Doctoral Dissertation

博士論文

Edge expansion parallel cascade selection molecular dynamics simulation (eePaCS-MD) for investigating protein dynamics

（エッジ拡張型並列カスケード選択分子動力学法（eePaCS-MD）を用いたタンパク質の動的構造探索）

A Dissertation Submitted for the Degree of Doctor of Philosophy

July 2020

令和２年７月博士（理学）申請

Department of Physics, Graduate School of Science,

The University of Tokyo

東京大学大学院理学系研究科

物理学専攻

Kenichiro Takaba

鷹羽　健一郎

Abstract

Proteins are inherently dynamical molecules that undergo large-scale conformational changes to exert its functions. To investigate the high anisotropic nature of protein dynamics, Molecular dynamics (MD) simulation is an essential computational tool that can elucidate the conformational transitions of proteins, providing time-dependent information on protein fluctuation at atomic resolution. However, observing conformational changes relevant to biological functions remains a challenge because these events tend to occur stochastically in a time scale longer than feasible MD simulation time. To overcome this difficulty, many enhanced conformational sampling methods have been proposed. However, some of the methods require an external force to enhance the conformational transition, which does not necessarily guarantee that the obtained trajectories follow the lowest energy pathway. Other methods do not need such external forces but may require pre-test of simulations to determine the simulation parameters which can be cumbersome. Therefore, an enhanced sampling method that can simulate protein conformations relevant to biological functions without external forces and does not require cumbersome parameter setting is attractive. In addition, a method that can simulate protein conformations starting from a single structure without the prior knowledge of other conformational states can be valuable.

This thesis focuses on the development of a new enhanced conformational sampling method, edge expansion parallel cascade selection molecular dynamics (eePaCS-MD), to investigate the large-amplitude collective motions of proteins with a focus on domain motions. eePaCS-MD is an efficient adaptive sampling method which does not require prior knowledge of the conformational transitions or external forces to enhance the conformational sampling. eePaCS-MD takes advantage of the fact that large-amplitude fluctuations of many proteins can be described in terms of only a few principal components (PCs). In this method, multiple independent MD simulations are iteratively conducted from initial structures randomly selected from the vertices of a multi-dimensional PC subspace with new initial velocities to help the simulated system to overcome the energy barrier. This subspace is defined by an ensemble of protein conformations sampled during previous cycles of eePaCS-MD. The edges and vertices of the conformational subspace are determined by solving the “convex hull problem”.

The conformational sampling efficiency of eePaCS-MD was assessed for the open-close transitions of glutamine binding protein, maltose/maltodextrin binding protein, and adenylate kinase. The free energy landscape of open-to-closed conformational transitions of glutamine binding protein was obtained by constructing a Markov state model from trajectories generated by eePaCS-MD. The

obtained free energy landscape showed an energy barrier separating the open and closed states where the open state was suggested to be energetically more favorable than the closed state. To further enhance the conformational sampling efficiency, eePaCS-MD was combined with accelerated MD, where the total computational cost of observing the open-close transitions can be reduced at most 36% compared to the original eePaCS-MD method. eePaCS-MD is expected to offer 1-3 orders of magnitude shorter simulation time compared to conventional MD simulation.

Acknowledgments

I would like to first thank Prof. Akio Kitao of the School of Life Science and Technology at the Tokyo Institute of Technology. I am grateful for giving me the opportunity to come and study in your lab. The door to your office was always open and I really enjoyed having discussions with you. I have learned so much not just about the research field but also the attitude toward research.

I would also like to thank Dr. Takemura Kazuhiro and Dr. Duy Phuoc Tran for the thoughtful discussions and advice throughout my research, and I am gratefully indebted to your valuable comments on the thesis. I also thank Dr. Hiroaki Hata who always made me wanted to visit the lab. I would like to pay my regards to Shibayama Hidemi and Iwasa Chisato for the help and support of the document works, allowing me to concentrate on my work.

I wish to express my greatest gratitude to Prof. Munehito Arai of the Graduate School of Science at the University of Tokyo for becoming my supervisor after Prof. Kitao moved to Tokyo Institute of Technology during my course of PhD program. You have supported and encouraged me throughout my years of study.

List of Figures

List of Tables

Abbreviations and Acronyms

ADK	Adenylate kinase
aMD	Accelerated molecular dynamics
cMD	Conventional molecular dynamics
cPCA	Cartesian coordinate principal component analysis
dPCA	distance-based principal component analysis
eePaCS	edge expansion parallel cascade selection
ntPaCS	nontargeted parallel cascade selection
MBP	Maltose/maltodextrin binding protein
MD	Molecular dynamics
MSM	Markov state model
OFLOOD	Outlier flooding
PaCS	Parallel cascade selection
PCA	Principal component analysis
PDB	Protein Data Bank
QBP	Glutamine binding protein
RMSD	Root-mean-square deviation
SACS	Self-avoiding conformational sampling
SDS	Structural dissimilarity sampling

Table of Contents

Chapter 1
Introduction

1.1 General Introduction

Biomolecules are essential for all living organisms which are involved in many biological functions required to maintain life. Biomolecules include various macromolecules such as proteins, lipids, carbohydrates, and nucleic acids, as well as small molecules represented by natural products and metabolites. Therefore, their structural and dynamical properties are not simply important to understand the fundamental mechanism of life, but also relevant for industrial purposes such as in the field of therapeutic medicine, as proteins have been implicated in many human diseases.[1,2]

Among the important biomolecules, proteins are responsible for regulating the cellular environment, playing critical roles in the body. For example, membrane proteins can receive and transmit signals during cell-to-cell communications and transport molecules in and out of cells to maintain the cell environment. Messenger proteins, such as hormones, can act as chemical messengers that aid the communication between different cells and tissues. Enzyme proteins cause biochemical reactions necessary to coordinate our bodily functions such as digestion and muscle contraction. Antibodies are proteins that bind to specific particles, such as viruses and bacteria, and protect the body from harmful infections. Fibrous proteins help maintaining the shape of cells and tissues with rigidity and elasticity.

The diverse functions of proteins are strongly correlated with their unique three-dimensional (3D) structures. Protein chains are biopolymers composed of more than 20 different amino acids, each of which indicates a unique structural feature. The protein structures are often referred to four distinct structures (primary, secondary, tertiary, and quaternary structure) regarding their structural aspects. Primary structure refers to the amino acid sequence. Secondary structure is the local structure element or motif, such as α-helices and β-sheets, being maintained by hydrogen bonds of local amino acids. Tertiary structure is the overall shape of a protein which is generally stabilized by various attractive interactions, such as hydrophobic interactions and salt bridges. Quaternary structure is a complex structure formed by several protein molecules.

Currently more than hundred thousand 3D structures of proteins in atomic resolution are solved

using various experimental methods, such as X-ray crystallography and nuclear magnetic resonance etc., giving insights into their biological functions.[3,4] Since proteins are inherently dynamical molecules that undergo conformational transitions which occur at a wide range of timescales (Figure 1.1), it is essential to study their dynamical behaviors;[5–7] especially the large-scale conformational changes as they are relevant to biological functions.[8–10] The highly anisotropic nature of protein dynamics is the key to efficiently induce the conformational change in macromolecular crowded environments.[11] Proteins exist in an ensemble of conformations around their native states that can be characterized by a rugged-free energy landscape with many local energy minima.[12] Hence, conformational change relevant to biological functions are considered as slow processes because multiple energy barriers must be crossed to induce the conformational change.

Experimental techniques can provide insights into their dynamical properties but it may be difficult to provide the necessary detailed information about the underlying conformational ensembles. Molecular dynamics (MD) simulation has been widely used to elucidate the conformational transitions of proteins, providing time-dependent information on protein fluctuation at atomic resolution. However, observing conformational changes relevant to biological functions is challenging because these events tend to occur stochastically in a time scale longer than feasible MD simulation time. Furthermore, the computational cost of MD simulations will increase with the system size. To overcome this difficulty, many enhanced conformational sampling methods have been proposed. For example, Targeted-MD[13,14] can enhance the conformational transitions of two-end states where subset of atoms is guided towards the product state via a steering force but the obtained transition pathway may not necessarily follow the lowest energy pathway.[14] In multicanonical MD,[15,16] a weight function is introduced so that the probability distribution of the potential energy is uniform, leading to a random walk in energy space. Although this method allows the system to overcome potential energy barriers and explore a wide range of phase space, pre-test of simulations are required to determine the optimal weight. Replica exchange MD[17] is another widely known enhanced sampling method where independent simulations of the same system (replicas) with slightly different ensemble conditions, such as temperatures, are periodically swapped between replicas to efficiently overcome the energy barriers. However, the number of replicas will increase as the system size increases and require more computer resource to achieve efficient sampling.[18]

Although each enhanced sampling method has its own strength and is useful in different situations, a method that can simulate protein conformations relevant to biological functions without external forces and does not require cumbersome parameter setting is attractive. In addition, a method that can simulate protein conformations starting from a single structure without the prior knowledge of other conformational states can be valuable, for example situations where a novel

protein structure is solved and its conformational transitions are unknown.

This thesis focuses on the development of a new conformational sampling method, edge expansion parallel cascade selection molecular dynamics (eePaCS-MD),[19] to investigate the large-amplitude collective motions of proteins with a focus on domain motions. eePaCS-MD is an efficient adaptive sampling method that does not require prior knowledge of the conformational transitions or external forces to enhance the conformational sampling. eePaCS-MD takes advantage of the fact that large-amplitude fluctuations of many proteins can be described in terms of only a few principal components (PCs).[11,20,21] The high sampling efficiency of eePaCS-MD is achieved by repeating multiple independent short MD simulations from structures that are rigorously located at the edge of a multi-dimensional PC subspace with new initial velocities which helps the simulated system to overcome the energy barrier.[22]

eePaCS-MD is expected to help generate new mechanistic hypotheses and support experimental work to further validate the hypotheses. For example, one can remove a bound ligand from an experimentally determined protein structure and then simulate the bound and unbound systems, or replace the bound ligand with other ligands, to see how ligand binding affects the protein dynamics and its functions.[23,24] Conformational sampling of proteins in their apo state can be also useful to investigate possible protein conformations and search for novel binding sites to develop new therapeutic drugs.[25] In addition, one can mutate one or more amino acid residues in the protein to explain or predict the effect of mutations.[26,27] The molecular environment of a simulated protein, such as salt concentration and pressure, can be changed to address how protein dynamics and its functions are affected by the molecular environment.[28,29]

1.2 Thesis Outline

Chapter 2 – Methodology and Analysis of Molecular Dynamics Simulations

This is the introductory chapter of the thesis and presents background information about molecular dynamics simulations regarding empirical force fields, numerical integration, thermostats, and barostats. It also describes enhanced sampling techniques and general analysis methods used in molecular dynamics simulations.

Chapter 3 – Edge Expansion Parallel Cascade Selection Molecular Dynamics

This is the main chapter of the thesis where edge expansion parallel cascade selection molecular dynamics (eePaCS-MD) is proposed as an efficient conformational sampling method.[19] Cartesian coordinate principal component analysis (PCA) is utilized to select the initial structures for

resampling. The conformational sampling efficiency of eePaCS-MD was assessed for the open-close transitions of glutamine binding protein, maltose/maltodextrin binding protein, and adenylate kinase. The free energy landscape of open-to-closed conformational transitions of glutamine binding protein was obtained by constructing a Markov state model from trajectories generated by eePaCS-MD. Furthermore, combinations of eePaCS-MD with accelerated MD can further enhanced the conformational sampling efficiency by at most 36%. eePaCS-MD is expected to offer 1–3 orders of magnitude shorter simulation time compared to conventional MD simulation.

Chapter 4 – Comparison of eePaCS-MD Utilizing Cartesian Coordinate PCA and Distance-based PCA

In this chapter, conformational sampling efficiency of eePaCS-MD using Cartesian coordinate PCA and Cα distance-based PCA (dPCA) is compared. The open-close transitions of adenylate kinase were investigated to assess eePaCS-MD utilizing two different PCA methods. The conformational sampling efficiency of eePaCS-MD was not affected by choice of the PCA method. Considering the computational complexity of dPCA which quadratically scales with the number of atoms, I have concluded that eePaCS-MD utilizing Cartesian coordinate PCA as the first choice of the PCA method, although the optimal choice will depend on the target.

Chapter 5 – Conclusions and Perspectives

A summary of my thesis and future perspective is given about this research area.

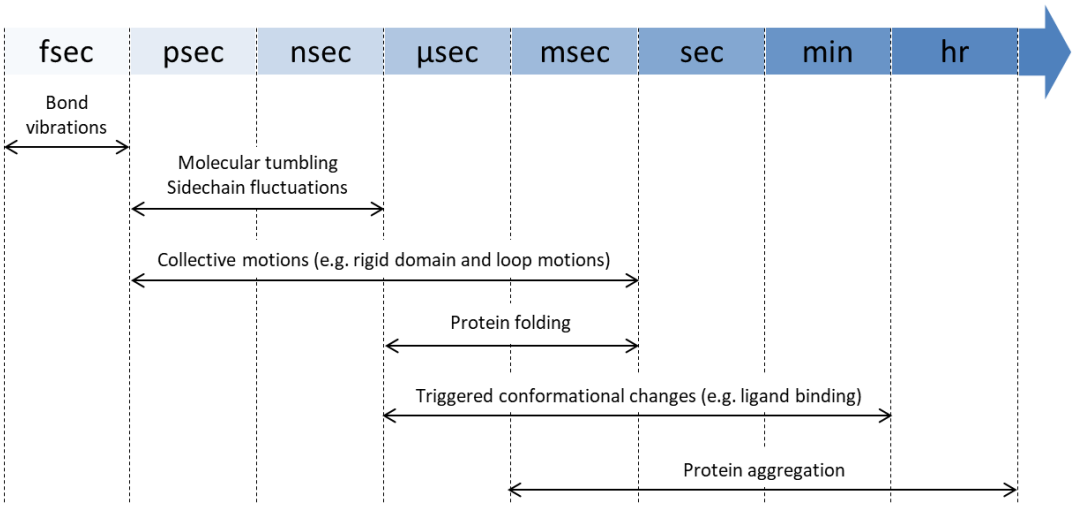

Figure 1.1: Typical timescale of protein dynamics.

Chapter 2
Methodology and Analysis of Molecular Dynamics Simulations

In this chapter, general introduction to molecular dynamics (MD) simulations and the computational techniques widely used to analyze trajectory data are introduced. Section 2.1 introduces the basics of MD simulations, such as empirical force fields, numerical integration and statistical ensembles. Section 2.2 discusses enhanced sampling methods to overcome the time limitation of conventional MD simulations where biomolecular processes of interest often exceeds the simulation time obtained by simple brute force simulations. Section 2.3 introduces computational techniques to analyze trajectory data.

2.1 Introduction

2.1.1 General introduction to molecular dynamics simulations

Molecular dynamics (MD) simulation is a computational approach widely used to elucidate the conformational transitions of biomolecular systems, such as proteins and nucleic acids, providing time-dependent information of protein dynamics at atomic resolution.[30–32] MD simulations can capture a wide variety of important biological processes, including conformational change,[8,23] ligand binding,[33,34] and protein folding.[35,36] The conformational sampling is driven by numerically integrating the Newtonian equations of motion:

$$m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} = \mathbf{F}_i, \qquad i = 1,2,\cdots N \tag{2.1}$$

$$\boldsymbol{F}_i = -\frac{dU}{d\boldsymbol{r}_i}, \qquad i = 1,2,\cdots N \tag{2.2}$$

where $\boldsymbol{r}_i$ and m_i represent the position and mass of atom i, respectively, $\boldsymbol{F}$ is the force which is derived from a given potential energy function $U(\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_\mathrm{N})$, and N is the total number of atoms of the system. The potential energy function is given by a set of parametrized functions called empirical force fields which will be discussed in Section 2.1.2.

The first all-atom MD simulation of a protein was performed in the late 1970s where bovine pancreatic trypsin inhibitor consisting of 58 residues was simulated *in vacuo*, for less than 10 ps.[37] Over the past years, improvements in algorithms, software, and computer hardware have allowed simulations of microsecond to millisecond timescales for systems with tens of thousands of atoms in solvated conditions.[23,35,38,39] Furthermore, simulations under cellular crowded environment provide the new atomic insights to how biological systems dynamically behave in reality.[40] The biomolecular system is often prepared from X-ray crystallography, nuclear magnetic resonance (NMR), cryo-electron microscopy, or homology modeling data. Experimentally solved protein structures are collected and distributed by the Worldwide Protein Data Bank (wwPDB),[3,4] which stores more than 160,000 structures to date and ~10,000 new structures are deposited annually (Figure 2.1). The advances in computational and structural biological field have led the use of MD simulations in the drug discovery industry, such as validation of ligand binding poses,[41] cryptic pocket predictions,[42,43] and protein-ligand binding free energy calculations.[44–46]

2.1.2 Potential energy function of biomolecular systems

2.1.2.1 Empirical force fields

Empirical force fields are sets of parameterized functions of atomic coordinates used to calculate the potential energy of a biomolecular system.[47–50] The force fields are parameterized by fitting and reproducing results of quantum mechanical calculations and experimental measurements. Commonly used force fields for biomolecular systems, such as AMBER,[51] CHARMM,[52] and OPLS[53], incorporate a similar functional form which consists of bonded and non-bonded energy terms (Figure 2.2). A general functional form of a force field is shown in Equation (2.3) and (2.4).

$$U_{total} = U_{bonds} + U_{angles} + U_{dihedrals} + U_{impropers} + U_{vdw} + U_{ele} \tag{2.3}$$

$$\begin{aligned} U_{total} = & \sum_{bonds} \frac{1}{2} k_b \left(r_{ij} - r_0\right)^2 + \sum_{angles} \frac{1}{2} k_\theta \left(\theta_{ijk} - \theta_0\right)^2 \\ & + \sum_{dihedrals} \frac{1}{2} k_\varphi \left[1 + \cos\left(n\varphi_{ijkl} - \gamma_n\right)\right] + \sum_{impropers} \frac{1}{2} k_\omega \left(\omega_{ijkl} - \omega_0\right)^2 \\ & + \sum_{i<j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{i<j} \frac{Q_i Q_j}{4\pi\varepsilon_0 r_{ij}} \end{aligned} \tag{2.4}$$

The bonded terms describe the interaction potentials between set of atoms that are covalently bonded, which include bond-stretching (U_{bonds}), bond angle-bending (U_{angles}), dihedral ($U_{dihedrals}$), and improper torsion ($U_{impropers}$) terms. The bond-stretching term is expressed as a

harmonic potential which corresponds to the first term in Equation (2.4), where k_b is the force constant, r_{ij} is the distance between two atoms i and j, and r_0 is the equilibrium bond distance. The bond angle-bending term is expressed similarly to those of the bond-stretching term. The dihedral angle is comprised of four consecutively connected atoms. A Fourier type expansion, as expressed as the third term in Equation (2.4), is used to characterize the dihedral potential where k_φ is the dihedral force constant (amplitude), n is dihedral periodicity, and γ_n is a phase of the dihedral angle φ_{ijkl}. Unlike the bonded and angle terms, the torsion term can have multiple energy maximum and minimum, and the torsion energies are often not so high to allow small deviations from an equilibrium structure. An improper dihedral is a special type of dihedral term used to maintain planarity in a molecular structure or to prevent transition to a configuration of opposite chirality, e.g., stereochemistry of a chiral carbon atom. The improper term can be expressed similarly to those of bonded and angle terms with a harmonic potential energy function. It is worth noting that improper dihedral term can be treated using the dihedral term when the periodicity is $n = 2$ and phase is $\gamma_n = 180$.

The non-bonded energy terms consist of two terms; the van der Waals interaction (U_{vdw}) and the electrostatic (U_{ele}) interaction. The van der Waals interaction is composed of repulsive and attractive interaction which is represented as a 6-12 Lennard-Jones potential given by:

$$U_{vdw}(r_{ij}) = 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{2.5}$$

where ε_{ij} and σ_{ij} represents the depth of the potential well and the distance at which the potential is zero, respectively, which are evaluated according to a mixing rule.[48] For example, in AMBER and CHARMM force fields, ε_{ij} and $\sigma_{ij}$are evaluated by geometric and arithmetic means, respectively. This approach is referred to as the Lorentz/Berthelot mixing rule as shown in Equation (2.6).

$$\begin{aligned} \varepsilon_{ij} &= \left(\varepsilon_{ii}\varepsilon_{jj}\right)^{1/2} \\ \sigma_{ij} &= \frac{1}{2}\left(\sigma_{ii} + \sigma_{jj}\right) \end{aligned} \tag{2.6}$$

Alternatively, OPLS force field applies the geometric mean for both ε_{ij} and σ_{ij}, thus other methods have been discussed elsewhere.[54] The r^{-12} describes the pauli exclusion which is a short-range repulsion force due to the overlap of electronic orbitals, required to prevent atoms of opposite charges from collapsing by the electrostatic attraction. The r^{-6} describes the weak

attractive force between two atoms due to interactions between permanent and induced dipoles. These interactions include the permanent dipole – permanent dipole (Keesom), permanent dipole – induced dipole (Debye), and induced dipole – induced dipole (London). The induced dipole – induced dipole interaction is also known as the dispersion interaction which is due to the instantaneous dipoles arising from fluctuations in the electronic distribution. The electrostatic (Coulomb) interaction between two charged atoms Q_i and Q_j is expressed as:

$$U_{ele}(r_{ij}) = \frac{Q_i Q_j}{4\pi\varepsilon_0 r_{ij}} \tag{2.7}$$

where ε_0 is the vacuum dielectric permittivity. In theory, multipole expansion, e.g., dipole and quadrupole, is required to accurately represent the quantum mechanical electrostatic potential. However, empirical force fields try to approximate this multipole expansion by assigning point charges localized at the nuclei of atoms, in order to reproduce the same electrostatic potential that would be given by the true electronic structure and electron density distribution.

2.1.2.2 Non-bonded exclusions

In practice, empirical force fields exclude the interactions between atoms separated by two- and three- consecutively covalent-bonding atoms (so-called 1-2 and 1-3 interaction) because they are redundant with the bond-stretching and angle-bending terms to reproduce the bond and angle geometries. However, the 1-4 interactions are usually kept to better reproduce the torsion barriers, for example by scaling down the 1-4 interaction of the van der Waals and electrostatic interactions.

2.1.2.3 Treatment of long-range interactions

The calculations of non-bonded pairwise interactions (van der Waals and electrostatic interactions) are the most time-consuming part of the simulation. A smooth truncation scheme can be applied to drive the non-bonded interactions to zero at a finite distance using cutoff distances to reduce the total computational cost. This may be acceptable for van der Waals interactions because pairwise interactions decay quickly with respect to r^{-6} where r is the distance between the interacting atoms. However, electrostatic interactions are fundamentally long-range and the contributions of energies and forces in the system from such a distant range can be non-trivial, hence severe errors can arise from neglecting electrostatic interactions beyond some cutoff distance. In practice, electrostatic interactions are widely calculated by introducing a periodic boundary condition which tiles repeating copies of the system. This makes it possible to include all long-range interactions by

summing over the real and reciprocal space using Particle-mesh Ewald (PME) method,[55] where the periodicity is used to take into account long-range electrostatic interactions including those of particles with their own periodic images.

2.1.3 Numerical integration

2.1.3.1 Verlet and velocity Verlet algorithms

To propagate the system, one starts with assigning an initial velocity to each atom. The initial velocity is usually assigned based on the Maxwell-Boltzmann distribution:

$$p(\boldsymbol{v}_i) = \left(\frac{m_i}{2\pi k_B T}\right)^{\frac{3}{2}} \exp\left(-\frac{m_i \boldsymbol{v}_i{}^2}{2\pi k_B T}\right) \tag{2.8}$$

where m_i and $\boldsymbol{v}_i = \left(v_i^{(x)}, v_i^{(y)}, v_i^{(z)}\right)$ are the mass and velocity of atom i, respectively, k_B is Boltzmann constant, and T is the given temperature. After the initial velocity assignment, the system is propagated by numerically integrating the Newtonian equations of motion. One of the most widely used numerical integrator is the velocity Verlet algorithm[56] which is a modification of the original Verlet algorithm.[57] The essential idea is to divide the integration step into small steps, each separated by a fixed time step Δt, and assumes that the positions and dynamics properties can be approximated in Taylor series expansion.

The Verlet algorithm considers the sum of the Taylor expansions corresponding to forward and reversed time steps of Δt.

$$\begin{cases} \boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i(t) + \Delta t \dot{\boldsymbol{r}}_i(t) + \frac{\Delta t^2}{2!}\ddot{\boldsymbol{r}}_i(t) + \frac{\Delta t^3}{3!}\dddot{\boldsymbol{r}}_i(t) + O(\Delta t^4) \\ \boldsymbol{r}_i(t-\Delta t) = \boldsymbol{r}_i(t) - \Delta t \dot{\boldsymbol{r}}_i(t) + \frac{\Delta t^2}{2!}\ddot{\boldsymbol{r}}_i(t) - \frac{\Delta t^3}{3!}\dddot{\boldsymbol{r}}_i(t) + O(\Delta t^4) \end{cases} \tag{2.9}$$

When these two expansions are added and rearranged, the equations give us:

$$\boldsymbol{r}_i(t+\Delta t) = 2\boldsymbol{r}_i(t) - \boldsymbol{r}_i(t-\Delta t) + \frac{\Delta t^2}{m_i}\boldsymbol{F}_i(t) + O(\Delta t^4) \tag{2.10}$$

$$\dot{\boldsymbol{r}}_i(t) = \frac{\boldsymbol{r}_i(t+\Delta t) - \boldsymbol{r}_i(t-\Delta t)}{2\Delta t} + O(\Delta t^2) \tag{2.11}$$

Equation (2.10) and (2.11) corresponds to the Verlet algorithm. Note that higher-order terms in the Taylor expansion are ignored for both positions and velocities. Although the velocities can be calculated with a second-order accuracy, i.e. $O(\Delta t^2)$, this is inconvenient as it may introduce some additional error in the energy and other properties which depend on the velocity. Another potential drawback is that positions are obtained by adding a small term, $\frac{\Delta t^2}{m_i}\boldsymbol{F}_i(t)$, to much larger terms which may lack numerical precision and give rise to substantial round off errors.

A related and more commonly used method is the velocity Verlet method where the positions and velocities are given by:

$$\boldsymbol{r}_i(t+\Delta t) = \boldsymbol{r}_i(t) + \Delta t \boldsymbol{v}_i(t) + \frac{\Delta t^2}{2m_i}\boldsymbol{F}_i(t) + O(\Delta t^4) \tag{2.12}$$

$$\boldsymbol{v}_i(t) = \boldsymbol{v}_i(t-\Delta t) + \frac{\Delta t}{m_i}\frac{\boldsymbol{F}_i(t) + \boldsymbol{F}_i(t-\Delta t)}{2} \tag{2.13}$$

This method is mathematically equivalent to the original Verlet algorithm, except the positions and velocities are given at the same time. The global error associated with the velocity Verlet algorithm is at the fourth order for the position, which is same as the Verlet algorithm. However, the precision is not influenced by the velocity as Equation (2.13) is equivalent to Equation (2.11). The round off errors can be minimized because the positions and velocities at $n\Delta t$ can be obtained by first summing the small terms over n steps, and then add-on to the larger terms.

$$\boldsymbol{r}_i(t+n\Delta t) = \boldsymbol{r}_i(t) + \Delta t \sum_{k=1}^{n} \boldsymbol{v}_i[t+(k-1)\Delta t] + \frac{\Delta t^2}{2m_i}\sum_{k=1}^{n}\boldsymbol{F}_i[t+(k-1)\Delta t] \tag{2.14}$$

$$\boldsymbol{v}_i(t+n\Delta t) = \boldsymbol{v}_i(t) + \frac{\Delta t}{m_i}\sum_{k=1}^{n}\frac{\boldsymbol{F}_i[t+(k-1)\Delta t] + \boldsymbol{F}_i(t+k\Delta t)}{2} \tag{2.15}$$

2.1.3.2 Choice of time step

The choice of time step Δt is important to achieve numerical stability and efficient sampling. If a large time step is used, the motion of molecules becomes unstable due to big errors occurring in the integration of equation of motion. If the time step is too small, then long computational time will be required to propagate the motion. The size of a time step is constrained by the time scale of the highest frequency motion in the system, which is typically the bond vibrations involving the hydrogen atoms. In general, time step of 1 fs (10^{-15} sec) is commonly used which is a magnitude

lower than the fastest time scale in the system. However, for computational efficiency, longer time step of 2 fs can be applied by constraining the hydrogen bond lengths and treating the water molecules as rigid body using algorithms such as SHAKE[58] and SETTLE.[59] Furthermore, recent advances in algorithms can provide even longer time step of ~4 fs.[60]

2.1.4 Application of MD simulations in NVT and NPT ensembles

2.1.4.1 Statistical ensembles

In classical mechanics, a state is a specific microscopic configuration of a system in a phase space. A certain state can be characterized by positions $\boldsymbol{q}$ and momenta $\boldsymbol{p}$ variables of the system. The Hamiltonian, H, which describes the total energy of the state is given by:

$$H(\boldsymbol{q},\boldsymbol{p}) = \sum_{i=1}^{N} \frac{{p_i}^2}{2m_i} + U(\boldsymbol{q}) \tag{2.16}$$

where the first term is the kinetic energy and $U(\boldsymbol{q})$ is the potential energy. A statistical ensemble is a collection of various states of an equilibrium macroscopic system under constraint conditions such as temperature, pressure, and volume. There exist different ensembles with different characteristics and some of the physically important ensembles are:

Microcanonical ensemble (NVE)

Microcanonical ensemble (NVE) is the statistical ensemble where the number of particles N, volume V, and energy E are constant. This ensemble is an approximation for an isolated system which cannot exchange energy or particles. The probability of all microstates of an isolated equilibrium system is considered equal and is given by:

$$\mathrm{P}(\boldsymbol{q},\boldsymbol{p}) \propto \delta[H(\boldsymbol{q},\boldsymbol{p}) - E] \tag{2.17}$$

Canonical ensemble (NVT)

Canonical ensemble (NVT) is the statistical ensemble where the number of particles N and volume V in the system is fixed, but the system is coupled to a heat bath (reservoir) at temperature T. Different microstates of the system can have different energies. The probability of finding a microstate i with energy E_i is given by:

$$\mathrm{P}(\boldsymbol{q},\boldsymbol{p}) \propto \exp(-\beta E_i) \tag{2.18}$$

where $\beta = {}^{1}/_{k_B T}$ and k_B is the Boltzmann constant. The averaged observable quantity *A* is given by:

$$\langle A \rangle = \frac{\sum_i A_i \exp(-\beta E_i)}{Z} \tag{2.19}$$

where $Z = \sum_i \exp(-\beta E_i)$ is the partition function and A_i is the microscopic property at state i.

Isothermal-isobaric ensemble (NPT)

Isothermal-isobaric ensemble (NPT) is the statistical ensemble where the number of particles N, pressure P, and temperature T are fixed, and allows the volume V and the energy to fluctuate. The isothermal-isobaric ensemble is one of most widely used ensembles as most of the real experiments are carried out under controlled conditions of temperature and pressure. The probability of finding a microstate i with energy E_i is given by:

$$\mathrm{P}(\boldsymbol{q}, \boldsymbol{p}) \propto \exp\{-\beta(E_i + PV)\} \tag{2.20}$$

Grand canonical ensemble (μVT)

Grand canonical ensemble (μVT) is the statistical ensemble where the volume V, temperature T, and chemical potential μ are fixed, but the number of particles and energy can exchange with the surrounding bath. This ensemble is particularly applicable to systems such as chemical reactions where the number of particles varies. The probability of finding a microstate i with energy E_i is given by:

$$\mathrm{P}(\boldsymbol{q}, \boldsymbol{p}) \propto \exp(-\beta E_i + \mu \beta N_i) \tag{2.21}$$

2.1.4.2 Thermostats and Barostats

Let us first consider a Hamiltonian system described by Equation (2.16). The Hamiltonian's canonical equation, which is the generalized form of Newton's equations of motion, is given by:

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} = \frac{p_i}{m_i} \tag{2.22}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} = -\frac{\partial U}{\partial q_i} \tag{2.23}$$

When the Hamiltonian system is simply solved using numerical integration discussed in Section 2.1.3, the generated ensemble is an NVE ensemble since $\frac{dH}{dt} = 0$ where the total energy is conserved. This is not the desired ensemble for biomolecular systems because it often makes sense to simulate systems used for experimental measurements, such as those at constant temperature and pressure. Generally, if such conditions are to be maintained, some thermostat and barostat algorithms need to be employed.[61]

Thermostats:

The temperature of a molecular dynamics simulation and the kinetic energies can be related using the equipartition theorem which is given by:

$$\langle \sum_{i=1}^{N} \frac{1}{2} m_i \boldsymbol{v}_i{}^2 \rangle = \frac{3}{2} N k_B T \tag{2.24}$$

where the angle brackets indicate the time-averaged quantity. When the temperature is calculated from a single snapshot, this quantity is referred to as the instantaneous temperature. The instantaneous temperature is not always equal to the target temperature but undergoes fluctuations around the target temperature. There are several methods employed by the thermostat algorithm to control the temperature.

One of the simplest thermostats to implement is the velocity rescaling method.[62] This method scales the velocities so that the temperature reaches the desired temperature. However, this approach does not allow fluctuations in temperature and the generated samples are isokinetic ensemble rather than the canonical ensemble. Berendsen thermostat,[63] also known as the weak coupling thermostat, is similar to the velocity rescaling approach but the temperature is allowed to slowly approach the target temperature. The velocities are scaled with a certain interval, such that the rate of temperature change is proportional to the difference in temperature, i.e., $\frac{dT}{dt} = -\frac{1}{\tau}(T - T_0)$, where τ is the relaxation time which determines how tightly the temperature bath and the system are coupled together and T_0 is the target temperature. The scaling factor is given by:

$$\lambda = 1 + \frac{\Delta t}{2\tau}\left(\frac{T_0}{T} - 1\right) \tag{2.25}$$

Although the Berendsen thermostat allows temperature fluctuations, the generated ensembles are not precisely the canonical ensemble.

An alternative approach is the Langevin thermostat[64] which controls the temperature to a reference temperature by inserting friction and stochastic terms in the equation of motion. The Langevin dynamics is given by:

$$m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} = F_i - m_i \gamma_i \frac{d\boldsymbol{r}_i}{dt} + R_i(t) \tag{2.26}$$

where γ_i is the collision frequency which determines the strength of the coupling to the heat bath. The stochastic force $R_i(t)$ is assumed to be uncorrelated with the positions and velocities of the particles and to be Gaussian with a mean zero and variance given by:

$$\langle R_i(t) R_j(t+\tau) \rangle = 2 m_i \gamma_i k_B T \delta_{ij} \delta(\tau) \tag{2.27}$$

The Langevin equation is known to achieve isothermal condition; however, the collision frequency is an effective friction coefficient of the system and should be carefully considered to maintain the temperature without significantly perturbing the dynamics of the system, where the dynamics will become microcanonical when $\gamma_i = 0$.

Another widely known approach is the Nosé-Hoover thermostat[65–67] which utilize an extended system to control the temperature. The general idea is to consider the heat bath as part of the system by introducing a fictious coordinate s, associated with an effective mass Q and conjugated momentum p_s of s. The equation of motion utilizing Nosé-Hoover thermostat is given by:

$$m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} = \boldsymbol{F}_i - m_i \xi \frac{d\boldsymbol{r}_i}{dt} \tag{2.28}$$

$$\frac{d\xi}{dt} = \frac{g k_B}{Q} (T(t) - T_0) \tag{2.29}$$

where ξ is a friction coefficient defined as $\xi = \frac{p_s}{Q}$, g is the number of degrees of freedom of the system, $T(t)$ is the instantaneous temperature defined as $T(t) = \frac{1}{g k_B} \sum_{i=1}^{N} \frac{\boldsymbol{p}_i{}^2}{m_i}$ and T_0 is the target temperature. Although the Nosé-Hoover thermostat is known to produce the canonical ensemble, the

temperature may be poorly controlled when the effective mass Q is too large, thus the dynamics of the system will become microcanonical when $Q \to \infty$.

Barostats:

The pressure can be measured using the virial theorem:

$$P = \frac{2}{3V}(\langle E_k \rangle - \langle \Xi \rangle) \tag{2.30}$$

where V is the volume of the system and E_k is the kinetic energy. Ξ is the internal virial for pair-additive potentials and is given by:

$$\Xi = -\frac{1}{2}\sum_{i<j} \boldsymbol{r}_{ij} \cdot \boldsymbol{F}_{ij} \tag{2.31}$$

$\boldsymbol{F}_{ij}$ is the force on particle i due to particle j, and $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$. These formulas give pressure as a time-averaged quantity and the instantaneous pressure is calculated using a single snapshot. The pressure will not always be equal to the target pressure but undergoes fluctuations around the target pressure.[68] There are several methods employed by the barostat algorithm to control the pressure.

The simplest approach is the volume rescaling where the volume of the system is modified such that the instantaneous pressure is exactly equal to the target pressure. This approach does not properly generate the isothermal-isobaric ensemble. Berendsen barostat[63] is analogous to the Berendsen thermostat where the pressure is weakly coupled to a "pressure bath" which slowly approaches to the target pressure. To maintain the system to a desired pressure, the atomic coordinates and volume are scaled periodically. The rate of change of pressure is proportional to the difference in pressure which is given by:

$$\frac{dP}{dt} = -\frac{1}{\tau}(P - P_0) \tag{2.32}$$

where τ is the relaxation time which determines how tightly the pressure bath and the system are coupled together and P_0 is the target pressure. Suppose we have an isotropic system with volume $V = L^3$ where L represents the box length of the system. An extra term $\alpha \boldsymbol{r}_i$ is added to the equation of motion which is given by:

$$\dot{\boldsymbol{r}}_i = \frac{\boldsymbol{p}_i}{m_i} + \alpha \boldsymbol{r}_i \tag{2.33}$$

where α is determined so that the instantaneous pressure is equivalent to the target pressure. The coordinates of the particles are given in terms of scaled variables $\tilde{\boldsymbol{r}}_i$:

$$\boldsymbol{r}_i = V^{\frac{1}{3}} \tilde{\boldsymbol{r}}_i \tag{2.34}$$

and the time derivative is given by:

$$\dot{\boldsymbol{r}}_i = V^{\frac{1}{3}} \dot{\tilde{\boldsymbol{r}}}_i + \frac{\dot{V}}{3V} \boldsymbol{r}_i = \frac{\boldsymbol{p}_i}{m_i} + \frac{\dot{V}}{3V} \boldsymbol{r}_i \tag{2.35}$$

With Equations (2.33) and (2.35), the volume change can be expressed as:

$$\dot{V} = 3\alpha V \tag{2.36}$$

The pressure change can be related to the isothermal compressibility β:

$$\frac{dP}{dt} = -\frac{1}{\beta V} \frac{dV}{dt} = -\frac{3\alpha}{\beta} \tag{2.37}$$

Hence, from Equations (2.32) and (2.37), the equation of motion in Equation (2.33) can be rewritten as:

$$\dot{\boldsymbol{r}}_i = \frac{\boldsymbol{p}_i}{m_i} + \frac{\beta}{3\tau}(P - P_0)\boldsymbol{r}_i \tag{2.38}$$

This represents a proportional scaling of coordinates and box length per time step from $\boldsymbol{r}_i$ to $\mu \boldsymbol{r}_i$ and L to μL, where μ is the scaling factor given by:

$$\mu = 1 + \frac{\beta \Delta t}{3\tau}(P - P_0) \cong \left[1 + \frac{\beta \Delta t}{\tau}(P - P_0)\right]^{\frac{1}{3}} \tag{2.39}$$

The value of the compressibility does not have to be precisely known and the relaxation time appears only as a ratio in the dynamics. In practice, compressibility of liquid water is often used. Many applications can be found that utilize Berendsen barostat[69–71] as this method will approach the target pressure realistically; however the ensemble it is sampling from is not well defined and cannot be guaranteed to be NPT ensemble.[68,72]

Alternatively, extended ensemble barostat methods, such as Parrinello-Rahman,[73] can be applied to control the pressure. The system is coupled to a fictitious pressure bath by adding an additional degree of freedom to the equations of motion which allows the shape of the simulation box to change during the simulation. Although the algorithm yields the correct ensemble, it is not recommended for equilibrium processes as it will not behave well if not near the target pressure.

2.2 Enhanced sampling methods

Conformational transitions relevant to biological functions are challenging to observe because these events tend to occur stochastically as rare events and the time scale of such phenomena can simply exceed feasible computational time simulated by brute force simulations, even if significant computational resources are employed. To overcome this difficulty, a wide variety of enhanced sampling techniques have been proposed to capture long-timescale events, where each technique can be useful for different situations. Some of the widely used methods are introduced.

2.2.1 Targeted MD

Targeted MD[13,14] is a class of nonequilibrium simulation which enhances the conformational transitions of two-end states (reactant and product) where a subset of atoms is guided towards the product state via a steering force. The force on each atom is derived from the gradient of the potential energy which is given by:

$$U_{TMD} = \frac{1}{2}k[RMSD(t) - RMSD_0(t)]^2 \tag{2.40}$$

where k is the spring constant, $RMSD(t)$ is the root-mean-square deviation (RMSD) at time step t measured from the product structure. $RMSD_0(t)$ is the prescribed RMSD at time step t which is slowly decreased to zero during the simulation. Although this method is convenient, the external force should be carefully set as the transition pathway may not necessarily follow the lowest energy pathway and yield irreversible pathways that are rarely accessible to the system at normal temperature.[14]

The free energy profile (potential of mean force) along the conformational transition pathway can be obtained using Jarzynski's equality[74] where the transition is characterized by the amount of work required to drive the transition.

$$\Delta F = -\frac{1}{\beta}\ln\langle e^{-\beta W}\rangle \tag{2.41}$$

Here, the bracket denotes an ensemble average over multiple independent simulations, and W is the work performed on the system by an external force. Since the exponential average can be dominated

by small values of the work which occur rarely, the logarithm of an exponential term can be expanded in terms of cumulants to reduce the exponential noise.[75,76]

2.2.2 Accelerated MD

In accelerated MD[77] the sampling efficiency is enhanced by increasing the escape rates from the potential wells (Figure 2.3). This is achieved by modifying the true potential $V(\boldsymbol{r})$ via a continuous non-negative boost potential $\Delta V(\boldsymbol{r})$ while maintaining the underlying shape of the true potential energy surface. The modified potential, $V^*(\boldsymbol{r}) = V(\boldsymbol{r}) + \Delta V(\boldsymbol{r})$, is described as:

$$\begin{cases} V^*(r) = V(r) + \dfrac{\big(E - V(r)\big)^2}{\alpha + \big(E - V(r)\big)}, & V(r) < E \\ V^*(r) = V(r), & V(r) \geq E \end{cases} \tag{2.42}$$

and the boost potential $\Delta V(\boldsymbol{r})$ is given by:

$$\Delta V(\boldsymbol{r}) = \frac{\big(E - V(\boldsymbol{r})\big)^2}{\alpha + \big(E - V(\boldsymbol{r})\big)}, \qquad V(r) < E \tag{2.43}$$

Here E is the potential energy threshold and α is the tuning parameter which determines the depth of the modified potential energy.

The boost potential can be applied to either the dihedral potential energy, total potential energy, or dihedral and total potential energy. The potential boost parameters associated with α can be defined by the number of protein residues, total number of atoms, and the average dihedral and/or total potential energies calculated from short conventional MD simulations.[78,79] The canonical ensemble distribution, $p(A)$, along a selected reaction coordinate $A(\boldsymbol{r})$ can be recovered by reweighting the probability distribution, $p^*(A)$, sampled by accelerated MD which is given by:

$$p(A_j) = p^*(A_j) \frac{\langle e^{\beta \Delta V(r)} \rangle_j}{\sum_{j=1}^{M} \langle e^{\beta \Delta V(r)} \rangle_j} \tag{2.44}$$

where M is the number of bins, $\langle e^{\beta \Delta V(r)} \rangle_j$ is the ensemble-averaged Boltzmann factor of $\Delta V(r)$ found in the j^{th} bin.[79] The exponential term can be approximated by Maclaruin series expansion or cumulant expansion to reduce the noise as the Boltzmann factor can be dominated by high boost potential values. The reweighted potential of mean force can be calculated as:

$$F(A_j) = -\frac{1}{\beta}\ln p(A_j) \tag{2.45}$$

2.2.3 Metadynamics

Metadynamics[80] enhances the rare-events occurrence by discouraging the system from revisiting the same phase space by introducing a repulsive bias potential to the original potential. The original potential $V(\boldsymbol{r})$ is modified by a history-dependent Gaussian potential deposited along the collective variable (CV) space which is expected to be suitable in describing the process of interest. The Gaussian potential is added every time interval τ_G and the biasing potential at time t is given by:

$$V_G(s(\boldsymbol{r}), t) = w \sum_{\substack{t'=\tau_G, 2\tau_G, 3\tau_G, \cdots \\ t'<t}} exp\left\{-\frac{[s(r) - s(r_G(t'))]^2}{2{\delta_s}^2}\right\} \tag{2.46}$$

where w and δ_s are the height and the width of the Gaussian. $s(\boldsymbol{r})$ and $\boldsymbol{r}_G(t')$ denote the CV and the trajectory of the system under the modified potential $V + V_G$, respectively. As the bias potential accumulates and fills the potential wells, the system is able to move in a less-barrier manner among the different states. The free energy profile along the CV can be reconstructed by simply changing the sign of Equation (2.46).

To achieve reasonable and accurate free energy profile, the height of the Gaussians must be sufficiently small compared to the main free energy barrier and the bias should not be added too frequently in time. This method is effective for exploring few CVs, however the performance can deteriorate rapidly with dimensionality because the computational effort required to discourage the visiting phase space will increase with the number of CVs. Deciding when to terminate a metadynamics simulation may not be straightforward as Gaussian potentials are added during the entire course of the simulation. As a result, the system will be pushed to explore high-energy regions and the calculated free energy will typically fluctuate around the correct value. Variants of metadynamics have been proposed to overcome these difficulties such as well-tempered metadyanics[81] where the height of the Gaussian potential decreases over time showing better convergence.

2.2.4 Multicanonical MD simulation

In multicanonical MD (McMD) simulation[15,16] the conformational sampling is enhanced by introducing a weight function so that the probability distribution of the potential energy $P_{mc}(E,T_0)$ is uniform.

$$P_{mc}(E,T_0) \propto n(E)e^{-W(E,T_0)} = const \tag{2.47}$$

Here E is the potential energy of the system, T_0 is the simulation temperature, $n(E)$ is the density of states, and $W(E,T_0)$ is the weight function. The flat energy probability distribution leads to random walk in energy space, allowing the system to overcome potential energy barriers and explore a wide range of phase space. In general, temperature T_0 is set to a sufficiently high temperature so that the conformation can overcome energy barriers in the conformational space. The multicanonical distribution can be reweighted to reproduce the canonical distribution at an arbitrary temperature T, if the weight function is accurately estimated, and is given by:

$$P_c(E,T) \propto P_{mc}(E,T_0)e^{-\beta E+W(E,T_0)} \tag{2.48}$$

Thus, the weight function is defined as

$$W(E) = \ln[n(E)] = \beta_0 E + \ln[P_c(E,T_0)] \tag{2.49}$$

$P_c(E,T_0)$ is the canonical energy distribution at temperature T_0. Since $W(E,T_0)$, i.e., $P_c(E,T_0)$, is not known *a priori*, the weight function is estimated and updated through iterative runs of McMD simulations until a flat distribution can be obtained from an converged estimate value of $P_c(E,T_0)$:

$$W^{(i+1)}(E) = W^{(i)}(E) + \ln P_{mc}^{(i)}(E,T_0) \tag{2.50}$$

where the i^{th} multicanonical distribution $P_{mc}^{(i)}(E,T_0)$ is obtained with the weight function $W^{(i)}(E)$. In practice, it is impossible to obtain an ideal weight factor that completely flattens the potential energy distribution. One criterion for a satisfactory weight factor is that as long as a random walk in potential energy space is achieved, the probability distribution $P_{mc}(E)$ does not have to be strictly flat but with a tolerance of an order of magnitude deviation.[82]

2.2.5 Replica exchange molecular dynamics

Replica exchange molecular dynamics (REMD)[17] seeks to enhance the conformational sampling by running independent simulations of the same system (replicas) with slightly different ensemble conditions and periodically swap replicas to efficiently overcome the energy barriers (Figure 2.4). In temperature REMD (T-REMD), also known as the parallel tempering, replicas with different temperatures are simulated parallel and swapping of neighboring replicas is attempted periodically with a given probability based on the Metropolis criterion that satisfies the detailed balance for swapping temperatures:

$$P_{i,j} = min\{1, exp[-(\beta_i - \beta_j)(E_j - E_i)]\} \tag{2.51}$$

where E is the potential energy, and subscripts i and j represent different replicas, respectively. To achieve optimal sampling performance, distribution of temperatures and number of replicas should be chosen with care. The highest temperature should be high enough to ensure that no replicas are trapped in local energy minima. For efficiency, each replica should spend equal amount of time at each temperature, suggesting that the temperatures should be chosen in a way that gives similar acceptance ratio between adjacent replicas. It was demonstrated that exchange acceptance probability of ~20% yields optimal performance;[83,84] however acceptance ratios of >10% is still acceptable.[17] Finding the optimal temperature distribution and number of replicas are non-trivial and several suggestions for setting these parameters have been offered.[85–87]

After a T-REMD simulation, unbiased populations of different substates can be obtained from the reference temperature of replicas. In addition, it is also possible to combine information from multiple replicas by performing a weighted-histogram analysis (WHAM)[17,88,89] to obtain the expectation value of any physical quantity at an arbitrary temperature. It should also be noted that analogous to T-REMD, replica exchange can be performed with order parameters other than temperatures, such as the use of different Hamiltonians (H-REMD).[18]

2.3 General analysis methods used in molecular dynamics simulations

2.3.1 Root-mean-square deviation and root-mean-square fluctuation

Root-mean-square deviation (RMSD) and root-mean-square fluctuation (RMSF) are frequently used measurements to give information on different aspects of biomolecular ensembles.[90] It often makes sense to compare the internal motions of the system; hence translational and rotation of atoms are removed from the system by superposing to a reference coordinate in advance of the analysis. RMSD is the average distance between the subset of atoms which is given by:

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{r} - \boldsymbol{r}_0)^2} \tag{2.52}$$

where n is number of atoms, $\boldsymbol{r}$ is the atomic coordinate, and $\boldsymbol{r}_0$ is the coordinate of the reference structure. RMSD is useful to measure the global similarity of two states and monitor the structural change during the simulation. RMSF is analogous to RMSD, which provides information about the local structural dynamics. RMSF is defined as the root mean-square-average distance between an atom and its average position in a given set of structures.

$$RMSF_k = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left\|\boldsymbol{r}_{i,k} - \langle \boldsymbol{r} \rangle_k\right\|^2} \tag{2.53}$$

Here $\langle \boldsymbol{r} \rangle_k$ is the average position of atom k over m structures. RMSF can be used to study protein dynamics such as protein flexibility, thermal stability, and prediction of disordered regions as it is related with B-factors in X-ray experiments which are given by:

$$RMSF_k = \sqrt{\frac{3B_k}{8\pi^2}} \tag{2.54}$$

2.3.2 Principal component analysis

Principal component analysis (PCA) is a multivariate statistical technique used to reduce the dimensionality of a biomolecular system.[20,21] The PCA represents a linear transformation that diagonalizes the covariance matrix so that the instantaneous linear correlations among the variables are removed. To obtain the internal motion of the system, translational and overall rotation from the trajectories are removed by superposing to a reference coordinate, which is often chosen to be the average coordinate determined self-consistently, prior to performing PCA.[91] Commonly, PCA of a MD simulation is performed by diagonalizing the (mass-weighted) covariance matrix of the atomic Cartesian coordinates, $\mathbf{A}$, where the element a_{ij} of matrix $\mathbf{A}$ is given by:

$$a_{ij} = \langle (q_i - \langle q_i \rangle)(q_j - \langle q_j \rangle) \rangle \tag{2.55}$$

where q_i and q_j represent the Cartesian coordinates of atom i and j, respectively, and the bracket denotes an average over ensemble of conformers. Diagonalization of matrix $\mathbf{A}$, i.e., $\mathbf{AV} = \mathbf{V\xi}$, leads to eigenvectors $\mathbf{V}$ and eigenvalues $\boldsymbol{\xi}$, which describes the direction of the motion and the magnitude along the corresponding eigenvector (principal component), respectively. Since the eigenvalues represent the mean-square fluctuations along the principal axes, the large-scale fluctuations or collective motions which dominate the biomolecular motions can be corresponded to the principal components with the large eigenvalues. The MD-generated trajectories $\mathbf{T}$ can be projected onto the mth principal component subspace by computing $\mathbf{T'v_m}$ where $\mathbf{v_m}$ is the mth eigenvector and the prime denotes a transposed matrix.

In general, 20 modes are usually more than enough to define an essential space that captures the motions governing biological functions.[92] However, the presence of large-scale motions makes it difficult to resolve small-scale motions because the former motion has much greater relative amplitude in atomic displacement. Cartesian coordinate-based PCA may reflect to some extent the dominant overall motion rather than smaller internal motion of the protein. PCA using Cartesian coordinates may not yield the correct rugged free energy landscape due to the artifact of the mixing of internal and overall motion, as separation of these two motions is essential to construct and interpret the energy landscape of a biomolecular system undergoing large structural rearrangement.[93] Internal coordinates such as backbone dihedral angles and distance-based measurements may be more advantageous as they provide natural separation of the external and internal motions.[94,95]

2.3.3 Clustering

Clustering is a data-mining technique which is widely used to group ensemble of conformers into similar structures based on a chosen distance measurement, such as distances between structures calculated via best-fit coordinate RMSD or some other Euclidean distances. A wide variety of algorithms have been applied in many studies to cluster molecular dynamics trajectories and search for similar structures which help gain an intuition of the biomolecular system. Clustering algorithms can be categorized into two major classes: hierarchical and non-hierarchical methods.

Hierarchical clustering seeks to build a hierarchy of clusters by merging pairs of clusters. For example, hierarchical clustering with the bottom-up approach (agglomerative hierarchical clustering) first starts with each data point as a single-point cluster and iteratively merges clusters into larger clusters until a desired number of clusters are reached. Commonly, distances between structures calculated via best-fit coordinate RMSD values or some other Euclidean distances are used to decide which clusters to merge. Hierarchical clustering can have different strategies to merge the data points such as single-linkage and average-linkage. Single-linkage uses the minimum distance, whereas average-linkage considers the average distance between members of two clusters. Average-linkage is expected to be one of the most useful approaches among various linkage algorithms.[96]

K-means clustering[97] is one of the most popular non-hierarchical clustering methods which aim to minimize the objective function given by:

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \left\| x_j - \mu_i \right\|^2 \tag{2.56}$$

where i is the cluster index, k is the defined number of clusters, x_j is the data point belonging to cluster C_i, and μ_i is the centroid of cluster C_i. $\left\| x_j - \mu_i \right\|^2$ is a chosen distance measure between point x_j and μ_i. K-means clustering first starts with assigning k random points as cluster centroids and subsequently all other points are assigned to the closest cluster centroid. The centroid of each cluster is recomputed and data points are reassigned to newly defined cluster centroids to minimize the objective function. This procedure is repeated until a stopping criterion is met, such as when centroids of newly formed clusters do not change or maximum number of iterations is reached. K-means algorithm does not necessarily find the most optimal solution, but instead approximates local minima of the objective function. The algorithm is also sensitive to the initial cluster centers, which are selected randomly, and is recommended to run multiple times with different random seeds

to reduce the initial dependency. K-means++[98] which is a modification of the standard K-means method can be applied to improve the initial guess of cluster center positions.

2.3.4 Markov state model analysis

The Markov state model (MSM)[99–101] is a discrete-state stochastic model which can provide insights into biomolecular mechanisms and predict stationary and kinetic quantities of long-timescale dynamics using a set of multiple simulations which are much shorter than the timescales of interest. This is achieved by coarse-graining the high-dimensional configuration space into n discrete microstates $S_{i=1,2\cdots n}$, and a conditional transition probability matrix, termed the transition matrix $\mathbf{T} \equiv \{T_{ij}\}$, is estimated from the simulation trajectories $\mathbf{x}$.

$$T_{ij}(\tau) = \text{Prob}\left(\mathbf{x}_{t+\tau} \in S_j \middle| \mathbf{x}_t \in S_i\right) \tag{2.57}$$

The transition matrix describes the chance of jumping from one state to another in some time interval τ which is referred to as the lag time. The eigen-decomposition of transition matrix $\mathbf{T}$ yields a set of eigenvectors and corresponding eigenvalues. The eigenvalues are related to the relaxation timescales of kinetic processes, and eigenvectors indicate the associated structural change occurring at these timescales. The relaxation timescale, also known as the implied timescale, is given by:

$$t_i = -\frac{\tau}{\ln\lambda_i} \tag{2.58}$$

where t_i is the relaxation timescale corresponding to the *i*th eigenvalue λ_i. The largest eigenvalue, λ_1, is always 1 for a model where networks are connected and satisfies the detailed balance condition. Thus, the corresponding (left) eigenvector represents the equilibrium distribution, i.e. stationary distribution. The equilibrium free energy of microstate i can be obtained using the stationary distribution $\boldsymbol{\pi}$ which is given by:

$$F_i = -\frac{1}{\beta}\ln(\pi_i) \tag{2.59}$$

The choice of a lag time is important to construct a valid Markov model. Plotting the relaxation timescales as a function of the lag time can give some indication of appropriate lag time.[102] According to the Chapman-Kolmogorov equation, $\mathbf{T}(n\tau) = \mathbf{T}(\tau)^n$, the relaxation times for a

Markov model with a lag time of $n\tau$ should be equal to those with a lag time of τ. Therefore, we can expect that the relaxation time level-off at a certain lag time if the model satisfies the Markov assumption. However, it should be noted that the convergence of the relaxation time does not guarantee Markovianity, as the eigenvectors would require being constant as well. One can also perform the Chapman-Kolmogorov test to further validate the model.

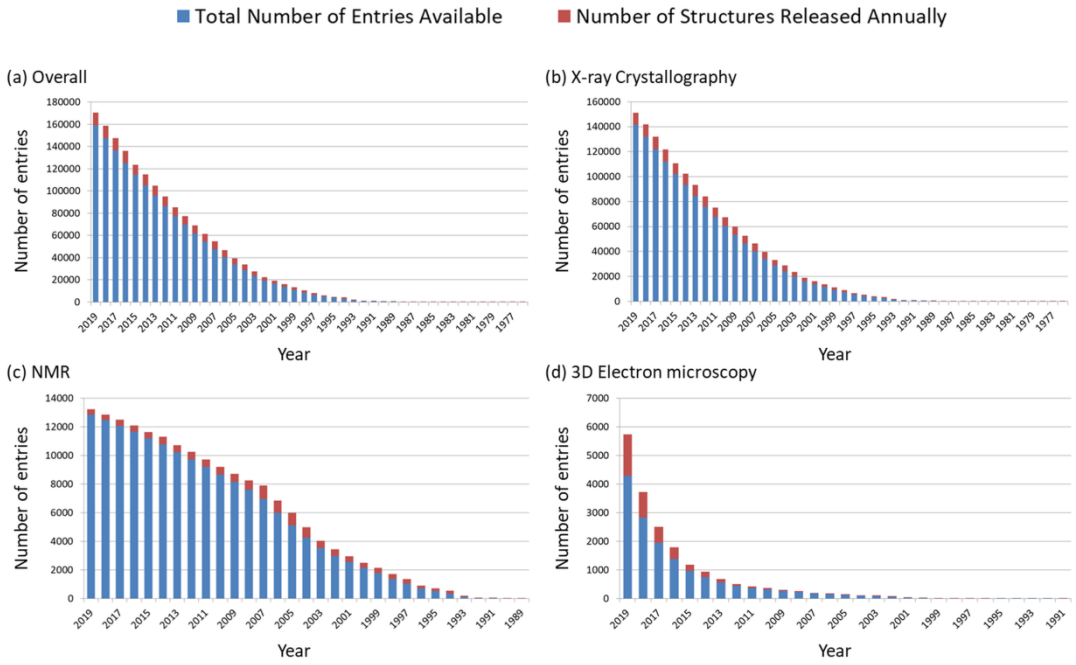

Figure 2.1: PDB statistical data of (a) overall, (b) X-ray crystallography, (c) NMR, and (d) 3D electron microscopy experiments. Data source taken from RCSB PDB (https://www.rcsb.org/).[103]

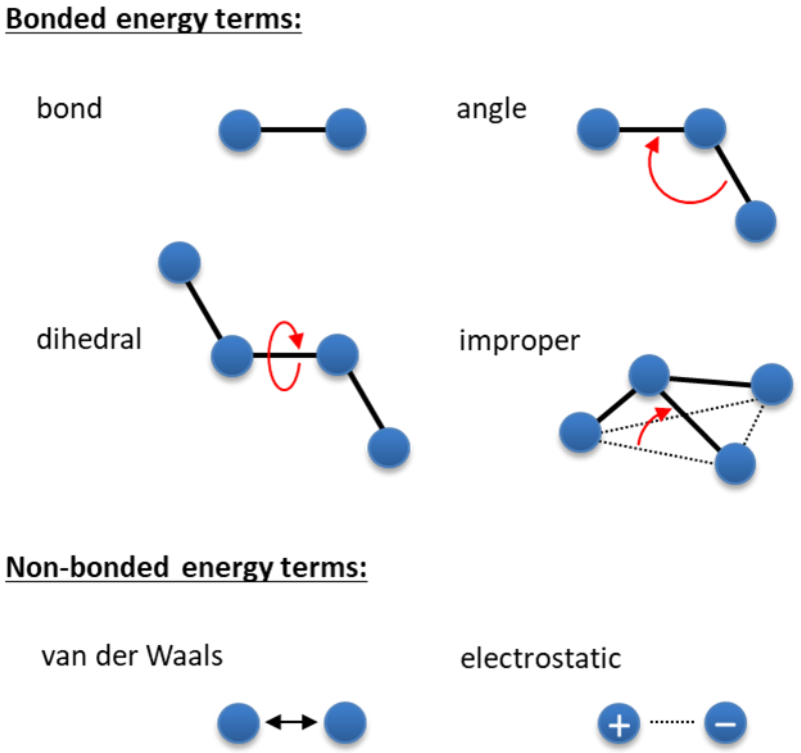

Figure 2.2: Schematic illustration of empirical force field with bonded and non-bonded energy terms.

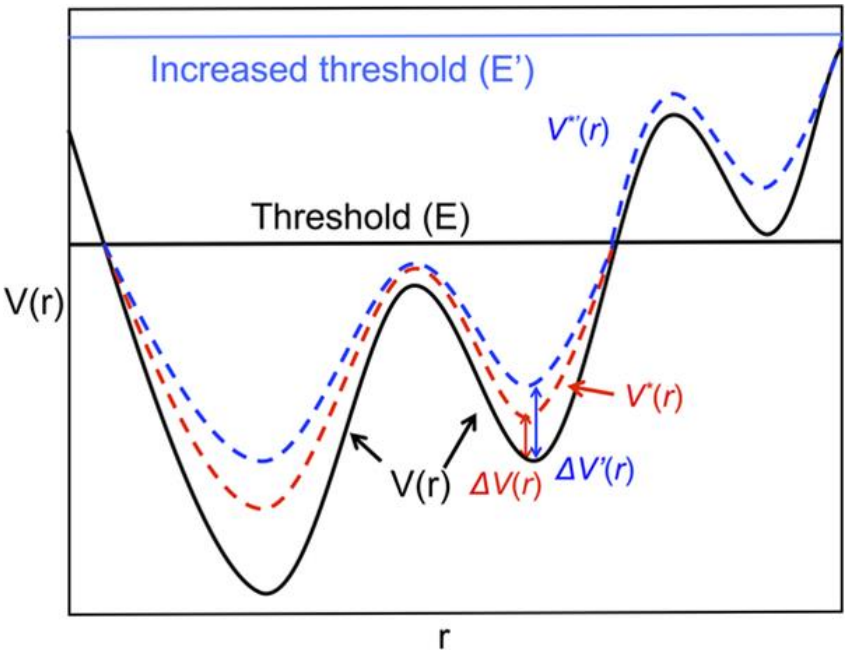

Figure 2.3: Schematic illustration of accelerated MD. $V(r)$, $\Delta V(r)$, and E represent the potential energy, boost potential energy and threshold energy, respectively. The red and blue dashed lines represent the modified potential energies using two different threshold energy, E and E', respectively. This figure is reprinted by Zhao B, Cohen Stuart MA, Hall CK (2017) Navigating in foldonia: Using accelerated molecular dynamics to explore stability, unfolding and self-healing of the β-solenoid structure formed by a silk-like polypeptide. *PLoS Comput Biol* **13**(3): e1005446. https://doi.org/10.1371/journal.pcbi.1005446.g011; licensed under Creative Commons Attribution (CC BY) License < https://creativecommons.org/licenses/by/4.0/>.

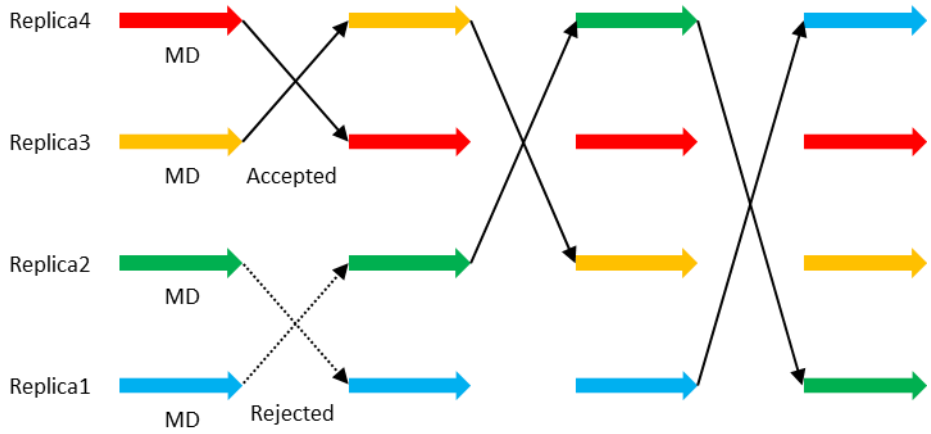

Figure 2.4: Schematic illustration of replica exchange molecular dynamics with temperature exchange. Each arrow represents a MD simulation with a different temperature.

Chapter 3
Edge Expansion Parallel Cascade Selection Molecular Dynamics Simulation

In this chapter, edge expansion parallel cascade molecular dynamics (eePaCS-MD) is introduced as a new enhanced sampling technique to investigate the large-amplitude collective motions of proteins with a focus on domain motions. Section 3.1 introduces the motivation of eePaCS-MD and addresses the current limitations of other related methods. Section 3.2 introduces the methodology and procedure of eePaCS-MD, as well as detailed description of the target systems used to assess the sampling efficiency of eePaCS-MD. The results and discussions are given in Section 3.3 and 3.4, respectively. Finally, the conclusion of this chapter is given in Section 3.5. This chapter is reproduced from Takaba, K.; Tran, D.P.; Kitao, A. Edge expansion parallel cascade selection molecular dynamics simulation for investigating large-amplitude collective motions of proteins. *J. Chem. Phys.* **2020**, 152 (22), 225101, with the permission of AIP Publishing.

3.1 Introduction

3.1.1 Parallel cascade selection molecular dynamics

Parallel cascade selection molecular dynamics (PaCS-MD)[104] is a class of adaptive sampling method which enhances the conformational transitions of two-end states. The PaCS-MD method dramatically enhances conformational transitions from one state to a target state without external perturbations using cycles of multiple independent MD simulations conducted in parallel. Each cycle consists of two major steps. In the first step, MD initial structures are selected from past simulation trajectories. The selected structures are the closest to the target structure defined by an appropriate quantity, e.g., root-mean-square deviation (RMSD). The second step involves conformational sampling by short parallel MD simulations. The selection of the initial structures considerably increases the probability of rare event occurrences to induce conformational change toward the target (Figure 3.1). The transition pathways obtained by PaCS-MD can be further analyzed by umbrella sampling along the pathways[104] or by performing Markov state model (MSM) analysis to obtain the free energy landscapes of protein conformational changes and protein-ligand bindings.[105,106]

One of the limitations of the original PaCS-MD method is the requirement of prior knowledge

of the product structure. Various extensions of PaCS-MD have been proposed over the past several years using distinct selection methods.[107–111] For example, nontargeted PaCS-MD (nt-PaCS-MD)[107] selects the structures that deviate most from the average structure based on Gram-Schmidt orthogonalization, whereas the outlier flooding method (OFLOOD)[108] selects outlier structures based on clustering techniques. The conformational sampling efficiency is enhanced by resampling from the edge or sparse distributions of the selected quantity, so that the probability along this quantity increases. Note that the conformational sampling is only enhanced against the selected quantity and not for the other coordinates (Figure 3.2). The utility of these methods has been demonstrated using several globular proteins, such as T4 lysozyme, glutamine binding protein, and maltose/maltodextrin binding protein. In all cases, the open-close transitions were observed within the simulation time scale of nanoseconds.

However, current methods still require improvement. For example, inefficient sampling can occur when the initial structures are not sufficiently distributed in a conformational space.[110] Furthermore, deterministic selection of initial structures may be problematic: we have observed the repeated selection of similar dead-end structures, resulting in inefficient sampling. In addition, some methods select the initial structures from all prior generated structures,[108–110] which requires keeping all the trajectories and makes the selection process time consuming. To address these issues, a new extension of eePaCS-MD was developed.

3.1.2 Edge expansion parallel cascade selection molecular dynamics

Edge expansion PaCS-MD (eePaCS-MD) is proposed as a new extension of PaCS-MD, which explores the large-amplitude collective motions of proteins with a focus on domain motions. eePaCS-MD brings together three important features: (i) conformational resampling from the structures rigorously located at the boundary of the sampled conformational space to improve sampling efficiency; (ii) reducing the number of candidates for the initial structures to quicken the selection process while retaining the information of the entire conformational space sampled by the simulation; and (iii) random selection of the initial structures to alleviate the risk of consecutively selecting dead-end structures. The first two features are realized by introducing the concept of "convex hull" defined as the smallest convex polygon that includes a given set of points in a multi-dimensional space (Figure 3.3(a) and (b)).[112] The convex hull is characterized by vertices and connecting edges. Computing the convex hull is a problem in computational geometry, which is widely applied to computer graphics, pattern recognitions,[113] image processing,[114,115] medical simulations,[116] home range estimations,[117] and animal epidemic forecasts.[118] In this study, eePaCS-MD-generated snapshots distributed in a several dimensional principal component (PC)

subspace determined by principal component analysis (PCA) are considered as the points constructing the convex hull. Identifying the snapshots as vertices reduces the number of candidate structures for selection. This procedure takes advantage of the fact that large-amplitude fluctuations of many proteins can be described in terms of only a few PCs.[11,20,21] Thus, eePaCS-MD randomly selects the initial structures from the vertices and enhances the concerted conformational transitions along a few PCs.

Here, we demonstrate the conformational sampling efficiency of eePaCS-MD for the open-close transitions of glutamine binding protein (QBP), maltose/maltodextrin binding protein (MBP), and adenylate kinase (ADK). Each was successfully simulated in ~10 ns of simulation time or less. eePaCS-MD is expected to offer 1-3 orders of magnitude shorter simulation time compared to conventional MD (cMD). Furthermore, we show that the combination of eePaCS-MD and accelerated MD (aMD)[77], which we call eePaCS-aMD, can further enhance conformational sampling efficiency, reducing the total computational cost of observing open-close transitions by at most 36%. We also compare eePaCS-MD with other related methods and conclude that the sampling efficiency of eePaCS-MD is slightly better or comparable.

3.2 Materials and methods

3.2.1 Procedure of eePaCS-MD

A flowchart for eePaCS-MD is shown in Figure 3.3(c). First, preliminary MD simulations (cycle 0) are performed, and the trajectories are subjected to PCA. The MD snapshots are projected onto a subspace spanned by a few PCs and a set of structures that constructs the convex hull is identified. The program package Qhull,[112] which implements the Quickhull algorithm, is applied to solve the convex hull problem (see Appendix for detail). A predefined number of initial structures are selected randomly from the vertices as the MD initial structures. Conformational sampling from the selected snapshots by parallel independent MD simulations (hereinafter called "*replicas*") is conducted with reinitialized velocities based on the Maxwell-Boltzmann distribution. After sampling, the MD trajectories from each replica and the vertices of the previous cycle are subjected to PCA, where the PCs are redefined to determine a new set of vertices and edges. Finally, the sampling process is repeated until it reaches a predefined number of cycles. Figure 3.3(b) shows the actual evolution of the vertices and edges for QBP.

3.2.2 Target systems

The conformational sampling efficiency of eePaCS-MD was assessed by applying it to three target proteins: QBP, MBP, and ADK (Figure 3.4). QBP and MBP are two-domain proteins widely studied by other methods related to eePaCS-MD.[107–111] We selected these proteins because they are suitable for examining the sampling efficiency of eePaCS-MD compared to the sampling efficiencies of previous studies. ADK was chosen as it has three domains and undergoes more complex movement[119,120] and thus is considered a more challenging target than QBP and MBP. Using ADK, we demonstrate the broader applicability of eePaCS-MD.

QBP and MBP are members of a large group of periplasmic binding proteins in gram-negative bacteria and are subunits of ATP binding cassette (ABC) transporters.[121–125] QBP and MBP move within the periplasm and serve as the initial receptor for the active transport of L-glutamine and maltose/maltodextrin, respectively. QBP and MBP consist of 226 and 370 residues, respectively, and comprise two globular domains linked by a hinge composed of two (QBP) or three (MBP) short linkers. The ligand binding site is located in the cleft between the two domains. X-ray crystallography and theoretical studies revealed that the two domains undergo a large amplitude domain motion upon ligand binding, from the open to closed structures.[121–128] QBP adopts an open form in its crystal apo state (Protein Data Bank (PDB) ID: 1GGG[123]) and takes a closed form in the holo state (PDB ID: 1WDN[125]). Similarly, MBP takes open and closed forms in the apo (PDB ID: 1OMP[122]) and holo (PDB ID: 1ANF[124]) states, respectively.

ADK is a nucleoside triphosphate kinase and is a monomeric enzyme that regulates the energy of a cell by balancing the relative abundance of AMP, ADP, and ATP via catalysis of the phosphoryl transfer reaction: $Mg^{2+} \cdot ATP + AMP \leftrightarrow Mg^{2+} + ADP + ADP$.[129,130] ADK comprises 214 residues and three domains: the LID domain that closes upon ATP binding, the NMP domain that closes upon AMP binding, and the core domain which shows no significant conformational change upon ligand binding. Conformational change between the open (PDB ID: 4AKE[130]) and closed (PDB ID: 1AKE[129]) states occurs as a multi-step clamshell-like movement.[120]

3.2.3 System preparation

The open and closed conformations of each protein were taken from the apo and holo crystal structures and employed as the starting structures for eePaCS-MD. In the closed state, the ligand was removed in the simulation system. The structures of the missing residues were modeled using PyMOL.[131] The protonation states of the amino acid residues were determined by H++.[132–134] For

QBP, Glu17 and Asp106 were protonated for both the open and closed states and all other residues were treated in their general tautomer states under physiological conditions. The AMBER ff14SB force field[51] was employed for all the proteins. Each system was solvated using TIP3P waters,[135] preserving crystal waters within 3.2 Å of each protein. Rectangular simulation boxes were constructed with a margin of at least 12 Å between the protein and the periodic box boundaries. The QBP system with a salt concentration of 100 mM KCl contained 11,702 and 13,159 water molecules for the open and closed states, respectively. The solvated MBP systems were neutralized by adding 8 Na^+ ions and each system contained 15,378 and 15,538 water molecules for the open and closed states, respectively. The systems for the open and closed states of ADK contained 12,534 and 10,312 water molecules, respectively, and 4 Na^+ ions were added to neutralize each system.

The GPU-accelerated version of AMBER14[136] was used for all simulations. Electrostatic interactions were treated with the particle mesh Ewald method[55] in which Lennard-Jones interactions and real-space electrostatic interactions were smoothly switched to zero at 10 Å. The systems of all three targets were first minimized with harmonic positional restraints imposed on the protein backbone atoms with a force constant of 10 kcal mol^{-1} $Å^{-2}$. The restraints were gradually reduced to zero during the minimization procedure. Subsequently, the system was subjected to 300 ps MD simulation with the NPT ensemble at 300 K and 1 bar, and an additional 200 ps was performed with the NVT ensemble at 300 K. For each case, a set of five distinct simulations was performed and the final structure was used as the initial structure for 10 preliminary MD simulations to generate the input for eePaCS-MDs. Isothermal and isobaric conditions were realized using the Langevin thermostat[137,138] with a friction constant of 2.0 ps^{-1} and the Berendsen barostat[63] with a pressure relaxation time of 1.0 ps. The MD time step was 2 fs, with hydrogen bonds constrained with the SHAKE[58] and SETTLE[59] algorithms.

3.2.4 Details of eePaCS-MD

The number of replicas (n_{rep}) and the total simulation cycles (n_{cyc}) were combinations of n_{rep} = 10 and 100 and n_{cyc} = 10, 15, 100, and 150, respectively, depending on the target system and the case study. The MD simulation time per cycle (t_{cyc}) was fixed to 100 ps and MD trajectories were saved every 1 ps. The eePaCS-MD simulation time ($t_{sim} = t_{cyc} \times n_{cyc}$) is proportional to the actual elapsed time or central processing unit (CPU) time, whereas the total accumulated MD time ($t_{tot} = n_{rep} \times t_{cyc} \times n_{cyc}$) indicates the total computational cost. For example, eePaCS-MD with n_{rep} = 10 and n_{cyc} = 100 requires t_{sim} = 10 ns and t_{tot} = 100 ns.

To investigate the effect of the number of PCs (n_{PC}) on sampling efficiency, eePaCS-MDs were

tested with n_{PC} = 2–5 using QBP. Based on the QBP results, we judged that n_{PC} = 4 is optimal and this value was used for MBP and ADK. Hereafter, for example, eePaCS-MD of QBP with n_{rep} = 10, n_{cyc} = 100 ps and n_{PC} = 2 starting from the open (OP) and closed (CL) states are referred to as $OPQ^{10,100,2}$ and $CLQ^{10,100,2}$, respectively, and Q indicates QBP. Similar indices to distinguish the target proteins and simulation conditions are used for MBP and ADK, as shown in the first column of Table 3.1. Five distinct trials of eePaCS-MDs from different initial structures were performed for both the open and closed states.

To further enhance the conformational sampling efficiency of eePaCS-MD, we also examined eePaCS-aMD, where aMD[77] was used instead of MD. In aMD, the true potential $V(\mathbf{r})$ is modified via a continuous non-negative boost potential $\Delta V(\mathbf{r})$ while maintaining the underlying shape of $V(\mathbf{r})$. The boost potential increases the escape rates from the potential wells and enhances the sampling efficiency. Here, dual boost aMD was applied where boost potentials were added to the total potential energy and the dihedral potential energy. The potential boost parameters (E_{pot}, α_{pot}) and dihedral boost parameters (E_{dih}, α_{dih}) were defined following the work of Pierce et al.[78] using a 150 ns total MD (30 ns MD × 5 trials per state) starting from the open and closed states.

3.2.5 Analysis

The goal of these applications is to generate natural conformational transition pathways from the open to closed states and vice versa without prior information on the other state. In this study, we judged that the conformational transitions to the other state were successfully simulated when the Cα RMSDs from the opposite state were within 1.5 Å. To define the reference structure for the RMSD calculation, five distinct 30 ns MDs were performed from the final structures of the aforementioned equilibration step for each case. The last 20 ns of five trajectories in the open or closed state were merged and subjected to agglomerative hierarchal clustering. The centroid structure of the most populated cluster was chosen as the representative structure of the open or closed state, and used as the reference for the RMSD calculation. Cα RMSD was measured from both the open and closed states to evaluate the conformational transitions with respect to n_{cyc}. $RMSD_{min}$ and $RMSD_{max}$ refer to the minimum and maximum Cα RMSDs measured from the opposite and initial structures, respectively. t_{1st} indicates the total computational cost required to reach the opposite state for the first time with different Cα RMSD criteria (3.5, 3.0, 2.5, 2.0, and 1.5 Å). In addition, we denote t_{min} as the eePaCS-MD time to reach $RMSD_{min}$. The eePaCS-MD trajectories were projected onto the first and second PC coordinates, where the subspaces were defined by the merged five distinct 30 ns cMD simulations of the open and closed states (total of 300 ns).

3.3 Results

3.3.1 Open-close transitions of QBP and the optimum number of PCs

We investigated the optimum value of n_{PC} by performing eePaCS-MDs of QBP with n_{rep} = 10, n_{cyc} = 100, and n_{PC} = 2–5, as summarized in Tables 3.1–3.3. For OPQ$^{10,100,2\text{-}5}$, $RMSD_{min}$ and t_{min} were in the range of 1.2–1.6 Å and 7.5–8.5 ns, respectively. OPQ^{10,100,3} achieved the lowest $RMSD_{min}$ (1.2 ± 0.3 Å) at t_{min} = 8.5 ± 0.8 ns, where 4/5 trials reached the closed state within n_{cyc} = 100. OPQ^{10,100,4} gave results comparable to OPQ^{10,100,3} but t_{min} was 9% shorter and the open-to-closed transitions were successful in all five trials. For CLQ$^{10,100,2\text{-}5}$, $RMSD_{min}$ and t_{min} were in the range of 1.3–2.5 Å and 5.8–9.5 ns, respectively. CLQ^{10,100,4} gave the lowest $RMSD_{min}$ (1.3 ± 0.3 Å) at t_{min} = 8.0 ± 2.2 ns, where 4/5 trials reached the open state. In the case of OPQ, t_{1st} tended to increase with n_{pc} against various RMSD criteria. Similar trend was observed with CLQ; however the trend was not as clear as OPQ since t_{1st} could not be averaged over all five trials even for rather high RMSD criteria (Table 3.2). Although the overall performance of eePaCS-MDs with n_{PC} = 3 and 4 were comparable, we chose n_{PC} = 4 for MBP and ADK, as the overall motions could be captured more efficiently by considering more PCs.

In most cases, the individual eePaCS-MD trials observed the open-close transition within 10 ns of simulation time, regardless of n_{PC} (Table 3.3). In contrast, with cMD the open-to-closed transition was not observed during 500 ns with the AMBER ff03 force field.[108] Figure 3.5 shows the average Cα RMSD ($RMSD_{min}$ and $RMSD_{max}$) and standard deviation with respect to n_{cyc}. Although $RMSD_{min}$ almost converged within n_{cyc}, $RMSD_{max}$ continued increasing. This is reasonable because higher energy states are reached as eePaCS-MD continues.

3.3.2 Open-close transitions of MBP and ADK

The MBP simulation results for MBP (OPM and CLM) are summarized in Tables 3.1 and 3.4. With OPM^{10,100,4}, $RMSD_{min}$ reached 1.5 ± 0.3 Å at t_{min} = 8.1 ± 1.8 ns; the corresponding value for CLM^{10,100,4} was 1.4 ± 0.9 Å at t_{min} = 6.9 ± 1.9 ns. Among the five trials with OPM^{10,100,4} and CLM^{10,100,4}, three and four trials were successful in reaching the opposite state within 100 cycles (t_{sim} ≤ 10 ns and t_{tot} ≤ 100 ns), respectively. The three unsuccessful trials, two and one trials from OPM and CLM, respectively, were able to reach the opposite state when n_{cyc} was extended to 150 cycles (Figure 3.6). A previous study reported that a 1 μs cMD simulation with the AMBER ff03 force field starting from the open state stayed in the initial state,[109] whereas the closed-to-open transition was observed within 500 ns of cMD simulation starting with the closed state.[110] Similarly, one of the 30

ns cMD simulations performed in the current study observed a closed-to-open transition and reflects the fact that the closed structure is stabilized by interactions with maltose/maltodextrin.

Table 3.1 and 3.5 shows the results for ADK with n_{cyc} = 150 and n_{PC} = 4. For OPA^{10,150,4}, $RMSD_{min}$ and t_{min} were 2.3 ± 0.6 Å and 10.3 ± 4.6 ns; the corresponding values for CLA^{10,150,4} were 2.4 ± 0.8 Å and 10.7 ± 3.4 ns, respectively. One trial of OPA^{10,150,4} observed the open-to-closed transition ($RMSD_{min} \leq$ 1.5 Å), and two and three trials of OPA^{10,150,4} and CLA^{10,150,4}, respectively, reached structures near the opposite states ($RMSD_{min} \leq$ 2.0 Å. See Table 3.5). The lowest $RMSD_{min}$ obtained by OPA^{10,150,4} and CLA^{10,150,4} were 1.5 Å and 1.7 Å, respectively, which showed nice overlap with the opposite structures as shown in Figure 3.7. These results also indicate that conformational sampling by eePaCS-MD can almost reach the opposite state. However, compared to QBP and MBP, simulating the open-close motion of ADK was more difficult, as expected from the aforementioned complex nature of ADK domain motion. For example, cMD simulations from the open state with AMBER ff03 did not approach to the closed crystal structure sufficiently within 1 μs.[139] Similarly, a 10 μs cMD simulation from the open state with AMBER ff12SB remained stable.[140] From the closed state, structures near the open crystal structure (~2 Å RMSD) were sometimes reached by 300–1,000 ns of cMD simulation with the AMBER ff03 force field.[139] We cannot rigorously compare these results with those of eePaCS-MD because of differences between the reference structures for the RMSD calculation and the different force fields; however, eePaCS-MD is expected to shorten the simulation time by 1–3 orders of magnitude compared with cMD simulation in the case of ADK.

For comparison, we also conducted five trials of 150 ns aMD for ADK. In the case of aMD from the open state, $RMSD_{min}$ from the closed state was 3.8 ± 0.5 Å, which is significantly greater than those with eePaCS-MDs, indicating that this direction is considered to be a difficult case for aMD. $RMSD_{min}$ from the closed state was 2.7 ± 0.6 Å, which is slightly worse than those with eePaCS-MDs.

3.3.3 Combination of eePaCS with accelerated MD (eePaCS-aMD)

Five distinct trials of eePaCS-aMDs were performed starting from the open and closed states (Table 3.1). The overall $RMSD_{min}$ were comparable to or slightly better than those obtained using eePaCS-MDs, with n_{PC} = 4. In addition, t_{min} decreased by 13–36% compared to the results of eePaCS-MD, except for CLM (4% increase). Next, the conformational spaces sampled by eePaCS-MDs and eePaCS-aMDs were compared, as shown in Figure 3.8. The merged cMD trajectories of the open and closed states (30 ns × 5 trials per state) are also shown for comparison. Broader conformational space was sampled by eePaCS-aMDs, indicating that the combination of

eePaCS with aMD can further enhance sampling efficiency. Similarly, the conformational space sampled by eePaCS-MD/aMDs are compared with the five trials of 150 ns aMD for ADK, as shown in Figure 3.9, indicating the high sampling efficiency of eePaCS-MD method.

In Figure 3.8, the PC coordinates used for the projection were determined using the cMD trajectories from both open and closed states. Practically, however, the structure of the opposite state may be unknown. As blind prediction, we examined whether the open and closed states are both distinguishable if we only employ the PC coordinates determined by the eePaCS-MD trajectories starting from one of the states (Figure 3.10). Similar to the results in Figure 3.8, the PC projections with the PC modes defined only by eePaCS-MD show that the initial and opposite states are clearly distinguished.

3.4 Discussion

3.4.1 Optimal n_{PC} and n_{rep}

Choosing the optimal value of n_{PC} is a nontrivial problem. As shown in Tables 3.1 and 3.2, t_{min} and t_{1st} of QBP tended to be shorter with fewer n_{PC}. This suggests that fewer n_{PC} results in fewer vertices, increasing the probability of selecting collective motions related to open-close transitions as long as important motions occur within the space spanned by the selected PCs. However, if there are too few n_{PC}, important protein motions might not be induced within the selected subspace. Figure 3.11 shows that the number of vertices significantly increases as n_{PC} increases, indicating that the number of vertices increases by a factor of three per dimension. To establish robust sampling corresponding to a variety of motions, we judged that n_{PC} = 4 is a reasonable default setting but should be adjusted depending on the target. We monitored the CPU time for solving the convex hull problem as a function of PC dimension (n_{PC}) before conducting the production runs (Figure 3.12). From n_{PC} = 7, the CPU drastically increased up to the order of seconds. Compared to this, 0.1 ns MD simulation per cycle with GPU Tesla K40c took 130–210 s, and the PCA calculation took a few seconds to tens of seconds. Therefore, eePaCS-MD up to n_{PC} = 8 is feasible in computational efficiency but higher dimension may be less efficient in computational time. If eePaCS-MD trial with n_{PC} = 4 is not so good, we would suggest to conduct one or two trials of eePaCS-MD by increasing n_{PC} up to 7 or 8. However, it should be noted that the number of vertex structures gradually increases as n_{cyc} evolves (Figure 3.11), meaning that the CPU time for the convex hull will also increase as n_{cyc} evolves. Therefore n_{rep} should be carefully adjusted to achieve sufficient selection rate of the vertex structures, i.e., n_{rep}/n_{vertex}, as discussed below.

Increasing n_{rep} is likely an efficient way to reduce simulation time and accelerate conformational transitions because the probability of rare event occurrence increases, as stated in an earlier paper.[105] To examine n_{rep} dependence, eePaCS-MDs with (n_{rep}, n_{PC}) = (100, 4) were also performed with n_{cyc} = 10 for QBP and MBP and 15 for ADK, so that t_{tot} is equal to that obtained with (n_{rep}, n_{PC}) = (10, 4), performed earlier. Five distinct trials were conducted starting from the open (OPQ^{100,10,4}, OPM^{100,10,4}, OPA^{100,15,4}) and closed states (CLQ^{100,10,4}, CLM^{100,10,4}, CLA^{100,15,4}. See Table 3.1). For all three targets, $RMSD_{min}$ obtained with (n_{rep}, n_{PC}) = (100, 4) were higher than that with (10, 4). In these cases, 77% (23/30) of the individual simulations showed $RMSD_{min}$ > 2.0 Å, indicating that the opposite state was not visited within the cycle limit (Tables 3.3–3.5). Next, we compared the evolution of $RMSD_{min}$ and $RMSD_{max}$ as a function of t_{tot} (Figure 3.13). Although eePaCS-MD with (n_{rep}, n_{PC}) = (100, 4) did not necessarily reach the opposite state as efficiently as (10, 4), it should be noted that the simulation time is one order of magnitude less than the latter case. The number of vertices with (100, 4) increased more rapidly compared to that with (10, 4) (Figure 3.14), indicating that the initial structures are more densely situated in the conformational space. Since the selection rates of the vertex structures, i.e., n_{rep}/n_{vertex}, with (n_{rep}, n_{PC}) = (100, 4) were several times higher than with (n_{rep}, n_{PC}) = (10, 4), more intensive sampling of the vertex structures was conducted using the former conditions. Here we have extended n_{cyc} of CLA^{100,15,4} to n_{cyc} = 50, i.e. CLA$_{extend}^{100,50,4}$, and achieved $RMSD_{min}$ comparable to those obtained by CLA^{10,150,4} but was able to reduce t_{min} by 64% (Table 3.1). As stated earlier, increasing n_{rep} is an efficient way to improve sampling efficiency; however, ten replicas were still adequate to promote the collective conformational transitions observed in this study.

In addition to the optimal n_{PC} and n_{rep}, it might be nontrivial to judge the convergence of the conformational sampling, because the conformational space will keep expanding as sampling is repeated from structures that are situated at the edge of the sampled conformational space. From our experiences, we recommend to use $RMSD_{max}$ as a criterion to stop the simulation, avoiding too much distortion from the original conformation. In the case of proteins similar to QBP, MBP, and ADK in size, we suggest to stop the simulation when $RMSD_{max}$ exceeds 8 Å.

3.4.2 Time evolution of PC subspaces

To examine the time evolution of the subspace spanned by the first few PCs, we introduced a quantity $R_{\alpha\beta}(t) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\boldsymbol{v}_{\alpha i}(t)\cdot\boldsymbol{v}_{\beta j}(t)\right)^2$ where $\boldsymbol{v}_{\alpha i}(t)$ is the eigenvector of the i-th PC at cycle t in trial α. $R_{\alpha\beta}(t)$ is unity if the subspace of trial α spanned by the first four PCs perfectly matches with that of trial β while this value vanishes if there is no correlation. The result with n = 4

is shown in Figure 3.15. In the beginning of eePaCS-MD, $R_{\alpha\beta}(t)$ of cycle 0 that was determined from the PCs of independent preliminary MDs ranged around 0.3–0.6, which indicates that the first four PC coordinates had some amount of correlation with those of the other trials from the beginning. This also shows that intrinsic anharmonic nature of the proteins can be partly captured during the short preliminary MDs. $R_{\alpha\beta}(t)$ converged to 0.6–0.7 and did not completely approach to unity, showing that there is some dependence on the initial conditions; thus the sampled conformational space differed among eePaCS-MD trials (Figures 3.16–3.18). Therefore, multiple trials of eePaCS-MD are suggested as we performed in this work.

Next, the inner products between distinct pairs of PCs from individual trials of eePaCS-MD simulations were calculated to investigate how PCs behaved during the simulation. Here we introduce a quantity $S_{ij}(t) = \left(\boldsymbol{v}_i(t) \cdot \boldsymbol{v}_j(t-1)\right)^2$ where $\boldsymbol{v}_i(t)$ is the eigenvector of the *i*-th PC at cycle *t*. $S_{ij}(t)$ represents how much the subspace spanned by the *j*-th PC at time $t-1$ is included in the *i*-th PC at cycle *t*. For example, $S_{11}(t) = 1$ means that the first PC at cycle *t* and $t-1$ share the same PC subspace, whereas $S_{12}(t) = 1$ indicates that the second PC at cycle $t-1$ changed to the first PC at cycle *t*. As a representative result, analysis of PC pairs of $i, j \leq 4$ applied to CLQ^{10,100,4} (trial 1 and trial 3) are shown in Figure 3.19. $S_{ij}(t)$ with $i = j$ tended to fluctuate throughout the simulation where the *i*-th PC at cycle *t* was partially captured by its neighboring PCs at cycle $t-1$; suggesting that the PCs flexibly change during eePaCS-MD to capture the large-amplitude fluctuations of proteins. In the case of CLQ^{10,100,4} (trial 1), $S_{11}(t)$ ranged around 0.8–1.0, meaning that the first PC did not change much during the simulation, whereas CLQ^{10,100,4} (trial 3) showed more fluctuations during the simulation. Interestingly, the RMSD$_{min}$ of the two trials (trial 1: 1.3 Å, trial 3: 1.9 Å) seems to differ respect to the behavior of $S_{11}(t)$.

We further examined whether successful eePaCS-MD trials that achieved low RMSD$_{min}$ can be related with certain PC behaviors during the simulation. Here another quantity $T_{i\tau}(t+\tau) = (\boldsymbol{v}_i(\tau) \cdot \boldsymbol{v}_i(t+\tau))^2$ is introduced where $\boldsymbol{v}_{i\tau}(t+\tau)$ is the eigenvector of the *i*-th PC at cycle $t + \tau$ and τ is a given fixed cycle. $T_{i\tau}(t+\tau)$ measures how much the *i*-th PC at cycle $t + \tau$ have change compared to those observed at $t = \tau$. The results with τ = 0, 10, 20, 30, 40, and 50 for eePaCS-MDs (n_{PC} = 4) applied to OPQ, CLQ, OPM, CLM, OPA, and CLA are shown in Figures 3.20–3.25, respectively. The relation between eePaCS-MD trials that achieved low RMSD$_{min}$ and $T_{i\tau}$ were not obvious for PCs higher than or equal to two ($i \geq 2$); however a slight trend was observed for the first PC (i = 1). In the case of eePaCS-MDs that showed high RMSD$_{min}$, such as trial 3 of CLQ (Figure 3.21(c): 1.9 Å) and trial 2 of CLM (Figure 3.23(b): 3.1 Å), $T_{i\tau}$ with i = 1 showed large fluctuations for various τ, indicating that the first PC was not clearly determined during the first 50 cycles of eePaCS-MD. In contrast, for eePaCS-MDs that were successful to achieve low RMSD$_{min}$, $T_{i\tau}$ with *i*

= 1 tended to behave more continuously, meaning that the first PC is determined in a certain direction or changes gradually respect to eePaCS-MD cycles. In eePaCS-MD, the conformational sampling is enhanced toward the selected quantity by increasing the probability along that direction. Therefore, eePaCS-MDs that are successful to determine the first PC which associates the largest amplitude motion after few tens of cycles is important to achieve efficient sampling. However, the conformational sampling could be inefficient even if the first PC is firmly defined as the PC may be irrelevant to the biological conformational change as seen with the third trial of OPA (Figure 3.24(c)).

3.4.3 Comparison of random and deterministic selections of initial structures

As mentioned in Section 3.1, the random selection of initial structures is one of the important features of eePaCS-MD. Here we show the pretest results of eePaCS-MD where initial structures are selected deterministically during the simulation. The procedure is similar to those described in Figure 3.3(c) except for step (iii), where the vertex structures were clustered into M = 10 clusters using agglomerative hierarchical clustering, and the largest RMSD snapshot with respect to the average structure from each cluster was selected as initial structures. Hereafter, we refer to this approach as deterministic-based approach and the original method which selects initial structures randomly as random-based approach. Since the simulation presented in Section 3.4 was one of the pretests, the simulation conditions of eePaCS-MD was slightly different from those described in Section 3.2, where 200 ps NVT simulation during equilibration was considered as the preliminary eePaCS-MD cycle, i.e. cycle 0, and MD trajectories were saved every 2 ps.

The performance of eePaCS-MD using deterministic-based approach, was investigated for QBP with n_{rep} = 10, n_{cyc} = 100, and n_{PC} = 2–5, as summarized in Tables 3.6–3.8. For $OPQ^{10,100,2\text{-}5}$, $RMSD_{min}$ and t_{min} were in the range of 1.2–1.3 Å and 5.3–7.0 ns, respectively. The deterministic-based approach gave comparable or slightly lower $RMSD_{min}$ values than random-based approach, thus t_{min} was 16–38% shorter. For $CLQ^{10,100,2\text{-}5}$, the $RMSD_{min}$ and t_{min} were in the range of 1.5–2.7 Å and 6.6–8.8 ns, respectively. Furthermore, the deterministic-based approach showed lower t_{1st} than the random-based approach against various RMSD criteria for most cases (Table 3.7). However, the random-based approach tended to show better results than deterministic-based approach for eePaCS-MDs using $n_{PC} \geq 4$ regrading $RMSD_{min}$ (Table 3.6).

In Figure 3.26, the initial structures from $(i+1)^{th}$ cycles were compared with the previous cycles to see how many replicas were subjected to sampling with different initial structures. eePaCS-MD

with random-based approach tends to select unique initial structures during the entire simulation. In deterministic-based approach, the number of unique initial structures rapidly decayed as n_{cyc} evolved and converged to 3–5, suggesting that similar initial structures were repeatedly selected for resampling. The same structures could be repeatedly selected for more than 20 cycles, as this was the case for CLQ[10,100,5]. This could result in inefficient sampling and expose risk in sampling energetically unfavorable structures. The simplest way to alleviate such risk is to select initial structures randomly as demonstrated in this study. Alternatively, application of reinforcement learning algorithms, such as Monte Carlo tree search[141] and multi-armed bandits,[142] can be applied to avoid being trapped in local states. In fact, the random selection of initial structures presented in this study can be regarded as an application of ε-greedy with ε = 1 which is one of the most famous and simplest multi-armed bandit algorithms.

In Figure 3.27, the time evolution of the subspace spanned by the first four PCs, i.e. $R_{\alpha\beta}(t)$, using random- and deterministic-based approaches are compared. In the case of deterministic-based approach (Figure 3.27(b)), $R_{\alpha\beta}(t)$ converged to 0.65–0.80 (OPQ) and 0.25–0.65 (CLQ) depending on the individual simulation trials, whereas $R_{\alpha\beta}(t)$ converged to ~0.65 for both OPQ and CLQ using the random-based approach (Figure 3.27(a)). Convergence to different $R_{\alpha\beta}(t)$ indicates that each individual trial samples different subspace regions and exhibits strong initial condition dependency.

3.4.4 Comparison with other related methods

The methodological features of eePaCS-MD are compared with those of other related methods in Table 3.9. In eePaCS-MD, the samplings are repeated from widely distributed structures that are rigorously located at the boundary of a conformational space to improve sampling efficiency. In contrast, SACS (Self-Avoiding Conformational Sampling)[109] and OFLOOD (Outlier FLOODing)[108] do not necessarily select the next starting structures from the edges but rather from near neighbors. In SACS, the initial structures are selected from unvisited PC subspaces from past cycles. The probability of selecting important motions is proportional to the ratio of the number of structures selected as initial structures versus the structures from unvisited PC subspaces from prior cycles. Sampling can be inefficient when this ratio becomes too small. In OFLOOD, the initial structures are randomly selected from sparsely distributed conformational spaces determined by clustering techniques. The second important feature of eePaCS-MD is its randomness in selecting the initial structures. The random selection of initial structures from the vertex structures used for resampling alleviates the risk of consecutively selecting unfavorable structures. SDS (Structural Dissimilarity Sampling)[110] and extended SDS[111] select the initial structures systematically based on inner products

among PCs. Another important feature of eePaCS-MD is that the information on the entire sampled conformational space is preserved with the minimal number of structures to accelerate the selection process. The vertex structures are the only structures stored throughout the simulation. In contrast, several related methods select the initial structures out of all prior generated structures,[108–110] which might require more storage, memory, and time for selection.

Next, we compared the sampling efficiency of eePaCS-MD and eePaCS-aMD ($n_{PC} = 4$) to those of other related methods for QBP, MBP, and ADK (Table 3.10). The results for QBP and MBP obtained by the related methods with reference numbers were taken from the original papers where the results are considered to be the best after parameter tuning, but each simulation was performed only once. We judged that we cannot reproduce the other methods exactly except for nt-PaCS-MD because the software to conduct them is not available. nt-PaCS-MD does not require any parameter tuning except for n_{rep}. Therefore, we conducted five trials of nt-PaCS-MDs for ADK from the open and closed states with $n_{rep} = 10$, which is the best parameter for both original nt-PaCS-MD and eePaCS-MD. We selected ADK because this target is the most challenging among the three targets investigated in this study and we expected that difference between the methods is more evident.

More than 3/5 eePaCS-MD trials were successful in observing the open-close transitions ($RMSD_{min} \leq 1.5$ Å) within $t_{tot} < 100$ ns for QBP and MBP (Tables 3.3 and 3.4). The $RMSD_{min}$ averaged over five trials of eePaCS-MDs for OPQ, CLQ, OPM and CLM simulations were 1.3 ± 0.2, 1.3 ± 0.3, 1.5 ± 0.3, and 1.4 ± 0.9 Å, respectively; the corresponding results for eePaCS-aMDs were 1.3 ± 0.1, 1.3 ± 0.1, 1.2 ± 0.1, and 1.3 ± 0.1 Å, respectively (Table 3.1). In contrast, the lowest $RMSD_{min}$ of QBP observed using other related methods (nt-PaCS-MD and OFLOOD) was no better than 1.7 Å for both OPQ and CLQ, despite requiring over twice the total computational cost as eePaCS-MD, i.e., $t_{tot} = 200$ ns. For MBP, $RMSD_{min}$ of OPM and CLM simulated by SACS, SDS, and extended SDS were in the range of 0.8–2.0 Å and t_{tot} ranged from 50–1000 ns. Several results from SDS and SACS showed lower $RMSD_{min}$ than eePaCS-MDs, but in most cases the simulations required a longer computational time. Among five trials of eePaCS-MD for ADK, two and three trials of OPA and CLA were successful in reaching the opposite state ($RMSD_{min} \leq 2.0$ Å), respectively. In contrast, nt-PaCS-MD succeeded only once in both cases. In addition, average $RMSD_{min}$ of eePaCS-MDs for OPA and CLA were 2.3 ± 0.6 and 2.4 ± 0.8 Å, respectively, which were smaller than the corresponding results of nt-PaCS-MDs (2.7 ± 0.8 and 2.7 ± 0.4 Å). Therefore, we judged that eePaCS-MD is more efficient than nt-PaCS-MD at least for this particular target.

The open-close transition of MBP is a large amplitude motion of two domains and is likely governed by the first two PCs. As anticipated, SACS employing the first two PCs ($SACS^{PC1,2}$, $t_{tot} =$

100 ns) applied to OPM showed lower $RMSD_{min}$ than eePaCS-MD with $n_{PC} = 4$. When SACS was compared under the similar simulation condition as eePaCS-MD, i.e., $SACS^{PC1-4}$ (t_{tot} = 100 ns), eePaCS-MD showed lower $RMSD_{min}$ than SACS. Extended SDS showed lower $RMSD_{min}$ and required less computational time than eePaCS-MD. This is expected because the former method samples from structures that deviate far from the initial structure promoting the open-close conformational change more intensively than the latter method. However, we speculate that the conformational space sampled by eePaCS-MD is much broader than that of extended SDS, as the latter method is more focused on enhancing sampling toward a certain direction in conformational space.

Table 3.10 also shows the comparison of t_{1st}. To estimate t_{1st} of QBP and MBP, we used the same RMSD criteria as those in the literatures (2 and 1 Å for QBP and MBP, respectively), so as to make a better comparison with the results of the other methods. As far as in the successful cases, eePaCS-MD/aMD tended to show shorter t_{1st} compared to the other methods except for CLQ and OPA. In the case of CLQ, average t_{1st} for eePaCS-MD/aMD were longer than that of nt-PaCS-MD, but the latter was conducted only once. The shortest t_{1st} for eePaCS-MD and eePaCS-aMD were 19 and 24 ns, which are shorter than that of nt-PaCS-MD (27 ns). OPM can be considered as a relatively difficult target because many of the other methods, as well as eePaCS-MD, did not reach the opposite state. Even in the successful cases, it took 89–660 ns in t_{1st}.

Table 3.11 summarizes the t_{1st} results of eePaCS-MD/aMD and nt-PaCS-MD against various RMSD criteria. For OPA, eePaCS-MD/aMD and nt-PaCS-MD required similar t_{1st} (~20 ns) to reach RMSD = 3.5 Å, where all five trials of each method were successful to satisfy the RMSD criteria. However, nt-PaCS-MD took ~60 ns in t_{1st} to reach RMSD = 3.0 Å which is twice as much computational time compared to eePaCS-MD/aMD (~30 ns). For CLA, 4/5 trials of eePaCS-MD and nt-PaCS-MD were successful to reach RMSD = 3.5 Å but t_{1st} was ~8% shorter for the former case. In the case of eePaCS-aMD, only 2/5 trials reached RMSD = 3.5 Å. Since the true potential of the system is artificially modified with a boost potential, eePaCS-aMD may had difficulty in capturing the PC subspace relevant to the conformational transitions toward the opposite state. The required t_{1st} to reach other RMSD criteria are difficult to estimate as less successful trials of eePaCS-MD/aMD and nt-PaCS-MD are observed as the criteria becomes stricter.

Although we cannot rigorously compare the eePaCS-MD results with those obtained using the other related methods, the conformational sampling efficiency of eePaCS-MD is expected to be better or comparable to that of the other related methods. It should be also noted that, for comparison, multiple trials of simulations are necessary because the sampled conformational space can depend on

the trial. In practice, several trials of eePaCS-MD are recommended.

3.4.5 Analysis of free energy landscape in combination with the Markov state model

Analysis of the free energy landscape can be performed by using the Markov state model (MSM) constructed from the trajectories generated by PaCS-MD.[105,106,143,144] The first approach only employed the PaCS-MD-generated trajectories[105,106] while the second approach also used additional MD trajectories together with the PaCS-MD trajectories so as to increase the statistics.[143,144] In this work, we used the second approach because eePaCS-MD tends to sample higher energy conformations so that statistics of low energy conformations was not sufficient without additional MDs.

We first performed the clustering of merged eePaCS-MD trajectories of OPQ[10,100,4] in the space spanned by the first 10 PCs and identified 496 highly-connected microstates from 500 clusters. Then, 496 cMD simulations started from the conformations of the cluster centers were independently conducted for 1 ns. After merging the trajectories obtained by eePaCS-MD and additional MDs, re-clustering into 496 microstates was conducted. Free energy landscape was calculated from the probabilities of the stationary distribution of the microstates obtained by MSM. The clusters were discretized by 100 trials of K-Means clustering[97] into clusters with K-Means++[98] for an initial guess of cluster center positions, ensuring the convergence of the clustering. We used the maximum likelihood estimator to build the Markov State Model with detailed balance condition by utilizing the PyEMMA package.[145]

The obtained free energy landscape of QBP is shown in Figure 3.28. We found that the global free energy minimum was situated in the open state as expected. Another free energy minimum was found in the closed state, and the transition state was located between the open and closed states. The positions of these energy minima are very close to those of the representative cMD structures of the open and closed states marked by the cross and square in Figure 3.8. Free energy of the closed state is 2.6 kcal/mol higher than that of the open state (global free energy minimum). Compared to the global minimum, the energy height of the transition state is 4.1 kcal/mol. The free energy difference between the two states and the barrier height determined with AMBER ff03 force field were reported to be ~2.6 and 5 k_BT (1.6 and 3.0 kcal/mol), respectively,[108] both of which are slightly lower by ~1.0 kcal/mol. Overall features of the 2D free energy landscape shown in the literature[146] are similar to that of Figure 3.28 but fee energy differences were not explicitly described in the paper. The MSM results indicate that eePaCS-MD can be combined with MSM to analyze the free energy landscape.

3.5. Conclusion

In this chapter, we proposed eePaCS-MD as an efficient adaptive conformational sampling method to investigate the large-amplitude motions of proteins. eePaCS-MD accelerates conformational sampling along a few large-amplitude PC modes by limiting the conformational space to be sampled at low dimensions. This treatment is expected to be efficient in exploring large protein motions such as domain motions but presumably not suitable for sampling more localized motions like loop flapping motion. We have demonstrated that eePaCS-MD can simulate open-close transitions along several collective degrees of freedom without prior knowledge of the opposite structure. In eePaCS-MD, resampling is repeated from the randomly selected initial structures that are rigorously located at the boundary of the conformational spaces identified as vertices of a convex hull spanned by several PCs. This resampling increases the probability of rare event occurrences, inducing conformational transitions to new conformational states, thus enhancing sampling efficiency. The information of the entire conformational space sampled by the simulation is stored as a set of vertex structures and is updated every cycle, which speeds up the selection process and improves the robustness of the method. We showed that eePaCS-MD successfully observed the open-close transitions of QBP, MBP, and ADK, requiring ~10 ns of simulation time on average, which is 1–3 orders of magnitude faster than conventional MD. Furthermore, we showed that the combination of eePaCS-MD and aMD (eePaCS-aMD) further enhances conformational sampling efficiency, with the total computational cost of observing the open-close transitions being reduced by 13–36%, except for one case where the cost increased by 4%. We also compared eePaCS-MD with other related methods and concluded that the sampling efficiency of eePaCS-MD is slightly better or comparable.

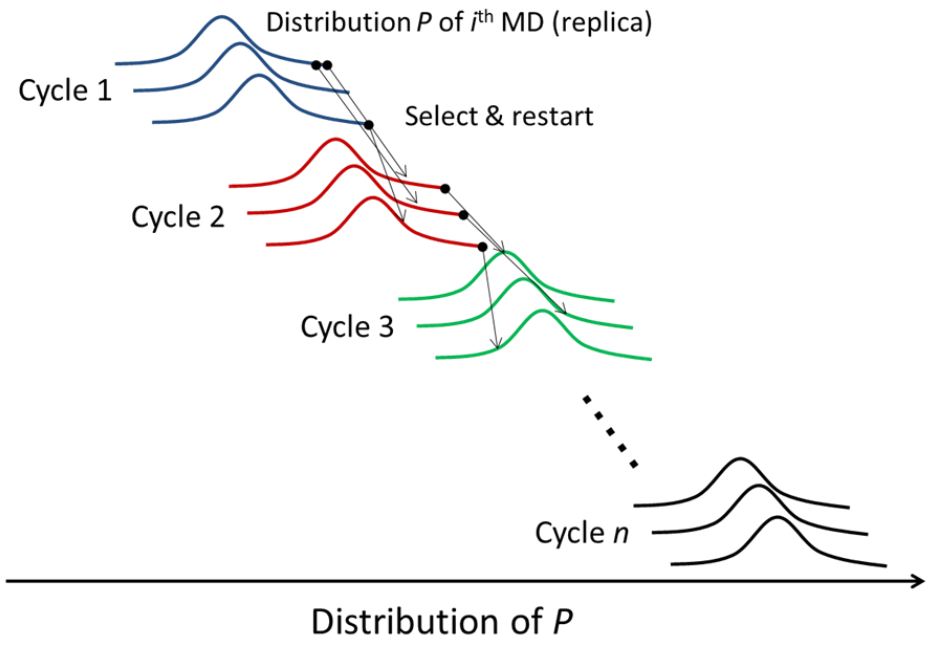

Figure 3.1: General concept of PaCS-MD. P is a selection quantity defined by the user.

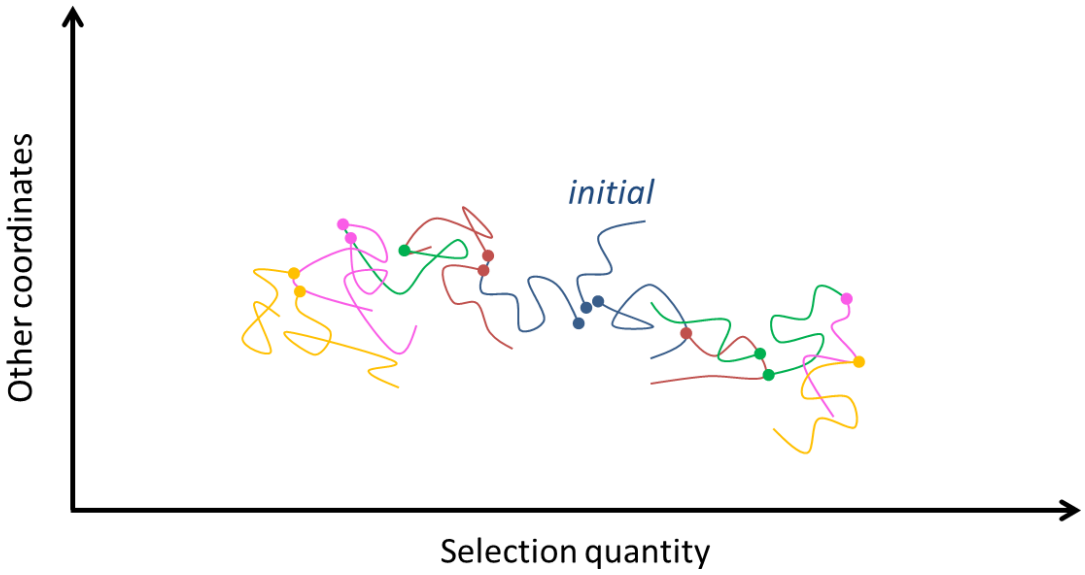

Figure 3.2: Conceptual trajectories generated by extensions of PaCS-MD that does not require the prior knowledge of the product structure. The filled circles indicate the starting point of each trajectory. The trajectories are sequentially generated in the order of blue, brown, green, magenta, and orange.

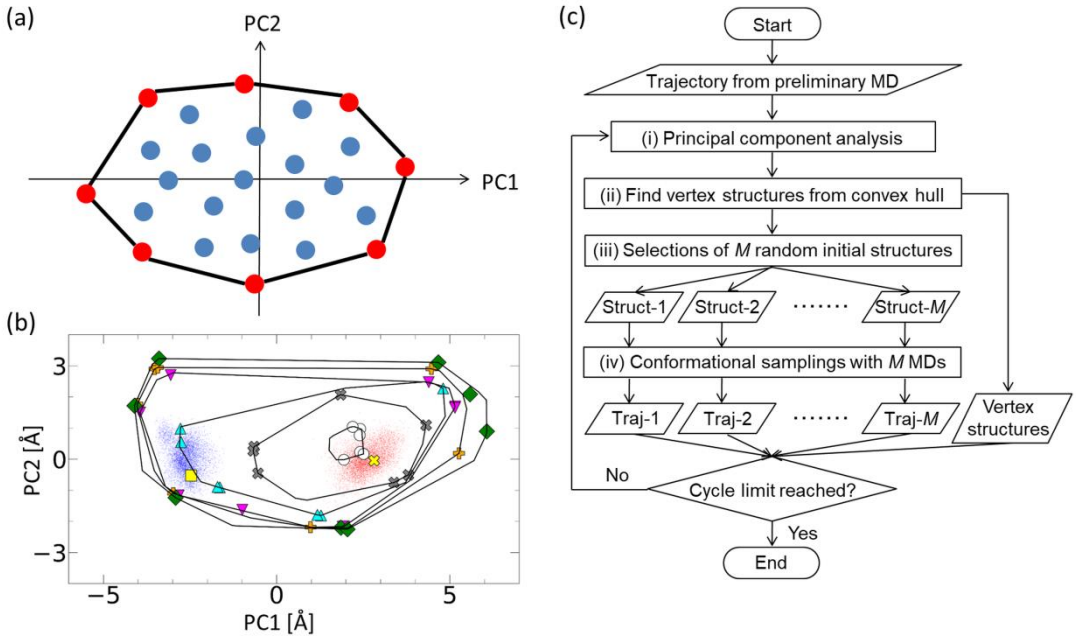

Figure 3.3: Procedure for conducting eePaCS-MD. (a) Schematic illustration of the convex hull. Each circle indicates a protein structure projected onto a subspace spanned by a set of PCs (PC1 and PC2). (b) The evolution of the vertices and edges, represented as colored markers and connecting black lines, respectively, for QBP. The vertices and edges are shown every 20 cycles (white: cycle 0, gray: 20, cyan: 40, magenta: 60, orange: 80, and green: 100). Cycle 0 refers to the preliminary MD which was initiated after the equilibration step (Section 3.2.3.). The red (open state) and blue (closed state) points represent five distinct 30 ns cMD simulations which were extended from the equilibration step. The yellow cross and square markers denote representative structures of the open and closed states which were defined by the extended 30 ns cMD simulations (Section 3.2.5), respectively. Note that the PCs used to obtain the vertices and edges are different from those used for projection. The vertices and edges were obtained from PCs that were redefined every eePaCS-MD cycle, and subsequently projected onto the first two PCs defined by the five distinct 30 ns cMD simulations starting from the open and closed states. (c) Flowchart for eePaCS-MD.

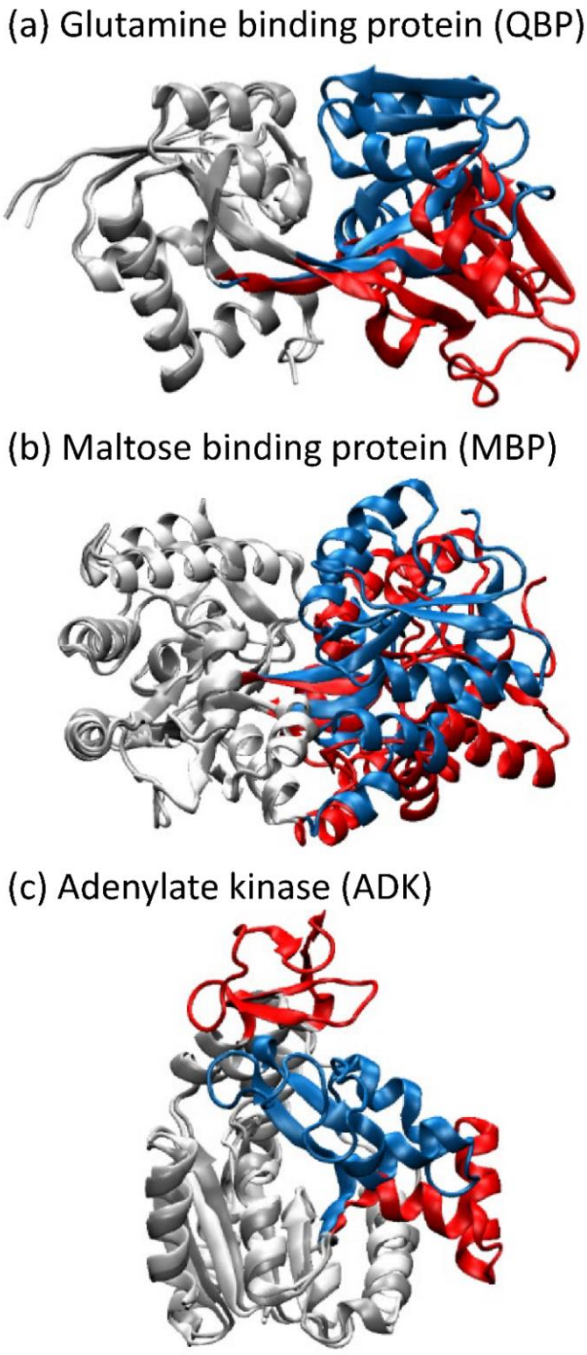

Figure 3.4: Superposition of the open and closed X-ray structures of (a) QBP: glutamine binding protein (open: PDB ID 1GGG, closed: 1WDN), (b) MBP: maltose/maltodextrin binding protein (open: 1OMP, closed: 1ANF), and (c) ADK: adenylate kinase (open: 4AKE, closed: 1AKE). The open and closed structures are depicted in red and blue, respectively, and the domains used for superposing the two states are shown in gray.

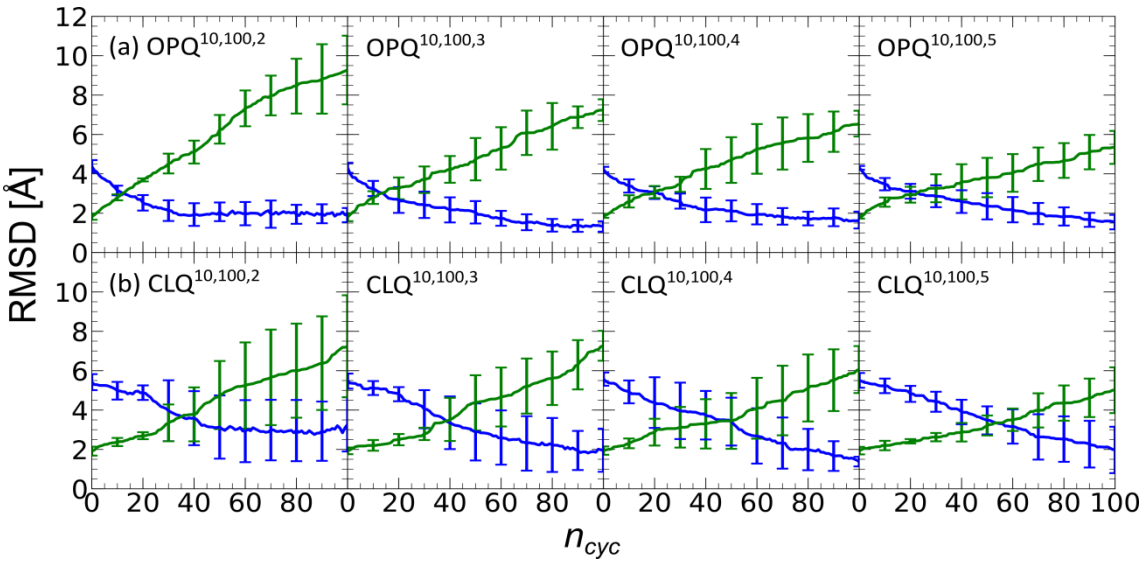

Figure 3.5: Evolution of Cα RMSD of QBP ($RMSD_{min}$ and $RMSD_{max}$) as a function of n_{cyc}. eePaCS-MD starting from (a) the open (OPQ) and (b) closed states (CLQ) are shown. Blue and green lines represent $RMSD_{min}$ and $RMSD_{max}$, respectively. The error bars indicate the standard deviations.

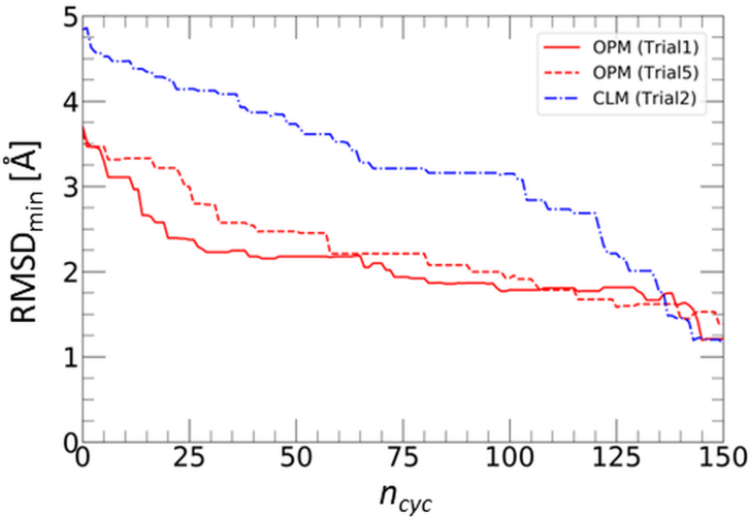

Figure 3.6: $RMSD_{min}$ profile of MBP obtained by eePaCS-MD with n_{cyc} extension. The three trials which were unsuccessful to reach the opposite state within 100 cycles were extended to 150 cycles. The trial number corresponds to individual trials described in Table 3.3.

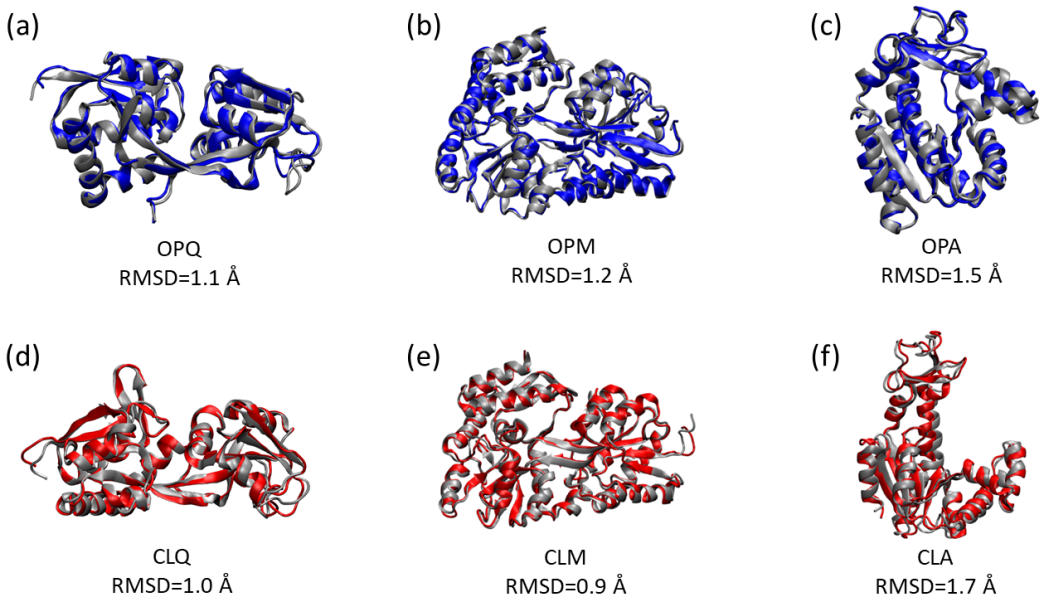

Figure 3.7: Superposition of the snapshots sampled by eePaCS-MD (gray) with the representative open and closed structures; and the corresponding Cα RMSD values. Representative closed structure (blue) and the closest snapshot to the representative closed structure sampled by eePaCS-MD starting from the open state of (a) QBP, (b) MBP, and (c) ADK are shown. Similarly, representative open structure (red) and the closest snapshot to the representative open structure sampled by eePaCS-MD starting from the closed state of (d) QBP, (e) MBP, and (f) ADK are shown.

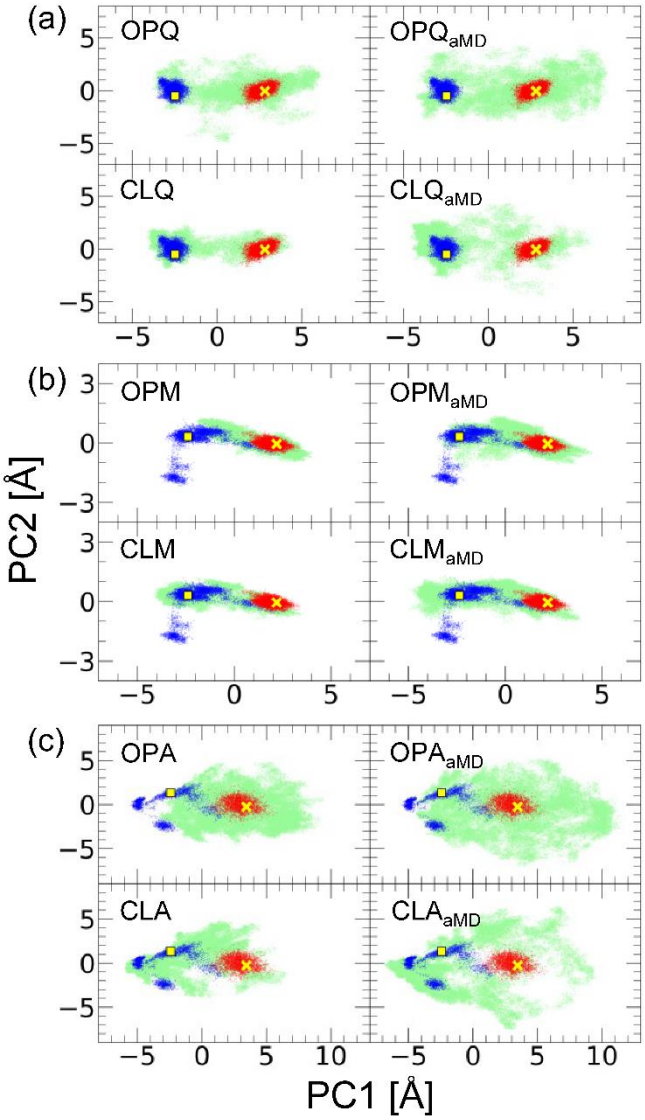

Figure 3.8: The eePaCS-MD/aMD trajectories (pale green) of (a) QBP, (b) MBP, and (c) ADK projected onto a subspace spanned by the first and second principal components (PC1 and PC2). The five distinct 30 ns cMD simulations starting from the open (red) and closed state (blue) are shown. The yellow cross and square markers denote representative structures of the open and closed states, respectively.

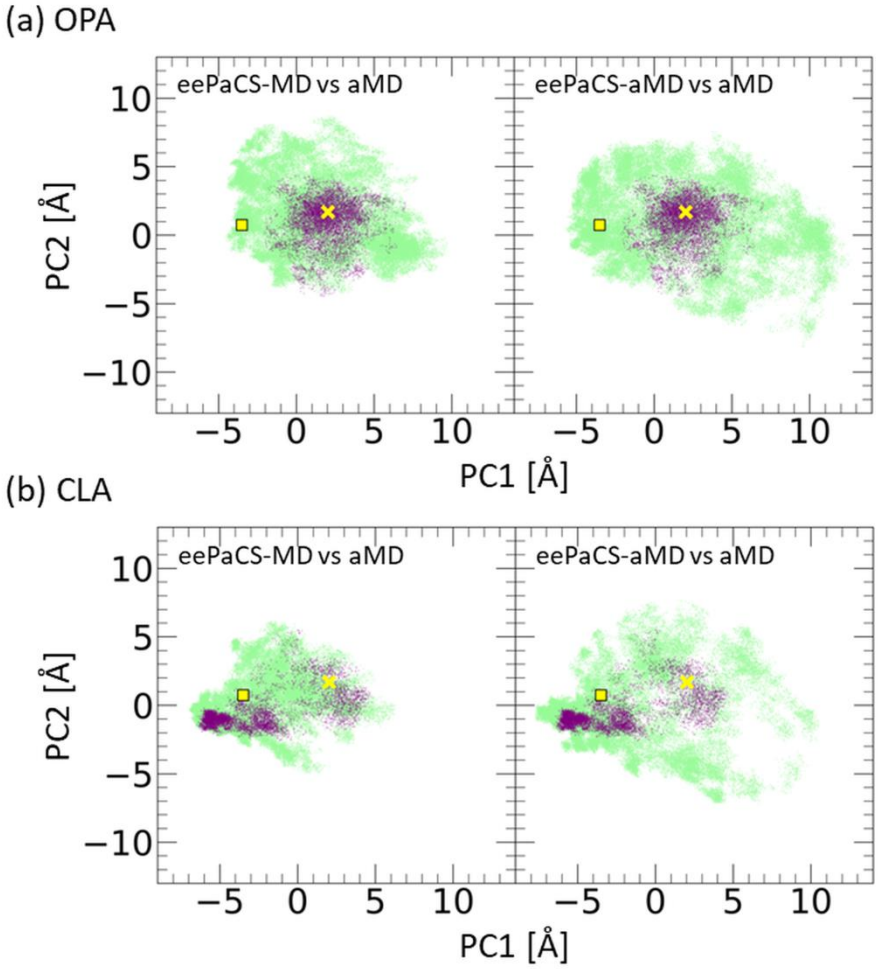

Figure 3.9: The eePaCS-MD/aMD trajectories (pale green) of ADK compared with five distinct 150 ns aMD (purple) simulations starting from the (a) open and (b) closed state. Trajectories were projected onto a subspace spanned by the first and second principal components (PC1 and PC2), similarly to those described in Figure 3.8.

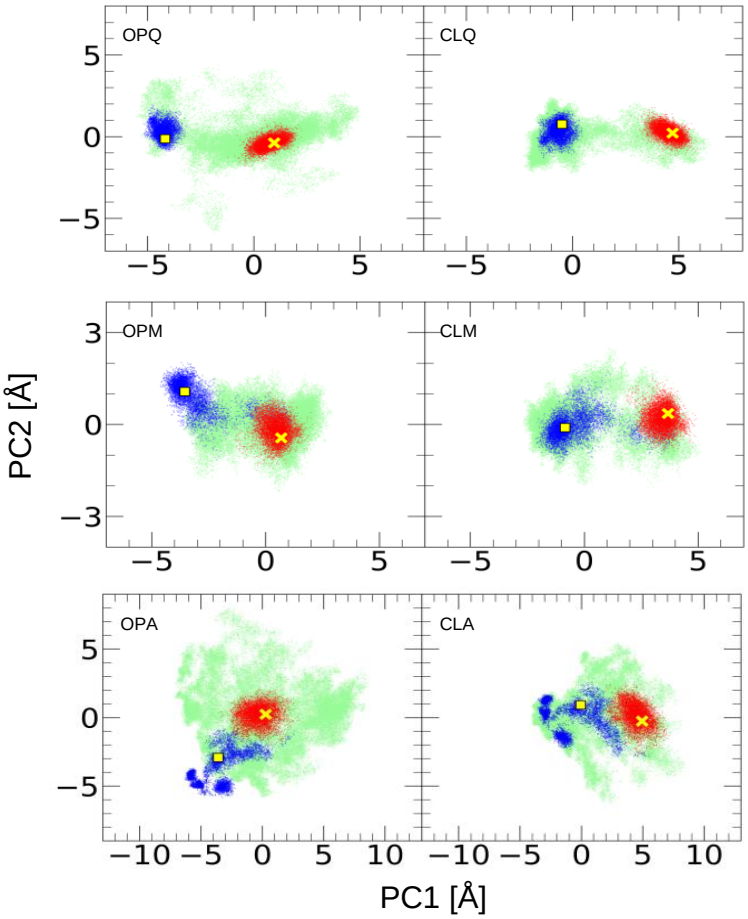

Figure 3.10: This figure is equivalent to Figure 3.8 but shows the trajectories projected onto a subspace spanned by the first two PC axes determined only from the eePaCS-MD trajectories. The five distinct 30 ns cMD simulations starting from the open (red) and closed state (blue) are shown. The yellow cross and square markers denote representative structures of the open and closed states, respectively.

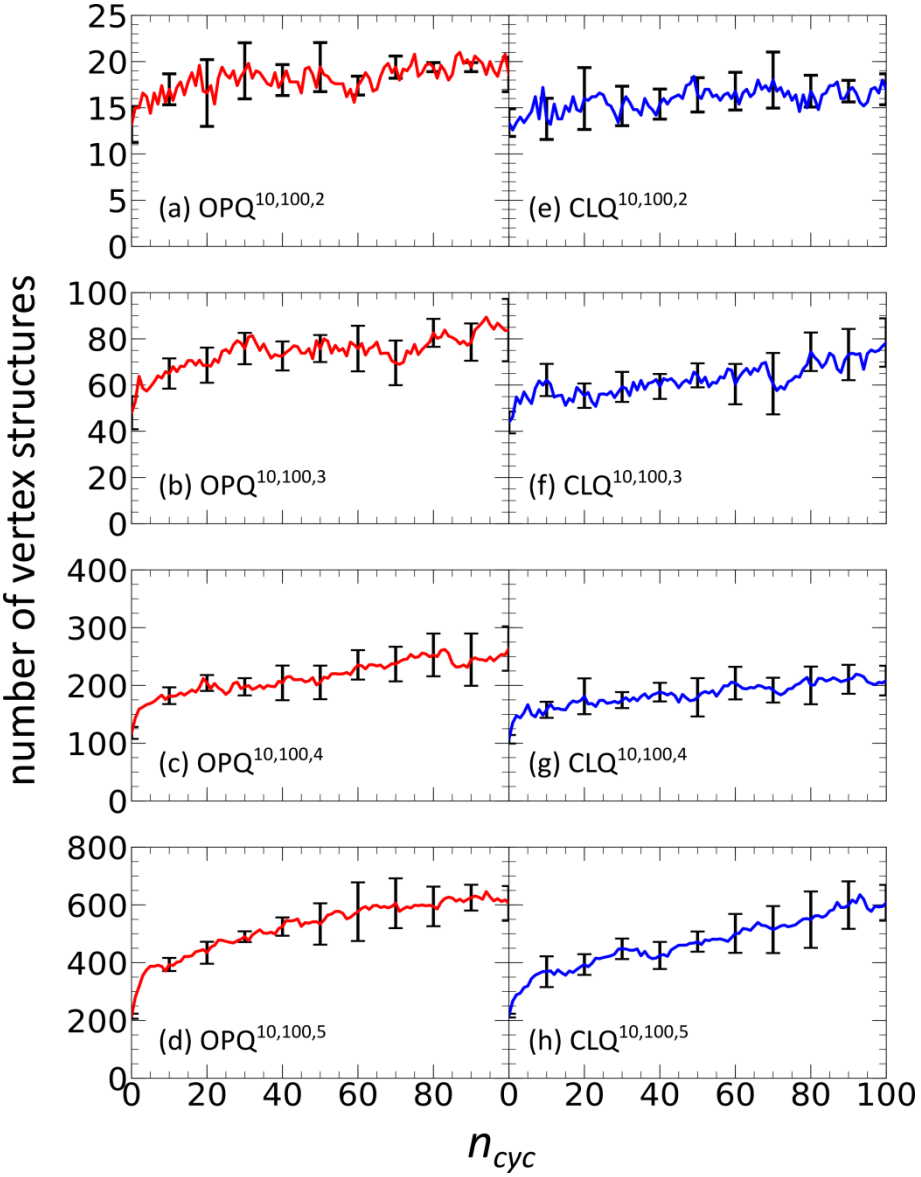

Figure 3.11: The number of vertex structures during eePaCS-MD of QBP starting from the open (red) and closed (blue) states. (a-d) and (e-h) represent the results for OPQ$^{10,100,2\text{-}5}$ and CLQ$^{10,100,2\text{-}5}$, respectively. The black error bars represent the standard deviations.

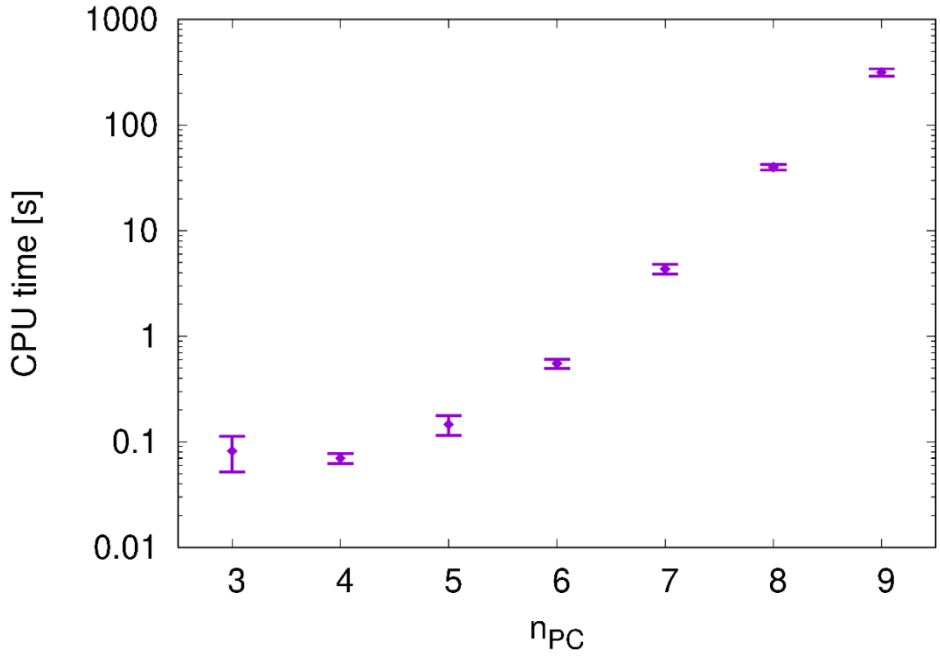

Figure 3.12: CPU time for solving the convex hull problem with different n_{PC}. One core of AMD Opteron 4122 (2.2 GHz) was used with 1,000 data points. The error bars indicate the standard deviations over 5 different datasets.

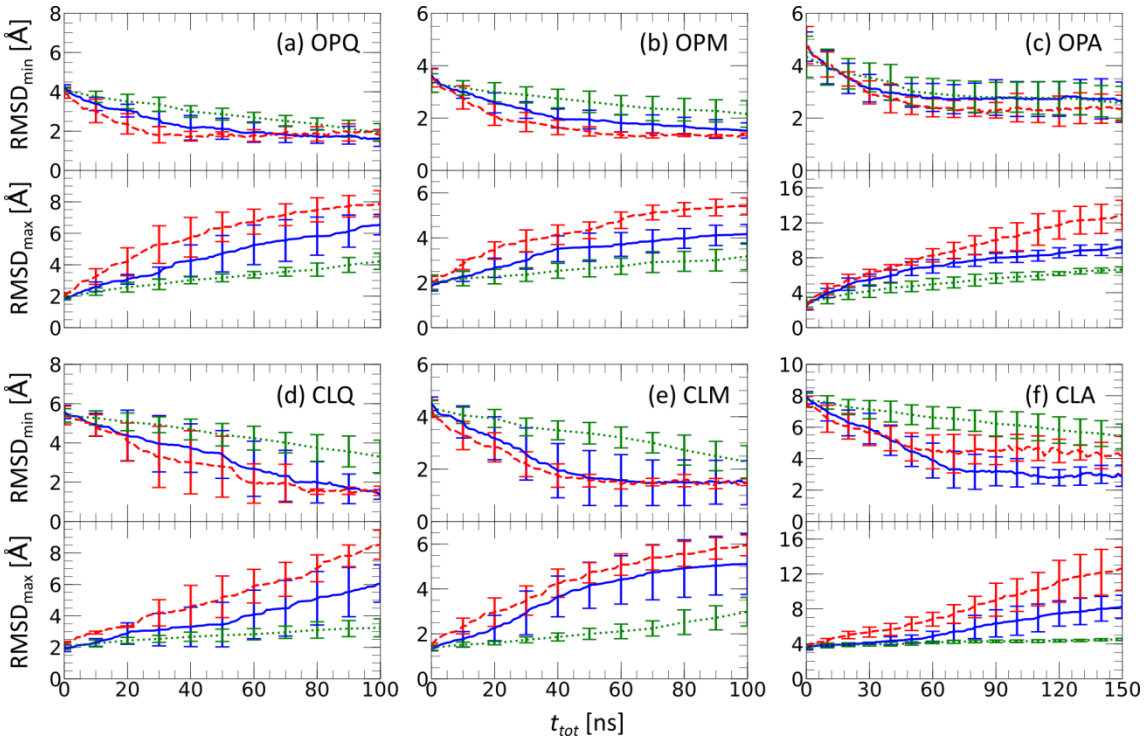

Figure 3.13: Comparison of evolution of Cα RMSD ($RMSD_{min}$ and $RMSD_{max}$) as a function of t_{tot} by eePaCS-MD with n_{rep} = 10 (blue) and 100 replicas (green) and eePaCS-aMD with n_{rep} = 10 (red). Simulations starting from the open state (a-c) and the closed state (d-f) of QBP, MBP, and ADK are shown. The error bars represent the standard deviations.

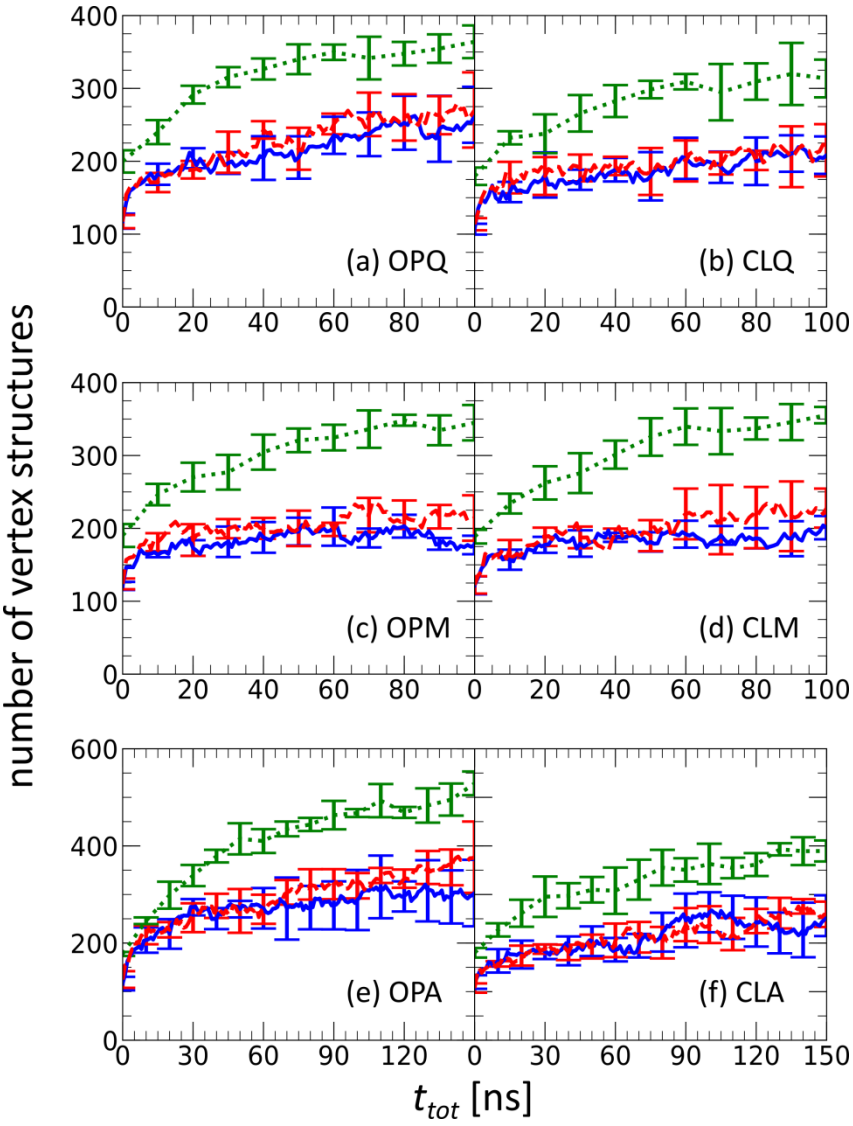

Figure 3.14: The number of vertex structures as a function of t_{tot}. The results of eePaCS-MD with n_{rep} = 10 (blue) and 100 (green), and eePaCS-aMD with n_{rep} = 10 (red) are shown. n_{PC} is fixed to 4. (a,b) QBP, (c,d) MBP, and (e,f) ADK. The error bars represent the standard deviations.

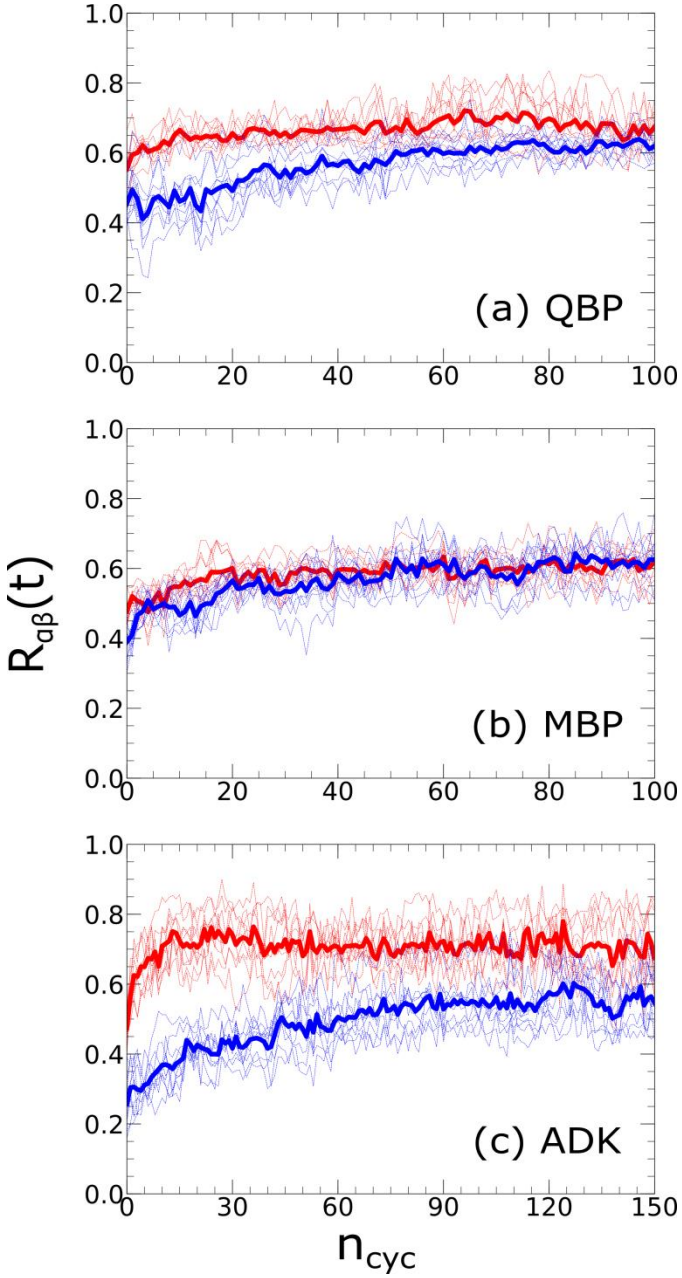

Figure 3.15: Similarity of the PC subspace spanned by the first four PCs between a pair of distinct eePaCS-MD trials at cycle *t*. The results for (a) QBP, (b) MBP, and (c) ADK from the open (red) and closed (blue) states are shown. The thin lines show the individual results and the thick lines indicate the average.

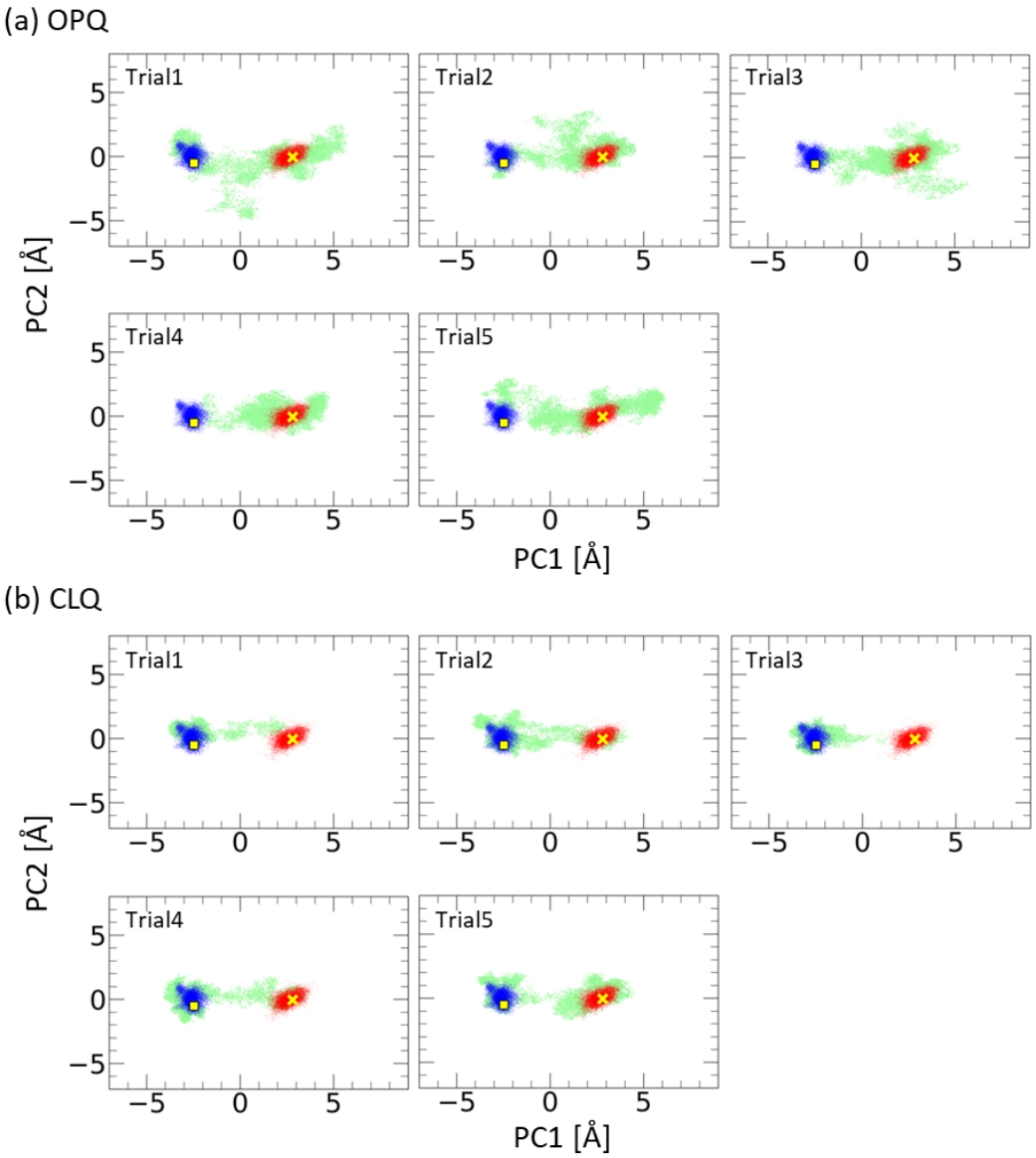

Figure 3.16: The five individual eePaCS-MD trajectories (pale green) of (a) OPQ and (b) CLQ projected onto a subspace spanned by the first and second principal components (PC1 and PC2). The five distinct 30 ns cMD simulations starting from the open (red) and closed state (blue) are shown. The yellow cross and square markers denote representative structures of the open and closed states, respectively.

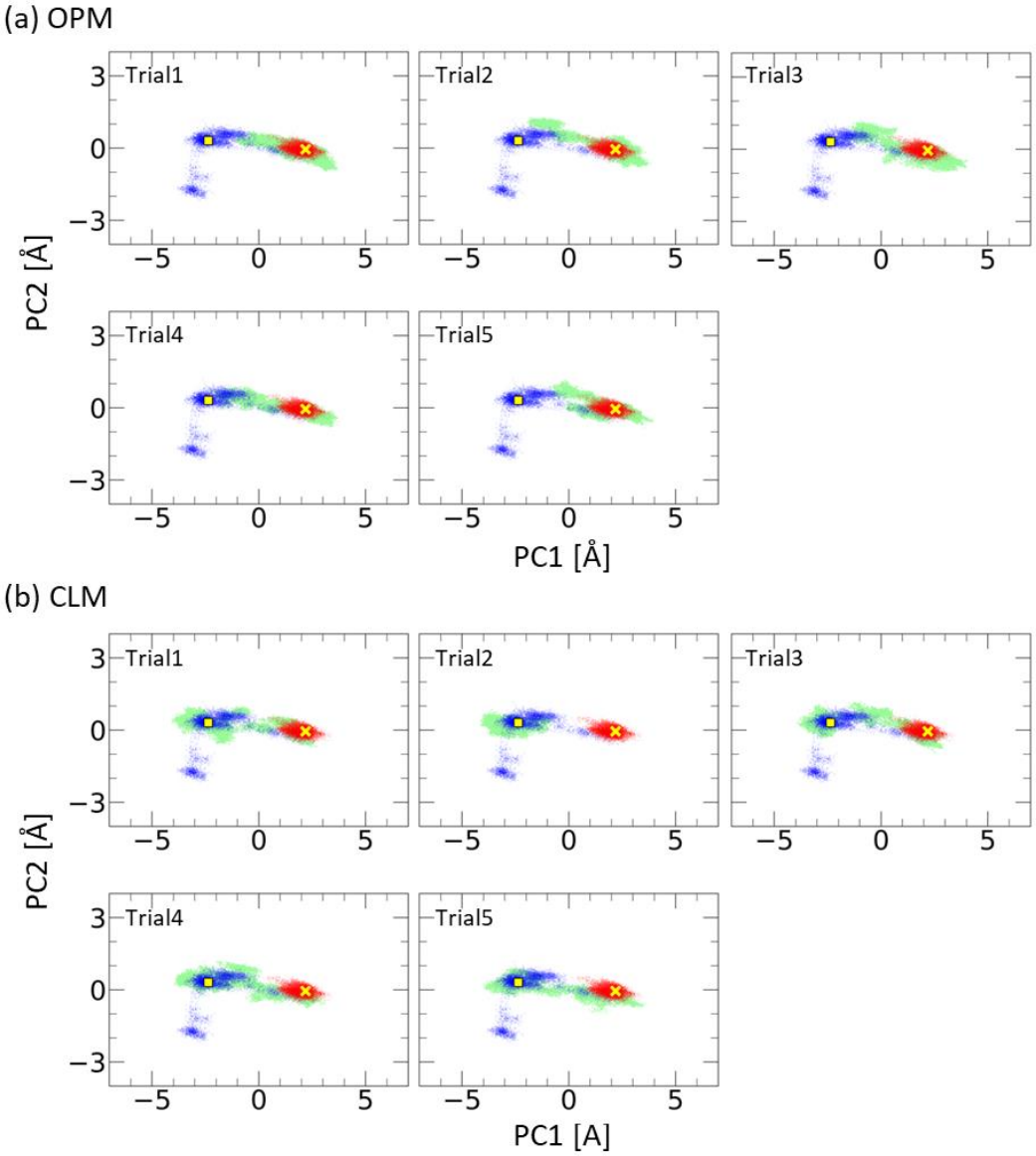

Figure 3.17: The five individual eePaCS-MD trajectories (pale green) of (a) OPM and (b) CLM projected onto a subspace spanned by the first and second principal components (PC1 and PC2). The meaning of the figure is same as those described in Figure 3.16.

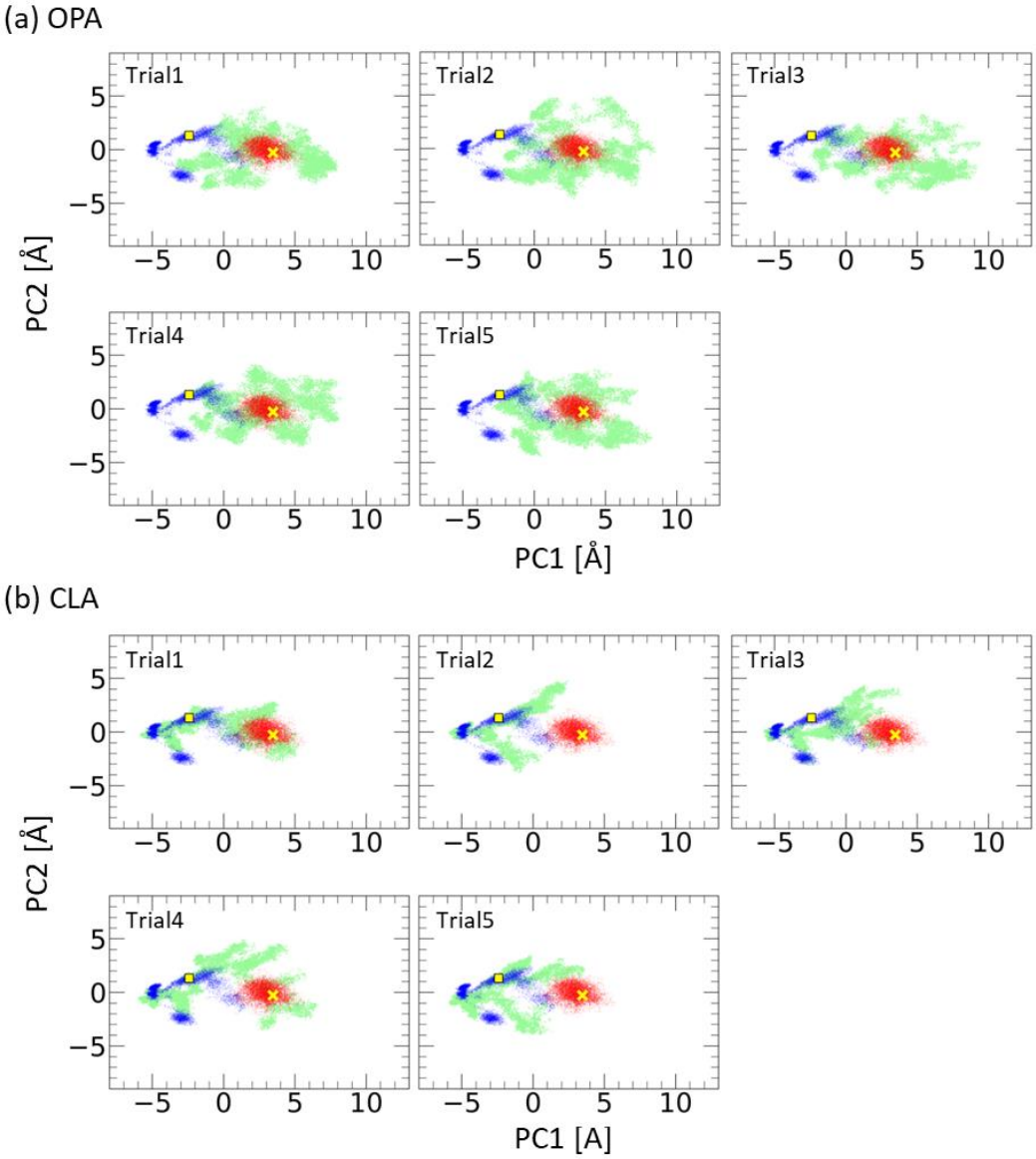

Figure 3.18: The five individual eePaCS-MD trajectories (pale green) of (a) OPA and (b) CLA projected onto a subspace spanned by the first and second principal components (PC1 and PC2). The meaning of the figure is same as those described in Figure 3.16.

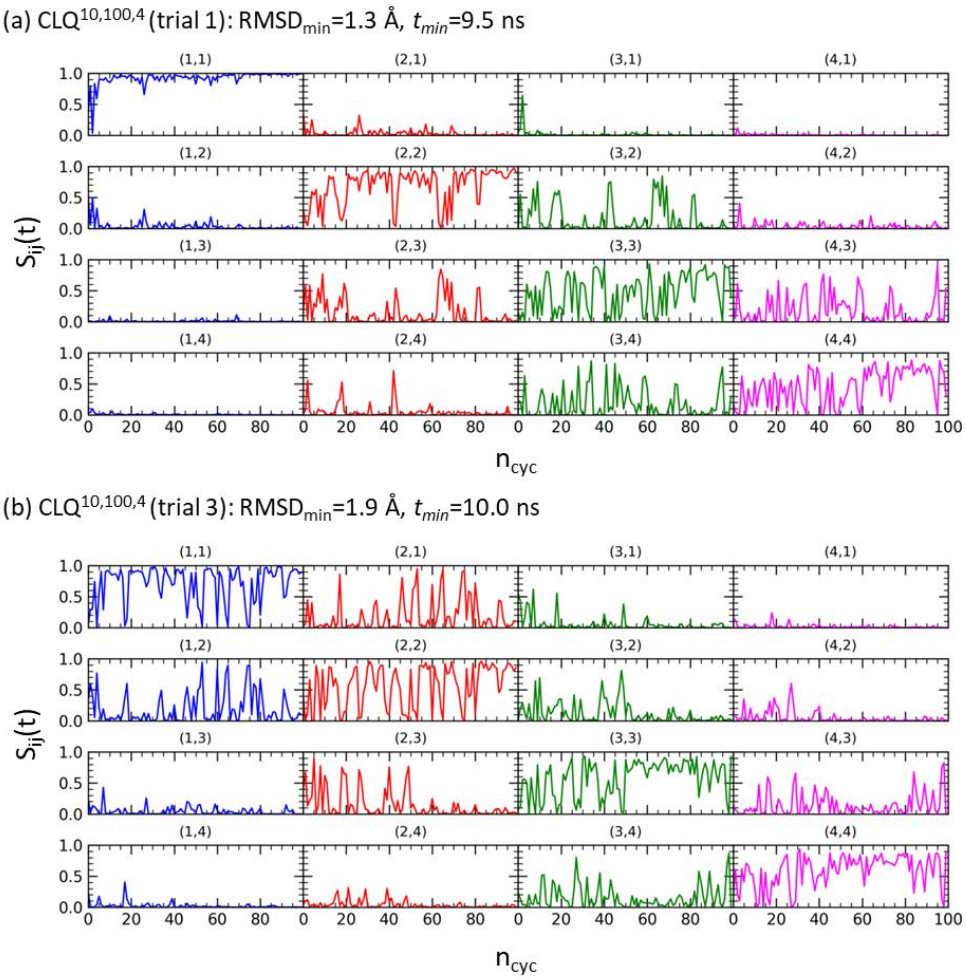

Figure 3.19: Evolution of inner products between distinct pairs of PCs. The results of $CLQ^{10,100,4}$ for (a) trial 1 and (b) trial 3 are shown. The numbers in brackets, (i, j), represents the inner products between the i-th PC at cycle t (blue: 1st PC, red: 2nd PC, green: 3rd PC, and magenta: 4th PC) and j-th PC at cycle $t - 1$. $RMSD_{min}$ and t_{min} are shown for references which were taken from Table 3.1.

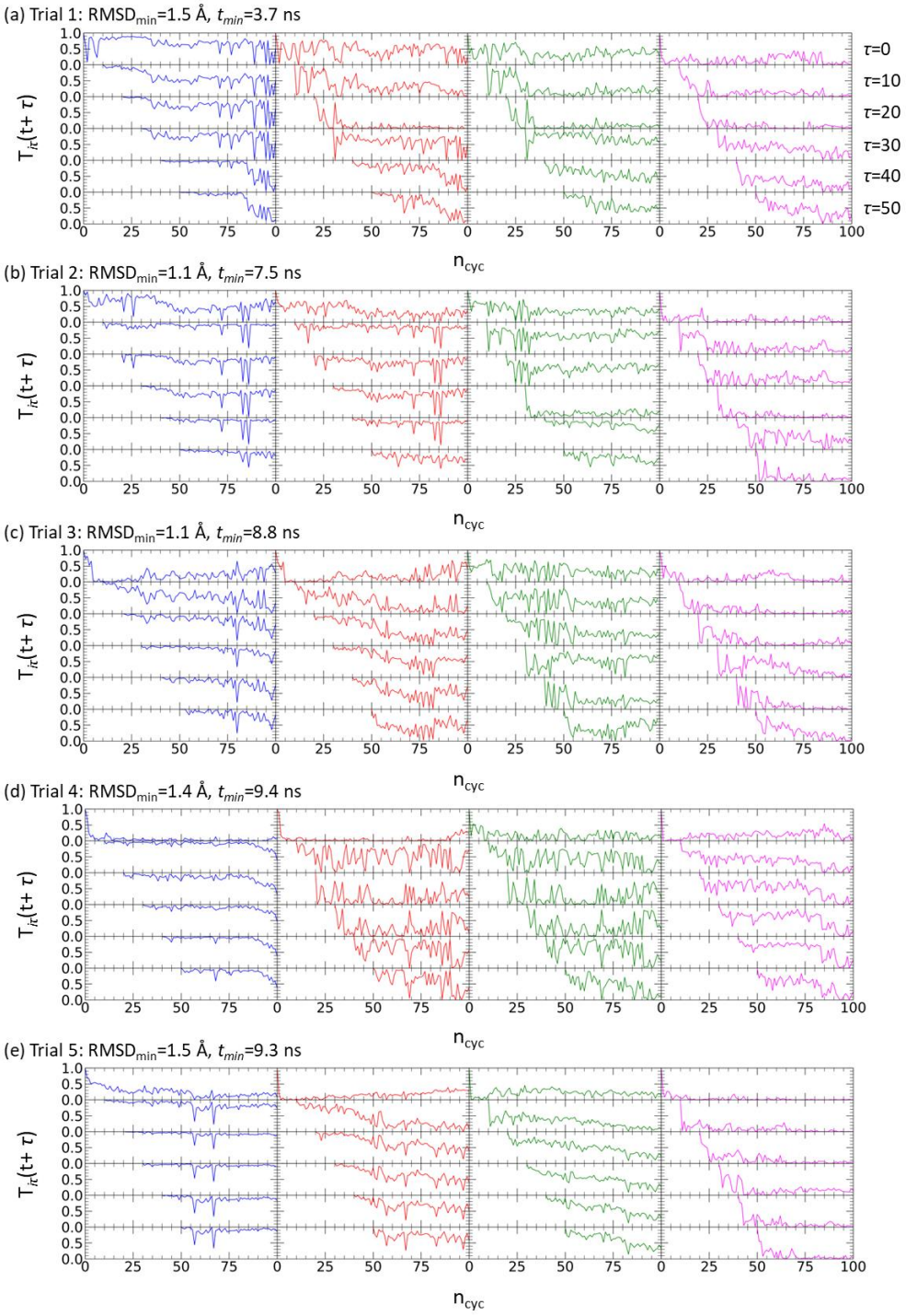

Figure 3.20: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for OPQ^{10,100,4}. PC inner products of the first (blue), second (red), third (green), and fourth (magenta) PCs at cycle $t + \tau$ and τ are shown. $RMSD_{min}$ and t_{min} are shown for references which were taken from Table 3.1.

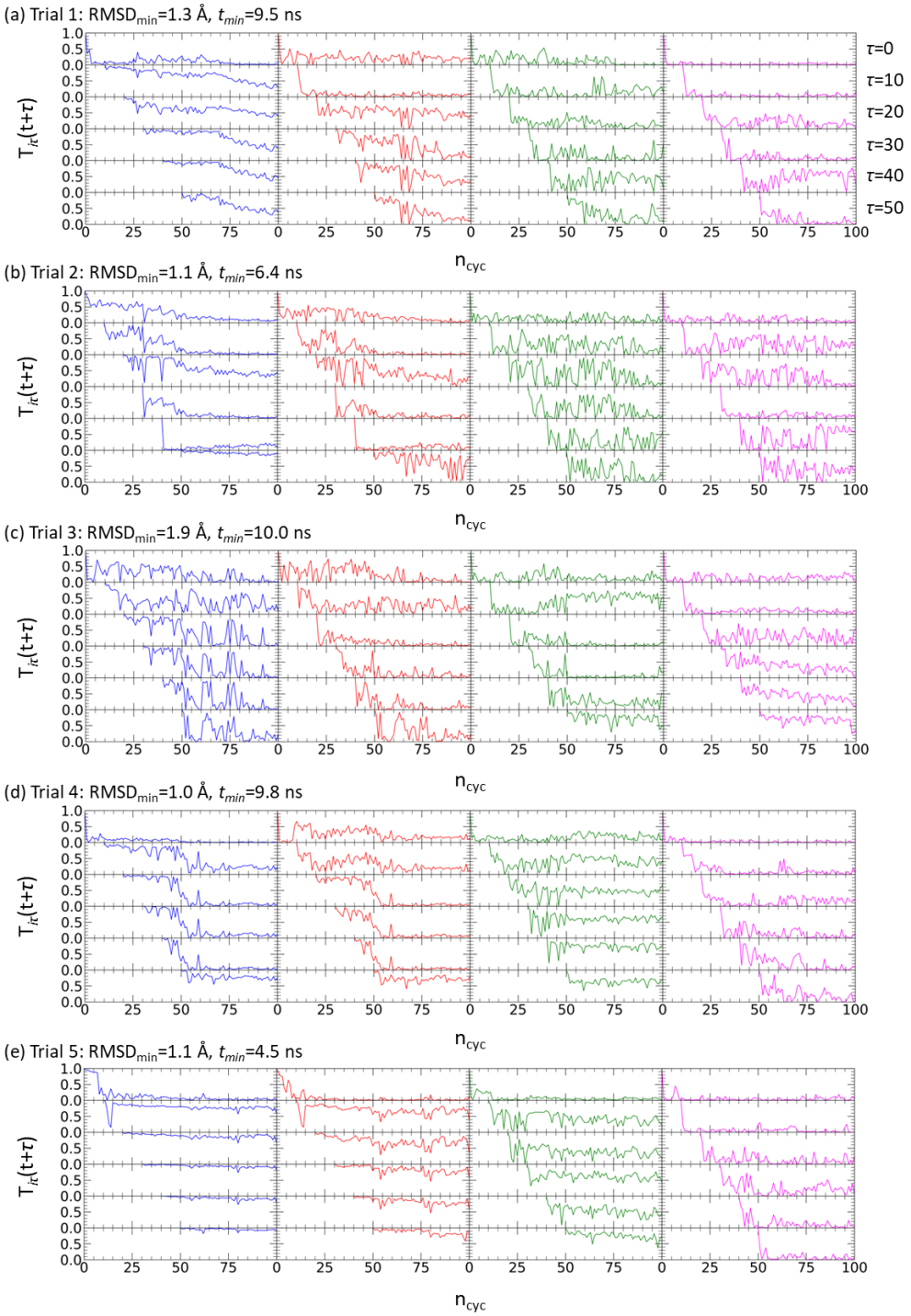

Figure 3.21: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for CLQ^{10,100,4}. The meaning of the figure is same as those described in Figure 3.20.

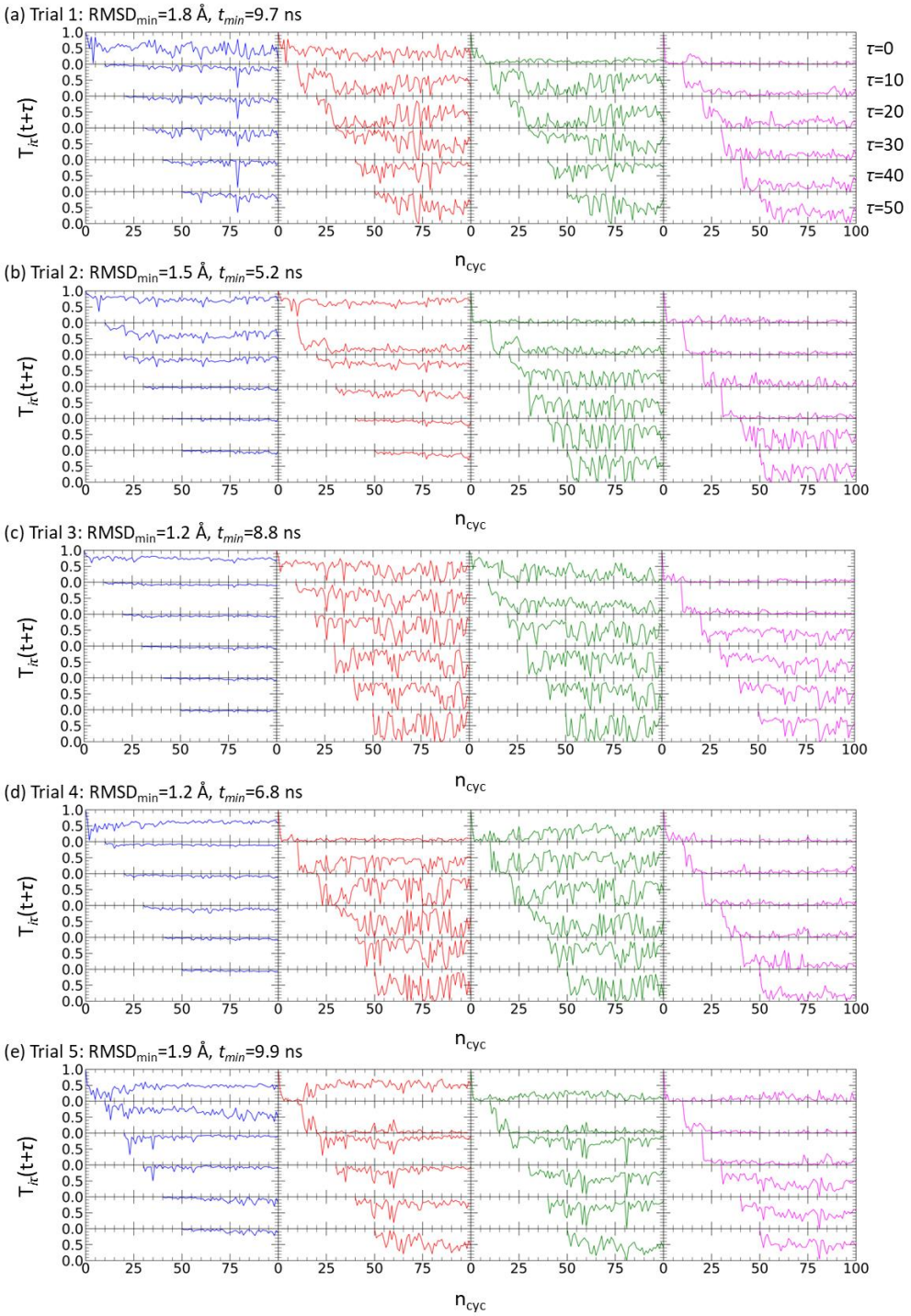

Figure 3.22: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for OPM[10,100,4]. The meaning of the figure is same as those described in Figure 3.20.

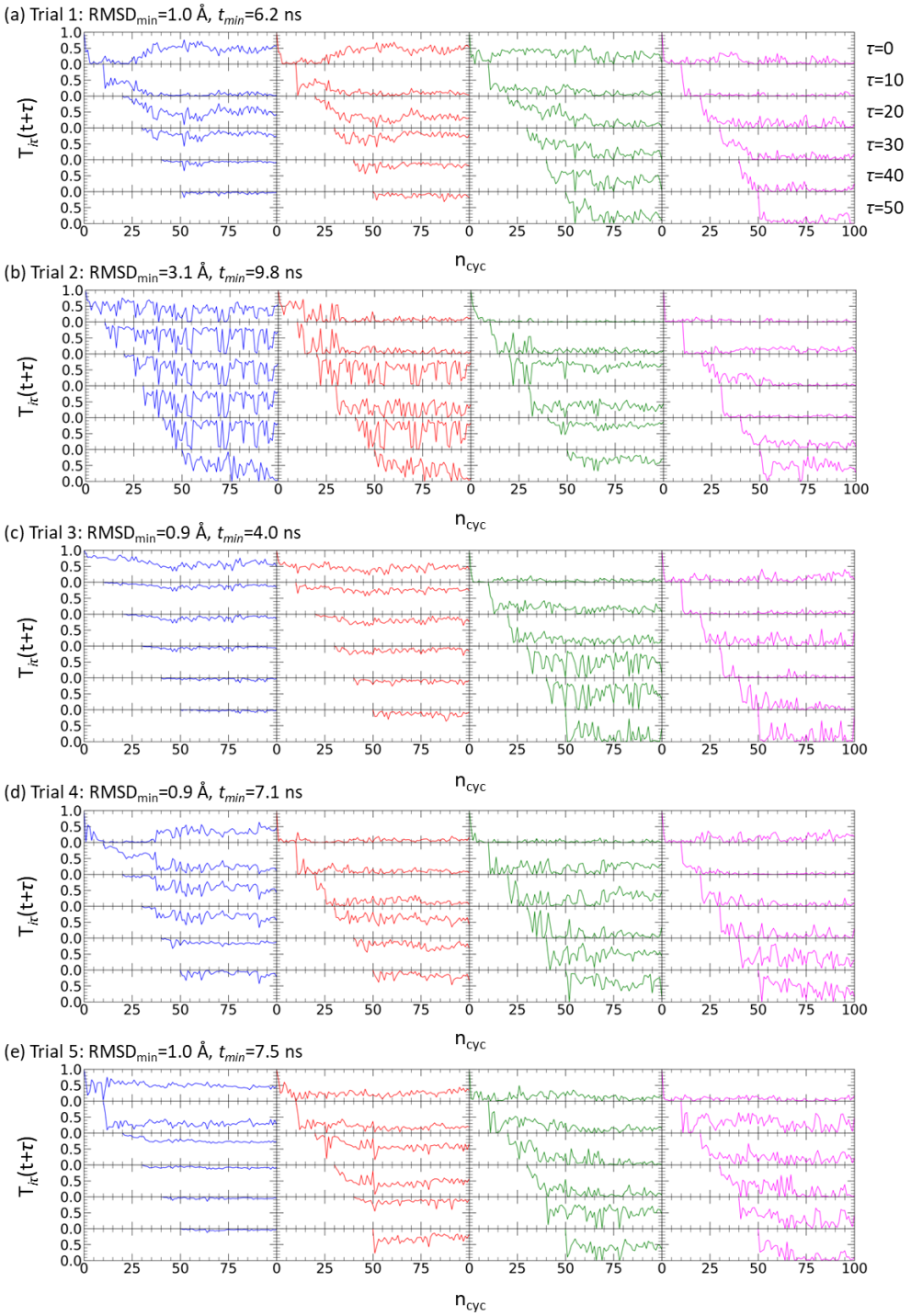

Figure 3.23: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for CLM^{10,100,4}. The meaning of the figure is same as those described in Figure 3.20.

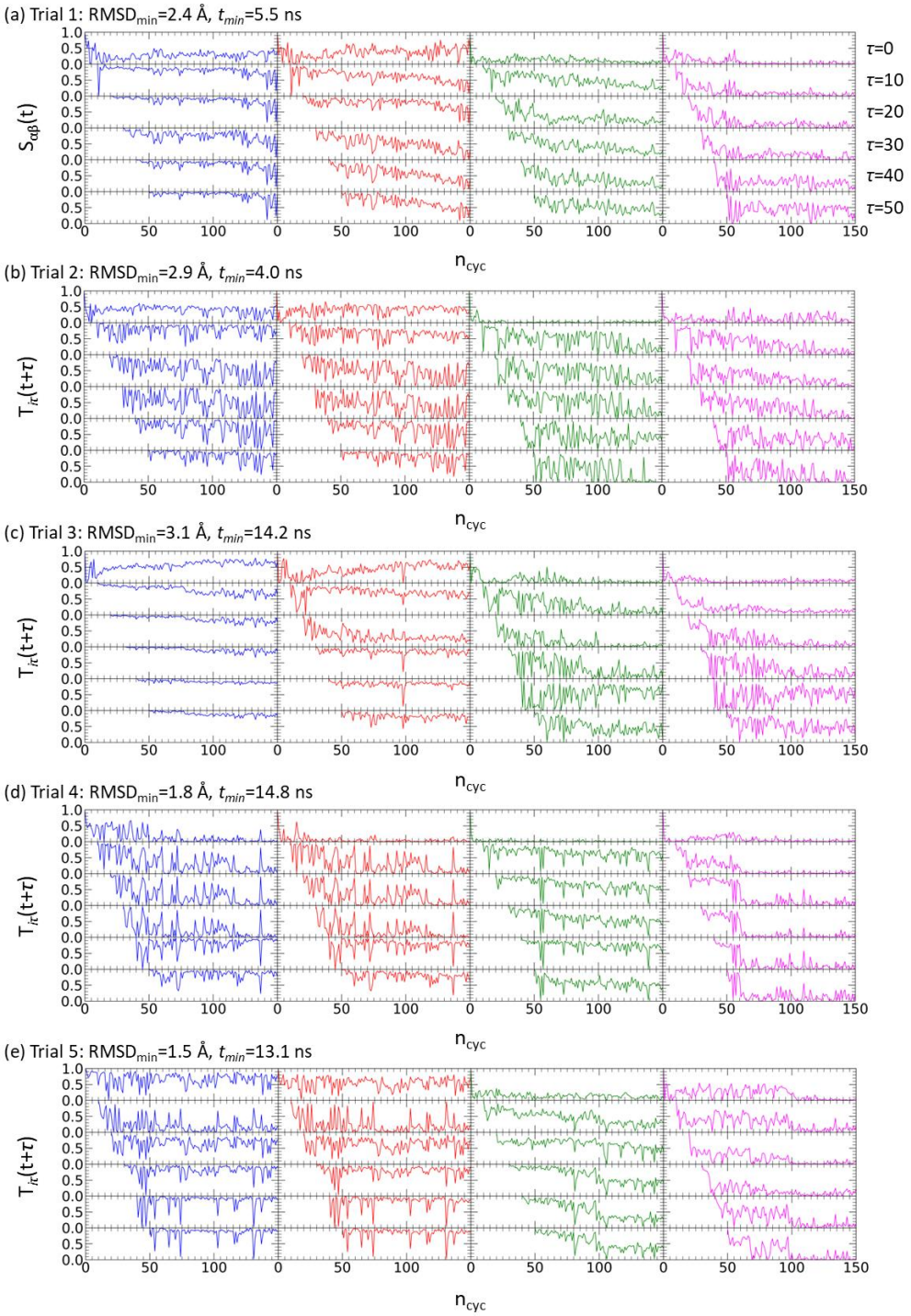

Figure 3.24: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for OPA[10,150,4]. The meaning of the figure is same as those described in Figure 3.20.

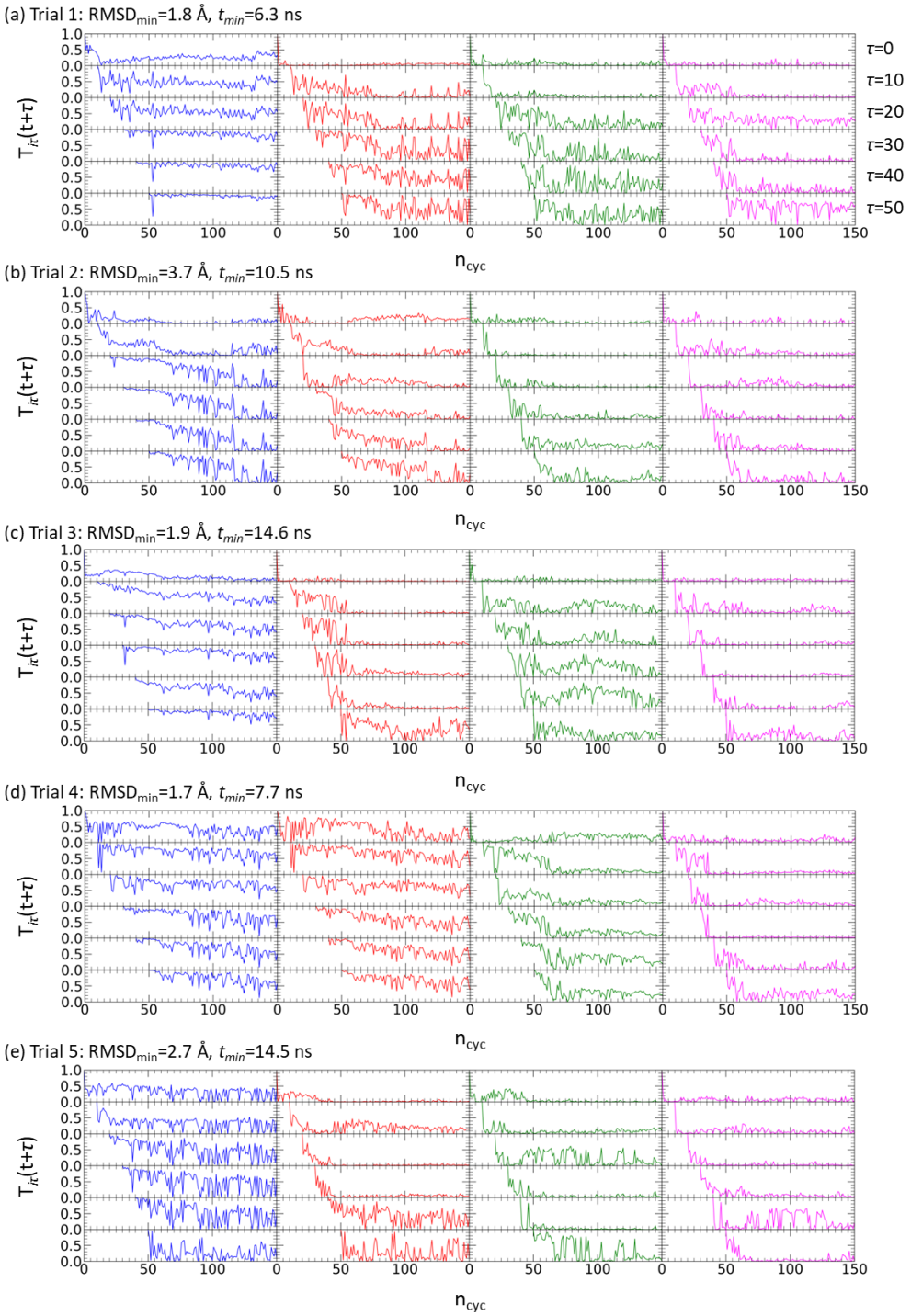

Figure 3.25: Evolution of inner products between *i*-th PCs with different reference cycles τ (0, 10, 20, 30, 40, and 50) for CLA^{10,150,4}. The meaning of the figure is same as those described in Figure 3.20.

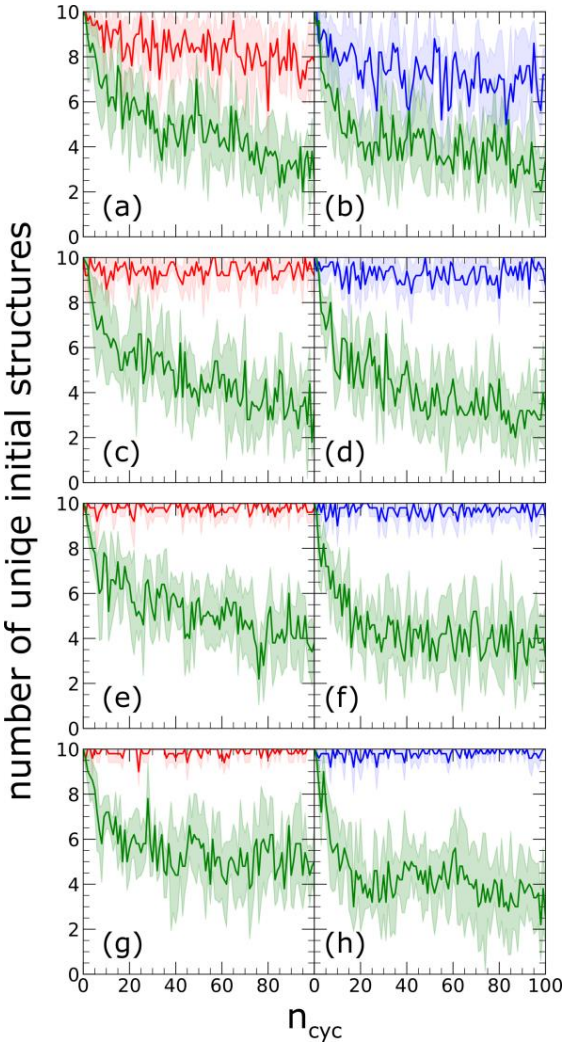

Figure 3.26: The number of unique initial structures found in $(i+1)^{th}$ cycle of eePaCS-MD with respect to the previous cycle using random-based approach and deterministic-based approach. eePaCS-MD results of (a) $OPQ^{10,100,2}$, (b) $CLQ^{10,100,2}$, (c) $OPQ^{10,100,3}$, (d) $CLQ^{10,100,3}$, (e) $OPQ^{10,100,2}$, (f) $CLQ^{10,100,4}$, (g) $OPQ^{10,100,5}$, and (h) $CLQ^{10,100,5}$ are shown. The lines in red and blue represents the results of OPQ and CLQ, respectively, with the random-based approach. The results from deterministic-based approach of OPQ and CLQ are both shown in green lines. Filled transparent areas represent the standard deviations.

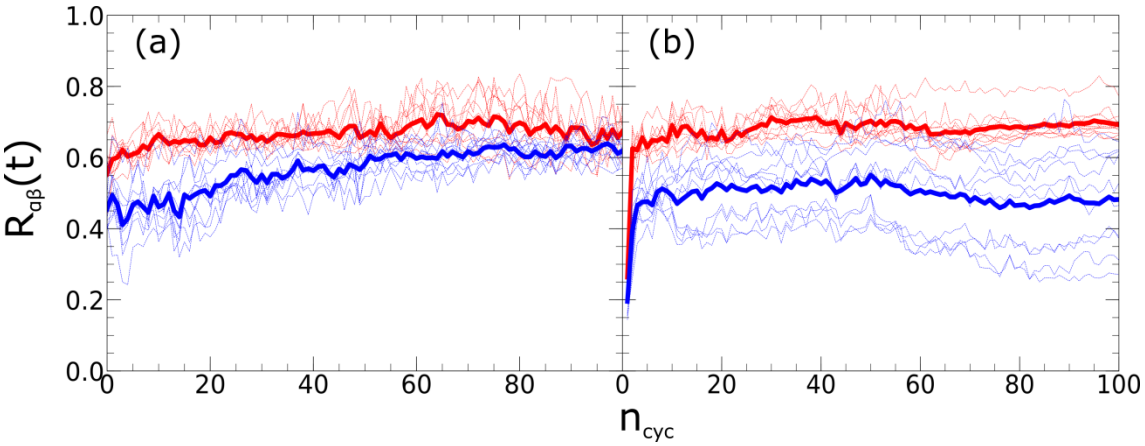

Figure 3.27: Similarity of the PC subspace spanned by the first four PCs between a pair of distinct eePaCS-MD trials at cycle *t* using (a) random- and (b) deterministic-based approach. The results for QBP from the open (red) and closed (blue) states are shown. The thin lines show the individual results and the thick lines indicate the average.

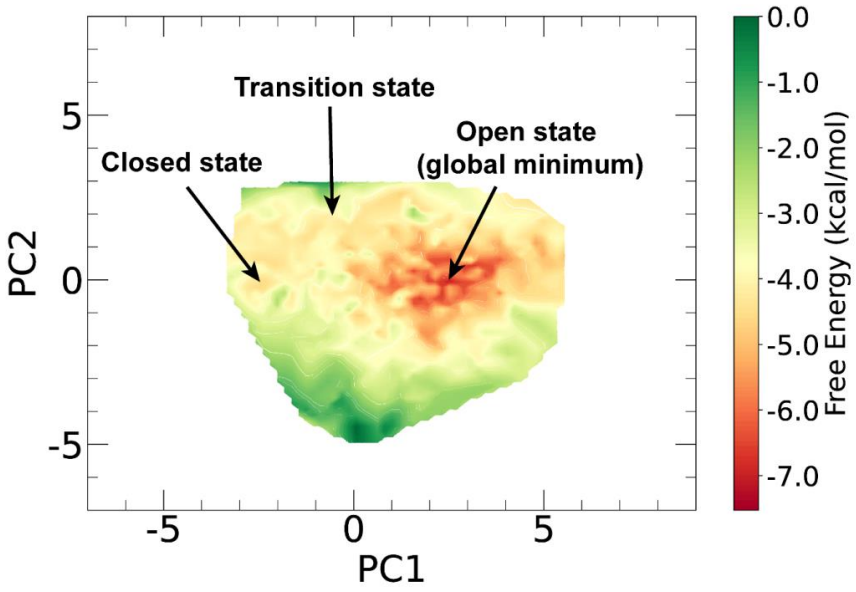

Figure 3.28: Free energy landscape of QBP determined by the Markov state model.

Table 3.1: Summary of the eePaCS-MD simulations applied to QBP, MBP, and ADK. The meanings of the simulation indices are described in the main text. Subscript "aMD" represents the simulation results of eePaCS-aMD. Average and standard deviations (values after ±) over five trials are shown.

Simulation index	n_{rep}	n_{cyc}	n_{PC}	$RMSD_{min}$ [Å]	$RMSD_{max}$ [Å]	t_{min} [ns]
QBP						
$OPQ^{10,100,2}$	10	100	2	1.6 ± 0.5	9.3 ± 1.8	7.5 ± 2.0
$OPQ^{10,100,3}$	10	100	3	1.2 ± 0.3	7.2 ± 0.5	8.5 ± 0.8
$OPQ^{10,100,4}$	10	100	4	1.3 ± 0.2	6.5 ± 0.7	7.7 ± 2.1
$OPQ^{100,10,4}$	100	10	4	1.9 ± 0.4	4.2 ± 0.5	1.0 ± 0.0
$OPQ_{aMD}{}^{10,100,4}$	10	100	4	1.3 ± 0.1	7.9 ± 0.8	4.9 ± 1.5
$OPQ^{10,100,5}$	10	100	5	1.5 ± 0.4	5.4 ± 0.9	8.3 ± 1.6
$CLQ^{10,100,2}$	10	100	2	2.5 ± 1.6	7.2 ± 2.6	5.8 ± 3.0
$CLQ^{10,100,3}$	10	100	3	1.7 ± 1.0	7.3 ± 0.8	7.0 ± 2.1
$CLQ^{10,100,4}$	10	100	4	1.3 ± 0.3	6.0 ± 1.2	8.0 ± 2.2
$CLQ^{100,10,4}$	100	10	4	3.3 ± 0.8	3.3 ± 0.6	1.0 ± 0.0
$CLQ_{aMD}{}^{10,100,4}$	10	100	4	1.3 ± 0.1	8.5 ± 0.9	6.4 ± 1.9
$CLQ^{10,100,5}$	10	100	5	1.9 ± 1.2	5.0 ± 1.2	9.5 ± 0.6
MBP						
$OPM^{10,100,4}$	10	100	4	1.5 ± 0.3	4.2 ± 0.4	8.1 ± 1.8
$OPM^{100,10,4}$	100	10	4	2.2 ± 0.5	3.2 ± 0.6	0.9 ± 0.1
$OPM_{aMD}{}^{10,100,4}$	10	100	4	1.2 ± 0.1	5.4 ± 0.4	6.7 ± 1.0
$CLM^{10,100,4}$	10	100	4	1.4 ± 0.9	5.1 ± 1.3	6.9 ± 1.9
$CLM^{100,10,4}$	100	10	4	2.3 ± 0.6	3.0 ± 0.7	1.0 ± 0.0
$CLM_{aMD}{}^{10,100,4}$	10	100	4	1.3 ± 0.1	5.9 ± 0.5	7.2 ± 1.8
ADK						
$OPA^{10,150,4}$	10	150	4	2.3 ± 0.6	9.3 ± 0.8	10.3 ± 4.6
$OPA^{100,15,4}$	100	15	4	2.6 ± 0.6	6.7 ± 0.3	1.4 ± 0.1
$OPA_{aMD}{}^{10,150,4}$	10	150	4	1.9 ± 0.4	12.9 ± 1.7	8.9 ± 3.3
$CLA^{10,150,4}$	10	150	4	2.4 ± 0.8	8.2 ± 1.3	10.7 ± 3.4
$CLA^{100,15,4}$	100	15	4	5.4 ± 0.8	4.5 ± 0.1	1.5 ± 0.0
$CLA_{extend}{}^{100,50,4}$	100	50	4	2.4 ± 0.8	10.9 ± 0.9	3.9 ± 0.8
$CLA_{aMD}{}^{10,150,4}$	10	150	4	3.2 ± 0.8	12.6 ± 2.5	9.3 ± 3.1

Table 3.2: Summary of the total computational cost, t_{1st} in ns, with different Cα RMSD criteria (3.5, 3.0, 2.5, 2.0, and 1.5 Å) measured from the opposite structure. The results of eePaCS-MD simulations applied to QBP, MBP, and ADK are shown. Average and standard deviations (values after ±) over the number of successful trials which are shown in brackets are computed for each RMSD criteria. NA means that one of the eePaCS-MD trials at the beginning satisfied the given RMSD criteria; hence t_{1st} is not shown.

Simulation index	3.5 [Å]	3.0 [Å]	2.5 [Å]	2.0 [Å]	1.5 [Å]
QBP					
$OPQ^{10,100,2}$	7.8 ± 1.6 (5)	14.0 ± 5.4 (5)	32.4 ± 28.9 (5)	41.5 ± 33.8 (4)	34.0 ± 4.5 (3)
$OPQ^{10,100,3}$	7.4 ± 4.8 (5)	15.6 ±7.2 (5)	32.0 ± 16.6 (5)	50.6 ± 21.2 (5)	56.0 ± 19.8 (4)
$OPQ^{10,100,4}$	9.0 ± 3.2 (5)	23.4 ± 16.4 (5)	36.4 ± 18.3 (5)	48.4 ± 22.8 (5)	67.8 ± 22.7 (5)
$OPQ^{10,100,5}$	11.6 ± 6.2 (5)	29.4 ± 16.1 (5)	44.6 ± 22.4 (5)	67.0 ± 25.1 (5)	70.0 ± 15.6 (3)
$OPQ_{aMD}^{10,100,4}$	7.4 ± 6.1 (5)	10.8 ± 6.8 (5)	14.4 ± 6.7 (5)	28.0 ± 10.8 (5)	40.4 ± 16.1 (5)
$CLQ^{10,100,2}$	35.3 ± 10.9 (3)	38.7 ± 12.4 (3)	44.3 ± 16.5 (3)	48.7 ± 20.9 (3)	36.0 ± 7.0 (2)
$CLQ^{10,100,3}$	44.3 ± 19.8 (4)	51.0 ± 22.5 (4)	54.8 ± 22.3 (4)	57.5 ± 22.2 (4)	61.3 ± 23.6 (4)
$CLQ^{10,100,4}$	53.8 ± 24.7 (5)	57.0 ± 25.7 (5)	60.6 ± 28.0 (5)	64.8 ± 28.4 (5)	59.3 ± 24.0 (4)
$CLQ^{10,100,5}$	40.5 ± 5.9 (4)	55.8 ± 9.0 (4)	67.8 ± 8.5 (4)	75.0 ± 12.9 (4)	89.0 ± 13.5 (4)
$CLQ_{aMD}^{10,100,4}$	39.0 ± 21.5 (5)	41.4 ± 20.8 (5)	44.2 ± 19.4 (5)	47.4 ± 19.0 (5)	52.8 ± 17.6 (5)
MBP					
$OPM^{10,100,4}$	NA	10.0 ± 7.7 (5)	23.8 ± 9.0 (5)	49.2 ± 25.9 (5)	53.7 ± 15.2 (3)
$OPM_{aMD}^{10,100,4}$	NA	5.8 ± 5.1 (5)	13.8 ± 6.2 (5)	24.0 ± 9.6 (5)	41.6 ± 13.1 (5)
$CLM^{10,100,4}$	20.4 ± 19.9 (5)	19.5 ± 9.7 (4)	27.0 ± 7.7 (4)	31.0 ± 10.1 (4)	37.0 ± 9.9 (4)
$CLM_{aMD}^{10,100,4}$	7.0 ± 6.3 (5)	16.4 ± 8.4 (5)	24.2 ± 10.8 (5)	33.2 ± 8.8 (5)	49.4 ± 20.2 (5)
ADK					
$OPA^{10,150,4}$	20.4 ± 9.2 (5)	31.0 ± 17.6 (4)	47.3 ± 24.5 (3)	102.0 ± 46.0 (2)	131 (1)
$OPA_{aMD}^{10,150,4}$	22.0 ± 12.9 (5)	28.8 ± 11.2 (5)	42.8 ± 10.5 (4)	76.3 ± 24.0 (4)	114 (1)
$CLA^{10,150,4}$	73.8 ± 27.5 (4)	81.5 ± 28.7 (4)	77.7 ± 26.0 (3)	92.3 ± 38.0 (3)	– (0)
$CLA_{aMD}^{10,150,4}$	95.0 ± 44.0 (2)	96.0 ± 44.0 (2)	108.5 ± 38.5 (2)	– (0)	– (0)

Table 3.3: Individual eePaCS-MD/aMD results for QBP. The unbracketed values from the upper and lower rows represent $RMSD_{min}$ and $RMSD_{max}$ in Å, respectively; and the values in brackets correspond to the number of cycles (n_{cyc}) required to reach $RMSD_{min}$ and $RMSD_{max}$.

Index	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$OPQ^{10,100,2}$	1.3 (45)	1.4 (83)	1.0 (59)	1.7 (100)	2.5 (90)
	12.1 (100)	8.7 (96)	7.7 (91)	10.5 (98)	7.5 (100)
$OPQ^{10,100,3}$	1.1 (85)	0.9 (92)	1.0 (95)	1.2 (76)	1.7 (78)
	6.3 (97)	7.1 (97)	7.2 (99)	7.7 (98)	7.8 (77)
$OPQ^{10,100,4}$	1.5 (37)	1.1 (75)	1.1 (88)	1.4 (94)	1.5 (93)
	7.4 (88)	6.5 (85)	6.1 (93)	5.6 (100)	7.1 (94)
$OPQ^{100,10,4}$	1.8 (10)	1.8 (10)	2.1 (10)	1.3 (10)	2.6 (10)
	4.4 (10)	4.2 (10)	4.1 (10)	5.0 (10)	3.4 (10)
$OPQ_{aMD}^{10,100,4}$	1.3 (60)	1.3 (38)	1.2 (30)	1.4 (45)	1.3 (72)
	7.5 (100)	8.0 (69)	8.4 (91)	9.0 (100)	6.6 (86)
$OPQ^{10,100,5}$	1.1 (56)	1.4 (72)	1.1 (95)	1.9 (92)	2.0 (99)
	6.7 (99)	5.4 (65)	5.4 (95)	5.3 (95)	4.0 (91)
$CLQ^{10,100,2}$	1.8 (80)	4.2 (99)	1.0 (48)	0.9 (52)	4.8 (10)
	9.1 (100)	4.5 (100)	8.3 (94)	10.4 (96)	3.9 (94)
$CLQ^{10,100,3}$	1.5 (96)	1.0 (37)	1.1 (56)	1.0 (76)	3.7 (83)
	6.1 (97)	8.0 (95)	8.2 (99)	7.2 (100)	6.8 (99)
$CLQ^{10,100,4}$	1.3 (95)	1.1 (64)	1.9 (100)	1.0 (98)	1.1 (45)
	5.6 (96)	6.9 (96)	4.0 (100)	6.3 (92)	7.4 (98)
$CLQ^{100,10,4}$	4.3 (10)	2.2 (10)	4.2 (10)	3.3 (10)	2.6 (10)
	2.8 (9)	4.1 (10)	2.5 (9)	3.2 (9)	3.7 (10)
$CLQ_{aMD}^{10,100,4}$	1.4 (91)	1.2 (52)	1.4 (64)	1.1 (77)	1.2 (36)
	6.9 (100)	9.3 (97)	8.7 (99)	8.3 (99)	9.5 (99)
$CLQ^{10,100,5}$	4.3 (91)	1.5 (98)	1.1 (85)	1.4 (100)	1.4 (100)
	3.0 (84)	4.9 (96)	6.2 (93)	4.8 (99)	6.2 (98)

Table 3.4: Individual eePaCS-MD/aMD results for MBP. The meaning of the table is same as those described in Table 3.3.

Index	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$OPM^{10,100,4}$	1.8 (97)	1.5 (52)	1.2 (88)	1.2 (68)	1.9 (99)
	3.8 (83)	4.6 (99)	4.2 (85)	4.7 (91)	3.6 (96)
$OPM^{100,10,4}$	2.8 (8)	2.2 (10)	2.6 (9)	1.5 (10)	1.7 (10)
	2.4 (5)	3.2 (10)	2.6 (9)	4.1 (10)	3.6 (10)
$OPM_{aMD}^{10,100,4}$	1.3 (65)	1.2 (82)	1.2 (52)	1.0 (74)	1.3 (64)
	5.0 (68)	5.6 (93)	5.6 (89)	5.9 (96)	4.9 (91)
$CLM^{10,100,4}$	1.0 (62)	3.1 (98)	0.9 (40)	0.9 (71)	1.0 (75)
	5.5 (92)	2.5 (88)	5.6 (65)	5.7 (98)	6.3 (95)
$CLM^{100,10,4}$	3.1 (10)	2.6 (10)	1.5 (10)	2.5 (10)	1.7 (10)
	2.1 (10)	2.6 (10)	3.7 (10)	2.8 (10)	3.8 (10)
$CLM_{aMD}^{10,100,4}$	1.4 (85)	1.1 (86)	1.3 (88)	1.2 (54)	1.3 (46)
	5.4 (98)	5.5 (89)	5.8 (77)	6.2 (99)	6.7 (99)

Table 3.5: Individual eePaCS-MD/aMD results for ADK. The meaning of the table is same as those described in Table 3.3.

Index	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$OPA^{10,150,4}$	2.4 (55)	2.9 (40)	3.1 (142)	1.8 (148)	1.5 (131)
	8.9 (147)	9.9 (150)	10.4 (148)	8.2 (143)	9.1 (147)
$OPA^{100,15,4}$	2.2 (12)	2.2 (15)	3.8 (14)	2.3 (14)	2.3 (15)
	6.2 (15)	6.4 (14)	6.9 (15)	7.0 (13)	6.8 (15)
$OPA_{aMD}^{10,150,4}$	1.5 (114)	2.6 (40)	1.6 (133)	2.0 (71)	1.8 (86)
	14.2 (140)	13.2 (149)	15.1 (149)	10.9 (146)	11.1 (147)
$CLA^{10,150,4}$	1.8 (63)	3.7 (105)	1.9 (146)	1.7 (77)	2.7 (145)
	9.1 (150)	6.6 (137)	7.2 (149)	10.3 (142)	8.0 (147)
$CLA^{100,15,4}$	4.1 (15)	6.4 (15)	5.5 (15)	5.0 (15)	6.0 (15)
	4.7 (15)	4.4 (15)	4.4 (14)	4.4 (14)	4.7 (15)
$CLA_{aMD}^{10,150,4}$	3.7 (60)	3.7 (97)	2.3 (70)	4.2 (88)	2.2 (149)
	13.7 (140)	12.1 (149)	16.0 (148)	12.8 (149)	8.4 (147)

Table 3.6: Summary of the eePaCS-MD simulations applied to QBP using deterministic-based approach. The meaning of the table is same as those described in Table 3.1. eePaCS-MD results using random-based approach are shown for comparison which were taken from Table 3.1.

Simulation index	n_{cyc}	n_{rep}	n_{PC}	$RMSD_{min}$ [Å]	$RMSD_{max}$ [Å]	t_{min} [ns]
Random						
$OPQ^{10,100,2}$	100	10	2	1.6 ± 0.5	9.3 ± 1.8	7.5 ± 2.0
$OPQ^{10,100,3}$	100	10	3	1.2 ± 0.3	7.2 ± 0.5	8.5 ± 0.8
$OPQ^{10,100,4}$	100	10	4	1.3 ± 0.2	6.5 ± 0.7	7.7 ± 2.1
$OPQ^{10,100,5}$	100	10	5	1.5 ± 0.4	5.4 ± 0.9	8.3 ± 1.6
$CLQ^{10,100,2}$	100	10	2	2.5 ± 1.6	7.2 ± 2.6	5.8 ± 3.0
$CLQ^{10,100,3}$	100	10	3	1.7 ± 1.0	7.3 ± 0.8	7.0 ± 2.1
$CLQ^{10,100,4}$	100	10	4	1.3 ± 0.3	6.0 ± 1.2	8.0 ± 2.2
$CLQ^{10,100,5}$	100	10	5	1.9 ± 1.2	5.0 ± 1.2	9.5 ± 0.6
Deterministic						
$OPQ^{10,100,2}$	100	10	2	1.2 ± 0.1	8.6 ± 1.1	5.7 ± 0.5
$OPQ^{10,100,3}$	100	10	3	1.2 ± 0.2	7.6 ± 1.8	5.3 ± 2.3
$OPQ^{10,100,4}$	100	10	4	1.2 ± 0.2	8.1 ± 1.0	5.8 ± 1.1
$OPQ^{10,100,5}$	100	10	5	1.3 ± 0.2	7.7 ± 2.0	7.0 ± 1.5
$CLQ^{10,100,2}$	100	10	2	1.9 ± 1.5	7.9 ± 2.4	6.6 ± 2.6
$CLQ^{10,100,3}$	100	10	3	1.5 ± 0.8	6.1 ± 1.5	8.8 ± 1.2
$CLQ^{10,100,4}$	100	10	4	2.6 ± 1.9	6.0 ± 1.8	7.3 ± 1.5
$CLQ^{10,100,5}$	100	10	5	2.7 ± 2.0	6.2 ± 2.2	8.5 ± 1.9

Table 3.7: Summary of the total computational cost, t_{1st} in ns, with different Cα RMSD criteria (3.5, 3.0, 2.5, 2.0, and 1.5 Å) obtained by eePaCS-MD simulations applied to QBP using deterministic-based approach. The meaning of the table is same as those described in Table 3.2. The results using random-based approach are shown for comparison.

Simulation index	3.5 [Å]	3.0 [Å]	2.5 [Å]	2.0 [Å]	1.5 [Å]
Random					
$OPQ^{10,100,2}$	7.8 ± 1.6 (5)	14.0 ± 5.4 (5)	32.4 ± 28.9 (5)	41.5 ± 33.8 (4)	34.0 ± 4.5 (3)
$OPQ^{10,100,3}$	7.4 ± 4.8 (5)	15.6 ±7.2 (5)	32.0 ± 16.6 (5)	50.6 ± 21.2 (5)	56.0 ± 19.8 (4)
$OPQ^{10,100,4}$	9.0 ± 3.2 (5)	23.4 ± 16.4 (5)	36.4 ± 18.3 (5)	48.4 ± 22.8 (5)	67.8 ± 22.7 (5)
$OPQ^{10,100,5}$	11.6 ± 6.2 (5)	29.4 ± 16.1 (5)	44.6 ± 22.4 (5)	67.0 ± 25.1 (5)	70.0 ± 15.6 (3)
$CLQ^{10,100,2}$	35.3 ± 10.9 (3)	38.7 ± 12.4 (3)	44.3 ± 16.5 (3)	48.7 ± 20.9 (3)	36.0 ± 7.0 (2)
$CLQ^{10,100,3}$	44.3 ± 19.8 (4)	51.0 ± 22.5 (4)	54.8 ± 22.3 (4)	57.5 ± 22.2 (4)	61.3 ± 23.6 (4)
$CLQ^{10,100,4}$	53.8 ± 24.7 (5)	57.0 ± 25.7 (5)	60.6 ± 28.0 (5)	64.8 ± 28.4 (5)	59.3 ± 24.0 (4)
$CLQ^{10,100,5}$	40.5 ± 5.9 (4)	55.8 ± 9.0 (4)	67.8 ± 8.5 (4)	75.0 ± 12.9 (4)	89.0 ± 13.5 (4)
Deterministic					
$OPQ^{10,100,2}$	6.2 ± 1.2 (5)	12.0 ± 1.8 (5)	17.4 ± 3.4 (5)	31.0 ± 8.8 (5)	47.8 ± 11.1 (5)
$OPQ^{10,100,3}$	8.2 ± 6.1 (5)	15.8 ± 9.8 (5)	23.6 ± 16.6 (5)	40.0 ± 29.1 (5)	33.5 ± 6.3 (4)
$OPQ^{10,100,4}$	3.6 ± 1.4 (5)	10.0 ± 2.8 (5)	16.8 ± 5.3 (5)	23.0 ± 9.1 (5)	40.5 ± 7.4 (4)
$OPQ^{10,100,5}$	6.4 ± 3.3 (5)	15.2 ± 6.0 (5)	27.8 ± 12.8 (5)	36.8 ± 10.1 (5)	46.5 ± 18.3 (4)
$CLQ^{10,100,2}$	41.0 ± 23.0 (4)	45.0 ± 24.5 (4)	50.0 ± 23.5 (4)	53.0 ± 23.8 (4)	56.8 ± 27.2 (4)
$CLQ^{10,100,3}$	48.4 ± 29.4 (5)	52.8 ± 30.3 (5)	44.0 ± 22.0 (4)	52.5 ± 27.8 (4)	58.8 ± 27.1 (4)
$CLQ^{10,100,4}$	39.3 ± 14.3 (3)	44.7 ± 13.3 (3)	51.0 ± 14.0 (3)	54.3 ± 12.7 (3)	57.7 ± 11.1 (3)
$CLQ^{10,100,5}$	34.0 ± 9.1 (3)	38.0 ± 8.3 (3)	40.0 ± 7.8 (3)	41.3 ± 7.8 (3)	43.3 ± 8.2 (3)

Table 3.8: Individual eePaCS-MD results for QBP using deterministic-based approach. The meaning of the table is same as those described in Table 3.3.

Index	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
$OPQ^{10,100,2}$	1.2 (55)	1.3 (61)	1.1 (54)	1.4 (64)	1.2 (51)
	7.9 (96)	9.5 (100)	6.9 (98)	9.8 (97)	8.7 (97)
$OPQ^{10,100,3}$	1.2 (53)	1.1 (38)	1.7 (98)	1.0 (44)	1.2 (34)
	9.9 (94)	7.0 (95)	4.8 (98)	9.1 (95)	7.1 (73)
$OPQ^{10,100,4}$	1.2 (46)	1.1 (76)	1.0 (65)	1.2 (48)	1.7 (53)
	7.8 (100)	7.4 (100)	9.5 (97)	6.7 (96)	8.9 (93)
$OPQ^{10,100,5}$	1.2 (52)	1.7 (67)	1.1 (96)	1.2 (63)	1.2 (73)
	7.1 (100)	11.4 (100)	5.7 (98)	6.6 (97)	7.7 (100)
$CLQ^{10,100,2}$	1.5 (100)	1.1 (44)	4.8 (91)	1.1 (61)	0.9 (34)
	6.0 (99)	8.7 (93)	4.3 (86)	9.4 (99)	11.1 (96)
$CLQ^{10,100,3}$	3.0 (100)	1.0 (69)	1.1 (78)	1.5 (100)	1.0 (93)
	4.2 (100)	8.1 (99)	6.4 (100)	4.8 (100)	7.2 (94)
$CLQ^{10,100,4}$	5.1 (68)	1.1 (92)	1.1 (73)	4.9 (85)	0.9 (49)
	4.4 (91)	6.2 (99)	8.4 (96)	3.6 (99)	7.2 (100)
$CLQ^{10,100,5}$	5.1 (99)	1.1 (100)	1.1 (48)	5.2 (95)	1.1 (83)
	3.3 (94)	8.9 (99)	7.6 (98)	3.7 (88)	7.3 (100)

Table 3.9: Summary of various methods related to eePaCS-MD.

Method	Selection method	Collective variable	Character of the initial structure	Selection process	Requirement of full trajectory search for resampling
eePaCS-MD	Convex hull	PCA	Edge	Random	No
nt-PaCS-MD[107]	Gram-Schmidt orthogonolization	RMSD	Non-edge	Random	No
OFLOOD[108]	Clustering	PCA	Non-edge	Random	Yes
SACS[109]	Histogram	PCA	Non-edge	Random	Yes
SDS[110]	Inner product	PCA	Non-edge	Deterministic	Yes
Extended SDS[111]	Inner product	PCA	Non-edge	Deterministic	No

Table 3.10: Sampling efficiencies of eePaCS-MD/aMD and the other related methods described in this chapter. The results of eePaCS-MD/aMD with n_{PC} = 4 are shown as a reference. The SACS and SDS superscripts show the number of PC dimensions applied in the method. The column of 'Ref' shows the reference number. t_{1st} is defined based on different Cα RMSD criteria (OPQ/CLQ: 2, OPM/CLM: 1, OPA/CLA: 2 Å) used in the references, and the numbers in the parentheses indicate the number of trials that satisfied the RMSD criteria. NA means that no information is provided in the original paper.

Target	Method	Ref	# trials	t_{cyc} [ps]	n_{rep}	n_{cyc}	t_{tot} [ns]	$RMSD_{min}$ [Å]	t_{1st} [ns]
OPQ	eePaCS-MD		5	100	10	100	100	1.3 ± 0.2	48.4 ± 22.8 (5)
	eePaCS-aMD		5	100	10	100	100	1.3 ± 0.1	28.0 ± 10.8 (5)
	nt-PaCS-MD	107	1	100	10	200	200	1.7	121 (1)
	OFLOOD	108	1	20	100	100	200	1.7	NA
CLQ	eePaCS-MD		5	100	10	100	100	1.3 ± 0.3	64.8 ± 28.4 (5)
	eePaCS-aMD		5	100	10	100	100	1.3 ± 0.1	47.4 ± 19.0 (5)
	nt-PaCS-MD	107	1	100	10	100	100	1.9	27 (1)
OPM	eePaCS-MD		5	100	10	100	100	1.5 ± 0.3	– (0)
	eePaCS-aMD		5	100	10	100	100	1.2 ± 0.1	74 (1)
	$SACS^{PM1,2}$	109	1	100	10	100	100	1.0	89 (1)
		109	1	100	50	50	250	1.0	125 (1)
		109	1	100	100	50	500	0.9	260 (1)
	$SACS^{PM1-3}$	109	1	100	10	100	100	2.0	– (0)
		109	1	100	50	100	500	1.4	– (0)
		109	1	100	100	100	1000	1.0	660 (1)
	$SACS^{PM1-4}$	109	1	100	10	100	100	1.9	– (0)
		109	1	100	50	100	500	1.3	– (0)
		109	1	100	100	100	1000	1.0	– (0)
	SDS^{PM10}	110	1	100	100	50	500	~1.0	~110 (1)
	SDS^{PM30}	110	1	100	50	50	250	~1.0	~125 (1)
		110	1	100	100	50	500	0.8	~150 (1)
	Extended SDS	111	1	100	10	50	50	~1.1	– (0)
		111	1	100	50	50	250	~1.0	~200 (1)
		111	1	100	100	50	500	~1.2	– (0)
CLM	eePaCS-MD		5	100	10	100	100	1.4 ± 0.9	57.5 ± 17.7 (4)
	eePaCS-aMD		5	100	10	100	100	1.3 ± 0.1	– (0)
	SDS^{PM10}	110	1	100	100	50	500	< 1.0	~60 (1)
	SDS^{PM30}	110	1	100	100	50	500	0.9	~100 (1)
OPA	eePaCS-MD		5	100	10	150	150	2.3 ± 0.6	102.0 ± 46.0 (2)
	eePaCS-aMD		5	100	10	150	150	1.9 ± 0.4	76.3± 24.0 (4)
	nt-PaCS-MD		5	100	10	100	150	2.7 ± 0.4	44 (1)
CLA	eePaCS-MD		5	100	10	150	150	2.4 ± 0.8	92.3 ± 38.0 (3)
	eePaCS-aMD		5	100	10	150	150	3.2 ± 0.8	– (0)
	nt-PaCS-MD		5	100	10	100	150	2.7 ± 0.7	149 (1)

Table 3.11: Summary of the total computational cost, t_{1st} in ns, with different Cα RMSD criteria (3.5, 3.0, 2.5, 2.0, and 1.5 Å) obtained by eePaCS-MD/aMD and nt-PaCS-MD simulations applied to OPA and CLA. The meaning of the table is same as those described in Table 3.2.

Simulation index	3.5 [Å]	3.0 [Å]	2.5 [Å]	2.0 [Å]	1.5 [Å]
OPA					
eePaCS-MD	20.4 ± 9.2 (5)	31.0 ± 17.6 (4)	47.3 ± 24.5 (3)	102.0 ± 46.0 (2)	131 (1)
eePaCS-aMD	22.0 ± 12.9 (5)	28.8 ± 11.2 (5)	42.8 ± 10.5 (4)	76.3 ± 24.0 (4)	114 (1)
nt-PaCS-MD	20.4 ± 15.3 (5)	60.8 ± 38.7 (4)	27 (1)	44 (1)	– (0)
CLA					
eePaCS-MD	73.8 ± 27.5 (4)	81.5 ± 28.7 (4)	77.7 ± 26.0 (3)	92.3 ± 38.0 (3)	– (0)
eePaCS-aMD	95.0 ± 44.0 (2)	96.0 ± 44.0 (2)	108.5 ± 38.5 (2)	– (0)	– (0)
nt-PaCS-MD	80.3 ± 40.6 (4)	65.0 ± 27.2 (3)	77.7 ± 24.2 (3)	149 (1)	– (0)

Chapter 4
Comparison of eePaCS-MD Utilizing Cartesian Coordinate PCA and Distance-based PCA

In this chapter, the conformational sampling efficiency of eePaCS-MD utilizing Cartesian coordinate PCA and distance-based PCA are compared. Section 4.1 introduces the motivation of using distance-based PCA in eePaCS-MD. Section 4.2 describes the procedure of eePaCS-MD utilizing distance-based PCA and the target system used to assess its performance. Section 4.3 discusses the results obtained by eePaCS-MDs utilizing Cartesian coordinate PCA and distance-based PCA, and compares the conformational sampling efficiency of the two methods. Finally, the conclusion of this chapter is given in Section 4.4.

4.1 Introduction

As described in Section 2.3.2, principal component analysis (PCA) is a powerful technique to reduce the high-dimensional MD trajectory data to a low-dimensional reaction coordinate. While various PCA methods have been proposed, probably the most commonly used PCA method is the Cartesian coordinate PCA.[11,20,21] To observe the internal motion of a system, it is required to first remove the external motion, i.e., overall translation and rotation, from the system which is usually achieved by instantaneously fitting each coordinates from the trajectory to a reference coordinate. Separation of external and internal motion is essential to achieve a well-resolved free energy landscape. However, such separation can be difficult even for relatively rigid systems.[94,147] Apart from Cartesian coordinates, internal coordinates, such as dihedral angles[93] and distance-based measures,[95] can be used as input data in a PCA. It has been shown that PCA utilizing internal coordinates can yield better-resolved energy landscapes than applying Cartesian coordinates.[93–95]

Given the above fact, it is debatable how the conformational sampling efficiency of eePaCS-MD can be influenced by the choice of internal coordinate PCA. In this study, the performance of eePaCS-MD utilizing Cartesian coordinate PCA (cPCA) and distance-based PCA (dPCA) are compared. While the optimal choice of internal coordinate PCA will depend on the specific molecule, dPCA was suggested as a versatile approach that balances the number of PCs required to show good convergence of the cumulative fluctuations of protein motions and to achieve

well-resolved free energy landscapes.[95]

Here, we show that the conformational sampling efficiency of eePaCS-MD for the open-close transitions of adenylate kinase (ADK) is not affected by the choice of the PCA method. eePaCS-MD utilizing dPCA gave similar performance as those obtained by cPCA.

4.2 Materials and methods

The conformational sampling efficiency of eePaCS-MD utilizing dPCA was assessed for the open-closed transitions of ADK. This target was chosen because it is the most challenging among the three targets (QBP, MBP, and ADK) investigated in the previous study and expected that differences between cPCA and dPCA will be more evident.

The procedure of eePaCS-MD utilizing dPCA is the same as those described in Section 3.2.1 where cPCA was simply replaced with dPCA. In dPCA, distances of *all* Cα atom pairs except for the first few atoms from the N- and C-terminals were excluded. The element of the covariance matrix, $\sigma_{\mu\nu}$, in dPCA is given by $\sigma_{\mu\nu} = \langle (D_\mu - \langle D_\mu \rangle)(D_\nu - \langle D_\nu \rangle) \rangle$ where μ, ν represents the atom pairs and D is the Cα distance of those pairs. dPCA was calculated utilizing MDTraj package.[148] The parameters of eePaCS-MD utilizing dPCA were chosen so that it matches the conditions with those simulated with cPCA described in Section 3.2.4. The number of replicas (n_{rep}), the total simulation cycles (n_{cyc}), and number of PCs (n_{PC}) were fixed to n_{rep} = 10, n_{cyc} = 150, and n_{PC} = 4 respectively. The MD simulation time per cycle (t_{cyc}) was fixed to 100 ps and MD trajectories were saved every 1 ps. Five distinct trials of eePaCS-MD from different initial structures were performed for both the open and closed states of ADK. The initial structures were taken from the final structures obtained by the five individual equilibration simulations described in Section 3.2.3 and were used to perform the preliminary eePaCS-MD cycles, i.e. cycle 0. To analyze the performance of eePaCS-MD, we introduced the same quantity measures as described in Section 3.2.5 such as $\mathrm{RMSD}_{\mathrm{min}}$ and t_{min}.

Hereafter, for example, eePaCS-MD utilizing cPCA and dPCA are referred to as $\mathrm{eePaCS\text{-}MD}_{\mathrm{cPCA}}$ and $\mathrm{eePaCS\text{-}MD}_{\mathrm{dPCA}}$, respectively. Similar indices are used for simulations of ADK starting from the open (OP) and closed (CL) states; for example, ADK starting from the open state utilizing dPCA is denoted as $\mathrm{OPA}_{\mathrm{dPCA}}$ where A indicates ADK.

4.3 Results and Discussion

The eePaCS-MD$_{dPCA}$ results for ADK (OPA$_{dPCA}$ and CLA$_{dPCA}$) are summarized in Tables 4.1–4.3. For OPA$_{dPCA}$, RMSD$_{min}$ reached 2.6 ± 0.5 Å at t_{min} = 11.3 ± 2.3 ns where RMSD$_{min}$ is 0.3 Å higher than OPA$_{cPCA}$. In the case of CLA$_{dPCA}$, RMSD$_{min}$ reached 2.0 ± 0.5 Å at t_{min} = 13.1 ± 1.3 ns where RMSD$_{min}$ is 0.4 Å lower than CLA$_{cPCA}$. For both OPA$_{dPCA}$ and CLA$_{dPCA}$, t_{min} increased by 10% and 22% compared to OPA$_{cPCA}$ and CLA$_{cPCA}$, respectively. In addition, eePaCS-MD$_{cPCA}$ tended to show lower t_{1st} than eePaCS-MD$_{dPCA}$ against various RMSD criteria; although meaningful comparison is somewhat difficult to make since t_{1st} was averaged over less successful trials as the RMSD criteria became stricter (Table 4.2). Among the five trials of OPA$_{dPCA}$, the open-to-closed transitions (RMSD$_{min}$ ≤ 1.5 Å) were not observed within 150 cycles, whereas one trial was successful in reaching the opposite state for CLA$_{dPCA}$ (Table 4.3). In contrast, for OPA$_{cPCA}$, one trial was successful to observe the open-to-closed transitions but all five trials failed to reach the opposite state for CLA$_{cPCA}$.

Next, the Cα RMSD$_{min/max}$ profiles of eePaCS-MD$_{cPCA}$ and eePaCS-MD$_{dPCA}$ were compared as shown in Figure 4.1. The RMSD$_{min}$ of eePaCS-MD$_{cPCA}$ tended to decay faster than eePaCS-MD$_{dPCA}$ for both OPA and CLA during the first ~90 cycles. The cumulative fluctuations obtained by the first four PCs of cPCA covered slightly larger fraction of the collective variance than dPCA during the first few tens of cycles, but no significant difference was found among the two PCA methods (Figure 4.2). The cumulative fraction at cycle 0 for OPA$_{cPCA}$, OPA$_{dPCA}$, CLA$_{cPCA}$, and CLA$_{dPCA}$ were 0.52, 0.52, 0.34, and 0.31, respectively, showing no remarkable difference. These results suggest that eePaCS-MD utilizing cPCA and dPCA captured the same amount of overall motion of ADK during the eePaCS-MD simulation.

The number of vertex structures obtained by eePaCS-MD$_{cPCA}$ and eePaCS-MD$_{dPCA}$ are compared as shown in Figure 4.3, suggesting that the selection rates of the vertex structures, i.e., n_{rep}/n_{vertex}, are not affected by the choice of the PCA method. In Figure 4.4, the time evolution of the subspace spanned by the first four PCs, i.e., $R_{\alpha\beta}(t)$, obtained by eePaCS-MD$_{cPCA}$ and eePaCS-MD$_{dPCA}$ are compared. In the beginning of eePaCS-MD$_{cPCA/dPCA}$, $R_{\alpha\beta}(t)$ of cycle 0 that was determined from the PCs of independent preliminary MDs were 0.47, 0.38, 0.26, and 0.22 for OPA$_{cPCA}$, OPA$_{dPCA}$, CLA$_{cPCA}$, and CLA$_{dPCA}$, respectively, indicating that the first four PC coordinates from cPCA had higher correlation with those of the other trials from the beginning compared to dPCA. $R_{\alpha\beta}(t)$ converged to ~0.70 and ~0.55 for OPA$_{cPCA/dPCA}$ and CLA$_{cPCA/dPCA}$, respectively, indicating that there are initial condition dependencies regardless of the PCA method.

The computational complexity of dPCA may be problematic for large biomolecular systems because of the quadratic scaling of the number of distances with respect to the size of the molecule. It has been shown that it is sufficient to include only relatively few selected distances in dPCA. Similar free energy landscapes were obtained using *all* Cα distances and selected distance pairs that are less than 8 Å apart in the reference structure.[95] Based on this fact, eePaCS-MDs utilizing dPCA with Cα atom pairs that are less than 8 Å apart from the initial structures of eePaCS-MDs were additionally simulated which are denoted as eePaCS-MD$_{\mathrm{dPCA(cutoff)}}$. Contrary to the expectation, eePaCS-MD$_{\mathrm{dPCA(cutoff)}}$ performed worse than the original eePaCS-MD$_{\mathrm{dPCA}}$, as shown in Tables 4.1–4.3. This suggests that it is important to include *all* Cα distances in dPCA to capture the intrinsic anharmonic nature of proteins during eePaCS-MD simulations.

4.4 Conclusion

In this chapter, eePaCS-MDs utilizing cPCA and dPCA were compared for the open-close conformational transitions of ADK. The above results suggest that the conformational sampling efficiency is not affected by the choice of the PCA method, at least for this particular target. Furthermore, dPCA with Cα atom pairs that are less than 8 Å apart from the initial structures of eePaCS-MDs were considered, suggesting that it is important to include *all* Cα distances in dPCA to capture the intrinsic anharmonic nature of proteins during eePaCS-MD simulations. cPCA and dPCA both have their own problems. In cPCA, separation of external and internal motion is not straightforward even for relatively rigid systems, whereas the computational complexity of dPCA scales quadratically with the number of atoms, which makes it unfeasible for large biomolecular systems. Considering the results obtained in this study, eePaCS-MD utilizing cPCA is recommended as a first choice of the PCA method, although the optimal choice of PCA method is expected to depend on the target system.

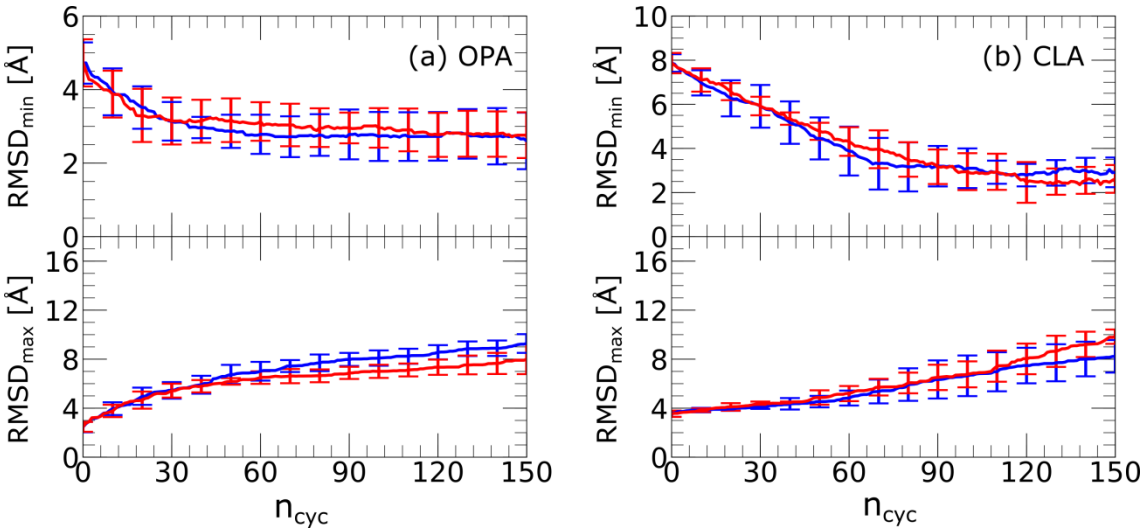

Figure 4.1: Cα RMSD ($RMSD_{min}$ and $RMSD_{max}$) profile as a function of n_{cyc} obtained by eePaCS-MD_{cPCA} (blue) and eePaCS-MD_{dPCA} (red). eePaCS-MDs starting from (a) the open (OPA) and (b) closed states (CLA) are shown. The error bars indicate the standard deviations.

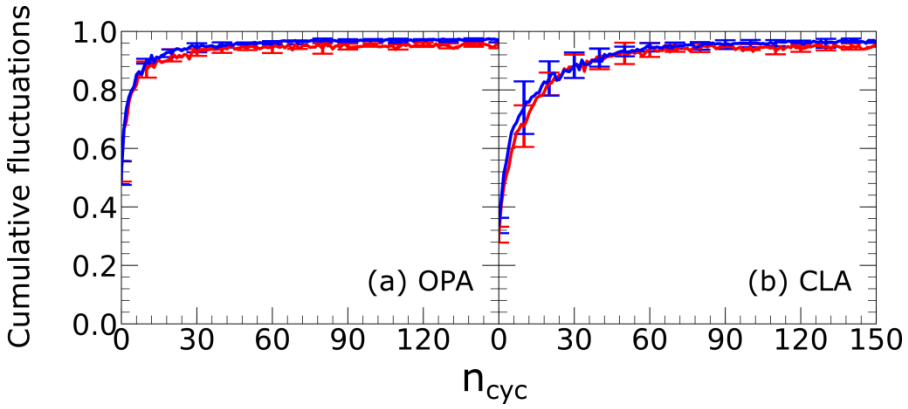

Figure 4.2: Evolution of cumulative fluctuations covered by the first four PCs as a function of n_{cyc} obtained by eePaCS-MD$_{cPCA}$ (blue) and eePaCS-MD$_{dPCA}$ (red). Results are shown for simulations starting from the open (a) and closed (b) states of ADK. The error bars represent the standard deviations.

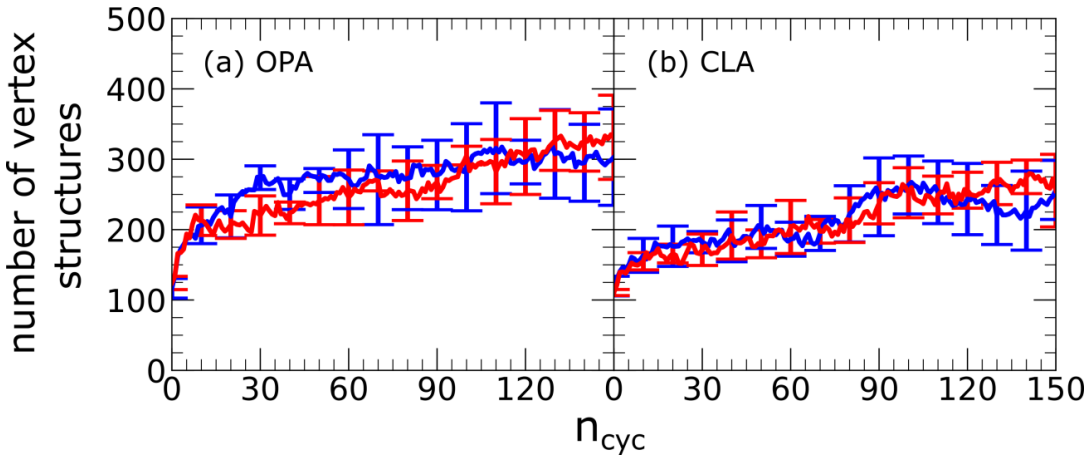

Figure 4.3: The number of vertex structures as a function of n_{cyc} obtained by eePaCS-MD$_{cPCA}$ (blue) and eePaCS-MD$_{dPCA}$ (red). Results are shown for simulations starting from the open (a) and closed (b) states of ADK. The error bars represent the standard deviations.

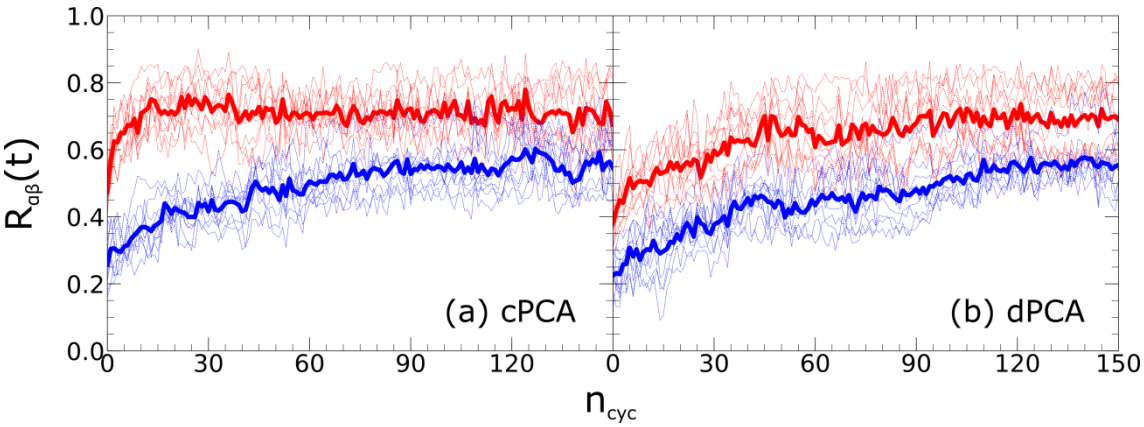

Figure 4.4: Similarity of the PC subspace spanned by the first four PCs between a pair of distinct eePaCS-MD trials at cycle *t* utilizing (a) cPCA and (b) dPCA. The results for ADK from the open (red) and closed (blue) states are shown. The thin lines show the individual results and the thick lines indicate the average.

Table 4.1: Summary of the eePaCS-MD simulations applied to ADK utilizing Cartesian coordinate PCA (cPCA) and distance-based PCA (dPCA). The subscript "cutoff" represents dPCA with Cα atom pairs that are less than 8 Å apart from the initial structures of eePaCS-MDs. The meanings of the simulation indices are described in the main text.

Simulation index	n_{cyc}	n_{rep}	n_{PC}	$RMSD_{min}$ [Å]	$RMSD_{max}$ [Å]	t_{min} [ns]
ADK						
OPA_{cPCA}	150	10	4	2.3 ± 0.6	9.3 ± 0.8	10.3 ± 4.6
OPA_{dPCA}	150	10	4	2.6 ± 0.5	8.0 ± 1.2	11.3 ± 2.3
$OPA_{dPCA(cutoff)}$	150	10	4	3.0 ± 0.5	5.2 ± 0.6	8.0 ± 3.5
CLA_{cPCA}	150	10	4	2.4 ± 0.8	8.2 ± 1.3	10.7 ± 3.4
CLA_{dPCA}	150	10	4	2.0 ± 0.5	9.9 ± 0.6	13.1 ± 1.3
$CLA_{dPCA(cutoff)}$	150	10	4	4.3 ± 1.1	7.1 ± 1.3	12.3 ± 0.9

Table 4.2: Summary of the total computational cost, t_{1st} in ns, with different Cα RMSD criteria (3.5, 3.0, 2.5, 2.0, and 1.5 Å) obtained by eePaCS-MD simulations applied to ADK utilizing Cartesian coordinate PCA (cPCA) and distance-based PCA (dPCA). The subscript "cutoff" represents dPCA with Cα atom pairs that are less than 8 Å apart from the initial structures of eePaCS-MDs. The meaning of the table same as those described in Table 3.2.

Simulation index	3.5 [Å]	3.0 [Å]	2.5 [Å]	2.0 [Å]	1.5 [Å]
ADK					
OPA_{cPCA}	20.4 ± 9.2 (5)	31.0 ± 17.6 (4)	47.3 ± 24.5 (3)	102.0 ± 46.0 (2)	131.0 (1)
OPA_{dPCA}	32.8 ± 28.3 (5)	17.3 ± 7.7 (3)	87.7 ± 52.6 (3)	117.0 (1)	– (0)
$OPA_{dPCA(cutoff)}$	40.3 ± 35.5 (4)	66.3 ± 29.2 (3)	68 (1)	– (0)	– (0)
CLA_{cPCA}	73.8 ± 27.5 (4)	81.5 ± 28.7 (4)	77.7 ± 26.0 (3)	92.3 ± 38.0 (3)	– (0)
CLA_{dPCA}	81.4 ± 25.2 (5)	91.6 ± 26.3 (5)	107.3 ± 21.2 (4)	109.7 ± 14.8 (3)	120 (1)
$CLA_{dPCA(cutoff)}$	76 (1)	77 (1)	83 (1)	– (0)	– (0)

Table 4.3: Individual eePaCS-MDs utilizing cPCA and dPCA for ADK. The meaning of the table is same as those described in Table 3.3.

Index	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
OPA_{cPCA}	2.4 (55)	2.9 (40)	3.1 (142)	1.8 (148)	1.5 (131)
	8.9 (147)	9.9 (150)	10.4 (148)	8.2 (143)	9.1 (147)
OPA_{dPCA}	1.8 (128)	3.2 (79)	3.2 (112)	2.4 (145)	2.5 (100)
	7.3 (141)	8.1 (150)	8.0 (150)	6.5 (141)	10.1 (141)
$OPA_{dPCA(cutoff)}$	2.9 (69)	3.4 (20)	2.9 (107)	3.6 (121)	2.2 (84)
	5.1 (147)	4.7 (108)	4.6 (116)	6.1 (112)	5.6 (124)
CLA_{cPCA}	1.8 (63)	3.7 (105)	1.9 (146)	1.7 (77)	2.7 (145)
	9.1 (150)	6.6 (137)	7.2 (149)	10.3 (142)	8.0 (147)
CLA_{dPCA}	2.0 (123)	2.3 (145)	2.6 (149)	1.7 (118)	1.3 (120)
	10.4 (140)	9.1 (150)	10.6 (148)	9.3 (139)	9.9 (147)
$CLA_{dPCA(cutoff)}$	5.0 (111)	4.0 (116)	5.3 (124)	2.2 (131)	4.8 (134)
	8.7 (141)	5.9 (150)	7.0 (144)	8.4 (150)	5.6 (149)

Chapter 5
Conclusions and Perspectives

Proteins are inherently dynamical molecules that undergo large-scale conformational changes to exert its functions.[8–10] To investigate the high anisotropic nature of protein dynamics, MD simulation is an essential computational tool that can elucidate the conformational transitions of proteins, providing time-dependent information on protein fluctuation at atomic resolution. However, observing conformational changes relevant to biological functions is challenging because these events tend to occur stochastically in a time scale longer than feasible MD simulation time. To overcome this difficulty, various enhanced methods have been proposed.[13–17,77,80] However, some of the methods require an external force to enhance the conformational transition, which does not necessarily guarantee that the obtained trajectories follow the lowest energy pathway. Other methods do not need such external forces but may require pre-test of simulations to determine the simulation parameters which can be cumbersome. Therefore, an enhanced sampling method that can simulate protein conformations relevant to biological functions without external forces and does not require cumbersome parameter setting is attractive. In addition, a method that can simulate protein conformations starting from a single structure without the prior knowledge of other conformational states can be valuable, for example situations where a novel protein structure is solved and its conformational transitions are unknown.

In this thesis, I have proposed edge expansion parallel cascade molecular dynamics (eePaCS-MD)[19] as an efficient adaptive conformational sampling method to investigate the large amplitude motions of proteins, which is an extension of the original PaCS-MD method.[104] eePaCS-MD can simulate open-close transitions along several collective degrees of freedom without the prior knowledge of the conformational transitions or external forces to enhance the conformational sampling. eePaCS-MD can help generate new mechanistic hypotheses and support experimental work to further validate the hypotheses. For example, one can remove a bound ligand from an experimentally determined protein structure and then simulate the bound and unbound systems to see how ligand binding affects protein dynamics and its functions.[23,24] Similarly, one can mutate one or more amino acid residues in the protein to explain or predict the effect of mutations.[26,27] Simply simulating a protein in their apo state to reveal possible protein conformations can be also valuable as it can lead to the discovery of novel binding sites and development of new therapeutic drugs.[25]

In Chapter 3, I have introduced the general concept and methodology of eePaCS-MD. In eePaCS-MD, sampling is repeated from *randomly* selected initial structures that are rigorously located at the boundary of the conformational spaces identified as vertices of a convex hull spanned by several principal components (PCs). This resampling increases the probability of rare event occurrences, inducing conformational transitions to new conformational states, thus enhancing sampling efficiency. In addition, each sampling is assigned with new initial velocities which help overcome the energy barriers. The information of the entire conformational space sampled by the simulation is stored as a set of vertex structures and is updated every cycle, which speeds up the selection process and improves the robustness of the method. The random selections of the initial structures alleviate the risk of consecutively selecting dead-end structures.

The conformational sampling efficiency of eePaCS-MD was demonstrated for the open-close transitions of glutamine binding protein, maltose/maltodextrin binding protein, and adenylate kinase. Each was successfully simulated in ~10 ns of simulation time on average which is expected to offer 1-3 orders of magnitude shorter simulation time than conventional MD. The free energy landscape of the conformational transitions can be obtained by constructing a Markov state model (MSM) using trajectories generated by eePaCS-MD, as demonstrated for the open-to-closed transition of glutamine binding protein. The obtained free energy landscape showed an energy barrier separating the open and closed states where the open state was suggested to be energetically more favorable than the closed state.

The simplicity and generality of eePaCS-MD is particularly appealing. This method can be implemented with any MD program by simple scripts and is available for any computational resource, such as supercomputers and cloud computing. As a demonstration, I have combined eePaCS-MD with accelerated MD and showed that the conformational sampling efficiency can be further enhanced, where the total computational cost of observing the open-close transitions was reduced at most 36% compared to the original eePaCS-MD method.

In Chapter 4, the conformational sampling efficiency of eePaCS-MD utilizing Cartesian coordinate PCA (cPCA) and distance-based PCA (dPCA) were compared for the open-close conformational transitions of adenylate kinase to investigate whether the choice of the PCA method affect the performance of eePaCS-MD. In dPCA, *all* Cα distance pairs and Cα distances of pre-selected atom pairs were considered. In the latter case, atom pairs with Cα distances less than 8 Å apart from the initial structures of eePaCS-MD were selected. In this study, it was suggested that considering *all* Cα distance pairs in dPCA is essential to capture the intrinsic anharmonic nature of proteins during eePaCS-MD simulations to promote the conformational sampling. The sampling

efficiency of eePaCS-MD utilizing cPCA and dPCA (*all* Cα distances) showed comparable results. Considering the computational complexity of dPCA which scales quadratically with the number of atoms, I have concluded that eePaCS-MD utilizing cPCA as a first choice of the PCA method, although the optimal choice is expected to depend on the target.

The current framework of eePaCS-MD, as well as other related methods, cannot directly calculate the free energy because the relationships among the generated trajectories are not obvious. Therefore, additional calculations such as umbrella sampling and MSM are required to obtain the free energy landscape. This two-step process is not necessarily a drawback, as insights into possible conformational changes can be obtained efficiently with a low computational cost. In some cases, the free energy landscape could be calculated by directly analyzing the PaCS-MD-generated trajectories by MSM without any additional simulation.[105,106] However, a concrete workflow and theoretical background in constructing reliable MSM with PaCS-MD-generated trajectories remain to be established.

The selection of initial structures that has the potential to exhibit new metastable states is the key to achieve efficient conformational sampling in eePaCS-MD. To this end, the application of reinforcement learning algorithms, such as Monte Carlo tree search[141] and multi-armed bandits,[142] may be worth considering to improve the process of selecting the initial structures. Such algorithms can be used to avoid initial structures that are trapped in local states and allocate computer resources to other initial structure candidates. In fact, eePaCS-MD proposed in this study which selects initial structures in a complete random fashion can be regarded as an application of ε-greedy with $\varepsilon = 1$ which is one of the most famous and simplest multi-armed bandit algorithms.

Appendix

A. Convex hull algorithm

When given a set of points, a convex hull is defined as the smallest convex which includes all the points. The points that construct the convex hull and its boundary lines are called vertices and edges, respectively. Computing the convex hull is a problem in computational geometry, which is widely applied to computer graphics, pattern recognitions,[113] image processing,[114,115] medical simulations,[116] home range estimations,[117] and animal epidemic forecasts.[118] Various algorithms have been proposed to solve the convex hull problem,[149] such as Graham's scan,[150,151] Jarvis's march,[152,153] and Quickhull.[154–156] Here the two-dimensional Quickhull algorithm is described.

Let us start with a set *S* containing *n* points (Figure A1(a)). First, the points with the minimum and maximum *x* coordinates, *a* and *b*, are determined where these points are one of the vertices of the convex hull (Figure A1(b)). The line formed by points *a*,*b* subdivides *S* into subsets *S1* and *S2*, which will be processed recursively. As an example, let us focus on *S1*.The next step is to find point *c* in *S1* such that the area of the triangle *abc* is maximized (Figure A1(c)). This process is equivalent to finding a point that has the furthest distance from line *ab*. Since the points lying inside the triangle *abc* cannot be part of the convex hull, these points are ignored in the latter steps (Figure A1(d)). To determine whether the point is inside the triangle, one can evaluate the orientation of an ordered triple of points (*p, q, r*), that is, if the points (*p, q, r*) form a clockwise cycle or not. The point orientation can be evaluated by calculating the determinant of a 3×3 matrix given by:

$$\mathrm{Orient}(p,q,r) = \det\begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix} \tag{A.1}$$

where the sign is negative if (*p, q, r*) forms a clockwise cycle, positive if counterclockwise, and zero if they are collinear. Since the determinant gives twice the signed area of the triangle formed by points (*p, q, r*), Equation (A.1) can be used to compute the furthest point from line *pq*. After point *c* is determined, *S1* can be further subdivided into *S11* and *S12* by lines *ac* and *bc*, respectively (Figure A1(d)). The next vertices, which are the furthest points from lines *ac* and *bc*, can be searched following the procedures described above until no more points are left.

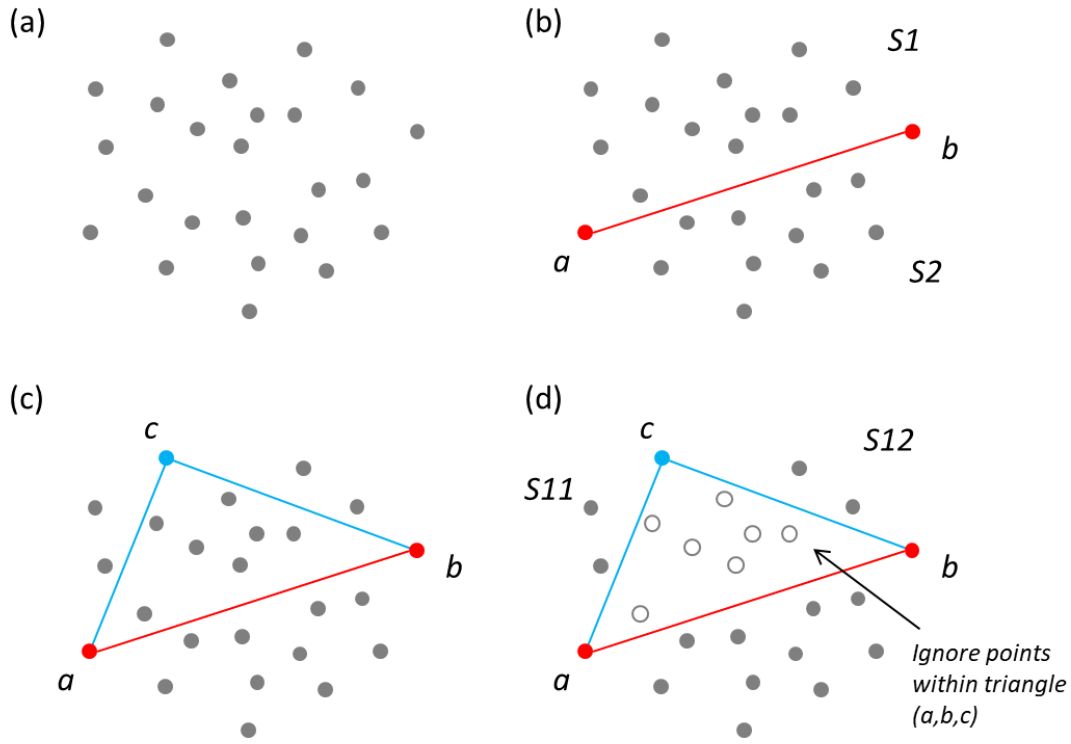

Figure A1: Schematic illustration of Quickhull algorithm. (a) Set *S* containing *n* points are shown as gray circles. (b) The points above and below line *ab* correspond to subsets *S1* and *S2*, respectively. (c) Point c represents the furthest point from line *ab*. (d) The points located outward from lines *ac* and *bc* correspond to subsets *S11* and *S12*, respectively.

Bibliography

[1] A.L. Hopkins and C.R. Groom, Nat. Rev. Drug Discov. **1**, 727 (2002).

[2] S.C. Bull and A.J. Doig, PLoS One **10**, e0117955 (2015).

[3] H. Berman, K. Henrick, and H. Nakamura, Nat. Struct. Mol. Biol. **10**, 980 (2003).

[4] wwPDB consortium, Nucleic Acids Res. **47**, D520 (2019).

[5] C. Kandt and L. Monticelli, in *Membrane Protein Structure Determination. Methods in Molecular Biology (Methods and Protocols)*, edited by J.-J. Lacapère (Humana Press, Totowa, NJ, 2010), pp. 423–440.

[6] G. Ben-Nissan and M. Sharon, Chem. Soc. Rev. **40**, 3627 (2011).

[7] Y. Xu and M. Havenith, J. Chem. Phys. **143**, 170901 (2015).

[8] J. Ma, P.B. Sigler, Z. Xu, and M. Karplus, J. Mol. Biol. **302**, 303 (2000).

[9] B.J. Grant, A.A. Gorfe, and J.A. McCammon, Curr. Opin. Struct. Biol. **20**, 142 (2010).

[10] I. Ernst, M. Haase, S. Ernst, S. Yuan, A. Kuhn, and S. Leptihn, Commun. Biol. **1**, 130 (2018).

[11] A. Kitao and K. Takemura, Curr. Opin. Struct. Biol. **42**, 50 (2017).

[12] K. Henzler-Wildman and D. Kern, Nature **450**, 964 (2007).

[13] J. Schlitter, M. Engels, and P. Krüger, J. Mol. Graph. **12**, 84 (1994).

[14] J. Apostolakis, P. Ferrara, and A. Caflisch, J. Chem. Phys. **110**, 2099 (1999).

[15] U.H.E. Hansmann, Y. Okamoto, and F. Eisenmenger, Chem. Phys. Lett. **259**, 321 (1996).

[16] N. Nakajima, H. Nakamura, and A. Kidera, J. Phys. Chem. B **101**, 817 (1997).

[17] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[18] H. Fukunishi, O. Watanabe, and S. Takada, J. Chem. Phys. **116**, 9058 (2002).

[19] K. Takaba, D.P. Tran, and A. Kitao, J. Chem. Phys. **152**, 225101 (2020).

[20] A. Kitao, F. Hirata, and N. Go, Chem. Phys. **158**, 447 (1991).

[21] A. Kitao and N. Go, Curr. Opin. Struct. Biol. **9**, 164 (1999).

[22] L.S.D. Caves, J.D. Evanseck, and M. Karplus, Protein Sci. **7**, 649 (1998).

[23] R.O. Dror, D.H. Arlow, D.W. Borhani, M. Jensen, S. Piana, and D.E. Shaw, Proc. Natl. Acad. Sci. U. S. A. **106**, 4689 (2009).

[24] N.R. Latorraca, N.M. Fastman, A.J. Venkatakrishnan, W.B. Frommer, R.O. Dror, and L. Feng, Cell **169**, 96 (2017).

[25] A. Stank, D.B. Kokh, J.C. Fuller, and R.C. Wade, Acc. Chem. Res. **49**, 809 (2016).

[26] S. Fukuyoshi, M. Kometani, Y. Watanabe, M. Hiratsuka, N. Yamaotsu, S. Hirono, N. Manabe, O. Takahashi, and A. Oda, PLoS One **11**, e0152946 (2016).

[27] M. Gur, E.A. Blackburn, J. Ning, V. Narayan, K.L. Ball, M.D. Walkinshaw, and B. Erman, J. Chem. Phys. **148**, 145101 (2018).

[28] H. Abdizadeh, A.R. Atilgan, and C. Atilgan, J. Phys. Chem. B **121**, 4778 (2017).

[29] H. Hata, M. Nishiyama, and A. Kitao, Biochim. Biophys. Acta - Gen. Subj. **1864**, 129395 (2020).

[30] M. Karplus and J.A. McCammon, Nat. Struct. Mol. Biol. **9**, 646 (2002).

[31] J.L. Klepeis, K. Lindorff-Larsen, R.O. Dror, and D.E. Shaw, Curr. Opin. Struct. Biol. **19**, 120 (2009).

[32] S.A. Hollingsworth and R.O. Dror, Neuron **99**, 1129 (2018).

[33] Y. Shan, E.T. Kim, M.P. Eastwood, R.O. Dror, M.A. Seeliger, and D.E. Shaw, J. Am. Chem. Soc. **133**, 9181 (2011).

[34] R.O. Dror, A.C. Pan, D.H. Arlow, D.W. Borhani, P. Maragakis, Y. Shan, H. Xu, and D.E. Shaw, Proc. Natl. Acad. Sci. U. S. A. **108**, 13118 (2011).

[35] P.L. Freddolino, F. Liu, M. Gruebele, and K. Schulten, Biophys. J. **94**, L75 (2008).

[36] K. Lindorff-Larsen, S. Piana, R.O. Dror, and D.E. Shaw, Science **334**, 517 (2011).

[37] J.A. McCammon, B.R. Gelin, and M. Karplus, Nature **267**, 585 (1977).

[38] A. Grossfield, M.C. Pitman, S.E. Feller, O. Soubias, and K. Gawrisch, J. Mol. Biol. **381**, 478 (2008).

[39] K. Lindorff-Larsen, P. Maragakis, S. Piana, and D.E. Shaw, J. Phys. Chem. B **120**, 8313 (2016).

[40] I. Yu, T. Mori, T. Ando, R. Harada, J. Jung, Y. Sugita, and M. Feig, eLife **5**, e19274 (2016).

[41] K. Liu and H. Kokubo, J. Chem. Inf. Model. **57**, 2514 (2017).

[42] Y.S. Tan, P. Śledź, S. Lang, C.J. Stubbs, D.R. Spring, C. Abell, and R.B. Best, Angew. Chem. Int. Ed. **51**, 10078 (2012).

[43] A. Bakan, N. Nevins, A.S. Lakdawala, and I. Bahar, J. Chem. Theory Comput. **8**, 2435 (2012).

[44] P. Kollman, Chem. Rev. **93**, 2395 (1993).

[45] H. Fujitani, Y. Tanida, M. Ito, G. Jayachandran, C.D. Snow, M.R. Shirts, E.J. Sorin, and V.S. Pande, J. Chem. Phys. **123**, 084108 (2005).

[46] L. Wang, Y. Wu, Y. Deng, B. Kim, L. Pierce, G. Krilov, D. Lupyan, S. Robinson, M.K. Dahlgren, J. Greenwood, D.L. Romero, C. Masse, J.L. Knight, T. Steinbrecher, T. Beuming, W. Damm, E. Harder, W. Sherman, M. Brewer, R. Wester, M. Murcko, L. Frye, R. Farid, T. Lin, D.L. Mobley, W.L. Jorgensen, B.J. Berne, R.A. Friesner, and R. Abel, J. Am. Chem. Soc. **137**, 2695 (2015).

[47] M. Levitt, M. Hirshberg, R. Sharon, and V. Daggett, Comput. Phys. Commun. **91**, 215 (1995).

[48] A.D. Mackerell, J. Comput. Chem. **25**, 1584 (2004).

[49] L. Monticelli and D.P. Tieleman, in *Biomolecular Simulations. Methods in Molecular Biology (Methods and Protocols)*, edited by L. Monticelli and E. Salonen (Humana Press, Totowa, NJ, 2013), pp. 197–213.

[50] A.P. Lyubartsev and A.L. Rabinovich, Biochim. Biophys. Acta - Biomembr. **1858**, 2483 (2016).

[51] J.A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K.E. Hauser, and C. Simmerling, J. Chem. Theory Comput. **11**, 3696 (2015).

[52] X. Zhu, P.E.M. Lopes, and A.D. Mackerell, Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 167 (2012).

[53] M.J. Robertson, J. Tirado-Rives, and W.L. Jorgensen, J. Chem. Theory Comput. **11**, 3499 (2015).

[54] A. Nikitin, Y. Milchevskiy, and A. Lyubartsev, J. Phys. Chem. B **119**, 14563 (2015).

[55] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, and L.G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).

[56] N.S. Martys and R.D. Mountain, Phys. Rev. E **59**, 3733 (1999).

[57] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten, Mol. Simul. **6**, 121 (1991).

[58] J.-P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[59] S. Miyamoto and P.A. Kollman, J. Comput. Chem. **13**, 952 (1992).

[60] C.W. Hopkins, S. Le Grand, R.C. Walker, and A.E. Roitberg, J. Chem. Theory Comput. **11**, 1864 (2015).

[61] E. Braun, J. Gilmer, H.B. Mayes, D.L. Mobley, J.I. Monroe, S. Prasad, and D.M. Zuckerman, Living J. Comp. Mol. Sci. **1**, 5957 (2019).

[62] L. V. Woodcock, Chem. Phys. Lett. **10**, 257 (1971).

[63] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, and J.R. Haak, J. Chem. Phys. **81**, 3684 (1984).

[64] W.F. van Gunsteren and H.J.C. Berendsen, Mol. Simul. **1**, 173 (1988).

[65] S. Nosé, J. Chem. Phys. **81**, 511 (1984).

[66] W.G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[67] D.J. Evans and B.L. Holian, J. Chem. Phys. **83**, 4069 (1985).

[68] M.R. Shirts, J. Chem. Theory Comput. **9**, 909 (2013).

[69] C. Hofsäß, E. Lindahl, and O. Edholm, Biophys. J. **84**, 2192 (2003).

[70] R. V. Swift and J.A. McCammon, Biochemistry **47**, 4102 (2008).

[71] A.T. Fenley, N.M. Henriksen, H.S. Muddana, and M.K. Gilson, J. Chem. Theory Comput. **10**, 4069 (2014).

[72] S.M.J. Rogge, L. Vanduyfhuys, A. Ghysels, M. Waroquier, T. Verstraelen, G. Maurin, and V. Van Speybroeck, J. Chem. Theory Comput. **11**, 5583 (2015).

[73] M. Parrinello and A. Rahman, J. Appl. Phys. **52**, 7182 (1981).

[74] C. Jarzynski, Phys. Rev. Lett. **78**, 2690 (1997).

[75] S. Park and K. Schulten, J. Chem. Phys. **120**, 5946 (2004).

[76] S. Wolf and G. Stock, J. Chem. Theory Comput. **14**, 6175 (2018).

[77] D. Hamelberg, J. Mongan, and J.A. McCammon, J. Chem. Phys. **120**, 11919 (2004).

[78] L.C.T. Pierce, R. Salomon-Ferrer, C.A.F. de Oliveira, J.A. McCammon, and R.C. Walker, J. Chem. Theory Comput. **8**, 2997 (2012).

[79] Y. Miao, W. Sinko, L. Pierce, D. Bucher, R.C. Walker, and J.A. McCammon, J. Chem. Theory Comput. **10**, 2677 (2014).

[80] A. Laio, A. Rodriguez-Fortea, F.L. Gervasio, M. Ceccarelli, and M. Parrinello, J. Phys. Chem. B **109**, 6714 (2005).

[81] A. Barducci, G. Bussi, and M. Parrinello, Phys. Rev. Lett. **100**, 020603 (2008).

[82] A. Mitsutake, Y. Sugita, and Y. Okamoto, J. Chem. Phys. **118**, 6664 (2003).

[83] A. Kone and D.A. Kofke, J. Chem. Phys. **122**, 206101 (2005).

[84] N. Rathore, M. Chopra, and J.J. de Pablo, J. Chem. Phys. **122**, 024111 (2005).

[85] D.A. Kofke, J. Chem. Phys. **117**, 6911 (2002).

[86] D.J. Earl and M.W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[87] A. Patriksson and D. van der Spoel, Phys. Chem. Chem. Phys. **10**, 2073 (2008).

[88] S. Kumar, D. Bouzida, R.H. Swendsen, P.A. Kollman, and J.M. Rosenberg, J. Comput. Chem. **13**, 1011 (1992).

[89] J.D. Chodera, W.C. Swope, J.W. Pitera, C. Seok, and K.A. Dill, J. Chem. Theory Comput. **3**, 26 (2007).

[90] A. Kuzmanic and B. Zagrovic, Biophys. J. **98**, 861 (2010).

[91] A. Kitao, in *Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems*, edited by Q. Cui and I. Bahar (Chapman & Hall/CRC Press, Boca Raton, FL, 2006), pp. 233-252.

[92] C.C. David and D.J. Jacobs, Methods Mol Biol. **1084**, 193 (2014).

[93] A. Altis, P.H. Nguyen, R. Hegger, and G. Stock, J. Chem. Phys. **126**, 244111 (2007).

[94] F. Sittel, A. Jain, and G. Stock, J. Chem. Phys. **141**, 014111 (2014).

[95] M. Ernst, F. Sittel, and G. Stock, J. Chem. Phys. **143**, 244114 (2015).

[96] J. Shao, S.W. Tanner, N. Thompson, and T.E. Cheatham, J. Chem. Theory Comput. **3**, 2312 (2007).

[97] S.P. Lloyd, IEEE Trans. Inf. Theory **28**, 129 (1982).

[98] D. Arthur and S. Vassilvitskii, in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, 2007), pp. 1027–1035.

[99] V.S. Pande, K. Beauchamp, and G.R. Bowman, Methods **52**, 99 (2010).

[100] J.D. Chodera and F. Noe, Curr. Opin. Struct. Biol. **25**, 135 (2014).

[101] B.E. Husic and V.S. Pande, J. Am. Chem. Soc. **140**, 2386 (2018).

[102] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J.D. Chodera, C. Schütte, and F. Noé, J. Chem. Phys. **134**, 174105 (2011).

[103] S.K. Burley, H.M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. Di Costanzo, C. Christie, K. Dalenberg, J.M. Duarte, S. Dutta, Z. Feng, S. Ghosh, D.S. Goodsell, R.K. Green, V. Guranović, D. Guzenko, B.P. Hudson, T. Kalro, Y. Liang, R. Lowe, H. Namkoong, E. Peisach, I. Periskova, A. Prlić, C. Randle, A. Rose, P. Rose, R. Sala, M. Sekharan, C. Shao, L. Tan, Y.-P. Tao, Y. Valasatava, M. Voigt, J. Westbrook, J. Woo, H. Yang, J. Young, M. Zhuravleva, and C. Zardecki, Nucleic Acids Res. **47**, D464 (2019).

[104] R. Harada and A. Kitao, J. Chem. Phys. **139**, 035103 (2013).

[105] D.P. Tran, K. Takemura, K. Kuwata, and A. Kitao, J. Chem. Theory Comput. **14**, 404 (2018).

[106] D.P. Tran and A. Kitao, J. Phys. Chem. B **123**, 2469 (2019).

[107] R. Harada and A. Kitao, J. Chem. Theory Comput. **11**, 5493 (2015).

[108] R. Harada, T. Nakamura, and Y. Shigeta, J. Comput. Chem. **37**, 724 (2016).

[109] R. Harada and Y. Shigeta, J. Chem. Inf. Model. **57**, 3070 (2017).

[110] R. Harada and Y. Shigeta, J. Chem. Theory Comput. **13**, 1411 (2017).

[111] R. Harada and Y. Shigeta, J. Comput. Chem. **38**, 1921 (2017).

[112] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa, ACM Trans. Math Software **22**, 469 (1996).

[113] R. Meier, F. Ackermann, G. Herrmann, S. Posch, and G. Sagerer, in *IEEE International Conference on Image Processing* (IEEE, 1995), pp. 552–555.

[114] N.M. Sirakov and P.A. Mlsna, in *2nd IEEE International Symposium on Biomedical Imaging: Macro to Nano* (IEEE, 2004), pp. 796–799.

[115] B. Yuan and C.L. Tan, Pattern Recogn. **40**, 456 (2007).

[116] F. Yaacoub, Y. Hamam, A. Abche, and C. Fares, in *32nd Annual Conference on IEEE Industrial Electronics (IECON)* (IEEE, 2006), pp. 3308–3313.

[117] J. Randon-Furling, S.N. Majumdar, and A. Comtet, Phys. Rev. Lett. **103**, 140602 (2009).

[118] E. Dumonteil, S.N. Majumdar, A. Rosso, and A. Zoia, Proc. Natl. Acad. Sci. U. S. A. **110**, 4239 (2013).

[119] Y. Matsunaga, H. Fujisaki, T. Terada, T. Furuta, K. Moritsugu, and A. Kidera, PLoS Comput. Biol. **8**, e1002555 (2012).

[120] S.L. Seyler and O. Beckstein, Mol. Simul. **40**, 855 (2014).

[121] J.C. Spurlino, G.Y. Lu, and F.A. Quiocho, J. Biol. Chem. **266**, 5202 (1991).

[122] A.J. Sharff, L.E. Rodseth, J.C. Spurlino, and F.A. Quiocho, Biochemistry **31**, 10657 (1992).

[123] C.-D. Hsiao, Y.-J. Sun, J. Rose, and B.-C. Wang, J. Mol. Biol. **262**, 225 (1996).

[124] F.A. Quiocho, J.C. Spurlino, and L.E. Rodseth, Structure **5**, 997 (1997).

[125] Y.-J. Sun, J. Rose, B.-C. Wang, and C.-D. Hsiao, J. Mol. Biol. **278**, 219 (1998).

[126] H.H. Loeffler and A. Kitao, Biophys. J. **97**, 2541 (2009).

[127] A. Kitao, J. Chem. Phys. **135**, 045101 (2011).

[128] S. Hayward and A. Kitao, J. Chem. Theory Comput. **11**, 3895 (2015).

[129] C.W. Müller and G.E. Schulz, J. Mol. Biol. **224**, 159 (1992).

[130] C.W. Müller, G.J. Schlauderer, J. Reinstein, and G.E. Schulz, Structure **4**, 147 (1996).

[131] W.L. DeLano, *The PyMOL Molecular Graphics System* (DeLano Scientific, San Carlos, CA 2002).

[132] J.C. Gordon, J.B. Myers, T. Folta, V. Shoja, L.S. Heath, and A. Onufriev, Nucleic Acids Res. **33**, W368 (2005).

[133] J. Myers, G. Grothaus, S. Narayanan, and A. Onufriev, Proteins **63**, 928 (2006).

[134] R. Anandakrishnan, B. Aguilar, and A. V. Onufriev, Nucleic Acids Res. **40**, W537 (2012).

[135] W.L. Jorgensen, J. Chandrasekhar, J.D. Madura, R.W. Impey, and M.L. Klein, J. Chem. Phys. **79**, 926 (1983).

[136] R. Salomon-Ferrer, D.A. Case, and R.C. Walker, Wiley Interdiscip. Rev. Comput. Mol. Sci. **3**, 198

(2013).

[137] R.W. Pastor, B.R. Brooks, and A. Szabo, Mol. Phys. **65**, 1409 (1988).

[138] R.J. Loncharich, B.R. Brooks, and R.W. Pastor, Biopolymers **32**, 523 (1992).

[139] D. Li, M.S. Liu, and B. Ji, Biophys. J. **109**, 647 (2015).

[140] C. Kobayashi, Y. Matsunaga, R. Koike, M. Ota, and Y. Sugita, J. Phys. Chem. B **119**, 14584 (2015).

[141] K. Shin, D.P. Tran, K. Takemura, A. Kitao, K. Terayama, and K. Tsuda, ACS Omega **4**, 13853 (2019).

[142] M.I. Zimmerman and G.R. Bowman, J. Chem. Theory Comput. **11**, 5747 (2015).

[143] D.P. Tran and A. Kitao, J. Chem. Theory Comput. **16**, 2835 (2020).

[144] H. Hata, Y. Nishihara, M. Nishiyama, Y. Sowa, I. Kawagishi, and A. Kitao, Sci. Rep. **10**, 2351 (2020).

[145] M.K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, J. Chem. Theory Comput. **11**, 5525 (2015).

[146] R. Harada, Y. Takano, T. Baba, and Y. Shigeta, Phys. Chem. Chem. Phys. **17**, 6155 (2015).

[147] L. Riccardi, P.H. Nguyen, and G. Stock, J. Phys. Chem. B **113**, 16660 (2009).

[148] R.T. McGibbon, K.A. Beauchamp, M.P. Harrigan, C. Klein, J.M. Swails, C.X. Hernández, C.R. Schwantes, L.-P. Wang, T.J. Lane, and V.S. Pande, Biophys. J. **109**, 1528 (2015).

[159] F.P. Preparata and M.I. Shamos, in *Computational Geometry* (Springer New York, New York, NY, 1985), pp. 89–143.

[150] R.L. Graham, Info. Proc. Lett. **1**, 132 (1972).

[151] A.M. Andrew, Info. Proc. Lett. **9**, 216 (1979).

[152] R.A. Jarvis, Info. Proc. Lett. **2**, 18 (1973).

[153] S.G. Akl, Info. Proc. Lett. **8**, 108 (1979).

[154] W. Eddy, ACM Trans. Math Software **3**, 398 (1977).

[155] A. Bykat, Info. Proc. Lett. **7**, 296 (1978).

[156] E. Mucke, Comput. Sci. Eng. **11**, 54 (2009).