Methods for Dynamical Analysis of Biomolecular Mechanisms in Molecular Dynamics Simulations

Methods for Dynamical Analysis of Biomolecular Mechanisms in Molecular Dynamics Simulations

by

Ka Chun Ho

Under the Direction of Donald Hamelberg, PhD

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2021

ABSTRACT

A deep understanding of the role of motions in the functional mechanisms of biomolecules can potentially speed up molecular dynamics (MD) simulations of the interaction of the same drug target with numerous drug candidates by reusing the simulation of the drug target. This deep understanding can also enable using dynamics as an additional tool to steer a drug toward its intended target and avoid side effects. To gain this understanding, we have modeled the internal motions, the structural hierarchy, and the dynamical hierarchy of biomolecules by analyzing MD simulations. We find that the diffusion coefficients of the internal motions of some biomolecules vary over time and have spectra with significant peaks. We mine and correlate the occurrence of frequent substructures in the major conformations of a protein and avoid the combinatorial explosion of configurations in the analysis. We model the contributions of residues to the coupled Markovian dynamics of multiple domains of a protein. The model discovers residue states that act as indicators of the domain states, and one domain acts as a switch that controls the dynamics of the other domains. Collectively, our results enable a hierarchical decomposition of the motion of biomolecules, extract insights of their functional mechanisms from MD simulations, and provide a foundation for designing architectures of artificial neural networks that extract dynamical and hierarchical insights from MD simulations.

INDEX WORDS: Molecular Dynamics Simulation

Methods for Dynamical Analysis of Biomolecular Mechanisms in Molecular Dynamics Simulations

by

Ka Chun Ho

Committee Chair: Donald Hamelberg

Committee: Binghe Wang

David Wilson

Electronic Version Approved:

Office of Graduate Services

College of Arts and Sciences

Georgia State University

December 2021

DEDICATION

To my parents, sister, and grandparents.

To logic, imagination, and reality.

To people who take tremendous risks to test and develop new ideas.

ACKNOWLEDGEMENTS

I want to thank my advisor, Professor Hamelberg. Our discussions of various methods, applications, and potential improvements for accelerating molecular dynamics simulation are particularly memorable. Those discussions made me realize the perils of sampling tremendously high dimensional spaces, the necessity of clever approximations, and the importance of roughness, diffusive motions, and weak interactions in the functional mechanism of biomolecules. His discussions of the fundamentals of molecular dynamics simulation contribute to my interest in phase space methods, dynamical systems, and the thermostat, which is the main source of stochasticity in the simulation. I also want to thank him for allowing me to have tremendous freedom in research and sending me to conferences, which have considerably broadened my horizons.

I want to thank Professor Wilson for discussing the advantages and limitations of experimental methods for measuring the kinetics and thermodynamics of biological processes. His discussions of calorimetry methods contribute to my interest in thermodynamics, the effect of temperature fluctuation on the diffusive motion of biomolecules, and dynamical systems.

I want to thank Professor Binghe Wang for being an inspiration, his advice on possible careers, and discussions for applying computational methods in drug discovery.

I want to thank Professor Irene Weber and Professor Robert Harrison for teaching the bioinformatics class. The class taught me to use Git to manage the revisions of computer programs and the importance of writing unit tests. These ideas allowed me to learn numpy, tensorflow, and C++ on my own.

I want to thank the former graduate advisor, Rita S. Bennet, who has been very kind and supportive. I want to thank my lab mates for discussions of interesting ideas.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

For a more concrete description of power spectrum CSES and cross-CSES, consider a wide sense stationary property *n* (Figure 2.2.3A). For any time *t*, the instantaneous distribution *N*[*t*] of *n* is a normal distribution with an average of zero and a variance of one. For any two different times t_1 and t_2, instantaneous distributions $N[t_1]$ and $N[t_2]$ are uncorrelated according to Pearson correlation. Consider also two purely second order cyclostationary properties *x* and *y* with measurements $x[t] = n[t]\sin(2\pi\times 2t)$ (Figure 2.2.3B) and $y[t] = n[t]\sin(2\pi\times 3t)$ (Figure 2.2.4C). The Pearson correlations between *x*[*t*] and *n*[*t*] (cyan curve in Figure 2.2.3D) and between *x*[*t*] and *y*[*t*] (magenta curve in Figure 2.2.3D) both approach zero if *x*, *y*, and *n* are measured for a long enough time. Pearson correlation suggests *x*[*t*], *y*[*t*], and *n*[*t*] are not correlated, even though they are related and share the same random factor *n*[*t*]. This is because Pearson correlation as a single number cannot

1 INTRODUCTION

1.1 Overview

Computational simulation has dramatically enhanced the understanding and exploitation of chemistry in biological systems. A drug is a chemical that treats disease by modifying the chemistry in life. Making a drug that alters the function of a target in a biological system is a laborious and time-consuming process that involves synthesizing many drug candidates and testing their toxicities and abilities to treat the disease. Simulating these tests in computers has significantly increased the efficiency, enhances the reliability, and reduces the cost of the drug development process.[1]

One of the problems with drugs is that they can have side effects. A drug has side effects when it interferes with the chemistry of unintended parts of the biological system.[2] For example, enzymes catalyze various reactions in a biological system. We often organize these enzymes into families. Many enzymes in the same family catalyze similar reactions and have similar shapes. While the enzymes in a family catalyze similar reactions, the biological role of these reactions can be dramatically different.

Consequently, a deeper understanding of the steps that enzymes and drugs take to achieve their function can reduce the side effects of drugs. These steps are the functional mechanism of the enzyme or the mechanism of action of the drugs. Since the functional mechanism consists of multiple steps, the mechanism is inherently dynamic instead of static. Indeed, biological molecules are not static structures and achieve their biological function through the motions of the parts of the molecules. Molecular dynamics (MD) simulation is a technique that simulates the motions of molecules with computers, which can produce a trajectory of shapes that a molecule can take over time. This technique has contributed to drug development by enabling calculation of the energy required for a drug candidate to bind to the drug target, refining the binding pose of the drug to the drug target, estimating the speed that a drug binds to the drug target, and simulating the steps in the delivery of a drug to its intended location in the human body.[3]

In the description of the functional mechanism, we often divide a biological molecule into building blocks of various sizes, including domains, secondary structures, sites, and residues. However, we don't have a corresponding understanding of the motions of biological molecules. For example, we cannot effectively model how the change of the motions of one part of a molecule would impact the motions of the whole. Specifically, we cannot predict how a modification of a drug would impact the motion of the drug target without simulating the motions of each pair of drug and drug target separately. Although all these simulations contain the same drug target, we cannot factor out the common dynamics of the drug target and only simulate the interactions between the drug target and a tremendous number of drugs. This factoring can significantly increase the quality and speed of discovering drugs with computers. A deeper understanding of the role of motions in the functional mechanism would also provide an additional means to avoid side effects by steering a drug toward its intended target.

To enable such a deeper understanding of the role of motions in the functional mechanism of biological molecules, I have worked on several projects. The first project explores the nature of the diffusive motions of peptides in water. The second project extracts the hierarchical relationships between local shapes in a protein. The third project models the relationship between the residues of a protein and the coupled dynamics of the protein domains.

1.2 Functional mechanism of Biomolecules

A biomolecule is a molecule with a biological function. The major categories of biomolecules include proteins, lipids, carbohydrates, and nucleic acids.[4] The functional mechanism of a system of biomolecules is the mechanism of a reaction in the system that achieves the biological function of the system.[5] Many biomolecules are macromolecules, molecules containing many atoms. Consequently, the description of the functional mechanisms of biomolecules often involves multiple layers of abstractions. For example, we can divide a protein into various domains. Each domain can contain multiple sites that are important in some steps of the functional mechanism, and each site consists of multiple amino acid residues.[5] Discovering the functional mechanisms of biomolecules is of central importance in the study of biomolecules. For example, a detailed understanding of the mechanism enables monitoring and modifying the function of biological systems.[6]

1.3 Molecular dynamics simulation

1.3.1 Overview

Molecular dynamics (MD) simulation is a method to use computers to simulate the trajectory of motions of a system of molecules.[7] Researchers have routinely used MD simulations to decipher the mechanism of biomolecules because of the unique advantages of MD simulations.[7-8] Firstly, biomolecules' mechanisms often contain multiple steps and are therefore inherently dynamic instead of static.

While MD simulation can simulate a trajectory of motions of every atom over time, many experimental techniques only provide a static description of a system of molecules. For example, crystallography, cryo-electronic microscopy, and neutron diffraction only provide static 3D structures. Secondly, MD simulation has significantly higher spatial and temporal resolutions than experimental techniques. The functional mechanism often involves transformations at various levels, including subunits, domains, sites, and residues. Without prior knowledge of the mechanism, all atoms of the macromolecules are potentially important. Therefore, high spatial and temporal resolutions are valuable in figuring out the mechanisms of biomolecules. However, such resolutions are not easily achievable by experimental techniques. For example, nuclear magnetic resonance (NMR) can observe the average distance between labeled atoms of a protein. Labeling all atoms of a protein is expensive[9] and NMR can observe several major conformations only if the transition between the several major conformations is relatively slow.[10] That means some of the faster steps in the mechanism are not directly observable by NMR. Therefore, MD simulation's ability to provide a highly detailed description of the motions of all atoms is valuable in understanding the mechanism of biomolecules.

1.3.2 Limitations of MD Simulation

Of course, each technique has its own limitation, and MD simulation is no exception. Firstly, MD simulation's accuracy is defined by the accuracy of the forcefield, which has steadily improved over time. While the accuracy of the protein forcefields has enabled many excellent predictions through MD simulation, the forcefield still doesn't exactly match reality.[11] Secondly, the amount of time simulated in typical MD simulation is still far shorter than the timescale of biological processes. This is called the sampling problem. While milliseconds MD simulations have been performed[12], this is still a far cry from the timescale of some biological phenomena, which can happen over seconds or minutes. Enhanced sampling techniques partially solve the sampling problem. These techniques can focus the sampling effort on the desirable dynamics or increase the timescales simulated by MD simulation. One of the consequents of the sampling problem is that the computational power required for MD simulation of the folding of a typical protein is infeasible. Therefore, performing an MD simulation of a system requires an initial structure from crystallography, NMR, cryo-EM, homology modeling, or other means.

1.3.3 Steps for Performing an MD simulation

The steps for performing an MD simulation typically consist of the following. Firstly, we prepare the starting structure of the system by obtaining an initial structure of the biomolecule and inserting solvent molecules and ions, such as water molecules, sodium ions, or potassium ions. Subsequently, we choose a forcefield and minimize the energy of the system. Next, we perform the actual MD simulation by selecting the thermostat and barostat and integrating the force, as defined by the forcefield, thermostat, and barostat, to obtain a trajectory of motion of the system. The integration follows the Newtonian equation of motion discretized over time.

1.3.4 Initial structure

For now, the simulation is not powerful enough to simulate the complete folding of a typical protein unless the protein is tiny. Therefore, MD simulations rely on x-ray crystallography, NMR, cryo-EM, or homology modeling to obtain an initial conformation of the protein.

1.3.5 Forcefield

Common forcefields for proteins include Amber, Charm, GROMOS. These forcefields generally split the potential energy into terms based on different kinds of interactions between the atoms. For example, the potential energy in the Amber forcefield has the following terms:

$$V(\vec{x}) = V_{bonds} + V_{angles} + V_{torsions} + V_{vdW} + V_{coulomb}$$

where the energy between covalent bonds is a sum of harmonic potential in terms of the deviation of the bond length from the equilibrium bond length:

$$V_{bonds} = \sum_{i \in bonds} k_i (r_i - r_i^0)^2$$

The energy in the bond angles is sum harmonic potential in terms of the deviation of the bond angle from the equilibrium bond angle:

$$V_{angles} = \sum_{i \in angles} c_i(\theta_i - \theta_i^0)^2$$

The energy in the torsion angle is a sum of Fourier series, which are each a sum of cosines that models the peaks and valleys in the torsion:

$$V_{torsions} = \sum_{i \in torsions} w_i(\omega_i)$$

$$w_i(\omega_i) = \sum_n \frac{1}{2} A_{i,n}[1 + \cos(n\omega_i + \gamma_i)]$$

The van der Waals interaction term is a sum of terms with 6 and 12 powers. The indexing is over pairs of atoms.

$$V_{vdW} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{A_{ij}}{r_{ij}^6} + \frac{B_{ij}}{r_{ij}^{12}}$$

The columbic interactions give the following energy:

$$V_{coulomb} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}$$

The function that defines the potential energy of every possible conformation of a class of molecules is a forcefield for the class of molecules. Since all we must do is to simulate these relatively simple differential equations (instead of doing a full quantum mechanical calculation and solving the computationally costly slater determinants), the method is computationally far cheaper than quantum mechanical methods. The columbic potential is generally evaluated with the particle mesh Ewald method.[13]

1.3.6 Evaluating Non-bonded Interactions with Particle Mesh Ewald Method

The total coulombic potential energy in a set of atoms is precisely the sum of the potential energy in the coulombic interaction between each of all pairs of atoms. Since each atom can form a pair with each of the remaining atoms, calculating the total coulombic potential energy is computationally time-consuming. While the number of pairs of atoms close to each other is relatively small, the number of pairs of atoms far apart is much larger. One approximation that avoids the costly summation is to sum only over the pairs of atoms close to each other, which is relatively small. Specifically, if the distance between two atoms is larger than the cutoff, we exclude the pair from the sum.

However, this approximation through summation with a cutoff is inaccurate for the coulombic potential energy. The magnitude of the non-bound interactions decreases slowly as the distance between the atoms increases. Therefore, the non-bonded interaction between atoms relatively far apart is still strong. Consequently, the approximation that excludes sum over pairs of atoms far apart is inaccurate.

According to the following equation, the particle mesh Ewald summation approximates the total non-bonded potential energy as a sum of two potential energies.

$$V_{columb} = \underbrace{\sum_{\vec{l}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{G_{ij}}{\left|\vec{r}_{ij} + \vec{l}\right|^2} \cdot \phi(r_{ij})}_{A} + \underbrace{\sum_{\vec{l}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{G_{ij}}{\left|\vec{r}_{ij} + \vec{l}\right|^2} \cdot \left[1 - \phi(r_{ij})\right]}_{B}$$

(Equation 1.1)

where $\vec{l}$ is a non-zero lattice vector, i and j indices the atoms, $\vec{r}_{ij}$ is a vector from the position of the i^{th} atom to the j^{th} atom, r_{ij} is the distance between the i^{th} and j^{th} atoms, G_{ij} is a constant contains the charges of the i^{th} and j^{th} atoms and the other constants in the coulomb law, and $\phi(r_{ij})$ is the convergence function that specifies the two summations are over which pairs of atoms.

These two potential energies are A. the non-bonded potential energy in the pairs of atoms that are not far apart and B. the non-bonded potential energy in the remaining pairs of atoms. The potential energy A is exactly the summation over all pairs of atoms close to each other. Particle mesh Ewald summation approximates potential energy B by Fourier transforming each term and summing the transformed terms over the frequency domain. The convergence function $\phi(r_{ij})$ is chosen such that A is tractable, B only requires summing over a low number of frequencies, and the overall approximation is accurate enough.[13] The periodicity of the interactions between one atom and all images of another atom in all periodic cells make the evaluation of this set of interactions a convolution. This convolution is efficiently computed by the convolution theorem and fast Fourier transform.[14]

1.3.7 Minimization

After adding solvent and ions to the system, we would minimize the energy of the system. This minimization step is often performed by the steepest descent. In the $(n + 1)^{\text{th}}$ iteration, the steepest descent algorithm updates the position of the atoms by the following equation:

$$\vec{r}_{n+1} = \vec{r}_n + \frac{\vec{F}_n}{max(|\vec{F}_n|)} \cdot h_n$$

(Equation 1.2)

where $\vec{r}_n$ is the coordinate of all atoms at the n^{th} iteration, $\vec{F}_n$ is the force on the atoms as defined by the forcefield in (Equation 1.6), $\max(|\vec{F}_n|)$ is the maximal component of $\vec{F}_n$, h_n is the maximal displacement.

1.3.8 Initial velocity

After preparing and minimizing the system, we would set the initial velocities of the atoms using the maxwell-Boltzman distribution at the beginning of the MD simulation. The probability density of the maxwell-Boltzmann distribution is as follows:

$$p(v_i) = \sqrt{\frac{m_i}{2\pi k_B T}} \exp\left(-\frac{m_i v_i^2}{2k_B T}\right)$$

(Equation 1.3)

where v_i is the i^{th} component of the velocities of all atoms, m_i is the mass of the atom, k_B is the Boltzmann constant, and T is the target temperature of the simulation. This distribution of velocities ensures that we are simulating a system at the target temperature.

1.3.9 Newtonian Equation of Motion

The simplicity of the physical model in MD simulation is one reason for the computational efficiency of the simulation. The simulation is based entirely on Newtonian physics. Every atom has an acceleration, velocity, and position. The relationship between these three followings the usual Newtonian physics, as described by the Newtonian equation of motion for objects with constant mass:

$$\frac{d\vec{x}(t)}{dt} = \vec{v}(t)$$

$$\frac{d\vec{v}(t)}{dt} = \vec{a}(t)$$

(Equation 1.4)

If the temperature of the simulated system is at absolute zero (0K), the force acting on each atom depends only on the current position of the atoms. Given the initial positions and velocities of all atoms and a function that defines the potential energy for all possible conformation, the differential equations describe the future motions of all atoms.

1.3.10 From forcefield to force

$$\vec{F}(\vec{x}(t)) = m\vec{a}(t)$$

(Equation 1.5)

The definition of potential energy defines the relationship between the potential energy function ($V(\vec{x})$), which is the forcefield, and the force:

$$-\nabla V(\vec{x}) = \vec{F}(\vec{x})$$

(Equation 1.6)

In other words, the negative of the gradient with respect to the coordinates of the atoms is the force acting on the atom. Specifically, the negative of the partial derivative of the total potential energy with respect to displacement of each atom along each axis is the force acting along the axis on the atom.

1.3.11 Integration over timesteps

Since computers don't have infinite precision, we can't integrate the equations to infinite time resolution. Therefore, it is necessary to discretize time to limit the time resolution. In the integration of differential equations, which include typical MD simulations, we split the time into multiple discrete points, called timesteps. There are various schemes to convert the equations over continuous-time to discrete-time. We strive to maintain energy, numerical stability, and time reversibility. Here are the methods to integrate the equation of motion. The numerical stability is advantageous because this would allow us to use bigger timesteps, which means we can use the same amount of real time to simulate a longer simulation time. Otherwise, it would blow up. Conservation of energy is crucial because it would avoid the accumulation of numerical errors, and the algorithm would have a minimal number of evaluations of the force. Due to the complexity of the forcefield, calculating the force is the most time-consuming step in the integration. One of the schemes with these desirable properties is the velocity Verlet algorithm.

The velocity Verlet algorithm updates the position and velocities of the atoms according to the following equation[15]:

$$\begin{cases} \vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\Delta t^2 \\ \vec{v}(t+\Delta t) = \vec{v}(t) + \frac{\vec{a}(t)+\vec{a}(t+\Delta t)}{2}\Delta t \end{cases}$$

(Equation 1.7)

where t is the current time, Δt is the timestep of the simulation, $\vec{x}$ is the positions of all atoms, $\vec{v}$ is the velocities of all atoms, and $\vec{a}$ is the acceleration of all atoms.

1.3.12 Adding a thermostat to the Newtonian Dynamics

We assume the simulated part of reality can exchange heat but cannot exchange atoms with the rest of the world. The rest of the world acts as a heat bath that provides heat to the part of reality that we simulate. We have various schemes to simulate this exchange of heat, called thermostats. For example, the Langevin thermostat adds a random force to all atoms. Before we add a thermostat to the simulation, the MD simulation is a numerical integration of the following differential equation.

$$ma = -\nabla V(x)$$

After adding a Langevin thermostat, we have the following stochastic differential equation that describes all possible paths that the system can go through.

$$ma = -\nabla V(x) - \gamma m v + \sqrt{2M\gamma k_B T}R(t)$$

Here, the $R(t)$ at any given point of time t has a Gaussian distribution. $R(t_1)$ and $R(t_2)$ are uncorrelated if $t_1 \neq t_2$. The dampening term and the strength of the gaussian process ensure the equations reproduce Boltzmann distribution when the system is at equilibrium.

1.3.13 Barostat

The goal of the barostat is to maintain the pressure of the simulation and ensure the simulation correctly reproduces thermodynamic ensembles.

In the isotropic version of the Berendsen barostat,[16] the pressure of the simulation approaches the target pressure by 1st order kinetics:

$$\frac{dP}{dt} = \frac{P_{ref} - P}{\tau_p}$$

where P_{ref} is the target pressure, P is the pressure within the simulation, and τ_p is the pressure coupling time constant. For every n_{PC} timesteps, the algorithm scales the positions of all atoms by:

$$\mu = 1 - \frac{n_{PC}\Delta t}{3\tau_p}\beta\left(P_{ref} - P\right)$$

where β is an estimation of the isothermal compressibility of the system. Notice that the Berendsen barostat does not yield the exact NPT ensemble.[17]

The Parinello-Rahman barostat allows the simulation box to scale and deform.[18] This barostat theoretically would give the correct NPT ensemble. The box vectors have the following equation of motion:

$$\frac{d^2\boldsymbol{B}}{dt^2} = V\boldsymbol{W}^{-\mathbf{1}}\boldsymbol{B}'^{-1}\left(\boldsymbol{P} - \boldsymbol{P}_{ref}\right)$$

where $\boldsymbol{B}$ is a 3 x 3 matrix with the box vectors as the columns, V is the volume of the box, $\boldsymbol{W}$ is a matrix parameter that determines the strength of the coupling, $\boldsymbol{P}$ is the current stress tensor that represents pressure at the boundary of the box, and $\boldsymbol{P}_{ref}$ represents the stress tensor for the external pressure over the box.

The equations of motion for each atom are:

$$\begin{cases} \dfrac{d^2\vec{r}_i}{dt^2} = \dfrac{\vec{F}_i}{m_i} - \boldsymbol{M}\dfrac{d\vec{r}_i}{dt} \\ \boldsymbol{M} = \boldsymbol{B}^{-1}\left[\boldsymbol{B}\dfrac{d\boldsymbol{B}'}{dt} + \dfrac{d\boldsymbol{B}}{dt}\boldsymbol{B}'\right]\boldsymbol{B}'^{-1} \end{cases}$$

where $\vec{r}_i$ is the coordinate for the i^{th} atom and $\vec{F}_i$ is the force acting on the i^{th} atom.

1.4 Markov State Model

Since MD simulations produce trajectories with numerous frames, methods for analyzing the trajectories are necessary for extracting biological insights from the simulation. Markov state model is a method for analyzing MD simulations.

Markov state model (MSM) assumes the system follows the Markovian assumption and the system is always in one of the N possible states. The Markovian assumption is valid if the system's distribution of possible future states depends only on the system's current state. In other words, the history of past states of the system doesn't influence the future state of the system.

The model describes the dynamics of the system through state and transition. At any point in time, the system has a probability of being in each of the possible states. Given the system's current state, the model describes the conditional probability to transit to each possible future state. The conditional probability of transition from each possible current state to each possible future state is constant over time. The time in the model can be either discrete or continuous. Since the MSM used for analyzing MD simulation are often discrete over time, I will describe the MSM for discrete-time here.

The $N \times N$ transition matrix T consists of all transition probabilities, the conditional probabilities of transitions between the states.

$$T_{ij} = p(S[n+1] = j | S[n] = i)$$

where T_{ij} is the element at the i^{th} row and j^{th} column of matrix T and $S[n]$ is the state at the n^{th} timestep of the MSM. Similarly, the probabilities for the system to be in each of the states are often organized into a vector $1 \times N$ state probability vector $\vec{x}$[t] with the following elements[19]:

$$x[t]_i = p(S[t] = i)$$

The following equation describes the evolution of the states of the MSM over time. [19]

$$x[t + \Delta t] = x[t]T$$

where Δt is the lag time of the MSM.

The Markov state model has a stationary probability, representing the system's probabilities of being in each state once the system reaches equilibrium. Once the system reaches equilibrium, the system will stay at equilibrium, and the state probability vector does not change. That means the stationary probabilities remain the same when the transition matrix transforms them. Therefore, the stationary probabilities are the elements of the left eigenvector $\vec{\eta}$ of the transition matrix with an eigenvalue of one. Hence, we have the following.[19]

$$\vec{\eta}T = \vec{\eta} \cdot \vec{1}$$

Besides the Markovian assumption of MSM, MSM for analyzing equilibrium MD simulation often has two additional assumptions. The model assumes that the system is ergodic, meaning any state can transit through any number of intermediate states to reach other states. The model also assumes detailed balance, which means the transition between the states is reversible, as specified by the following equation.[19]

$$\eta_i T_{ij} = \eta_j T_{ji}$$

2 OSCILLATORY DIFFUSION AND SECOND ORDER CYCLOSTATIONARITY IN ALANINE TRIPEPTIDE FROM MOLECULAR DYNAMICS SIMULATION

This chapter is adapted with permission from *J. Chem. Theory Comput.* 2016, 12, 1, 372–382.
 https://pubs.acs.org/doi/10.1021/acs.jctc.5b00876

2.1 Introduction

Molecular dynamics (MD) simulation distinctly offers a description of motions of biomolecules at atomic resolution. This description could potentially explain allosteric mechanism in subcellular processes that is not offered by crystallography, functional assays, nuclear magnetic resonance, or other methods at comparable resolution. However, this advantage is realized only if the mechanism is extracted as humanly understandable insight, assuming the force field is accurate enough. Without statistical methods, it is almost impossible to analyze MD simulations efficiently, completely, or objectively because MD simulations generate a large volume of data. Moreover, we should seek to extract as much insight as possible from the simulations, since MD simulations also take a considerable amount of time to carry out. These considerations show importance of robust statistical methods in the analysis of MD simulations.

There are several methods that can be used to extract mechanisms from MD simulations. Some methods extract the overall mechanism without analyzing detailed causes of the mechanism. For example, a Markov state model of MD simulations attempts to extract transition probabilities between conformations.[20] This overall description alone would not trace the flow of information through different parts of the biomolecule. Understanding how information flows through different parts of a biomolecule can potentially help to design drugs that can modify this flow. Extracting this flow of information would require a multiscale description from atomic interactions to the overall mechanism.

For bottom-up methods, the first step is usually to construct correlation maps, which reflect correlations between all pairs of atoms or residues. The measure of correlation is often dynamic cross correlation[21] or a common estimation of mutual information[22] of certain properties. Principal component analysis and network analysis of the correlation map can subsequently detect atoms or residues that dynamically function as a group and the interactions between these groups.[23] Alternatively, quasi-normal-mode analysis can be used on the mass-weighted correlation map to estimate the frequency of motions and configurational entropy.[24-25] Measures of information transfer can also be used.[22] For example, transfer entropy estimates the strength and direction of information transfer as conditional mutual information,[22, 26-31] and a description of the mechanism can similarly be built bottom-up.

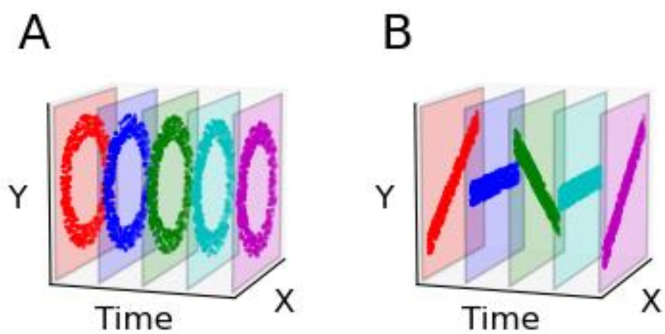

Figure 2.1.1 Scatter plot of joint distributions of properties X and Y at various time. (A) X and Y have a wide sense stationary and nonlinear relationships. (B) X and Y have nonstationary and linear relationships. Two properties are jointly nonstationary if their relationship, as reflected by their joint probability distribution, changes over time.

The foundations of these bottom-up methods are the measures of correlation. An ideal measure of correlation should characterize the total strength of all forms of relationships between properties. This strength of the relationship between two properties is reflected by the ability to predict one property from the other property. Correctly accounting for nonlinear and nonstationary relationships are two recognizable challenges in designing an ideal measure (Figure 2.1.1).[32] Compared to this ideal measure, the measures of correlation commonly used for analysis of MD simulations can be inadequate because they assume wide sense stationarity. When the relationship between two properties changes over time, mutual information, Pearson correlation, and dynamic cross correlation can only reflect the time-averaged relationship.

Here is an example that illustrates the effect of nonstationarity on Pearson correlation, commonly used in linear regression (Figure 2.5.1). Consider two students A and B who stochastically visit a library everyday associated with property S. If student A visits the library on a day t, then $S_A(t) = 2$; otherwise $S_A(t) = -2$. For student B, $S_B(t)$ is similarly defined. On Mondays, students A and B either both visit the library or both do not visit the library. This means that the covariance between $S_A(t)$ and $S_B(t)$ on Mondays is always 4. On Wednesdays, either A or B visits the library but not both. This means that the covariance between $S_A(t)$ and $S_B(t)$ on every Wednesday is always -4. For any day that is neither a Monday nor a Wednesday, $S_A(t)$ and $S_B(t)$ are uncorrelated and have a covariance of 0. If we apply covariance to the two trajectories $S_A(t)$ and $S_B(t)$ over a long time, the overall covariance is zero. While the overall covariance is zero, $S_A(t)$ and $S_B(t)$ are clearly related, especially on Mondays and Wednesdays. Covariance failed to detect the relationship between $S_A(t)$ and $S_B(t)$ because they are not jointly wide sense stationary. This means the relationship between $S_A(t)$ and $S_B(t)$ change over time t. In this case, the relationship between $S_A(t)$ and $S_B(t)$ oscillates with a period of one week. Correlations that oscillate over time will be called an oscillatory correlation. This example also shows that Pearson correlation cannot detect oscillatory correlation. This is because Pearson correlation is the normalized covariance.

Dynamic cross-correlation similarly cannot correctly characterize relationships that change or oscillate over time. This is because dynamic cross-correlation on one dimension is Pearson correlation. Pearson correlation is widely known for its inability to correctly account for nonlinear relationships. Mutual information can account for nonlinear relationships in correlation. However, mutual information also cannot correctly account for nonstationary relationships.[33] Mutual information is a form of Kullback-Leiber divergence,[28, 34] which cannot correctly account for nonstationarity.[32, 35]

In general, if a measure of correlation does not change upon reordering the trajectory, it cannot correctly characterize nonstationary relationships (Figure 2.5.2). This means that a correlation analysis over the distribution of all conformations in a trajectory cannot account for nonstationary relationships. This is because rearranging the order of the frames of the trajectory would not change the result of the analysis. Mutual information, dynamic cross-correlation, and distance correlation coefficient[36] do not change over reordering of frames in a trajectory. They all cannot correctly account for nonstationary relationships.

Given that current correlation analysis for MD simulations cannot correctly account for nonstationary relationships, can such relationships exist in internal motion of a molecule at thermal equilibrium with the solvent? NMR experiments show that the side chain and main chain motions of protein at thermal equilibrium with water occur on different time scales.[37] These motions on different time scales are correlated. The faster local atomic fluctuations can facilitate the slower collective motions. Motions with faster time scales have a higher percentage of high frequency motions. For example, in a more viscous liquid, the correlation time is longer,[38] and the motions of molecules have a higher percentage of low frequency components. This means that protein motions of different frequencies are correlated because protein motions of different time scales are correlated. Two-dimensional infrared spectroscopy also shows that motions of different frequencies are correlated in a peptide.[39] Mathematically, correlation between components of distinct frequencies is a nonstationary correlation.[40] This is because stationary relationships do not change over time, have no periodicity, and do not favor specific frequencies. This means that stationary relationships cannot contribute to correlation between motions of different frequencies. A mathematical proof can be found in the Supplementary Information. The above observations have the following implication. First, oscillatory correlation represents a correlation between motions of specific and different time scales. This correlation includes correlation between main chain motions and side chain rotations and correlation between slower collective motion and faster local motions. Secondly, oscillatory correlation can occur at thermal equilibrium because correlation between motions on different timescales exists at thermal equilibrium. Third, current correlation analysis of MD simulations cannot correctly account for correlation between motions on different timescales because these analyses cannot correctly account for nonstationary correlations.

To characterize nonstationary relationships in molecular motion, we can assume the motions of atoms are jointly cyclostationary instead of jointly stationary. Joint second-order cyclostationarity[41] means the relationship between two properties, as reflected by the instantaneous covariance, oscillates over time (Figure 2.5.3). This instantaneous covariance is defined over an ensemble of conformations that varies over time (Figure 2.5.3). This time-varying ensemble is reasonable because existence of normal mode and vibrational spectrum show the ensemble of conformations change over time. An oscillating hook spring is an example of a system with a time-varying ensemble. This is because the collection of possible conformations of an oscillating hook spring changes over time. We can characterize this oscillatory relationship by estimating the spectrum of the instantaneous covariance. Peaks in this spectrum reflect the oscillation of the instantaneous covariance. We can also define an oscillatory instantaneous variance of a property as the oscillatory instantaneous covariance between a property and the same property. This oscillatory instantaneous variance has the following connection with models of diffusion.

A diffusion model can be used in analyzing internal motions of a molecule. For example, Smoluchowski equation and Ornstein-Ulenbeck processes can be used to model the rotation of a peptide bond.[42] Both of these equations describe diffusion that has a wide sense stationary position, velocity, force, and acceleration.[43] Diffusion coefficient,[42] mean squared displacement over time,[44] autocorrelation,[12, 42] and memory function[45] can be used to estimate parameters for this model. Yet, all these characterizations cannot reveal or characterize nonstationarity in motion. This is because these characterizations all use time averaging or ensemble averaging. Time averaging discards time-varying properties like oscillation of instantaneous diffusion coefficient. Ensemble averaging removes oscillation if each group of chemically identical molecules in the ensemble does not favor a particular phase of the oscillation of the parameters. Modelling molecular motion by Smoluchowski equation, Ornstein-Ulenbeck processes, or other models of stationary diffusion cannot account for nonstationarity. A time-varying diffusion coefficient can be used to extend these models for modeling nonstationary motions. An acceleration with oscillatory instantaneous variance means an instantaneous diffusion coefficient that oscillates over time. This means, on average, that the speed of the diffusion oscillates between fast and slow. This form of diffusion will be called an oscillatory diffusion. This chapter will focus on characterizing the oscillatory instantaneous variance, which is a form of second-order cyclostationarity, in the motion of alanine tripeptide.

2.2 Theoretical Considerations

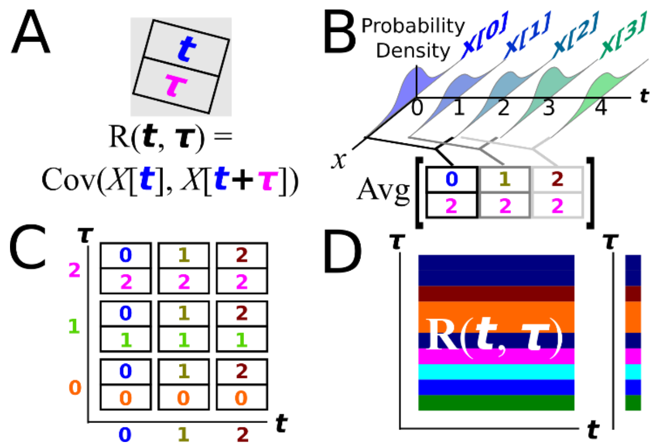

Figure 2.2.1 (A) For each time t and delay τ, the correlation structure $R_X(t, \tau)$ is the covariance between instantaneous probability distributions $X[t]$ and $X[t + \tau]$. $R_X(t, \tau)$ evaluated over time t and delay τ is represented by the box at top. (B) The filled shapes each represents the probability distribution $X[t]$ of property x at time t. These probability distributions are the instantaneous probability distributions. At time t, the average and variance of $X[t]$ are respectively the instantaneous average and instantaneous variance of property x at time t. Unnormalized autocorrelation $C_X(\tau)$ for delay $\tau = 2$ is average of $R_X(t, 2)$ over time t. Each box is linked to $X[t]$ and $X[t + \tau]$ represented by filled shapes. (C) Plot of $R_X(t, \tau)$ over t and τ. (D) Correlation structure of any wide sense stationary property is constant over time t for each τ. The square array of boxes at left represents such a correlation structure. The colors are constant over time t for each delay τ. The single column of boxes at right represents the unnormalized autocorrelation, which is the average over time t of the square array at left.

A property x is wide sense stationary if it satisfies two requirements. Firstly, its instantaneous average $\tilde{\mu}_x[t]$ must be constant over time t (Figure 2.2.1).[46] Secondly, its correlation structure must be constant over time t for each delay τ (Figure 2.2.1).[46] This implies the instantaneous average and correlation structure both have no dependence on time t. This means that Harmonic oscillation is not wide sense stationary, according to this definition of wide sense stationarity. The position of a harmonic oscillator changes over time. This means the instantaneous average is not constant over time. Notice also that the degree of damping of a motion is not directly related to wide sense stationarity. A heavily damped motion can be wide sense stationary or nonstationary. A discussion of damping, wide sense stationarity, and possible forms of nonstationarity in molecule motion can be found in the Supplementary Information.

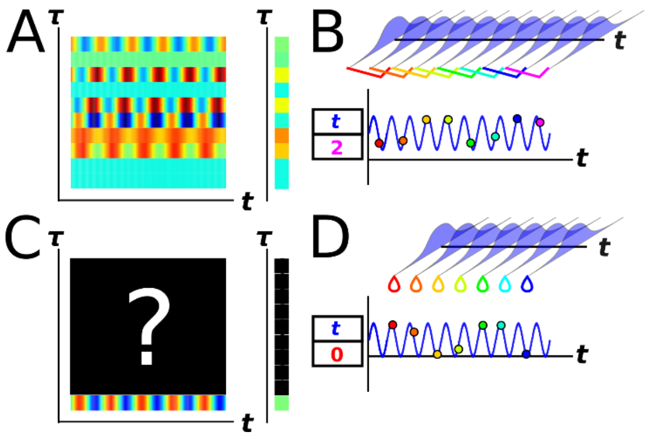

Figure 2.2.2 (A) Correlation structure $R_X(t, \tau)$ of a second order cyclostationary component *x*. The column at left is the autocorrelation. (B) Rainbow colored lines represent covariances between $X[t]$ and $X[t + 2]$. (C) Correlation structure $R_Y(t, \tau)$ of a component y with oscillatory instantaneous variance. While $R_Y(t, 0)$ must be oscillatory over time *t*, the rest of the correlation structure can have any values and is represented by a white question mark. (D). $R_Y(t, 0)$.

A component x of a property is second order cyclostationary if its correlation structure for each delay τ is either periodic or constant over time *t* (Figure 2.2.2A and Figure 2.2.2B).[47-49] This is exemplified by Figure 2.2.2A. Each row of the correlation structure in Figure 2.2.2A corresponds to one time delay τ. Each row is either constant or periodic over time t. A row that is constant over time t has a single color. A row that is oscillatory over time *t* is represented with alternating colors. The component *x* is purely second order cyclostationary if it satisfies two additional requirements.[50] Firstly, it should have a constantly zero instantaneous average over time. Secondly, it should contain no wide sense stationary component. A subset of purely second order cyclostationary components is components with pure oscillatory instantaneous variance. A component *y* with oscillatory instantaneous variance with frequency *f* must have a correlation structure $R_Y(t, \tau)$ that oscillates over time *t* with frequency *f* for delay $\tau = 0$ (Figure 2.2.2C). This is because the part of the correlation structure with zero delays τ is the instantaneous variance (Figure 2.2.2D). For non-zero delays $\tau > 0$, this correlation structure $R_Y(t, \tau)$ can have any values

(Figure 2.2.2C). The component *y* has pure oscillatory instantaneous variance if it contains no wide sense stationary component and no harmonic component.

Instead of investigating all forms of second order cyclostationarity, we investigate pure oscillatory instantaneous variance in internal motions of a biomolecule and do not consider correlations with non-zero delays. A test for pure oscillatory instantaneous variance in acceleration of internal motions would serve two purposes. Firstly, this would reflect the extent of a component of motion that is not accounted for by current correlation analysis of MD simulations. Secondly, this test would show whether oscillatory diffusion can exist in internal motion of biomolecules. Such a test has not been applied to biomolecular MD simulations, to the best of our knowledge. For each property x measured for a total time T, the test we employ consists of subtracting the overall average μ_x from the measurements $x[t]$ and then calculating two spectra. The first spectrum is the power spectrum $S_{xx}[\omega]$ (Equation 2.1),[51] which is used to detect harmonic components. Peaks on this spectrum represent a harmonic oscillation with amplitude A and frequency f as a peak with magnitude $A^2/2$ and frequency f. All purely second order cyclostationary components are invisible on this power spectrum. The second spectrum is a squared envelope spectrum (SES)[47, 52] with the peak at zero frequency removed. Squared envelope spectrum is a power spectrum of the squared deviation of the property from its overall average. The removed peak at zero frequency represents the overall variance σ^2_x of the data, which reflects contributions from all components. For ease of comparison with the first spectrum, the frequency is halved. The magnitude is also doubled and the square root taken to form a corrected squared envelope spectrum (CSES) (Equation 2.2). The CSES represents each harmonic component of a property with amplitude A and frequency f as a peak with magnitude $A^2/2$ and frequency f. Each component with oscillation in the instantaneous variance of the property with amplitude $A^2/2$ and frequency $2f$ would also generate a peak with magnitude $A^2/2$ and frequency f. The remaining components of the property contain no harmonic component and no component with oscillatory instantaneous variance. These remaining components would contribute to a constant baseline across all frequencies on this spectrum. The total magnitude at a frequency is the sum of magnitudes of these three contributions. A comparison of these two spectra would show the contribution of components with pure oscillatory instantaneous variance at each frequency.

$$S_{\mathrm{xx}}[\omega] = 2\left|\frac{1}{T}\mathrm{FT}_{x[t]}(\omega)\right|^2 \text{ for } \omega>0 \text{, where}$$

$$\mathrm{FT}_x(\omega) = \sum_{t=0}^{T}(x[t]-\mu_x)\, e^{-2\pi i\omega t} \text{, and}$$

$$\mu_x = \frac{1}{T}\sum_{t=0}^{T} x\,[t]$$

(Equation 2.1)

$$\mathrm{CSES}[\omega] = \sqrt{2\cdot\mathrm{SES}[2\omega]} \text{ for } \omega>0 \text{, where}$$

$$\mathrm{SES}[\omega] = 2\left|\frac{1}{T}\mathrm{FT}_{x^2}(\omega)\right|^2,$$

$$\mathrm{FT}_{x^2}(\omega) = \sum_{t=0}^{T}[(x[t]-\mu_x)^2-\sigma_x^2]e^{-2\pi i\omega t},$$

$$\mu_x = \frac{1}{T}\sum_{t=0}^{T} x\,[t] \text{, and } \sigma_x^2 = \frac{1}{T}\sum_{t=0}^{T}(x[t]-\mu_x)^2$$

(Equation 2.2)

A test for correlation involving second order cyclostationary motions would show if this form of motion could transmit through a molecule. In general, correlation structures involving second order cyclostationary components are oscillatory over time t and may be also oscillatory over delay, τ.[41] We employed a test for the oscillation over time t of the part of the cross correlation structure with the zero delay τ. This oscillation would be called an oscillatory correlation. Consider two properties x and y measured over total time T. The measurements for properties x and y are $x[t]$ and $y[t]$, respectively. The instantaneous distributions of x and y at time t are $X[t]$ and $Y[t]$, respectively. The cross correlation structure is the covariances between instantaneous distributions of $X[t]$ and $Y[t + \tau]$ for all pairs of time t and time $t + \tau$. To test for the oscillatory correlation, we can first estimate an instantaneous product. For each point of time t, this estimation is the product of the measurements $x[t]$ and $y[t]$, each subtracted by its average over all time t. The power spectrum of this estimation would reflect the oscillatory correlation between the two properties when the harmonic components in the two properties are negligible. This spectrum would be called a cross-SES. We can define a cross-CSES (Equation 2.3) with corrections identical to the corrections used in CSES. When the harmonic components are all negligible, peaks on the cross-CSES reflect correlations involving purely second order cyclostationary components. This is because of the following. When the harmonic components are negligible, the instantaneous average of each property is constant over time and is equal to the average of the property over all time. In this case, the instantaneous product estimates the instantaneous covariance between the instantaneous distributions $X[t]$ and $Y[t]$ of the two properties at each time, t. Correlations between wide sense stationary components have no dependence on time t and cannot generate a peak on the cross-CSES. Correlations between the harmonic components are negligible. The remaining forms of correlations that can generate a peak on the cross-CSES are correlations involving purely second order cyclostationary components. Correlation between two pure oscillatory instantaneous variances of the same frequency f would generate a peak at frequency f on the cross-CSES. Correlation between a pure oscillatory instantaneous variance of one frequency and the wide sense stationary component, or a pure oscillatory instantaneous variance of another frequency, would generate peaks at other frequencies. Notice that while all peaks on the cross-CSES represent a correlation involving a purely second order cyclostationary component, the cross-CSES however do not reveal all forms of correlations involving second order cyclostationary components. This is because oscillation of the cross correlation structure over time *t* for a non-zero separation time τ is not taken into accounted. Consider a property *h* that is purely second order cyclostationary but has no oscillatory instantaneous variance. The cross-CSES between property *h* and itself is the CSES of property *h*. The CSES of property *h* has no peak despite property *h* should relate to itself. This is because the instantaneous variance of property *h* has no oscillation over time *t*.

$$\text{Cross-CSES}[\omega] = \sqrt{2 \cdot \text{Cross-SES}[2\omega]}$$

for ω>0 where

$$\text{Cross-SES}[\omega] = 2\left|\frac{1}{T}\text{FT}_{xy}(\omega)\right|^2,$$

$$\text{FT}_{\text{xy}}(\omega) = \sum_{\text{t=0}}^{T}(x[t] - \mu_x)\left(y[t] - \mu_y\right)e^{-2\pi \text{i}\omega \text{t}},$$

$\mu_x = \frac{1}{T}\sum_{\text{t=0}}^{T} x\,[t]$, and $\mu_y = \frac{1}{T}\sum_{\text{t=0}}^{T} y\,[t]$

(Equation 2.3)

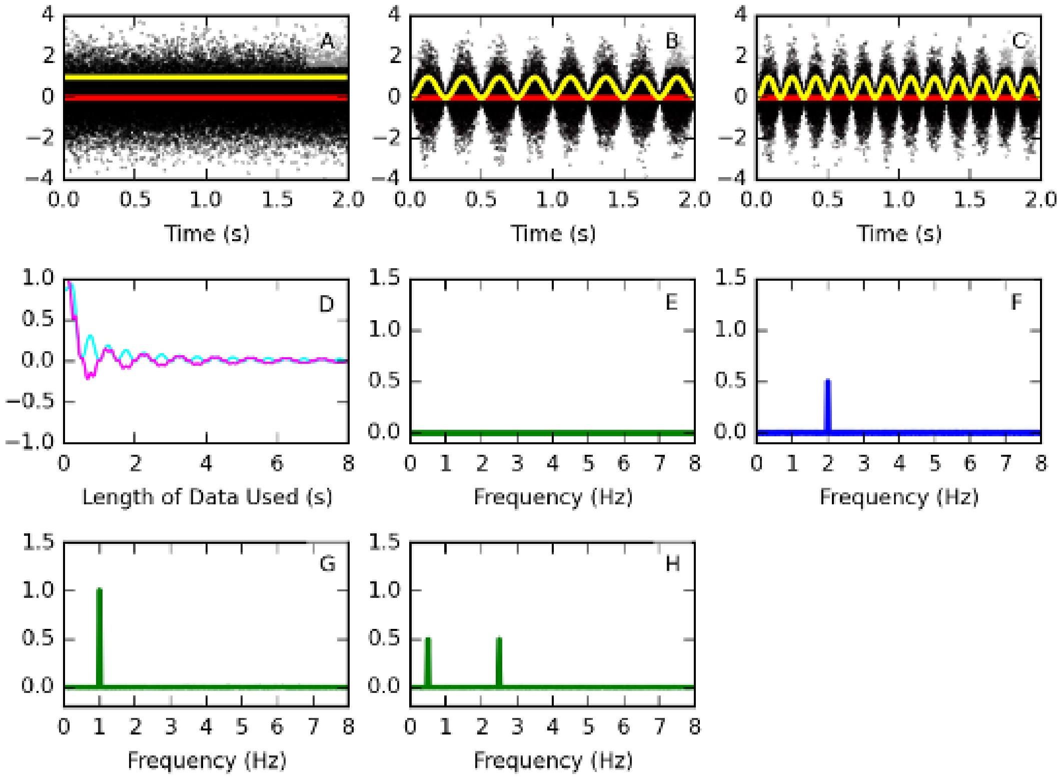

Figure 2.2.3 The yellow curve, red curve, and black dots represent the instantaneous variance, instantaneous average, and measurements of the property in (A), (B), and (C), respectively. The measurements of the properties described in (A), (B), and (C) are $n[t]$, which has a normal distribution of zero average and one variance for each time t and is statistically independent from $n[t_2]$ at any other time t_2, $x[t] = n[t]\sin(2\pi\times 2t)$, and $y[t] = n[t]\sin(2\pi\times 3t)$, respectively. The cyan curve in (D) represents the Pearson correlations between $x[t]$ and $n[t]$ measured for varying lengths of time. The magenta curve in (D) represents the Pearson correlations between $x[t]$ and $y[t]$ measured for varying lengths of time. Power spectrum of $x[t]$ (E), CSES of $x[t]$ (F), cross-CSES between $x[t]$ and $n[t]$ (G), and cross-CSES between $x[t]$ and $y[t]$ (H) are also plotted.

For a more concrete description of power spectrum CSES and cross-CSES, consider a wide sense stationary property n (Figure 2.2.3A). For any time t, the instantaneous distribution $N[t]$ of n is a normal distribution with an average of zero and a variance of one. For any two different times t_1 and t_2, instantaneous distributions $N[t_1]$ and $N[t_2]$ are uncorrelated according to Pearson correlation. Consider also two purely second order cyclostationary properties x and y with measurements $x[t] = n[t]\sin(2\pi\times 2t)$ (Figure 2.2.3B) and $y[t] = n[t]\sin(2\pi\times 3t)$ (Figure 2.2.4C). The Pearson correlations between $x[t]$ and $n[t]$ (cyan curve in Figure 2.2.3D) and between $x[t]$ and $y[t]$ (magenta curve in Figure 2.2.3D) both approach zero if x, y, and n are measured for a long enough time. Pearson correlation suggests $x[t]$, $y[t]$, and $n[t]$ are not correlated, even though they are related and share the same random factor $n[t]$. This is because Pearson correlation as a single number cannot reflect the oscillatory relationship between $x[t]$ and $n[t]$, which have same sign only when $\sin(2\pi\times 2t)$ is positive. Pearson correlation instead reflects the time average of the same and opposite sign cases and approaches zero when x and n are measured for a long time. Similarly, Pearson correlation is not suitable for reflecting the oscillatory correlation between x and y and the oscillatory correlations between second order cyclostationary components of different frequencies.

While the power spectrum can detect harmonic oscillation, the purely second order cyclostationary $x[t]$ is invisible on the power spectrum (Figure 2.2.3E). Contrary to the power spectrum in Figure 2.2.5E, the CSES correctly showed the oscillation of instantaneous variance of $x[t]$ as a single peak at 2 Hz with amplitude $1^2/2$ (Figure 2.2.3F) as expected. The cross-CSES is defined to show oscillatory correlation between two properties. This oscillatory correlation can be caused by correlation between two second order cyclostationary components of same or different frequencies, or correlation between a second order cyclostationary component and a stationary component. Correlation between a second order cyclostationary component with frequency f and a wide sense stationary component would generate a peak with frequency $f/2$ on the cross-CSES. For example, the cross-CSES between $x[t]$ and $n[t]$ has a peak at 1 Hz (Figure 2.2.3G) because $x[t]$ has pure oscillatory instantaneous variance and $n[t]$ is wide sense stationary. The oscillatory correlation between two second-order cyclostationary components of the same frequency would generate a peak at the frequency on the cross-CSES. For example, the cross-CSES between $x[t]$ and $x[t]$ is identical to the CSES of $x[t]$ (Figure 2.2.3F) and has a single peak at 2 Hz, which is the frequency of the oscillatory instantaneous variance of $x[t]$. Oscillatory correlation between two second order cyclostationary components of different frequencies would generate peaks at the halved sum and difference frequencies on the cross-CSES. For example, the cross-CSES between $x[t]$ and $y[t]$ has peaks at (3 – 2)/2 and (3 + 2)/2 Hz (Figure 2.2.3H) and the frequencies of the oscillatory instantaneous variances of $x[t]$ and $y[t]$ are 2 and 3 Hz, respectively.

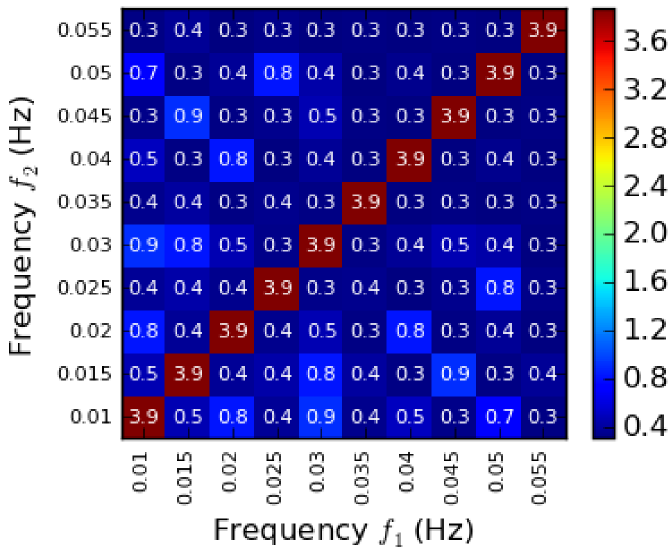

Figure 2.2.6 Mutual information between x'[t] = x[t]/σx and y'[t] = y[t]/σy, where x[t] = n[t]sin(2πf1t) and y[t] = n[t]sin(2πf2t), n[t] is a white noise with 0 average and 1 standard deviation, σy and σx are respectively the standard deviation of x[t] and y[t], and the unit of t is seconds. The grid used for calculating the mutual information is a 1000 × 1000 grid with a cell size of 0.008 × 0.008.

Mutual information cannot correctly account for nonstationary relationship as shown in Figure 2.2.7. Regardless of the frequencies f_1 and f_2, x' can always be predicted from y without any error. This is because $x'[t]$ is proportional to $y'[t]\sin(2\pi f_1 t)/\sin(2\pi f_2 t)$ except when $\sin(2\pi f_2 t)$ is exactly 0. The number of data points with $\sin(2\pi f_2 t) = 0$ is negligible. This means that the x' and y' should always be perfectly correlated regardless of the frequencies f_1 and f_2. However, the mutual information is different for different f_1 and f_2. The mutual information between x' and y' for $f_1 \neq f_2$ is much lower than that for $f_1 = f_2$. This shows that mutual information suggests the strength of relationship is much lower when f_1 and f_2 are not equal, which is not true. The strength of the relationship between x' and y' should be the same regardless of f_1 and f_2. This shows that mutual information cannot correctly account for nonstationary relationships.

The spectrum of a stochastic process is also stochastic. A power spectrum of a single sample of a white noise process does not have uniform magnitude across frequencies. The expected power spectrum is the average of the power spectrum of an infinite number of trajectories of the white noise process. This expected power spectrum can be estimated by smoothing the power spectrum of a single sample of the stochastic process (Figure 2.5.4A). The distortions caused by this estimation are small when the peaks in the expected spectrum are significantly wider than the length of the smoothing window. This smoothed spectrum is called a smoothed periodogram.[53] The spectrum of a property of a molecular dynamics simulation depends on the sampling frequency of the property. Cutting down the rate of saving the trajectory by half is cutting down the sampling frequency of the property by half. If the actual frequency of oscillation of a property is greater than half of the sampling frequency, aliasing will occur. The half of the sampling frequency is the Nyquist frequency. Aliasing means the peaks beyond the Nyquist frequency will be folded and become a peak with frequency lower than the Nyquist frequency (Figure 2.5.4B). An example of aliasing is illustrated in Figure 2.5.4C and Figure 2.5.4D. Power spectrum, CSES, and cross-CSES are all affected by aliasing. Aliasing affect these spectra in the same way as illustrated in Figure 2.5.4C and Figure 2.5.4D. This is because Fourier transform is affected by aliasing.[54]

2.3 Results and Discussion

Dihedral Angle Name	Definition
A	from bond a through b to c
B	from bond b through c to d
C	from bond c through d to e
D	from bond d through e to f
E	from bond e through f to g
F	from bond f through g to h
G	from bond g through h to I

Figure 2.3.1 Stick model of alanine tripeptide with names of the main-chain signed dihedral angles.

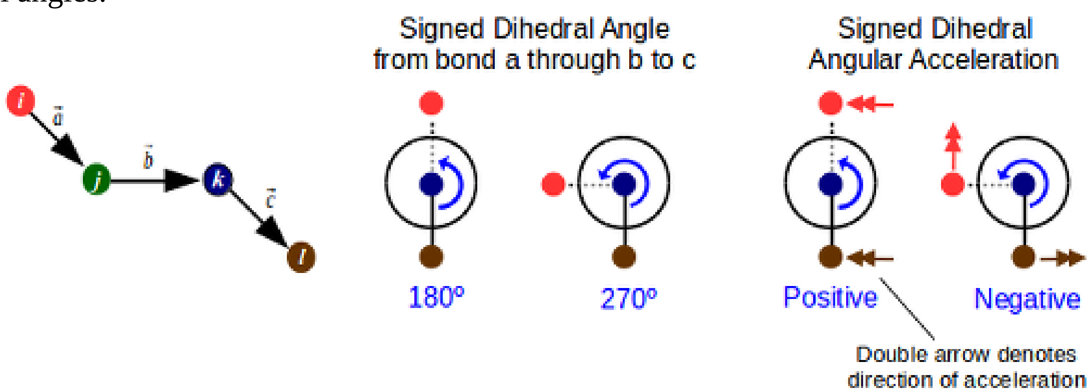

Figure 2.3.2 Signed dihedral angle and signed dihedral angular acceleration on Newman projection.

To test for components of oscillatory instantaneous variance and correlation involving second-order cyclostationary components in a biological model system, signed dihedral angular acceleration (SDAA) of main chain signed dihedral angles of alanine tripeptide (Figure 2.3.1) were used. The second time derivative of the signed dihedral angle θ of Blondel and Karplus[55] (Figure 2.3.2 and (Equation *2.4*)) is the SDAA (Figure 2.3.2 and (Equation *2.5*)), where $\vec{F}_i, \vec{F}_j, \vec{F}_k, \vec{F}_l$, m_i, m_j, m_k, and m_l are net forces on and masses of atom *i*, *j*, *k*, and *l* as defined in Figure 2.3.1 and the vectors $\vec{a}, \vec{b}, \vec{c}$ are as defined in Figure 2.3.1. In the derivation of SDAA, all first time derivatives were set to zero because the potential energy, force, moment of inertia, and acceleration have no dependence on velocity. The use of SDAA avoided the inaccuracy of on-step velocity of leapfrog integrator[56] and divisions by zero in the definition of the unsigned dihedral angle at 0° and 180°.[55] Leapfrog integrator was used because it provides better conservation of energy and causes lower resonance artifacts, compared to the velocity Verlet integrator.[56]

$\theta = \text{atan2}(y, x)$, where $\vec{u} = \vec{a} \times \vec{b}$, $\vec{v} = \vec{b} \times \vec{c}$,

$y = \vec{u} \times \vec{v} \cdot \vec{b}$, and $x = \vec{u} \cdot \vec{v}\left|\vec{b}\right|$

(Equation 2.4)

$\theta'' = \frac{xy'' - yx''}{x^2 + y^2}$ where

$y'' = \vec{u}'' \times \vec{v} \cdot \vec{b} + \vec{u} \times \vec{v}'' \cdot \vec{b} + \vec{u} \times \vec{v} \cdot \vec{b}''$,

$x'' = (\vec{u}'' \cdot \vec{v} + \vec{u} \cdot \vec{v}'')\left|\vec{b}\right| + (\vec{u} \cdot \vec{v})\vec{b}'' \cdot \vec{b}/\left|\vec{b}\right|$,

$\vec{u}'' = \vec{a}'' \times \vec{b} + \vec{a} \times \vec{b}''$, $\vec{v}'' = \vec{b}'' \times \vec{c} + \vec{b} \times \vec{c}''$,

$\vec{a}'' = \vec{F}_j/m_j - \vec{F}_i/m_i$, $\vec{b}'' = \vec{F}_k/m_k - \vec{F}_j/m_j$,

and $\vec{c}'' = \vec{F}_l/m_l - \vec{F}_k/m_k$

(Equation 2.5)

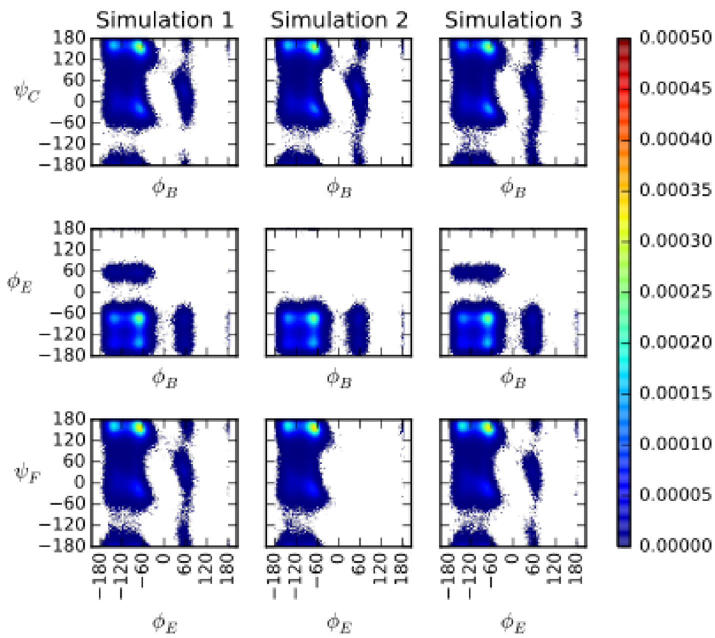

Figure 2.3.3 Ramachandran plot of alanine tripeptide in three 100 ns simulations under the NVE ensemble. The color represents the estimated probability density. The subscript of the Greek letters represents the identity of the dihedral angles as defined in Figure 2.3.1.

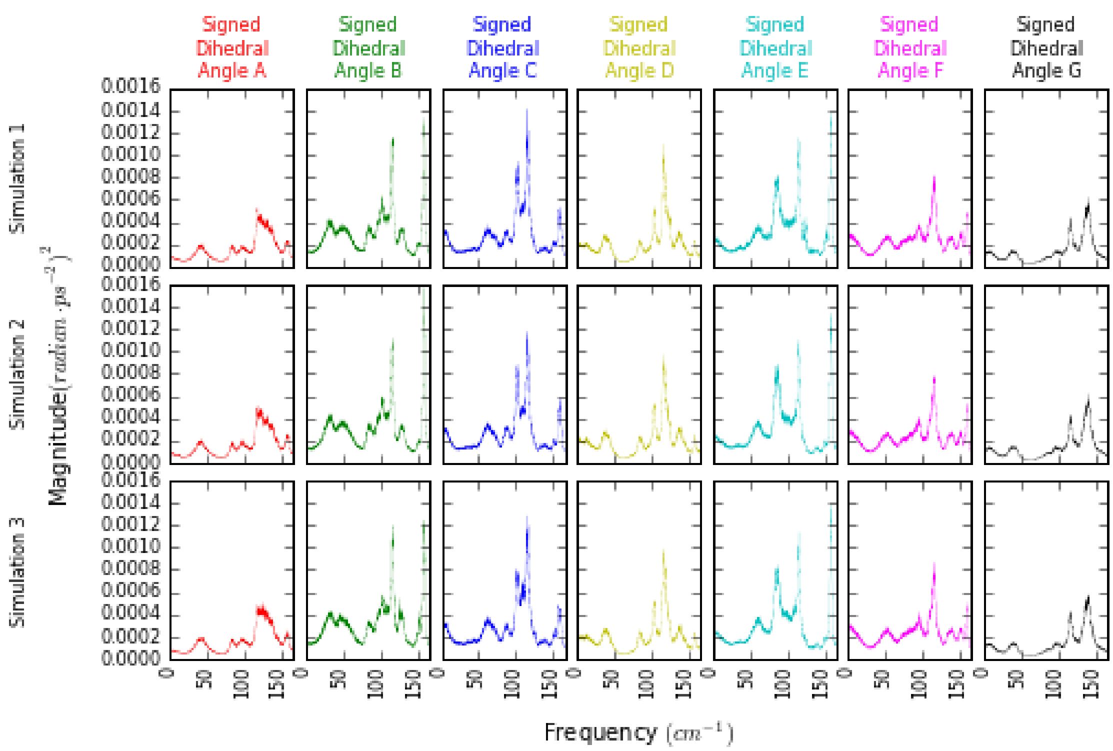

Figure 2.3.4 Smoothed power spectra of the signed dihedral angular accelerations (SDAA) of each of the seven dihedral angles defined in Figure 2.3.1. The power spectrum of

SDAA without smoothing can be found in Figure 2.5.5. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in water under the NVE ensemble.

Three simulations of 100 ns under the NVE ensemble were performed. The sampling of the conformations of the peptide is reflected by the Ramachandran plot in Figure 2.3.4. For each of the seven signed dihedral angles, the smoothed power spectra of SDAA (Figure 2.3.4) calculated from the three simulations agree with each other (Figure 2.5.6 and Figure 2.5.1). The power spectrum before smoothing can be found in Figure 2.5.5. Smoothing was used to estimate the expected magnitude at each frequency of the spectrum. This is because spectrum of a stochastic process has stochastic magnitude.

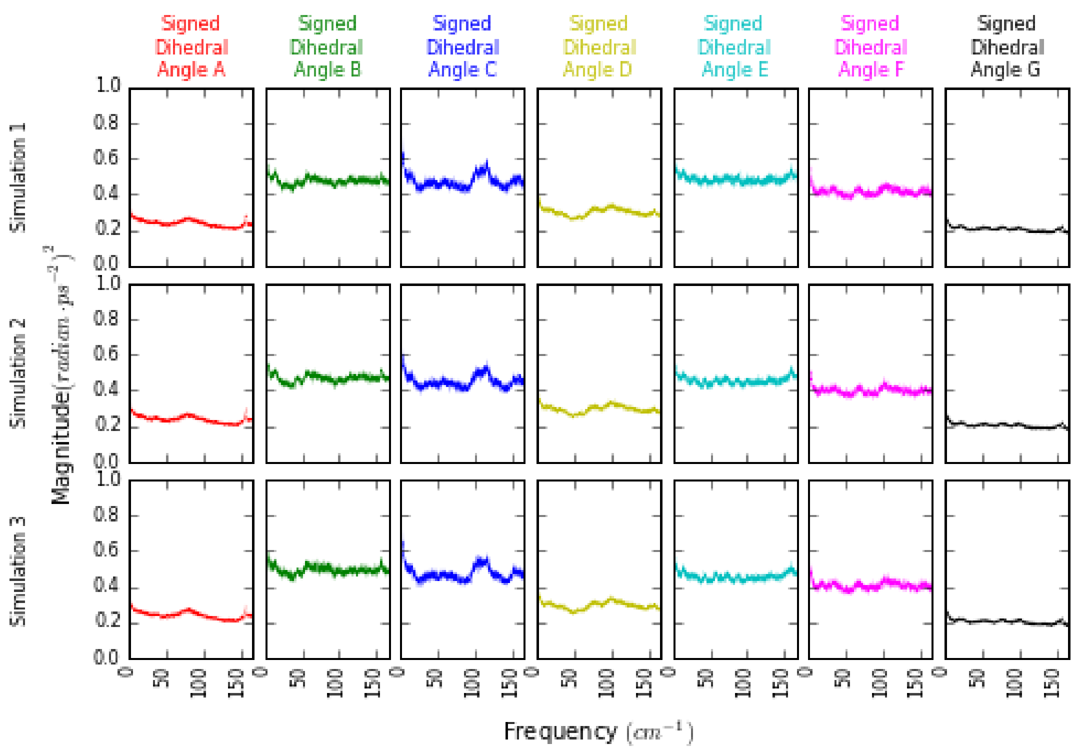

Figure 2.3.5 Smoothed corrected squared envelope spectrum (CSES) of the signed dihedral angular acceleration (SDAA) of the seven signed dihedral angles defined in Figure 2.3.1. The CSES of SDAA without smoothing can be found in Figure 2.5.7. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in water under the NVE ensemble.

Peaks on the power spectrum of SDAA (Figure 2.3.4) reflect the intensity of harmonic oscillation in SDAA. Peaks on the CSES of SDAA (Figure *2.3.5*) reflect the combined intensity of components with pure oscillatory instantaneous variance and components with harmonic oscillation in SDAA at each frequency. The harmonic components in the SDAA were insignificant because the peaks on the power spectrum of SDAA (Figure 2.3.4) are far lower than the peaks on the CSES of SDAA (Figure 2.3.5). This shows that rotations of the dihedral angles were heavily over-damped. This implies that the peaks on the CSES of SDAA purely reflect components with pure oscillatory instantaneous variance. The baseline of CSES of SDAA of signed dihedral angles A, D, and G are lower than baseline of CSES of SDAA of other main chain signed dihedral angles (Figure 2.3.5). This reflects that signed dihedral angles A, D, and G are around peptide bonds and have lower freedom of rotation.

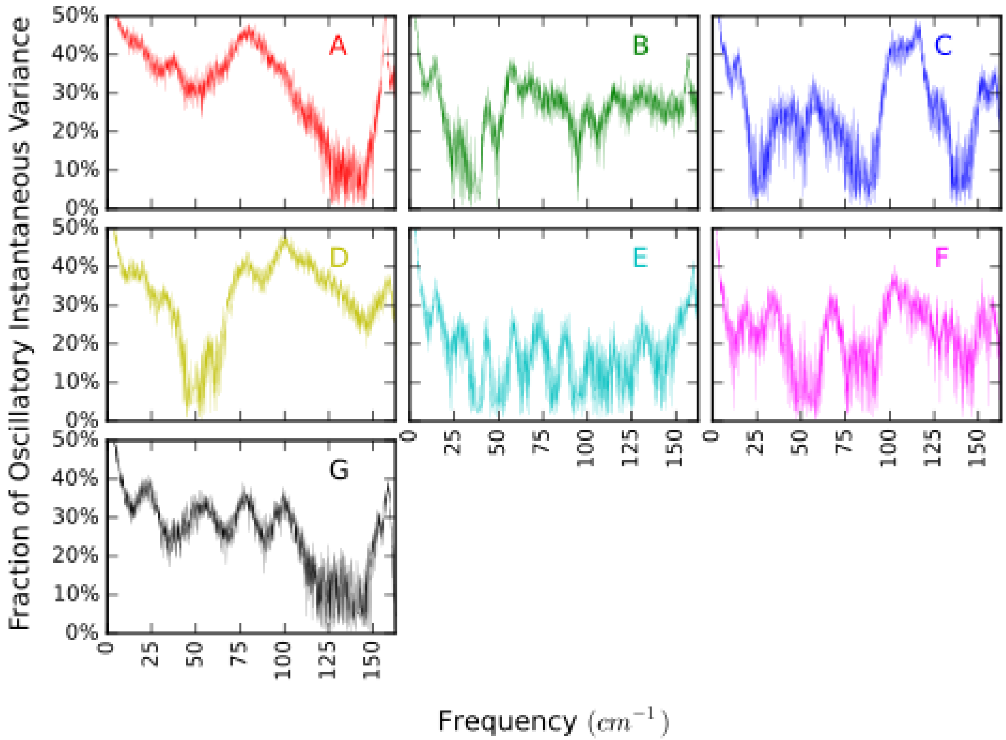

Figure 2.3.6 Fraction of oscillatory instantaneous variance in the signed dihedral angular acceleration (SDAA) of the seven signed dihedral angles defined in Figure 2.3.1. The letter in each subplot denotes the identity of the signed dihedral angles that the subplot describes. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in water under the NVE ensemble.

For each of the seven main chain signed dihedral angles defined in Figure 2.3.1, the CSES of SDAA calculated from the three simulations have good agreement with each other (Figure 2.5.6). This means that an average of the three CSES of each signed dihedral angle would provide a less noisy description of the SDAA. The CSES of SDAA (Figure 2.3.5) have baselines that represent the variance of wide sense stationary component and peaks that represent components with pure oscillatory instantaneous variances because the harmonic components are negligible (Figure 2.3.4). To calculate the fraction of oscillatory instantaneous variance at each frequency, the lowest point of each smoothed CSES was extracted, after smoothing the CSES with a running average with a window that was 1/10 of the frequency range of the CSES and then finding the lowest point. The magnitude of this lowest point, say *l*, reflects the variance of the part of the SDAA that has no oscillatory instantaneous variance, distributed evenly over all frequencies. The fraction of oscillatory instantaneous variance in all motion at frequency ω was calculated as:

$$K(\omega) = \sqrt{\frac{CSES(\omega) - l}{CSES(\omega)}} \times 100\%$$

(Equation 2.6)

This fraction of instantaneous variance oscillation is shown in Figure 2.3.6. Most peaks in the fractions of oscillatory instantaneous of the seven signed-dihedral angles have 20% to 40% magnitudes (Figure 2.3.6). This shows that pure oscillatory instantaneous variance and oscillatory diffusion are significant. The strength of second order cyclostationarity in SDAA is likely higher than the strength of oscillatory instantaneous variance detected by the CSES of SDAA because some forms of second order cyclostationarity can have no oscillation over time in instantaneous variance. This significant oscillatory instantaneous variance also suggests that significant correlation would not be accounted for in current methods because second order cyclostationarity was found to be significant. The fraction of oscillatory instantaneous variance, smoothed power spectrum, and smoothed CSES for SDAA of MD simulation of alanine tripeptide in vacuum can be found in Figure 2.5.8 and Figure 2.5.9. The magnitude of harmonic components in vacuum simulation is about one to two times the magnitude of harmonic components in explicit water. The magnitude of CSES of SDAA in ~~the~~ vacuum is comparable to that in water. The fractions of oscillatory instantaneous variance in vacuum are about 0-20% different from the corresponding fractions of oscillatory instantaneous variance in water (Figure 2.3.6).

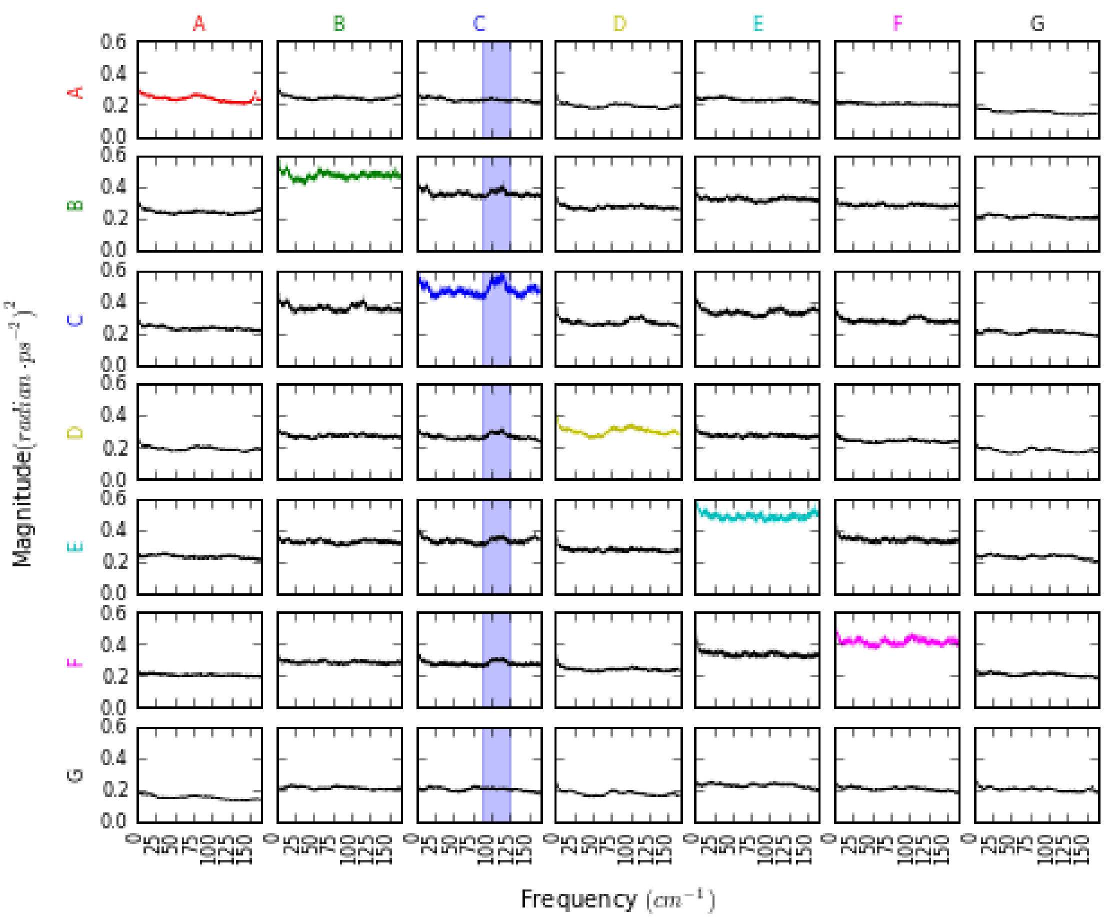

Figure 2.3.7 Smoothed corrected cross squared envelope spectrum (cross-CSES) between the seven signed dihedral angular accelerations (SDAA) as defined in Figure 2.3.1. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in explicit water under the NVE ensemble.

The cross-CSES between dihedral angle C and each of dihedral angle B, C, D, E, and F had peaks in the highlighted range of frequencies (Figure 2.3.8). This means second order cyclostationary motions of the same range of frequencies caused the oscillatory correlations between SDAA of these dihedral angles. This shows that the second order cyclostationary motion could transmit between these dihedral angles. This implies second order cyclostationary motions can transmit through a molecule and could have implications for the dynamics of biomolecules.

2.4 Conclusion

Measures of correlations are the foundation of many bottom-up analysis methods for molecular dynamics (MD) simulations.[23-25] The common measures of correlation used in analysis of MD simulations include dynamic cross correlation,[21] Pearson correlation,[57] and a common estimation of mutual information.[22] These measures of correlation can be inadequate because they do not correctly account for correlations involving nonstationary components.[33, 58] Internal motions of biomolecules have nonstationary components because motions of different frequencies are correlated in biomolecules,[40] as shown by 2D-IR experiments.[39] This nonstationary component of motion includes purely second order cyclostationary components,[41, 47, 52] which include components with pure oscillatory instantaneous variance. Oscillatory instantaneous variance in acceleration has a physical interpretation of an oscillatory diffusion. Significant components with pure oscillatory instantaneous variance were found in the heavily over-damped signed dihedral angular acceleration (SDAA) of main-chain dihedral angles of alanine tripeptide from MD simulations. The fraction of oscillatory instantaneous variance in the SDAA was found to be about 40% at some frequencies by a comparison of the power spectrum of SDAA and corrected squared envelope spectrum (CSES) of SDAA. This form of oscillation was found to transmit between dihedral angles of alanine tripeptide, as shown by the cross-CSES of the simulations. More sophisticated cyclostationary analysis of MD simulations may provide additional insights into dynamical motions of biomolecules. This is because this characterization can better account for nonstationary motions and correlations between motions of different time scales than current correlation analysis of MD simulations.

2.5 Supplementary Information

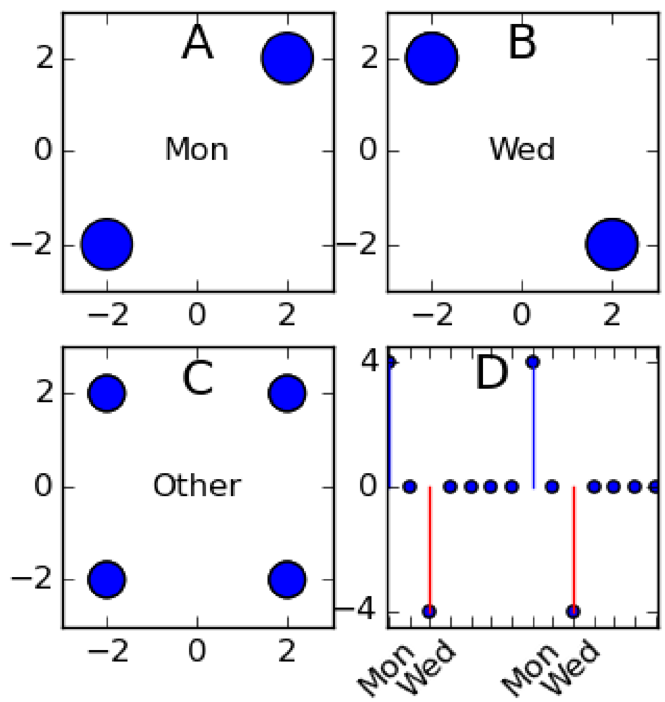

Figure 2.5.1 Two students A and B who stochastically visit a library on everyday associated with property S. If student A visits the library on a day t, then $S_A(t) = 1$; otherwise $S_A(t) = -1$. For student B, $S_B(t)$ is similarly defined. (A) Joint probability distribution of $S_A(t)$ and $S_B(t)$ on each Monday. Size of dot represents probability density. (B) Joint probability distribution of $S_A(t)$ and $S_B(t)$ on each Wednesday. (C) Joint probability distribution of $S_A(t)$ and $S_B(t)$ on each day that is neither a Monday nor a Wednesday. (D) Covariance between $S_A(t)$ and $S_B(t)$ on each day.

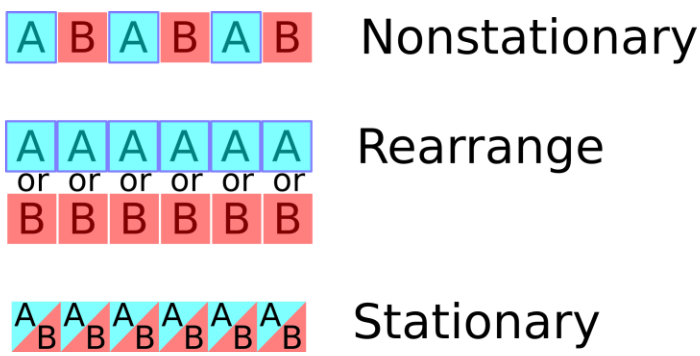

Figure 2.5.2 Suppose properties *X* and *Y* have two probable relationships *A* and *B*. If the relationship alternate between *A* and *B* over time, the relationship is nonstationary. Suppose we randomly rearrange the data such that all possible rearrangements are equally probable. Then at all points of time, relationships *A* or *B* are equally likely. This means in the rearranged data the relationship between *X* and *Y* doesn't change over time and is stationary. Consider a correlation measure that does not change upon reordering the data over time. This measure would have the same value for the original and the rearranged data. This means the measure cannot distinguish the nonstationary and stationary cases. This means the measure cannot correctly account for nonstationary relationships. For example, this correlation measure may not find any relationship in the original or the reordered data. Yet, relationships that periodically appear and disappear, such as relationships *A* and *B* in the original data, can still exist.

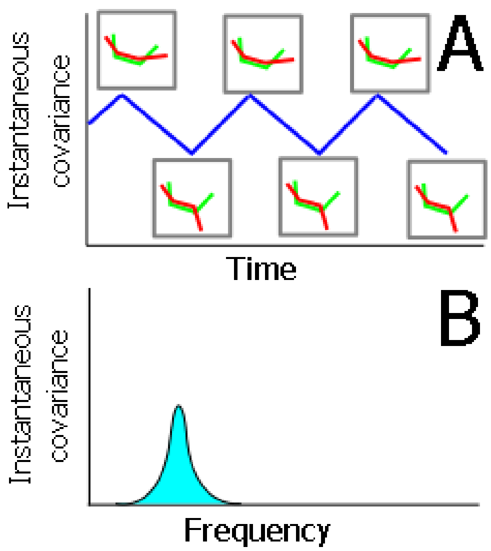

Figure 2.5.3 (A) Each black box contains possible conformations of a molecule at a specific time. The black boxes represent an ensemble of conformations that varies over time. For each of the black box we can calculate covariance between two properties. The blue curve represents this covariance, which varies over time. (B) The spectrum of the instantaneous covariance depicted in (A).

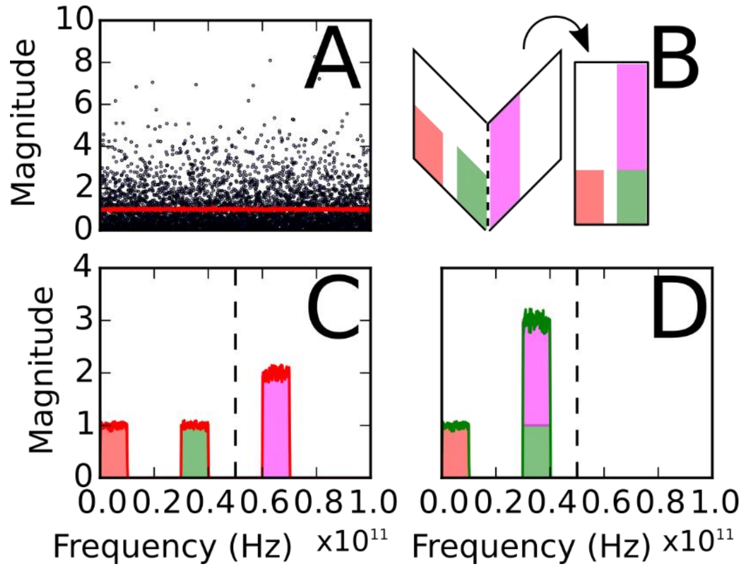

Figure 2.5.4 (A) Spectrum of a white noise signal is represented by black dots. The red curve represents the smoothed power spectrum of the white noise with a moving average of 1000 points over frequency. (B) Schematic diagram for frequency folding upon halving the sampling rate. (C) Red curve is the smoothed spectrum of a synthesized signal. The color of the filled rectangles represent the different peaks. The black dashed line indicates the center of the frequency axis. (D) The sampling rate of the signal in C is halved by discarding half of the data points in the signal in C. The green curve is the smoothed spectrum of this new signal. The filled rectangle represents how the peaks in C with corresponding filled color contribute to peaks in this spectrum.

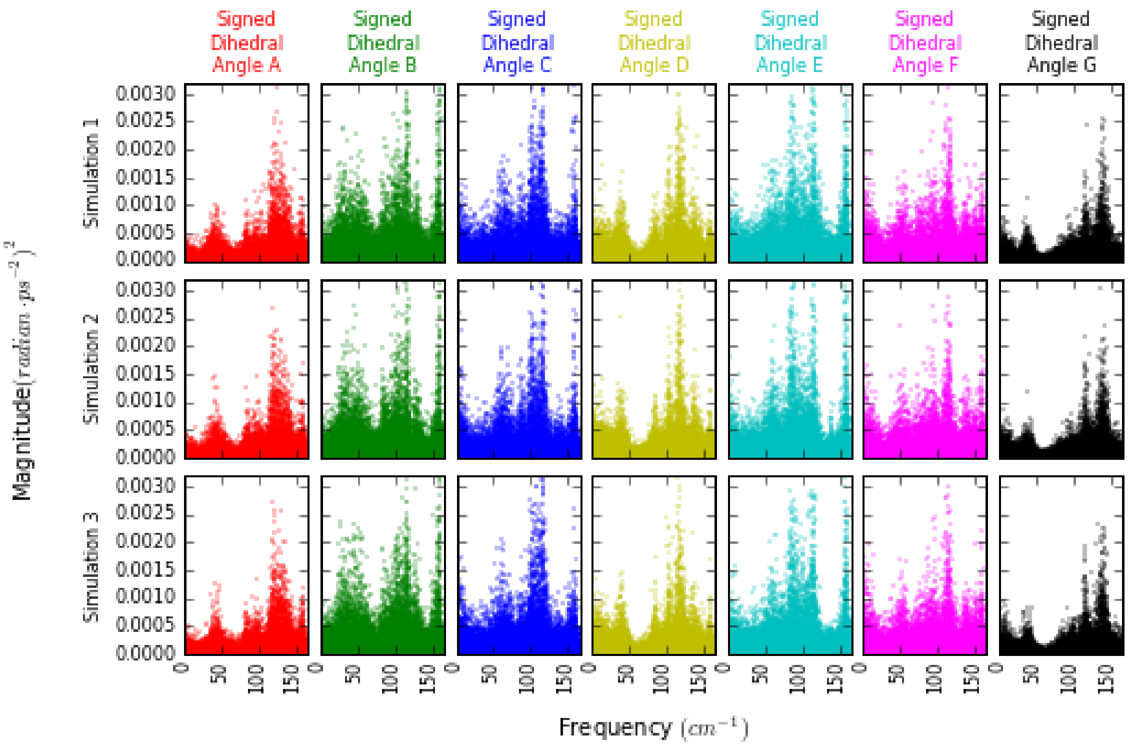

Figure 2.5.5 Power spectra of the signed dihedral angular accelerations (SDAA) of each of the seven dihedral angles defined in Figure 2.3.1. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in water under the NVE ensemble.

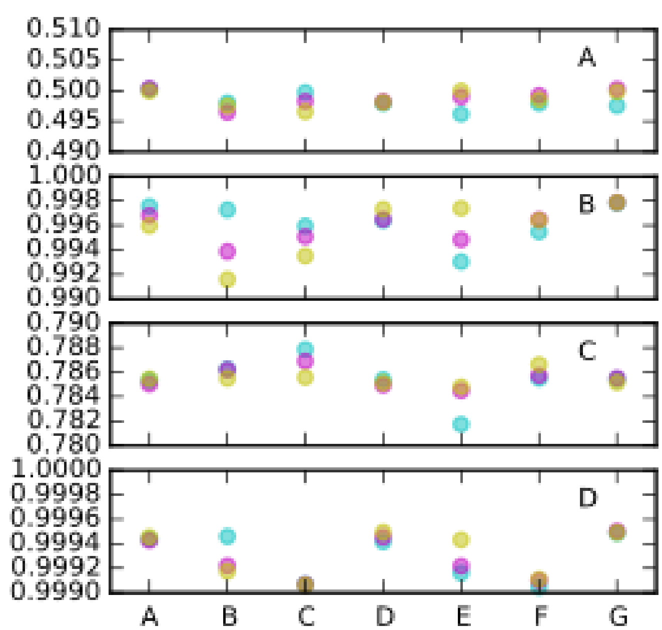

Figure 2.5.6 Pairwise similarity of spectrum as measured by a measure based on Schwartz inequality.[59] The labels of the horizontal axis is the identity of the signed dihedral angular acceleration (SDAA) as defined in Figure 2.3.1 and Figure 2.3.2. The cyan, magenta, and yellow dots respectively represent the similarities between the spectrum of the first and second simulations, between the spectrum of the first and third simulations, and between the spectrum of the second and third simulations. (A), (B), (C) and (D) are respectively the pairwise similarity between the power spectrum, smoothed power spectrum, corrected squared envelope spectrum (CSES), and smoothed CSES of the SDAA of the same dihedral angle in the three simulation. A similarity greater than 0.99, 0.999, and 0.9999 respectively represent an acceptable, good, and excellent fit.[59] The similarity between the spectrum greatly increases upon smoothing. This shows the expected power spectrum estimated by smoothing and calculated from the three simulations are acceptably similar to each other. The expected CSES estimated by smoothing and calculated from the three simulations have good similarity.

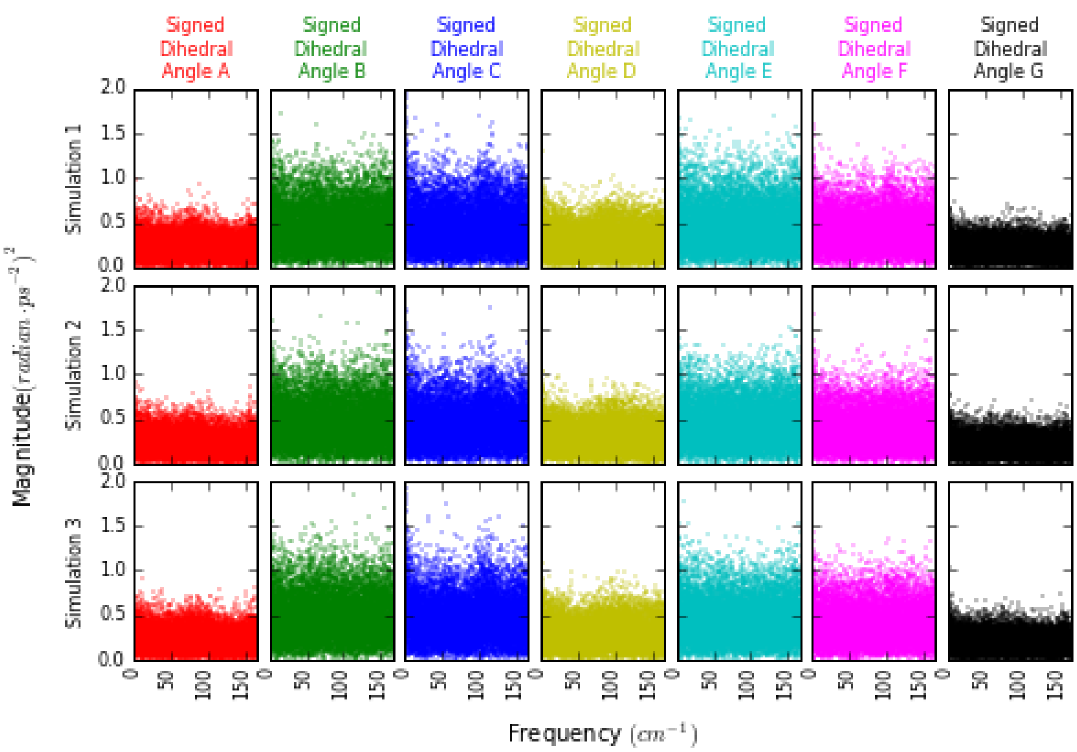

Figure 2.5.7 Corrected squared envelope spectrum (CSES) of the signed dihedral angular accelerations (SDAA) of each of the seven dihedral angles defined in Figure 2.3.1. The SDAA are calculated from the three 100 ns simulations of alanine tripeptide in water under the NVE ensemble.

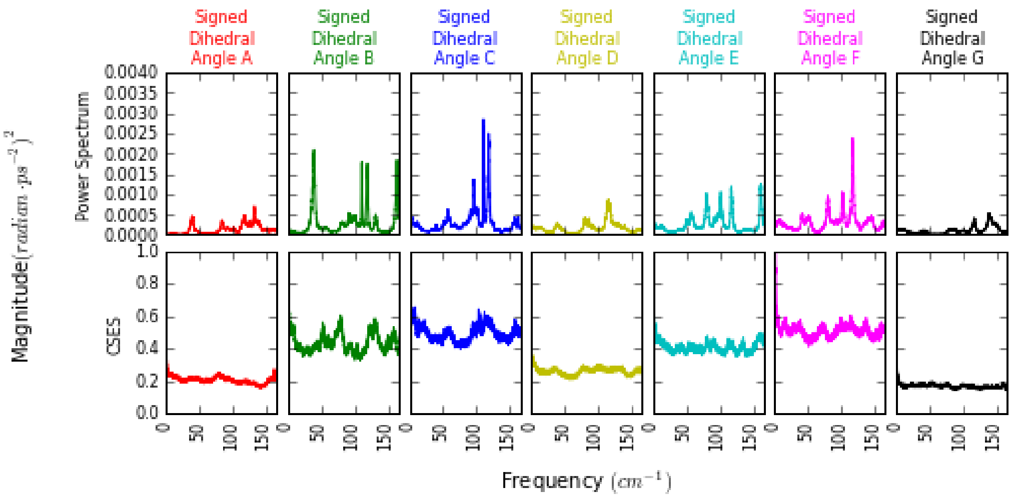

Figure 2.5.8 Smoothed power spectrum and smoothed corrected squared envelope spectrum (CSES) of the signed dihedral angular accelerations (SDAA) of each of the seven dihedral angles defined in Figure 2.3.1. The SDAA are calculated from one 100 ns simulation of alanine tripeptide in vacuum under the NVE ensemble. The magnitude of harmonic components in the vacuum simulation is about one to two times of the magnitude of harmonic components in the three simulation of alanine tripeptide in water. The magnitude of CSES of SDAA in the vacuum simulation is comparable to that in the three simulations of alanine tripeptide in water under NVE ensemble.

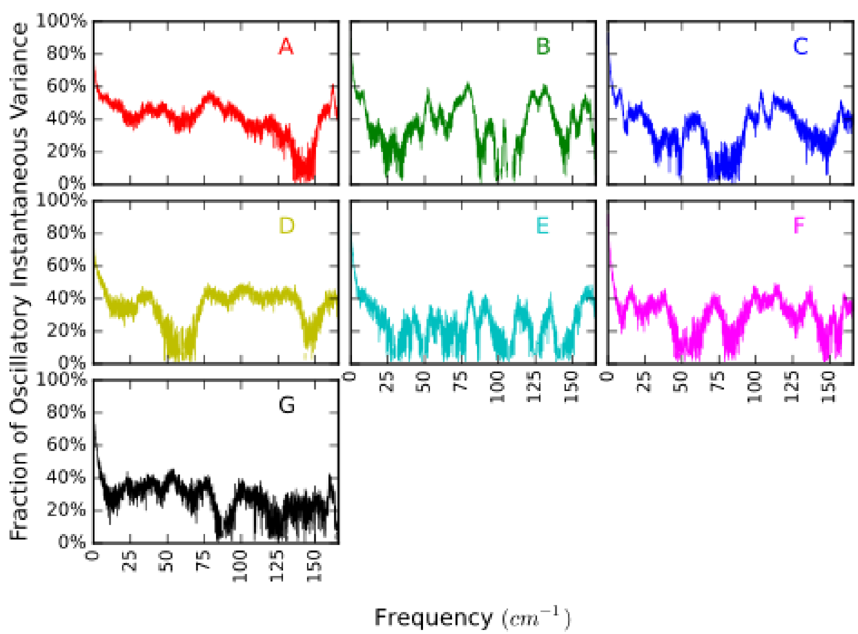

Figure 2.5.9 Fraction of oscillatory instantaneous variance in the signed dihedral angular acceleration (SDAA) of the seven signed dihedral angles defined in Figure Figure 2.3.1. The letter in each subplot denotes the identity of the signed dihedral angles that the subplot describes. The SDAA are calculated from one 100 ns simulation of alanine tripeptide in vacuum under the NVE ensemble. The fractions of oscillatory instantaneous variance for the vacuum simulation

are about 0-20% different from the corresponding fractions of oscillatory instantaneous variance for the simulations of alanine tripeptide in water in Figure 2.5.9.

2.6 Simulation of Alanine Tripeptide in Vacuum under NVE Ensemble

The 100 ns simulation of alanine tripeptide were performed as follow. The frame at 135.5 ns of the 511 ns NPT equilibration simulation of alanine tripeptide in water was extracted. Water was removed from the frame. The geometry of alanine tripeptide in the frame was used to start an NVE simulation with initial velocity drawn from a Maxwell-Boltzmann distribution of 300K. No barostat, no thermostat, no constraint of any bond, or no periodic boundary condition were used.

Infinite cutoff for short range coulombic and short-range Van der Waals forces were used. The simulation also had translation and rotation around center of mass removed, group cutoff scheme, and time steps of 0.5 fs. Trajectory frames were saved every 100 time steps.

2.7 Mathematical Proof of Correlation between Components of Different Frequencies Implies Nonstationary Relationship.

Consider stochastic motion of an object with *x*-coordinate $X(t)$ at time t. $X(t)$ has the following two properties:

1. $X(t)$ and itself are jointly wide sense stationary
2. Average of $X(t)$ over a long time is zero

According to the definition of wide sense stationarity (Figure 2), the first property implies $E\{X(t)\}$ is constant over time. $E\{X(t)\}$ is the instantaneous average of $x(t)$ at time t where E{.} denotes expectation.

Notice that we can rearrange integration and expectation in the following way:

$$E\left\{\int_0^1 X(t)dt\right\} = E\left\{\lim_{n\to\infty}\sum_{i=0}^{n} X\left(\frac{i}{n}\right)\times\frac{1}{n}\right\} = \lim_{n\to\infty}\sum_{i=0}^{n} E\left\{X\left(\frac{i}{n}\right)\right\}\times\frac{1}{n} = \int_0^1 E\{X(t)\}dt$$

This is because $E\{X_1 + X_2\} = E\{X_1\} + E\{X_2\}$ and $E\{cX_1\} = cE\{X_2\}$ for any constant c and any two random variables X_1 and X_2. $X(t)$ is a random variable for each time t.

The second property of $X(t)$ means:

$$0 = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} X(t)dt$$

We have the following because $E\{X(t)\}$ is constant over time.

$$0 = E\{0\} = E\left\{\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} X(t)dt\right\} = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} E\{X(t)\}dt = E\{X(t)\}$$

The complex-valued Fourier transform of $X(t)$ is the following:

$$\tilde{X}(\omega) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} X(t)e^{-i2\pi\omega t}dt = FT\{X(t)\}$$

We have the following because $E\{X(t)\} = 0$ for all time t.

$$E\{\tilde{X}(\omega)\} = E\left\{\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} X(t)e^{-i2\pi\omega t}dt\right\} = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} E\{X(t)\}e^{-i2\pi\omega t}dt = 0 + 0i$$

because for each t, $X(t)$ is a random variable, $e^{-i2\pi\omega t}$ is constant, and $E\{cX_1\} = cE\{X_2\}$.

The covariance between two frequency components of x at frequencies $\omega + \xi$ and ω is:

$$E\{[\tilde{X}(\omega+\xi) - E\{\tilde{X}(\omega+\xi)\}][\tilde{X}(\omega) - \tilde{X}(\omega)]^*\}$$

where the asterisk denotes complex conjugate.

The above expression is equal to the following:

$E\{\tilde{X}^*(\omega+\xi)\tilde{X}(\omega)\}$ because as $E\{\tilde{X}(\omega)\} = 0 = E\{\tilde{X}(\omega+\xi)\}$

$$= E\left\{\left(\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)e^{-i2\pi(\omega+\xi)t}dt\right)^* \left(\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)e^{-i2\pi t}dt\right)\right\}$$

$$= E\left\{\left(\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)e^{-i2\pi\omega t}e^{-i2\pi\xi t}dt\right)^* \left(\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} x(t)e^{-i2\pi\omega t}dt\right)\right\}$$

$$= E\left\{\left(FT\{x(t)e^{-i2\pi\xi t}\}\right)^* FT\{x(t)\}\right\}$$

$= E\left\{FT\{\lim_{S\to\infty} \frac{1}{2S}\int_{-S}^{S} \left(x(s)e^{-i2\pi\xi s}\right)^* x(s+t)ds\}\right\}$ by the cross correlation theorem

$$= E\left\{FT\{\lim_{S\to\infty} \frac{1}{2S}\int_{-S}^{S} x(s)^* e^{i2\pi\xi s}x(s+t)ds\}\right\}$$

as complex conjugate is distributive over multiplication

$= E\left\{FT\{\lim_{S\to\infty}\frac{1}{2S}\int_{-S}^{S}x(s)e^{i2\pi\xi s}x(s+t)ds\}\right\}$

as x(t) is real-valued and complex conjugate of a real number is the real number.

$= FT\{\lim_{S\to\infty}\frac{1}{2S}\int_{-S}^{S}E\left\{x(s)e^{i2\pi\xi s}x(s+t)\right\}ds\}$

$= FT\{\lim_{S\to\infty}\frac{1}{2S}\int_{-S}^{S}E\{x(s)x(s+t)\}e^{i2\pi\xi s}ds\}$

As $x(t)$ is wide sense stationary, $E\{x(s)x(s+t)\} = C(t)$

is a non-stochastic function that depends on t but not on s.

$= FT\{\lim_{S\to\infty}\frac{1}{2S}\int_{-S}^{S}C(t)e^{i2\pi\xi s}ds\}$

$= FT\{C(t)\times\lim_{S\to\infty}\frac{1}{2S}\int_{-S}^{S}e^{i2\pi\xi s}ds\}$

$= FT\{C(t)\times 0\}$ if $\xi \neq 0$

The average of the complex exponential over s is zero if the frequency ξ is nonzero.

$= 0$ if $\xi \neq 0$

The above shows that if $x(t)$ is jointly wide sense stationary with itself, covariance between any two of its frequency component at different frequencies is zero.

This means that when the covariance between any two frequency components of $y(t)$ of different frequencies is nonzero, the following is true. $y(t)$ and itself are not jointly wide sense stationary. This means that $y(t)$ is nonstationary. This assumes $y(t)$ has zero average over long time.

Similarly, if the covariance between a frequency component of $g(t)$ at one frequency and a frequency component of $h(t)$ at a different frequency is nonzero, $g(t)$ and $h(t)$ are jointly wide sense nonstationary. This assumes $g(t)$ and h(t) have zero averages over long time.

Granger's description of Cramér's spectral representation of stationary random processes also contain the above conclusions in page 424-425.[60] The review by Gardner and others stated: "wide-sense stationary processes have autocorrelation functions that are independent of time, depending on only the lag parameter, and all distinct spectral components are uncorrelated."[61] This means if two distinct spectral components of a random process are correlated, the random process is nonstationary.

2.8 Damping, Wide Sense Stationarity, and Forms of Nonstationarity in Molecular Motion

While harmonic oscillation is a form of nonstationarity, it is not necessarily the main form of nonstationarity in internal motions of biomolecules. Internal motions of biomolecules, especially low frequencies motions, are often overdamped and diffusive.[62-63] Overdamping means that these motions are nearly aperiodic, and friction dominates these motions.[62] The friction is provided by water and other parts of the biomolecules. These motions therefore have constant instantaneous averages because they are aperiodic. The rates of these motions are reflected by their diffusion coefficients. The instantaneous diffusion coefficient is proportional to the instantaneous slope of the mean squared displacement versus time.[64] The friction against these motions can be wide sense stationary or nonstationary. Both Brownian diffusion[43] and harmonically confined Brownian diffusion[43] have wide sense stationary friction and acceleration. The above considerations show that overdamping does not necessarily mean these motions are wide sense stationary, because their correlation structures may not be constant over time t for each of all time delays, τ.

The nonstationary component in the overdamped diffusive internal motions of a biomolecule at thermal equilibrium with water has some additional restrictions. Firstly, this nonstationary component is not a harmonic oscillation. This is because the motions are aperiodic as they are overdamped. Secondly, the separations between atoms of the biomolecule are limited by interactions between the atoms, such as covalent bonds. This implies that the biomolecule is unlikely to have a property with a linear trend of constant rise or fall. For example, a constant drop of total energy suggests that a simulation is not at equilibrium.[65] Finally, this nonstationary component may be chaotic with nearly no periodicity in many statistical measures[66] or second order cyclostationary.

3 COMBINATORIAL COARSE-GRAINING OF MOLECULAR DYNAMICS SIMULATIONS FOR DETECTING RELATIONSHIPS BETWEEN LOCAL CONFIGURATIONS AND OVERALL CONFORMATIONS

https://pubs.acs.org/doi/10.1021/acs.jctc.8b00333

3.1 Introduction

Molecular dynamics (MD) has become a well-established technique to study the dynamics of biomolecules, complementing experiments to provide more detailed atomistic explanation of sub-cellular processes.[67] Over the years, tremendous effort has been put in improving the molecular mechanics force fields.[68-71] Also, many new algorithms have been proposed to extend the time scale of the simulations.[72] Generating simulation data of biomolecules is now becoming routine with graphical processing units,[73-74] high performance computers, or specialized computers.[75-76] However, representing this massive amount of data to make sense of the information, that is, finding the needle in the haystack, is an additional barrier to fully understanding the relationship between dynamics and function. For example, the analysis is never efficient, objective, or thorough without using appropriate statistical approaches and making the appropriate connection to experiments.

The relationship between different regions of a biological macromolecule is key to understanding their function. For example, allostery is a causal relationship between an allosteric site and an active site that are distal from each other.[77] The function of biomolecules is often explained through the interactions between their domains, secondary structures, and smaller components at various spatial scales. Changes of two local configurations, such as three-dimensional (3D) arrangements of amino acid side chains, secondary structures, or domains, of a protein could correlate to each other even if the changes are far apart or occur at distinct spatial scales. The dynamical relationship between different local configurations of the protein could provide valuable insights into the nature of the communication between different allosteric sites, for example. More specific examples include the spatial arrangement of the subunits of hemoglobin dictates the cooperative binding of oxygen,[78] the change of the packing of the alpha helices within subunit c of ATP synthase can affect the translocation of proton,[79] and the opening and closing of the gates formed by side chains[80-81] can contribute to the mechanism of enzymes. Therefore, being able to detect and summarize the frequently occurring local configurations (substructures) of proteins at different resolutions and the relationships between groups of co-occurring substructures would provide valuable insights into protein function.

However, current methods used to analyze MD simulations could neither automatically discover frequent 3D local configurations nor analyze their relationships.[82-87] Clustering of conformations[88-89] categorizes conformations by differences over the overall shape, such as an open and a closed state of a protein, and disregards variations over finer details, such as local configurations. Contact and correlations are time-averaged measurements and cannot describe the time-varying conformation and local configurations in a MD simulation. Therefore, diagonalization of a correlation matrix or clustering of pairwise correlations[90] or contacts[91] cannot categorize the local configurations.

We propose an approach to automatically discover and concisely summarize frequently occurring 3D local configurations (or substructures) in MD simulations of molecular systems. The approach is referred to as Frequent Substructure Clustering (FSC). FSC automatically summarizes the hierarchy of frequent 3D local configurations across all spatial scales, and the relationship between the occurrences of the substructures could be described using correlation analysis, for example. FSC could detect correlation between 3D local configurations of different parts of a protein even if they are distal from each other. For example, if site 1 had shape A whenever site 2 had shape B, then shape A and B would be merged to form a new shape, regardless of the distance between sites 1 and 2. The arrangements of the side chains, secondary structures, and domains could therefore be summarized hierarchically in a multi-scale fashion by sequentially merging the shapes.

Moreover, the hierarchical summary from FSC also leads to a more dynamic and precise definition of structural components of a molecular system. The traditional definition of domains is often biased by a static view of biomolecules and persists throughout the analysis of MD simulations even though the conformation and shape change over time. For example, the definition of domains may be based on experimental crystal structures or sequences of amino acids.[92] Alternatively, clustering the time-averaged strength of interactions between atoms in a MD simulation leads to clusters representing dynamic domains.[93-94] These definitions are optimal for some but not all conformations because the conformations and network of relationships change dynamically over time in MD simulations. Frequent Substructure Clustering can identify frequently occurring local 3D shapes that do not necessarily persist over the entire simulation. The local shapes at a large spatial scale within the hierarchical summary are an automatic definition of domains that are optimal for explaining all major conformations at that spatial scale.

In many cases the mechanism of a protein could first be described through interactions between its domains, and the details of the side chain configurations could be taken into consideration only if necessary. Providing such a multi-scale description allows for determining the relationship, for example correlation, between two configurations of a protein at distinct scales. This multi-scale description is advantageous because the change of a local configuration at a relatively small part of a protein, such as an active site, could have a large effect on the function of the whole protein. Additionally, a multi-scale description of multiple conformations is usually terser than a single-scale description because a multi-scale description avoids repeating the description of the features shared by the conformations and describe the details only if they are found to be important. Discovering the frequent local configurations (substructures) of a system at various resolutions and summarizing the relationships between them provides a multi-scale description of the system. The advantage of developing more rigorous clustering methods and providing multi-scale descriptions of biomolecular systems could augment the use of other methods, such as Markov State Model,[89] to study the function of sub-cellular processes and highlight the role of local configurations in biomolecular systems.

3.2 Computational Methods

3.2.1 Frequent Substructure Clustering

We propose Frequent Substructure Clustering (FSC) to concisely summarize 3D local configurations across all spatial resolutions in an MD simulation as summarized in Figure 3.2.1. The method breaks each conformation during an MD simulation (Figure 3.2.1A) into overlapping small local configurations represented by substructures (Figure 3.2.1B and Figure 3.2.1C). Each substructure is a set of discretized pairwise distances (see below) between selected atoms. Subsequently, the method clusters the substructures using hierarchical clustering (Figure 3.2.1D). Every step of the hierarchical clustering merges two clusters of substructures that co-occur most often to form a larger cluster. Unlike other methods, FSC preserves the 3D shapes of the local configurations because each step of FSC preserves the discretized distances and the pairwise distances between all the atoms of a molecule can define the shape of a molecule.

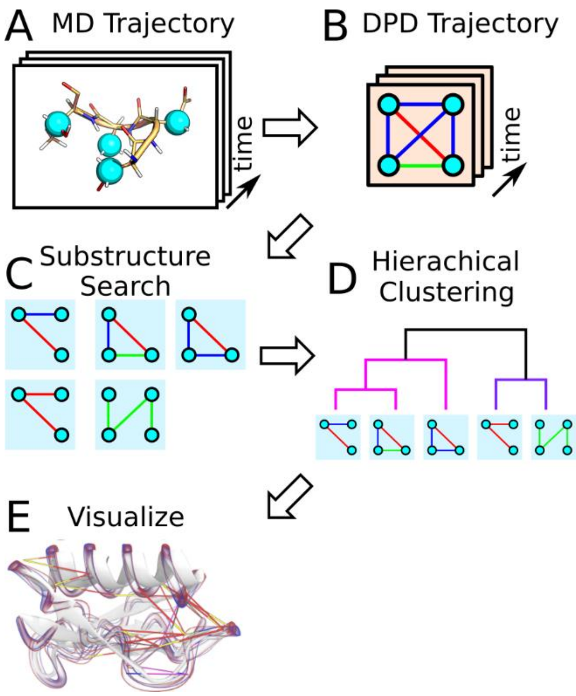

Figure 3.2.1 Steps of frequent substructure clustering. (A) Trajectory from a molecular dynamics simulation. Blue spheres represent C_β atoms. (B) Discretized pairwise distance (DPD) trajectory between C_β atoms. Red, blue, and green lines represent discretized distances. Cyan circles represent atoms. (C) Combinatorial search for substructures. Each substructure represents a local configuration in terms of discretized distances. (D) Hierarchical clustering. Magenta and purple lines represent two clusters of substructures. (E) Visualization of a cluster of substructures.

3.2.2 Discretization of Pairwise Distances

The positions of the C_β atoms were extracted from each frame of the MD simulation. For each pair of C_β atoms, the distances between the pair were discretized into five bins with equal counts (Figure 2). The dynamic range of the histogram was from the minimal distance to the maximal distance. Notice that for each pair of atoms, the conversion was different because the histograms in Figure 2C are different.

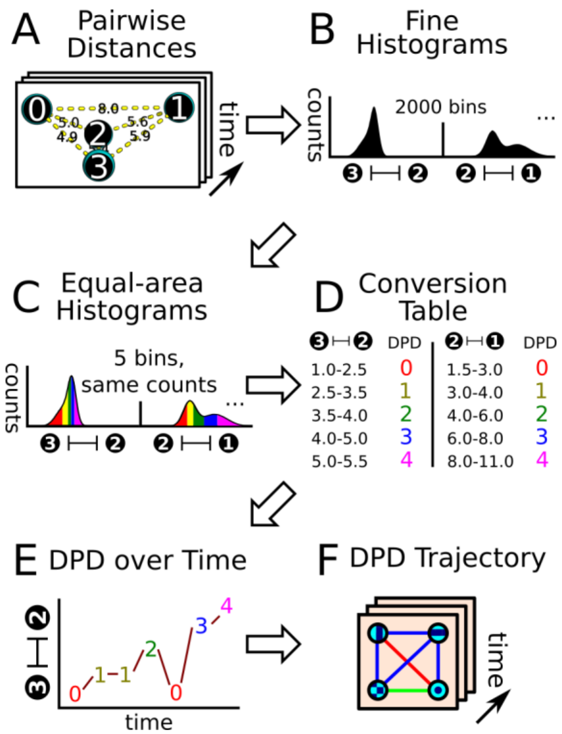

Figure 3.2.2 Discretization of pairwise distances. (A) Pairwise distances between atoms in each frame of a molecular dynamics (MD) simulation. (B) Histogram with 2000 bins for distances between each pair of atoms. Two histograms are shown. (C) Lump the bins of each histogram in B to 5 final bins with equal counts. (D) The histogram in C represents a table to convert pairwise distance to discretized pairwise distance (DPD). (E) The plot of discretized distance between a pair of atoms over time. (F) A trajectory of DPD over time. Circles represent atoms. The coloring of lines represents values of the discretized distance.

3.2.3 Definition and Occurrence of Substructures

The definition and occurrence of substructures in the simulation trajectory are summarized in Figure 3.2.3. The simulation trajectory was transformed into a Discretized Pairwise Distance (DPD) trajectory (Figure 3.2.2), with each frame representing a DPD graph (Figure 3.2.2F and Figure 3.2.3A). Each snapshot of the trajectory represents a graph that is made of atoms as the nodes and the discretized distances between pair of atoms as the edges. Frequent substructures are sub-graphs of specific values of discretized pairwise distances with specific atom pairs that occur frequently throughout the simulation or DPD trajectory (Figure

3.2.3B). A frequent substructure is identified if it occurs in more than 20% of the trajectory and is not a subset of anther substructure.

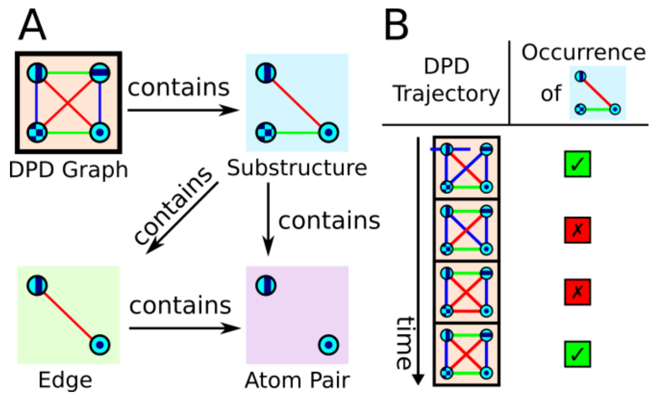

Figure 3.2.3 Example of discretized pairwise distance graph, substructure, edge, atom pair, containment, and occurrence. (A) Circles represent atoms. Arrows indicate relationships of containment. Colors of lines represent values of discretized distances between the atoms. (B) Example of occurrence of a substructure.

3.2.4 Combinatorial Search for Frequent Substructures

The distances between pairs of atoms were discretized and converted to occurrence (over the trajectory) vectors of 5 edges corresponding to 5 bins of discretization (Figure *3.2.2*). The n^{th} element of the occurrence vector of an edge is 1 if the edge occurs in the n^{th} frame of the MD simulation (Figure 3.2.4). Otherwise, the n^{th} element is 0. The support of an edge is the number of MD frames containing the edge, which is the sum of elements of the occurrence vector (Figures 4A and 4B). The occurrence vector of a substructure is the element-wise product of occurrence vectors of all edges in the substructure (Figure 3.2.4C).

The CloseGraph[95] algorithm (Figure S7) was adapted to search for all frequent substructures. The algorithm performed a depth-first search over the search tree for each starting edge. In the search tree, each node contains a substructure and occurrence vector of the substructure. Each child of a node contains an extension of the substructure in the parent. Each extension adds a new edge lower than all existing edges according to a canonical ordering (Figure 3.5.4) of edges (Figure 3.2.4D). Initially, the search tree is pruned to remove all substructures with a support that is less than a defined threshold of 20% (Figure 3.2.4E); that is, a substructure should occur in at least 20% of the simulation frames to be considered and the substructure should not be a subset of another frequent substructure. The substructure search was performed on C_β atoms of GB3 with a 20% support threshold and five final bins of discretized distance as discussed above to limit the number of substructures and to account for the memory requirements of the clustering steps. The occurrence vectors of all frequent substructures found were saved as a sparse matrix.

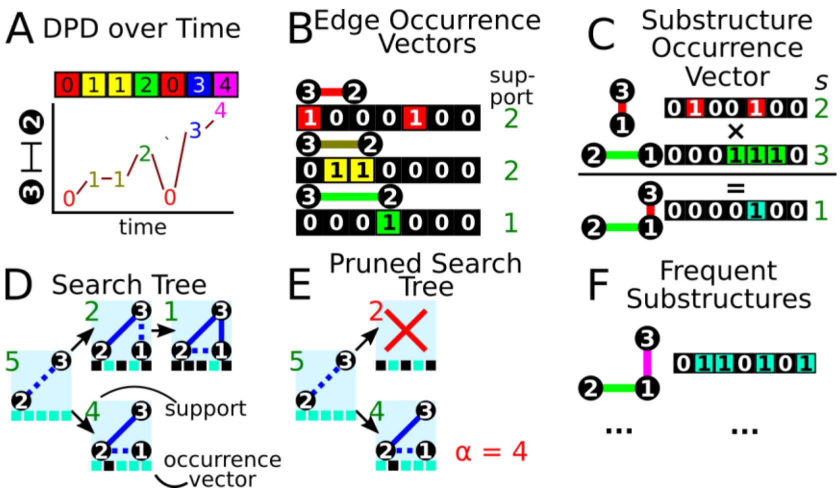

Figure 3.2.4 Combinatorial search for frequent substructures. (A) Both the plot and the colored squares represent a discretized distance between each pair of atoms over time. The colored squares represent a vector of values. (B) Occurrence vectors for three of the five edges between a pair of atoms. (C) Calculate occurrence vector of a substructure from occurrence vectors of edges. (D) Search tree of substructures starting at the edge (3, 2). Each arrow connects a node to its children. Only substructures consisting of blue edges are shown. The dashed lines represent the lowest edge in a substructure. (E) Pruning the search tree to remove all substructures with a support threshold that is less than four. (F) Frequent substructures and their occurrence vectors.

3.2.5 Hierarchical Clustering of Frequent Substructures

The *k*-means clustering (Figure 3.5.6) of 51,022 occurrence vectors of frequent substructures produced 500 rough clusters (Figure 3.2.5B). The *k*-means clustering used *k*-means++ initialization to enhance quality of the clustering (Figure 3.5.4). The *k*-means clustering was used to reduce the data to be clustered using hierarchical clustering by about 100-fold because hierarchical clustering requires more time and memory than *k*-means clustering.[96-97] The centroids from *k*-means clustering were then clustered using hierarchical clustering with Pearson distance and Ward's linkage.[98-99] Pearson distance is 1 minus Pearson correlation. The dendrogram from the hierarchical clustering was cut to form six intermediate clusters (Figure 3.2.5C). The supplementary information contains the description of the hierarchical clustering.

Each of the six intermediate clusters from hierarchical clustering contained multiple centroids from *k*-means clustering. Each *k*-means centroid in the intermediate clusters was replaced by occurrence vectors in the corresponding *k*-means cluster. This replacement converted the intermediate clusters of centroids to the final clusters of occurrence vectors of substructures (Figure 3.2.5D). Since each occurrence vector corresponded to a substructure, these final clusters were also clusters of substructures (Figure 3.2.5E). Clusters in the following description of FSC refer to these final clusters.

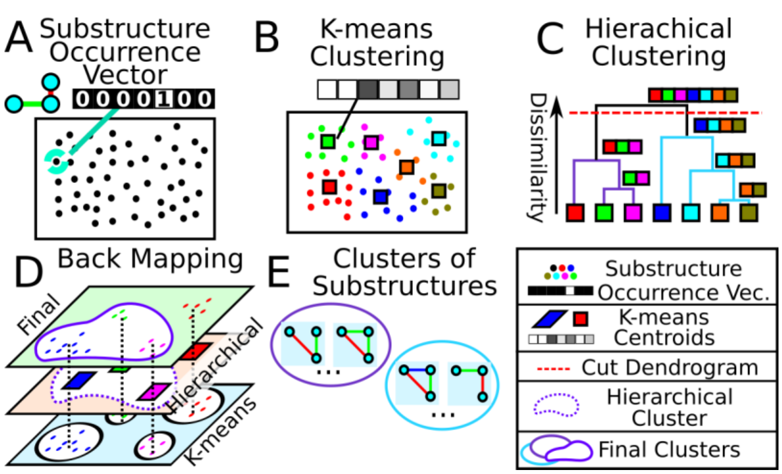

Figure 3.2.5 Clustering of substructures. (A) Occurrence vectors of each frequent substructure is a vector in an n-dimensional space. (B) k-means clustering of the points in A. Colors represent the clusters. The grayscale array is the vector representation of a centroid. (C) Hierarchical clustering of the centroids in B. Each branching point of the dendrogram represents merging two clusters to form a new cluster. The cut produces two clusters represented by purple and cyan lines. (D) Converting a cluster of centroids in C to a final cluster of substructures. (E) Two final clusters of substructures.

3.2.6 Calculating Dynamic Cluster Attributes

Visualizing a cluster of substructures requires merging the substructures into a single representation. The relationship between a cluster and substructure is static because a cluster always contains the same substructures. However, the membership of a cluster can also be interpreted as changing dynamically. In this interpretation, a cluster dynamically contains a substructure if the substructure occurs in the current simulation frame and the cluster statically contains the substructure. This interpretation leads to the following definitions for visualizing a cluster.

Firstly, dynamic size reflects the degree of presence of a cluster in a simulation frame. It is defined as the sum of sizes of all substructures that both belong to the cluster and occur in the simulation frame (Figure 3.2.6B). Size of a substructure is the number of edges contained in the substructure (Figure 3.2.6A). Secondly, dynamic containment of an edge by a cluster reflects whether the cluster dynamically contains a substructure that contains the edge. This is 1 if the cluster statically has at least one substructure that occurs both in the simulation frame and contains the edge. Otherwise, this is 0 (Figure 3.2.6C). Similarly, dynamic containment of an atom pair by a cluster reflects whether the cluster dynamically contains a substructure that contains an edge between the atom pair (Figure 3.2.6D). Finally, the weighted average importance of an edge to a cluster reflects the importance of an edge to a cluster. It is a weighted average of dynamic containment with dynamic size as weights over all simulation frames (Figure 3.2.6E). The weighted average importance of an atom pair to a cluster is similarly defined.

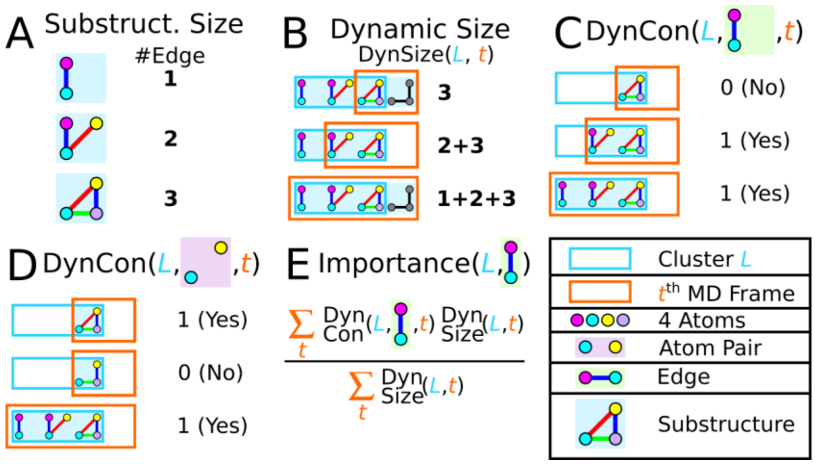

Figure 3.2.6 Dynamic cluster attributes. (A) Size of a substructure. (B) Dynamic size of a cluster of substructures. (C) Venn diagrams illustrating dynamic containment of an edge by a cluster at a simulation frame. (D) Venn diagrams illustrating dynamic containment of an atom pair by a cluster at a simulation frame. (E) Weighted average importance of an edge to a cluster.

3.2.7 Visualization of the Clusters of Substructures

The representative structure of a cluster was defined as the conformation in the simulation frame with the highest dynamic size for the cluster. To visualize the specification of local spatial relationships in each cluster, the weighted average importance of each pair of C_β atoms to the cluster were drawn as cylinders overlaid on the representative structure. The coloring of the cylinder connecting each pair of atoms reflects the ratio between the weighted average importance of the five edges containing the atom pair. To avoid generating an overly complicated diagram, cylinders connecting C_β atoms that were farther than 10 Å apart on the representative structure were not drawn. One overall histogram was calculated for all values of the weighted average importance of each atom pair to each cluster. If the weighted average importance of an atom pair to the cluster is within 20^{th} percentile in the histogram, the cylinder connecting the atom pair was also not drawn.

To visualize the general shape of the overall conformations containing substructures from a cluster, we define reference spline density and weighted spline density (Figure 3.2.7). For each cluster, the weighted spline density represents a weighted average of conformations aligned to the representative structure with the degree of presence of the cluster as weight. The reference spline density represents an unweighted average of conformations aligned to the representative structure. The difference between the reference and weighted spline density reflected a shift of the distribution of the overall conformation upon biasing the local configurations toward substructures in the cluster. The difference between the weighted and reference spline densities instead of the weighted spline density was interpreted to reduce the effect of the representative structure, which was different for each of the six clusters.

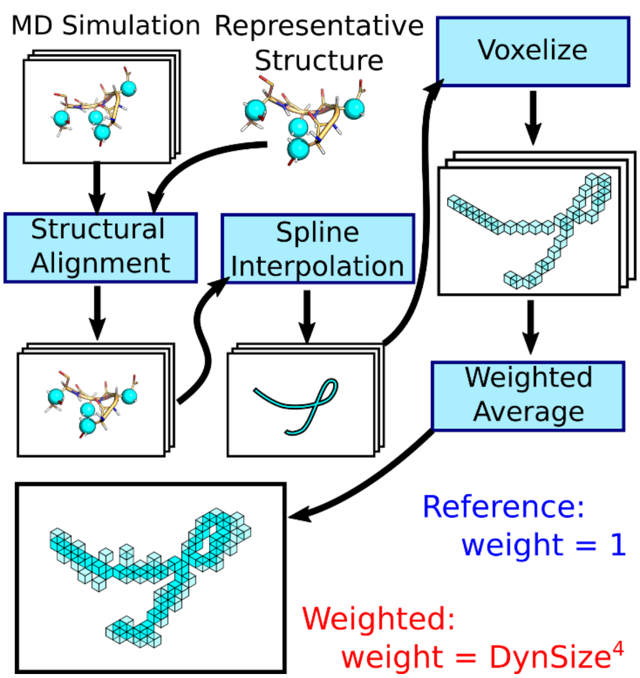

Figure 3.2.7 Calculation of spline density for each cluster L of substructures. (A) Conformation in the current simulation frame. (B) Aligning the green conformation to the brown representative structure of the cluster. (C) A spline fitting of the C_β atom of the aligned conformation. (D) The spline was converted to voxels. Subsequently, a weighted average of the voxel frames was calculated. For calculating the reference spline density, all weights were set to 1. For calculating the weighted spline density, the weights were $DynSize(L, t)^4$, which is the fourth power of the dynamic size of the cluster at simulation frame t.

3.2.8 Software Implementations

All plotting and analysis steps were performed with Python 3 programs written that used the Matplotlib,[100] Numpy,[101] Scipy,[101] and MDAnalysis[102] libraries, and VMD[103] except for the following. The *k*-means clustering and *k*-means++ initialization were performed by C++ program written using OpenMP,[104] Open MPI,[105] and a sparse representation of occurrence vector.

3.2.9 Molecular Dynamics Simulation

The crystal structure of GB3 (PDB id: 1P7F)[106] was solvated in a rhombic dodecahedron box of TIP3P[107] water that extended at least 1.0 nm from the surface of the protein. The energy of this system was minimized by the steepest descent algorithm until the maximum force fell below 100 $kJ \cdot mol^{-1} \cdot nm^{-1}$ with an initial step size of 0.01 nm. Subsequently, a 100 ps heating to 300 K under the NVT ensemble, an equilibration of 10 ns under the NPT ensemble with a Berendsen barostat[108] with 1.0 ps time constant and 1.0 bar reference pressure, an equilibration of 100 ns under the NPT ensemble, and a 760 ns production simulation were performed. Unless otherwise specified, all simulation steps were performed with Gromacs 5.2 with the Amber99SB-ILDN force field,[109] a Parrinello-Rahman barostat[18] with 2 ps time constant set at 1 bar, a modified Berendsen thermostat[110] at 300 K with a 0.1 ps time constant, and a velocity Verlet integrator[111] with 2 fs time steps. The long-range electrostatic was treated by the particle mesh Ewald method[112] with cubic interpolation and 0.16 nm grid for fast Fourier transform. The cutoffs for electrostatics and van der Waals interactions were both 1.0 nm. All bonds of the structure were constrained with fourth order LINCS[113] with one iteration.

3.3 Results and Discussions

We introduce Frequent Substructure Clustering (FSC) to automatically discover and summarize frequently occurring 3D local configurations (local shapes) across all spatial scales for MD simulations of molecular systems. In general, FSC requires more computing power than Principal Component Analysis (PCA), clustering of conformations[88-89], and clustering of pairwise correlations[90] (REF) or contact[91] (REF) because the number of pairwise distances between atoms is less than the number of 3D local configurations, which are defined as combinations of pairwise distances. However, current analysis methods of MD simulations do not automatically discover or summarize 3D shapes of local configurations.

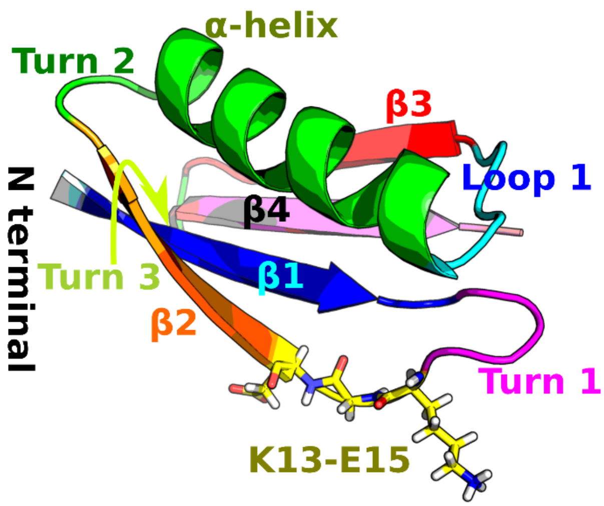

Figure 3.3.1 Naming of various parts of GB3. Turn 3 is between β3 and β4.

Analyzing the molecular dynamics simulation of the GB3 protein (Figure 3.3.1), we show that FSC can find the correlation between shapes at locations that are far apart. The capability of correlating shapes at distal locations is important for understanding allosteric mechanisms. The spline densities of the clusters are different at the same region, showing that FSC can differentiate major local shapes of the same region. For example, the shapes of turn 1 are different across the clusters (Figure 3.3.3). Additionally, the cylinders in cluster 5 specify a short distance between turn 2 and the α-helix (Figure 3.3.3). The cylinders in cluster 5 also specify a K13-E15 region that bends more than the same region in the reference spline densities (Figure 3.3.3). Therefore, a short distance between turn 2 correlates with a K13-E15 region that bends more, which is a feature distal to turn 2. Since FSC cluster substructures by co-occurrence, shape of any substructure in a cluster would correlate with shape of another substructure in the cluster, the general shape of the whole cluster, or any part of the general shape of the cluster.

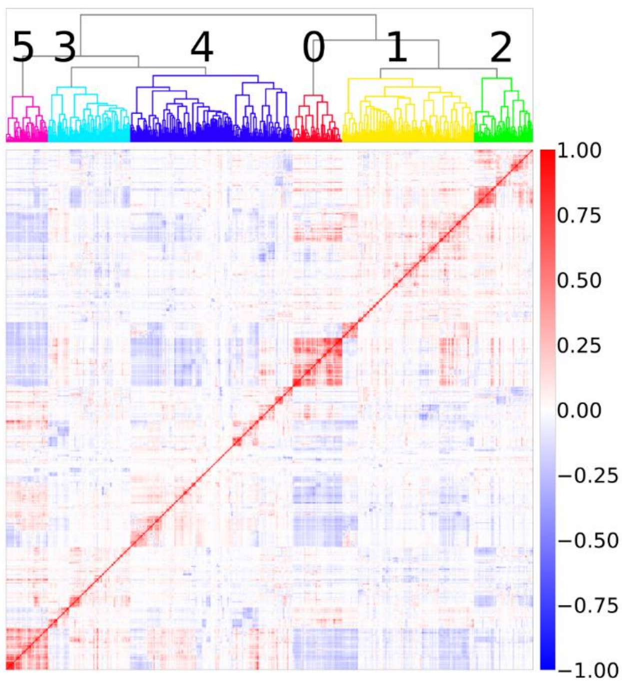

Figure 3.3.2 Dendrogram from hierarchical clustering of the k-means centroid with Pearson distance and Ward's linkage. The k-means centroids are from k-means clustering of the occurrence vector of each frequent substructure formed by C_β atoms from a molecular dynamics simulation of the GB3 protein. The heat map is the Pearson correlation between the k-means centroids. The numeric labels represent six clusters from the hierarchical clustering.

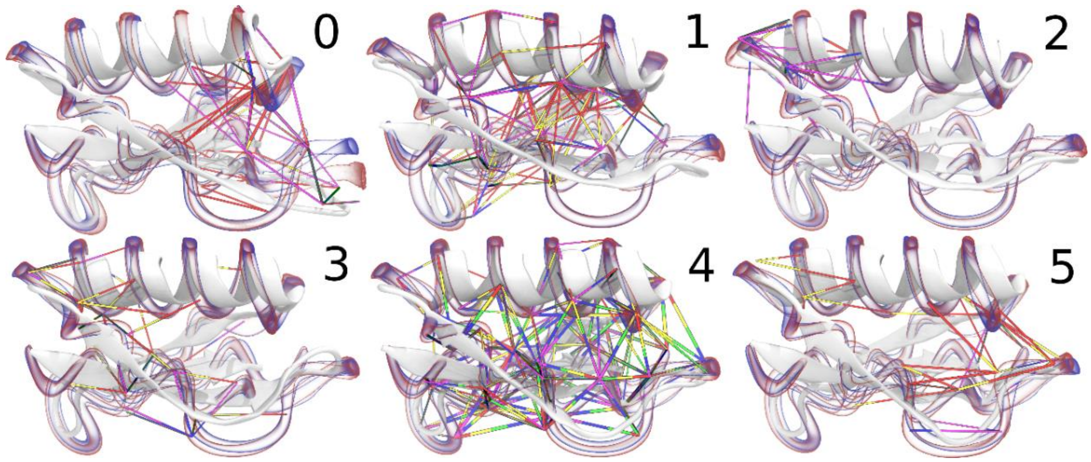

Figure 3.3.3 Spline densities, cylinders, and representative structures of the clusters of substructures. Red and blue surfaces are iso-surfaces of weighted (red) and reference (blue) spline densities of each cluster with 0.01 iso-value. The numeric labels indicate the indices of the clusters. The cross-section area of each cylinder is proportional to the weighted average importance of the atom pair to the cluster. The coloring of each cylinder indicates the ratio between the weighted average importance of the five edges containing the atom pair to the cluster. The five edges correspond to five bins of discretized distance. Red, yellow, green, blue, magenta corresponds to the bins of discretized distances between each atom pair from short to long. Insignificant cylinders are filtered out as described in the Method Section.

Also, the FSC method found two opposing sets of major 3D local configurations of GB3. Each set described a general shape of a large part of GB3. The dendrogram shows the clusters can be divided into two groups (Figure 3.3.2). Group A consists of clusters 0, 1, 2, and group B consists of clusters 3, 4, 5. The correlation matrix shows that clusters in the same group tend to co-occur, while clusters in the two different groups tend to not co-occur (Figure 3.3.2). Clusters in each group localizes at the left, center, and right region of GB3 (Figure 3.3.3). Therefore, local configurations described by clusters in each group together described the shape of a large part of GB3. This shows that the general shape of GB3 can be described in terms of the local shapes represented by the clusters. This result shows that the optimal division of the GB3 protein into 3D components is not static and not obvious from the overall conformation. The two groups represent two ways to divide the GB3 protein into domains. For some conformations, one way is more appropriate than the other way. The boundary, size, and shape of these domains are different between the two groups. The result also shows the rigidity of a region may change as the conformation of the protein changes because FSC clusters substructure by co-occurrence. When a region of a protein consistently has a rigid shape, all substructures describing any part of the shape would belong to the same cluster. On the other hand, if the arrangement of two rigid regions constantly changes, the substructures describing part of the shapes of the two regions would form two clusters. In this case, no cylinder would connect the two regions because the spatial relationship between the two regions is flexible.

The relationship, such as the correlation, between local and global configurations could also be identified by FSC. The weighted spline density of a cluster describes general shape of the whole GB3 when its local configurations are biased toward the local configurations in the cluster. The weighted spline density is compared against the reference spline density to reduce the effect of the representative structure, which is different for each cluster. Compared to the reference, the weighted spline densities of cluster 0 has turns 1 bends more away from the alpha helix and turn 3 bends more toward the beta sheets (Figure 3.3.3). However, the local configurations in cluster 0 describe turn 1 but not turn 3, as shown by the cylinders (Figure 3.3.3). Therefore, the comparison shows that the shape of the local configuration near turn 1 in cluster 0 correlates with conformations in which turn 3 bends more toward the beta sheet. Similarly, the shape of the local configuration near turn 2 in cluster 5 correlates with conformations in which turn 3 bends more away from the beta sheet, as shown by clusters 3 and 5 (Figure 3.3.3). Correlation between the different components of a protein at different spatial scales is important in understanding protein function. For example, the active site and allosteric site are small relative to the rest of the protein. Therefore, the causal relationship between the two sites may potentially involve propagating signals across multiple spatial scales. That is, the binding of a ligand to an active site could perturb the local arrangement of the protein residues that could lead to distal functional relevant conformational changes.[114] The results suggest that the correlation analysis of the substructures from FSC can provide 3D relationships between different spatial scales of a protein. The spatial scales could include side chains, secondary structures, and domains in a protein. While this result correlates the occurrences of conformation and the clusters, correlations between local configurations across other spatial scales can be similarly analyzed because the hierarchical clustering in FSC provides a summary of local configurations across all spatial scales.

In general, FSC provides a multi-resolution summary of 3D local configurations. For example, the spatial resolutions of the six clusters are different. Each cluster is a representation of local shapes of the protein. Cluster 4 has a larger uncertainty in local shape and a lower spatial resolution, since most of the cylinders in cluster 4 have three to four colors (Figure 3.3.3). Unlike cluster 4, most of the cylinders in cluster 1 only have one to two colors (Figure 3.3.3). Therefore, cluster 1 has a higher spatial resolution than cluster 4. The difference in resolutions of the clusters is limited by the number of bins during discretization of the pairwise distances.

Multi-resolution is a great strategy for concisely representing a system at a high resolution. For example, a multi-resolution force field may represent a molecular system as a sum of coarser interactions and finer details. Since the finer details are more local, this representation may limit the scope and number of non-local interactions in the force field. However, Coarse-grained (CG) force fields for simulating biological macromolecules are generally single-resolution instead of adaptive multi-resolution. Coarse-grained force-fields often describe molecular systems at a lower resolution with beads that represents multiple atoms. Force-matching,[115-116] reverse Monte Carlo,[117] and iterative Boltzmann inversion[118] can create a

CG force-field from an all-atom (AA) force field. Alternatively, a CG forcefield could be created with manually defined beads and parametrization against AA simulations.[119] The spatial resolution of the CG force field is the size of the beads. Other possibility is to allow each bead to have multiple states that map to several configurations of the atoms represented by the beads.[120] Application of adaptive resolution scheme seems currently limited to solvent and switching between two resolutions.[121] Since FSC discovers frequent local configurations across all spatial scales from MD simulations, the occurrence of these local configurations may serve as a basis for describing a coarse-grained force field with adaptive multi-resolution for macromolecules.

3.4 Conclusion

Analysis of three-dimensional (3D) local configurations and the relationships between them have implications for understanding the function of biological systems. However, current methods for analyzing molecular dynamics (MD) simulations do not effectively extract, categorize, and analyze 3D local configurations. We presented frequent substructure clustering (FSC), which is a method to automatically discover and summarize 3D local configurations across all spatial scales in MD simulations. The summary consists of hierarchical clustering of overlapping 3D local configurations in terms of discretized pairwise distances. Analyzing the MD simulation of the C_β atoms of the GB3 protein, FSC found two opposing groups of three clusters of frequent 3D local configurations. FSC described the correlations between local and global shapes, including shapes of regions that are far apart from each other. The results suggest that analyzing MD simulations through 3D local configurations may be worthy of further investigation.

3.5 Supplementary Information

3.5.1 Prefix Tree

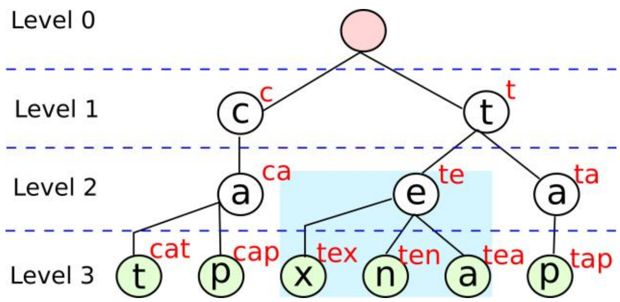

Figure 3.5.1 A prefix-tree. Each circle represents a node. Black letter in each circle represents the actual value of the node. The red letters represent the word represented by each node. The red node is the root of the tree. Each green node is a leaf node. “e” is parent of “x”, which is child of “e.” The blue shaded area indicates a subtree of the whole tree. This subtree is a tree consisted of the node “e” and all its descendants.

A tree is a hierarchical organization of nodes. In a tree, node *A* at a level may connect to node *B* at next level. In this case node *A* is parent of node *B*, and node *B* is a child of node *A*. Each node in a tree can only have one parent. Each node in a tree can have 0 or multiple children. Two nodes are siblings if they have the same parent. The node with no parent is root of a tree. A node with no child is a leaf node. The level of a node is the number of steps from the node to the root. Within a tree, a subtree consists of a node and all its descendants. A search tree is a tree of objects to be searched.

In a prefix tree, each node *C* represents the word formed by concatenating characters of the nodes along a path from the root to node *C*.

3.5.2 Depth First Traversal of a Tree

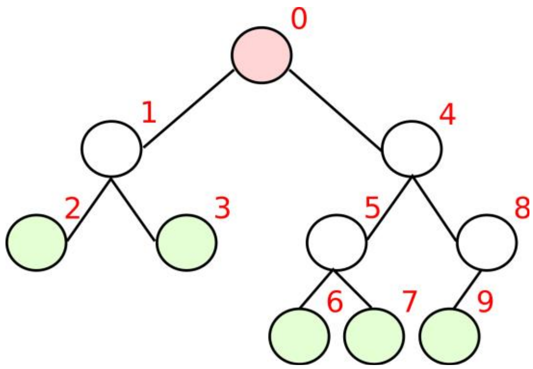

Figure 3.5.2 Depth first traversal of a binary tree. Each circle is a node. The red node is the root of the tree. Each green node is a leaf node. The red numbers indicate the order of visiting the nodes in a left-to-right depth first traversal of the tree.

Traversing a tree is to visit each node of a tree for exactly once. Depth first traversal of a tree visits the nodes by depth first (Figure 3.5.2). Depth first search is a depth first traversal of a search tree.

3.5.3 Graph

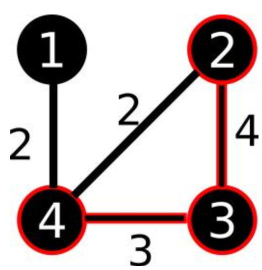

Figure 3.5.3 Graph and subgraph. Each circle represents a node while each line represents an edge. White numbers are labeling of nodes. Black numbers are labeling of edges. The highlighted part is a subgraph of the whole graph, which is a supergraph of the highlighted part.

A graph *G* is a set *V* of nodes and a set *E* of edges between the nodes. Both vertex and edge can have labels. Edges of an undirected graph has no direction. A subgraph *S* of graph *G* is a set *V*’ of nodes and a set *E*’ of edges between the nodes in *V*’, where *V*’ is a subset of *V* and *E*’ is a subset of *E*. If *S* is a subgraph of *G*, *G* is a supergraph of *S*.

3.5.4 Closed Frequent Subgraph

A database *D* contains a set of graphs. The support of a graph *S* in the database *D* is the number of graphs in *D* that are supergraphs of *S*. A graph *S* is a frequent subgraph of database *D* if its support is above a support threshold α.

The subgraphs of a database follows the apriori property[122], which states if graph *A* is subgraph of *B*, support of *A* in database *D* is greater than or equal to support of *B* in database *D*.

A closed frequent subgraph *C* of database *D* is a frequent subgraph of *D* and *C* is not a subgraph of any frequent subgraph of database *D*.

The set of all closed frequent subgraphs of database *D* is far smaller than the set of all frequent subgraphs of database *D*. Since each frequent subgraph is a subgraph of a closed frequent subgraph, the set of all closed frequent subgraph is a very concise representation of the set of all frequent subgraphs.

3.5.5 DFS code of a graph

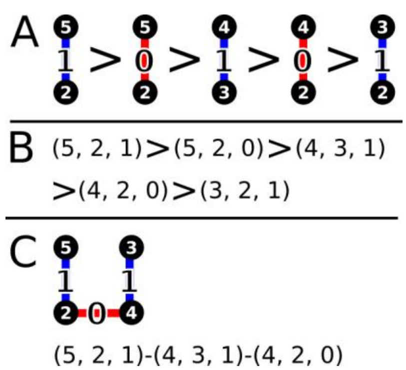

Figure 3.5.4 DFS code of a graph. (A) Each black circle represents a node. Each line represents an edge. White numbers represent labeling of nodes. Black numbers represent labeling of edges. The edges are sorted first by the larger label of node in each edge, then by the smaller label of node, and finally by the label of edge. (B) tuple representation of edges. (C) DFS code of a graph.

Consider a graph G where both nodes and edge are labeled by numbers. Each edge *e* connecting nodes n_1 and n_2 can be represented by a tuple (a, b, c) where a and b are labels of n_1

and n_2, $a \geq b$, and c is the label of e. We can define a comparison of two edges over the tuple representation. For edge $e_1 = (a_1, b_1, c_1)$ and edge $e_2 = (a_2, b_2, c_2)$, if $a_1 > a_2$, then $e_1 > e_2$. If $a_1 = a_2$ and $b_1 > b_2$, then $e_1 > e_2$. If $a_1 = a_2$, $b_1 = b_2$ and $c_1 > c_2$, then $e_1 > e_2$. When a list of edges of the graph G is sorted from large to small according to the comparison, the list is the DFS code of G.[95]

The lexicographical order[95] of two DFS code is like alphabetical ordering of two words. A DFS code A is a word if we treat each edge in A as a letter. For example, (5, 4, 2) > (5, 4, 2)-(4, 3, 3) > (3, 4, 0)-(2, 1, 4).

The DFS code used in the original CloseGraph[95] algorithm uses a tuple of 5 numbers to represent an edge. Since the elemental type of atoms do not change, we used a tuple of 3 numbers to represent an edge.

3.5.6 Close Graph

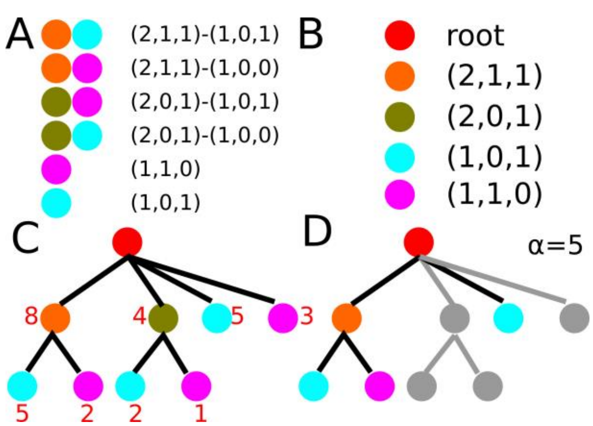

Figure 3.5.5 Prefix tree of DFS code of all graphs of a hypothetical database D. (A) DFS code of all graphs in the database. (B) Representing each edge by filled circle or by tuple. (C) Prefix tree of the DFS codes in A. Red number indicates hypothetical support of the graph S with a DFS code represented by the node. (D) Pruning of the search tree with a support threshold of 5. The gray nodes are removed. All nodes with a support less than 5 and their descendants are removed.

We can use a prefix tree of DFS codes of all graphs in the database as a search tree (Figure 3.5.5C). The apriori property means all supergraphs of an infrequent subgraph are infrequent subgraph. Therefore, we can prune the search tree by getting rid of any node representing an infrequent subgraph, along with all descendants of the node (Figure 3.5.5D).

The CloseGraph[95] algorithm is a depth first search of the pruned search tree. The leaf nodes of the search tree represent closed frequent subgraphs.

3.5.7 Description of K-means clustering

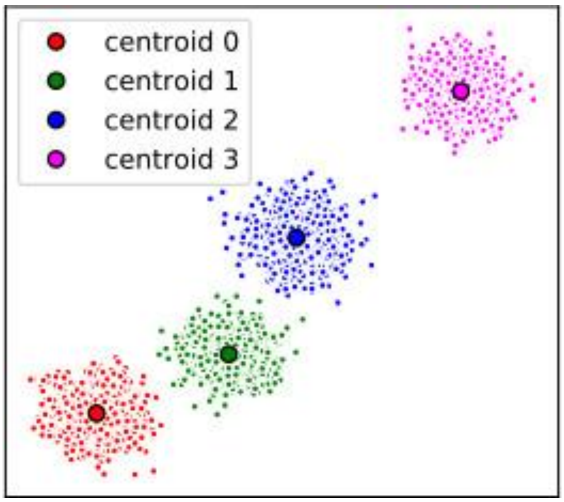

Figure 3.5.6 A k-means clustering of data points to 4 clusters. The initial centroids are chosen by k-means++ initialization. All data points belonging to the cluster are colored to the same color as the centroid of the cluster.

K-means clustering is a method to divide data points into *k* clusters by minimizing the sum of squared errors (SSE).[123] For each cluster, the SSE is the sum of squared Euclidean distance from each data point in the cluster to the average of all data points of the cluster. SSE represents the compactness of a cluster. The data points can be thought of as sand particles forming wells in a sand table. Initially, we throw *k* marbles to the sand table. Each marble represents the current estimation of the centroids of the *k* clusters. At the end, the marbles roll to the bottom of the wells. Assigning each data point to the closest marble would divide the data points into *k* clusters.

For minimizing SSE, we can use Lloyd's algorithm. Firstly, for each data point (sand particle), we assign the data point to the closest estimation of centroid (position of the marble) according to Euclidean distance. This would form *k* groups of data points. Then we calculate the average of each group of data points and treat that as the new estimation of centroid (new position of the marble). These two steps are repeated until the centroids do not change significantly.

The distance measure for *k*-means clustering must be Euclidean distance because Euclidean distance can guarantee convergence of the Lloyd's algorithm.[124] The SSE is guarantee to consistently reduce in each steps of the algorithm until this reaches a local minimum. Many similarity measures, such as Pearson correlation, do not have this guarantee.[125]

Description of K-means++ initialization

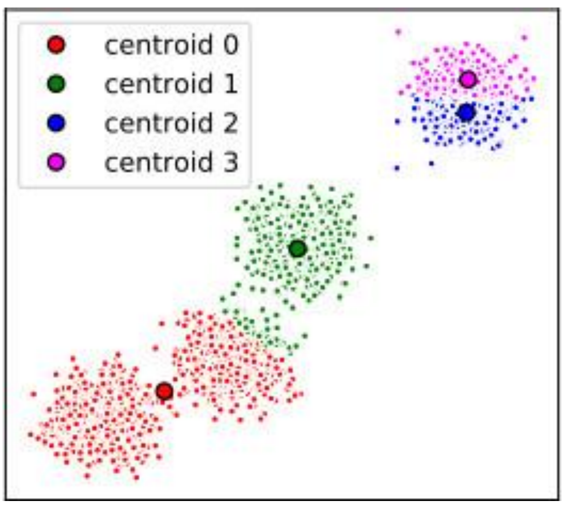

Figure 3.5.7 A sub-optimal k-means clustering of data points to 4 clusters. The initial centroids are randomly and independently chosen from the data points. All data points have equal chance to be chosen as the initial centroid. All data points belonging to the cluster are colored to the same color as the centroid of the cluster.

Suppose initially the sand particles form 5 wells, and we randomly throw 5 marbles. Unluckily, two marbles are thrown into the same well. The sand particles of the well with two marbles would be divided between the two marbles (Figure 3.5.7). This can happen because k-means only guarantee to find centroids that locally instead of globally minimize the SSE. K-means++ initialization tries to avoid this situation by sequentially choosing the initial centroids from the data points. Each choice of an initial centroid is biased to favor data points that are far away from existing centroids.[126]

3.5.8 Description of Hierarchical Clustering and Ward's linkage

A hierarchical clustering of data points would group the data points into a hierarchy of clusters with increasing size. Each data point is initially treated as a cluster. Agglomerative clustering is a subset of hierarchical clustering. In each step of agglomerative clustering, the two least dissimilar clusters are merged.

The measurement for similarity depends on a linkage function and a metric.

Ward's linkage would join the two clusters that has the least increase of variance upon joining. Scipy's implementation of hierarchical clustering maintains a dissimilarity matrix between the current clusters at each step.[127] The (u, v)th entry of the new dissimilarity matrix after joining clusters *s* and *t* into cluster *u* is[127]:

$$d(u,v)^2 = \frac{|v|+|s|}{T} d(v,s)^2 + \frac{|v|+|t|}{T} d(v,t)^2 - \frac{|v|}{T} d(s,t)^2$$

where *v* is each of all current clusters that are not *s* or *t*, $|*|$ denotes cardinality of a cluster, and $T = |v| + |s| + |t|$.

Ward's linkage was chosen because it tends to produce clusters of equal size.[98]

Speed Comparison between K-means and Hierarchical Clustering

Hierarchical clustering is usually much slower than *k*-means clustering. Time complexity for hierarchical clustering with the nearest neighbor chain algorithm is $O(N^2)$.[96] The time complexity for *k*-means clustering is $O(Ndki)$ where N is the number of d-dimensional data points and i is the number of iteration *k*-means required to converge.[128] The actual *d* of each data point is much lower than the number of simulation frame because the occurrences of the substructures were represented as sparse vectors, and the occurrences of each substructure are few and far between over time.

3.5.9 Two Stage Clustering

We performed a two-stages clustering because hierarchical clustering uses much more memory and require more time than *k*-means clustering especially when the number of substructures *N* is large. However, we still want to perform a hierarchical clustering because this allows generating an arbitrary number of representations, instead of just six representations.

To approximate a direct hierarchical clustering over occurrence vectors of the substructures, we use a two stages clustering. The numerous occurrence vector of the substructure was first roughly clustered to a relatively small number of clusters by *k*-means clustering. Subsequently, the centroids from the *k*-means clustering is clustered by hierarchical clustering with Ward's linkage and correlation distance. The 1st stage of rough clustering is not particularly important if it divides the data points into many small clusters.

3.5.10 Rationale for Choosing Pearson Distance in Hierarchical Clustering

Pearson distance is 1 minus Pearson correlation, which is:

$$d(x[t], y[t]) = 1 - \frac{x[t] - \bar{x}}{s_x} \cdot \frac{y[t] - \bar{y}}{s_y}$$

where $x[t]$ and y[t] are time series, $\bar{x}$ is average of $x[t]$, and s_x is standard deviation of $x[t]$.

Pearson distance instead of Euclidean distance was used because Pearson distance can be converted to Pearson correlation to reflect if the data points correlate with each other. The rough clustering by *k*-means may not generate clusters that are similar in frequency to occur or contain similar number of substructures. For example, the substructures in one rough cluster A may tend to occur at the same time more often than the substructures in another rough cluster B. This means the centroid of A has a centroid with a higher time average and a lower standard deviation than the centroid of B. This can happen because the substructures contain different numbers of atoms. A substructure that requires more atoms to be in a particular geometry would be less likely to occur. To normalize these differences between the *k*-means clusters, the hierarchical clustering used Pearson distance instead of Euclidean distance. This would allow merging two rough clusters A and B of substructures when the (dynamic sizes) of A and B have magnitude that are quite different but tend to increase or decrease at the same time.

3.5.11 Choice of Parameters

Each pairwise distance between the atoms was discretized to five instead of more bins because of limitation of computational power and memory. The program for discovering the frequent substructure runs quickly. But the *k*-means clustering program is much slower. Only beta carbons instead of every atom of the protein are analyzed because of the same limitation in computational power and memory. The more atoms, the more combination of atoms, the more frequent substructures, and the more time and memory the k-means clustering program takes. Currently, the k-means program we wrote requires distributing all input data across memory in a cluster of computers before performing k-means++ initialization[126] and k-means clustering[129]. Using a streaming k-means algorithm[130] would reduce the memory requirement and would be more suitable for bigger data.

4 A MULTI-STEP, MULTI-COMPONENT, MULTI-SCALE, AND RATIONAL APPROACH TO EXPLAINING MECHANISMS FROM MOLECULAR DYNAMICS SIMULATIONS

4.1 Introduction

Mechanism of a biological molecule is inherently dynamic, multi-scale, and multi-component. For example, allostery describes a small motion at one part of a protein that can lead to a significant change of the interaction between multiple domains at a much larger scale in hemoglobin.[131] Mechanism of protein aggregation can be described by kinetics of interactions between various residues, sites, domains, and multiple proteins, which are entities at different spatial scales.[132-133] Molecular dynamics (MD) simulation has become a powerful technique that has been routinely used to study the mechanism of biomolecules in subcellular processes. One of the reasons is that with enough computing power to sample the relevant conformational states, an appropriately accurate force-field, and correct simulation method, MD simulations can describe the motions of every atom of proteins. Therefore, we can potentially extract a dynamic, multi-scale, and multi-component mechanism of the residues, sites, domains, and proteins from the simulation. However, since MD simulation provides a large amount of data[134], analysis techniques are necessary to extract such a description of the mechanism from the simulation.

Markov state model (MSM)[135] of conformations is a powerful method to summarize the dynamics of the whole protein. The model assumes the protein is always in one of the several states and describes the probabilities of transitioning from a particular state at the current timestep to every other possible state at a future timestep. The model further assumes the dynamics follows the Markovian assumption, which means that the probability that the protein is in a future state depends only on which state the protein is currently in.[136] The separation between the current and future timesteps is the lag time.[135] For example, suppose a protein has an opened state and a closed state. A MSM of conformations would describe which set of conformations in the simulation belongs to which state. The model would also describe the probability to transit from the current state to the state at 100 ns later, if the lag time is 100 ns. If the current state is the opened state, the model describes the probability to transition to the closed state in 100 ns later.

However, while an MSM of conformations can summarize the dynamics of the whole protein, this model is not suitable for describing the dynamics of the detailed parts of the protein. Knowledge about the dynamics of substructures of the proteins is valuable because some small parts of the protein can have a previously unknown and important role in the mechanism, such as the three amino acid residues that form the catalytic triad in chymotrypsin.[137] Without knowing or assuming which set of residues is important in the mechanism, we may build an MSM of conformations with one state for each combination of major configurations of the amino acid residues to find this set of important residues. Yet, such an MSM is impractical because of a combinatorial explosion of the required number of states. For example, for a peptide with 16 residues and each residue with 2 major configurations, the required number of states to describe the major configurations of the whole protein is $2^{16} = 65536$. That means building an MSM of conformations that has a high enough resolution to describe the dynamics of all residues would require 65536 states, which is intractable. Moreover, the model could describe the combinations of states of all residues, instead of the states of individual residues. Therefore, reducing the number of states is necessary for developing a Markovian model that describes the detailed dynamics of all parts of the protein.

To reduce the number of states required to describe a multi-component dynamics, we can use the coupled MSM model.[138] In addition to the Markovian assumption, the coupled MSM model assumes that the overall system is always in one of several overall states. The overall system consists of several parts and each overall state of the system corresponds to a set of states for all parts of the system. The model describes the overall state, the states of each part, and the probabilities to transit from the current overall state to all possible future overall states. For example, suppose a protein has 3 domains. Each domain is always in one of two states: A state or B state. Then, each overall state is a combination of the states of the 3 domains. For example, the 3rd overall state corresponds to A state of the 1st domain, A state at the 2nd domain, and B state at the 3rd domain. The model would also describe the probability to transit from the 3rd overall state at the current timestep to the 4th overall state at the future timestep. After fitting this model, we can simplify the coupled MSM by extracting a smaller coupled MSM for a subset of domains given the state of another domain or allosteric site. For example, two domains might have highly correlated states and behave as a single domain only when the third domain is in the on state. The model does not assume the dynamics of the domains are always independent. Domain A and B may conditionally have strong interaction only when domain C is in the off state. This is contrary to the approach used in independent Markov decomposition.[139]

To further reduce the required number of states to describe the dynamics of the residues and connect the dynamics of the domains with the dynamics of the residues, we developed a hidden coupled Markov model. In a hidden Markov model,[140] the dynamics of the hidden states of the system is the same as a Markov state model, except the hidden states are not directly observable. For each hidden state, the visible data follows a distinct probability distribution. The conditional probability distributions of the visible data given each of all hidden states are called the output distributions. For example, suppose the weather is not directly observable but the percentage of people who bring an umbrella is observable. The weather has two states: rainy and sunny. Transition between rainy and sunny weather follows a Markov state model. When the weather is rainy, 90% of people use umbrellas. When the weather is sunny, 5% of people use umbrellas. The probability distribution of people that use umbrellas given a sunny weather is one of the two output distributions.

The relationship between the hidden state and the visible data is a dimension reduction. With prior knowledge of the relationship between the state and the visible data, the dimension reduction and the fitting of the Markov state model of the hidden states can be split into two steps. For example, the MD simulation can go through dimension reduction steps that consist of principal component analysis, calculation of dihedral angles, and K-means clustering[135]. After the dimension reduction, we get a trajectory of states and can fit an MSM as a separated step. Since the dimension reduction steps summarizes the conformations, the resulting MSM is a MSM of conformations. This MSM of conformation can be considered as a hidden Markov model because the state of the protein is not directly observable in the simulation. However, with prior knowledge of the relationship between the hidden state and the visible data, we can use an MSM and a known dimension reduction step to achieve the same result as in a hidden Markov model.

A hidden Markov model is more valuable when the relationship between the state and the visible data is unknown. For example, in VAMPnet[141], all output distributions are represented by one artificial neural network (ANN). Same is true for independent Markov decomposition,[139] which is a decomposition of the transition probabilities of the VAMPnet. Since the ANN models a complex relationship between the hidden state and the visible data, we are unlikely to know the relationship between the hidden state and the visible data before fitting the hidden Markov model. Moreover, the artificial neural network (ANN) in VAMPnet is a fully connected multilayer perceptron[142] that consists of nonlinear transformation[143] with numerous parameters, and the weights in the ANN that represents the output distributions have no clear interpretation.[144] Therefore, VAMPnet does not directly describe a relationship between the residues and the hidden states that is interpretable, which is a disadvantage in the analysis of MD simulations.

In our model, the dynamics of the states of the domains are hidden. The overall state of the whole molecule consists of the states of all domains of the molecule. The number of overall states is the number of combinations of the domain states. When each of D domain can have two possible domain states, the number of overall states is 2^D. The transition of this overall state over time follows a MSM. This MSM is the hidden part of the overall model, which is a hidden Markov model. The number of the domains D is a meta-parameter that must be chosen before fitting the model.

We will arbitrarily assume that each protein consists of multiple domains, each domain has two possible domain states called A and B, and each residue has two clusters of configurations called up and down. The A, B, up, and down states do not imply any biological function of the domains or residues. When the system consists of D domains and each domain has 2 domain states, the number of overall states is the number of combinations of the states of all domains, which is 2^D. We believe abstracting each domain and residue into two states is adequate for designing a unified model that is interpretable and relate the states of the residues with the dynamics of the domain. Notice that our model doesn't inherently limit the number of domain states or the number of residue states to two.

The visible part of our hidden coupled MSM consists of one Dirichlet distribution for the whole protein and one dynamic graphical model for each domain state of each domain. The Dirichlet distribution determines which residues belong to which domain. Consequently, fitting the model automatically determines the which residue belong to which domain without relying on any prior knowledge. The dynamic graphical model describes the energies of the configuration of the states of all residues of each domain. Each domain state corresponds to a model of disposition of each residue toward the up or down states and the energies of pairwise interaction between the residues.

The dynamic graphical model is an Ising model or Potts model with half of the spins as the current state of the residues and the other half of the spins as the future state of the residues.[145] The Potts model is same as an Ising model, except each spin has more than two possible states.[146-147] Like the other Ising models, the dynamic graphical model assumes each residue has a spin with either 2 or more possible values. Each configuration of spins has an energy, which is the sum of self-energy and pairwise interaction energy of the spins. The probability distribution of all possible configurations of spins follows the Boltzmann distribution. We choose the dynamic graphical model because it offers a clear interpretation for the relationship between the hidden dynamics of the domains and the visible states of the residues.

For a system with N sites, the dynamic graphical model is an Ising model or Potts model with 2N sites. The model describes the conditional probability distribution of the possible future configuration of spins of a system, given the current configuration of spins of the system. Each site can have 2 or more possible spins, depending on if we are using Ising or Potts model. Half of the sites represents the current state of the system. The remaining half of the sites represents the future state of the system. The time separation between the current and future states is again the lag time. There is no interaction between the sites that represents the state of the system at each point of time. For example, all sites that represents the future state of the system do not interact with each other. Each site that is part of the current state of the system has interaction with all sites that are part of the future state of the system.

We preprocess the trajectory from MD simulation for the dynamic graphical models within the overall model by the following method. We cluster the configuration of each residue into two clusters. And arbitrarily assign the two clusters as spin up or spin down. The correspondence between the cluster and the spin is not part of the model. Instead, the model only describes the relative energies of the two clusters, as modeled by the Ising model. We also arbitrarily divide the configurations of each side chain into two clusters. Dividing the configuration into more clusters and make the model to have more than two kinds of spins for each residue is certainly possible. However, for the purpose of designing a unified model that is interpretable and relate the states of the residues with the dynamics of the domain, we think that abstracting the configurations into two clusters is adequate.

Overall, we propose a unified statistical model that can simultaneously account for the three aspects of mechanism by extending the MSM. The model describes transition of the states of all domains by a coupled hidden Markov model. Fitting the model would automatically assign each residue to one of the domains. The state of each domain is defined by the state of the network of interactions between the residues of the domain. Specifically, each state of each domain is modelled by a dynamic graphical model. The whole model describes contribution of interactions between residues to the dynamics of domain interactions. We test the model by fitting a millisecond MD simulation of Bovine pancreatic trypsin inhibitor (BPTI) obtained from D. E. Shaw Research to the model.[134]

4.2 Method

The original millisecond trajectory of a MD simulation of BPTI was obtained from D. E. Shaw Research.[134] The original trajectory was transformed to a trajectory of states of all residues by clustering the geometries of each residue in all frames of the simulation to two states by K-means clustering.[129] For each residue, the distance used in the K-means clustering is Euclidean distance between corresponding pairwise Euclidean distances between the atoms of the residue in two frames of the simulation. Since some side chains have rotational symmetry, we test all combinations of rotations that does not affect the geometry to find the minimal distance between the geometries in the two frames.

Subsequently, 512 models were initialized with random parameters. For each model, our program proposed a new model by randomly selecting one of the residues and randomly assign the residue to another domain. Subsequently, the program performs 10 steps of stochastic gradient descent (SGD)[148] with a learning rate of 1.0 and over a rough log-likelihood for the proposed models. The proposed models are randomly accepted as the new model when the old likelihood plus a shift and proposed likelihood satisfies the metropolis criteria. The magnitude of the shift was adaptively adjusted to allow 5% of the models to accept the new proposal. After 3 repetitions of the metropolis and SGD steps, the program selects the model with the highest likelihood. The program then calculates a more accurate likelihood for the selected model. If the accurate likelihood of the selected model is better than the old best likelihood, the selected model becomes the current best model. Afterwards, the program repeats the above steps except for the initialization of parameters.

Finally, the program sets the assignment of residues to domains of all models to the assignment of residues to domains of the best model. The set of probabilities that a residue belongs to each of all domains is modelled by a Dirichlet distribution and fitted with metropolis algorithm.[149] Afterwards, the program fit the Ising model[145] layer with iteratively weighted least square implemented in the scipy library. Then, the program performed the same repetition of steps except for the initialization of parameters.

The rough log-likelihood was calculated from the ten segments of 100 frames. Each segment of each model was randomly and independently selected from the trajectory of states of all residues. The more accurate log-likelihood was calculated from 10 segments of 10,000 frames. The lag time was 100 ns. We set the number of domains to three because the correlation map between states of the residues show three blocks along the diagonal. The program was written with python, mdtraj[150], and tensorflow[151] and was executed on GPU.

The probability distribution of the positions of all carbon atoms in the simulation was estimated by the following method. For each residue, we calculate one weighted 3D histogram of the positions of all carbon atoms in the residue over all simulation frames. For the probability distribution weighted for each overall state, the weight is the probability of the frame being in the overall state. For the unweighted probability distribution, the weight is 1. For each overall state, the magnitude of the differences per residue between the distribution of the conformations is the sum over all space of the absolute difference between the probability distribution weighted for the overall state and the unweighted probability distribution.

4.3 Results and Discussion

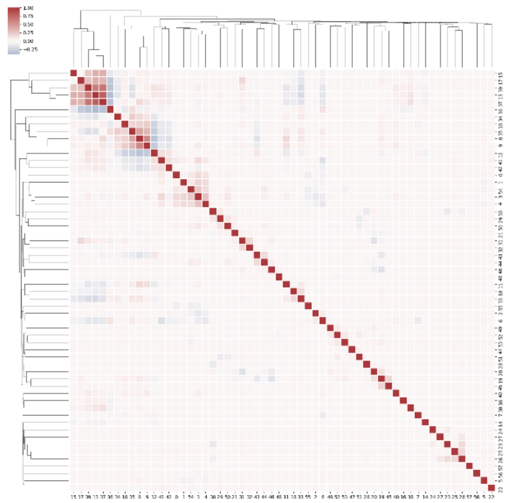

Figure 4.3.1 Correlation matrix between states of the residues of BPTI in the MD simulation. Indices of the columns and rows are residue indices. The rows and columns are rearranged by hierarchical clustering by the seaborn library.

The correlation map contains three blocks along the diagonal (Figure 4.3.1). This shows that the residues can be roughly divided into three groups. Within each group the residues are closely related to each other.

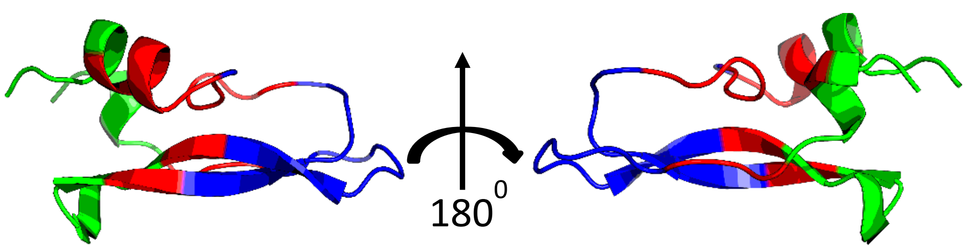

Figure 4.3.2 Two views of the three domains in the model.

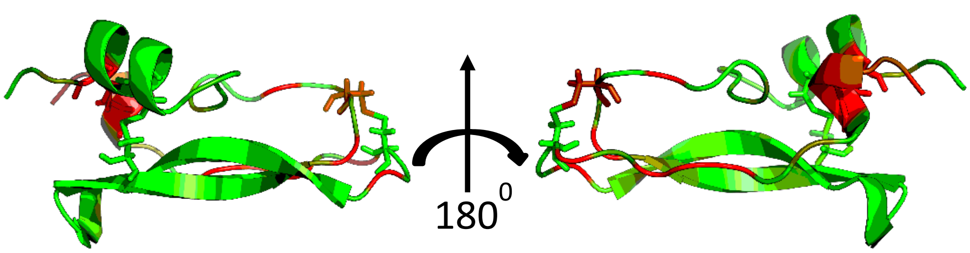

Figure 4.3.3 Two views of the absolute difference between the field strength parameters of the two states of all domains. Red is high and green is low. The stick representation shows the disulfide bonds.

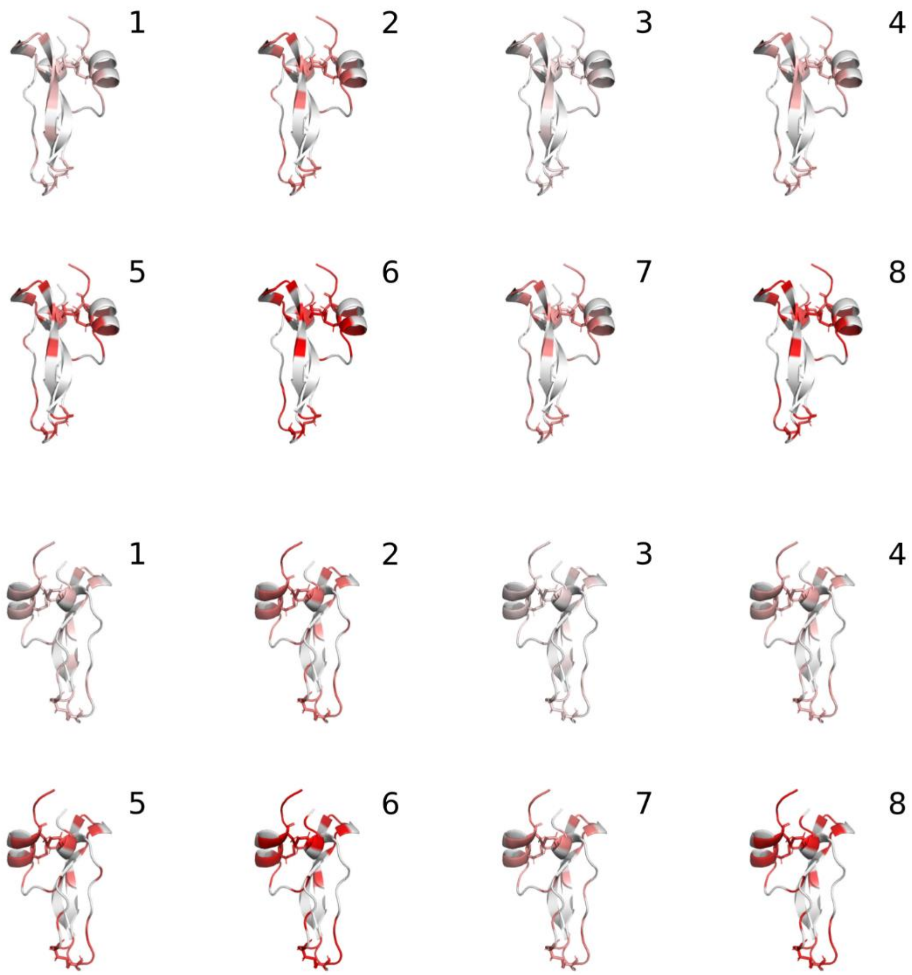

Figure 4.3.4 Magnitude of the differences per residue between the distribution of the conformations in the frames with higher than 90% probability of being in an overall state and the distribution of the conformations in all the frames. All frames were aligned to the crystal structure of BPTI before calculating the differences. The numbering of the subfigures corresponds to the modes. The coloring of each residue reflects the atomic average of the absolute difference between the probability distributions of the positions of the carbon atoms in the selected frames and in all frames. Red is high and white is low. For each overall state, the diagram doesn't color the residues with a difference less than the median of all differences for all residues.

The three domains in the best model are spatially contiguous and the three domains do not mix spatially (Figure 4.3.2). This shows that the coupled Markov model layer describes domain dynamics and the Ising model layer describes residues interactions and there is no mixing of responsibilities. If the coupled Markov layer used the coupling to describe some of the residue interactions, the three domains would mix because the interactions between the residues is local. Localized features differentiate the two states of the domains (Figure *4.3.3*). This shows that the state of each domain is defined by some key regions. Therefore, the domain dynamics correlates to stronger change in these regions. Developing ligands that target these regions may change the domain dynamics.

The major differences in geometry are over the green and blue domains defined in Figure 4.3.2, which include the disulfide bonds, threonine 32, and the loop region near the C14-C38 disulfide bond. Overall states 5 – 8 have a deeper red color than overall state 1 – 4 (Figure 4.3.4). That means when the state of the red domain is B, the distribution of the conformation of the protein has bigger deviation from the distribution of the conformation of the protein over all simulation frames. Comparing with Figure 4.3.3, Figure 4.3.4 has more residues marked as different. This is expected because Figure 4.3.3 shows the difference in the energetics of the residues between the two states of each of all domains while Figure 4.3.4 reflects the difference in geometry. The change of the energetics of one residue can still affect the geometry of a set of neighboring residues when the energies of all possible arrangements of the residues in this set remain the same. Therefore, Figure 4.3.3 shows fewer residues having a significant difference in energetics between the A and B domain states. The energetics in Figure 4.3.3 is a more succinct description of the source of the geometric differences in Figure 4.3.4. Figure 4.3.4 shows the distribution of configurations of threonine 32 (T32) in the protein in each overall state is significantly different from the distribution of the configuration of T32 over the whole simulation. Contrary to Figure 4.3.4, Figure 4.3.3 shows threonine 32 does not have a significant difference of energies in the two states of the domains identified. In fact, substituting threonine 32 by an alanine would only slightly change the configuration of the nearby Y21 and F4 side chains and without any further effect on the crystal structure.[152]

Table 4.3.1 Transition matrix of the coupled hidden Markov model. The self-transition probability is the probability to transit from a mode to the same mode.

Overall State	Domain States Now			Domain States at 100 ns Later			P(Overall Self-Transition)
	Red	Green	Blue	P(Red is B)	P(Green is B)	P(Blue is B)	
1	A	A	A	8.69%	75.22%	9.15%	20.56%
2	A	A	B	6.32%	13.24%	98.46%	80.02%
3	A	B	A	2.13%	93.57%	3.26%	88.59%
4	A	B	B	32.29%	96.04%	93.89%	61.06%
5	B	A	A	62.40%	88.60%	14.00%	6.12%
6	B	A	B	94.64%	86.93%	99.33%	12.29%
7	B	B	A	44.65%	82.72%	17.19%	30.58%
8	B	B	B	92.71%	85.68%	99.41%	78.96%

The coupled transition matrix shows the states of the domains are quite dynamic and the protein have multiple stable overall states. Since a more stable overall state has a higher self-transition probability, overall states 2, 3, and 8 are quite stable. Overall state 4 seems to be a meta-stable transition state between overall states 3, 2 and 8. Overall state 4 is meta-stable because overall state 4 has a self-transition probability of 61.06%. Overall state 8 has 7.29%

chance to transit to overall state 4. Overall state 4 has 3.96% chance to transit to overall state 2 and 6.1% chance to transit to overall state 3 (Table 4.3.1).

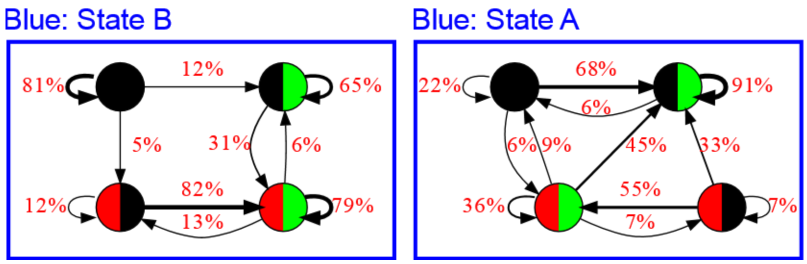

Figure 4.3.5 Conditional transition probabilities between states of the red and green domains, given the state of the blue domain. Red and green color in a circle represents that the red or green domains are in the B state. Black color in a circle represents that the red or green domains are in the A state. Thickness of the arrow and red number represent transition probability. Transition probabilities less than 5% are omitted.

Regardless of the current domain states of the red and green domains, the future domain state of the blue domain tends to be the same as the current domain state of the blue domain (Table 4.3.1). Therefore, the transition of the domain state of the blue domain is approximately independent from the current states of the red and green domains. Consequently, it is reasonable to consider the conditional transition probabilities of the domain states of the red and green domains given the domain state of the blue domain. The domain dynamics of the red and green domains change dramatically as the domain state of the blue domain changes from B to A (Figure *4.3.5*). For example, as the domain state of the blue domain changes from B to A, the self-transition probability of the node that represents the domains states of the red and green domains both being B decreases from 81% to 22% (Figure *4.3.5*). That means the stability decreases significantly. When BPTI binds to trypsin, the loop around the C14-C38 disulfide bond in the blue domain is at the interface between BPTI and trypsin.[153] Since the current domain state of the blue domain is approximately independent from the future domain state of the other domains and the domain dynamics of the red and green domains change dramatically as the domain state of the blue domain changes, the blue domain is a switch that correlates to the dynamics of the red and green domain. The blue domain is a switch despite the geometrical features that distinguishes each overall state from all overall states are over both the green and blue domains (Figure *4.3.4*).

4.4 Conclusion

In conclusion, the coupled transition matrix of the model showed that the blue domain is a switch. While the domain states of the other domain mostly don't affect the domain state of blue domain, the domain state of the blue domain significantly affects the interactions between the red and green domains. The model reveals contribution of residue interactions to dynamics of domain interactions. Domain state of each domain is defined by important localized interactions between residues instead of all the interactions within the domain. Developing drugs that target these key interactions may change the domain dynamics. We have presented a multi-step, multi-component, and multi-scale model that can explain the contribution of residue interactions to coupled domain dynamics over time.

4.5 Supplementary Information

Table 4.5.1 Table of Symbols

Symbol	**Meaning**
$(\vec{a})_i$	The i^{th} element of the vector $\vec{a}$.
T	The length of the trajectory of observation.
L	The number of hidden states.
O_t	Random variable for the observation at time t.
o_t	The value of the observation at time t.

$P(O_{2:4} = o_{2:4})$	The probability that the random variables for the observation at time $t = 2, 3, 4$ equal to the actual value of the observation at $t = 2, 3, 4$. In other words, the symbol means $P(O_2 = o_2, O_3 = o_3, O_4 = o_4)$.
$P(O_{2:4})$	Abbreviation for $P(O_{2:4} = o_{2:4})$.
H_t	Random variable for the hidden state at time t.
$\vec{\eta}$	A row vector encoding the initial probability distribution of the hidden state at time $t = 1$. Specifically, $(\vec{\eta})_m = P(H_1 = m)$.
$\vec{x}_t$	A row vector encoding the probability distribution of the hidden state at time t. Specifically, $(\vec{x})_m = P(H_t = m)$.
$\vec{z}_t$	A row vector encoding the probability distribution of the observation at time t. Specifically, $(\vec{z})_k = P(O_t = k)$.
A_{lm}	A $L \times L$ matrix. The element at the l^{th} row and m^{th} column of the transition matrix A. $A_{lm} = P(H_t = m \vert H_{t-1} = l)$ for all time t.
B_{mk}	The emission probabilities. $B_{mk} = P(O_t = k \vert H_t = m)$
Φ_t	A $L \times L$ diagonal matrix. The diagonal elements are the probabilities for the model to generate the current observation for each of the possible hidden states. $(\Phi_t)_{mm} = P(O_t = o_t \vert H_t = m)$
Θ	Random variables for all parameters of the model

θ	Values of all parameters of the model
$\ln[P(O_{2:4} = o_{2:4} \vert \Theta = \theta)]$	Log-likelihood for the current values of the parameters
$\ln[P(O_{2:4} \vert \Theta)]$	Abbreviation of the log-likelihood expression

We will assume all probabilities are conditional over the current values of all parameters of the model. We may *omit* the condition of $\Theta = \theta$ in the following discussion of the probabilities.

The following equations describe the evolution of the probability distribution of the hidden state over time in a hidden Markov model.

$$\begin{cases} \vec{x}_{t-1}A = \vec{x}_t \\ \vec{x}_t B = \vec{z}_t \end{cases}$$

(Equation 4.1)

The probabilities of all possibilities sum to one. Therefore, we have:

$$\begin{cases} \sum_{m=1}^{L} (\vec{\eta})_m = \sum_{m=1}^{L} P(H_1 = m) = 1 \\ \forall l: \sum_{m=1}^{L} A_{lm} = \sum_{m=1}^{L} P(H_t = m | H_{t-1} = l) = 1 \\ \forall m: \sum_{k=1}^{K} B_{mk} = \sum_{k=1}^{K} P(O_t = k | H_t = m) = 1 \end{cases}$$

(Equation 4.2)

We can treat the actual trajectory of observation as a sample of all possible trajectories of observations that the model can generate. The likelihood for the current values of all parameters θ is the probability that the model generates the trajectory of observations, which is:

$$P(O_{1:T} = o_{1:T} | \Theta = \theta) = P(O_{1:T} | \Theta) = P(O_{1:T})$$

(Equation 4.3)

The forward-backward algorithm[154] is a method for calculating the log-likelihood and the trajectory of the probability distributions of hidden states over time, given the values of the parameters, the observation, and the emission probabilities. The algorithm defines the forward probabilities of a hidden Markov model as:

$$(\vec{\alpha}_t)_m = P(H_t = m, O_{1:t} = o_{1:t} | \Theta = \theta) = P(H_t = m, O_{1:t} | \Theta) = P(H_t = m, O_{1:t})$$

(Equation 4.4)

Then we can calculate the likelihood and forward probabilities by the following equations:

$$P(O_{1:T}) = \vec{\eta} \left[\prod_{\tau=1}^{T} (\Phi_\tau A) \right] \vec{1}^{\intercal}$$

(Equation 4.5)

$$\vec{\alpha}_t = \begin{cases} \vec{\eta}\Phi_1 & \text{for } t = 1 \\ \vec{\eta} \left[\prod_{\tau=1}^{t-1} (\Phi_\tau A) \right] \Phi_t & \text{for } t > 1 \end{cases}$$

(Equation 4.6)

The probability for the model to generate a trajectory of observation that exactly matches the actual trajectory of observation decreases as the length of the trajectory increases. Therefore, the forward probabilities would become vanishing small as t increases. Consequently, the definition of the forward probabilities is not numerically stable.

We can rewrite the log-likelihood of the model as:

$$P(O_{1:T}) = P(O_1)P(O_2|O_1)P(O_3|O_{1:2})P(O_4|O_{1:3}) \cdots P\left(O_T \middle| O_{1:(T-1)}\right) = \prod_{\tau=1}^{T} c_\tau$$

(Equation 4.7)

where

$$c_\tau = \begin{cases} P(O_1) & \text{for } \tau = 1 \\ P\left(O_\tau \middle| O_{1:(\tau-1)}\right) & \text{for } \tau > 1 \end{cases}$$

(Equation 4.8)

Then the log-likelihood is:

$$\ln[P(O_{1:T})] = \sum_{\tau=1}^{T} c_\tau$$

(Equation 4.9)
Define the numerically stable forward probabilities as:

$$\left(\hat{\vec{\alpha}}_t\right)_m = P(H_t = m|O_{1:t}) = \frac{P(H_t = m, O_{1:t})}{P(O_{1:t})} = \frac{(\vec{\alpha}_t)_m}{\prod_{\tau=1}^{t} c_\tau}$$

(Equation 4.10)
Then we have:

$$\hat{\vec{\alpha}}_t = \frac{\vec{\alpha}_t}{\prod_{\tau=1}^{t} c_\tau}$$

$$= \begin{cases} \dfrac{\vec{\eta}\Phi_1}{c_1} & \text{for } t = 1 \\ \dfrac{\vec{\eta}[\prod_{\tau=1}^{t-1}(\Phi_\tau A)]\Phi_t}{\prod_{\tau=1}^{t} c_\tau} & \text{for } t > 1 \end{cases}$$

$$= \begin{cases} \vec{\eta}\widetilde{\Phi}_1 & \text{for } t = 1 \\ \vec{\eta}\left[\prod_{\tau=1}^{t-1}\left(\widetilde{\Phi}_\tau A\right)\right]\widetilde{\Phi}_t & \text{for } t > 1 \end{cases}$$

(Equation 4.11)

where

$$\widetilde{\Phi}_t = \frac{\Phi_t}{c_t}$$

(Equation 4.12)

Notice that

$$\hat{\hat{\alpha}}_t \vec{1}^\intercal = \sum_{m=1}^{L} \left(\hat{\hat{\alpha}}_t\right)_m = \sum_{m=1}^{L} P(H_t = m | O_{1:t}) = 1$$

(Equation 4.13)

Then we have:

$$1 = \sum_{m=1}^{L} \left(\hat{\hat{\alpha}}_t\right)_m = \sum_{m=1}^{L} \left[\vec{\eta}\left(\frac{\Phi_1}{c_1}A\right)\left(\frac{\Phi_2}{c_2}A\right)\cdots\left(\frac{\Phi_{t-1}}{c_{t-1}}A\right)\frac{\Phi_t}{c_t}\right]_m$$

(Equation 4.14)

Therefore,

$$c_1 = c_1 \cdot 1 = c_1 \cdot \sum_{m=1}^{L} \left(\hat{\hat{\alpha}}_1\right)_m = c_1 \cdot \sum_{m=1}^{L} \left[\vec{\eta}\frac{\Phi_1}{c_1}\right]_m = \sum_{m=1}^{L} [\vec{\eta}\Phi_1]_m = \sum_{m=1}^{L} (\vec{\alpha}_1)_m = \vec{\alpha}_1 \vec{1}^\intercal$$

For $t > 1$,

$$c_t = c_t \cdot 1 = c_t \cdot \sum_{m=1}^{L} \left(\hat{\hat{\alpha}}_t\right)_m = c_t \cdot \sum_{m=1}^{L} \left[\vec{\eta}\left(\frac{\Phi_1}{c_1}A\right)\left(\frac{\Phi_2}{c_2}A\right)\cdots\left(\frac{\Phi_{t-1}}{c_{t-1}}A\right)\frac{\Phi_t}{c_t}\right]_m$$

$$= \sum_{m=1}^{L} \left[\underbrace{\vec{\eta}\left(\frac{\Phi_1}{c_1}A\right)\left(\frac{\Phi_2}{c_2}A\right)\cdots\left(\frac{\Phi_{t-2}}{c_{t-2}}A\right)\frac{\Phi_{t-1}}{c_{t-1}}}_{\hat{\hat{\alpha}}_{t-1}} A\Phi_t\right]_m$$

$$= \sum_{m=1}^{L} \left(\hat{\hat{\alpha}}_{t-1} A \Phi_t\right)_m = \hat{\hat{\alpha}}_{t-1} A \Phi_t \vec{1}^\intercal$$

Consequently,

$$c_t = \begin{cases} \vec{\alpha}_1 \vec{1}^\intercal & \text{for } t = 1 \\ \hat{\hat{\alpha}}_{t-1} A \Phi_t \vec{1}^\intercal & \text{for } t > 1 \end{cases}$$

(Equation 4.15)

Given the trajectory of observation, transition and initial probabilities, and the emission probabilities, we can calculate the numerically stable forward probabilities and normalization factors by the forward-backward algorithm.[154]

For $t = 1$,

1. $\vec{\alpha}_1 \leftarrow \vec{\eta}\Phi_1$
2. $c_1 \leftarrow \vec{\alpha}_1 \vec{1}^\intercal$
3. $\hat{\vec{\alpha}}_1 \leftarrow \frac{1}{c_1}\vec{\alpha}_1$

For $t > 1$,

1. $\vec{b}_t \leftarrow \hat{\vec{\alpha}}_{t-1} A \Phi_t$
2. $c_t \leftarrow \vec{b}_t \vec{1}^\intercal$
3. $\hat{\vec{\alpha}}_t \leftarrow \frac{1}{c_1}\vec{b}_t$

Define the backward probabilities as

$$\left(\vec{\beta}_t\right)_m = \begin{cases} P\left(O_{(t+1):T} \middle| H_t = l\right) & \text{for } 1 \le t \le T-1 \\ 1 & \text{for } t = T \end{cases}$$

(Equation 4.16)

We have the following

$$\vec{\beta}_t^\intercal = \begin{cases} A\left[\prod_{\tau=t+1}^{T} (\Phi_\tau A)\right]\vec{1}^\intercal & \text{for } 1 \le t \le T-1 \\ \vec{1}^\intercal & \text{for } t = T \end{cases}$$

(Equation 4.17)

Notice that the probability for the model to generate a trajectory of observation that exactly matches the actual trajectory of observation is vanishingly small for a moderately long trajectory. Therefore, these backward probabilities would become vanishingly small for a moderately large t.

Define the numerically stable backward probabilities as

$$\left(\hat{\hat{\beta}}_t\right)_l = \begin{cases} \dfrac{P\left(O_{(t+1):T} \middle| H_t = l\right)}{P\left(O_{(t+1):T} \middle| O_{1:t}\right)} & \text{for } 1 \le t \le T-1 \\ 1 & \text{for } t = T \end{cases}$$

(Equation 4.18)

For $1 \le t \le T-1$, we have:

$$\hat{\hat{\beta}}_t = \frac{P\left(O_{(t+1):T} \middle| H_t = l\right)}{P\left(O_{(t+1):T} \middle| O_{1:t}\right)} = \frac{\beta_t}{\prod_{\tau=t+1}^{T} c_\tau} = \frac{A\left[\prod_{\tau=t+1}^{T} (\Phi_\tau A)\right]\vec{1}^\intercal}{\prod_{\tau=t+1}^{T} c_\tau} = A\left[\prod_{\tau=t+1}^{T} \left(\tilde{\Phi}_\tau A\right)\right]\vec{1}^\intercal$$

Consequently,

$$\hat{\hat{\beta}}_t = \begin{cases} A\left[\prod_{\tau=t+1}^{T} \left(\tilde{\Phi}_\tau A\right)\right]\vec{1}^\intercal & \text{for } 1 \le t \le T-1 \\ \vec{1}^\intercal & \text{for } t = T \end{cases}$$

(Equation 4.19)

The Markovian assumption assumes the probability distributions of the future states depend only on the current state. Therefore, when the current state is known, the past state has no effect on the probability distribution of the future states. In hidden Markov models, the probability of the observation at any time depends only on the hidden state at the time. Consequently, when the current hidden state is known, past and current observations have no effect on the probability distribution of the future observations. Therefore, we have

$$P\left(O_{(t+1):T} \middle| H_t = m, O_{1:t}\right) = P\left(O_{(t+1):T} \middle| H_t = m\right)$$

With the above simplification, we can show that the state probabilities given the trajectory of observations are the product of the numerically stable forward and backward probabilities.

$$P(H_t = m | O_{1:T}) = \frac{P(H_t = m, O_{1:T})}{P(O_{1:T})} = \frac{P\left(H_t = m, O_{1:t}, O_{(t+1):T}\right)}{P(O_{1:T})}$$

$$= \frac{P(H_t = m, O_{1:t})P\left(O_{(t+1):T} \middle| H_t = m, O_{1:T}\right)}{P(O_{1:T})}$$

$$= \frac{P(H_t = m, O_{1:t})P\left(O_{(t+1):T} \middle| H_t = m\right)}{P(O_{1:T})}$$

$$= \frac{P(H_t = m, O_{1:t})}{P(O_{1:t})} \cdot \frac{P\left(O_{(t+1):T} \middle| H_t = m\right)}{P\left(O_{(t+1):T} \middle| O_{1:t}\right)}$$

$$= P(H_t = m | O_{1:t}) \cdot \frac{P\left(O_{(t+1):T} \middle| H_t = m\right)}{P\left(O_{(t+1):T} \middle| O_{1:t}\right)}$$

$$= \left(\hat{\hat{\alpha}}_t\right)_m \left(\hat{\hat{\beta}}_t\right)_m$$

(Equation 4.20)

Stochastic gradient descent can optimize the parameters of a hidden Markov model if we can calculate the gradient of the log-likelihood.

According to (Equation 4.5), the likelihood is:

$$P(O_{1:T}) = \vec{\eta} \left[\prod_{\tau=1}^{T} (\Phi_\tau A)\right] \vec{1}^{\intercal}$$

According to (Equation 4.7), the likelihood is:

$$P(O_{1:T}) = \prod_{\tau=1}^{T} c_\tau$$

From (Equation 4.5) and (Equation 4.7), the gradient of the log-likelihood with respect to a parameter of the model ξ is:

$$\frac{\partial \ln[P(O_{1:T})]}{\partial \xi} = \frac{1}{P(O_{1:T})} \cdot \frac{\partial P(O_{1:T})}{\partial \xi} = \frac{1}{\prod_{\tau=1}^{T} c_\tau} \cdot \frac{\partial}{\partial \xi} \left[\vec{\eta} \left[\prod_{\tau=1}^{T} (\Phi_\tau A)\right] \vec{1}^{\intercal}\right]$$

(Equation 4.21)

The following assumes the transition matrix A is not dependent on the other parameters of the model.

According to matrix calculus, we have:

$$\left(\frac{\partial A}{\partial A_{lm}}\right)_{l'm'} = \delta_{ll'}\delta_{mm'} = \begin{cases} 1 & \text{if } l = l' \text{ and } m = m' \\ 0 & \text{otherwise} \end{cases}$$

(Equation 4.22)

where δ_{ij} is the Kronecker delta.

Therefore, we have:

$$\left(\vec{x}\frac{\partial A}{\partial A_{lm}}\right)_{m'} = \sum_{l'=1}^{L}(x)_{l'}\left(\frac{\partial A}{\partial A_{lm}}\right)_{l'm'} = (x)_l\left(\frac{\partial A}{\partial A_{lm}}\right)_{lm'} = (x)_l\delta_{mm'} = \begin{cases} x_l & \text{if } m = m' \\ 0 & \text{otherwise} \end{cases}$$

(Equation 4.23)

and

$$\vec{x}\frac{\partial A}{\partial A_{lm}}\vec{y}^{\intercal} = \left(\vec{x}\frac{\partial A}{\partial A_{lm}}\right)\vec{y}^{\intercal} = \sum_{m'=1}^{L}(x)_l\delta_{mm'}(y)_{m'} = (x)_l(y)_m$$

(Equation 4.24)

From the product rule in matrix calculus, we have:

$$\frac{\partial(\Phi_\tau A)}{\partial A_{lm}} = \Phi_\tau\frac{\partial \mathrm{A}}{\partial A_{lm}} + \frac{\partial \Phi_\tau}{\partial A_{lm}}A = \Phi_\tau\frac{\partial \mathrm{A}}{\partial A_{lm}}$$

(Equation 4.25)

By equations (Equation 4.21), derivative of the log-likelihood with respect to an element in the transition matrix is:

$$\frac{\partial \ln[P(O_{1:T})]}{\partial A_{lm}} = \frac{1}{\prod_{\tau=1}^{T}c_\tau}\cdot\frac{\partial}{\partial A_{lm}}\left[\vec{\eta}\left[\prod_{\tau=1}^{T}(\Phi_\tau A)\right]\vec{1}^{\intercal}\right]$$

$$= \frac{1}{\prod_{\tau=1}^{T}c_\tau}\cdot\vec{\eta}\frac{\partial \prod_{\tau=1}^{T}(\Phi_\tau A)}{\partial A_{lm}}\vec{1}^{\intercal}$$

According to the product rule of matrix calculus, the above expression is equal to the following.

$$\frac{1}{\prod_{\tau=1}^{T} c_\tau} \cdot \vec{\eta} \frac{\partial \prod_{\tau=1}^{T} (\Phi_\tau A)}{\partial A_{lm}} \vec{1}^\intercal$$

$$= \frac{1}{\prod_{\tau=1}^{T} c_\tau} \cdot \vec{\eta} \begin{pmatrix} & \frac{\partial(\Phi_1 A)}{\partial A_{lm}} (\Phi_2 A) \cdots (\Phi_{T-1} A)(\Phi_T A) \\ + & (\Phi_1 A) \frac{\partial(\Phi_2 A)}{\partial A_{lm}} \cdots (\Phi_{T-1} A)(\Phi_T A) \\ \cdots & \\ + & (\Phi_1 A)(\Phi_2 A) \cdots \frac{\partial(\Phi_{T-1} A)}{\partial A_{lm}} (\Phi_T A) \\ + & (\Phi_1 A)(\Phi_2 A) \cdots (\Phi_{T-1} A) \frac{\partial(\Phi_T A)}{\partial A_{lm}} \end{pmatrix} \vec{1}^\intercal$$

According to (Equation 4.25), the above expression is equal to the following.

$$\frac{1}{\prod_{\tau=1}^{T} c_\tau} \cdot \vec{\eta} \begin{pmatrix} & \Phi_1 \frac{\partial \mathrm{A}}{\partial A_{lm}} (\Phi_2 A) \cdots (\Phi_{T-1} A)(\Phi_T A) \\ + & (\Phi_1 A) \Phi_2 \frac{\partial \mathrm{A}}{\partial A_{lm}} (\Phi_3 A) \cdots (\Phi_{T-1} A)(\Phi_T A) \\ \cdots & \\ + & (\Phi_1 A)(\Phi_2 A) \cdots (\Phi_{T-2} A) \Phi_{T-1} \frac{\partial \mathrm{A}}{\partial A_{lm}} (\Phi_T A) \\ + & (\Phi_1 A)(\Phi_2 A) \cdots (\Phi_{T-1} A) \Phi_T \frac{\partial \mathrm{A}}{\partial A_{lm}} \end{pmatrix} \vec{1}^\intercal$$

According to (Equation 4.23) and (Equation 4.24) the above expression is equal to the following.

$$\frac{1}{\prod_{\tau=1}^{T} c_\tau} \cdot \begin{pmatrix} & [\vec{\eta} \Phi_1]_l \left[(\Phi_2 A) \cdots (\Phi_{T-1} A)(\Phi_T A) \vec{1}^\intercal\right]_m \\ + & [\vec{\eta} (\Phi_1 A) \Phi_2]_l \left[(\Phi_3 A) \cdots (\Phi_{T-1} A)(\Phi_T A) \vec{1}^\intercal\right]_m \\ \cdots & \\ + & [\vec{\eta} (\Phi_1 A)(\Phi_2 A) \cdots (\Phi_{T-2} A) \Phi_{T-1}]_l \left[\Phi_T A \vec{1}^\intercal\right]_m \\ + & [\vec{\eta} (\Phi_1 A)(\Phi_2 A) \cdots (\Phi_{T-1} A) \Phi_T]_l \left[\vec{1}^\intercal\right]_m \end{pmatrix}$$

According to (Equation 4.12) and (Equation 4.25), the above expression is equal to the following.

$$\begin{pmatrix} & [\vec{\eta}\tilde{\Phi}_1]_l[(\tilde{\Phi}_2 A)\cdots(\tilde{\Phi}_{T-1}A)(\tilde{\Phi}_T A)\vec{1}^\intercal]_m \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)\tilde{\Phi}_2]_l[(\tilde{\Phi}_3 A)\cdots(\tilde{\Phi}_{T-1}A)(\tilde{\Phi}_T A)\vec{1}^\intercal]_m \\ \cdots & \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)(\tilde{\Phi}_2 A)\cdots(\tilde{\Phi}_{T-2}A)\tilde{\Phi}_{T-1}]_l[\tilde{\Phi}_T A\vec{1}^\intercal]_m \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)(\tilde{\Phi}_2 A)\cdots(\tilde{\Phi}_{T-1}A)\tilde{\Phi}_T]_l[\vec{1}^\intercal]_m \end{pmatrix}$$

$$= \begin{pmatrix} & [\vec{\eta}\tilde{\Phi}_1]_l[\tilde{\Phi}_2 \times A(\tilde{\Phi}_3 A)\cdots(\tilde{\Phi}_{T-1}A)(\tilde{\Phi}_T A)\vec{1}^\intercal]_m \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)\tilde{\Phi}_2]_l[\tilde{\Phi}_3 \times A(\tilde{\Phi}_4 A)\cdots(\tilde{\Phi}_{T-1}A)(\tilde{\Phi}_T A)\vec{1}^\intercal]_m \\ \cdots & \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)(\tilde{\Phi}_2 A)\cdots(\tilde{\Phi}_{T-2}A)\tilde{\Phi}_{T-1}]_l[\tilde{\Phi}_T \times A\vec{1}^\intercal]_m \\ + & [\vec{\eta}(\tilde{\Phi}_1 A)(\tilde{\Phi}_2 A)\cdots(\tilde{\Phi}_{T-1}A)\tilde{\Phi}_T]_l[\vec{1}^\intercal]_m \end{pmatrix}$$

According to (Equation 4.11), $A\vec{1}^\intercal = \vec{1}^\intercal = \hat{\hat{\beta}}_T$. Therefore, according to (Equation 4.12) and (Equation 4.19), the above expression is equal to the following.

$$\begin{pmatrix} & (\hat{\hat{\alpha}}_1)_l\left(\tilde{\Phi}_2\hat{\hat{\beta}}_2\right)_m \\ + & (\hat{\hat{\alpha}}_2)_l\left(\tilde{\Phi}_3\hat{\hat{\beta}}_3\right)_m \\ \cdots & \\ + & (\hat{\hat{\alpha}}_{T-1})_l\left[\tilde{\Phi}_T\hat{\hat{\beta}}_T\right]_m \\ + & (\hat{\hat{\alpha}}_T)_l \end{pmatrix}$$

Therefore,

$$\frac{\partial \ln[P(O_{1:T})]}{\partial A_{lm}} = (\hat{\hat{\alpha}}_T)_l + \sum_{\tau=1}^{T-1}(\hat{\hat{\alpha}}_\tau)_l\left[\tilde{\Phi}_{\tau+1}\hat{\hat{\beta}}_{\tau+1}\right]_m$$

(Equation 4.26)

The following assumes the transition matrix A is not dependent on the state-dependent emission probabilities $\phi_{m,t}$. In other words, $\frac{\partial A}{\partial \phi_{m,t}}$ is the zero matrix.

According to the product rule of matrix calculus, we have:

$$\frac{\partial(\Phi_t A)}{\partial \ln \phi_{m,t}} = \Phi_t\frac{\partial A}{\partial \ln \phi_{m,t}} + \frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}A = 0 + \frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}A = \frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}A$$

(Equation 4.27)

By renaming variable $\mathrm{y} = \ln x$, we have:

$$\frac{\partial x}{\partial \ln x} = \frac{\partial e^{\ln x}}{\partial \ln x} = \frac{\partial e^{y}}{\partial y} = e^{y} = x$$

Consequently, we have:

$$\left(\frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}\right)_{l'm'} = \delta_{mm'}\delta_{ml'}\frac{\partial \phi_{m,t}}{\partial \ln \phi_{m,t}} = \delta_{mm'}\delta_{ml'}\phi_{m,t}$$

$$\left[\vec{x}\left(\frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}\right)\right]_{m'} = \sum_{l'=1}^{L}(x)_{l'}\left(\frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}\right)_{l'm'} = \sum_{l'=1}^{L}(x)_{l'}\delta_{mm'}\delta_{ml'}\phi_{m,t} = \delta_{mm'}(x)_m\phi_{m,t}$$

$$\vec{x}\left(\frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}\right)\vec{y}^{\intercal} = \left(\vec{x}\frac{\partial \Phi_t}{\partial \ln \phi_{m,t}}\right)\vec{y}^{\intercal} = \sum_{m'=1}^{L}\delta_{mm'}(x)_m\phi_{m,t}(y)_m = (x)_m\phi_{m,t}(y)_m$$

(Equation 4.28)

By (Equation 4.21) and (Equation 4.28), we have:

$$\frac{\partial \ln[P(O_{1:T})]}{\partial \ln \phi_{m,t}} = \frac{1}{\prod_{\tau=1}^{T} c_\tau}\cdot\frac{\partial}{\partial \ln \phi_{m,t}}\left[\vec{\eta}\left[\prod_{\tau=1}^{T}(\Phi_\tau A)\right]\vec{1}^{\intercal}\right]$$

$$= \frac{1}{\prod_{\tau=1}^{T} c_\tau}\cdot\vec{\eta}(\Phi_1 A)\cdots(\Phi_{t-1}A)\frac{\partial(\Phi_\tau A)}{\partial \ln \phi_{m,t}}(\Phi_{t+1}A)\cdots(\Phi_T A)\vec{1}^{\intercal}$$

$$= \frac{1}{\prod_{\tau=1}^{T} c_\tau}\cdot[\vec{\eta}(\Phi_1 A)\cdots(\Phi_{t-1}A)]_m\phi_{m,t}\left[(\Phi_{t+1}A)\cdots(\Phi_T A)\vec{1}^{\intercal}\right]_m$$

Since Φ_t is a diagonal matrix, the above expression is equal to:

$$\frac{1}{\prod_{\tau=1}^{T} c_\tau}\cdot[\vec{\eta}(\Phi_1 A)\cdots(\Phi_{t-1}A)\Phi_t]_m\left[(\Phi_{t+1}A)\cdots(\Phi_T A)\vec{1}^{\intercal}\right]_m$$

By equations (Equation *4.11*), (Equation *4.12*), (Equation 4.19), and (Equation 4.20), the above expression is equal to:

$$\left[\vec{\eta}\left(\widetilde{\Phi}_1 A\right)\cdots\left(\widetilde{\Phi}_{t-1}A\right)\widetilde{\Phi}_t\right]_m\left[\left(\widetilde{\Phi}_{t+1}A\right)\cdots\left(\widetilde{\Phi}_T A\right)\vec{1}^{\intercal}\right]_m$$

$$= \left(\hat{\hat{\alpha}}_t\right)_m \left(\hat{\hat{\beta}}_t\right)_m$$

$$= P(H_t = m|O_{1:T})$$

Consequently,

$$\frac{\partial \ln[P(O_{1:T})]}{\partial \ln \phi_{m,t}} = \left(\hat{\hat{\alpha}}_t\right)_m \left(\hat{\hat{\beta}}_t\right)_m = P(H_t = m|O_{1:T})$$

(Equation 4.29)

Each residue has a Dirichlet distribution that models which domain contains the residue. This Dirichlet distribution has the following probability density function.

$$\rho(\vec{\upsilon}) = \frac{1}{K} \prod_{i=1}^{D} \upsilon_i^{\lambda-1}$$

where D is the number of domains. K is the normalization constant.

$\vec{\upsilon}$ is the membership vector. When $\upsilon_i = 1$, the residue fully belongs to the i^{th} domain. When $\upsilon_i = 0$, the residue fully does not belong to the i^{th} domain. Dirichlet distribution constraints $\vec{\upsilon}$ by the following equations.

$$\begin{cases} \forall i: \upsilon_i \in [0, 1] \\ \displaystyle\sum_{i=1}^{D} \upsilon_i = 1 \end{cases}$$

We chose a λ to make $\vec{\upsilon}$ have a high chance of having one element equal to 1 and the remaining elements equal to 0. In practice, we picked $\lambda = 0.9$.

We implement the constraints of the membership vector through two transformations[155].

The first transformation f is from a $1 \times (D - 1)$ vector $\vec{z}$ in the unit cube to the $1 \times D$ vector $\vec{\upsilon}$ in the probability simplex.

$$f(\vec{z}) = \vec{\upsilon}$$

where

$$z_j = \frac{v_j}{1 - \sum_{a=1}^{j-1} v_a}$$

The first transformation transforms a point in the probability simplex to the unit cube. Notice that the transformation transforms the $1 \times D$ vector $\vec{v}$ to a $1 \times (D-1)$ vector $\vec{z}$.

$$z_j = \frac{v_j}{1 - \sum_{a=1}^{j-1} v_a}$$

The model would fit the transformed membership vector. At the end of the fitting, we recover the membership vector with the inverse transform.

5 CONCLUSION

I have worked on three projects about three significant challenges in understanding the role of motion in the functional mechanism of biomolecules through analyzing MD simulations. The three challenges are modeling the diffusive internal motion of biomolecules, taming the combinatorial explosion of the number of local shapes, and simplifying the hierarchy of motions. The first project finds the diffusive internal motions of a biomolecule have diffusion coefficients that vary periodically over time. The second project extracts the local shapes that frequently occur and find the correlation between the local shapes and the overall conformation. The third project models the contribution of each residue to the coupled dynamics of the domains of a protein. The results identify residues that correlate to the state of the domains, and one domain is a switch that controls the dynamics of the other domains.

The functions of biological molecules depend on both the motions and structures of the biological molecules. Therefore, a deep understanding of motion has profound consequences for the understanding and application of biomolecules. We would understand the motion of biomolecules directly as motions instead of a structure with unclear drives for motion. Deep knowledge of the motions of biomolecules would enable factoring the motions. The factoring can significantly speed up the MD simulations of many different drugs bound to the same target by factoring out the motions of the same drug target. Such a systematic understanding of motion would also provide a model that predicts if a tiny modification of a drug can cause a significant change in the behavior of the target. Without this prediction, potentially, we have to perform a full MD simulation for each tiny modification of the drug.

Motion can potentially become another tool for reducing side effects by ensuring the drug would only affect its intended target. For example, structurally similar biomolecules are often the source of drugs' unintended interactions and side effects. However, two structurally similar biomolecules may have very different motions. Therefore, developing drugs that target both the motion and structure may reduce side effects.

A reductionist understanding of motion can benefit both the design of experiments and computational modeling. Since the function of a biomolecule depends on both its structure and motion, the understanding of motion can benefit the creation of artificial proteins with novel functions. The synthesis of molecules depends on the rearrangement of atoms and bonds. A deep understanding of the motions of molecules can also benefit synthetic chemistry and other branches of chemistry.

A possible future direction would be to factor the motion of biomolecules into the motions of the parts with neural network architectures based on the ideas in the three projects. The factoring would describe how the change of the motion of one part affects the motions of the other parts. Artificial neural networks can potentially complement theories in attaining a deep understanding of motion. In MD simulations, Newtonian dynamics describe the detailed motions of every atom. Although Newtonian dynamics is useful for performing the simulations, this detailed description is not an intuitive model of the overall behavior. A more abstract model of the general behavior in terms of the motion of the parts is valuable. However, biomolecules are often nonlinear. Since theories often have difficulty describing nonlinear systems in general, a hybrid approach to model the overall motion of biomolecule may be more promising. For example, the coupling between the motions of the parts is often complex and nonlinear. Artificial neural networks can model a suitable nonlinear transformation of the couplings that turn the nonlinear relationships into linear relationships that are far more intuitive and interpretable for humans. If a partial understanding of the intuitive part of motions is sufficient, we can use theory to explain only the intuitive and decomposable part of motions. Of course, the artificial neural network is not a solution for all problems. Unless carefully designed, an artificial neural network might be a black box without offering any margin of error, an explicit guarantee of reliability, or explanation.

REFERENCES

1. Lin, X.; Li, X.; Lin, X., A Review on Applications of Computational Methods in Drug Screening and Design. *Molecules* **2020,** *25* (6), 1375.
2. Ferguson, F. M.; Gray, N. S., Kinase inhibitors: the road ahead. *Nature Reviews Drug Discovery* **2018,** *17* (5), 353-377.
3. De Vivo, M.; Masetti, M.; Bottegoni, G.; Cavalli, A., Role of Molecular Dynamics and Related Methods in Drug Discovery. *Journal of Medicinal Chemistry* **2016,** *59* (9), 4035-4061.
4. Mackerell Jr, A. D., Empirical force fields for biological macromolecules: Overview and issues. *Journal of Computational Chemistry* **2004,** *25* (13), 1584-1604.
5. Laberge, M.; Yonetani, T., Molecular dynamics simulations of hemoglobin A in different states and bound to DPG: effector-linked perturbation of tertiary conformations and HbA concerted dynamics. *Biophysical journal* **2008,** *94* (7), 2737-2751.
6. Lu, H.; Zhou, Q.; He, J.; Jiang, Z.; Peng, C.; Tong, R.; Shi, J., Recent advances in the development of protein–protein interactions modulators: mechanisms and clinical trials. *Signal Transduction and Targeted Therapy* **2020,** *5* (1), 213.
7. Karplus, M.; McCammon, J. A., Molecular dynamics simulations of biomolecules. *Nature Structural Biology* **2002,** *9* (9), 646-652.
8. Hollingsworth, S. A.; Dror, R. O., Molecular Dynamics Simulation for All. *Neuron* **2018,** *99* (6), 1129-1143.
9. Klopp, J.; Winterhalter, A.; Gébleux, R.; Scherer-Becker, D.; Ostermeier, C.; Gossert, A. D., Cost-effective large-scale expression of proteins for NMR studies. *Journal of Biomolecular NMR* **2018,** *71* (4), 247-262.
10. Teilum, K.; Kunze, M. B. A.; Erlendsson, S.; Kragelund, B. B., (S)Pinning down protein interactions by NMR. *Protein Sci* **2017,** *26* (3), 436-451.
11. Tian, C.; Kasavajhala, K.; Belfon, K. A. A.; Raguette, L.; Huang, H.; Migues, A. N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; Simmerling, C., ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of Chemical Theory and Computation* **2020,** *16* (1), 528-552.
12. Maragakis, P.; Lindorff-Larsen, K.; Eastwood, M. P.; Dror, R. O.; Klepeis, J. L.; Arkin, I. T.; Jensen, M. Ø.; Xu, H.; Trbovic, N.; Friesner, R. A.; Palmer, A. G.; Shaw, D. E., Microsecond molecular dynamics simulation shows effect of slow loop dynamics on backbone amide order parameters of proteins. *The journal of physical chemistry. B* **2008,** *112* (19), 6155-6158.
13. Wells, B. A.; Chaffee, A. L., Ewald Summation for Molecular Simulations. *Journal of Chemical Theory and Computation* **2015,** *11* (8), 3684-3695.
14. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G., A smooth particle mesh Ewald method. *The Journal of Chemical Physics* **1995,** *103* (19), 8577-8593.
15. Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R., A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics* **1982,** *76* (1), 637-649.
16. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R., Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics* **1984,** *81* (8), 3684-3690.

17. Rogge, S. M. J.; Vanduyfhuys, L.; Ghysels, A.; Waroquier, M.; Verstraelen, T.; Maurin, G.; Van Speybroeck, V., A Comparison of Barostats for the Mechanical Characterization of Metal–Organic Frameworks. *Journal of Chemical Theory and Computation* **2015,** *11* (12), 5583-5597.
18. Parrinello, M.; Rahman, A., Polymorphic Transitions in Single Crystals: A New Molecular Dynamics Method. *Journal of Applied Physics* **1981,** *52* (12), 7182-7190.
19. Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F., Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics* **2011,** *134* (17), 174105.
20. Pande, V. S.; Beauchamp, K.; Bowman, G. R., Everything you wanted to know about Markov State Models but were afraid to ask. *Methods (San Diego, Calif.)* **2010,** *52* (1), 99-105.
21. Upadhyay, S. K., Dynamics of Gal80p in the Gal80p-Gal3p complex differ significantly from the dynamics in the Gal80p-Gal1p complex: implications for the higher specificity of Gal3p. *Molecular BioSystems* **2014,** *10* (12), 3120-3129.
22. Kamberaj, H.; van der Vaart, A., Extracting the Causality of Correlated Motions from Molecular Dynamics Simulations. *Biophysical Journal* **2009,** *97* (6), 1747-1755.
23. Ghosh, A.; Vishveshwara, S., A study of communication pathways in methionyl- tRNA synthetase by molecular dynamics simulations and structure network analysis. *Proceedings of the National Academy of Sciences of the United States of America* **2007,** *104* (40), 15711-15716.
24. Hayward, S.; de Groot, B. L., Normal Modes and Essential Dynamics. In *Molecular Modeling of Proteins*, Kukol, A., Ed. Humana Press: Totowa, NJ, 2008; Vol. 443, pp 89-106.
25. Karplus, M.; Kushick, J. N., Method for estimating the configurational entropy of macromolecules. *Macromolecules* **1981,** *14* (2), 325-332.
26. Rosenblum, M. G.; Pikovsky, A. S., Detecting direction of coupling in interacting oscillators. *Physical Review E* **2001,** *64* (4), 045202.
27. Paluš, M.; Stefanovska, A., Direction of coupling from phases of interacting oscillators: An information-theoretic approach. *Physical Review E* **2003,** *67* (5), 055201.
28. Schreiber, T., Measuring Information Transfer. *Physical Review Letters* **2000,** *85* (2), 461-464.
29. Staniek, M.; Lehnertz, K., Symbolic Transfer Entropy. *Physical Review Letters* **2008,** *100* (15), 158101.
30. Vejmelka, M.; Paluš, M., Inferring the directionality of coupling with conditional mutual information. *Physical Review E* **2008,** *77* (2), 026214.
31. Frenzel, S.; Pompe, B., Partial Mutual Information for Coupling Analysis of Multivariate Time Series. *Physical Review Letters* **2007,** *99* (20), 204101.
32. Huang, H.-Y.; Ombao, H.; Stoffer, D. S., Discrimination and classification of nonstationary time series using the SLEX model. *Journal of the American Statistical Association* **2004,** *99* (467), 763-774.
33. Vu, V. Q.; Yu, B.; Kass, R. E., Information In The Non-Stationary Case. *Neural computation* **2009,** *21* (3), 688-703.
34. Lange, O. F.; Grubmüller, H., Generalized correlation for biomolecular dynamics. *Proteins: Structure, Function, and Bioinformatics* **2006,** *62* (4), 1053-1061.
35. Dahlhaus, R., On the Kullback-Leibler information divergence of locally stationary processes. *Stochastic Processes and their Applications* **1996,** *62* (1), 139-168.
36. Roy, A.; Post, C. B., Detection of long-range concerted motions in protein by a distance covariance. *Journal of chemical theory and computation* **2012,** *8* (9), 3009-3014.

37. Henzler-Wildman, K. A.; Lei, M.; Thai, V.; Kerns, S. J.; Karplus, M.; Kern, D., A hierarchy of timescales in protein dynamics is linked to enzyme catalysis. *Nature* **2007,** *450* (7171), 913-916.
38. Kuimova, M. K., Mapping viscosity in cells using molecular rotors. *Physical Chemistry Chemical Physics* **2012,** *14* (37), 12671-12686.
39. Woutersen, S.; Hamm, P., Structure Determination of Trialanine in Water Using Polarization Sensitive Two-Dimensional Vibrational Spectroscopy. *The Journal of Physical Chemistry B* **2000,** *104* (47), 11316-11320.
40. Mellors, R.; Vernon, F.; Thomson, D., Detection of dispersive signals using multitaper dual-frequency coherence. *Geophysical Journal International* **1998,** *135* (1), 146-154.
41. Gardner, W. A., The spectral correlation theory of cyclostationary time-series. *Signal processing* **1986,** *11* (1), 13-36.
42. Johnson, Q.; Doshi, U.; Shen, T.; Hamelberg, D., Water’s Contribution to the Energetic Roughness from Peptide Dynamics. *Journal of Chemical Theory and Computation* **2010,** *6* (9), 2591-2597.
43. Barndorff-Nielsen, O. E.; Jensen, J. L.; Sørensen, M., Some stationary processes in discrete and continuous time. *Advances in Applied Probability* **1998,** *30* (4), 989-1007.
44. Amadei, A.; de Groot, B. L.; Ceruso, M. A.; Paci, M.; Di Nola, A.; Berendsen, H. J. C., A kinetic model for the internal motions of proteins: Diffusion between multiple harmonic wells. *Proteins: Structure, Function, and Bioinformatics* **1999,** *35* (3), 283-292.
45. Chen, M.; Li, X.; Liu, C., Computation of the memory functions in the generalized Langevin models for collective dynamics of macromolecules. *The Journal of Chemical Physics* **2014,** *141* (6), 064112.
46. Wang, D.; Li, Y.; Nie, P., A study on the Gaussianity and stationarity of the random noise in the seismic exploration. *Journal of Applied Geophysics* **2014,** *109*, 210-217.
47. Borghesani, P.; Pennacchi, P.; Ricci, R.; Chatterton, S., Testing second order cyclostationarity in the squared envelope spectrum of non-white vibration signals. *Mechanical Systems and Signal Processing* **2013,** *40* (1), 38-55.
48. Gardner, W. A., Exploitation of spectral redundancy in cyclostationary signals. *Signal Processing Magazine, IEEE* **1991,** *8* (2), 14-36.
49. Ait-Sghir, K.; badaoui, M. E.; Guillet, F.; Aboutajdine, D.; Dron, J.-P., Application of the cyclostationairity for the cutting tool diagnosis. *Mechanics & Industry* **2014,** *15* (06), 497-507.
50. Bonnardot, F.; Randall, R. B.; Guillet, F., Extraction of second-order cyclostationary sources—Application to vibration analysis. *Mechanical Systems and Signal Processing* **2005,** *19* (6), 1230-1244.
51. Bair, W.; Koch, C.; Newsome, W.; Britten, K., Power spectrum analysis of bursting cells in area MT in the behaving monkey. *The Journal of neuroscience* **1994,** *14* (5), 2870-2892.
52. Randall, R. B.; Antoni, J.; Chobsaard, S., The relationship between spectral correlation and envelope analysis in the diagnostics of bearing faults and other cyclostationary machine signals. *Mechanical systems and signal processing* **2001,** *15* (5), 945-962.
53. Wahba, G., Automatic smoothing of the log periodogram. *Journal of the American Statistical Association* **1980,** *75* (369), 122-132.
54. Costa, A. H.; Boudreaux-Bartels, G. F., An overview of aliasing errors in discrete-time formulations of time-frequency representations. *Signal Processing, IEEE Transactions on* **1999,** *47* (5), 1463-1474.

55. Blondel, A.; Karplus, M., New formulation for derivatives of torsion angles and improper torsion angles in molecular mechanics: Elimination of singularities. *Journal of computational chemistry* **1996,** *17* (9), 1132-1141.
56. Mazur, A. K., Common molecular dynamics algorithms revisited: accuracy and optimal time steps of Störmer–Leapfrog integrators. *Journal of Computational Physics* **1997,** *136* (2), 354-365.
57. Bolboaca, S. D.; Jantschi, L., Pearson versus Spearman, Kendall's Tau Correlation Analysis on Structure-Activity Relationships of Biologic Active Compounds. *Leonardo Journal of Sciences* **2006,** *5* (9), 179-200.
58. Hoover, K. D., Nonstationary Time Series, Cointegration, and the Principle of the Common Cause. *The British Journal for the Philosophy of Science* **2003,** *54* (4), 527-551.
59. Romero, J.; García-Beltrán, A.; Hernández-Andrés, J., Linear bases for representation of natural and artificial illuminants. *J. Opt. Soc. Am. A* **1997,** *14* (5), 1007-1014.
60. Granger, C. W. J., Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica* **1969,** *37* (3), 424-438.
61. Gardner, W. A.; Napolitano, A.; Paura, L., Cyclostationarity: Half a century of research. *Signal Processing* **2006,** *86* (4), 639-697.
62. Ma, J., Usefulness and Limitations of Normal Mode Analysis in Modeling Dynamics of Biomolecular Complexes. *Structure* **2005,** *13* (3), 373-380.
63. McCammon, J. A., Protein dynamics. *Reports on Progress in Physics* **1984,** *47* (1), 1.
64. Or, D.; Leij, F. J.; Snyder, V.; Ghezzehei, T. A., Stochastic model for posttillage soil pore space evolution. *Water Resources Research* **2000,** *36* (7), 1641-1652.
65. Hünenberger, P. H., Thermostat Algorithms for Molecular Dynamics Simulations. In *Advanced Computer Simulation*, Holm, C.; Kremer, K., Eds. Springer: Berlin, 2005; Vol. 173, pp 105-149.
66. Carroll, T., Chaotic communications that are difficult to detect. *Physical Review E* **2003,** *67* (2), 026207.
67. Adcock, S. A.; McCammon, J. A., Molecular Dynamics: Survey of Methods for Simulating the Activity of Proteins. *Chemical Reviews* **2006,** *106* (5), 1589-1615.
68. Hart, K.; Foloppe, N.; Baker, C. M.; Denning, E. J.; Nilsson, L.; MacKerell, A. D., Optimization of the CHARMM Additive Force Field for DNA: Improved Treatment of the BI/BII Conformational Equilibrium. *Journal of Chemical Theory and Computation* **2012,** *8* (1), 348-362.
69. Ivani, I.; Dans, P. D.; Noy, A.; Pérez, A.; Faustino, I.; Hospital, A.; Walther, J.; Andrio, P.; Goñi, R.; Balaceanu, A.; Portella, G.; Battistini, F.; Gelpí, J. L.; González, C.; Vendruscolo, M.; Laughton, C. A.; Harris, S. A.; Case, D. A.; Orozco, M., Parmbsc1: a refined force field for DNA simulations. *Nature Methods* **2015,** *13*, 55.
70. Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C., ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *Journal of Chemical Theory and Computation* **2015,** *11* (8), 3696-3713.
71. Zhu, X.; Lopes, P. E. M.; MacKerell, A. D., Recent Developments and Applications of the CHARMM force fields. *Wiley interdisciplinary reviews. Computational molecular science* **2012,** *2* (1), 167-185.
72. Doshi, U.; Hamelberg, D., Towards fast, rigorous and efficient conformational sampling of biomolecules: Advances in accelerated molecular dynamics. *Biochimica et Biophysica Acta (BBA) - General Subjects* **2015,** *1850* (5), 878-888.

73. Götz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C., Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *Journal of Chemical Theory and Computation* **2012,** *8* (5), 1542-1555.
74. Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C., Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *Journal of Chemical Theory and Computation* **2013,** *9* (9), 3878-3888.
75. Shaw, D. E.; Grossman, J. P.; Bank, J. A.; Batson, B.; Butts, J. A.; Chao, J. C.; Deneroff, M. M.; Dror, R. O.; Even, A.; Fenton, C. H.; Forte, A.; Gagliardo, J.; Gill, G.; Greskamp, B.; Ho, C. R.; Ierardi, D. J.; Iserovich, L.; Kuskin, J. S.; Larson, R. H.; Layman, T.; Lee, L.-S.; Lerer, A. K.; Li, C.; Killebrew, D.; Mackenzie, K. M.; Mok, S. Y.-H.; Moraes, M. A.; Mueller, R.; Nociolo, L. J.; Peticolas, J. L.; Quan, T.; Ramot, D.; Salmon, J. K.; Scarpazza, D. P.; Schafer, U. B.; Siddique, N.; Snyder, C. W.; Spengler, J.; Tang, P. T. P.; Theobald, M.; Toma, H.; Towles, B.; Vitale, B.; Wang, S. C.; Young, C., Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE Press: New Orleans, Louisana, 2014; pp 41-53.
76. Shaw, D. E.; Dror, R. O.; Salmon, J. K.; Grossman, J. P.; Mackenzie, K. M.; Bank, J. A.; Young, C.; Deneroff, M. M.; Batson, B.; Bowers, K. J.; Chow, E.; Eastwood, M. P.; Ierardi, D. J.; Klepeis, J. L.; Kuskin, J. S.; Larson, R. H.; Lindorff-Larsen, K.; Maragakis, P.; Moraes, M. A.; Piana, S.; Shan, Y.; Towles, B., Millisecond-scale molecular dynamics simulations on Anton. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, ACM: Portland, Oregon, 2009; pp 1-11.
77. Nussinov, R.; Tsai, C.-J., Allostery in Disease and in Drug Discovery. *Cell* **2013,** *153* (2), 293-305.
78. Koshland, D. E.; Némethy, G.; Filmer, D., Comparison of Experimental Binding Data and Theoretical Models in Proteins Containing Subunits. *Biochemistry* **1966,** *5* (1), 365-385.
79. Rastogi, V. K.; Girvin, M. E., Structural changes linked to proton translocation by subunit c of the ATP synthase. *Nature* **1999,** *402*, 263.
80. Fishelovitch, D.; Shaik, S.; Wolfson, H. J.; Nussinov, R., Theoretical Characterization of Substrate Access/Exit Channels in the Human Cytochrome P450 3A4 Enzyme: Involvement of Phenylalanine Residues in the Gating Mechanism. *The Journal of Physical Chemistry B* **2009,** *113* (39), 13018-13025.
81. Xin, Y.; Gadda, G.; Hamelberg, D., The Cluster of Hydrophobic Residues Controls the Entrance to the Active Site of Choline Oxidase. *Biochemistry* **2009,** *48* (40), 9599-9605.
82. Wang, W.; Jiang, C.; Zhang, J.; Ye, W.; Luo, R.; Chen, H.-F., Dynamics Correlation Network for Allosteric Switching of PreQ1 Riboswitch. *Scientific Reports* **2016,** *6*, 31005.
83. Vettoretti, G.; Moroni, E.; Sattin, S.; Tao, J.; Agard, D. A.; Bernardi, A.; Colombo, G., Molecular Dynamics Simulations Reveal the Mechanisms of Allosteric Activation of Hsp90 by Designed Ligands. *Scientific Reports* **2016,** *6*, 23830.
84. Ricci, C. G.; Silveira, R. L.; Rivalta, I.; Batista, V. S.; Skaf, M. S., Allosteric Pathways in the PPARγ-RXRα nuclear receptor complex. *Scientific Reports* **2016,** *6*, 19940.
85. Long, S.; Tian, P., Nonlinear backbone torsional pair correlations in proteins. *Scientific Reports* **2016,** *6*, 34481.
86. McClendon, C. L.; Friedland, G.; Mobley, D. L.; Amirkhani, H.; Jacobson, M. P., Quantifying Correlations Between Allosteric Sites in Thermodynamic Ensembles. *Journal of Chemical Theory and Computation* **2009,** *5* (9), 2486-2502.

87. Cheng, S.; Song, W.; Yuan, X.; Xu, Y., Gorge Motions of Acetylcholinesterase Revealed by Microsecond Molecular Dynamics Simulations. *Scientific Reports* **2017,** *7* (1), 3219.
88. Shao, J.; Tanner, S. W.; Thompson, N.; Cheatham, T. E., Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. *Journal of Chemical Theory and Computation* **2007,** *3* (6), 2312-2334.
89. Pande, V. S.; Beauchamp, K.; Bowman, G. R., Everything You Wanted to Know about Markov State Models but were Afraid to Ask. *Methods* **2010,** *52* (1), 99-105.
90. McClendon, C. L.; Kornev, A. P.; Gilson, M. K.; Taylor, S. S., Dynamic Architecture of a Protein Kinase. *Proceedings of the National Academy of Sciences* **2014,** *111* (43), E4623-E4631.
91. Miao, Y.; Nichols, S. E.; Gasper, P. M.; Metzger, V. T.; McCammon, J. A., Activation and Dynamic Network of the M2 Muscarinic Receptor. *Proceedings of the National Academy of Sciences of the United States of America* **2013,** *110* (27), 10982-10987.
92. Granata, D.; Ponzoni, L.; Micheletti, C.; Carnevale, V., Patterns of coevolving amino acids unveil structural and dynamical domains. *Proceedings of the National Academy of Sciences of the United States of America* **2017,** *114* (50), E10612-E10621.
93. Dziubiński, M.; Daniluk, P.; Lesyng, B., ResiCon: a method for the identification of dynamic domains, hinges and interfacial regions in proteins. *Bioinformatics* **2016,** *32* (1), 25-34.
94. Bernhard, S.; Noé, F., Optimal Identification of Semi-Rigid Domains in Macromolecules from Molecular Dynamics Simulation. *PLOS ONE* **2010,** *5* (5), e10491.
95. Yan, X.; Han, J., CloseGraph: Mining Closed Frequent Graph Patterns. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM: Washington, D.C., 2003; pp 286-295.
96. Müllner, D., Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* **2011**.
97. Vrahatis, M. N.; Boutsinas, B.; Alevizos, P.; Pavlides, G., The New k-Windows Algorithm for Improving the k-Means Clustering Algorithm. *Journal of Complexity* **2002,** *18* (1), 375-391.
98. Ferreira, L.; Hitchcock, D. B., A Comparison of Hierarchical Methods for Clustering Functional Data. *Communications in Statistics - Simulation and Computation* **2009,** *38* (9), 1925-1949.
99. Ward, J. H., Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association* **1963,** *58* (301), 236-244.
100. Hunter, J. D., Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **2007,** *9* (3), 90-95.
101. van der Walt, S.; Colbert, S. C.; Varoquaux, G., The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* **2011,** *13* (2), 22-30.
102. Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O., MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations. *Journal of Computational Chemistry* **2011,** *32* (10), 2319-2327.
103. Humphrey, W.; Dalke, A.; Schulten, K., VMD: Visual Molecular Dynamics. *Journal of Molecular Graphics* **1996,** *14* (1), 33-38.
104. Dagum, L.; Menon, R., OpenMP: An Industry Standard API for Shared-memory Programming. *IEEE Computational Science and Engineering* **1998,** *5* (1), 46-55.
105. Gabriel, E.; Fagg, G. E.; Bosilca, G.; Angskun, T.; Dongarra, J. J.; Squyres, J. M.; Sahay, V.; Kambadur, P.; Barrett, B.; Lumsdaine, A. In *Open MPI: Goals, Concept, and Design of a*

Next Generation MPI Implementation, European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting, Springer: 2004; pp 97-104.
106. Ulmer, T. S.; Ramirez, B. E.; Delaglio, F.; Bax, A., Evaluation of Backbone Proton Positions and Dynamics in a Small Protein by Liquid Crystal NMR Spectroscopy. *Journal of the American Chemical Society* **2003,** *125* (30), 9179-9191.
107. Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L., Comparison of Simple Potential Functions for Simulating Liquid Water. *Journal of Chemical Physics* **1983,** *79* (2), 926-935.
108. Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R., Molecular Dynamics with Coupling to an External Bath. *Journal of Chemical Physics* **1984,** *81* (8), 3684-3690.
109. Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E., Improved Side-chain Torsion Potentials for the Amber ff99SB Protein Force Field. *Proteins: Structure, Function, and Bioinformatics* **2010,** *78* (8), 1950-1958.
110. Bussi, G.; Donadio, D.; Parrinello, M., Canonical Sampling through Velocity Rescaling. *Journal of Chemical Physics* **2007,** *126* (1), 014101.
111. Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R., A Computer Simulation Method for the Calculation of Equilibrium Constants for the Formation of Physical Clusters of Molecules: Application to Small Water Clusters. *J. Chem. Phys.* **1982,** *76* (1), 637-649.
112. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G., A Smooth Particle Mesh Ewald Method. *J. Chem. Phys.* **1995,** *103* (19), 8577-8593.
113. Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M., LINCS: A Linear Constraint Solver for Molecular Simulations. *Journal of Computational Chemistry* **1997,** *18* (12), 1463-1472.
114. Olsson, U.; Wolf-Watz, M., Overlap between Folding and Functional Energy Landscapes for Adenylate Kinase Conformational Change. *Nature Communications* **2010,** *1*, 111.
115. Davtyan, A.; Dama, J. F.; Voth, G. A.; Andersen, H. C., Dynamic force matching: A method for constructing dynamical coarse-grained models with realistic time dependence. *The Journal of Chemical Physics* **2015,** *142* (15), 154104.
116. Mullinax, J. W.; Noid, W. G., Generalized Yvon-Born-Green Theory for Molecular Systems. *Physical Review Letters* **2009,** *103* (19), 198104.
117. Lyubartsev, A. P.; Naômé, A.; Vercauteren, D. P.; Laaksonen, A., Systematic hierarchical coarse-graining with the inverse Monte Carlo method. *The Journal of Chemical Physics* **2015,** *143* (24), 243120.
118. Moore, T. C.; Iacovella, C. R.; McCabe, C., Derivation of coarse-grained potentials via multistate iterative Boltzmann inversion. *The Journal of Chemical Physics* **2014,** *140* (22), 224104.
119. Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H., The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *The Journal of Physical Chemistry B* **2007,** *111* (27), 7812-7824.
120. Dama, J. F.; Sinitskiy, A. V.; McCullagh, M.; Weare, J.; Roux, B.; Dinner, A. R.; Voth, G. A., The Theory of Ultra-Coarse-Graining. 1. General Principles. *Journal of Chemical Theory and Computation* **2013,** *9* (5), 2466-2480.
121. Tarenzi, T.; Calandrini, V.; Potestio, R.; Giorgetti, A.; Carloni, P., Open Boundary Simulations of Proteins and Their Hydration Shells by Hamiltonian Adaptive Resolution Scheme. *Journal of Chemical Theory and Computation* **2017,** *13* (11), 5647-5657.

122. Agrawal, R.; Srikant, R., Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers Inc.: 1994; pp 487-499.
123. Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y., An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002,** *24* (7), 881-892.
124. Selim, S. Z.; Ismail, M. A., K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1984,** *PAMI-6* (1), 81-87.
125. R. Ackermann, M.; Blömer, J.; Sohler, C., *Clustering for metric and non-metric distance measures*. 2008; Vol. 6, p 799-808.
126. Arthur, D.; Vassilvitskii, S., k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics: New Orleans, Louisiana, 2007; pp 1027-1035.
127. Jones, E.; Oliphant, T.; Peterson, P. Scipy v1.0.0 Reference Guide. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html (accessed 04/02/2018).
128. Vrahatis, M. N.; Boutsinas, B.; Alevizos, P.; Pavlides, G., The New k-Windows Algorithm for Improving thek -Means Clustering Algorithm. *Journal of Complexity* **2002,** *18* (1), 375-391.
129. Lloyd, S., Least squares quantization in PCM. *IEEE Transactions on Information Theory* **1982,** *28* (2), 129-137.
130. Ackermann, M.; Raupach, C.; Lammersen, C.; Sohler, C.; Märtens, M.; Swierkot, K., StreamKM++: A Clustering Algorithm for Data Streams. In *2010 Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments (ALENEX)*, Society for Industrial and Applied Mathematics: 2010; pp 173-187.
131. Yuan, Y.; Tam, M. F.; Simplaceanu, V.; Ho, C., New look at hemoglobin allostery. *Chem Rev* **2015,** *115* (4), 1702-1724.
132. Zakharova, E.; Horvath, M. P.; Goldenberg, D. P., Structure of a serine protease poised to resynthesize a peptide bond. *Proceedings of the National Academy of Sciences* **2009,** *106* (27), 11034.
133. Morris, A. M.; Watzky, M. A.; Finke, R. G., Protein aggregation kinetics, mechanism, and curve-fitting: A review of the literature. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* **2009,** *1794* (3), 375-397.
134. Shaw David, E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror Ron, O.; Eastwood Michael, P.; Bank Joseph, A.; Jumper John, M.; Salmon John, K.; Shan, Y.; Wriggers, W., Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010,** *330* (6002), 341-346.
135. Husic, B. E.; Pande, V. S., Markov State Models: From an Art to a Science. *Journal of the American Chemical Society* **2018,** *140* (7), 2386-2396.
136. Malmstrom, R. D.; Lee, C. T.; Van Wart, A. T.; Amaro, R. E., Application of Molecular-Dynamics Based Markov State Models to Functional Proteins. *Journal of Chemical Theory and Computation* **2014,** *10* (7), 2648-2657.
137. Cassidy, C. S.; Lin, J.; Frey, P. A., A New Concept for the Mechanism of Action of Chymotrypsin: The Role of the Low-Barrier Hydrogen Bond. *Biochemistry* **1997,** *36* (15), 4576-4584.

138. Brand, M.; Oliver, N.; Pentland, A. In *Coupled hidden Markov models for complex action recognition*, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 17-19 June 1997; 1997; pp 994-999.
139. Hempel, T.; del Razo, M. J.; Lee, C. T.; Taylor, B. C.; Amaro, R. E.; Noé, F., Independent Markov decomposition: Toward modeling kinetics of biomolecular complexes. *Proceedings of the National Academy of Sciences* **2021,** *118* (31), e2105230118.
140. Noé, F.; Wu, H.; Prinz, J.-H.; Plattner, N., Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules. *The Journal of Chemical Physics* **2013,** *139* (18), 184114.
141. Mardt, A.; Pasquali, L.; Wu, H.; Noé, F., VAMPnets for deep learning of molecular kinetics. *Nature Communications* **2018,** *9* (1), 5.
142. Rosenblatt, F., *The perceptron, a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory: 1957.
143. Nair, V.; Hinton, G. E. In *Rectified linear units improve restricted boltzmann machines*, Icml, 2010.
144. Rudin, C., Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* **2019,** *1* (5), 206-215.
145. Yap, W. T.; Saroff, H. A., Application of the ising model to hemoglobin. *Journal of Theoretical Biology* **1971,** *30* (1), 35-39.
146. Potts, R. B., Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society* **2008,** *48* (1), 106-109.
147. Anishchenko, I.; Kundrotas, P. J.; Vakser, I. A., Contact Potential for Structure Prediction of Proteins and Protein Complexes from Potts Model. *Biophys J* **2018,** *115* (5), 809-821.
148. Herbert, R.; Sutton, M., A Stochastic Approximation Method. *The Annals of Mathematical Statistics* **1951,** *22* (3), 400-407.
149. Hastings, W. K., Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970,** *57* (1), 97-109.
150. McGibbon, Robert T.; Beauchamp, Kyle A.; Harrigan, Matthew P.; Klein, C.; Swails, Jason M.; Hernández, Carlos X.; Schwantes, Christian R.; Wang, L.-P.; Lane, Thomas J.; Pande, Vijay S., MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal* **2015,** *109* (8), 1528-1532.
151. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M., Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* **2016**.
152. Islam, M. M.; Sohya, S.; Noguchi, K.; Yohda, M.; Kuroda, Y., Crystal structure of an extensively simplified variant of bovine pancreatic trypsin inhibitor in which over one-third of the residues are alanines. *Proceedings of the National Academy of Sciences* **2008,** *105* (40), 15334.
153. Borjigin, J.; Nathans, J., Bovine pancreatic trypsin inhibitor-trypsin complex as a detection system for recombinant proteins. *Proceedings of the National Academy of Sciences* **1993,** *90* (1), 337.
154. Baum, L. E.; Petrie, T.; Soules, G.; Weiss, N., A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics* **1970,** *41* (1), 164-171.

155. Betancourt, M. In *Cruising the simplex: Hamiltonian Monte Carlo and the Dirichlet distribution*, AIP Conference Proceedings 31st, American Institute of Physics: 2012; pp 157-164.