the conversion coefficient from number of timesteps to seconds, we run 5 simulations with cluster sizes 5,10,15,20 and 25 using the unity vr app but without the human input. The final results of median finish time in minutes vs. cluster size are shown in Fig. 4.10.

For cluster sizes < 10, the purely computaional method is fast enough that half of the runs finish within 3 minutes. It's reasonable to assume with those cluster sizes, a human will not improve the efficiency of the optimization process or even harm it as the process finishes before a human can form an intuition and helping the algorithm when it's reached a difficult point in the optimization. Thus the cluster sizes used for perfomance evaluation are chosen to be 10 to 15.

Each participant was asked to perform 3 trials with different cluster sizes. Before these 3 formal trials, they were given a cluster of size less than 10 as a practice trial to help them get familiar with app and gain essential knowledge about the problem they need to solve. Table 4.2 shows the records of the participant data. In summary, we had 22 participants: half female, half male. All the participants were associated with the chemistry department. Among them, one was a faculty, one was an instructor, 19 were graduate students and one was an Undergraduate student.

To compare the performance of the purely computational method and the hybrid method, we need to choose a quantity that characterize the performance of

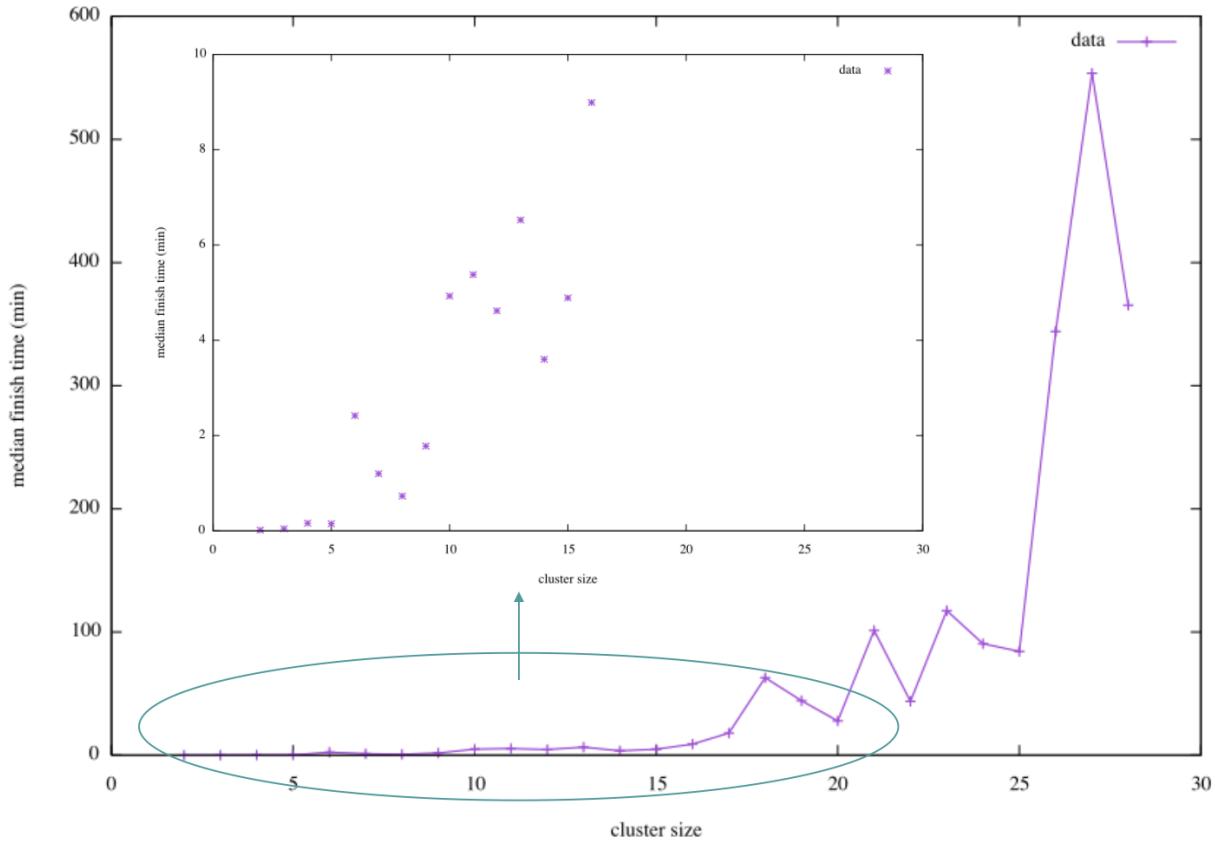


Figure 4.10: Median finish time (elapsed time) for different cluster sizes with purely computational method.

Id	cluster sizes	Gender	Occupation
1	10,11,12	F	Faculty(Chemistry)
2	11,12,13	M	Graduate Student(Chemistry)
3	12,13,14	M	Graduate Student(Chemistry)
4	13,14,15	M	Graduate Student(Chemistry)
5	10,11,12	M	Graduate Student(Chemistry)
6	11,12,13	F	Graduate Student(Chemistry)
7	12,13,14	M	Graduate Student(Chemistry)
8	13,14,15	F	Graduate Student(Chemistry)
9	10,11,12	F	Instructor(Chemistry)
10	11,12,13	M	Graduate Student(Chemistry)
11	12,13,14	F	Graduate Student(Chemistry)
12	13,14,15	M	Graduate Student(Chemistry)
13	10,11,12	F	Graduate Student(Chemistry)
14	11,12,13	F	Graduate Student(Chemistry)
15	12,13,14	F	Graduate Student(Chemistry)
16	13,14,15	F	Graduate Student(Chemistry)
17	10,11,12	M	Graduate Student(Chemistry)
18	11,12,13	M	Undergraduate Student(Chemistry)
19	12,13,14	M	Graduate Student(Chemistry)
20	13,14,15	M	Graduate Student(Chemistry)
21	10,11,12	F	Graduate Student(Chemistry)
22	11,12,13	F	Graduate Student(Chemistry)

Table 4.2: Participant records. (Data for participant 14 was discarded due to the device overheating during the experiment.)

Size	Computational Method			Hybrid Method		
	#trials	avg	std	#trials	avg	std
10	100	16552	14910.3	7	4635.6	2091.7
11	100	18218	14304.4	14	7852.9	7546.0
12	100	14377	12047.0	17	9128.1	8576.3
13	100	20710	15881.5	14	11320.8	10415.2
14	100	10973	8071.2	9	8403	8198.8
15	100	14766	9870.4	5	15020	10841.6

Table 4.3: Algorithm runtimes (number of timesteps) for different cluster sizes.

different algorithms.

The runtime is a natural choice. Table 4.3 shows the results of algorithm runtimes for different cluster sizes with both methods. The data is plotted in Fig. 4.11.

To understand how human involvement and cluster size affect the runtime of the algorithm, we performed a linear mixed model analysis using R [84]. The fixed factors we choose are the method (computational or hybrid), cluster sizes and an interactive term between the two. Because each participant did trials with several different cluster sizes, we can not consider these two factors to be independent. Also, the participants we had are only a random sample of the overal human population. The effect of the hybrid method depends on the particular sample of the participants. To take these issues into account, we incorporate the participant id as a random effect. The result of the linear mixed model is shown as the following:

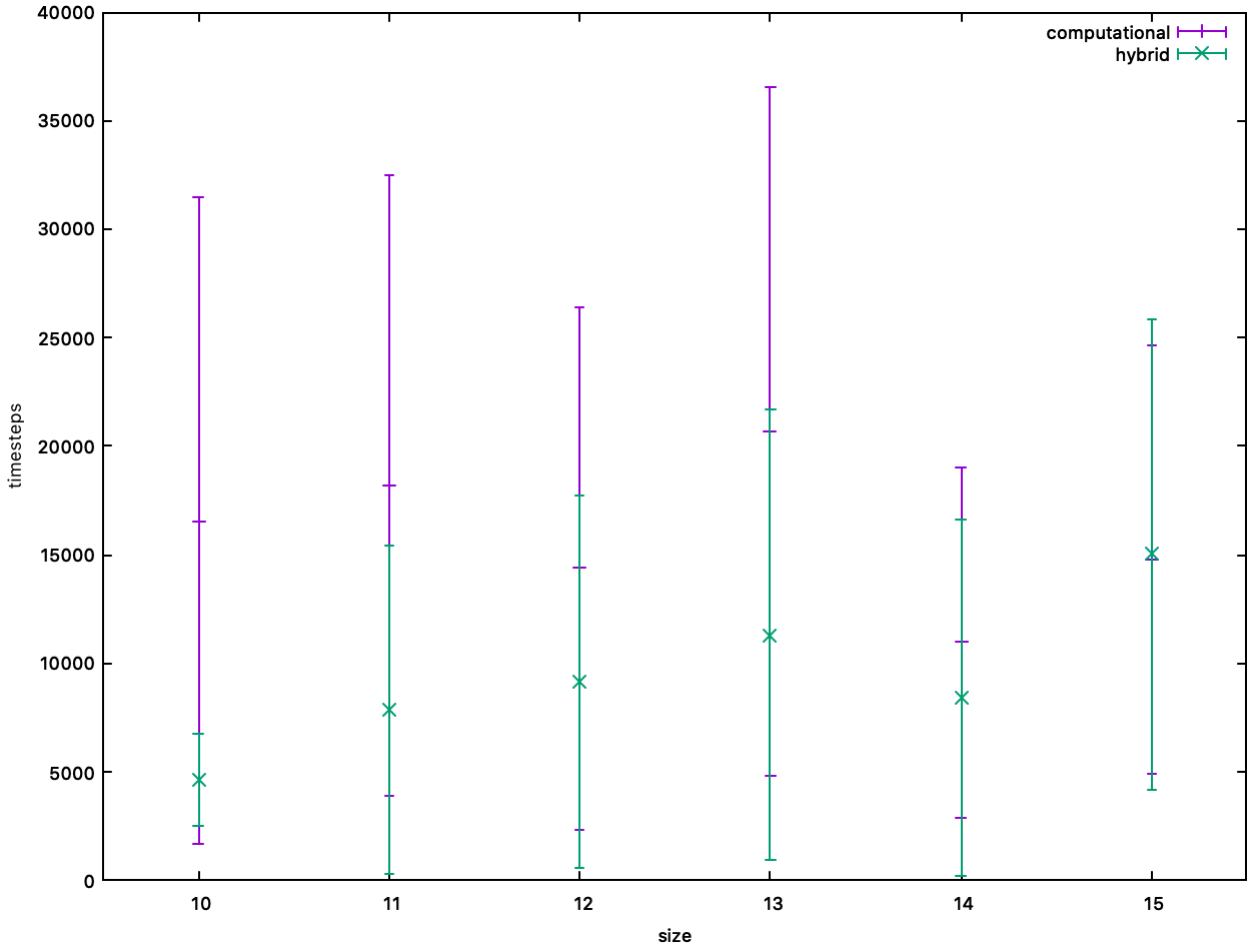


Figure 4.11: The plot of algorithm runtimes (number of timesteps) vs cluster sizes. The error bars represent the standard deviation.

```
Linear mixed-effects model fit by maximum likelihood
 Data: data
      AIC      BIC  logLik
  14364.2 14400.14 -7174.1

Random effects:
 Formula: ~method | participant_id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev       Corr
(Intercept) 4.020201e-01 (Intr)
method      2.656559e+00 -0.001
Residual    1.271974e+04

Fixed effects: timesteps ~ method + cluster_size + method:cluster_size
                        Value Std.Error  DF   t-value p-value
(Intercept)          24622.41  3847.735 636  6.399195  0.0000
method              -35293.20 15137.724  20 -2.331474  0.0303
cluster_size          -695.18   304.985 636 -2.279381  0.0230
method:cluster_size   2289.58  1210.664 636  1.891175  0.0591
 Correlation:
                     (Intr) method clstr_
method               -0.254
cluster_size         -0.991  0.252
method:cluster_size   0.250 -0.993 -0.252

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.3455961 -0.6919452 -0.2605430  0.3482647  4.9853531

Number of Observations: 660
Number of Groups: 22
```

The result shows that, with $\alpha = 0.05$, both method (p-value = 0.030) and cluster size (p-value = 0.023) have statistically significant effects on the runtime of the algorithm. Although the interactive term (p-value = 0.059) is not significant at the $\alpha = 0.05$ level, the value is near the $\alpha = 0.05$ cutoff, and might deserve further consideration. In other words, the effect of method on the timesteps required varies as the cluster size changes. To visualize the interaction effect, we plot the timesteps required vs. the cluster size for both hybrid method and the computational method, as shown in Fig.4.12. The shaded gray area surrounding the trend lines represents

the 95% confidence interval for the predicted timestep value at a specific cluster size. For cluster size 10 to 13, the 95% confidence intervals of both methods don't overlap and the expected timesteps required by the hybrid method is lower than the expected value for the purely computational method. As cluster sizes increase, the confidence intervals of both method gradually overlap, showing no statistically significant difference between the average timesteps required of both methods. The results suggest that human participation improves the performance of the optimization algorithm. However, such improvements tends to be diminished as the cluster sizes increase.

To further investigate the role humans play in the optimization process, we plot the minimum energy found till each timestep $E_{\min}$ vs the timestep t for both computational and hybrid methods, as shown in Fig. 4.13 and 4.14.

The faster $E_{\min}$ decreases, the more efficient the optimization method is. From the graphs we can see $E_{\min}$ decreases dramatically at the begining of the optimization. As the $E_{\min}$ approaches the global minimum, the rate of decrease gets slower due to the alogrithm frequently revisiting local minima that have been explored before and taking longer time to escape deeper minima. At the end of the optimization, the rate of decrease is almost zero.

For each cluster size i, the change of $E_{\min}$ with t is fitted by

$$E_{\min} = a\left[\exp(-t^{\gamma}) - \exp(-t_e^{\gamma})\right] + E_{\min,\text{global}}, \tag{4.4}$$

where a and γ are the fitting parameters, $E_{\min,\text{global}}$ is the global minimum energy for cluster of size i, and t_e is the timestep that $E_{\min,\text{global}}$ is found. The fitting results for both computational and hybrid methods are shown in Fig 4.13 and 4.14.

The parameter γ represent the efficiency of the optimization method. The larger the γ is, the faster the alogrithm finds the global minimum. Table 4.4 shows γ

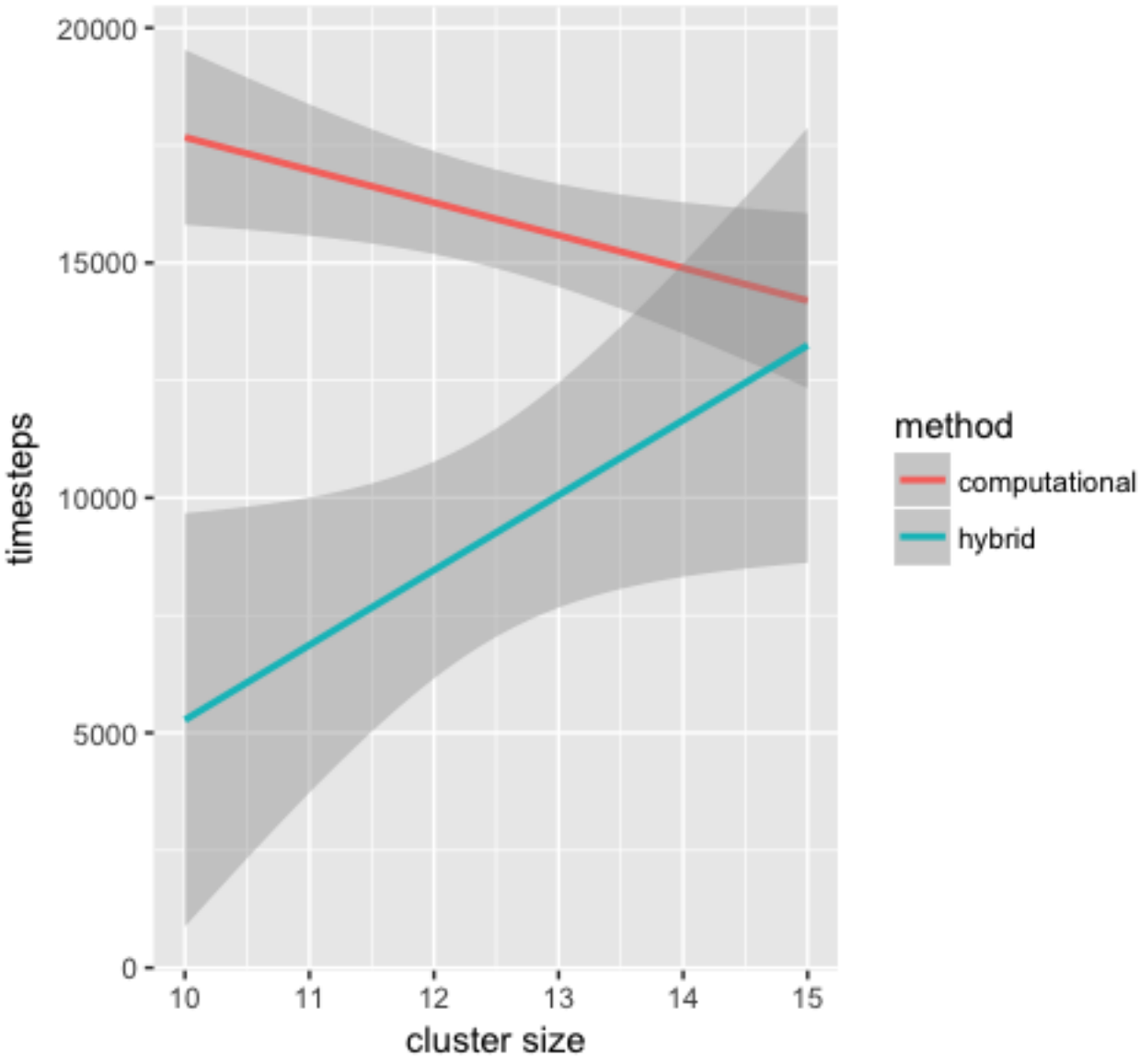
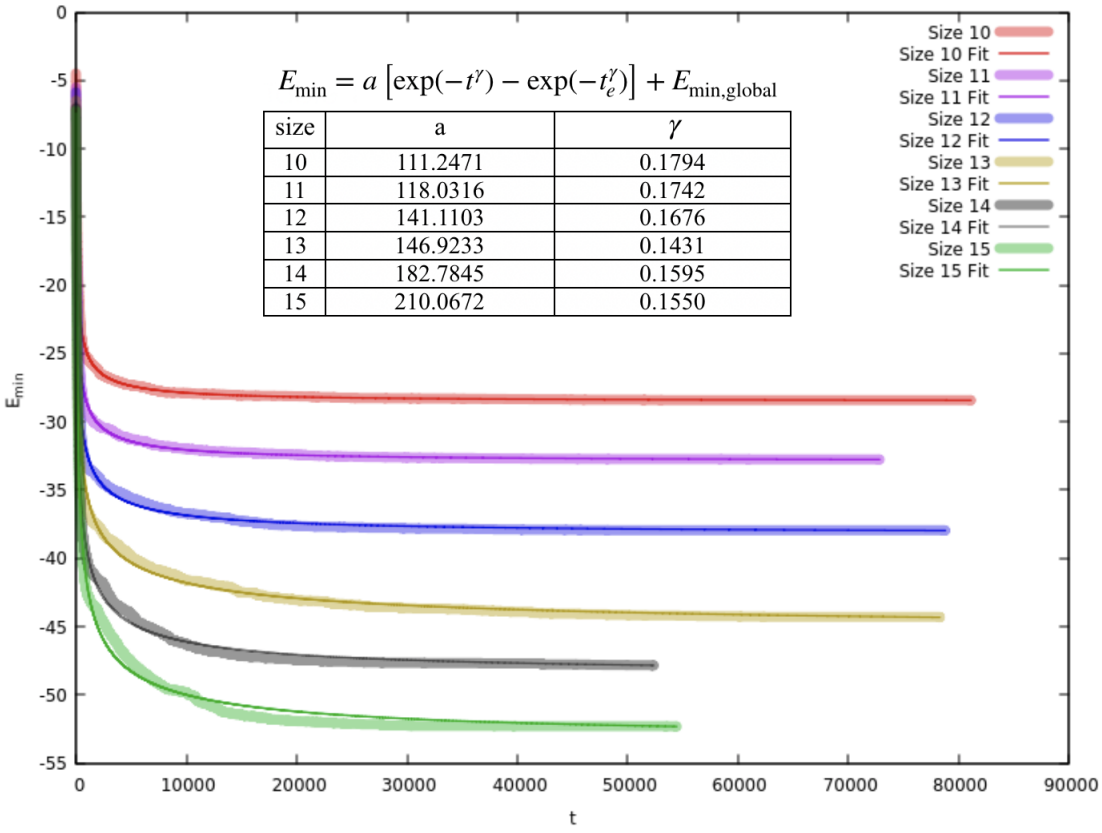
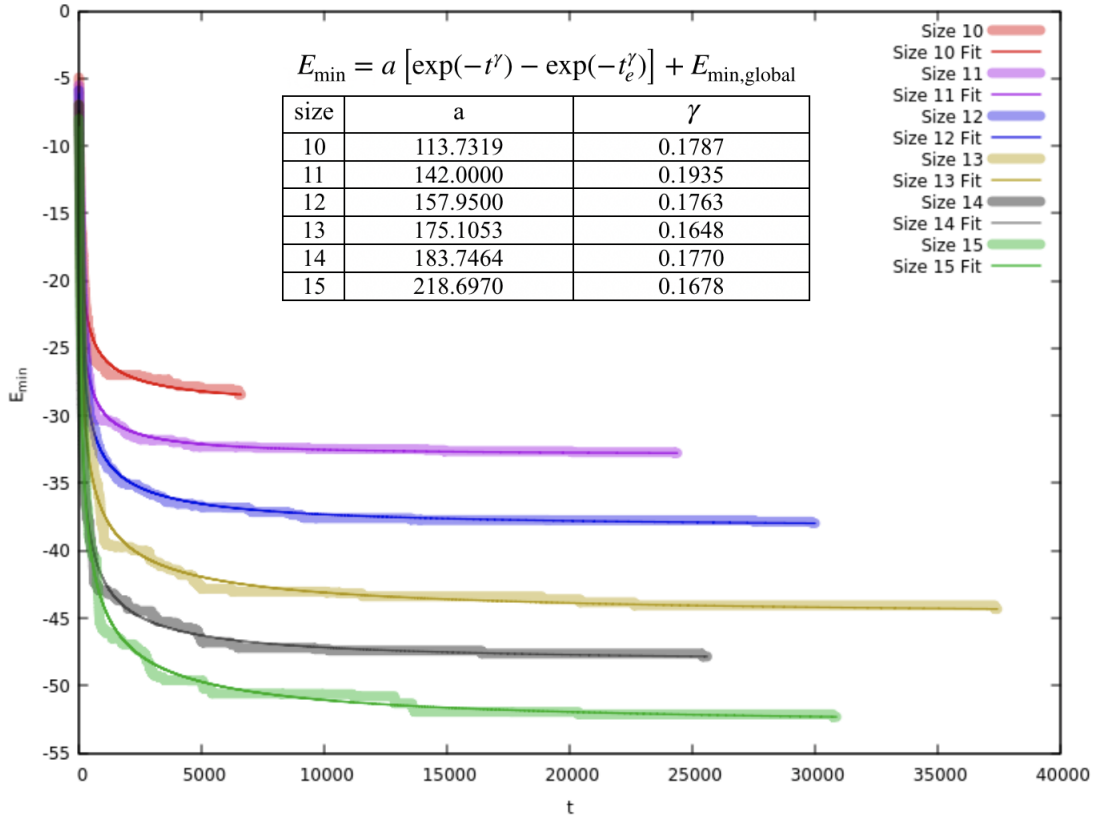


Figure 4.12: The effect of the human involvement changes with the cluster size. The shaded gray area surrounding the trend lines represents the 95% confidence interval for the predicted timestep value at a specific cluster size.



size	a	γ
10	111.2471	0.1794
11	118.0316	0.1742
12	141.1103	0.1676
13	146.9233	0.1431
14	182.7845	0.1595
15	210.0672	0.1550

Figure 4.13: The performance data for purely computational method. $E_{\min}$ is the minimum energy found for each timestep t. t starts with 1. Each data point is an average of 100 runs.



size	a	γ
10	113.7319	0.1787
11	142.0000	0.1935
12	157.9500	0.1763
13	175.1053	0.1648
14	183.7464	0.1770
15	218.6970	0.1678

Figure 4.14: The performance data for hybrid method. $E_{\min}$ is the minimum energy found for each timestep t. t starts with 1. Data points for size 10, 11, 12, 13, 14 and 15 are averages of 6, 12, 15, 13, 9 and 5 runs, respectively.

Size	Computational Method	Hybrid method
10	0.1794	0.1787
11	0.1742	0.1935
12	0.1676	0.1763
13	0.1431	0.1648
14	0.1595	0.1770
15	0.1550	0.1678

Table 4.4: γ values for different cluster sizes.

Size	Computational Method		Hybrid Method	
	meidan	avg	median	avg
10	12354	16552	5069	4636
11	13480	18218	4746	7853
12	11577	14377	7041	9128
13	16358	20710	5701	11321
14	9026	10973	5157	8403
15	12257	14766	12996	15020

Table 4.5: Average runtimes (number of timesteps) vs median runtimes (number of timesteps) for different cluster sizes.

values for clusters of different sizes for both computaional and hybrid methods.

In general, as the cluster sizes increases, the gamma vaule decreases. This is expected since the number of local minimum the depth of local minimum increase as the cluster size increase causing the algorithm to be less efficient. The inversely proportional relationship between γ and cluster size n is quite prominent for computational method as shown in Figure 4.15. It's worth noting that for cluster size 13, the computational alogrithm has a particular hard time optimize the cluster structure.

With the help of human, we find that the optmization efficiency increase except for cluster of size 10. This is not consistent with previous conclusions we made based on the linear mixed model analysis. Following are some possible explanations.

1. Average might not be a good characterization for the runtime data. Table 4.5

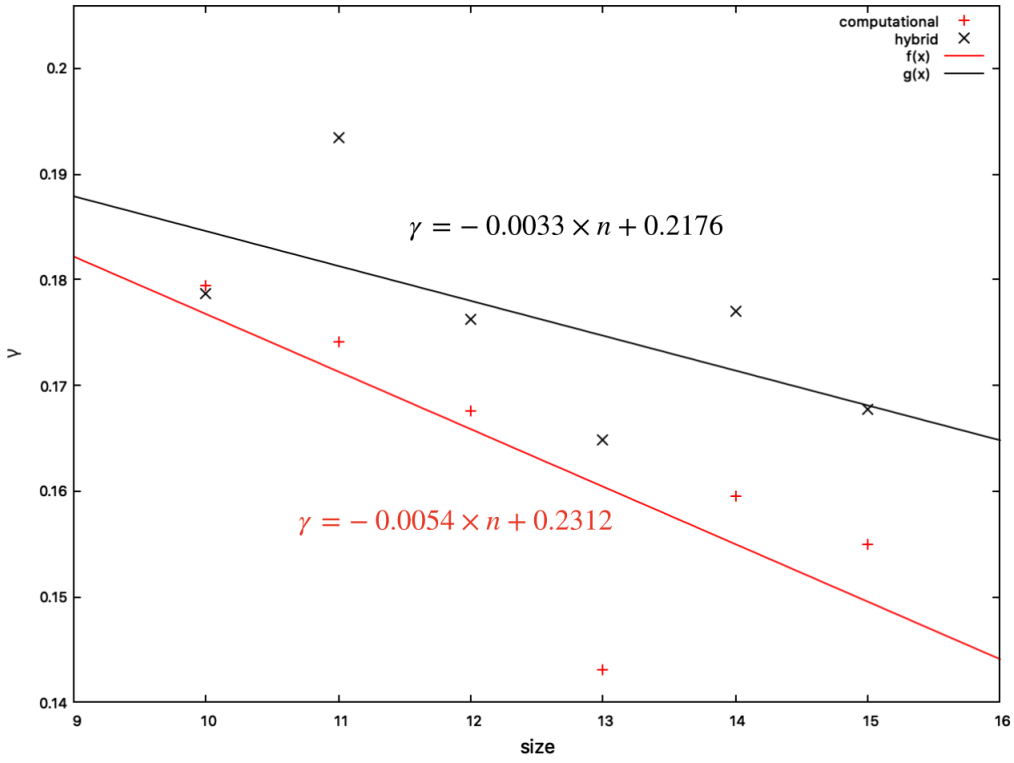


Figure 4.15: γ as a function of cluster size (n).

shows the average runtimes vs the median runtimes. The medians are smaller than the average except for cluster of size 10 with hybrid method, indicating the runtime distribution is not symmetrical, but right-skewed. This indicates average might not be a good characterization for the runtime data. The unusual trend of cluster size 10 with hybrid method migh be attributed to under sampling since the sample size is only 7.

2. There are sampling errors associated with γ. The γ value is based on the function fitting of average $E_{\min}$ vs t. With large sampling sizes, especially for the hybrid method, the trend of $E_{\min}$ over t can potential change and a different γ value could be obtained.

3. The function used to fit the $(t, E_{\min})$ doesn't match the true function underlying the data.

4. For size 10, the data shows that the starting point of $E_{\min}$ for the hybrid method is smaller than the starting point of $E_{\min}$ for the computational one. This could be due to humans are able to help to select a starting configuration with lower energy or simply sampling errors.

5. For cluster of size 14 and 15, γ values suggests humans improving the computationl method while the linear mixed model analysis suggests there's no statistically significant difference between efficency of computaional method and hybrid method.

   It could be because that humans are able to make large moves that allows the alogritm to explore different regions in the potential energy surface faster, causing an initial faster decrease in $E_{\min}$, thus larger γ values. This also causing the strong dependency of γ on cluster sizes with the computationl method to be

weakenend with the hybrid method, as shown in fig 4.15. But to find the actual global minimum, the alogritm needs to visit a specific basin with extremely small area compared to the whole energy surface, causing a very long tail of very slow decrease in the $E_{\min}$. As the cluster sizes increase, the structure gets more complicated and humans are not able to help much with identifying the exact region that are relavent to the global minimum structure, causing the overall average runtime to be the same for both method.

Or this controdiction is simply due to the sampling errors.

Futher studies with more data are needed to investigate those possible explanations and draw a more confident conclusion about the impact of human inputs on the optimization process.

4.4 Conclusions

In this project, we explore the idea that human intelligence can be integrated into a computational optimization algorithm to allow faster optimization process. By using homogeneous LJ clusters and a simple Metropolis Monte Carlo coupled with steepest descent optmization alogrithm, we are able to show preliminarily that human does have some positive impact on the structure optimization process, at lease for cluster of certian sizes. We hypothesize that the increased efficency of the optimization is attributed to human helping the alogrithm to start with a initial structure that has lower energy compared to a randomly selected one, make large moves to cover larger areas in the potential energy surface and escape a deep local minimum.

While only preliminarily, the result is significant as it provides a new strategy to improve the existing optimization algorithms that can potentially solve difficult optmization problems are currently unsolvable.

While promising, future works are needed to confirm our result and investigate in details under what conditions and how exactly human can help with the optimization. And the insights gained by studying the human behaviors during the optimization can potenitally be transfered to new strategies or improve existing strategies for computational algorithms.

Chapter 5

Future Work

In this section, we discuss the potential future works for the project.

First, more data should be collected with more participants to allow more confident conclusions.

We hypothesis that in our test system, human intelligence can only help the optimization with cluster of sizes in a certain range. If the cluster size is too small, the compuational method finish too fast for human to help. If the cluster size is too large, the cluster structure is too complicated for human to form intuitions that are helpful to the optimization alogrithm. To test this hypothesis and find out the critical cluster sizes, data with a large range of cluster size need to be collected and studied.

During the trials, we observed that some participants provide no inputs during the optimization process. Features such as recording user interaction counts and user interaction types can be added to the VR app to allow a more accurate study of the impact of human intelligence on the optimization process.

The lack of inputs can be caused by either the participants don't have any intuition that they think will help the optmization process or the participants are not fully engaged in the task.

In the former case, we should investigate how the background of the participants affects the ability of forming intuitions and helping the optimization process by grouping the participants into groups such as "general public", "undergraduates", "graudates", "professors" based on their supposed knowledge about chemistry and compare participants' performance in different groups. Currently, the participant are given a cluster of size < 10 for them to get familiar with different operations in the app and gain the necessary knowledge that are important to form good intuitions that help with the optmization. A multi-stage in-app interactive introduction section can be used instead, to allow the first-time user to learn the essentials faster.

In the second case, we can add more game-like features such as sound effects to better engage the participants and encourge them to actively thinking and solving the optimization problem.

Efforts can also be made to study the human behaviors that help with the optmization. Even though human approach the problem intuitively, we can identify partterns and find strategies in those intuitive behaviors. This can potentially provides insights on mechanism of optimization process and improvements to the existing computationl optimization algorithms.

At last, we can try to incorporate human intelligence with the state-of-art optimization algorithms to solve more complicated optimization problem.

Appendices

Appendix A Informed Consent Form

Information about Being in a Research Study
Clemson University

Human-Guided Global Optimization of Molecular Structures via Virtual Reality

KEY INFORMATION ABOUT THE RESEARCH STUDY

Voluntary Consent: Professor Steven J Stuart is inviting you to volunteer for a research study. Steven J Stuart is a chemistry professor at Clemson University conducting the study with Wenxing Zhang, a chemistry PhD student.

You may choose not to take part and you may choose to stop taking part at any time. You will not be punished in any way if you decide not to be in the study or to stop taking part in the study.

Alternative to Participation: Participation is entirely voluntary and the only alternative is to not participate.

Study Purpose: The purpose of this research is to test whether using human input in a virtual reality (VR) environment can improve the rate at which a computational method can find optimal structures for a molecular cluster.

Activities and Procedures: Your part in the study will be to use a VR app, and provide input during optimization of several structures. You will be instructed in the use of a VR headset, and a VR app through which you can manipulate a molecular cluster displayed in VR. You will then engage in several trials in which you will assist an automatic computational algorithm in searching for optimal (low-energy) structures for the molecular cluster.

Participation Time: You may be asked to participate in one or two one hour sessions.

Risks and Discomforts: There is a possibility of certain risks or discomforts that you might expect if you take part in this research. Some users complain of motion sickness or migraine headaches when using VR technology for prolonged periods of time. You will be able to stop at any point if you begin to experience any discomfort while participating.

Possible Benefits: You will not benefit directly from taking part in this study, aside from any enjoyment derived from playing a video game-like app, and perhaps gaining some chemical intuition and knowledge about the stability of cluster structures. The research does have broader possible benefits to the fields of chemistry, education, and computer science, in advancing the methods used for structure optimization, and developing methods at the boundary between human interfaces with computational technology.

EQUIPMENT AND DEVICES THAT WILL BE USED IN RESEARCH STUDY

A VR headset and a cellphone with a VR app will be used in this study. You might experience motion sickness or migraine when using the VR headset for a long time. Notify the research team immediately if you experience any discomforts during the study. If you continue to experience any discomforts after the study, contact your preferred healthcare provider and notify

the research team. Clemson University has not set aside funds to compensate you for any injury, complication or related medical care that may arise from participation in this study.

PROTECTION OF PRIVACY AND CONFIDENTIALITY

The results of this study may be published in scientific journals, professional publications, or educational presentations.

No personal information will be collected during the study. The only data collected will be the progress and performance of the optimization, and whether or not a human was contributing to the optimization. This data will not be collected with any identifying information about individual participants.

The information collected during the study could be used for future research studies or distributed to another investigator for future research studies without additional informed consent from the participants or legally authorized representatives.

We might be required to share the information we collect from you with the Clemson University Office of Research Compliance and the federal Office for Human Research Protections. If this happens, the information would only be used to find out if we ran this study properly and protected your rights in the study.

CONTACT INFORMATION

If you have any questions or concerns about your rights in this research study, please contact the Clemson University Office of Research Compliance (ORC) at 864-656-0636 or irb@clemson.edu. If you are outside of the Upstate South Carolina area, please use the ORC's toll-free number, 866-297-3071. The Clemson IRB will not be able to answer some study-specific questions. However, you may contact the Clemson IRB if the research staff cannot be reached or if you wish to speak with someone other than the research staff.

If you have any study related questions or if any problems arise, please contact Prof. Steven J Stuart at Clemson University at 369 Hunter Laboratory, ss@clemson.edu.

CONSENT

By participating in the study, you indicate that you have read the information written above, been allowed to ask any questions, and you are voluntarily choosing to take part in this research. You do not give up any legal rights by taking part in this research study.

Participant's signature: _____ Date: _____

Print name: _____

A copy of this form will be given to you.

Bibliography

[1] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 08 2010.

[2] Mathias S. Jorgensen, Michael N. Groves, and Bjork Hammer. Combining evolutionary algorithms with clustering toward rational global structure optimization at the atomic scale. *Journal of Chemical Theory and Computation*, 13(3):1486–1493, 2017. PMID: 28186745.

[3] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

[4] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 1984.

[5] Gary S. Grest and Kurt Kremer. Molecular dynamics simulation for polymers in the presence of a heat bath. *Phys. Rev. A*, 33:3628–3631, May 1986.

[6] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984.

[7] William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31:1695–1697, Mar 1985.

[8] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998. PMID: 24889800.

[9] Robert B. Best, Xiao Zhu, Jihyun Shim, Pedro E. M. Lopes, Jeetain Mittal, Michael Feig, and Alexander D. MacKerell. Optimization of the additive charmm all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain $\chi 1$ and $\chi 2$ dihedral angles. *Journal of Chemical Theory and Computation*, 8(9):3257–3273, 2012. PMID: 23341755.

[10] Jay Ponder and David Case. Force fields for protein simulations. protein simulations. *Adv. Prot. Chem*, 66:27–85, 02 2003.

[11] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996.

[12] Ron Elber. Perspective: Computer simulations of long time dynamics. *The Journal of Chemical Physics*, 144(6):060901, 2016.

[13] William A. Eaton, Victor Muñoz, Stephen J. Hagen, Gouri S. Jas, Lisa J. Lapidus, Eric R. Henry, and James Hofrichter. Fast kinetics and mechanisms in protein folding. *Annual Review of Biophysics and Biomolecular Structure*, 29(1):327–359, 2000. PMID: 10940252.

[14] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman J.C Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics*, 23(3):327 – 341, 1977.

[15] Hans C Andersen. Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52(1):24 – 34, 1983.

[16] Keith J. Laidler and M. Christine King. Development of transition-state theory. *The Journal of Physical Chemistry*, 87(15):2657–2664, 1983.

[17] Donald G. Truhlar, Bruce C. Garrett, and Stephen J. Klippenstein. Current status of transition-state theory. *The Journal of Physical Chemistry*, 100(31):12771–12800, 1996.

[18] Peter G. Bolhuis, David Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53(1):291–318, 2002. PMID: 11972010.

[19] Arthur F. Voter. A method for accelerating the molecular dynamics simulation of infrequent events. *The Journal of Chemical Physics*, 106(11):4665–4677, 1997.

[20] Arthur F. Voter. Hyperdynamics: Accelerated molecular dynamics of infrequent events. *Phys. Rev. Lett.*, 78:3908–3911, May 1997.

[21] Arthur F. Voter, Francesco Montalenti, and Timothy C. Germann. Extending the time scale in atomistic simulation of materials. *Annual Review of Materials Research*, 32(1):321–346, 2002.

[22] Danny Perez, Blas P. Uberuaga, Yunsic Shim, Jacques G. Amar, and Arthur F. Voter. Chapter 4 accelerated molecular dynamics methods: Introduction and recent developments. volume 5 of *Annual Reports in Computational Chemistry*, pages 79 – 98. Elsevier, 2009.

[23] John G. Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, 3(5):300–313, 1935.

[24] David A. Pearlman. Determining the contributions of constraints in free energy calculations: Development, characterization, and recommendations. *The Journal of Chemical Physics*, 98(11):8946–8957, 1993.

[25] Robert W. Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

[26] T. P. Straatsma and J. A. McCammon. Multiconfiguration thermodynamic integration. *The Journal of Chemical Physics*, 95(2):1175–1188, 1991.

[27] G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187 – 199, 1977.

[28] Shankar Kumar, John M. Rosenberg, Djamal Bouzida, Robert H. Swendsen, and Peter A. Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[29] Beata Dworakowska and Krzysztof Dołowy. Ion channels-related diseases. *Acta biochimica Polonica*, 47:685–703, 02 2000.

[30] Benoît Roux and Roderick MacKinnon. The cavity and pore helices in the kcsa k+ channel: electrostatic stabilization of monovalent cations. *Science*, 285 5424:100–2, 1999.

[31] S. Long, X. Tao, E. Campbell, and R. MacKinnon. Atomic structure of a voltage-dependent k+ channel in a lipid membrane-like environment. *Nature*, 450, 1 2007.

[32] Stephen B. Long, Ernest B. Campbell, and Roderick MacKinnon. Crystal structure of a mammalian voltage-dependent shaker family k+ channel. *Science*, 309(5736):897–903, 2005.

[33] Simon Bernèche and Benoît Roux. Energetics of ion conduction through the k+ channel. *Nature*, 414:73–77, 2001.

[34] Morten Ø. Jensen, David W. Borhani, Kresten Lindorff-Larsen, Paul Maragakis, Vishwanath Jogini, Michael P. Eastwood, Ron O. Dror, and David E. Shaw. Principles of conduction and hydrophobic gating in k+ channels. *Proceedings of the National Academy of Sciences*, 107(13):5833–5838, 2010.

[35] Morten Ø. Jensen, Vishwanath Jogini, Michael P. Eastwood, and David E. Shaw. Atomic-level simulation of current–voltage relationships in single-file ion channels. *The Journal of General Physiology*, 141(5):619–632, 2013.

[36] Simon Bernèche and Benoît Roux. A microscopic view of ion conduction through the k+ channel. *Proceedings of the National Academy of Sciences*, 100(15):8644–8648, 2003.

[37] Fatemeh Khalili-Araghi, Emad Tajkhorshid, and Klaus Schulten. Dynamics of k+ ion conduction through kv1.2. *Biophysical journal*, 91 6:L72–4, 2006.

[38] Tugba G Kucukkal, Feras Alsaiari, and Steven J Stuart. Modeling ion permeation in wild-type and mutant human α7 nachr ion channels. *Journal of Theoretical and Computational Chemistry*, 17(07):1850045, 2018.

[39] Simone Furini and Carmen Domene. Selectivity and permeation of alkali metal ions in k+-channels. *Journal of Molecular Biology*, 409(5):867 – 878, 2011.

[40] Ilsoo Kim and Toby W. Allen. On the selective ion binding hypothesis for potassium channels. *Proceedings of the National Academy of Sciences*, 108(44):17963–17968, 2011.

[41] Ameer Thompson, Ilsoo Kim, Timothy D Panosian, Tina M Iverson, Toby W Allen, and Crina Nimigean. Mechanism of potassium-channel selectivity revealed by na+ and li + binding sites within the kcsa pore. *Nature structural & molecular biology*, 16:1317–24, 11 2009.

[42] Morten Ø. Jensen, Vishwanath Jogini, David W. Borhani, Abba E. Leffler, Ron O. Dror, and David E. Shaw. Mechanism of voltage gating in potassium channels. *Science*, 336(6078):229–233, 2012.

[43] Albert C. Pan, Luis G. Cuello, Eduardo Perozo, and Benoît Roux. Thermodynamic coupling between activation and inactivation gating in potassium channels revealed by free energy molecular dynamics simulations. *The Journal of General Physiology*, 138(6):571–580, 2011.

[44] Fatemeh Khalili-Araghi, Vishwanath Jogini, Vladimir Yarov-Yarovoy, Emad Tajkhorshid, Benoît Roux, and Klaus Schulten. Calculation of the gating charge for the kv1.2 voltage-activated potassium channel. *Biophysical Journal*, 98(10):2189 – 2198, 2010.

[45] Radu A. Miron and Kristen A. Fichthorn. Accelerated molecular dynamics with the bond-boost method. *The Journal of Chemical Physics*, 119(12):6210–6216, 2003.

[46] Kristof M. Bal and Erik C. Neyts. Merging metadynamics into hyperdynamics: Accelerated molecular simulations reaching time scales from microseconds to seconds. *Journal of Chemical Theory and Computation*, 11(10):4545–4554, 2015. PMID: 26889516.

[47] Penghao Xiao, Juliana Duncan, Liang Zhang, and Graeme Henkelman. Ridge-based bias potentials to accelerate molecular dynamics. *The Journal of Chemical Physics*, 143(24):244104, 2015.

[48] A. L. Hodgkin and R. D. Keynes. The potassium permeability of a giant nerve fibre. *The Journal of Physiology*, 128(1):61–88, 1955.

[49] Sunhwan Jo, Taehoon Kim, and Wonpil Im. Automated builder and database of protein/membrane complexes for molecular dynamics simulations. *PloS one*, 2:e880, 02 2007.

[50] Orientations of proteins in membranes (opm) database. `http://opm.phar.umich.edu/`.

[51] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, 26(16):1781–1802, 2005.

[52] Ulrich Essmann, Lalith Perera, Max L. Berkowitz, Tom Darden, Hsing Lee, and Lee G. Pedersen. A smooth particle mesh Ewald method. *jcp*, 103(19):8577–8593, November 1995.

[53] Philip W. Fowler, Enrique Abad, Oliver Beckstein, and Mark S. P. Sansom. Energetics of multi-ion conduction pathways in potassium ion channels. *Journal of Chemical Theory and Computation*, 9(11):5176–5189, 2013. PMID: 24353479.

[54] A. Grossfield. Wham: the weighted histogram analysis method, version 2.0.9. `http://membrane.urmc.rochester.edu/content/wham`.

[55] David J. Wales and Harold A. Scheraga. Global optimization of clusters, crystals, and biomolecules. *Science*, 285(5432):1368–1372, 1999.

[56] L T Wille and J Vennik. Computational complexity of the ground-state determination of atomic clusters. *Journal of Physics A: Mathematical and General*, 18(8):L419–L422, jun 1985.

[57] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[58] John H. Holland. Genetic algorithms. *Scientific American*, 267(1):66–73, 1992.

[59] Optimization Genetic Algorithms in Search and Machine Learning. *Goldberg, David*. Addison-Wesley Professional, 1989.

[60] Sigurd Schelstraete and Henri Verschelde. Finding minimum-energy configurations of lennard-jones clusters using an effective potential. *The Journal of Physical Chemistry A*, 101(3):310–315, 1997.

[61] F. H. Stillinger and T. A. Weber. Nonlinear optimization simplified by hypersurface deformation. *Journal of Statistical Physics*, 52(5-6):1429–1445, Sep 1988.

[62] Lucjan Piela, Jaroslaw Kostrowicki, and Harold A. Scheraga. On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method. *The Journal of Physical Chemistry*, 93(8):3339–3346, 1989.

[63] Jaroslaw Pillardy and Lucjan Piela. Molecular dynamics on deformed potential energy hypersurfaces. *The Journal of Physical Chemistry*, 99(31):11805–11812, 1995.

[64] Wolfgang Wenzel and K. Hamacher. A stochastic tunneling approach for global minimization of complex potential energy landscapes. *Physical Review Letters*, 82, 03 1999.

[65] David J. Wales and Jonathan P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[66] Jonathan P. K. Doye and David J. Wales. Thermodynamics of global optimization. *Phys. Rev. Lett.*, 80:1357–1360, Feb 1998.

[67] Jonathan P. K. Doye, David J. Wales, and Mark A. Miller. Thermodynamics and the global optimization of lennard-jones clusters. *The Journal of Chemical Physics*, 109(19):8143–8153, 1998.

[68] Virtual Reality. *Steven M. LaValle*. Cambridge University Press, 2017.

[69] Andrew Gelman. Analysis of variance: Why it is more important than ever. *The Annals of Statistics*, 33(1):1–31, 2005.

[70] Matthew D. Wolf and Uzi Landman. Genetic algorithms for structural cluster optimization. *The Journal of Physical Chemistry A*, 102(30):6129–6137, 1998.

[71] Nathalie Tarrat, Mathias Rapacioli, Jérôme Cuny, Joseph Morillo, Jean-Louis Heully, and Fernand Spiegelman. Global optimization of neutral and charged 20- and 55-atom silver and gold clusters at the dftb level. *Computational and Theoretical Chemistry*, 1107:102 – 114, 2017. Structure prediction of nanoclusters from global optimization techniques: computational strategies.

[72] Andrea Grosso, Marco Locatelli, and Fabio Schoen. Solving molecular distance geometry problems by global optimization algorithms. *Comput. Optim. Appl.*, 43(1):23–37, May 2009.

[73] Fedor N. Novikov and Ghermes G. Chilov. Molecular docking: theoretical background, practical applications and perspectives. *Mendeleev Communications*, 19(5):237 – 242, 2009.

[74] A. Neumaier. Molecular Modeling of Proteins and Mathematical Prediction of Protein Structure. *SIAM Review*, 39:407–460, January 1997.

[75] Jingfa Liu, Yuanyuan Sun, Gang Li, Beibei Song, and Weibo Huang. Heuristic-based tabu search algorithm for folding two-dimensional ab off-lattice model proteins. *Computational Biology and Chemistry*, 47:142 – 148, 2013.

[76] J. M. García-Martínez, E. M. Garzón, J. M. Cecilia, H. Pérez-Sánchez, and P. M. Ortigosa. An efficient approach for solving the hp protein folding problem based on uego. *Journal of Mathematical Chemistry*, 53(3):794–806, Mar 2015.

[77] Andreas Krämer, Marco Hülsmann, Thorsten Koeddermann, and Dirk Reith. Automated parameterization of intermolecular pair potentials using global optimization techniques. *Computer Physics Communications*, 185:3228–3239, 12 2014.

[78] Mark Dittner, Julian Müller, Hasan Metin Aktulga, and Bernd Hartke. Efficient global optimization of reactive force-field parameters. *Journal of Computational Chemistry*, 36(20):1550–1561, 2015.

[79] Constantin Brif, Raj Chakrabarti, and Herschel Rabitz. Control of quantum phenomena: past, present and future. *New Journal of Physics*, 12(7):075008, jul 2010.

[80] Julien Jorda, Michael R. Sawaya, and Todd O. Yeates. *CrowdPhase*: crowdsourcing the phase problem. *Acta Crystallographica Section D*, 70(6):1538–1548, Jun 2014.

[81] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Snehal Gaikwad, Sungroh Yoon, Adrien Treuille, and Rhiju Das. Rna design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, 111(6):2122–2127, 2014.

[82] Jens Jakob W. H. Sørensen, Mads Kock Pedersen, Michael Munch, Pinja Haikka, Jesper Halkjær Jensen, Tilo Planke, Morten Ginnerup Andreasen, Miroslav Gajdacz, Klaus Mølmer, Andreas Lieberoth, and Jacob F. Sherson. Exploring the quantum speed limit with computer games. *Nature*, 532(7598):210–213, April 2016.

[83] Alexander Kawrykow, Gary Roumanis, Alfred Kam, Daniel Kwak, Clarence Leung, Chu Wu, Eleyine Zarour, Phylo players, Luis Sarmenta, Mathieu Blanchette, and Jérôme Waldispühl. Phylo: A citizen science approach for improving multiple sequence alignment. *PLOS ONE*, 7(3):1–9, 03 2012.

[84] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.