Using Persistent Homology for Topological Analysis of Protein Interaction Network of Candida Antarctica Lipase B Molecular Dynamic Simulation Model

by

Samin Tajik

B.Sc., Shahid Beheshti University, 2014
M.Sc., Brock University, 2018

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Mathematics and Sciences

Department of Physics

BROCK UNIVERSITY

January 12, 2023

(Signature) ______________________

Department of Physics

Brock University
St.Catharines, Canada

Date ______________________

Abstract

In this work, we aim to examine the activity of one of the most efficient and commonly used lipases, Candida Antarctica Lipase B (CalB), from the perspective of multiple computational techniques. To this end, we first conduct a series of Molecular Dynamics Simulations on CalB in different conditions to analyze the conformational changes of the protein and probe its unusual high-temperature activity. Next, we build the protein interaction network of amino acids for CalB to study pairwise interactions between amino acids (nodes) and probe the protein in terms of statistical features of links' distribution. Finally, we employ an algebraic topology-based method to study the protein interaction network from a broader perspective. The "Persistent Homology (PH) method" is then presented as a way to exceed pairwise interactions and examine protein networks in terms of patterns of interaction between the nodes. Persistent Homology studies the evolution of the protein interaction network's topological features (homology groups) in different states. Employing topological analysis, we compare the active form of CalB at high temperatures to its inactive states to account for possible topological contributions to the protein functionality. By discovering a prominent 1-dimensional hole in the active form of the protein, we highlight the role of higher-order interaction patterns in the network. Moreover, using the evolution of topological features, we study topological changes in protein networks and show the decline in the total number of 1-dimensional features as the protein loses activity and compactness over time. Accordingly, we propose that the protein's general conformational changes and three-dimensional structure are not the only facets contributing to its active state. Instead, we suggest examining the topology of the protein interaction network, referred to as different dimensional holes of the networks, as a higher dimensional analysis should be used to account for protein functionality. Hence, in this work, we desire to present that one needs to consider topological features acting as patterns of interaction between the components to study, examine or predict the folding of polypeptide chains into active structures.

Contents

List of Figures

Acknowledgements

I wish to express my sincere gratitude to my advisor and the chair of physics Prof. Thad Harroun. I appreciate the time, wisdom, and patience you were willing to devote in these years to grant me an excellent opportunity to follow my goals. It was an honor working under your supervision. Besides my supervisor, I would like to thank my committee members, Prof. Kirill Samokhin and Prof. Edward Sternin, for supporting me in understanding the subjects and helping me greatly with their helpful advice and recommendations. Moreover, I would like to extend my sincerest thanks and appreciation to Prof. Fereidoon S. Razavi and Prof. Maureen Reedyk for always believing in me and helping me through this journey. Last but not least, I want to thank my mother and father for always being there for me during this path.

Chapter 1

Introduction: Background of the Research

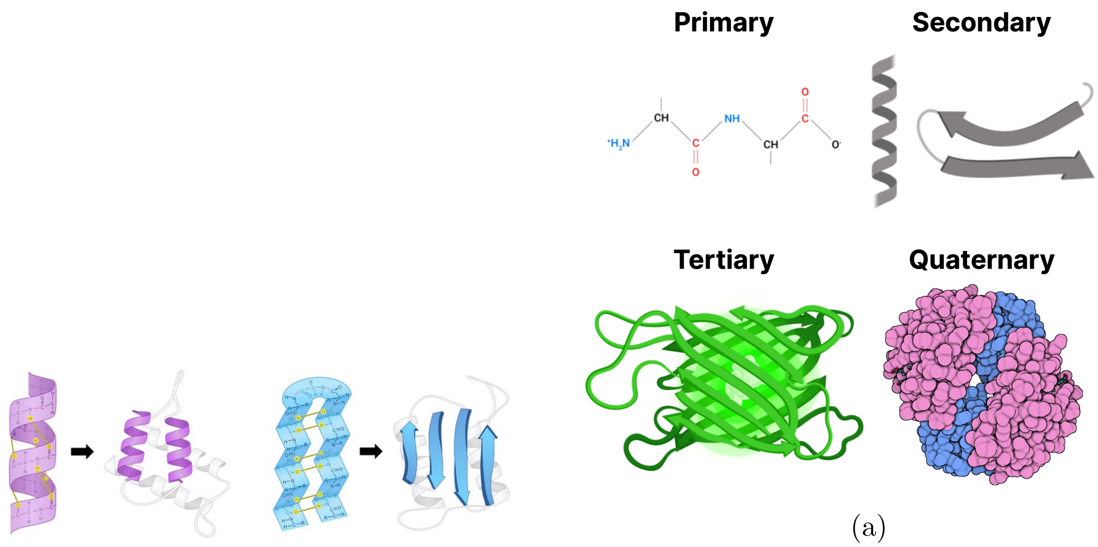
1.1 Proteins : Building Blocks of Life

Lipids, proteins, and carbohydrates are the primary constituents of the architecture of a living organism. Billions of microscopic molecular machines are working inside every cell in our body, allowing our eyes to detect light, our neurons to fire, and the 'instructions' in our DNA to be read. These tiny machines or major components of all living systems are proteins. Proteins are found in all living systems, from bacteria and viruses to the unicellular organisms and higher mammals, underpinning not just the biological procedures in our body but every biological procedure in every living thing. They're the building blocks of life.

Proteins are biological macromolecules that are involved intimately in the biological process for all the chemical reactions occurring in cells, and hence, can function as enzymes, hormones, receptors, channels, transporters, antibodies, and support structures inside and outside cells. Based on their sequence in the polypeptide chain, protein is unique in fulfilling its role. Properly folded proteins consisting of a precise sequence of amino acids, folds into a particular 3D structure required to function correctly. Attraction and repulsion between 20 different amino acids yield these strings to fold and form curls, loops, and pleats.

Levels of Protein Folding

There are several levels of protein folding, each with a particular type of bonding, turning the amino acid sequence (primary structure) into its final 3D structure- the tertiary structure and quaternary structure of various chains. Protein secondary structure refers to the protein's backbone's local conformation, stabilized by intramolecular and intermolecular (hydrogen bonding) interactions, and contains two prevalent types: Alpha Helices and Beta Sheets (see Fig 1.1), which are the initial state of the folding process. The alpha helix (α-helix) retains a right-handed spiral conformation, with every backbone N-H group hydrogen bonds to the backbone C=O group of the amino acid located four residues earlier along the protein sequence. Beta sheets made



(a)

Figure 1.1: **a. Protein Secondary Structure** consisting alpha helices coloured in purple and beta strands coloured in blue [7]. **b. Different levels of protein structure** from primary structure of amino acid sequence, secondary structure of helices and sheets, and tertiary structure of 3D structure. Protein quaternary structure refers to proteins that themselves are composed of multiple protein chains [2]

of two or more parallel or anti-parallel adjacent beta-strands (β-strand) is a stretch of polypeptide chains, typically 3 to 10 amino acids long, with almost fully extended backbones[18].

Therefore, we need to focus on their three-dimensional structure and corresponding structural characteristics to study protein function. Although prediction of the precise configuration of proteins and resolving how these long-chain peptides have folded into their stable configuration is still a controversial problem in science, any improperly folded, unfolded, or denatured protein will lose its functionality. For example, a very recent study by a team of Stanford University researchers[13] revealed the relationship between aging and protein aggregation, suggesting that by perturbing the machinery that preserves the stability of some proteins, animals tend to age quickly, and if the quality control pathways are enhanced genetically, they tend to live longer. While no one knows how to predict the folding of these long peptides into their ultimate 3D structure, these harmful effects of perturbing the stable conformations and disturbing the function of different proteins has been the subject of many recent studies[45, 8].

1.2 Candida Antarctica Lipase B (CalB)

Enzymes are a type of proteins that perform as biological catalysts, increasing the rate of a reaction by lowering the reaction's activation energy, and lipases are versatile enzymes hydrolyzing fat distributed throughout living organisms. Among lipases, Candida Antarctica lipase B(CalB) is a widely studied lipase with numerous registered patents and applications in the pharmaceutical, chemical, and food industries [9, 23]. CalB is a significantly efficient catalyst for hydrolysis of carboxyl ester bonds at low temperature in water and esterification in organic solvents with high thermal stability [23, 45].

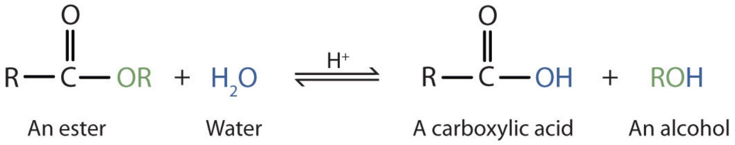


Figure 1.2: **Hydrolysis Reaction Mechanism** [1]

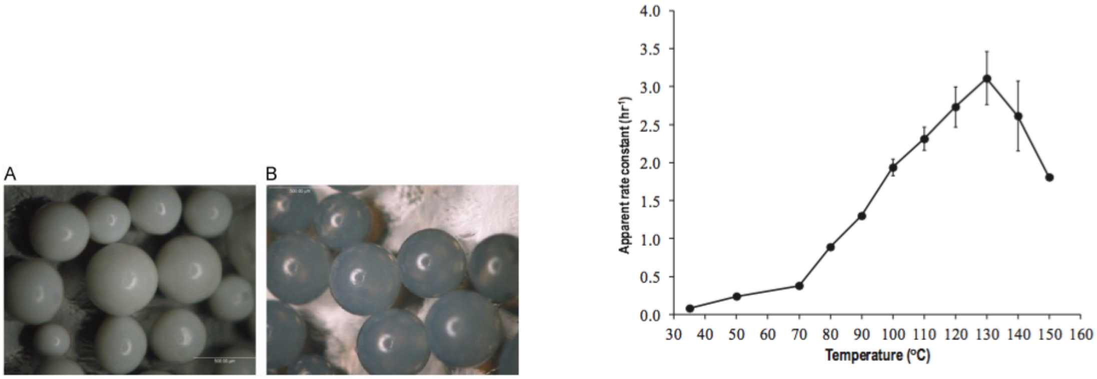
CalB is a α/β-hydrolase type protein composed of 317 amino acids and a secondary structure of seven β-sheets and 10 α-helices [38] is used in a variety of industries due to its compatibility with a wide range of substrates, thermal stability, and stability in organic solvents[52].

1.3 Protein Active Site and Catalytic Triad of CalB

The part of the enzyme where the substrate molecules (substance on which an enzyme acts) bind and undergo a chemical reaction is called the active site of proteins. If the shape of the active site were changed by an external agent, like a change in temperature or pH, the enzyme would not be able to catalyze the reaction. The 3D structure of each protein molecule provides a framework to confirm that the active site, containing multiple amino acids, is precisely in the proper orientation and all of these active site residues are in the appropriate configuration relative to one another. Hence, the interaction between a substrate and enzyme depends on both the active site's physical shape and the active site's chemistry, including the performance of hydrogen bond donors and acceptors. CalB as a member of the α/β hydrolase fold family is known to utilize the serine (Ser)- histidine (His)- aspartate (Asp) triad as its catalytic site [49, 38], and the analysis of the CalB catalytic triad consisting of S105, D187, and H224 is one of the essential ways to examine the activity of the protein under different conditions.

1.4 Experimental Background of Thermal Stability of CalB

Critical operational considerations, such as preventing denaturation at high temperatures being of significant concerns to overcome when generalizing the use of enzymes on an industrial scale, have made CalB one of the most used lipases. According to different studies, the immobilized form of CalB is quite thermostable, particularly under nonaqueous conditions where the catalyst remains active for many hours in the presence of high concentration of reactants. In aqueous solutions, however, the lipase becomes inactive quickly at temperatures as low as 40°C. Explaining the thermostability of CalB in the non-aqueous environment and improving the thermal stability of the enzyme without negatively affecting its activity has been the subject of many recent studies [58]. In previous study by Frampton et al.[20], where the immobilized form of CalB (commercially available under Novozyme 435-N435) was proven to be quite thermostable under non-aqueous conditions, CalB esterification proceeded without bulk solvent, and the reaction rate increased with increasing temperature up to more than 130°C. The protein was active up to 150°C, as shown in Fig. 1.3b.



(a)

(b)

Figure 1.3: **a. Lewatit beads** where CalB is immobilized on the surface. **b. Esterification reaction rate** as a function of temperature measured by Frampton et al [20].

The authors could not account for the high thermal stability of the enzyme as the reactions proceeded well over 100°C, and this was our initial motivation to adopt multiple computational methods to examine the protein at the structural level to uncover possible contributing factors to this unusual behavior. To examine its activity at temperatures that would destroy most proteins, we used molecular simulation techniques to analyze the structural behaviors and microstructural properties. We also probed the protein topologically at the network level to reveal its topology-function relationship.

1.5 Layout of the Thesis and Computational Approaches to Study the Protein

In this study, through multiple computational techniques, we attempt to uncover what is unique about this protein allowing it to be active in temperatures that destroy most other proteins. Our discussions, results, and analysis using different methods are in the following direction:

- Chapter 2: Through a series of **Molecular Dynamics Simulations**, we first study CalB on the molecular scale, demonstrate its unfolding process in extreme conditions and highlight its stable and active state and conformation at high temperatures. We also probe the active site of the protein and examine the possibility of active site shielding to explore the contributing factors in high-temperature activity of CalB.

- Chapter 3: In chapter 3, we introduce the **Theory of Complex Systems and Complex Networks**, and from the three-dimensional configuration, we build the Protein Residue Network of CalB (Protein Interaction Network). We compare some statistical features of the complex network of CalB in different states and look at its active form in high temperatures from a new approach. The protein interaction network uses amino acids as nodes and studies pairwise interaction between the pairs. The statistical analysis of pairwise interactions in different states studies and compares the distribution of bonds between residues in the active and inactive states of the protein. These networks provide input for further analysis in the later sections.

- Chapter 4: In chapter 4 of the thesis, we introduce a new computational technique based on Algebraic Topology to study the protein interaction network of CalB. Using **Topological Data Analysis** with **Persistent Homology Approach**, we search for the global and local effects of topological fingerprints on the unusual activity of CalB in high temperatures. Topological features (fingerprints) examine the evolution of homology generators in different dimensions (connected components, holes, and voids) and compare the general topology of the protein in different states. By topological analysis, we go beyond pairwise interaction between amino acids and probe the patterns of interactions of the network. We believe that these higher-order interactions between amino acids (pattern of interactions) play an essential role in studying the protein folding process to the final 3D configuration of proteins. Hence, in exploring the protein interaction networks of proteins, other than pairwise interactions, one must consider the interaction pathways between the nodes representing themselves in the network's topological features.

- Chapter 5: Chapter 5 contains conclusion and future works suggestions.

Chapter 2

Method of Molecular Dynamics Simulation and Results

2.1 Basics of Molecular Dynamics Simulation

Most of the computer simulations are based on the assumption that the motions of atoms and molecules (particles) can be explained by the laws of classical mechanics[21]. Molecular Dynamics (MD) simulation is a well-established computational method that aims to compute the equilibrium and transport properties of classical many-body systems. In recent years MD simulation is used to describe the dynamical properties of proteins and other macromolecules to provide structural interpretations of experimental data [21]. The basic strategy of MD simulation is to numerically solve Newtonian equations of motion in a Laplacian framework. Noting that in a deterministic universe, knowing the precise location and momentum of any particle (or mass) in the universe, their past and future position and momentum for any given time can be determined from the laws of classical mechanics. MD simulations in many aspects function very similarly to real experiments. In MD simulation a model system consisting of N atoms will be chosen, for which the Newtonian equations of motion will be solved for the system until it reaches its equilibrium state. For many classical systems, the equations of motion for a system of N interacting atoms with positions r_i and masses m_i take the following form:

$$m_i \frac{\partial^2 r_i}{\partial t^2} = F_i, \quad i = 1...N,: \tag{2.1}$$

in which the forces acting on the atoms are derived from:

$$F_i = -\frac{\partial V}{\partial r_i}, \tag{2.2}$$

where V is the potential function, including non-bonded pairwise interaction terms:

$$V(r_1, ...r_N) = \sum_{i<j} V_{ij}(r_{ij})$$

such as Lennard-Jones, and Coulombic and also intramolecular bonding interactions. For a Lennard-Jones system the non-bonded part of the potential function is defined by:

$$V^{(LJ)}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{2.3}$$

where r_{ij} is the distance between two interacting particles, σ is a measure of the range of the potential and ϵ is the strength (depth of the potential well, also known as dispersion energy). The LJ potential is positive (repulsion) for small values of r, and negative (attraction) for large values, with a minimum of $-\epsilon$ at $2^{1/6}\sigma$. If electrostatic charges exist we need to add the appropriate Coulomb potential term.

$$V^{(Coulomb)}(r_{ij}) = \frac{Q_i Q_j}{4\pi\epsilon_0 r_{ij}}, \tag{2.4}$$

where Q_i and Q_j are the charges and ϵ_0 is the permittivity of free space. To overcome computational difficulties, program always uses a cut-off radius for LJ and sometimes for Coulomb interaction such that they become zero beyond a certain cut-off distance r_c. To define the value for cut-off, MD simulation softwares such as Gromacs [34] defines the minimum-image convention technique, which considers only one image of each particle in the periodic boundary conditions for a pair interaction, so the cut-off radius cannot exceed half the box size. For simulation of molecules program also needs to consider the intramolecular bonding interactions defined as:

$$\begin{aligned} V_{intramolecular} = \frac{1}{2} \sum_{bonds} K^{r}_{ijk}(r_{ij} - r_{eq})^2 + \frac{1}{2} \sum_{bendangles} K^{\theta}_{ijk}(\theta_{ijk} - \theta_{eq})^2 + \\ + \frac{1}{2} \sum_{torsionangles} \sum_{m} K^{\phi,m}_{ijkl}(1 + cos(m\phi_{ijkl} - \gamma_m)). \end{aligned} \tag{2.5}$$

As it can be noted from the equation, each “bond”, taking a harmonic form (quadratic) with a defined equilibrium separation, considers the separation between adjacent pairs of atoms, and each “bend angle” is constructed between successive bonds, considering three atom coordinates. The “torsion angles” then are defined in terms of three connected bonds and four atomic coordinates. In order to conduct Molecular Dynamics Simulation we used GROMACS [34] Molecular Dynamics Simulation Package which is mainly designed for simulations of proteins. In order to simulate the dynamics of proteins GROMACS defines variety of ”Forcefields” which are computational methods that can be used to estimate the forces between atoms within molecules and also between molecules. A forcefield of any simulation package will specify the strength parameters, intramolecular potentials and all the other necessary constants such as parameters needed for any solvent that is used in the simulation. When a force acting on each particle has been acquired at a specific time t, numerical integration of the equations of motion, employing several algorithms yields a new particle position at a time $t + \delta t$. Where δt will be dictated by the algorithm of the program. To begin the simulation, one should set initial positions and velocities for all particles in the system. These conditions are chosen consistent with the structure that is being simulated. For MD simulation the coordinates are usually extracted from an experimentally determined crystal structure deposited in a public database. If the initial velocities are not known the program generates initial atomic velocities $v_i, i = 1..3N$ from Maxwell-Boltzman velocity distribution:

$$\rho(v_i) = \sqrt{\frac{m_i}{2\pi kT}} \exp\left(-\frac{m_i v_i^2}{2kT}\right). \tag{2.6}$$

Note that since the system is equilibrated in the canonical ensemble and will naturally tend to its equilibrium state, this choice of initial velocity does not affect the result of the simulation. The temperature of the simulation is found by the total kinetic energy of the N -particle system, and can be controlled using the method of temperature coupling to an external heat bath. If the resulting total energy will not correspond exactly to the required temperature T, a correction is made through scaling all velocities so that the total energy corresponds exactly to T.

Algorithm and Application

We require a good algorithm in order to integrate Newtonian equations of motion and obtain the system trajectory. For a proper MD simulation, an acceptable accuracy for relatively larger time steps is important since a longer time step is equivalent to fewer evaluations of the forces and a more efficient simulation. The default MD integrator in GROMACS the so-called leap-frog algorithm uses positions os atoms at time t and velocities at time $t - \frac{1}{2}\delta t$, and updates positions and velocities, using $F(t)$. When extremely accurate integration with temperature and/or pressure coupling is needed, the preferred method is velocity Verlet integrators[34]. In our simulation we used Verlet method as for most MD applications, Verlet-like algorithms are adequate; however, occasionally it is convenient to employ higher-order algorithms. In velocity Verlet method, positions r and velocities v at time t are used to integrate the equations of motion:

$$r(t + \delta t) \;=\; r(t) + \delta t v + \frac{\delta t^2}{2m} F(t) \tag{2.7}$$

$$v(t+\delta t) \;=\; v(t) + \frac{\delta t}{2m}[F(t) + F(t+\delta t)]. \tag{2.8}$$

Provided that the potential range is cut-off, the program then applies periodic boundary condition which allows each atom to interact with their nearest image in a periodic array. Hence, the procedure to build and simulate our system and analyze its dynamics contained:

- Obtaining the initial structure.
- Selecting the preferred forcefield based on experimental data and generating the starting topology of the system.
- Choosing the appropriate solvent and neutralizing the system, this step contains adding ions to the system to cancel the extra charge.
- Relaxing the system at its minimized energy and equilibrating it to the desired temperature and pressure using NVT and NPT ensembles.
- Conducting the time evolution simulation and analyzing the results.

Candida Antarctica Lipase B (CalB) crystal structure used in our Molecular Dynamics simulation is obtained from the protein data bank [53] under PDB accession code: 1TCA which the experimental crystal structure was obtained by X-RAY diffraction and the 50 *ns* simulations are carried out using GROMACS software [34] using CHARMM36m forcefield, which is known to be an enhanced forcefield for folded and disordered proteins [26]. The protein is centred in a simple cubic box with $0.1nm$ from the box edges as the unit cell, and solvated using GROMACS TIP3P water model, $8M$ urea, and glycerol as different solvents. After neutralization and energy minimization using the steepest-descent algorithm the system equilibration and heating to the desired temperature of 300K, 323K, 423K in water, 350K and 423K in glycerol and 423K and 480K in urea and pressure of 1 bar was through an NVT and NPT ensemble respectively.

2.2 Results and Analysis of Molecular Dynamics Simulations of CalB

In order to analyze the results and quantify the final topology and conformational changes of the structure, after the 50 ns simulation is carried out, we examine the snapshots of the protein in time schematically, calculate the root-mean-square deviation (RMSD), which quantifies the structural stability concerning the starting

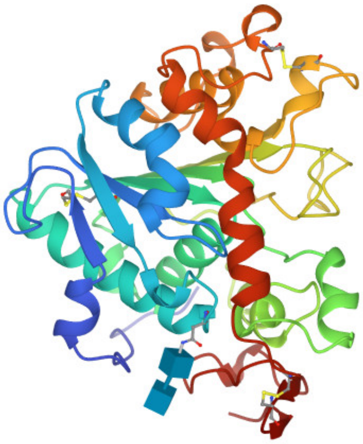


Figure 2.1: **1TCA**, LIPASE B FROM CANDIDA ANTARCTICA, downloaded from Protein Data Bank [53].

structure of the protein and radius of gyration (R_g) values of CalB structure in different temperatures and plot these quantities as a function of time. We also study the active site of CalB in terms of the distances between active site residues to account for the activity of CalB in different states.

The root mean square deviation RMSD of certain atoms in a molecule relative to a reference structure (usually the backbone atoms), can be calculated as:

$$\mathrm{RMSD}(t) = \left[\frac{1}{M} \sum_{i=1}^{N} m_i \big| r_i(t) - r_i^{\mathrm{ref}} \big|^2 \right]^{1/2}, \tag{2.9}$$

where $M = \sum_i m_i$, r^{ref} is the reference structure and $r_i(t)$ shows the position of atom i at time t. As proteins can be fitted on the backbone atoms and accordingly RMSD will follow to be computed for the backbone, in our calculations we have plotted the RMSD and analyzed its deviation with respect to the backbone of CalB.

Furthermore, in order to have an estimate for the compactness of a structure, one can compute the radius of gyration, which is defined as:

$$R_g^2 = \left(\frac{\sum_i^N |(r_i - R_{cm})|^2 m_i}{\sum_i m_i} \right), \tag{2.10}$$

where m_i is the mass of atom i and r_i stands for the positions of atoms with respect to the center of mass of the molecule $R_{cm} = N^{-1} \sum_{i=1}^{N} r_i$. If a protein is stably folded,

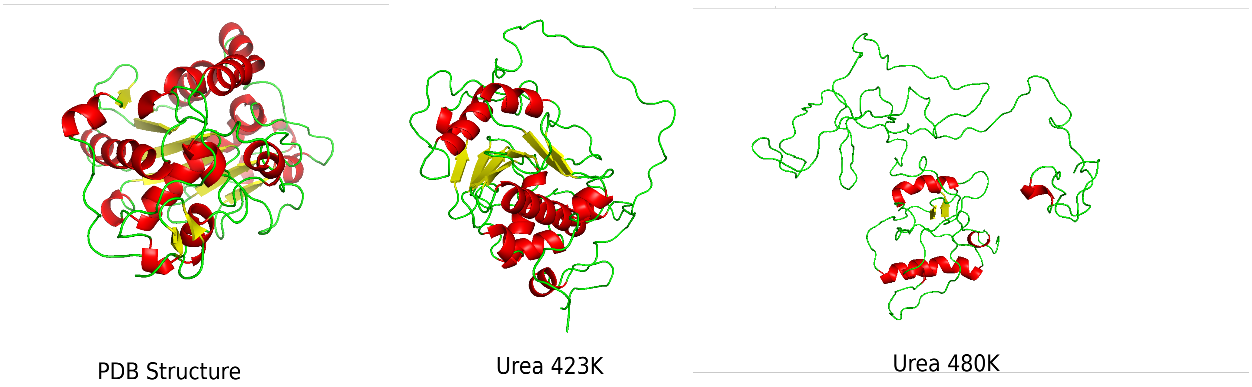


Figure 2.2: **Starting structure and snapshots of overall CalB**, after 50 ns simulation in Urea at 423 K and 480 K.

and compact the R_g value will likely maintain a relatively steady value, and as it starts to unfold, the corresponding R_g value is expected to increase as a function of time, and to decrease during protein folding. In the following sections we are going to use RMSD and $\mathrm{R_g}$ analysis together with an examination of the active site of CalB to account for its conformational stability and activity in different conditions.

2.2.1 Results of MD Simulation of CalB in Urea

Regarding the analysis of the molecular dynamics simulation of protein, our first step was to show how denaturation looks like in computer simulation. In the 1930s, urea became the most typically used denaturant agent in protein folding - unfolding studies. It has been suggested that by forming hydrogen bonds with protein amino acid side chains, urea induces protein denaturation [3]. To simulate the unfolding process of CalB, we followed a paper by Monhemi and colleagues where the protein was placed in a box of $8M$ urea and water [38], and conducted a series of $50ns$ MD simulations, subjecting CalB to high-temperature simulations at $423K$ and $480K$. In order to schematically represent the final and unfolded state of CalB snapshots of the overall CalB structure after the $50ns$ simulation at both high temperatures are depicted in Fig 2.2. The figure confirm that the protein unfolds and accordingly loses most of its secondary structure in Urea. For CalB in Urea at $480K$ the overall structure of CalB is further lost due to complete unfolding at this high temperature in the presence of urea.

Fig. 2.3 shows the evolution of the RMSD in time, as a good quantitative measure

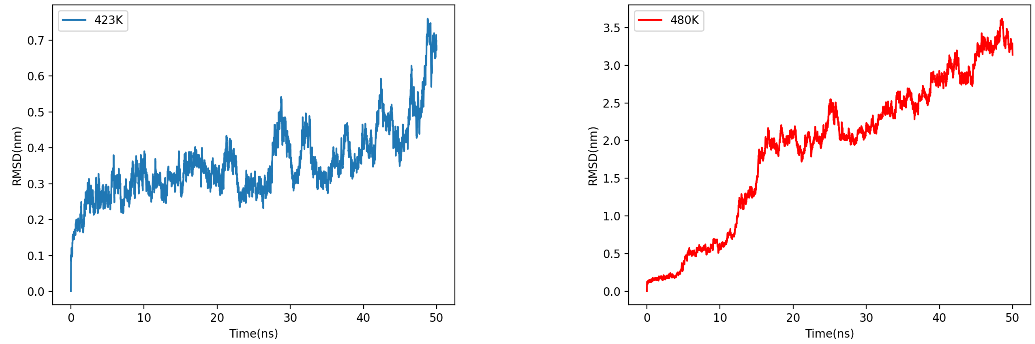


Figure 2.3: **Root Mean Square Deviation** RMSD, of CalB in urea after 50 ns simulation at 423K and 480K.

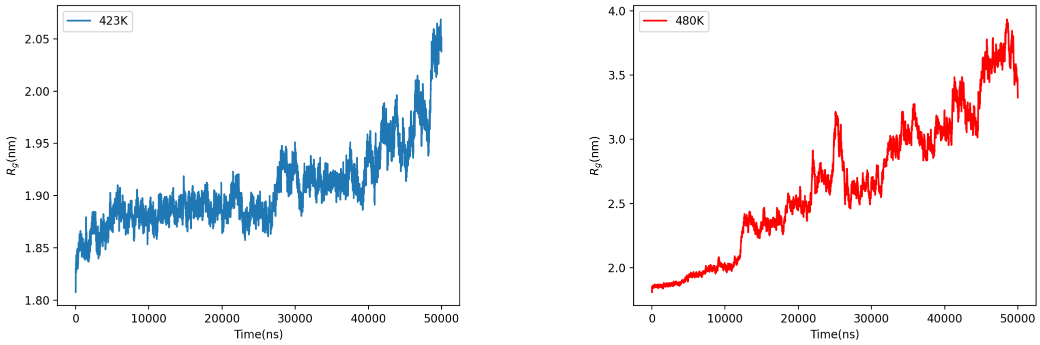
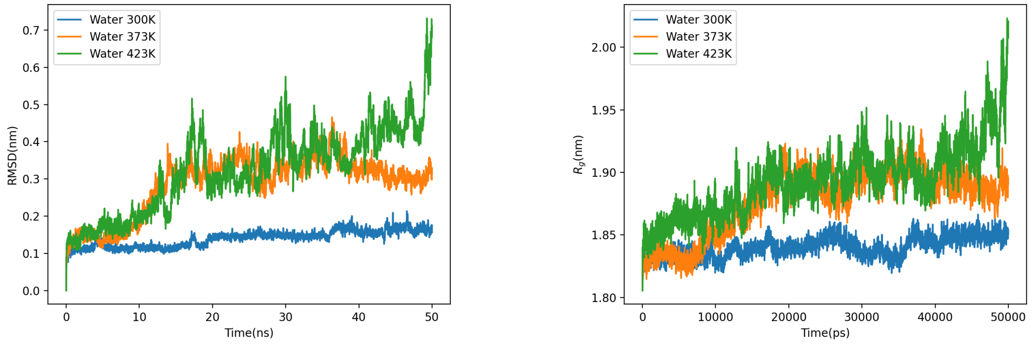


Figure 2.4: **Radius of Gyration** R_g, of CalB in urea after 50 ns simulation at 423K and 480K.



(a)

(b)

Figure 2.5: RMSD **and R_g of CalB** , after 50 ns simulation in pure water at different temperatures.

for protein unfolding and losing its compactness in both $423K$ and $480K$. As shown, the value of RMSD gradually increased and reached its highest values after 50 ns, in both temperatures. For CalB in urea at $480K$ as expected, this increase was dramatically larger as the protein loses more of its secondary and tertiary structure. For further studying the compactness of CalB, we then analyzed the radius of gyration for CalB in our simulations, and plotted it as a function of time in Fig 2.4. The value of R_g increases from about $1.8nm$ to $2.05nm$ and $4.0nm$ for CalB in urea at $423K$ and $480K$ respectively. The high deviation value for RMSD, and R_g, together with the schematic representation of CalB in urea, confirmed the denaturation of CalB in urea both at 423K and at 480K. Please note that the unfolding process doesn't stop, and we expect further unfolding of protein as time exceeds 50ns.

2.2.2 Results of MD simulation of CalB in Water

The hypothesis of high-temperature stability and activity of CalB is then tested by placing the protein in a box of pure water. We used TIP3P water model of the GROMACS package[34], and conducted another set of $50ns$ MD simulations at various temperatures of 300K, 373K, 423K, and 480K. The RMSD and R_g values are plotted in Fig 2.5. RMSD values of CalB during the simulation in water for a series of temperatures are shown in Fig 2.5a. Likewise, equilibrium conformation of the system in water is also examined for protein structure compactness using the R_g value, which for a range of temperatures are plotted as a function of time for CalB in Fig 2.5b. As expected, the results of MD simulation in water show protein is more compact in water as compared to urea. In water at $300K$, protein remained compact with steady RMSD and R_g values. At $373K$ protein partially unfolds and RMSD and R_g values increase slightly to $0.3nm$ and $1.89nm$ respectively. In water at $423K$ protein seems to unfold by a large deviation and will further lose its compactness and stability as the values of RMSD and R_g increase to about $0.6nm$ and from $1.8nm$ to $2nm$ respectively.

2.2.3 High Temperature Activity of CalB in glycerol

Our next step was to choose a non-aqueous environment for high-temperature MD simulation. Glycerol was chosen for its alcohol groups, high boiling point, and the availability of trusted force-field parameters. Glycerol has three hydroxyl groups and can be efficiently used as solvent and acyl acceptor in the transesterification to produce the corresponding alcohol [56]. Immobilized CalB was previously used as a catalyst for transesterification where glycerol is utilized as both the solvent and the acyl acceptor in the kinetic resolution of ester racemates [15]. Accordingly, we repeated our high temperature $50ns$ simulations in this solvent and examined the conformational changes, stability, and compactness of the protein using the schematic snapshots representation, RMSD and R_g respectively. Fig 2.6 compares every $10ns$ snapshots of the protein in water and glycerol during the $50ns$ simulations in terms of protein secondary and tertiary structural loss and conformational change. As noted, protein remains folded both in water at $300K$ and in glycerol 423K and start losing its 3D structure in water at $423K$. In Fig 2.7 we have plotted the result of RMSD and R_g analysis in glycerol at two different temperatures. As we can compare the structure to the protein dynamics in water, we find that the overall structure of the protein remains compact and stable in glycerol even at high temperatures as the protein RMSD and R_g values follow same patterns for both water $300K$ and glycerol at $423K$. Hence, our results of RMSD and R_g, together with schematic representation of the protein confirmed that CalB is stable and compact in water at $300K$ and in glycerol both at $350K$ and remains active up to $423K$.

2.2.4 Active Site Residues and Analysis

As discussed previously in chapter 1, the active site of CalB is the conserved catalytic triad consists of Ser (S105), Asp (D187), and His (H224). In Fig2.8 we show the crystal structure of CalB in grey with active site residues coloured in red(His), yellow(Asp) and green(Ser). In the next step in order to analyze the bonding lengths of the triad, and study the activity of CalB in both solvents in different temperatures, we examined the minimum pairwise distances between the triad residues and plotted these quantities as a function of time. In the Ser His Asp catalytic triad, the three residues form a charge-transfer relay network, with His playing the central role

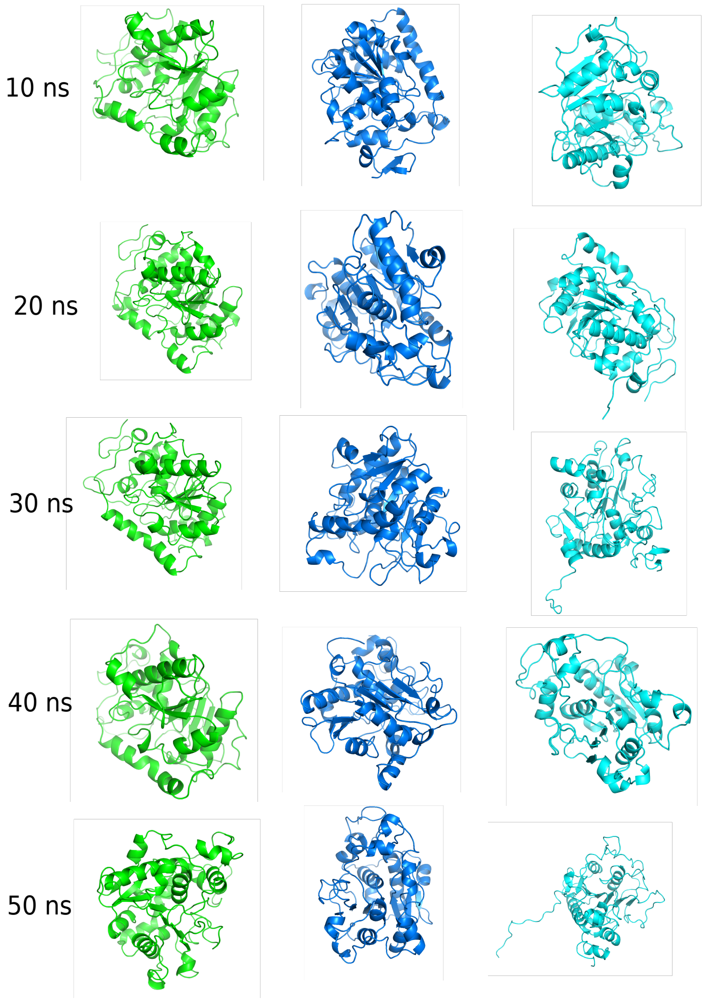


Figure 2.6: **Snapshots of CalB in different solvents.** From left to right: in glycerol at 423K, in water at 300K and in water at 423K

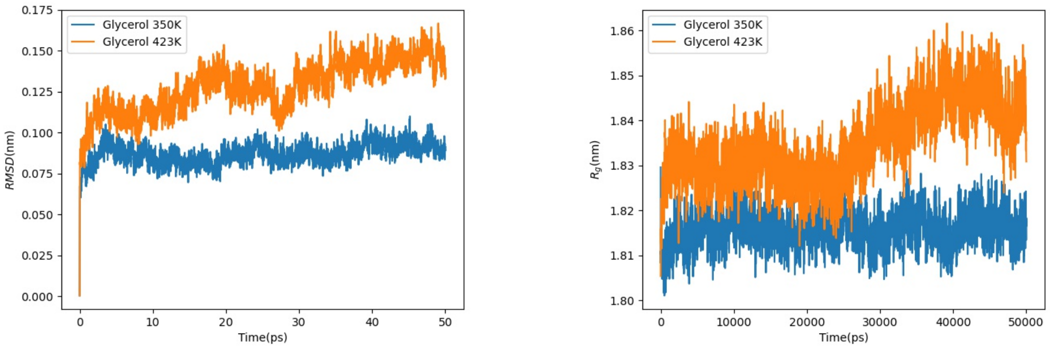


Figure 2.7: RMSD **and R_g of CalB**, after 50 ns simulation in glycerol at two different temperatures

and serine being hydrogen-bonded to Histidine, which is further hydrogen-bonded to Aspartate. Hence, in order to account for the activity of the protein, we will probe the pairwise distances between His to Ser and His to Asp to ensure that the triad is at the proper reach from one another and confirm the existence of hydrogen bonds between them.

As it can be noted from Fig 2.9, the active site residues of the protein seem to be moving away from one another in all three simulations of CalB in water in different temperatures. In water at low temperatures $300K$, even though the structure looks relatively compact, and stable according to RMSD and R_g values, additional examination of the active site shows that the catalytic triad minimum distances start to increase at the beginning of the simulation. To further analyze the high-temperature activity of CalB in glycerol in terms of active site we also studied the minimum distance between active site residues as a function of time for protein in glycerol at $423K$ and the corresponding results are plotted in Fig 2.10. As it can be noted from the figure the active site analysis of CalB in glycerol at high temperatures shows the distances between active site components in this solvent is stable even at high temperature which further confirms the protein activity in this condition. Accordingly, we conclude that in glycerol solvent, the conformation of the active site maintains its stability, whereas, in the polar solvents, even in lower temperatures $(300K)$ despite the compactness of the overall conformation, the interaction between water molecules and active site residues destroys the hydrogen bonding between catalytic triad as the distances are not remained stable.

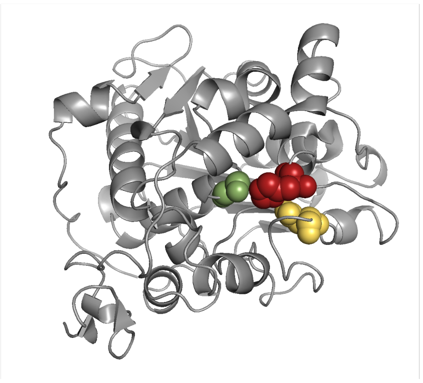


Figure 2.8: **Active Site Residues of CalB**, consisting His224 (red), Asp187 (yellow), Ser105 (Green)

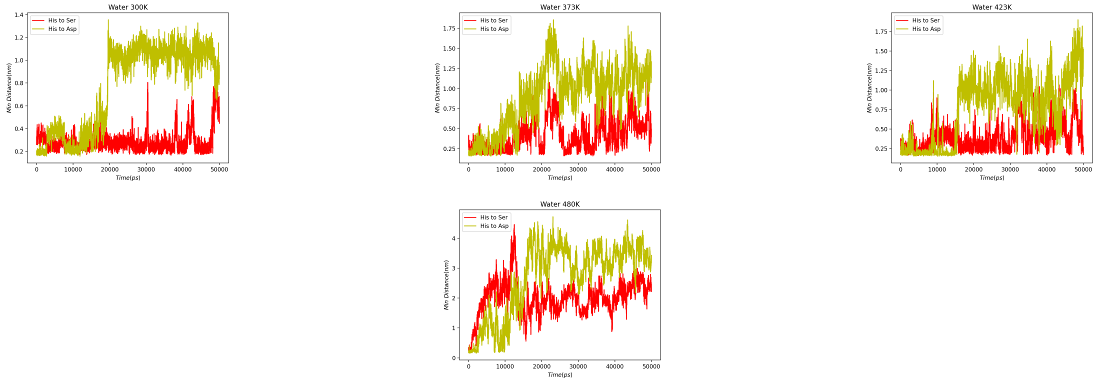


Figure 2.9: **Minimum Distance Between Active Site Residues**, after 50 ns simulation in pure water at different temperatures.

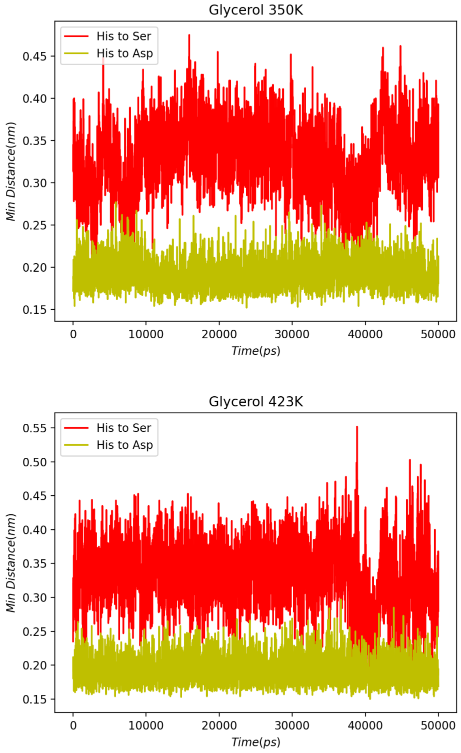


Figure 2.10: **Minimum Distance Between the Active Site**, after 50 ns simulation in glycerol at 350K and 423K.

2.2.5 CalB Lid and the Effect of Alpha-5 on the Activity of the Protein

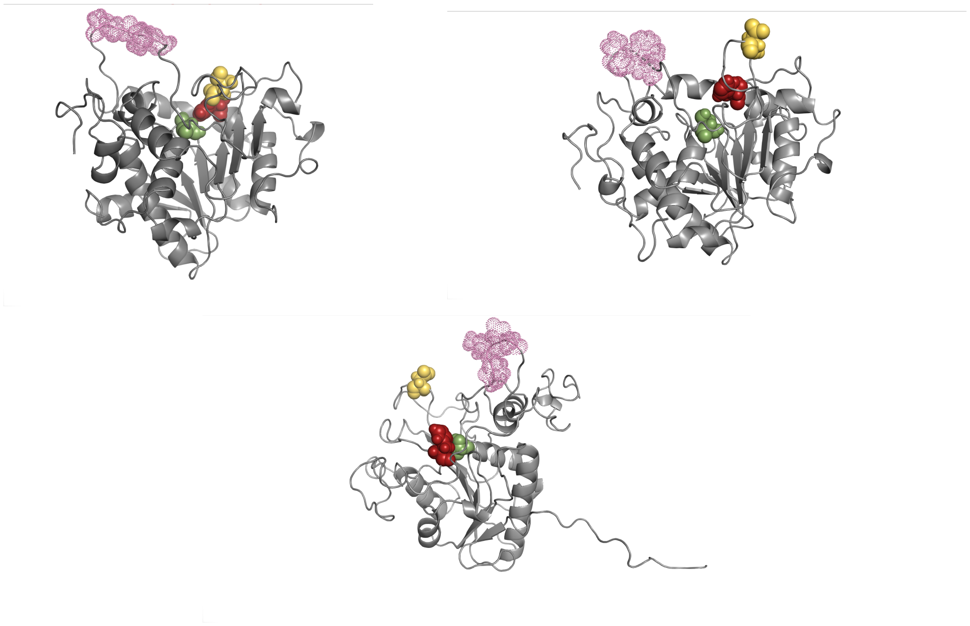


Figure 2.11: **Active state of the protein in glycerol** at 423K (top left). The active site residues are represented in the sphere with His in red, Asp in yellow-orange, and Ser in light green. The top right panel shows the folded state of CalB in water at 300K with active residues moved away, and the bottom panel represents the inactive and unfolded state of CalB in water at 423K. The Alpha-5 helix is colored in pink

Previous studies on the activity of CalB have shown that this enzyme has not shown any significant interfacial activation in prior experiments, and the further observations led to the determination that this behavior may stem from the absence of a lid. The lid of any enzyme controls the enzyme activity and is known to be responsible for protecting the active site. For CalB, this regulation of the access to the active site is still is a matter of controversy. However, it was also reported that CalB has two α-helixes α 5 and α 10 surrounding the active site, whose movement could play the role of the lid of the lipase. Their motions which is diminished in aqueous media significantly affect the catalytic properties of the enzyme ([48]). In another study by Kumaresan [33], the substantial role played by the flexible α5-helix in CalB is noted, where their results imply that the α5 helix in the native protein seems to unwind, and its movement allows it to partially cover the active site region. The study shows α5 and α10 helices, having a hydrophobic nature, present themselves in the path to the active site of the enzyme. Fig 2.11 shows 3 conformations of CalB in water in two temperatures and glycerol together with the position of active site residues (His224 in red, Asp187 in yellow, and Ser105 in green), and $\alpha 5$ helix colored in pink. The top left panel of the figure shows the folded (according to RMSD, and R_g results), and active state of the CalB in glycerol at $423K$, the catalytic triad is depicted in spheres with His colored in red, Asp in yellow-orange, and Ser in light green. The top right panel displays the folded state of CalB in water at $300K$ with active residues dragged away, and the bottom panel illustrates the inactive and unfolded state of CalB in water at $423K$. The schematic representation of our MD simulations results (Fig.2.11), showed that $\alpha 5$ unfolds in glycerol at 423K, where the protein appears to be folded and active, with active site residues held tight together, does not unfold in water at 300K and unfolds in water at 423K. As mentioned earlier, the existence of a lid and its role in the activity of CalB in high temperatures is still controversial. In the following chapters, we employ the method of complex system and topological analysis to look further into the underlying structure and topology-function relationship of CalB.

2.3 Chapter Conclusion and Remarks

To summarize this chapter, we want to highlight how the Molecular Dynamics simulation technique provided us with the starting point of our analysis and how its deficiencies instructed us to proceed to other approaches. Using MD simulation, after reproducing CalB unfolding process through simulation in urea, we analyzed the 3D structure and activity of the protein in water and glycerol. We used the results of RMSD and R_g to verify the folded state of CalB in water at 300K and glycerol at 423K. Furthermore, we analyzed the protein in terms of its active site residues' minimum distances from one another to probe the activity of the protein. Hence, we demonstrated the high-temperature activity of CalB in glycerol solvent and confirmed that despite the folded general configuration, protein becomes inactive in water even at lower temperatures around 300K. Consequently, we inferred that to examine the activity of the protein at high temperatures, account for its unusual behavior, and distinguish between folded-active (glycerol 423K) and folded-inactive (water300K) structures of CalB, we require a more suitable measure. Finally, after testing multiple computational methods, we employed the complex network analysis approach, which maps the protein system to a network of nodes and edges and examines it using pairwise interactions between the nodes.

Chapter 3

Complex System Theory and Protein Residue Network

3.1 Introduction to The Theory of Complex Systems

The science of complex systems is a remarkable mix of various subjects such as physics, biology, and social sciences. Complex system theory studies systems that contain many interacting elements from which various properties or functionality can emerge[50]. As we noticed earlier in MD simulation, forces are derivative of potentials, and classical systems trajectories can be fully understood and predicted using classical mechanics.

Contrary to physical methods where we don't specify which particle is interacting with which, interactions are particular in complex systems. If more than two elements interact, interactions will be described by time-dependent networks: $M_{ij}^{\alpha}(t)$, where i and j label the elements in the system and α is the interaction type. Networks characterize the strength and type of the interaction between elements of the system through correlation or interaction matrix elements [50]. The theory of complex networks at the heart of complex system theory uses these connected networks to track which elements interact with others in which ways. Complex network theory lies in the intersection between graph theory and statistical mechanics, employing networks to study the properties and dynamics of complex systems. In the following section, we introduce some important concepts and definitions used in complex network theory (based on [19, 42, 14]) to calculate some significant statistical features of the protein residue network of CalB in different states.

Statistical Features of Networks

In the simplest layout, a network is a set of points in space, joined together by some lines, which in the language of graph theory, are referred to as nodes (or vertices) and links (or edges) of the graphs. In the mathematical representation of graph theory, we define a network as: $M = (V, E, f)$, where V is a finite set of nodes, $E \subseteq V \otimes V$

is a set of edges, and f is a mapping [1] that associates elements of E to the binary set $\mathbb{Z}_2 = \{0, 1\}$:

$$f : E \rightarrow \mathbb{Z}_2 \tag{3.1}$$

The above definition for networks works well when we have "unweighted" networks. For any weighted network, in which the weights of the links take a range of values, we replace the map f by the weight function ω, and for a weighted network, we define $M = (V, E, \omega)$.

There are multiple different ways to represent networks of complex systems mathematically. In matrix representation for an undirected network (no direction is defined for edges) with N nodes, one creates a so-called Adjacency Matrix based on the list of all the edges between paired nodes (edge list). If we denote an edge between vertices i and j by $e_{ij} = (i, j)$, then one can determine the complete network by giving the value of N and a list of all edges.

Definition 3.1.1 (The Adjacency Matrix)**.** The adjacency matrix A of a simple graph is the matrix with elements A_{ij} such that:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Complex Network Theory classifies networks based on the nature of nodes and links. One of the most famous classes of networks that is the subject of our study is the one whose links between pairs of nodes represent interactions determined by physical forces based on distances, called geometric graphs. In this context, protein residue networks are among the most notable examples.

3.2 Protein Residue Network

Protein residue network (also referred to as protein interaction network) for any protein can be constructed by considering each C_α atom of each amino acid as nodes. Two nodes then connect, and one builds a link between them provided that the corresponding C_α atoms are separated by at most a certain specified distance. Hence, a cut-off is defined for the system under which an edge exists between two nodes. This model maps the protein's all-atom conformation to a graph with amino acid residues as vertices, and all (covalent and non-covalent) interactions between them as edges [37, 29]. Fig 3.1 represents three dimensional conformation of the protein in all-atom representation on the left and the network representation of CalB on the right, in which the set of 3-dimensional coordinates of the location of the residues provides

[1] See Definition A.1 in Chapter 4

a point cloud data (PCD), each amino acid residue represented by its C_α atom is a node, and each edge between two nodes is constructed if the distance between a pair of C-alpha atoms is within the cut-off value defined according to bond lengths. This network model of protein networks usually has the cut-off of about 7Å.

As mentioned earlier, a widespread statistical method to analyze a protein is through protein interaction network and analyzing the network obtained by thresholding the distances (sparse network). However, in this work, instead of the binary adjacency matrix, we define and use the weighted adjacency matrix to examine the CalB system as a fully connected network to prevent any information loss for our further analysis. The **adjacency matrix** A of a weighted network is the matrix with elements A_{ij} such that: $A_{ij} = 0$, when there's no edge between vertices i and j, and the A_{ij} value is equal to the weights of the corresponding connections $A_{ij} = \omega(e_{ij}) = \omega_{ij}$ if the edge between them exist. One method to define the adjacency matrix for a point cloud is from the corresponding distances between the nodes by having the weights defined as: $\omega = d_{ij}$ where d is a metric defined on the space, e.g., the Euclidean distance on the Euclidean space. One can use other definitions for adjacency matrices based on various correlations between the nodes of the system. For example, another widespread way to define weights for point cloud is $\omega = 1/d_{ij}$.

3.2.1 Protein Residue Network of CalB

To create the protein residue network of CalB we first take the protein structure's dynamic model (since the system changes over time) and define dynamical distances between the 317 points. Thus a protein now will be represented by a 317×317 distance matrix, representing our adjacency matrix for each snapshot, which will provide the input of our further analysis. In Fig. 3.2a we first plotted the distance matrix for the crystal structure of CalB. As it can be noted from the figure, the value of the distances between the nodes in the crystal structure varies from 0 to about 50Å. The resulting tabulation of distances can be presented graphically in the form of a frequency histogram. In Fig. 3.2b we have plotted the frequency histogram of the distances between the nodes. In this representation rectangles are constructed over each interval with their height being proportional to the number of class frequencies of weights in the crystal structure which better represents how the links of the network are distributed among 0 to 50Å. The total number of nodes (the size) of the protein network is ($N = 317$), giving us the corresponding frequency histogram for N^2 edges. There are 317 links at class zero which correspond to the diagonal of the distance matrix ($\omega_{ii} = \omega_{jj} = 0$). We can see from the figure that in the crystal structure, most of the nodes are within 20 to 30Å from one another, where the peak of the graph is located.

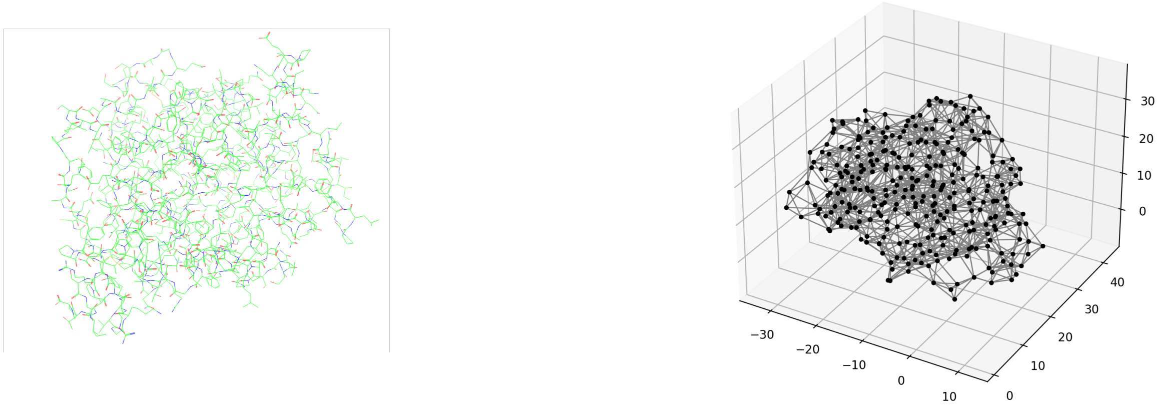
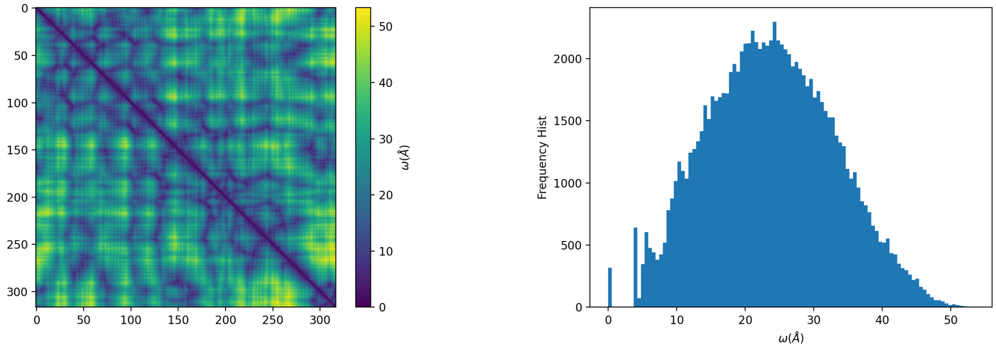


Figure 3.1: **All atom representation of CalB** downloaded from Protein Data Bank (1TCA) on the left, **and Protein Interaction Network** of CalB on the right using cut-off of 0.7nm, where the black dots represent alpha carbons and provide the nodes of the graph. We set the cutoff distance to 7Å, and edges connect the nodes with black solid lines if the nodes are within this distance.



(a) (b)

Figure 3.2: **(a) The adjacency matrix** (distance matrix) for the crystal structure of CalB (1TCA), and **(b) the histogram of the distance matrix values**. The adjacency matrix has 317 × 317 elements and the distances between these elements can be compared using the colorbar. Since $\omega_{ii} = \omega_{ij} = 0$ all the distances on diagonal are set to zeros, and accordingly there exist 317 links in class zero.

3.2.2 Statistical Analysis of CalB Interaction Network in Different States

To analyze the complex network of protein statistically, we choose three different states of CalB according to our previous RMSD and R_g and active site analysis:

- Glycerol 423K: Folded and active state of CalB
- Water 300K: Folded and inactive state of CalB
- Water 423K: Unfolded and the inactive state of CalB

In the following sections we compare some statistical features for protein network in these states. In Fig .3.3, we first plotted the graph representation of the final snapshot(after 50 ns) of protein interaction networks. Black dots represent the nodes of the graph (α-carbons of each amino acids) and red dots highlight the active site residues. From this figure we can see the position of active site residues embedded in the framework of the overall network for each case. These figures also represent how the active site residues seem to be held closer in glycerol. In 3.3c we noticed that a segment of the protein has drifted away from the rest of the structure during the unfolding process.

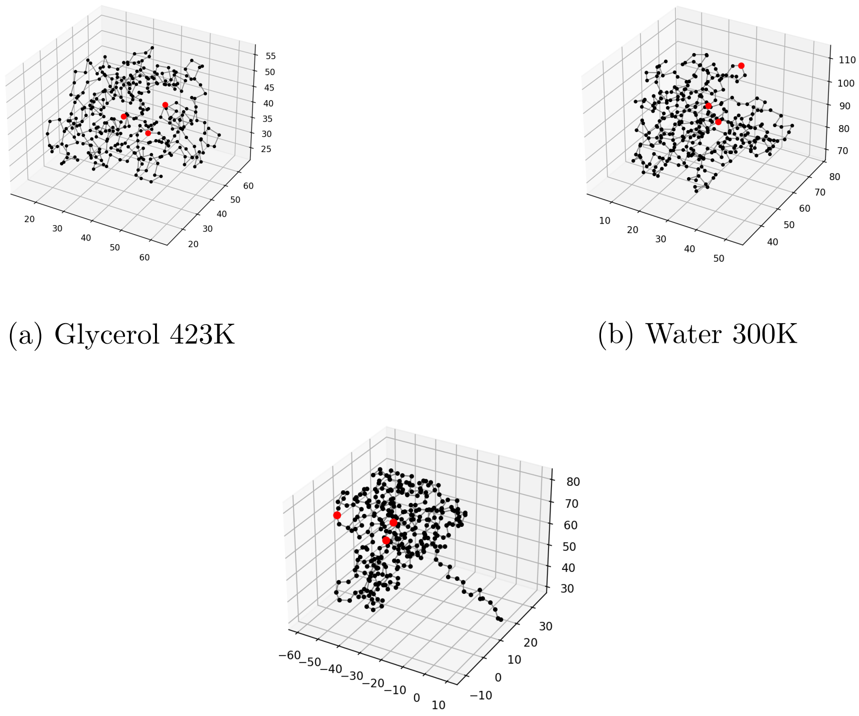
Since the adjacency matrix (distance matrix), is time dependent with elements for each protein interaction network changing in time ($A(t)$), we need to calculate and compare the time averaged adjacency matrix in different states. Therefore, for each time dependent element of the matrix, $A_{ij} = \omega_{ij}$, we need to consider the 50 ns time-averaged value and build the time-averaged adjacency matrix ($\bar{A}$) accordingly. If we denote the size of the network (number of nodes) by N, for $\mathbb{Z}_N = \{1, 2, ..., N\}$, we have:

$$\forall (i,j) \in \mathbb{Z}_N^2 : \quad \omega_{ij}(t) = d(\vec{x_i}(t), \vec{x_j}(t)) \equiv \left[\sum_{\delta=1}^{3} (x_i^\delta(t) - x_j^\delta(t))^2 \right]^{1/2} , \tag{3.3}$$

where d is the Euclidean distance between the nodes. Hence, for each trajectory of protein from the MD simulation result we have obtained time series:

$$\{\omega_{ij}(t)\}_{t=1}^{T} , \tag{3.4}$$

where T is the number of time frames. Therefore, our system of protein network evolving in time can be described by a collection of time series of the above form (${}^N P_2$ number of time series). Now taking average in time yields:



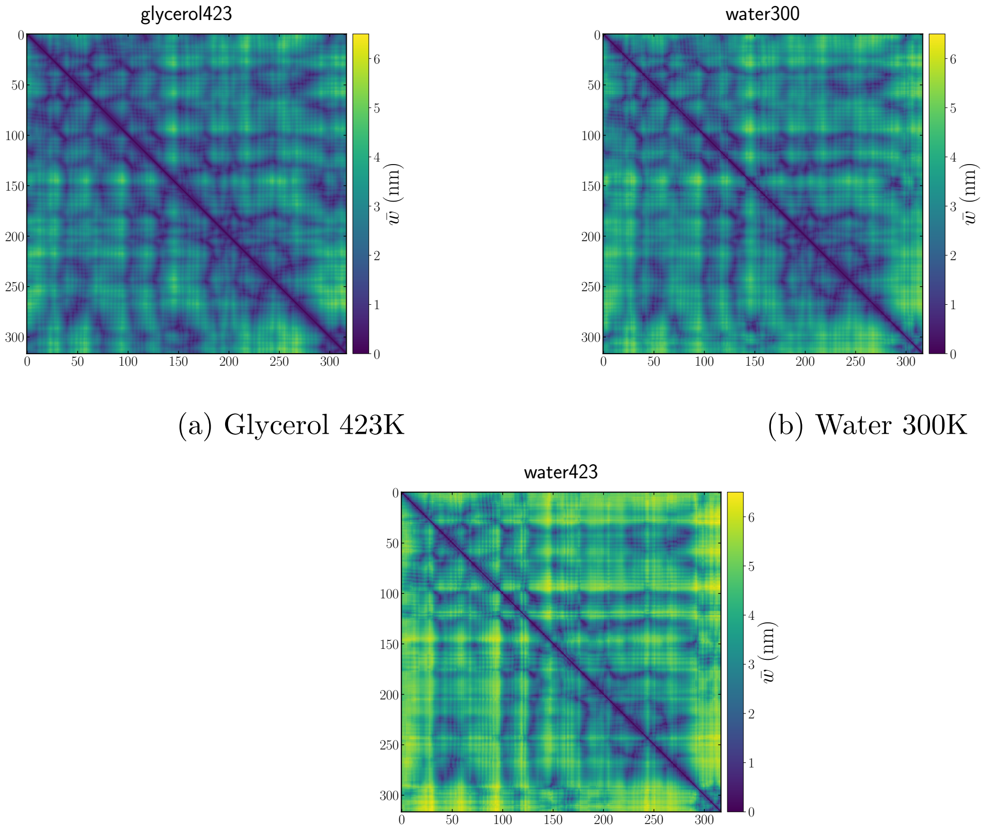
(a) Glycerol 423K

(b) Water 300K

(c) Water 423K

Figure 3.3: **Schematic Representation of Protein Interaction Network** of CalB in three different states. a. In glycerol at 423K, b. In water at 300K and c. In water at 423K. The black dots represent the nodes of the graph which are the C_α of each amino acid and the red dots represent active site nodes.

$$\bar{\omega}_{ij} = \langle \omega_{ij}(t) \rangle_t = \frac{1}{T} \sum_{t=1}^{T} \omega_{ij}(t), \tag{3.5}$$



(a) Glycerol 423K

(b) Water 300K

(c) Water 423K

Figure 3.4: **Time averaged adjacency matrices** for protein in glycerol at 423K, Water at 300K, and Water at 423K. The color bar shows the ranges of distances(weights) in *nm* for each matrix. The elements of the matrices present the distances between each pair of residues in different states.

which can be used to plot the weighted adjacency matrix($\bar{A}$) for each protein network with having $\bar{\omega}_{ij}$ as the elements. In Fig 3.4 we represented time averaged adjacency matrices for protein residue network of amino acids in these three states. As we can note from the heat maps in different states, the protein network includes the most extensive ranges for matrix elements for Water at 423K, where the structure unfolds, and amino acid residues take considerable distances from one another. The large amount of deviation from the rest of the network in water 423K is highlighted in bright yellow for some residues near 99 and 25 for instance. Which emphasize that these residues are far from all the others and remind us about how protein unfolds in water at 423K. As expected for protein in glycerol at 423K and water at 300K where the 3D structure is folded (according to RMSD and R_g) all the distances between the nodes of the network are within the same range compared to the crystal structure and are smaller compared to the unfolded case (water423K). Darker spots on the adjacency matrices are those residues that are close to each other in space and are assumed to interact with one another (diagonal represents the distances from themselves). Hence, the adjacency matrices of the protein interaction network of CalB in different states can provide a good starting point to study the pairwise interaction between the nodes (amino acids) and will be discussed in the next section. Moreover, we are very interested in exploring the observed patterns of these darker dots on the plots. Therefore, in the later chapter, we utilize the method of Topological Data Analysis to detect these interaction patterns and analyze them.

3.2.3 Probability Distribution Functions

The probability distribution function of the time-averaged distance matrix provides a mathematical model for the population histogram and is obtained by normalizing the frequency curve[2]. Fig 3.5 shows the 50 ns time averaged probability distribution function (PDF) for protein in different states. As we can see, the density starts to increase, has a peak and decreases at higher thresholds. It goes to zero as the values of weights go beyond around 5nm in glycerol, confirming that our network doesn't have any distance between pair of nodes at these distances, and the amino acid residues are always within 5 nm. This number goes up slightly for the protein network in water at $300K$ and, as expected, takes its maximum value of above 6 nm for protein in water at $423K$. Protein Network has different distribution function in different states in terms of the order of mean, the spread of distances and the measure of the symmetry. As we can see from this plot, for protein in glycerol the probability of finding distances between amino acid residues is maximum at around 2 nm. The spread peak has dropped in value and also has taken a notable shift to higher weights in water, representing that, in water, we detect higher distances between the nodes, which on average caused the peak to shift to higher weights, which in water at $423K$ it increases to about 5 nm.

[2] The statistical analysis of each state is also plotted separately in Appendix B

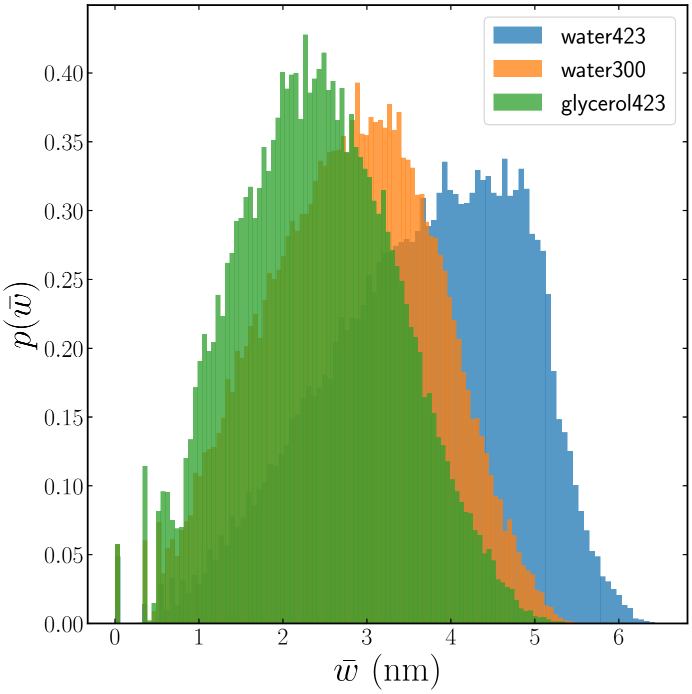


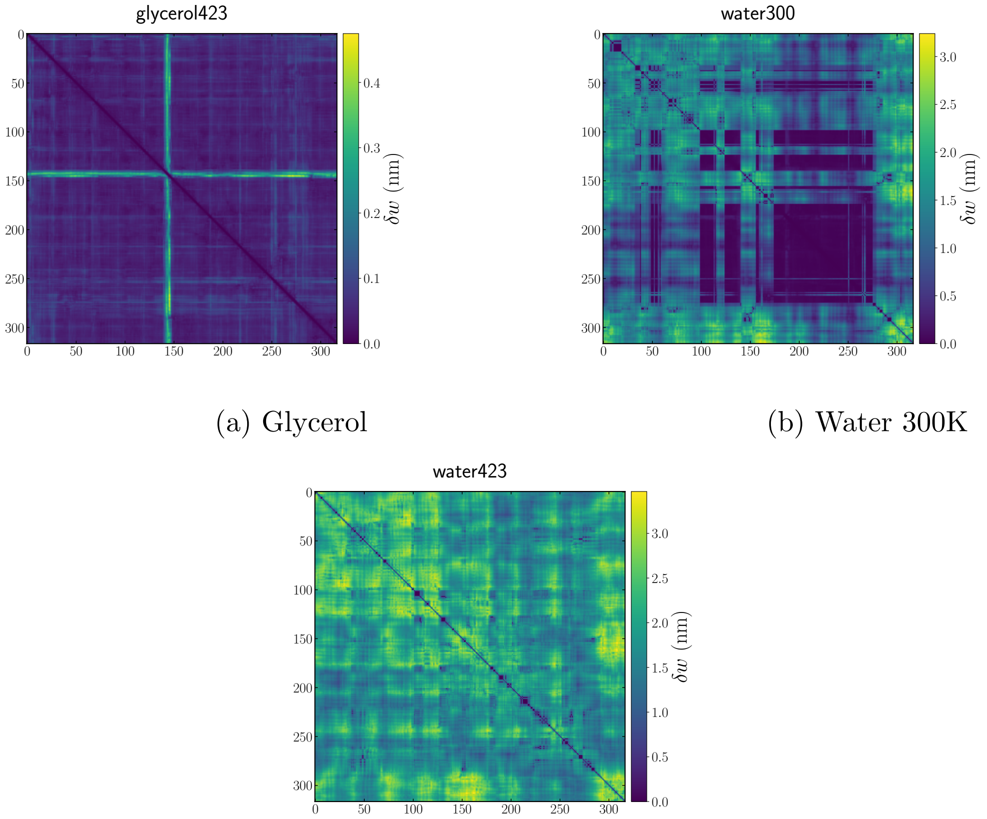
Figure 3.5: Time Averaged Probability Distribution of links for protein residue network in glycerol(green), water 300K(orange) and water 423K(blue) calculated from eq 3.5. Horizontal axis shows the time averaged weights (distances) and the vertical axis shows the normalizes frequency histogram for each class representing the probability of corresponding ω between the nodes.

3.2.4 Standard Deviation

The variance of a set of measurements $y_1, y_2, y_3, ..., y_n$ is defined as the average of the square of the deviations of the measurements about their mean [36]. The standard deviation is equal to the positive square root of the variance. Generally, the standard deviation is a measure of the spread of a distribution. However, since the value of weights(ω_{ij}) is changing in time, one should note that the standard deviation for our time-dependent distribution is not a measure of the spread of the previous distribution function, but the deviation that the distribution of links has in time (range of motion). Hence, to perform the calculation for standard deviation and mean we need to refer to the time series, introduced in 3.4. This deviation for each matrix element (weight) can be calculated using $\delta\omega_{ij}$ where:

$$(\delta\omega_{ij})^2 = \left\langle (\omega_{ij}(t) - \bar{\omega}_{ij})^2 \right\rangle_t = \frac{1}{T}\sum_{i=1}^{T}(\omega_{ij}(t) - \bar{\omega}_{ij})^2. \tag{3.6}$$

In Fig 3.6 we showed the Standard Deviation of the distribution for the network in three different states. The value of standard deviation reveals that for protein in glycerol, almost all of the distances between the pairs fluctuate within $0.5nm$ interval. For protein in water at $300K$, however, we note the number of bright dots has changed dramatically as can be seen from Fig 3.6b and this number increases to $3nm$. This confirms the existence of a broad spatial distribution for the protein network in water $300K$ and, accordingly, more scattered distance fluctuations between the nodes. In that state the lower right area of the matrix (where the active site is located) has the least fluctuation and seems frozen compared to the rest of the molecule. Hence, we concluded that although protein does not unfold in water at $300K$, according to RMSD and R_g, the distances from amino acid residues go through high deviation and despite the folded general configuration of the protein, links between the nodes on average fluctuate highly compared to the active state in glycerol. For water at $423K$, the number of pairs with high standard deviation increases, and we notice higher fluctuations throughout the whole system. In this state we also see many bright spots close to the diagonal (about80 to 90, 110 to 120 for instance), which shows that the alpha carbons despite being close to their neighbours on the chain, go through pairwise fluctuations. For protein in glycerol, we observed that even though the whole structure doesn't fluctuate in time, some residues around residue 150 seem to have a very high fluctuation compared to the whole system, causing two symmetric bright yellow lines to stand out in Fig 3.6a. We hypothesized that this can be due to the movement and unfolding of α_5 where residues 142 to 146 are moving a lot with respect to all other amino acids, contributing to the activity of the protein. In Fig 3.7, we plotted the distribution function for the standard deviation of the weighted network. The horizontal axis shows the value of standard deviation,



(a) Glycerol

(b) Water 300K

(c) Water 423K

Figure 3.6: Standard Deviation of probability density function of distances for protein network in different states

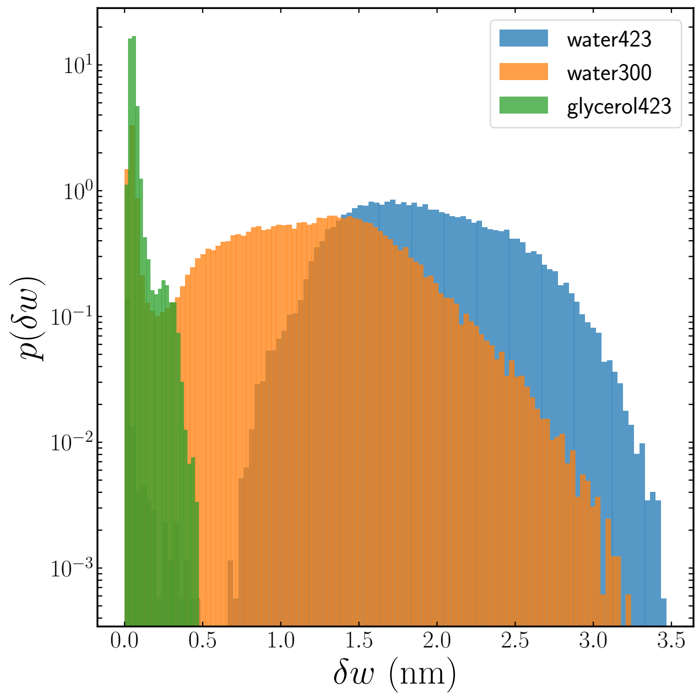


Figure 3.7: Figure shows the histogram of standard deviation, calculated from eq 3.6 for network in different states. Standard deviation highlight the changes that adjacency matrix go through during the simulation.For glycerol other than some residues most of the others go through minimal fluctuation. This number has increased dramatically in water.

and the vertical axis stands for the frequency of the number of pairwise distances. As we can deduce from the figure, the peak of the distribution locates in a smaller standard deviation for glycerol. Since the standard deviation is averaged in time, we can conclude that, on average, the number of pairs with a very small (close to zero) standard deviation is maximum in glycerol. Having a minimum frequency for the tail of the distribution implies that a tiny number of residues tend to fluctuate higher from others in time. Thus in glycerol, on average, we can claim that the oscillations of the pairwise distances in time are almost constant; nevertheless, the distributions of standard deviation in water at both temperatures look flatter.

3.2.5 Mean Probability Distribution

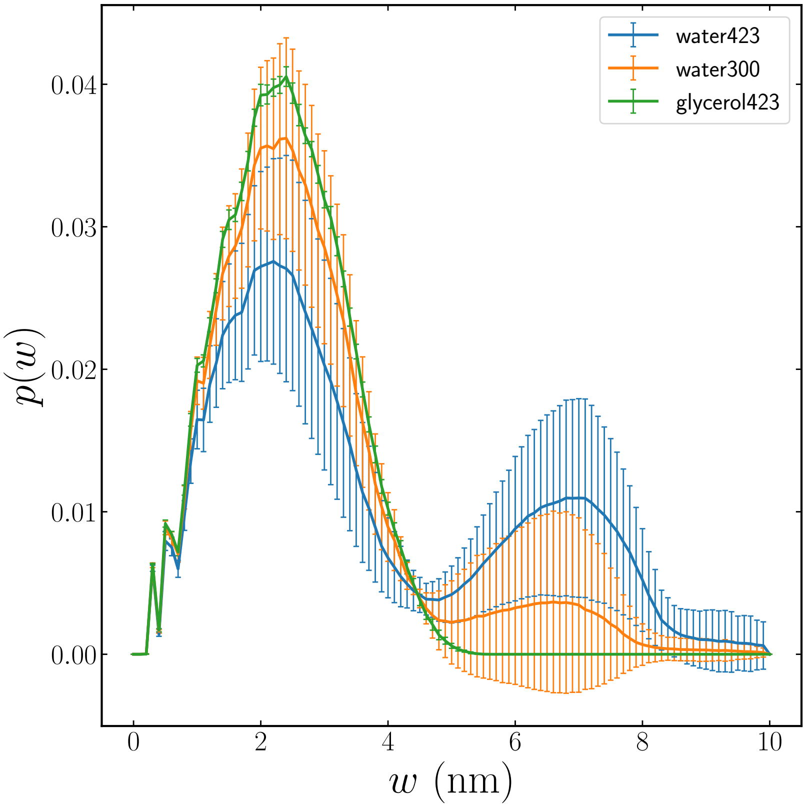


Figure 3.8: Figure shows the distribution of mean for network in different states. The horizontal axis represents weights and the vertical axis is the probability distribution. The larger error bars implies the greater fluctuations of distances in time.

To emphasize the dynamics of the protein in time, we also plotted the probability distribution of the mean of the time series with error bars representing each time frame in Fig 3.8. The vertical axis is the probability distribution of the weights, and the horizontal axis represents the value of weights ω in nm. For protein in glycerol, we marked only one peak at smaller weights around 2 nm, indicating that most amino acids are located at that distance. The tiny error bar also emphasizes that these minimum distances are kept throughout the whole simulation in time. For water at $300K$ we detect a small number of residues in a higher weights with a relatively larger error bar indicating the fluctuations in time. The error bar has its most significant value for protein in water at 423K, and the value of the second peak at around $\omega = 7nm$ has increased as the protein unfolds. Our analysis confirms that even though the protein in water at 300K maintains its folded state, the distances between amino acid pairs show another tiny peak at a higher threshold. For water at 423K as the number of distances in higher threshold increases, the value of this second peak increases, giving the previous shift in the average distribution function of Fig 3.5.

3.3 Chapter Conclusion and Remarks

To summarize this chapter, we want to emphasize how complex system theory helped us analyze the protein's dynamics in different temperatures using pairwise interaction between amino acids and how it will provide us with the input to look into the general topology of the protein. We used the theory of complex networks to map the three-dimensional protein structure composed of more than 2000 atoms (2325) to a network of 317 nodes in position space (R), with each node representing amino acid residues and weighted edges illustrating the distances between them. We used statistical features of the network, such as probability distribution of lengths, standard deviation, and the mean of distance distribution in time to highlight the differences between the folded-active and the folded-inactive state of CalB. We showed that CalB dynamics in high temperature (423K) in glycerol solvent have a small links' distribution variance in time. Hence, we concluded that the protein network in its active state at high temperatures holds structures that are persistent in time. In order to examine these persistent structures throughout the whole network, one can go beyond pairwise interactions between the nodes and employ methods based on patterns of interaction and accordingly obtain a global measure for the network. In the next chapter, we present how we utilized a new computational method based on analyzing the network's topology to characterize these persistent structures. Intuitively, looking at a network from a topological point of view is like describing a building through its floors, bedrooms, and hallways instead of its building blocks. In the most straightforward wording, using the mathematical tool of topology, we explore "empty spaces" in the topological spaces built for the data sets to characterize them.

Chapter 4

Topological Data Analysis of Protein Interaction Network

4.1 Topological Background

This subject mainly concerns a brief introduction to algebraic topology concepts that we will use to analyze the networks from a new perspective. Provided that a thorough learning of the subject can be found in references like [41] and [25]. Topology is a branch of mathematics that is mainly involved in studying spaces (topological spaces) to compare them and categorize them. Algebraic Topology, employs algebra to examine "shapes" and their properties in topological space. These properties are mostly independent of continuous transformation and remain invariant under stretching and shrinking.

To a non-mathematician, Topology is the "rubber-sheet" or "squishy" version of Geometry, as it focuses on features of space that stay intact under continuous deformation, and squares and circles will become equivalent. In this context, homology helps us understand whether two things are the same topologically, and to examine their topological differences by counting how many fundamental holes they have. In Fig 4.1, we show different dimensional topological spaces together with their equivalent geometrical spaces and network representation. To analyze a shape topologically, we explore 0-2-dimensional fundamental holes of the topological spaces created for different data sets. For example, in one dimension, we refer to 1-dimensional holes as rings, and in the 2-dimension, as voids (or cavities).

As we will see in the later sections, we use so-called Betti numbers to count the number of fundamental holes of different dimensions. Bear in mind that not all the features of topological spaces can be detected by eyes, and we will further see how the concept of simplices and simplicial complexes has become very helpful when it comes to more complicated objects. The algebra that is utilized in the theory of Algebraic Topology is mostly from the language of group theory. Here, we review some important concepts and definitions to build a necessary foundation for our research and analysis.

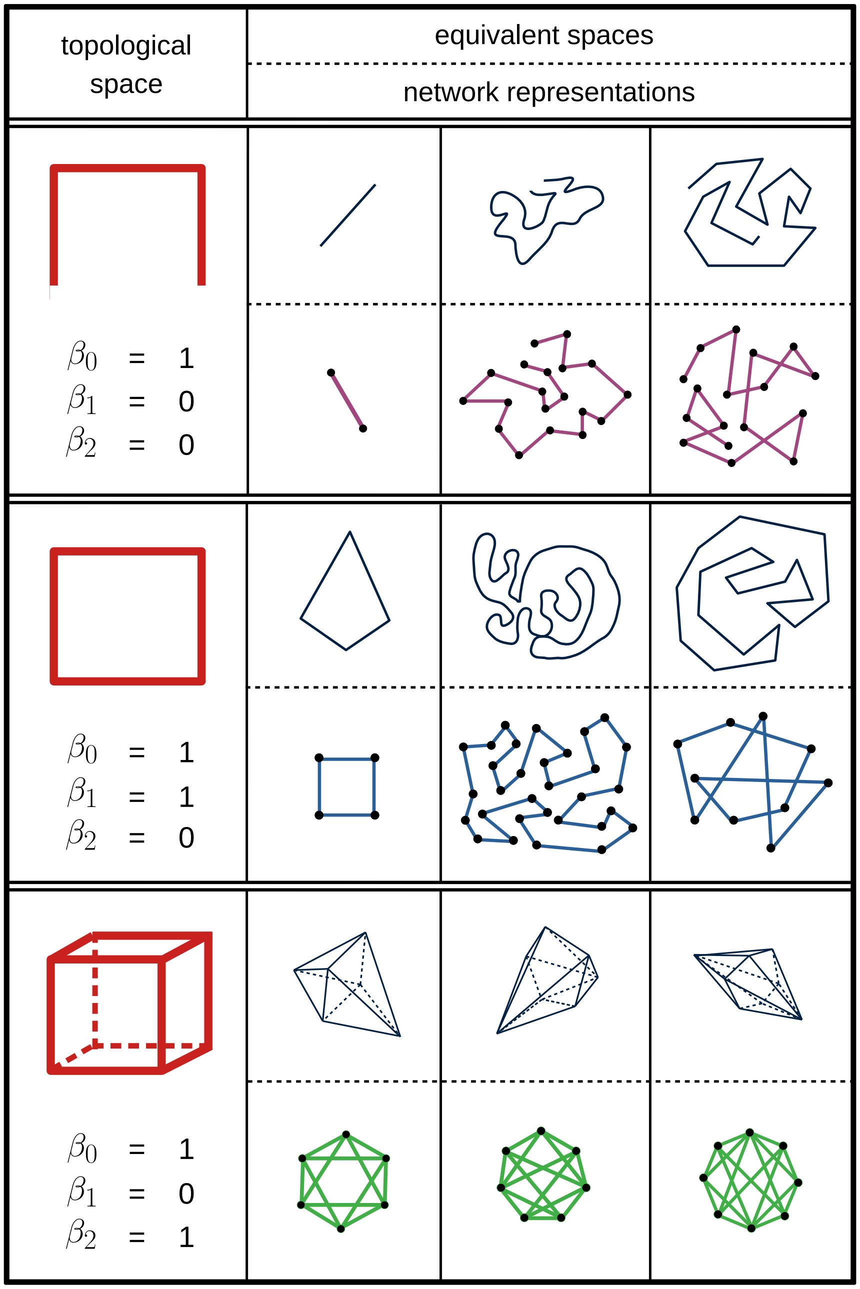


Figure 4.1: Figure shows a simple example for topological spaces in terms of simple geometrical spaces and network representation. We used β_0, β_1 and β_2 to count number of fundamental holes of each representation.

Topological spaces and Topological Invariant

Even though physicists usually assume all the spaces they deal with to be equipped with some metrics, this is not always true, since in fact, metric spaces form a subset of manifolds, where manifolds themselves form a subset of topological spaces [41]. The concept of topological spaces mostly originated from the generalization of the study of the real line and Euclidian space[40]. Touching base with the mathematical foundation required for understanding the method of topological amalysis, we define a few more basic concepts such as topology, topological spaces, simplicial complexes and topological invariants which will become important in the following sections. We begin by defining a topology.

Definition 4.1.1 (Topology)**.** For any subset X of D-dimensional Euclidean space: $\mathbb{R}^D$, a **topology** on X is a collection $\mathcal{T}$ of subsets of X with following properties:

- $\emptyset$ and $X \in \mathcal{T}$.
- The union of elements of any subcollection of $\mathcal{T}$ is in $\mathcal{T}$.
- The intersection of elements of any subcollection of $\mathcal{T}$ is also in $\mathcal{T}$.

Definition 4.1.2 (Topological Spaces)**.** A **topological space** is an ordered pair (X, $\mathcal{T}$) that consists of a set X and a topology $\mathcal{T}$ on X.

Remark. We call sets that belong to collection $\mathcal{T}$ open sets of X.

Remark. A subset A of X is closed if its complement in X is an open set.

Definition 4.1.3 (Continuous map)**.** For topological spaces (X, $\mathcal{T}_X$) and (Y, $\mathcal{T}_{\mathcal{Y}}$) a map $f : X \to Y$ is a **continuous map** if the inverse image (see def.A.1.3) of any element in $\mathcal{T}_Y$ is an element in $\mathcal{T}_X$.

Definition 4.1.4 (Homeomorphism)**.** For two topological spaces (X, $\mathcal{T}_X$) and (Y, $\mathcal{T}_{\mathcal{Y}}$) (we will use short notation X and Y for topological spaces moving forward) a map $f : X \to Y$ is a **homeomorphism** if it is continuous and bijective with its inverse $f^{-1} : Y \to X$ also being continuous.

Remark. If there exist a homeomorphism between X and Y, X is said to be **homeomorphic** to Y and vise versa.

Intuitively speaking, from the perspective of topology, two topological spaces are said to be homeomorphic to one another, if using continuous deformation we can transform one into the other; that is without tearing them or pasting [41]. Consider

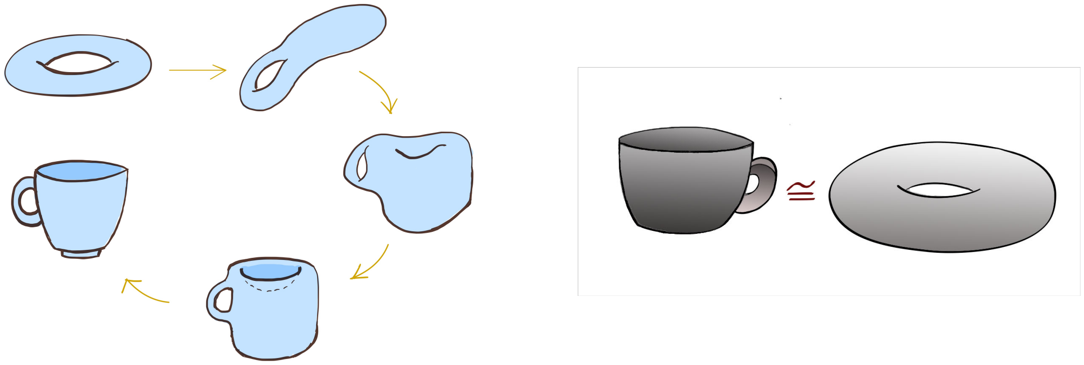


Figure 4.2: A coffee cup is homeomorphic to a doughnut. Reproduced from the concepts explained in [41]

a coffee cup that is said to be homeomorphic to a donut(see Fig 4.2), in this context, topology allows an enormous group of homeomorphism, deforming one object by stretching and shrinking, and it is only through cutting the curve that we convert its topology from a closed loop to a path. Fig 4.3 shows some examples of homeomorphisms. Topology with the main object of its study "topological spaces" as the most general form of space, categorizes a shape based on its connectivity, which can be thought of as its number of pieces, loops, or the presence of a boundary. Topology then examines these topological spaces based on whether they still retain a notion of connectivity[60]. Hence, homeomorphism is an equivalence relation that divides all topological spaces into equivalence classes.[1]

Definition 4.1.5 (Topological Invariant)**. Topological invariants** are those quantities which are conserved under homeomorphism.

Remark. If two spaces have different topological invariants they are not homeomorphic to each other.

Examples of topological invariants could be a number such as the number of connected components of the space, an algebraic structure such as a group or ring, constructed out of the space. Topological invariants of spaces could also be a property like connectedness and compactness, Euler characteristics, or homology groups (Betti numbers).

[1] See Definition A.1.6 and Definition A.1.7 in the AppendixA.

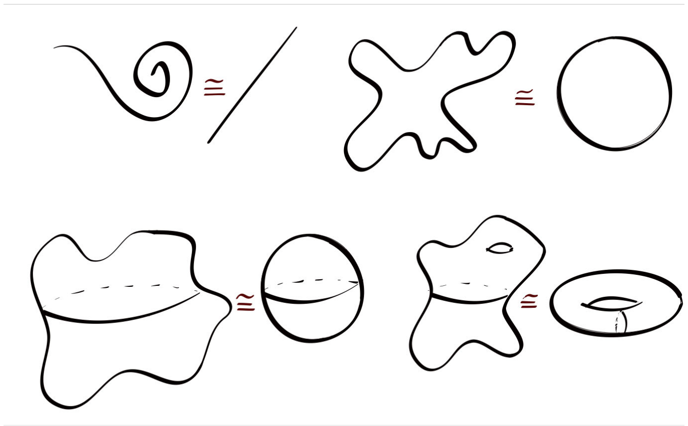


Figure 4.3: Examples of homeomorphic spaces. Reproduced from concepts explained in [41].

4.2 Homology Groups of a Simplicial Complex

We begin our discussion of the topological analysis of data by introducing the homology theory and homology groups. One should bear in mind that one of the most helpful guides in categorizing spaces, and topologically analyzing their characteristics, mathematically elaborated into the theory of homology groups, is to "find an area without boundaries within the space, that is not a boundary of any area itself"[41]. In other words, to analyze our data using Algebraic Topology, we search for the existence of a loop of any dimension that is not a boundary of some area itself. In this concept, we look for the existence of a "hole of some dimension" within the loop to classify different spaces. Analyzing a complex topological space and trying to understand it, mathematicians have developed a technique of imagining it built up from smaller pieces called simplices. For example, for a surface like a torus (donut), one can triangulate that surface and track how they are connected. Furthermore, using the idea of simplices and simplicial complexes, one can detect the loops and holes of topological spaces. This section will see how the concept of simplices has helped us understand the topological structures of complicated spaces.

4.2.1 Simplices and Simplicial Complex

Before describing (simplicial) homology groups, we introduce the class of spaces that defines them: class of polyhedra. A polyhedron is a space that can be built from "building blocks" such as line segments, triangles, tetrahedra, and their higher-dimensional analogs by “gluing them together” along with their faces [40]. In this section, we first define simplexes as the building blocks of polyhedra to build the basics of homology groups.

Definition 4.2.1 (Simplexes (Simplices))**.** A k simplex σ_k spanned by $c_0, c_1, ..., c_k$ is a set of all point x in R^D $(D \geq k)$ such that:

$$\sigma^k = \left\{ x \in R^D \mid x = \sum_{i=0}^{k} c_i x_i, c_i \geq 0, \sum_{i=0}^{k} c_i = 1 \right\} \equiv < x_0, x_1, ..., x_k > \subseteq R^D \tag{4.1}$$

Remark. The set of parameters $(c_0, ..., c_r)$ is called barycentric coordinate of x. Note that for any k-simplex representing a k-dimensional object the vertices x_i must be geometrically independent.[2]

Illustrations of simplexes, building blocks of a polyhedron, are in 0-dimension (0-simplex $\sigma_0 =< x_0 >$) a point or a vertex, a 1-simplex $\sigma_1 =< x_0, x_1 >$ being a line segment or an edge, a 2-simplex $\sigma_2 =< x_0, x_1, x_2 >$ defined to be a triangle with its interior included and a 3-simplex $\sigma_3 =< x_0, x_1, x_2 >$ is a solid tetrahedron. Fig 4.4 demonstrate 0-, 1-, 2- and 3-simplexes.

Definition 4.2.2 (Simplicial Complex)**.** A collection ψ of simplexes “nicely” fitted together is a simplicial complex. Such that:

- Every face of a simplex of ψ is in ψ
- The intersection of any two simplexes of ψ is a face of each of them.

Remark. The dimension of a simplicial complex is defined to be the largest dimension of simplexes in it (All graphs are simplicial complexes of dimension 1).

We will soon notice the basics of simplicial homology theory concerns assigning to each simplicial complex a chain complex followed by its homology group. This approach will be used to characterize topological spaces, which will look like polyhedra, to cover a manifold by a process called triangulation.

4.2.2 Homology Groups, and Betti numbers

Homology with the help of so-called Homology groups and Betti numbers quantitatively detects loops and holes in various dimensions of the simplicial complex to give insights into the way a topological space is connected. In this section, we are going

[2] See Definition A.1.8 in the Appendix

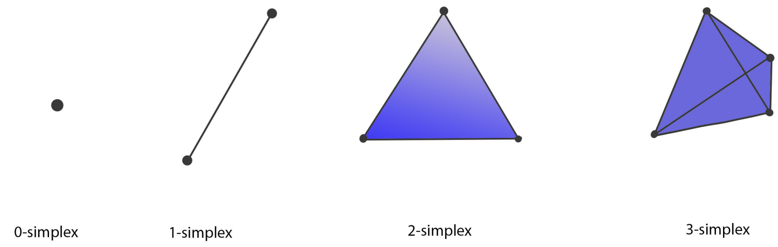


Figure 4.4: Low dimensional simplices: A vertex, An edge, Triangle, and a a solid tetrahedron

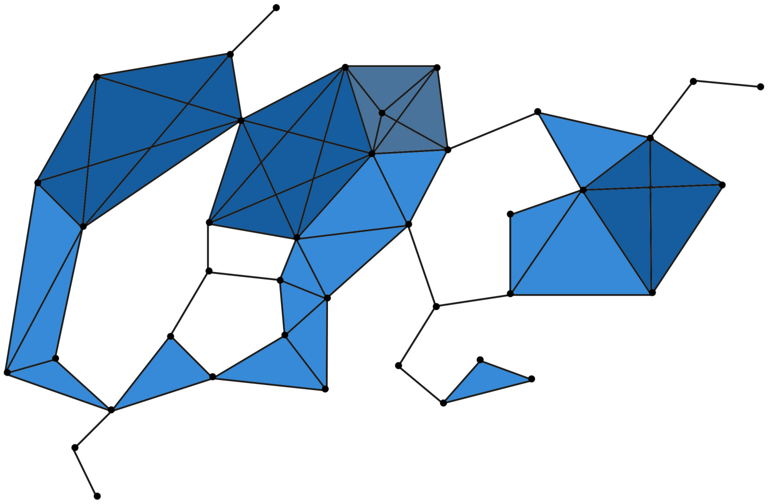


Figure 4.5: An example of a simplicial complex

to define Homology groups, detecting holes and loops indirectly by looking at the space surrounding them, and Betti numbers as a way to count them. We begin our discussion by describing chains, chain groups, and cycles of any simplicial complex to establish the necessary background.

Definition 4.2.3 (k-chain)**.** For a simplicial complex ψ, a **k-dimensional chain (or k-chain)** is a formal sum of k-simplices in ψ such that:

$$c_k = \sum_{i \in I} a_i \sigma_k^i, \tag{4.2}$$

where a_i are coefficient and σ_k^i are k-simplices and I is an index set.

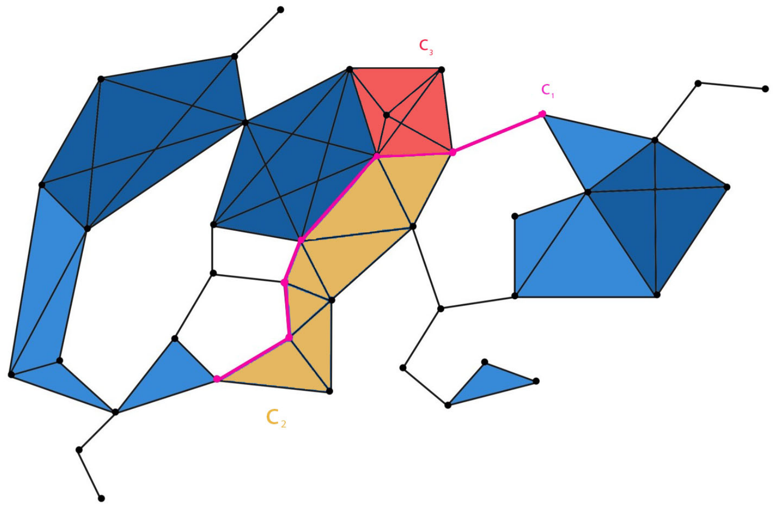


Figure 4.6: 1-chain c_1 (colored in pink), 2-chain c_2 (colored in yellow), 3-chain c_3 (colored in red).

Fig 4.6 shows some of the k-chains of simplicial complex example in Fig 4.5. The commutative group generated by all the k-chains of ψ is called a k-dimensional chain group, denoted by $C_k(\psi)$ For example, regarding a simple graph of vertices and edges 0-dimensional chains C_0 elements are integral linear combination of vertices and C_1 elements are linear combinations of edges. Here, in order to see what cycles of a simplicial complex are and how they can be detected using homology, we will consider the relationship between chains of different dimensions and define the boundary of chains in the following section. We will notice how the concept of the boundary has become significant in homology theory and how it has enabled us to relate different k-chain groups to detect the cycles of a simplicial complex.

Definition 4.2.4 (Boundary operator of a simplicial complex)**.** The k-dimensional **boundary operator** denoted by ∂_k is a topological operator that upon acting on any simplex σ_k gives a $k-1$ chain of σ_k. We write:

$$\partial_k(\sigma_k) \equiv \sum_{i=0}^{k} (-1)^i [x_0, x_1, ..., x_{i-1}, x_{i+1}, ..., x_k]. \tag{4.3}$$

Boundary operator is a homomorphism that maps k-simplices to their boundaries and consequently k-dimensional chain group C_k to $(k-1)$-dimensional chain group C_{k-1}:

$$... \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} ... \rightarrow C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \emptyset, \tag{4.4}$$

where we call the sequence 4.4 of chains and homomorphism a chain complex. Fig 4.7 represents boundary operator acting on different dimensional simplices. Accordingly, one can define a k-dimensional cycle (k-cycle) Z_k as a k-chain c_k that is mapped to empty set by boundary operator, $\partial_k(c_k) = \emptyset$. This leads to create a subspace Z_k, so-called k-dimensional cycle group (k-cycle group), of vector space C_k.

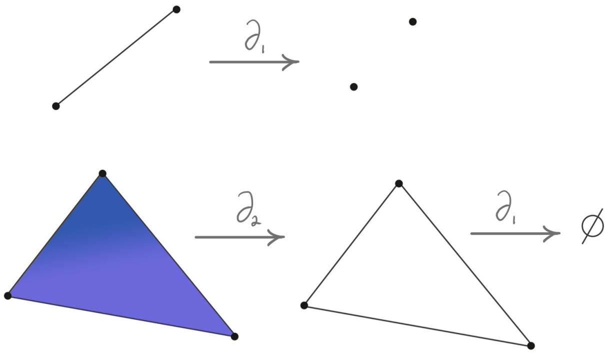


Figure 4.7: Results of boundary operation on simplices of different dimensions.

Definition 4.2.5 (k-cycle)**.** A k-chain c_k that has no boundary is a k**-cycle** Z_k which satisfies the following relation:

$$\partial_k(c_k) = \emptyset. \tag{4.5}$$

The set of all k-cycles is a subgroup of $C_k(\psi)$ and is called the k-cycle group. Note that $Z_k(\psi) = Ker\partial_k$, and for $k = 0$, $\partial_0 c_0$ vanishes.

Definition 4.2.6 (k-boundary)**.** k-dimensional chain c_k is called k**-dimensional boundary or k-boundary** b_k if it represents the boundary of $(k+1)$ chain.

The set of k-boundaries $B_k(\psi)$ is a subgroup of $C_k(\psi)$ and is called k**-dimensional boundary group (or k-boundary group)**. Note that $B_k(\psi) = Im\ \partial_{k+1}$. Since "boundaries have no boundary", we can write:

$$\partial_k(b_k) = \partial_k(\partial_{k+1}(c_{k+1})) = \emptyset \tag{4.6}$$

Definition 4.2.7 (k-th homology group)**.** k**-th homology group** H_k of a simplicial complex psi is the quotient group of k-cycles Z_k modulo the group of boundaries B_k. We write:

$$H_k = Z_k/B_k. \tag{4.7}$$

The set defined by 4.2.7 basically represents the set of all k-chains of a simplicial complex that have no boundaries and are not themselves a boundary of any spaces. Consider expression 4.7 in 1 dimension, we have $H_1 = Z_1/B_1$, taking the quotient intuitively means looking for cosets of B_1 in Z_1, which in one dimension, implies finding the 1-cycles (loops) and disregarding the boundaries (cycles that are boundaries of some 2d cells), and those are the H_1 of the simplicial complex. Note that homology groups are topological invariants.

Definition 4.2.8 (k-th Betti number)**.** The k**-th Betti number** of a simplicial complex β_k is defined by:

$$\beta_k(\psi) = \mathrm{dim}(\mathrm{H_k}(\psi)) \tag{4.8}$$

Betti number as a topological invariant of the complex is the dimension of the k-homology group of the complex and represents the number of k-dimensional holes in any ψ. Hence, β_0 counts the number of connected components, β_1 counts the number of loops, and β_2 counts the number of voids of any simplicial complex and so on.

4.3 Method of Topological Data Analysis

Topological Data Analysis (TDA) is a subarea of computational topology, utilizing the language of algebraic topology, to develop novel topological techniques for robust analysis of various categories of scientific data [60]. To do so, TDA, employing a collection of powerful tools, represents characteristics and properties of data structures in topological fingerprints to quantify the "shapes" of these data sets [39]. In the previous section, we introduced homology as a tool from algebraic topology that identifies different features of a topological space such as annulus, sphere, torus, or more complicated manifolds and defined homology groups and Betti numbers as a way for homology theory to characterize these spaces from one another by quantifying their homological features.

When can TDA become useful?

In switching from network to topological analysis, the first question one should ask is if the system of interest is the right fit for the method and how topology can help with analyzing this system. To address this question, one should first consider if higher-order interactions (more than pairwise) can become important and if the global patterns of interactions between elements can affect any functionality in the scale of the whole system. For such systems, one can apply TDA methods to acquire a vision of the global topology of the system to disclose the system's inherent structure and function. Moreover, one should consider if the topological loops or cavities of the structure can have any significance, and how these features can be interpreted.

In network science, we translate data into a graph of nodes and edges and proceed with statistical techniques to analyze the pairwise interactions. However, for topological data analysis, we translate data into a simplicial complex and employ topological schemes (such as Persistent Homology) to examine the global patterns of interactions as an example of higher-order interaction analysis. In the following section, we present various types of inputs for TDA, describe how simplicial complexes are created and "filtered," and how the pattern of interactions are quantitatively examined through topological fingerprints.

Possible Inputs for Persistent Homology Analysis:

- Point Cloud
- Networks
- Scalar Fields
- Time Series

Vietoris-Rips complex

In order to form the desired simplicial complex from a complex network, or a point cloud, Vietoris-Rips(VR) complex (the Rips complex) method is one of the most common approaches. To build a Rips complex, we first define a distance metric (symmetric $N \times N$ matrix) of pairwise distances between data points and define a proximity parameter ϵ. Then, for each $\epsilon > 0$, we construct a simplicial complex ψ_ϵ based on the condition that every collection of $k + 1$ data points is a k-simplex provided that the pairwise distance between points is less than ϵ [51]. Hence, the 0-simplices are the vertices or data points themselves. Two points form an edge between them (1-simplex) whenever they are within distance ϵ of one another. Likewise, three vertices can form a triangle (2-simplex) whenever they are pairwise within that distance ϵ; correspondingly, a 3-simplex (a tetrahedron) is formed whenever four points are pairwise within ϵ of one another. Fig 4.8 shows an example of a VR complex, in which the grey circles represent $\epsilon/2$ balls so that an edge connects two vertices if their balls intersect.

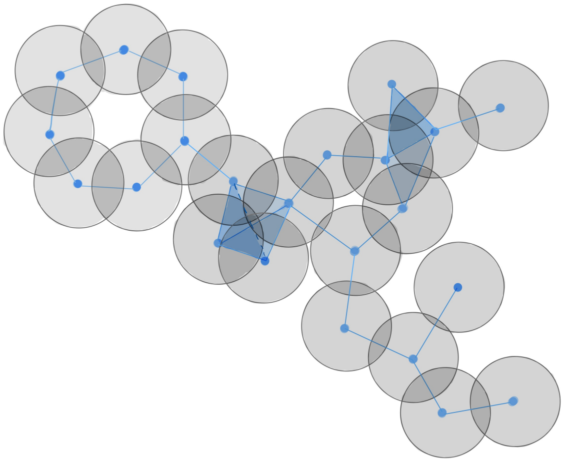


Figure 4.8: Point cloud data (vertices) form 0-simplices, and if their surrounded area (grey filled circle) intersect, they create a 1-simplex (an edge). Three vertices then form a 2-simplex (a triangle) if edges pairwise connect them, and four vertices form a 3-simplex (a tetrahedron) if edges pairwise connect them.

4.4 Persistent Homology

Homological analysis of a simplicial complex created for a scientific data set $\mathscr{D}$ means finding the homology groups of the simplicial complex $\psi(\mathscr{D})$, $\{H_k(\psi(\mathscr{D}))\}_{k=0}^{\dim(\psi(\mathscr{D}))}$ to topologically comprehend the simplicial complex $\psi(\mathscr{D})$ of the shape of the data. In this section, we will see how scientific data could be viewed as a model of a topological space to quantify its homology by creating connections between their elements, varying the scale over which these connections are made, and looking for characteristics that persist across scales. This method is called Persistent Homology(PH). Persistent Homology has been employed in a wide range of applications to uncover the topological features of data, including neuroscience, Natural Language Processing(NLP), biological and medical studies, computer vision, and sensor networks [46, 59, 12, 31]. In Persistent Homology approach we explain how to use Betti numbers defined in Eq 4.8 of a simplicial complex ψ for the analysis of graphs. We base our discussion and definitions on [55, 17, 28, 51, 16, 24, 35].

4.4.1 Filtration

After creating a Rips complex ψ as a global object from a set of discrete N data points. In this section we demonstrate how we can apply Persistent Homology to extract topological fingerprints of the desired simplicial complex to characterize our scientific data. We start by giving a definition for subcomplexes as:

Definition 4.4.1 (Subcomplex of a simplicial complex)**.** A **subcomplex** of a simplicial complex is a subset of simplices that satisfy the properties of a simplicial complex.

We expect the choice of ϵ to highly affect the formation of the consequent Rips subcomplexes and their homology. For example, considering Fig 4.8, one can predict that with small values of ϵ, the Rips subcomplex will mainly consist of isolated vertices. In contrast, the whole data set will become a single connected component for larger values. Hence, upon varying the threshold, we can now build sequences of simplicial complexes that form subcomplexes of one another, resulting in a family of subcomplexes ranging from small ϵ into those for larger values. We have obtained an inclusion map of simplicial complexes for $\epsilon_1 \leq \epsilon_2 \leq ... \leq \epsilon_k$, along which we can define filtration:

Definition 4.4.2 (Filtration)**.** A **filtration** of a simplicial complex ψ is a nested sequence of subcomplexes starting with the empty complex $\emptyset$ and ending with the full simplicial complex:

$$\emptyset \equiv \psi_{\epsilon_1} \subseteq \psi_{\epsilon_2} \subseteq ... \subseteq \psi_{\epsilon_{k-1}} \subseteq \psi_{\epsilon_k} \equiv \psi. \tag{4.9}$$

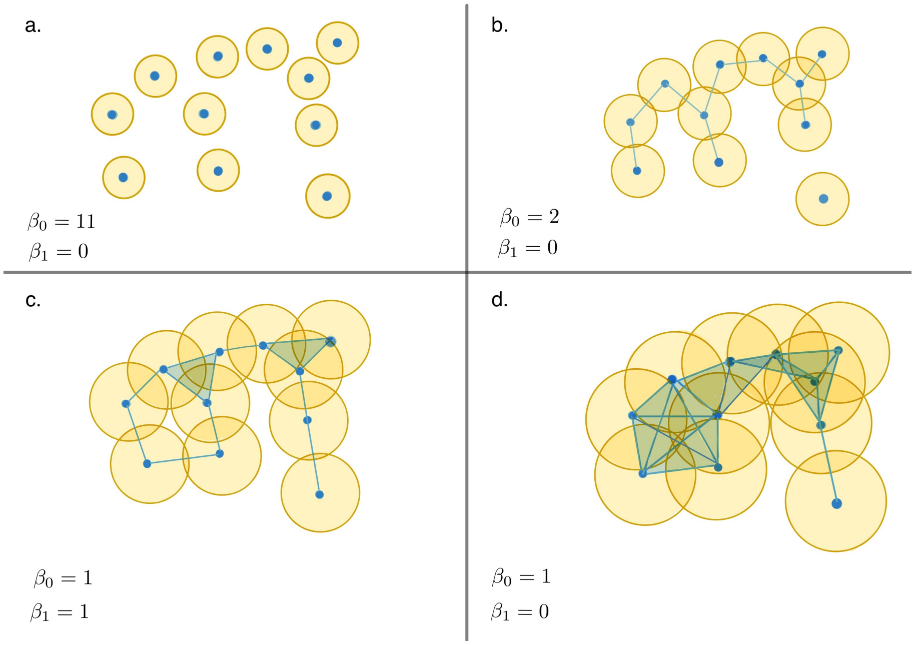


Figure 4.9: Example of a simple filtration process of a Vietoris-Rips complex where upon increasing the threshold from a to d (the radius of yellow circles) 0 to 1, the Rips complex undergoes through some topological changes. Betti numbers are used to keep track of these topological features. Betti0 β_0 counts the number of connected components, Betti1 β_1 counts the number of 1-dimensional holes(loops).

4.4.2 Persistence: Birth and Death of a Homology Class

For any simplicial complex $\psi^{\mathscr{D}}(\omega)$ formed over proximity parameter ω where $\omega \in \{0, 1, 2, ..., n\}$, we can find homology groups in every step $i \in \{0, 1, 2, ..., n\}$ of the filtered simplicial complex $\Phi(\psi^{\mathscr{D}}(\omega))$. Where there exist only one proximity parameter ω_i for each filtration step as:

$$\forall i \in \{0, 1, 2, ..., n\} \quad \exists \quad \omega_i \in [\omega_{min}, \omega_{max}] \quad ; \quad \begin{cases} \omega_0 = \omega_{min} \\ \omega_n = \omega_{max}. \end{cases} \tag{4.10}$$

Persistence of homology groups H_k of the simplicial complex $\psi^{\mathscr{D}}(\omega)$, or the k-homology group is defined as the collection of maps such that:

$$\left\{\phi_i \,|\, \phi_i : H_k(\psi^{\mathscr{D}}(\omega_i)) \rightarrow \{0, i\}\right\}_{i=0}^{n}, \tag{4.11}$$

and for each homology class h_k of the k-homology group $H_k(\psi^{\mathscr{D}}(\omega_i))$, we can define the persistence homology as:

$$\forall h_k^{(i)} \in H_k(\psi^{\mathscr{D}}(\omega_i))) : \quad \phi_i(h_k^{(i)}) = \begin{cases} i & ; \quad \exists h_k^{i+1} \in H_k(\psi^{\mathscr{D}}(\omega_{i+1}))) : h_k^{(i)} \cong h_k^{(i+1)} \\ 0 & ; \quad otherwise \end{cases} \tag{4.12}$$

Thus, from the persistent homology perspective, during filtration process, different topological possibilities will appear and disappear for the Rips complex. Which using the persistence we follow when a prominent topological feature, such as a homology class, of the simplicial complex $\psi^D(\omega)$ first arises in the filtration process and when it vanishes. In another words birth and death of a homology class follow as:

Definition 4.4.3 (Birth and Death of a homology class). A homology class $h_k^i \in H_k(\psi^{\mathscr{D}}(\omega))$ is born at $\psi^{\mathscr{D}}(\omega_i)$ if h_k^i is an element of $H_k(\psi^{\mathscr{D}}(\omega_i))$ but is not in the image of the inclusion map $\psi^{\mathscr{D}}(\omega_{i-1}) \hookrightarrow \psi^{\mathscr{D}}(\omega_i)$, and the homology class h_k^i dies entering $\psi^{\mathscr{D}}(\omega_{j+1})$ if h_k^i in an element of $H_k(\psi(\omega_j))$ but is not in the image of the inclusion map $\psi^{\mathscr{D}}(\omega_j) \hookrightarrow \psi^{\mathscr{D}}(\omega_{j+1})$,

where we denote the filtration step in which h_k^i is born with $\omega_b = \omega_i$, and the filtration step in which h_k^i dies as $\omega_d = \omega_{(j+1)}$. Hence the lifetime (persistence) of each homology class can be found using:

$$l(h_k^i) = \omega_d - \omega_b. \tag{4.13}$$

Persistent Homology uses Betti numbers, birth and death step of each k-homology to detect and study these topological fingerprints of the data sets. The topological features we examine in the simplicial complex, include connected components denoted by β_0, 1-dimensional holes (loops: β_1), and 2-dimensional holes (voids: β_2). Regarding connected components of the complex, for instance, PH keeps track of the threshold at which each connected component appeared (born) and the threshold at which two separate components merged (died). Likewise, we track the threshold that each hole(one or two dimensional) is formed (born) and when it is filled (dies) [4]. Fig 4.9 shows a simple example of a filtration process. The filtration starts with 11 nodes(0-simplices), and accordingly eleven Betti0, $\beta_0 = 11$. In the second filtration we have vertices (0-simplices) connected through (1-simplices) and the subcomplex has 2 connected components ($\beta_0 = 2$, no loop detected). Upon increasing the threshold (the radius of disks) we get a loop in the third stage of filtration ($\beta_1 = 1$) which later disappears in the last stage where the complex has become one component ($\beta_0 = 1$). Using the evolution of these Betti numbers now we can examine different topological features in data, appearing and disappearing.

4.4.3 Persistence Barcode and Persistence Diagram

Persistent Barcodes(PB) and Persistent Diagrams(PD) are the most common way to visualize persistent homology through a graphical representation. These plots are being used in topological data analysis as representations of PH to summarize topological information of the data-set.

Barcodes

Persistence Barcode for the persistence of a k-homology (k-Persistence Barcode) is a way to visually represent the lifetime of a k-homology in $\mathbb{R}^2$. Where the horizontal axis shows $\omega_i \in [\omega_0, \omega_n]$, and the vertical axis presents homology classes of k-homology groups h_k^i. Please note that the ordering is arbitrary. Each homology class in represented using the corresponding persistence interval:

Definition 4.4.4 (Persistence interval)**.** The persistence interval for a homology class $h_k^i \in H_k(\psi^{\mathscr{D}}(\omega_i))$ is given by $[\omega_b(h_k^i), \omega_d(h_k^i))$.

Persistence Barcode, thus, can be thought of as a version of Betti numbers defined in Eq.4.8 to depict persistence intervals of homology classes.

Definition 4.4.5 (Barcode)**.** A k-dimensional barcode for a filtered simplicial complex $PB_k(\psi_\omega^{\mathscr{D}})$ is a collection of horizontal line segments on a plane representing the persistentce intervals of homology generators of the $k-th$ homology group arbitrarily ordered along the vertical axis.

$$PB_k(\psi_\omega^{\mathscr{D}}) = \big\{ \big[\omega_b(h_k^i), \omega_d(h_k^i)\big) \quad | \quad i = 0, 1, 2, .., n \big\} \tag{4.14}$$

Fig 4.10 shows an example of the process of tracking topological fingerprints and filtration of a Vietoris-Rips complex over proximity parameter ϵ. The top four figures indicate the Rips complex of 18 points for different values of ϵ. As we can note from the figure, we can draw the life spans of these topological features as life bars, the length of which reveals how persistent a component or hole is before it merges or is filled [51]. The horizontal axis of the bottom plots of Fig 4.10 tracks the radius of the circle or the threshold of proximity parameter, and the vertical lines correspond to these four levels of ϵ. The number of horizontal bars gives the number of topological features at their starting threshold (birth) and death, which arranging them together for any complex provides us with the corresponding Persistence Barcode (PB) being trcked on the vertical axis, with the x-axis showing the varying threshold, and each bar corresponding to a topological feature for that Rips complex [47].

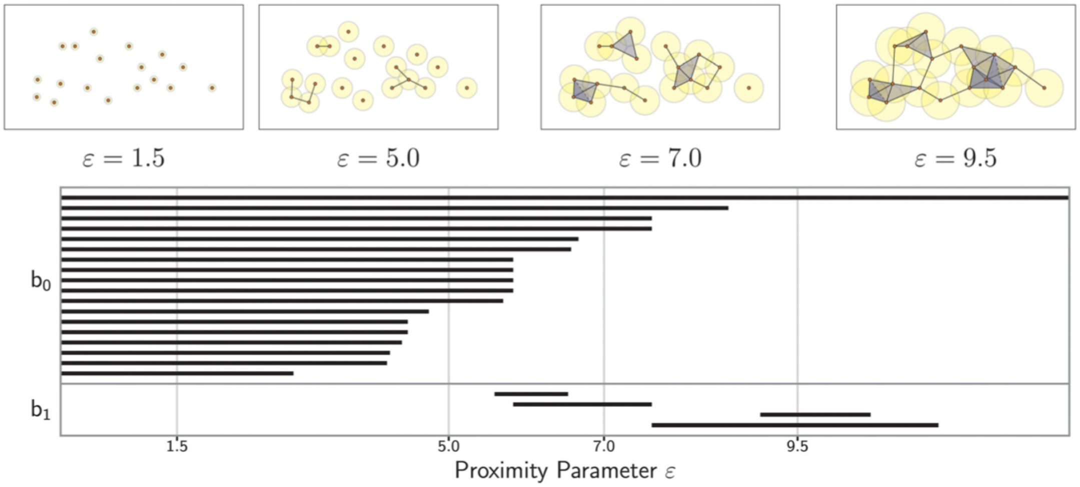


Figure 4.10: Example of a Vietoris-Rips complex together with the persistent barcode for the birth and death of 1-dimensional holes of the complex [51].

Persistence Diagram

Persistence Diagrams (PD) are another practical presentation of the persistence of topological features in different network weight. In the PD plot of the k^{th} dimension for weighted complex ψ, $PD_k(\psi_\omega^{\mathscr{D}})$, any topological feature is represented by a point (persistence pair) in a 2-dimensional Euclidean space. Therefore, persistent diagrams(PD), as another representation of homology groups over filtration, yields as a set of points scheming topological features of the data in terms of persistent pairs represented by $P^{h_k}(w_b, w_d)$ in Euclidean space R^2. Since for any k-homology group h_k^i we always have $\omega_d(h_k^i) > \omega_b(h_k^i)$, all the persistence pairs of the k-dimensional Persistence Diagram $PD_k(\psi)$ will appear above the main diagonal. As the distance from the main diagonal increases we have a feature with a longer lifetime, and more persistence. To obtain a persistence diagram from a persistence barcode plot, we transform the birth beginning point and the death endpoint of all the bars as x-y coordinates in a death-vs-birth plane. In Fig 4.11 using a schematic representation, we generated an example of persistence diagram obtained from the corresponding persistence barcode. Hence, in this context, using PB and PD of different dimension, Persistent Homology tracks these topological fingerprints that persist across a range of ω, and one could use the “lifetime” of these features to analyze characteristics of the data-set.

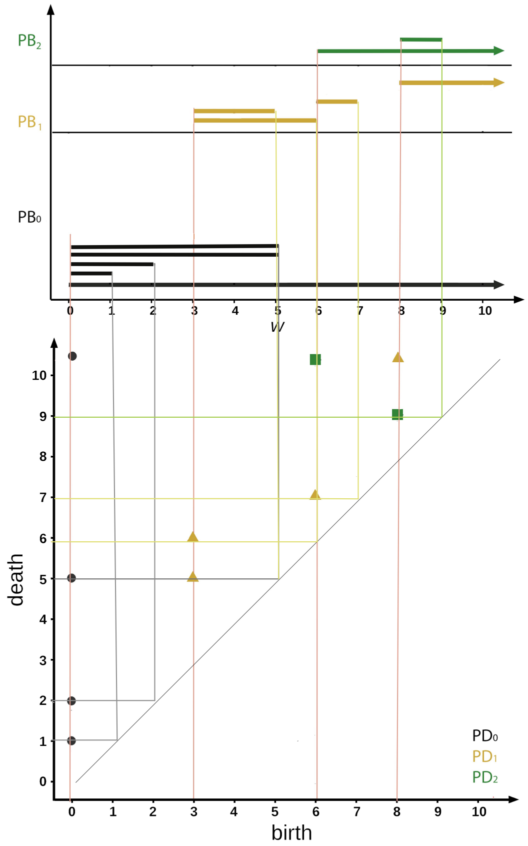


Figure 4.11: Persistence diagram (planar display) obtained from corresponding persistence barcode through translating each bar to a (birth, death) pair as x-y coordinates on the plane.

4.4.4 Betti Curves

Betti curve β_k is the frequency distribution function of persistence homology for k-homology groups $\phi_i(h_k^i)$ of a filtered simplicial complex $\psi_\omega^{\mathscr{D}}$. Betti curves track the evolution of k-dimensional holes in different dimensions over varying thresholds. For example, β_0 counts the number of connected components, where growing the number of 0-holes (β_0) implies the lack of links (1-simplices) connecting them. β_1 tracks the evolution of 1-holes (loops), where a rise in their number indicates the absence of triangles (2-simplices) to connect the nodes (agents) of a subnetwork, and β_2 tracks the evolution of 2-dimensional holes (voids, or cavities) of the data-set.

4.5 Overview of Previous Topological Studies On Proteins

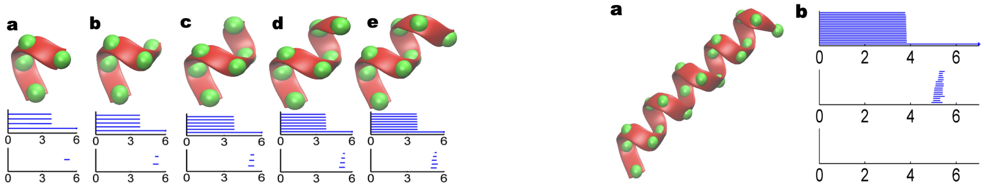


Figure 4.12: Topological Analysis of an alpha helix using slicing method for the coarse-grain representation. Every four C_α builds a one-dimensional loop in the filtration process, and by adding one more C_α atom, one more β_1 will be generated as shown in b, c, d, and e. For alpha Helix with PDB accession code: 1C26 with 19 residues, the figure shows 16 short-lived bars in the β_1 panel [57]

.

In recent decades, employing Topological Data Analysis(TDA) to study different aspects of protein has been the subject of novel studies [5, 10, 11, 27, 54]. In 2014, for the first time, Xia and Wei [57] introduced the application of TDA and, more precisely, persistent homology to extract molecular topological fingerprints. They proposed the slicing method to track the geometric origin of protein topological invariants for all-atom and coarse-grained representations of alpha helices and beta sheets. Regarding topological features of helices, using slicing method, they proposed that for coarse grained model, each α-helix the first four C_α atoms of any helix in the starting crystal structure contribute to a one-dimensional loop, where adding one more C_α generates one additional loop and, therefore, one more short-lived bar in the PB diagram (see Fig 4.12). Regarding Beta sheets for the coarse grained model,

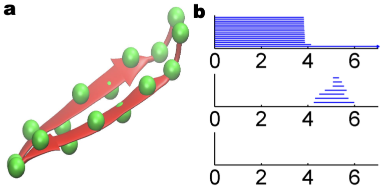


Figure 4.13: Tracking topological fingerprints of the coarse-grained model of two beta strands (2JOX). The figure shows eight pairs of residues (16 C_α) and accordingly 16 bars in the β_0. Every two pairs of atoms contribute one loop to make up 7 bars in the $\beta - 1$ [57].

it has been reported that any two pairs of C_α atoms contribute a one-dimensional loop, yielding bars in the PB1 panel. They accordingly built and analyzed elastic network models for proteins. Furthermore, they developed correlation matrix-based filtration and generated persistent barcodes to model proteins' flexibility based on the correlation between protein compactness, rigidity, and connectivity. Ultimately, they utilized the evolution of obtained topological features to denote protein topological transition during the folding process. We initially studied different computational approaches that have used topological methods to clarify the proteins' function using the residues' static spatial coordinates. For instance, Edelsbrunner & Harer [16] explored applications of topology-based computational methods to predict protein interactions and protein docking. Moreover, Gameiro et al. [22] paper aimed to clarify the relationship between the compressibility of a protein and its molecular geometric structure. They used alpha filtration on several proteins' persistence diagrams and compared the experimental compressibility of most studied proteins. Using Persistent Diagrams, they found a measure of compressibility based on topological features (tunnels and cavities) and designated a transparent correlation between their topological measure and the experimentally-determined compressibility of most proteins. Kovacev-Nikolic et al. (2016) [32] used TDA and precisely the method of persistent homology and dynamical distances to analyze protein binding and indicated a clear separation between the final closed and open protein conformations using TDA. They also suggested that the active site residues and allosteric pathway residues of the protein under their study is located in the vicinity of the most persistent loop in the corresponding filtered Vietoris-Rips complex to confirm the importance of topological fingerprints. The input for their topological analysis was a 370 × 370 matrix of dynamical distances for each conformation. However, these intuitive approaches are inefficient as protein undergoes different local and global conformational changes in time or when it goes though the unfolding process. We aim to capture the evolution of topological features in time to account for possible topological contributions to protein activity.

4.6 Results of Persistent Homology Analysis of Protein Interaction Network of CalB

As formerly mentioned, a k-hole of different dimensions in the space is a subspace with "no boundary and is not being a boundary itself," demonstrating a lack of higher-order connections between the nodes of data. Tracking the number of k-dimensional holes of the network by plotting their evolution as a function of weight in terms of Betti numbers (β), Persistent Barcode, and Persistent Diagram is one crucial point for analyzing the topological features of different protein networks. In order to use CalB 3D structure as an input for Topological study, we could utilize both the all-atom structure as an example of point-cloud or use the Protein Residue Network we built earlier using α Carbons. The all-atom model from MD Simulation provides an atomic characterization of the protein, whereas the residue network of CalB as a coarse-grained representation illustrates the protein molecule with a reduced number of degrees of freedom and can sufficiently highlight significant features of the structure. Hence, we chose the residue network as an efficient input for our further Topological studies. Regarding filtration process, since Vietoris-Rips filtration is widely used in practice, we focused our effort on studying the topological properties of the protein network of this particular kind of filtration. We utilized Vietoris-Rips filtration method of Dionysus Python package [44] to form simplicial complexes (simplices), and calculate the number of topological fingerprints of the network for both the crystal structure and also tracked the evolution of topological features in time. We further tested the robustness of our analysis using different filtration methods such as Alpha filtration and other packages such as Gudhi libraries [43].

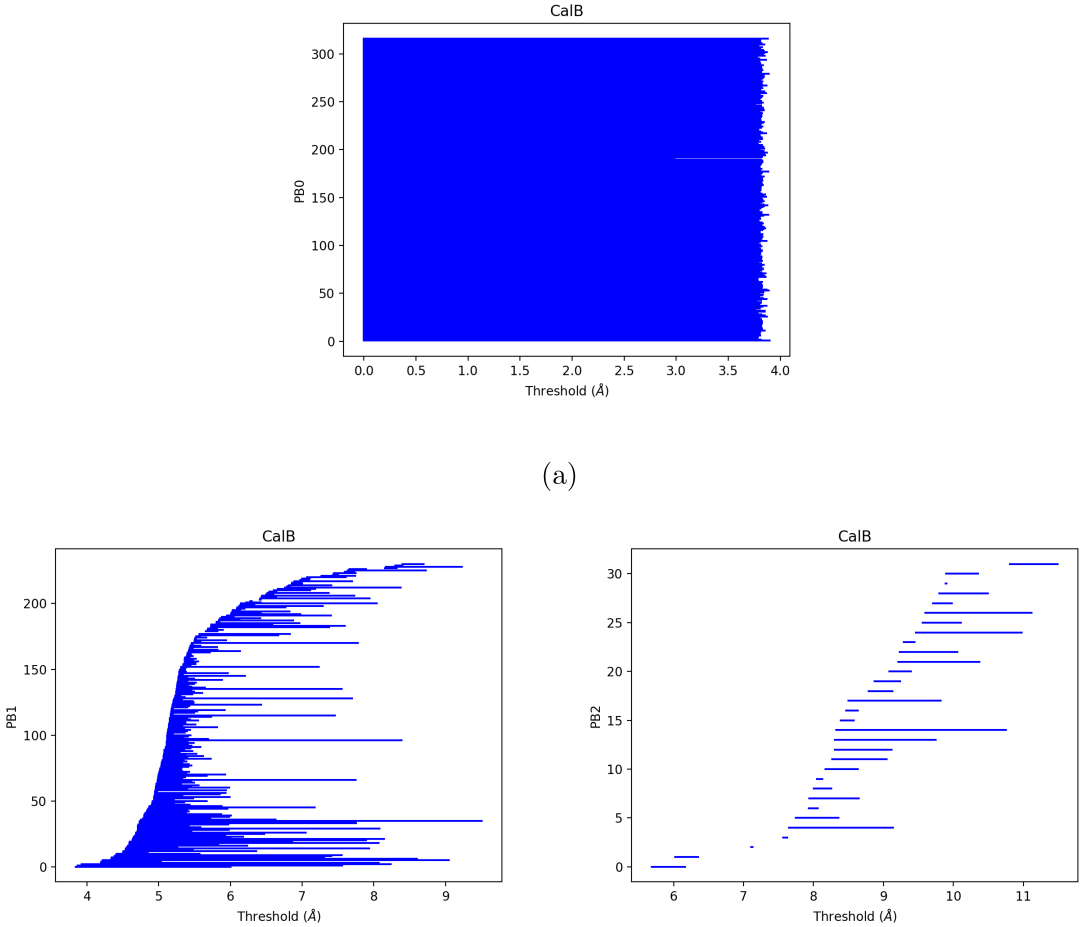
4.6.1 Persistent Homology on Protein Residue Network of CalB Crystal Structure

CalB 3D structure downloaded from protein data bank (1TCA) consists of nine beta sheets and sixteen helices, with its topological features including isolated entities, rings, and cavities. Each helix has a coiling conformation, with each spiral in the backbone made of 3.6 amino acid residues, connected through a hydrogen bonding, and each beta-sheet consists of 3 to 10 amino acids. According to the work of Xia [57], each alpha helices and beta sheets, together with neighbouring structures will generate their corresponding topological features which can be detected using persistent barcodes. As protein goes through different conformational changes in time, we can detect the topological changes and their effect on the functionality of the protein.

In Fig 4.14, we first plot the 0-2 dimensional persistent barcode for the crystal structure of CalB. As previously mentioned, PB_0 counts the number 0-dimensional holes (connected components), PB_1 counts the number of 1-dimensional holes (loops), and PB_2 is responsible for counting 2-dimensional holes(cavities) of the network over varying the threshold. As it can be noted from the graph, there are 317 bars in PB_0 (connected components) at small thresholds, representing the number of nodes of the graph (α carbons of the protein). Moving to bigger weights, persistent bars in PB_0 start to disappear as nodes get connected and loops and cavities are generated. The number of loops and cavities of the CalB crystal structure at varying threshold can be tracked by PB_1 (4.14b) and PB_2 (4.14c) respectively. As Figure shows we see the maximum number of loops at around 5Å, representing that at this distance most of the patterns of interaction in terms of topological features exist. This number according to the study by Xia [57] mostly come from the secondary structure. We also noticed that the most persistent loop of the network dies at around 9Å and we didn't detect any loop at higher threshold. Fig 4.14c shows cavities of the network. The number of 2-dimensional holes (cavities) increases as more loops are filled and generates the planes of cavities. Hence, at higher thresholds (above 8Å) upon disappearing the loops of the network PB_2 increases. As the threshold keeps increasing beyond 11Å, the general shape of the VR complex made from our protein network would not undergo significant topological changes and stays as a single component with no one or two-dimensional holes. Figure 4.15 shows 0-2 dimensional persistent diagram for protein network of crystal structure of CalB. As it can be seen from 4.15, the network becomes one-component above 4Å. Persistent pairs with longer lifetime tend to be located further from the main diagonal and correspond to more robust topological feature. One-dimensional holes (loops) of data-sets are usually referred to as interaction pathways between the nodes, and our analysis shows that for crystal structure of CalB the most persistent loop of the network has a lifetime of around 5Å. Likewise, the most persistent two-dimensional hole (cavity) of the crystal structure has the lifetime of 2.4Å. Since, as we previously seen the helices and sheets of the secondary structure only contribute to the loops of the data-set, the cavities (2-d holes) are only the result of 3D structure and the way these helices and sheets fold.

4.6.2 Persistent Homology on MD Simulation Results

By examining the protein interaction network from topological framework, in this section we aim to shed some lights on the patterns of interaction between residues in different systems. Applying PH on the weighted complex networks of different systems of the protein, we analyze the evolution of the dimension of the k-homology group of the topological space β_k. As mentioned earlier each Betti number in each dimension

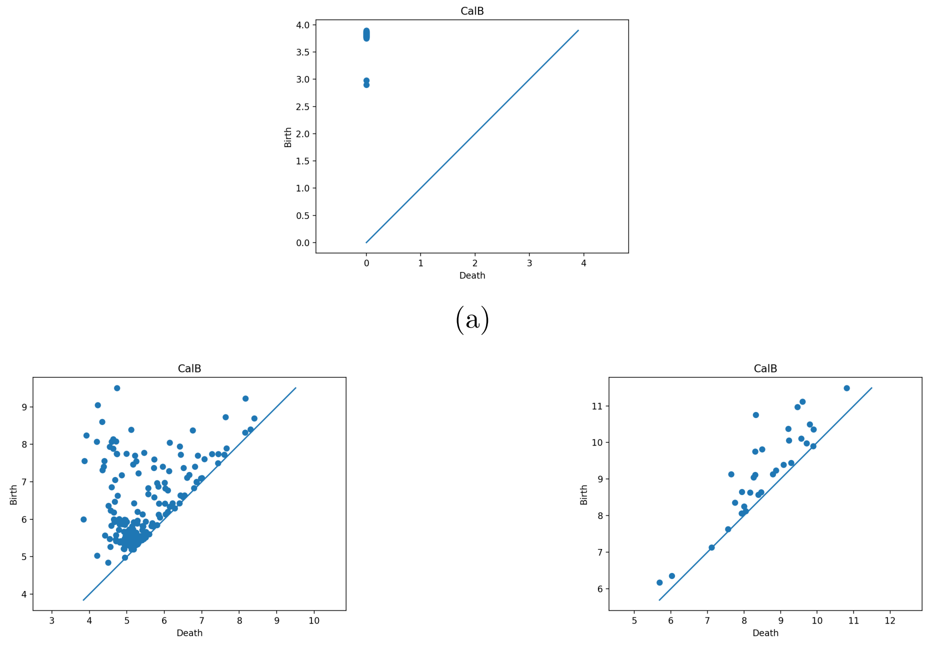


(a)

(b)

(c)

Figure 4.14: a. 0-dimensional Persistent Barcode, b. 1-dimensional Persistent Barcode and, c. 2-dimensional Persistent Barcode over varying threshold for protein interaction network of CalB crystal structure.



(a)

(b)

(c)

Figure 4.15: a. 0-dimensional Persistent Diagram, b. 1-dimensional Persistent Diagram and, c. 2-dimensional Persistent Diagram, over varying threshold for protein interaction network of CalB crystal structure

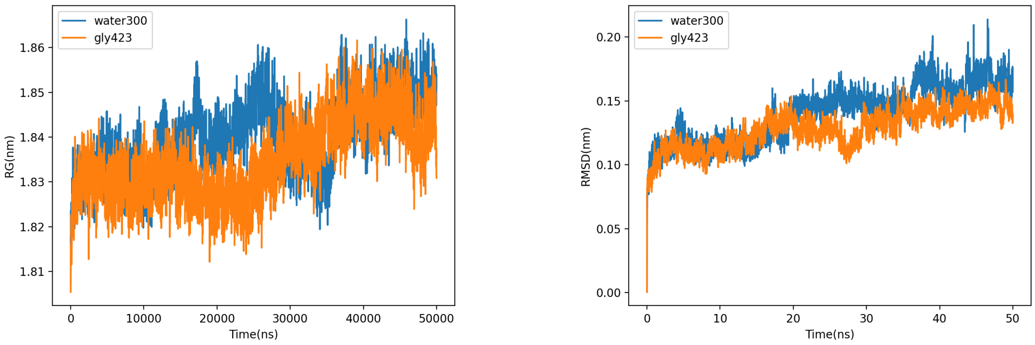


Figure 4.16: RMSD and Rg values for CalB shows protein maintains the same 3-dimensional structure in both states

demonstrates the number of k-dimensional topological hole. CalB structure seems to be folded in both glycerol at 423K and in water at 300K, where Fig 4.16 shows the patterns of RMSD and R_g in these two states; however, active site analysis showed how active site residues move away from one another in water. This motivated us to look into topological changes of the protein network of CalB in time for different states to look for possible topology-function relation that can not be detected in three dimensional conformation of the protein. We applied Persistent Homology technique to the MD simulation results of CalB protein interaction network to firstly topologically compare the folded, active, and inactive state of CalB, and account for the possible contribution of topological features on the activity of the protein and also to study from the topological perspective how local changes in the network can create functionality at the scale of the entire network.

In order to apply TDA to the protein interaction network, we built the filtered Rips complex $\psi_\epsilon(\mathscr{D}, t)$ for each time frame ($0 < t <$ 50ns) of the adjacency matrix introduced in chapter 3, over varying ϵ, where $\epsilon = \omega_{ij}(t)$ defined in Eq 3.3, and $\mathscr{D} = A(t)$ and use persistent homology to look for $k = 0, 1, 2$ dimensional homology groups h_k in each time frame. Moreover, by using barcodes and persistence diagrams, we can look more closely into which local topological features are more critical, and their role in the entire network can be examined. Hence, we first report the analysis of Persistent Barcode(PB) and Diagrams(PD) for the final time-frame ($t = 50ns$) of Rips simplicial complex $\psi_\epsilon(\mathscr{D})$ to find the most prominent local topological features and search for the possibility of their effect on the functionality at the scale of the entire network. We wish to extract information about the “most important” or “most robust” topological features observed from the length of bars in the PB diagram, which correspond to persistent features with the most extended “lifetime” over varying the threshold. In representing the $k = 0, 1, 2$ homology groups, we use different colors for different systems and plot each homology group generators on a separate chart. To show the evolution of Betti curves in different dimensions (β_0, β_1, β_2), since the total number of topological features is time-dependent and changes during our simulation, we calculate Betti curves for each time frame ($\beta_k(\underline{t})$ for 50 time frames for $50ns$ simulation), and take the average in time and plot $\bar{\beta}_k$ curves for protein network in different states. Moreover, we track the total number of 1-dimensional holes of the data-set for different states of CalB in time, which can be assumed as the pathways of interactions between residues, and will ultimately represent so called “Persistent Entropy” plots of the networks in different states to uncover our insights into the patterns of interactions in active and inactive states of CalB.

0-dimensional Topological Changes (Connected Components)

We used the final snapshot of CalB after $50ns$ simulation in different conditions as the input for local topological changes analysis and plotted the PB_0 and PD_0 for the protein network. Figure 4.17 illustrates the 0-dimensional topological changes of the protein networks versus the network weight (distances) in different systems. This figure shows that the protein network becomes fully connected in all three systems upon the last filtration stage ($\omega > 0.4$), and no isolated entity is left throughout the whole network.

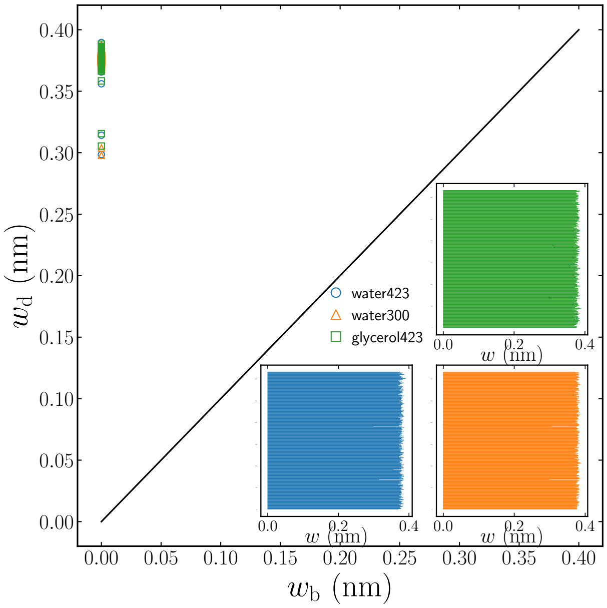


Figure 4.17: 0-dimensional Persistence Barcode and Persistence Diagram for protein network in three different states. The 0-dimensional topological fingerprints are born in the first stage of filtration and died in the last stage at about 0.4nm

Fig 4.18 shows the time-averaged evolution of β_0 versus the varying weight of protein network in different states. The β_0 curve can be used to demonstrate the bond length information. One can notice from the β_0 curve that the number of β_0(connected components) for the protein networks falls sharply at about 0.4 nm, which physically implies the bond length between amino acids and is reflected in the distance-based filtration. Hence, reaching this threshold, amino acid residues start to bind, causing the network to become one component and stays as one component until the last filtration stage, and no isolated components persist when proceeding to higher threshold values. The β_0 curves for water at 300K and 423K networks indicate that some isolated components persist longer in the "inactive" state of CalB. The

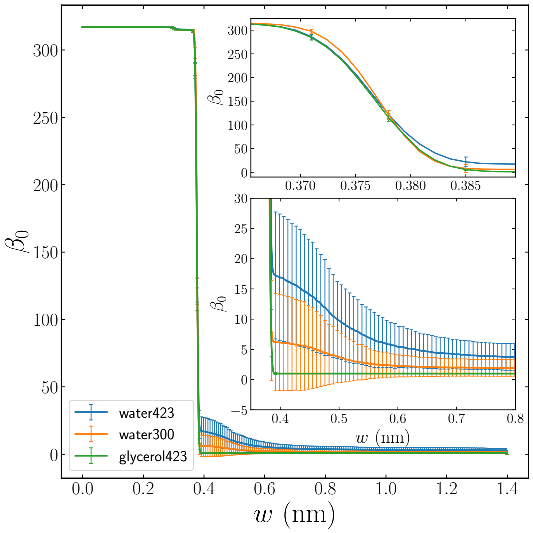


Figure 4.18: 0-dimensional topological changes (β_0) curve for protein network in different states averaged over time. The error bars for each curve shows the topological changes in time. The system becomes one component in the last stage of filtration.

error bars emphasize how the total number of topological features changes over time. We have used two subplots in Fig 4.18 to emphasize the sharpness of the curve in glycerol network and highlight the larger error bars for protein network in water. The sharpness of curves and the error bar value can also be interpreted using the analysis of link distribution function plotted in chapter 3. Our probability distribution function analysis of weights (Fig 3.5) of the network implies that in glycerol network total number of links with stronger interaction is higher compared to water system. The sharpness of glycerol mean of the distribution in Fig3.8 and the standard deviation in Fig 3.7 shows that this system undergoes the topological evolution more promptly, as their links are restricted to smaller values.

1-dimensional Topological Holes (Loops)

PB_1 and PD_1 trace the lifetime of 1-dimensional topological features (loops) of the network and can provide a suitable representation of how local topological changes can affect the entire network. Upon calculating and plotting a 1-dimensional persistence barcode and persistence diagram for the final time frame of our simulation, we can shed some light on the most persistent or most robust topological feature of the network and its effect on the entire system. In Fig 4.19 we have plotted the 1-dimensional Persistence Barcode and Persistence Diagram for protein networks in different states. Relative to the persistence barcode of the crystal structure, we no-

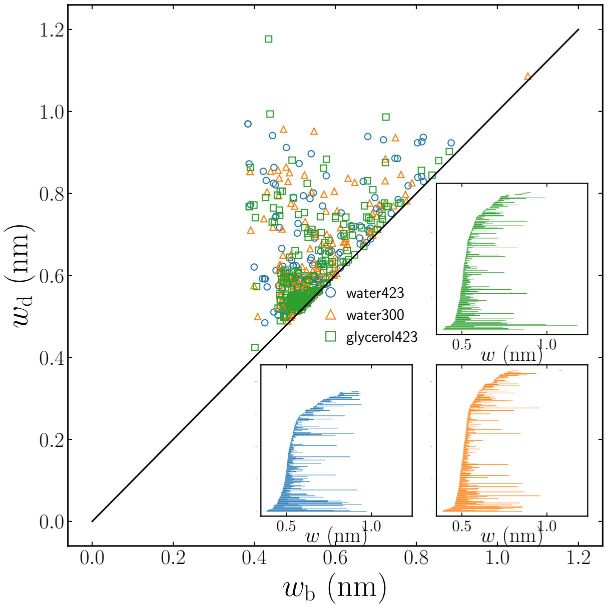
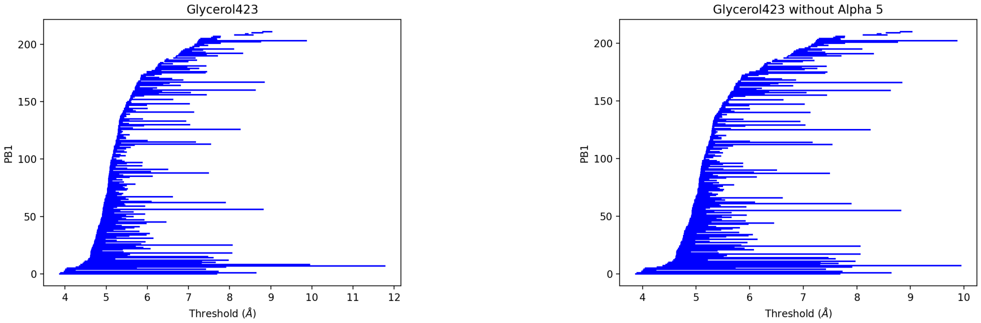


Figure 4.19: 1-dimensional Persistence Barcode and Persistence Diagram for protein network in three different states. The figure shows that the most persistent loop of the network with the lifetime of about 0.7nm dying at around 1.2nm appears in glycerol network



(a)

(b)

Figure 4.20: (a)Persistent Barcode of CalB after $50ns$ in glycerol at $423K$. (b) same frame after removing helix $\alpha5$. The barcode shows how the most persistent bar was due to the existence of helix $\alpha5$.

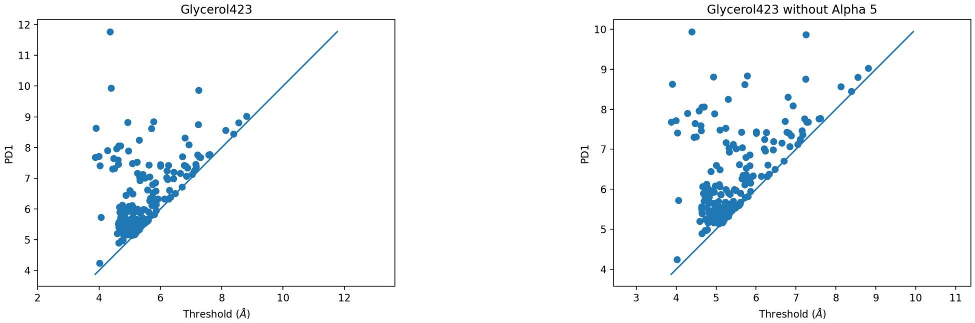
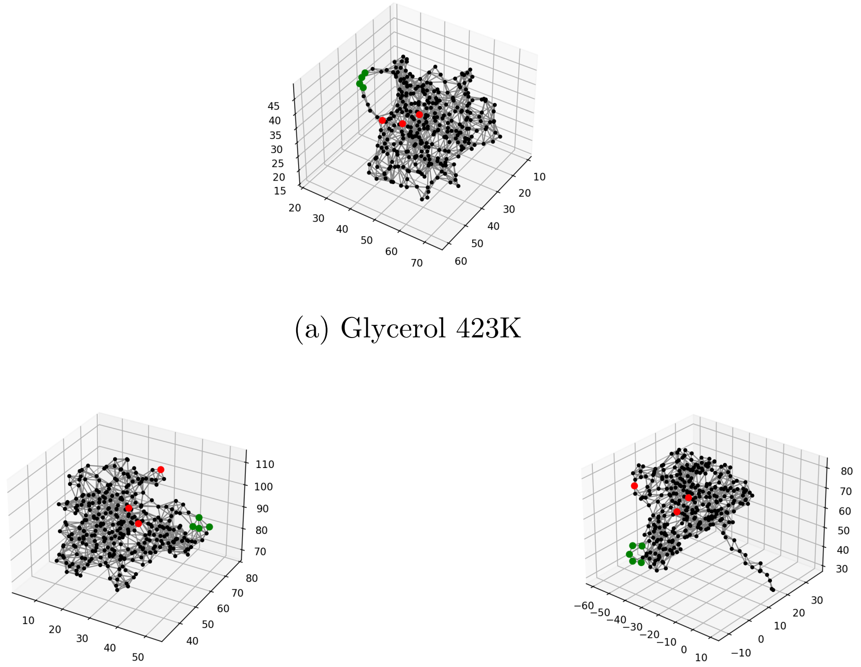


Figure 4.21: (a)Persistent Diagram of CalB after $50ns$ in glycerol at $423K$. (b) same frame after removing helix $\alpha 5$. The most persistent feature of the diagram which dies at around 12Å disappears after removing the helix.



(a) Glycerol 423K

(b) Water 300K

(c) Water 423K

Figure 4.22: Protein residue network of CalB in different states, plotted with α_5 residues colored in green and active site in red.

ticed that the lifetime of the most persistent features has increased for the protein network in all three systems. As loops of the biological systems networks are usually referred to as the pathways of interaction, one can suppose that the network nodes are constructing their paths of interaction via these loops. The 1-dimensional loops of the network have decreased slightly in water at $300K$ both in total number and also lifetime and has reduced substantially in water at $423K$. For the protein network in the glycerol system, we detect a robust 1-dimensional topological hole with the most significant lifetime compared to others of around 0.7nm. The figure shows how the noticed persistent 1-d feature (the longest bar in PB and the most persistent pair in PD), being born at around 0.4nm and dying at around 1.2nm, tends to stand out in both subplots. As this prominent feature should note some significant interaction, we first hypothesized that this feature might stem from the active site region in glycerol. Our further analysis proved that this notable feature, appearing in the active state of the protein residue network, corresponds to topological alteration of $\alpha - 5$. In Figure 4.20, and Figure 4.21 we remarked that upon removing $\alpha - 5$ from the same frame, this topological feature disappears from persistence barcode and persistence diagram respectively, which further highlights the role of $\alpha - 5$ in the protein activity. Hence, to account for the local topology-function relationship of protein complex networks of CalB, we believe that the persistence of the topological fingerprint of conformational changes of alpha-5 (CalB "lid"), which gave rise to the most persistent 1 bar, affects the active site and the activity of the protein in high temperature. In Figure 4.22 we have plotted the protein residue network in different systems and highlighted the active site residues in red and alpha-5 in green to schematically show the interaction pathway created by this persistent feature.

Figure 4.23 illustrates the evolution of 1-dimensional topological features for the protein network in different systems. The evolution of β_1 for all three systems start to increase at about 0.4 nm as amino acid residues start to bind, and as we observed in Fig 3.5 the probability of links goes up. The one-dimensional interaction pathways between nodes will create the maximum number of loops at around 0.5 nm. This number decreases and goes to zero at $\omega >$ 1nm. As the cutoff distance for protein interaction networks are usually taken to be around 0.7nm, we notice upon adding more links in that distance the β_1 curve shows another peak at around 0.7nm. We can see how Betti curves error bar increases as the protein lose its activity and compactness in time in the water. Interestingly, for protein in water at $300K$ and glycerol, even though protein is considered folded and they both followed the same RMSD patterns, in the case of water, we observed a more oversized error bar and, accordingly, a decline in the total number of features [3]. This difference is more apparent in Fig 4.24 where we have plotted the total number of one-dimensional holes of graphs versus

[3] Betti curves for each state are plotted with separate scales in Appendix B

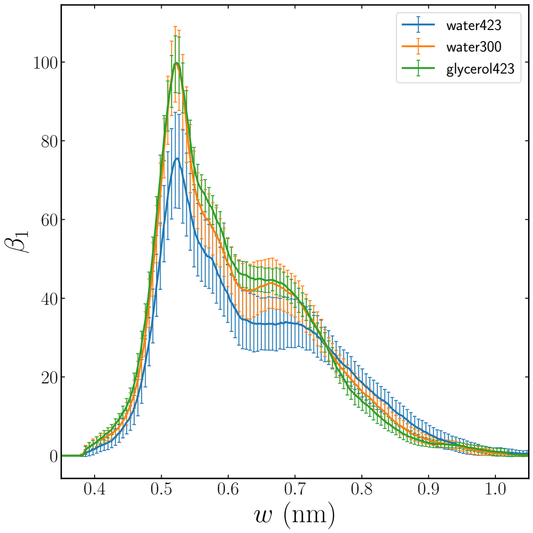
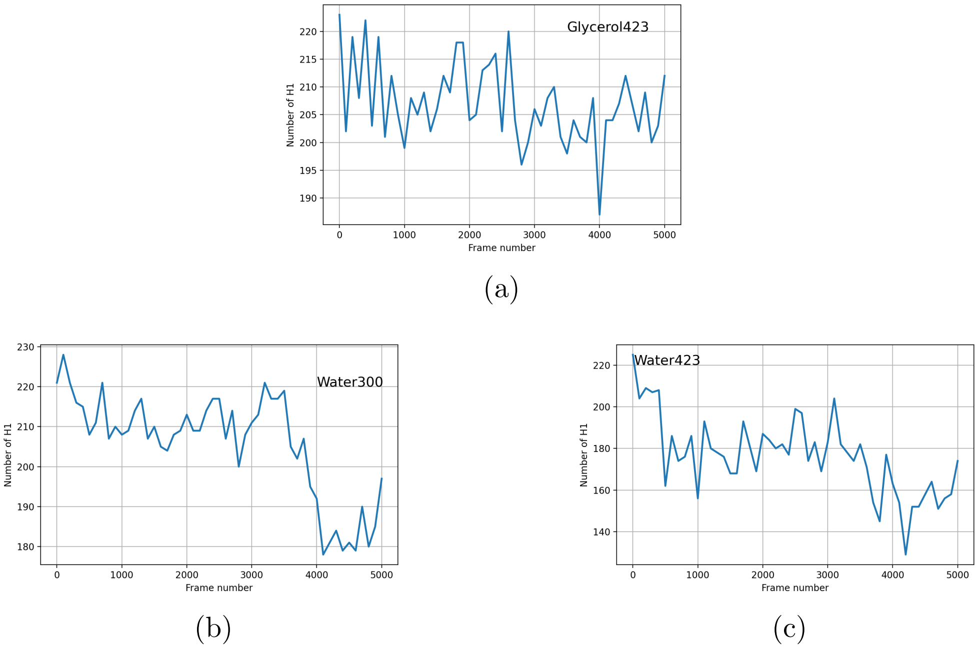


Figure 4.23: The evolution of β_1 curve shows the topological changes of one-dimensional hole(loops) of the network. The 1-hole can be considered as the pathways of interaction between the nodes. For our network having dynamic in time, we have plotted 50ns time-averaged β_1 curves for all three systems. The first peak occurs at around 0.5nm.

time. Upon the unfolding process of the protein at 423K and losing the active site activity in water at $300K$, we notice a more significant decline in the total number of features. Hence, as the protein loses its activity in time, we hypothesized that the corresponding protein interaction network loses some of its topological features.

2-dimensional Topological Holes (Voids)

2-dimensional holes or voids of the filtered simplicial complex created for protein network, can be traced using the PB_2, PD_2 and β_2 curves. We noticed total number of voids has dropped in water $423K$ upon unfolding of CalB. The β_2 curves for CalB in different states are plotted in Fig 4.26, which represents that the total number of voids has dropped in water. The distribution for 2-dimensional topological holes of the network has its first peak around 0.8 to $0.9nm$, upon the decline in the β_1. This peak has decreased in value and is shifted to higher weights for CalB network in its inactive state. In smaller values of ω where the cut-off distance for pairwise interaction ($\omega < 0.7$nm) of amino acids is, protein network in water at $300K$ shows the most number of cavities. In higher weights $1.1 < \omega < 1.3$ and before all the topological holes disappear we see that the cavities in glycerol system persist longer, showing that in the active state of CalB some prominent 2d-holes persist.



(a)

(b)

(c)

Figure 4.24: Total Number of H_1 for protein in glycerol at 423K (a), Water at 300K (b), and Water 423K (c) over 50 snapshots in 50ns time span

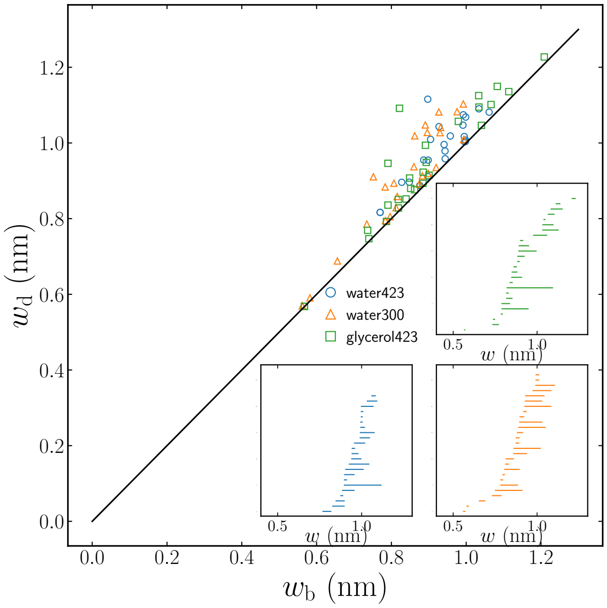


Figure 4.25: 2-dimensional Persistence Barcode and Persistence Diagram for protein network in three different states.

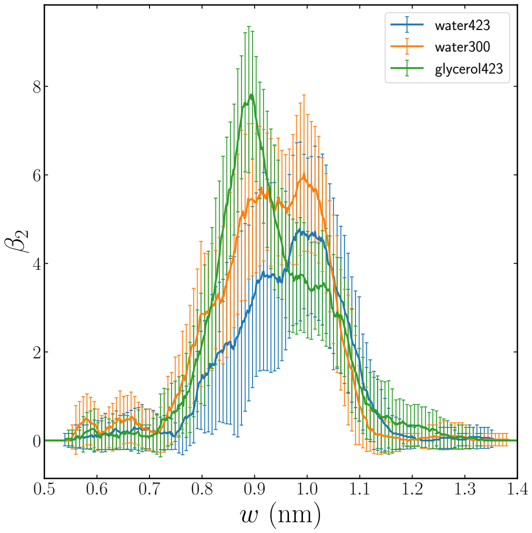


Figure 4.26: The evolution of β_2 curve shows the topological changes of two-dimensional hole(voids) of the network. For our network having dynamic in time, we have plotted 50ns time-averaged β_2 curves for all three systems.

4.6.3 Shannon Entropy and Persistent Entropy

Shannon introduced the "Shannon entropy" concept in 1948, where a measure of the uncertainty of the occurrence of a particular event, given partial information about the system, was proposed. In the context of information theory, where there exists a precise meaning for the information content of a probability distribution, the Shannon entropy (or entropy of mixing) for any probability distribution is defined as:

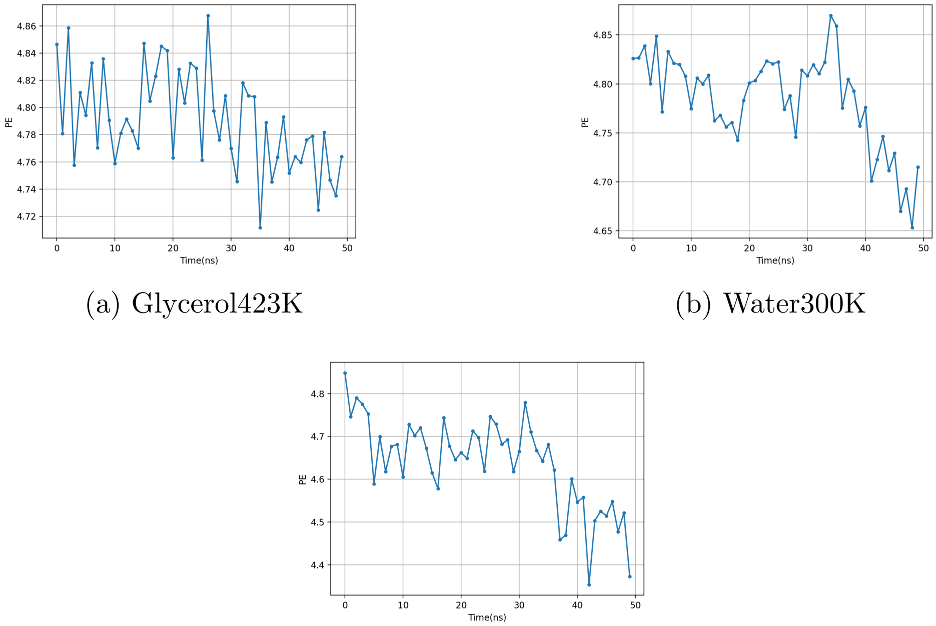
$$S = -\sum_{i=1}^{M} p(i) \ln p(i), \tag{4.15}$$

where we have considered a random variable with a discrete set of outcomes $S = x_i$ occurring with probabilities $p(i)$, for $i = 1, .., M$. S is a measure of dispersity (disorder) of the distribution[30].

Using the concept of Shannon entropy from information theory, TDA introduces a new measure of entropy, so-called "Persistent Entropy, which can calculate how much the construction of filtration is ordered and a way to measure how different the bars of a barcode are in length[6]. Given a filtration $\phi(\psi^D(\theta)) = (\psi^D(\theta_i))_{i=0}^n$ and the corresponding persistence diagram $dgm(\phi) = \{a_i = (x_i, y_i) | 1 \leq i \leq n\}$ (being $x_i < y_i$ for all i), let $L = \{\ell_i = y_i - x_i | 1 \leq i \leq n\}$. The persistent entropy $E(\phi)$ of ϕ is calculated as follows:

$$E(\phi) = -\sum_{i=1}^{L} \frac{\ell_i}{S_L} \ln \frac{\ell_i}{S_L}, \tag{4.16}$$

where $S_L = \sum_{i=1}^{n} \ell_i$. Hence, the maximum of persistent entropy occurs when all the bars in the persistence barcode are of equal length (i.e., $\ell_i = \ell_j$ for all $1 \leq i, j \leq n$) and this value decreases as more bars of different lengths become present in the barcodes. We have compared the persistent entropy for protein residue network of CalB in glycerol, water300K and water423K, we see that the persistent entropy of the network decreases by a greater amount as the protein loses its structure and activity where we find the greatest decline for water 423K.



(a) Glycerol423K

(b) Water300K

(c) Water423K

Figure 4.27: The Persistent Entropy for protein in glycerol at 423K, and water at 423K and 300K

Chapter 5

Conclusions

To this end, we first conducted a series of Molecular Dynamics (MD) simulations on one of the most widely studied lipases, Candida Antarctica lipase B (CalB), with a wide range of industrial applications and high thermal stability. MD simulation results provide us with trajectories of thousands of atoms that build the system in time. Furthermore, MD simulation analysis defines various measures based on these atomic coordinates to globally and locally account for the folded structure and activity of the protein. For instance, using Root Mean Square Deviation (RMSD) and Radius of gyration, we examine the protein's overall structure, and employing local analysis such as distances between active site residues, we probe the activity of the protein.

In Chapter 2 the results of MD simulation on CalB confirmed that the protein is folded and active in glycerol at high temperatures (423K) and loses its activity in water even at lower temperatures (300K). We also examined the structure of CalB "lid" ($\alpha 5$: residue 142-146) under different conditions and discovered that in the active state of CalB, even though the overall structure is folded, $\alpha 5$ unfolds. Furthermore, despite losing its activity in the water, the results of RMSD and R_g confirm that the CalB structure doesn't unfold at 300K in water. Hence, we concluded that we require a more suitable measure than MD simulation to distinguish between folded states of CalB, study the effect of $\alpha 5$ on the activity of CalB, and examine its high-temperature activity.

To account for the activity of the protein at very high temperatures and study the internal conformational changes occurring in the confinement of the protein, we constructed the corresponding protein residue network of CalB. The protein residue network (protein interaction network) is built using amino acids as nodes and distances between them as weighted edges. Therefore, for the protein consisting of 317 amino acids, we obtained a 317×317 matrix (called adjacency matrix for the network) with matrix elements representing the weight values. We utilized the adjacency matrix to analyze the statistical features of the network. Using the probability distribution function of distances, standard deviation, and mean of links' distribution over time, we statistically distinguished between the folded state of CalB in water at 300K and in glycerol. We confirmed that a higher standard deviation for the network distribution function in water contributes to protein-losing activity while maintaining its folded three-dimensional structure. Moreover, we found that in spite of a tiny standard deviation in time for CalB distribution function in glycerol, residues 142-146 (CalB lid) tend to stand out from the whole structure. From chapter 3, we concluded that when the protein is active in high-temperature pairwise interactions in the network form structures that are persistent in time.

The main question we aim to address from our further analysis concerns whether protein activity only depends on its three-dimensional structure or whether there are deeper patterns between residues that we need to consider studying protein activity. While network analysis emphasizes the essence of structural analysis of the components of the protein network and highlights the importance of little fluctuations in pairwise interaction between amino acids for the active state, we further analyzed the overall topology of the protein to search for features in the general interactions patterns that contribute to the activity. In Chapter 4, we employed the Topological Data Analysis (TDA) method, which characterizes the system's topology based on algebraic topology concepts and definitions.

By examining the topological features of the corresponding protein residue network, we studied the association between the protein activity and the evolution of its topological fingerprints. Firstly, regarding the local conformational changes and their effect on the active site, we found the most persistent feature of the network in 1 dimension (referred to as interaction pathways) corresponds to the residue 142-146 in the active state. Moreover, we compared the protein's total number of topological features in time and observed a greater decline in the total number in time as the protein loses its activity and compactness. To this end, we aim to establish that to study the activity of proteins, other than the 3D structure of the protein of interest, the topological fingerprints of the protein network are essential. However, we want to suggest that we have to extend the simulation time scale to confirm this decline by a greater amount.

This work is the first attempt at looking at the dynamical structural properties of a protein, including during unfolding, through the idea of topological data analysis. Previous studies have used TDA analysis of static protein structures or artificial protein dynamics. We are using for the first time with a more realistic molecular dynamics simulation.

Appendix A

Additional notes and algebra

A.1 Some Additional Definitions from Algebraic Topology

Definition A.1.1 (Map)**.** A map. f from X to Y is a rule by which for any element $x \in X$ we assign an element $y \in Y$ [41]. We write:

$$f : X \to Y \tag{A.1}$$

Remark. We call X the domain and Y the range of map f.

If f is defined by some explicit formula:

$$if \quad Y = f(x); \quad f : x \to f(x). \tag{A.2}$$

Definition A.1.2. Maps with certain properties bear special names:

- A map $f : X \to Y$ is **injective (one to one)** if any distinct pair elements of X is assigned to a distinct pair elements of Y. We write:

$$\forall x, x' \in X : \quad if \quad x \neq x' \to f(x) \neq f(x') \tag{A.3}$$

- A map $f : X \to Y$ is **surjective (onto)** if for each element of set Y there exist at least one element x in X to be assigned by the map:

$$\forall y \in Y \quad \exists x \in X; \quad f(x) = y \tag{A.4}$$

- A map $f : X \to Y$ is **bijective** and invertible if it is both injective and surjective

Definition A.1.3 (Inverse Image)**.** If more than two elements in X correspond to same $y \in Y$ **inverse image** of y is defined as a subset of X whose elements are mapped. We write:

$$f^{-1}(y) = \{x \in X \mid f(x) = y\} \tag{A.5}$$

.

Definition A.1.4 (Image)**.** The **image** of the map is defined as:

$$f(x) = \{y \in Y \mid y = f(x) \quad for \quad x \in X\} \subset Y \tag{A.6}$$

Definition A.1.5 (Relation)**.** A **relation** R defined in set X is a subset of X^2. If a point $(a, b) \in X^2$ is in R: (aRb)

Definition A.1.6 (Equivalence Relation)**.** An **equivalence relation** $\sim$ is a relation that satisfies:

- $a \sim a$
- If $a \sim b \rightarrow b \sim a$
- If $a \sim b, b \sim c \rightarrow a \sim c$

Definition A.1.7 (Equivalence Class)**.** A **class** $[a]$ is made of all the elements $x \in X$ such that $x \sim a$:

$$[a] = \{x \in X \mid x \sim a\} \tag{A.7}$$

Definition A.1.8 (Geometrically Independent)**.** n+1 points in space are **geometrically independent** if their corresponding n vectors are linearly independent. Set $\{a_0, a_1, ..., an\}$ of points in R^N with real scalars t_i is geometrically independent such that:

$$\sum_{i=0}^{n} t_i = 0, \sum_{i=0}^{n} t_i a_i = 0, \quad implies \quad t_0 = t_1 = ... = t_n = 0 \tag{A.8}$$

Definition A.1.9 (Homomorphism)**.** A map $f : X \rightarrow Y$ is called a **homomorphism** if for any two sets X and Y together with their certain algebraic structure (product, addition, etc) f preserves these algebraic structures. We write:

$$\forall x, x' \in X \quad f(x \odot_x x') = f(x) \odot_y f(x\prime), \tag{A.9}$$

where $\odot_x$ and $\odot_y$ are arbitrary algebraic operations in X and Y respectivele.

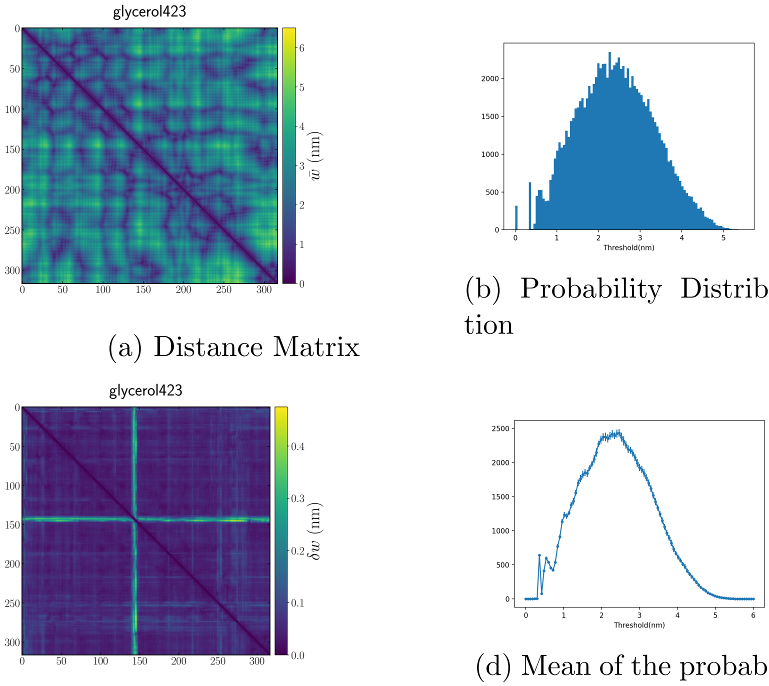
Definition A.1.10 (Isomorphism)**.** If a homomorphism f is bijective, f is called an **isomorphism** and X is said to be **isomorphic** to Y. We write: $X \cong Y$

Appendix B

Further Plots

B.1 Statistical Analysis of CalB Network in Different States

This section includes for each system, statistical analysis of CalB interaction network on a separate figure.



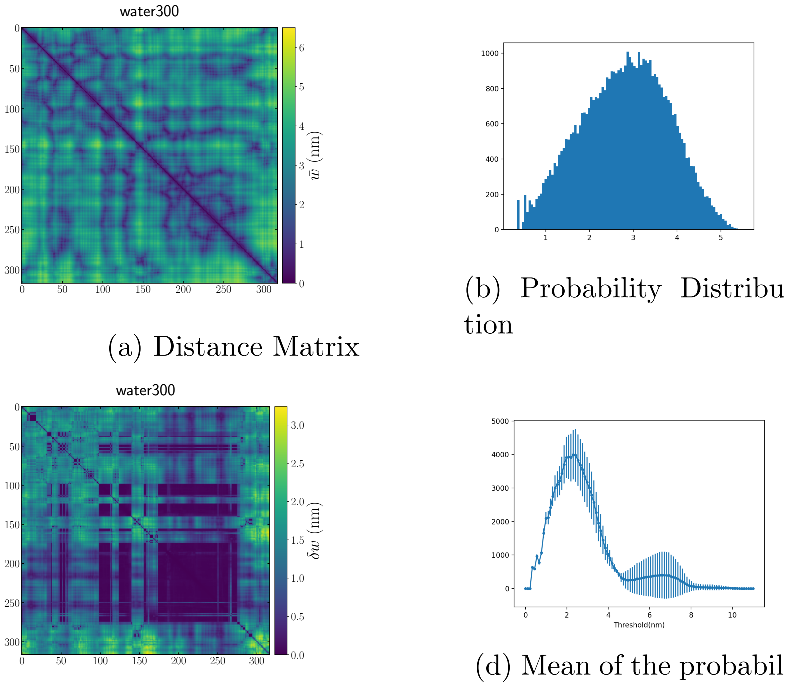
(a) Distance Matrix

(b) Probability Distribution Function

(c) Standard Deviation

(d) Mean of the probability distribution function

Figure B.1: Figure shows statistical analysis of CalB network in glycerol at 423K



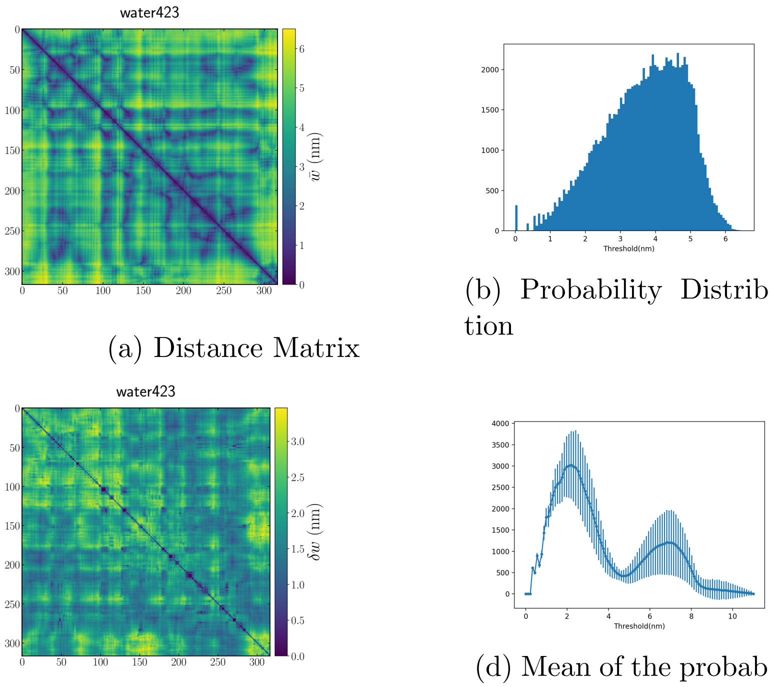
(a) Distance Matrix

(b) Probability Distribution Function

(c) Standard Deviation

(d) Mean of the probability distribution function

Figure B.2: Figure shows statistical analysis of CalB network in water at 300K



(a) Distance Matrix

(b) Probability Distribution Function

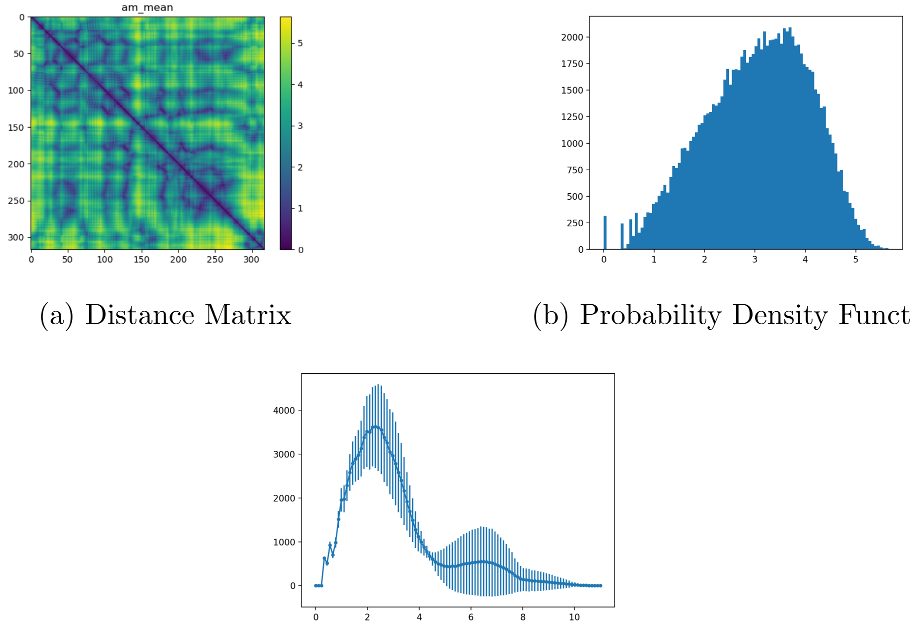
(c) Standard Deviation

(d) Mean of the probability distribution function

Figure B.3: Figure shows statistical analysis of CalB network in water at 423K

B.2 Statistical Analysis and Persistent Homology on Protein Residue Network in Urea

Here in Fig B.4 we show the distance matrix, probability density function, mean, standard deviation, and the evolution of topological features of CalB in urea at $423K$. Also In Fig B.5 we show 0 and 1-dimensional betti curves together with persistent entropy for CalB network in Urea.



(a) Distance Matrix

(b) Probability Density Function

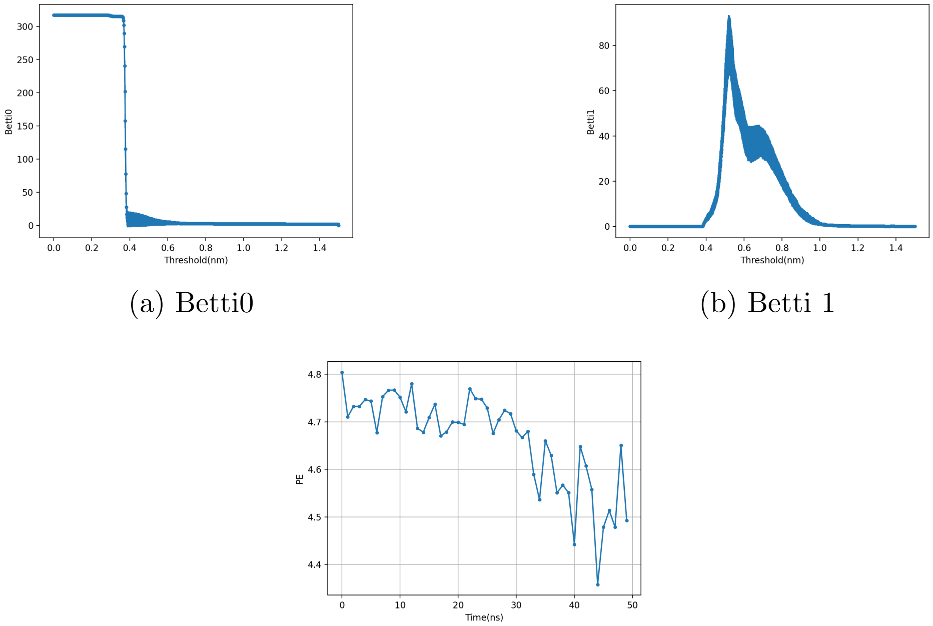
(c) Mean

Figure B.4: For the protein interaction network of CaB in urea we have plotted distance matrix, PDF and mean

B.3 Topological Data Analysis Results

B.3.1 Betti Curves

Here we show $0-1$-dimensional Betti curves for protein in each state.

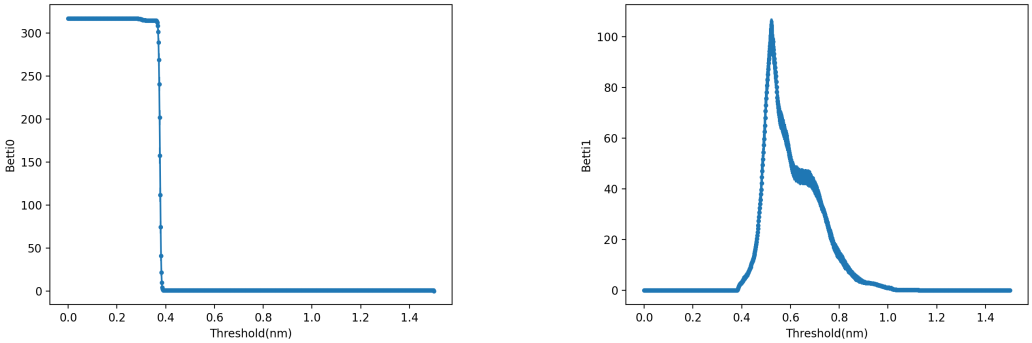


(a) Betti0

(b) Betti 1

(c) Persistent Entropy

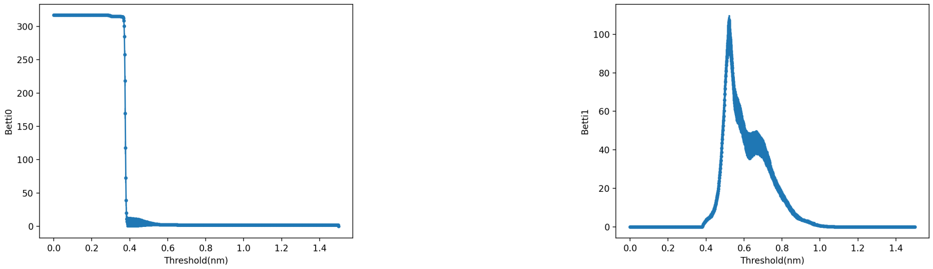
Figure B.5: For the protein interaction network of CaB in urea we have plotted 0 and 1-dimensional betti curves and persistent entropy



(a) β_0

(b) β_1

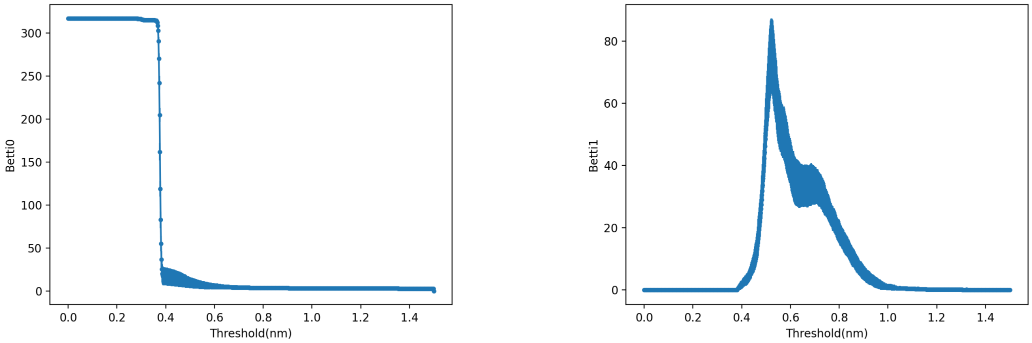
Figure B.6: the evolution of zero(left) and one(right) dimensional hole (loops) as a function of threshold for CalB in Glycerol at 423K



(a) β_0

(b) β_1

Figure B.7: the evolution of zero(left) and one(right) dimensional hole (loops) as a function of threshold for CalB in Water at 300K



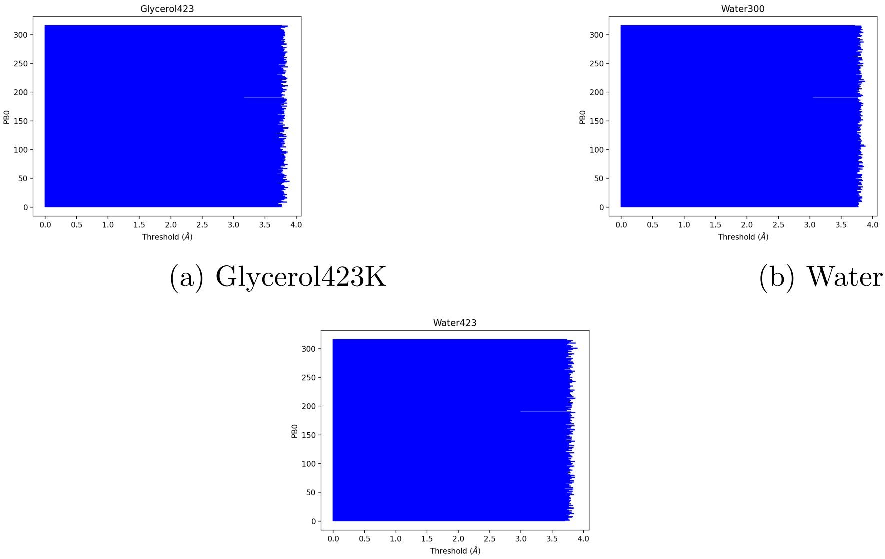
(a) β_0

(b) β_1

Figure B.8: Figure shows the evolution of zero(left) and one(right) dimensional hole (loops) as a function of threshold together with the corresponding total number of features as a function of frame number for CalB in water at 423K

B.3.2 Persistence Barcode and Diagram

In the following we show for each state 0-2 dimensional persistence barcode and persistence diagram.

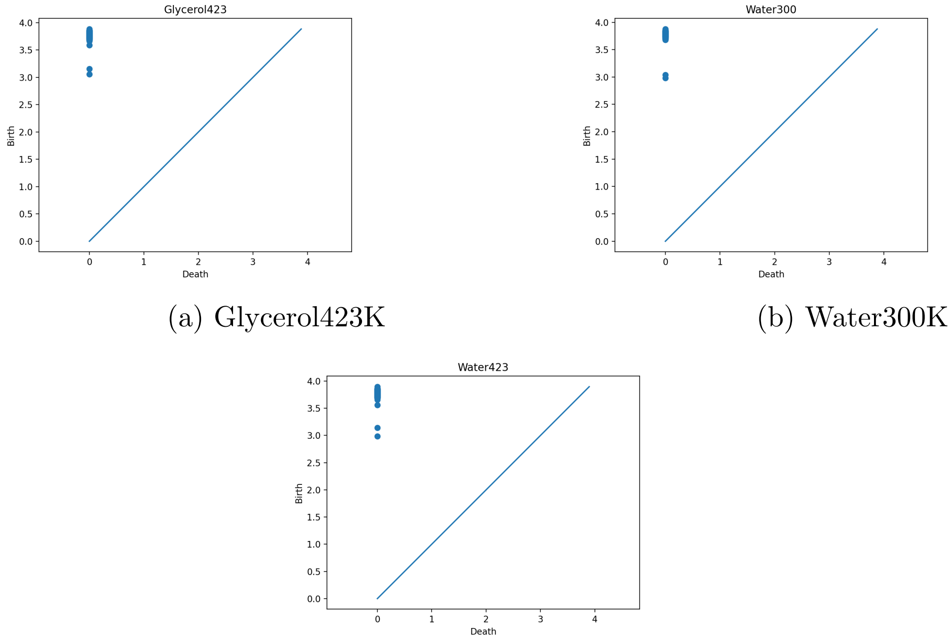


(a) Glycerol423K

(b) Water300K

(c) Water423K

Figure B.9: The 0-dimensional Persistent Barcode for protein in water at 300K , Glycerol at 423K and Water at 423K



(a) Glycerol423K

(b) Water300K

(c) Water423K

Figure B.10: The 0-dimensional Persistent Diagram for protein in glycerol at 423K, water at 300K and Water at 423K

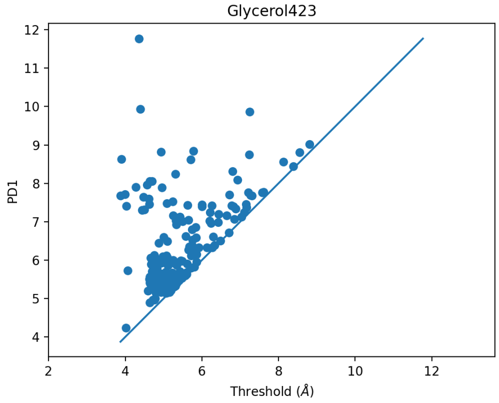


Figure B.11: (a)Persistent Diagram of CalB after 50ns in glycerol

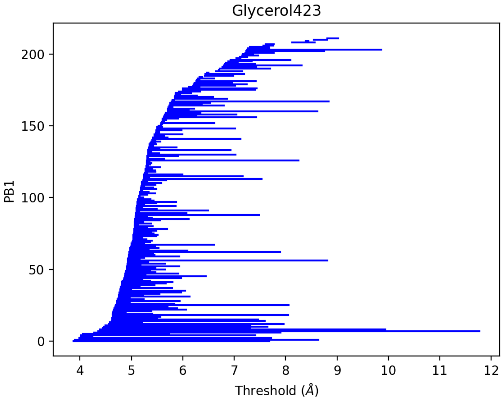


Figure B.12: (a)Persistent Diagram of CalB after 50ns in glycerol

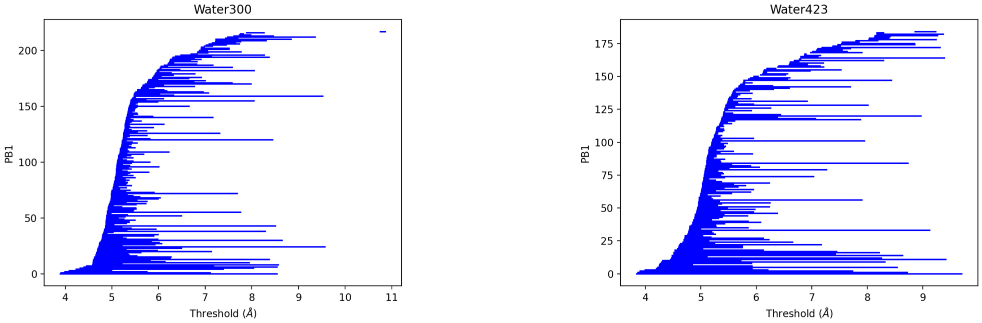


Figure B.13: (a)Persistent Barcode of CalB after 50ns in water at 300K and 423K.

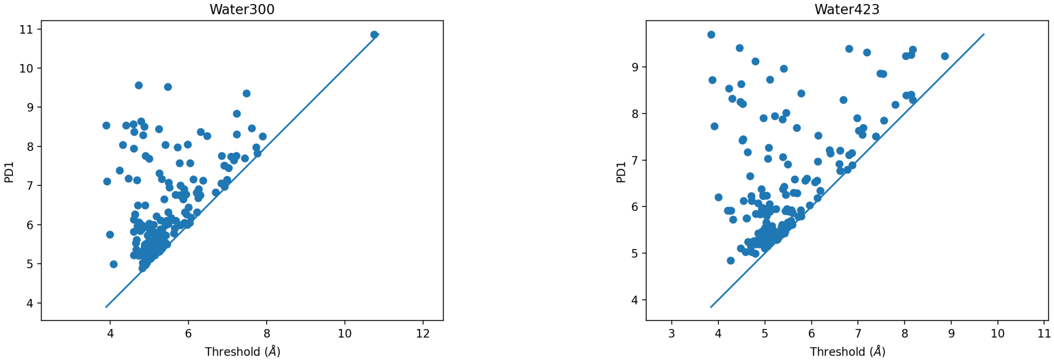
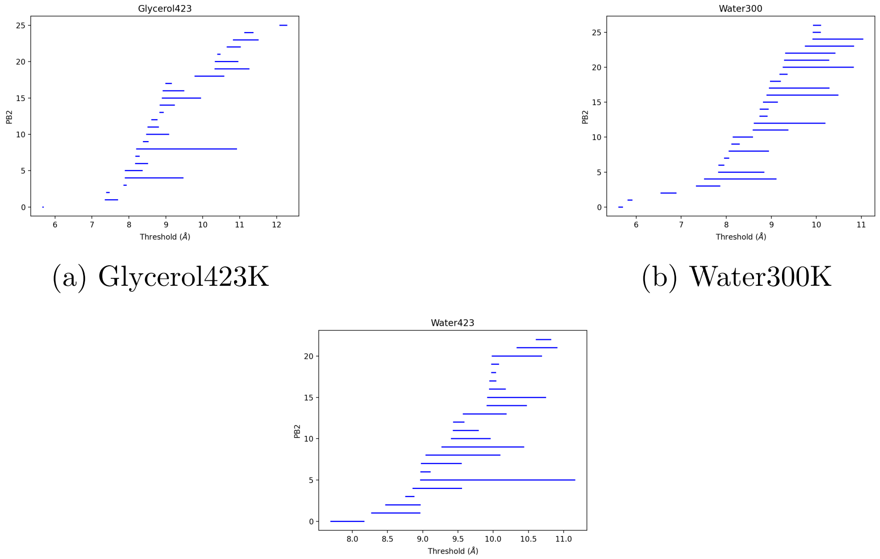


Figure B.14: (a)Persistent Diagram of CalB after 50ns in different temperatures in water

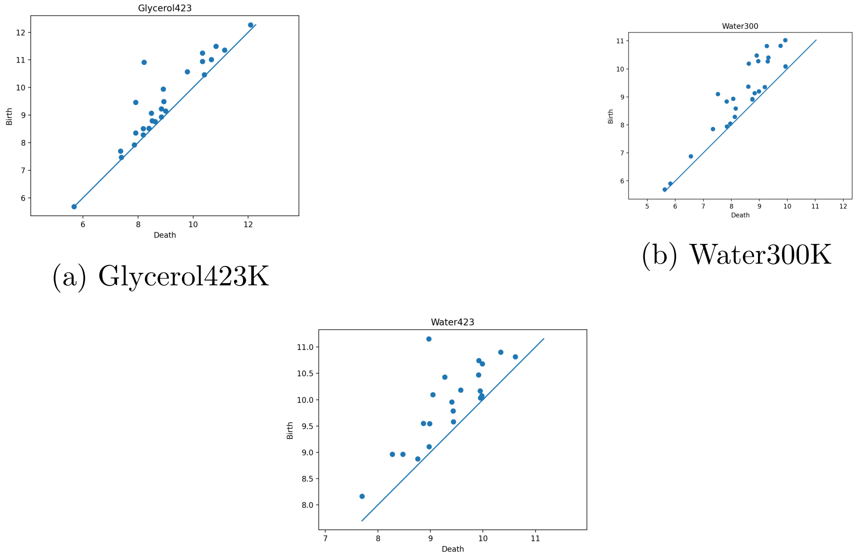


(a) Glycerol423K

(b) Water300K

(c) Water423K

Figure B.15: The 2-dimensional Persistent Barcode for protein in glycerol at 423K, water at 300K, and Water at 423K



(a) Glycerol423K

(b) Water300K

(c) Water423K

Figure B.16: The 2-dimensional Persistent Diagram for protein in water at 300K , Glycerol at 423K and Water at 423K

Appendix C

TDA Analysis Source Code

C.1 The source code used to calculate distance matrix, plot distance matrix and distribution

locations.py

```
import numpy as np
from Bio import PDB
from Bio import *
from Bio.PDB.PDBParser import PDBParser
from Bio.PDB.Polypeptide import PPBuilder
import dionysus
import matplotlib.pyplot as plt




def distance(p_i, p_j):
    D = len(p_i)
    dis = 0
    for d in range(D):
        dis += (p_i[d] - p_j[d]) ** 2

    return(dis ** (1/2))


########################################################################


parser = PDB.PDBParser()
struct = parser.get_structure('1tca','proteinwater423.pdb')

data = []

```

```python
subnode_number = []

for model in struct:
    for chain in model:
        for residue in chain:
            c = 0

            k = residue.get_id()
            for atom in residue:

                if atom.get_name() == 'CA' :
                    c += 1

                    X,Y,Z = atom.get_coord()
                    data.append([X,Y,Z])
                    subnode_number.append(c)


N = len(subnode_number)

data = np.array(data)


nodes = []

for i in range(N):
    nodes.append(data[sum(subnode_number[:i]) : sum(subnode_number[:i])+subnode_nu

nodes = np.array(nodes)


distance_matrix = np.zeros((N,N))

for i in range(N):
    for j in range(i+1, N):

        d_ij = []

        for p_i in nodes[i]:
            for p_j in nodes[j]:

                dist = distance(p_i, p_j)
                d_ij.append(dist)

        #print(min(d_ij))

        distance_matrix[i][j] = distance_matrix[j][i] = min(d_ij)


np.save('locations', data)


np.save('distance_matrix', distance_matrix)

print('distance_matrix saved')

```

Topology.py

```
import numpy as np
import networkx as nx
from Bio import PDB
from Bio import *
from Bio.PDB.PDBParser import PDBParser
from Bio.PDB.Polypeptide import PPBuilder
import dionysus
import matplotlib.pyplot as plt




def distance(p_i, p_j):
    D = len(p_i)
    dis = 0
    for d in range(D):
        dis += (p_i[d] - p_j[d]) ** 2

    return(dis ** (1/2))

#########################################################################

```

```


distance_matrix = np.load('distance_matrix.npy')

flat = distance_matrix.flatten()

plt.hist(flat, bins=100)
plt.show()

N = len(distance_matrix)


plt.imshow(distance_matrix)
plt.colorbar()
plt.show()


edge_list = []

for i in range(N):
    for j in range(i+1, N):
        if distance_matrix[i][j] != 0 :
            edge_list.append(distance_matrix[i][j])
        else:
            edge_list.append(np.inf)

edge_list = np.array(edge_list)

unq = np.unique(edge_list)

print(unq[-2])

max_dim_sim = 3
the_max = 14
the_min = 0


filt = dionysus.fill_rips(edge_list, max_dim_sim, the_max)
ph = dionysus.homology_persistence(filt)
pds = dionysus.init_diagrams(ph, filt)
```

```



np.save('pds', pds)


loops = []

for pp in pds[1]:
    loops.append([pp.birth, pp.death])

np.savetxt('loops', loops)

print('pds saved')
```

C.2 The source code used to plot for each frame betti curves, persistent barcodes and persistent diagrams

topologyplot.py

```
import numpy as np
import networkx as nx
from Bio import PDB
from Bio import *
from Bio.PDB.PDBParser import PDBParser
from Bio.PDB.Polypeptide import PPBuilder
import dionysus
import matplotlib.pyplot as plt



pds = np.load('pds.npy',allow_pickle=True)

max_dim_sim = len(pds)


for d in range(max_dim_sim-1):
    dionysus.plot.plot_bars(pds[d])
    plt.savefig('PB'+str(d)+'.pdf')
    print(len(pds[d]))
    plt.title('PB'+str(d))
    plt.show()


for d in range(max_dim_sim-1):
    dionysus.plot.plot_diagram(pds[d])
    print(len(pds[d]))
    plt.title('PD'+str(d))
    plt.savefig('PD'+str(d)+'.pdf')
    plt.show()


the_max = 14
the_min = 0


Betti_curve_d = []
birth_curve_d = []
death_curve_d = []
pdf_PB_d = []
PE_d = []
PI_d = []
mean_bars_d = []

STEP = 50
DELTA = float((the_max-the_min)/STEP)

step_Bettis = 1000
step = 100

delta = float((the_max-the_min)/step)

delta_Bettis = float((the_max-the_min)/step_Bettis)

thrs_Bettis = np.arange(the_min, the_max+delta_Bettis, delta_Bettis)

for d in range(max_dim_sim-1):

    bars = []
    baRs = []
    Betti_curve = np.zeros(step_Bettis+1)
    birth_curve = np.zeros(step+1)
    death_curve = np.zeros(step+1)
    PI = np.zeros((STEP+1,STEP+1))

    births = []
    deaths = []

    for p in pds[d]:

        births.append(p.birth)
        deaths.append(min(p.death,the_max))

        if p.death != np.inf :
            k_min = int((p.birth-the_min) / delta_Bettis)
            k_max = int((p.death-the_min) / delta_Bettis)

            k_birth = int((p.birth-the_min) / delta)
            k_death = int((p.death-the_min) / delta)

            K_MIN = int((p.birth-the_min) / DELTA)
            K_MAX = int((p.death-the_min) / DELTA)
            PI[K_MIN][K_MAX] += 1


            birth_curve[k_birth] += 1
            death_curve[k_death] += 1

            for k in range(k_min, k_max):
                Betti_curve[k] += 1
            bars.append(p.death - p.birth)
            baRs.append(p.death - p.birth)

        else:
            k_min = int((p.birth-the_min) / delta_Bettis)
            k_max = int((the_max-the_min) / delta_Bettis)

            k_birth = int((p.birth-the_min) / delta)
            k_death = int((the_max-the_min) / delta)

            K_MIN = int((p.birth-the_min) / DELTA)
            K_MAX = int((the_max-the_min) / DELTA)
            PI[K_MIN][K_MAX] += 1

            birth_curve[k_birth] += 1
            death_curve[k_death] += 1

            for k in range(k_min, k_max):
                Betti_curve[k] += 1
            bars.append(the_max - p.birth)

    pdf_PB = np.zeros(step+1)
    PE = 0
    Bar = np.sum(baRs)

    mean_bars = np.max(baRs)
    mean_bars_d.append(mean_bars)

    for bar in baRs:
        kl = int((bar-the_min) / delta)
        pdf_PB[kl] += 1
        probab = float(bar / Bar)
        PE -= (probab * np.log(probab))

    #PE = float(PE / Bar)

    Betti_curve_d.append(Betti_curve)
    birth_curve_d.append(birth_curve)
    death_curve_d.append(death_curve)
    pdf_PB_d.append(pdf_PB)
    PE_d.append(PE)
    PI_d.append(PI)


np.save('Betti_curve_d', Betti_curve_d)

for d in range(max_dim_sim-1):
    plt.plot(thrs_Bettis, Betti_curve_d[d], marker='.')
    plt.savefig('betti'+str(d))
    plt.show()
    #plt.plot(thrs_Bettis, birth_curve_d[d], marker='.')
    #plt.show()
    #plt.plot(thrs_Bettis, death_curve_d[d], marker='.')
    #plt.show()

#for d in range(max_dim_sim-1):
 #   dionysus.plot.plot_diagram(pds[d])
  #  print(len(pds[d]))
   # plt.title('PD'+str(d))
    #plt.savefig('PD'+str(d)+'.pdf')
    #plt.show()
```

C.3 The source code used to find time averaged betti curves

timeaveragedbetti.py

```
import numpy as np
import pandas
import scipy.spatial
import mdtraj
import dionysus
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from mpl_toolkits.mplot3d import Axes3D


def dis(p1,p2):
    D = len(p1)
    distance = 0
    for d in range(D):
        distance += (p1[d]-p2[d])**2
    return(distance**0.5)


#get_ipython().run_line_magic('matplotlib', 'inline')

```

```
calb_residues = 317
coordinates = np.zeros((calb_residues,3))
number_of_frames = 5001
step_frames = 100


time_step = int(number_of_frames/step_frames)


the_min = 0
the_max = 1.4
step = 1000
delta = float((the_max-the_min)/step)
thrs = np.arange(the_min, the_max+delta, delta)


am_frame = []
diagrams_0_frame = []
diagrams_1_frame = []
diagrams_2_frame = []


store0 = pandas.HDFStore('test0.h5d')
store1 = pandas.HDFStore('test1.h5d')
store2 = pandas.HDFStore('test2.h5d')


for read_frame in range(0, number_of_frames, step_frames) :
    #
    # LOAD THE FRAME
    #
    trajectory = mdtraj.load_xtc("md_0_1.xtc",
                                 top="1tcagly423.gro",
                                 frame=read_frame)
    #
    # CREATE A (317,3) MATRIX OF THE COORDINATES OF THE CHOSEN ATOM
    #
    topology = trajectory.topology
    alpha_carbon_indexes = topology.select("name CA")



    for index in range(calb_residues) :
        coordinates[index] = trajectory.xyz[0,alpha_carbon_indexes[index]]


    #print(coordinates)
    #plt.scatter(coordinates[:,0],coordinates[:,1],marker='.')
    #plt.show()
    #fig = plt.figure()
    #ax = fig.add_subplot(projection='3d')
    #ax.plot3D(coordinates[:,0], coordinates[:,1], coordinates[:,2], marker='.',
    #plt.show()



    #
    # DO TDA ON THAT POINT CLOUD
    #
    #print(coordinates[:,0])

    xyz = []

    am = np.zeros((317,317))
    for i in range(317):
        xyz.append(coordinates[i])
        for j in range(i+1,317):
            am[i][j] = am[j][i] = dis(coordinates[i],coordinates[j])


    am_frame.append(am)
    #pdist = scipy.spatial.distance.pdist(coordinates)
    #print(coordinates.shape)
    #print(type(coordinates))
    #print(pdist.shape)

    xyz = np.array(xyz)

    np.save('am_'+str(read_frame), am)


    filtration = dionysus.fill_rips(xyz, 3, the_max)


    #filtration = dionysus.fill_rips(coordinates, 2, the_max)
    persistence = dionysus.homology_persistence(filtration)
    diagrams = dionysus.init_diagrams(persistence,filtration)
    #
    # ADD EACH FEATURE TO THE DENSITY MATRIX
    #
    #for i,pd in enumerate(diagrams[1]) :
         #for j,cutoff in enumerate(cutoff_distance) :
                #if (pd.birth < cutoff) and (pd.death > cutoff) :
                    #density[j] += 1


    births0 = []
    deaths0 = []
    for pp in diagrams[0] :
        births0.append(pp.birth)
        deaths0.append(pp.death)


    df0 = pandas.DataFrame({"birth":births0, "death":deaths0})
    key = f"frame{read_frame:04d}"
    store0.put(key,df0)




    births1 = []
    deaths1 = []
    for pp in diagrams[1] :
        births1.append(pp.birth)
        deaths1.append(pp.death)

    df1 = pandas.DataFrame({"birth":births1, "death":deaths1})
    key = f"frame{read_frame:04d}"
    store1.put(key,df1)

    births2 = []
    deaths2 = []
    for pp in diagrams[2] :
        births2.append(pp.birth)
        deaths2.append(pp.death)

    df2 = pandas.DataFrame({"birth":births2, "death":deaths2})
    key = f"frame{read_frame:04d}"
    store2.put(key,df2)


    diagrams_0_frame.append(diagrams[0])
    diagrams_1_frame.append(diagrams[1])
    diagrams_2_frame.append(diagrams[2])



    np.save('pd_'+str(read_frame), diagrams)




store0.close()
store1.close()
store2.close()



am_frame = np.array(am_frame)
am_mean = np.mean(am_frame, axis=0)
am_std = np.std(am_frame, axis=0)
np.save('am_mean', am_mean)
np.save('am_std', am_std)


#np.save('diagrams_0_frame', diagrams_0_frame)
#np.save('diagrams_1_frame', diagrams_1_frame)
#np.save('diagrams_2_frame', diagrams_2_frame)
```

```
####################


pdf_am_frame = []
min_am = 0
max_am = 10
step_am = 100
delta_am = float((max_am-min_am)/step_am)
thrs_am = np.arange(min_am, max_am+delta_am, delta_am)

betti_curve_0_frame = []
betti_curve_1_frame = []
betti_curve_2_frame = []


entropy_0_frame = []
entropy_1_frame = []
entropy_2_frame = []
pbar_frame = []
for time in range(time_step):
    pdf_am = np.zeros(step_am+1)
    for val in am_frame[time].flatten():
        if min_am < val <= max_am :
            k = int((val-min_am)/delta_am)
            pdf_am[k] += 1
    pdf_am_frame.append(pdf_am)
    np.save('pdf_am_'+str(time), pdf_am)


    betti_curve_0 = np.zeros(step+1)

    lifetimes0 = []


    for pp in diagrams_0_frame[time] :

        if pp.death != np.inf :
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((pp.death-the_min)/delta)

            lifetimes0.append(pp.death-pp.birth)


        else:
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((the_max-the_min)/delta)


            lifetimes0.append(the_max-pp.birth)


        betti_curve_0[k_min:k_max] += 1


    L = sum(lifetimes0)


    entropy_0 = 0


    for l in lifetimes0:


        entropy_0 -= (l/L) * np.log(l/L)


    entropy_0_frame.append(entropy_0)
    betti_curve_0_frame.append(betti_curve_0)




    betti_curve_1 = np.zeros(step+1)
    lifetimes1 = []


    for pp in diagrams_1_frame[time] :
        if pp.death != np.inf :
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((pp.death-the_min)/delta)


            lifetimes1.append(pp.death-pp.birth)


        else:
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((the_max-the_min)/delta)


            lifetimes1.append(the_max-pp.birth)


        betti_curve_1[k_min:k_max] += 1


    L = sum(lifetimes1)



    entropy_1 = 0

    for l in lifetimes1:
        entropy_1 -= (l/L) * np.log(l/L)


    #pbar = float(max(lifetimes1) / np.mean(lifetimes1.pop(np.argmax(lifetimes1)
    #pbar = float(max(lifetimes1) / np.mean(lifetimes1))

    sort = np.sort(lifetimes1)
    pbar = sort[-1]/np.mean(sort[:-2])
    pbar_frame.append(pbar)

    entropy_1_frame.append(entropy_1)

    betti_curve_1_frame.append(betti_curve_1)




    betti_curve_2 = np.zeros(step+1)
    lifetimes2 = []


    for pp in diagrams_2_frame[time] :


        if pp.death != np.inf :
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((pp.death-the_min)/delta)


            lifetimes2.append(pp.death-pp.birth)


        else:
            k_min = int((pp.birth-the_min)/delta)
            k_max = int((the_max-the_min)/delta)


            lifetimes2.append(the_max-pp.birth)


        betti_curve_2[k_min:k_max] += 1


    L = sum(lifetimes2)



    entropy_2 = 0

    for l in lifetimes2:
        entropy_2 -= (l/L) * np.log(l/L)



    sort = np.sort(lifetimes2)
    pbar = sort[-1]/np.mean(sort[:-2])
    pbar_frame.append(pbar)

    entropy_2_frame.append(entropy_2)



    betti_curve_2_frame.append(betti_curve_2)




pdf_am_frame = np.array(pdf_am_frame)
pdf_am_mean = np.mean(pdf_am_frame, axis=0)
pdf_am_std = np.std(pdf_am_frame, axis=0)

np.save('pdf_am_mean', pdf_am_mean)
np.save('pdf_am_std', pdf_am_std)

betti_curve_0_frame = np.array(betti_curve_0_frame)
betti_curve_0_mean = np.mean(betti_curve_0_frame, axis=0)
betti_curve_0_std = np.std(betti_curve_0_frame, axis=0)
betti_curve_1_frame = np.array(betti_curve_1_frame)
betti_curve_1_mean = np.mean(betti_curve_1_frame, axis=0)
betti_curve_1_std = np.std(betti_curve_1_frame, axis=0)
betti_curve_2_frame = np.array(betti_curve_2_frame)
betti_curve_2_mean = np.mean(betti_curve_2_frame, axis=0)
betti_curve_2_std = np.std(betti_curve_2_frame, axis=0)


bottleneck_distance_matrix_0 = np.zeros((time_step,time_step))
bottleneck_distance_matrix_1 = np.zeros((time_step,time_step))
bottleneck_distance_matrix_2 = np.zeros((time_step,time_step))


for i in range(time_step):
    for j in range(i+1, time_step):


        bottleneck = dionysus.bottleneck_distance(diagrams_0_frame[i], diagrams_0_
        bottleneck_distance_matrix_0[i][j] = bottleneck_distance_matrix_0[j][i] =

        bottleneck = dionysus.bottleneck_distance(diagrams_1_frame[i], diagrams_1_
        bottleneck_distance_matrix_1[i][j] = bottleneck_distance_matrix_1[j][i] =

        bottleneck = dionysus.bottleneck_distance(diagrams_2_frame[i], diagrams_2_
        bottleneck_distance_matrix_2[i][j] = bottleneck_distance_matrix_2[j][i] =



print(time_step)
np.save('Betti_curve_mean_0', betti_curve_0_mean)
np.save('Betti_curve_mean_1', betti_curve_1_mean)
np.save('Betti_curve_mean_2', betti_curve_2_mean)
np.save('Betti_curve_std_0', betti_curve_0_std)
np.save('Betti_curve_std_1', betti_curve_1_std)
np.save('Betti_curve_std_2', betti_curve_2_std)
np.save('bottleneck_distance_matrix_0', bottleneck_distance_matrix_0)
np.save('bottleneck_distance_matrix_1', bottleneck_distance_matrix_1)
np.save('bottleneck_distance_matrix_2', bottleneck_distance_matrix_2)
print(bottleneck_distance_matrix_0)
print(bottleneck_distance_matrix_1)
print(bottleneck_distance_matrix_2)


### plots :

plt.imshow(am_mean)
plt.title('am_mean')
cbar = plt.colorbar()
cbar.set_label('Weights in $(nm)$', rotation=90)
plt.savefig('am_mean')
plt.show()

flat = am_mean.flatten()
plt.hist(flat, bins=100)
plt.savefig('Averaged PDF')
plt.xlabel('Threshold(nm)')
plt.show()

plt.imshow(am_std)
plt.title('am_std')
cbar = plt.colorbar()
cbar.set_label('Weights in $(nm)$', rotation=90)
plt.savefig('am_std')
plt.show()

flat = am_std.flatten()
plt.hist(flat, bins=100)
plt.savefig('pdf_am_std')
plt.show()


plt.errorbar(thrs_am[:len(pdf_am_std)], pdf_am_mean, pdf_am_std, marker='.')
plt.savefig('pdf_am_mean')
plt.savefig('Mean Value of PDF in Time')
plt.xlabel('Threshold(nm)')
plt.show()

plt.plot(pbar_frame, marker='.')
plt.savefig('pbar')
plt.title('PBar')
plt.xlabel("Time(ns)")
plt.ylabel("Lifetime")
plt.grid(True)
plt.show()


d = 0

plt.errorbar(thrs[:len(betti_curve_0_mean)], betti_curve_0_mean, betti_curve_0_std
plt.xlabel("Threshold(nm)")
plt.ylabel("Betti0")
plt.show()



d = 1
plt.errorbar(thrs[:len(betti_curve_1_mean)], betti_curve_1_mean, betti_curve_1_std
plt.xlabel("Threshold(nm)")
plt.ylabel("Betti1")
plt.show()


d = 2
plt.errorbar(thrs[:len(betti_curve_2_mean)], betti_curve_2_mean, betti_curve_2_std
plt.xlabel("Threshold(nm)")
plt.ylabel("Betti2")
plt.show()



###

d = 0
plt.plot(range(time_step), entropy_0_frame, marker='.')
plt.show()



d = 1
plt.plot(range(time_step), entropy_1_frame, marker='.')
plt.show()

d = 2
plt.plot(range(time_step), entropy_2_frame, marker='.')
plt.show()

###


d = 0
plt.imshow(bottleneck_distance_matrix_0)
plt.colorbar()
plt.show()



d = 1
```

```
plt.imshow(bottleneck_distance_matrix_1)
plt.colorbar()
plt.show()

d = 2
plt.imshow(bottleneck_distance_matrix_2)
plt.colorbar()
plt.show()



```

Bibliography

[1] Hydrolysis of Esters, jan 31 2022. [Online; accessed 2022-12-02].

[2] A Walk in the Forest. A Primer for Protein Structure. `https://walkintheforest.com/Content/Posts/A+Primer+for+Protein+Structure`, 2020.

[3] Jorge Almarza, Luis Rincon, Ali Bahsas, and Francisco Brito. Molecular mechanism for the denaturation of proteins by urea. *Biochemistry*, 48(32):7608–7613, 2009.

[4] Erik J Amézquita, Michelle Y Quigley, Tim Ophelders, Elizabeth Munch, and Daniel H Chitwood. The shape of things to come: Topological data analysis and biology, from molecules to organisms. *Developmental Dynamics*, 249(7):816–833, 2020.

[5] Gil Amitai, Arye Shemesh, Einat Sitbon, Maxim Shklar, Dvir Netanely, Ilya Venger, and Shmuel Pietrokovski. Network analysis of protein structures identifies functional residues. *Journal of Molecular Biology*, 344(4):1135–1146, 2004. ISSN 00222836. doi: 10.1016/j.jmb.2004.10.055. URL `https://linkinghub.elsevier.com/retrieve/pii/S0022283604013592`.

[6] Nieves Atienza, Rocio Gonzalez-Diaz, and Matteo Rucco. Persistent entropy for separating topological features from noise in vietoris-rips complexes. *Journal of Intelligent Information Systems*, 52(3):637–655, 2019.

[7] BioNinja. Protein structure. `https://ib.bioninja.com.au/higher-level/topic-7-nucleic-acids/73-translation/protein-structure.html`, 2021.

[8] Fredrik Björkling, Sven Erik Godtfredsen, and Ole Kirk. The future impact of industrial lipases. *Trends in Biotechnology*, 9(1):360–363, 1991.

[9] Uwe Bornscheuer, Oscar-Werner Reif, Ralf Lausch, Ruth Freitag, Thomas Scheper, Fragiskos N Kolisis, and Uldrich Menge. Lipase of pseudomonas cepacia for biotechnological purposes: purification, crystallization and characterization. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1201(1):55–60, 1994.

[10] Zixuan Cang, Lin Mu, Kedi Wu, Kristopher Opron, Kelin Xia, and Guo-Wei Wei. A topological approach for protein classification. *Computational and Mathematical Biophysics*, 3(1), 2015. ISSN 2544–7297. doi: 10.1515/mlbmb-2015-0009. URL `https://www.degruyter.com/document/doi/10.1515/mlbmb-2015-0009/html`.

[11] Zixuan Cang, Elizabeth Munch, and Guo-Wei Wei. Evolutionary homology on coupled dynamical systems with applications to protein flexibility analysis. *Journal of Applied and Computational Topology*, 4(4):481–507, 2020. ISSN 2367-1726, 2367-1734. doi: 10.1007/s41468-020-00057-9. URL `https://link.springer.com/10.1007/s41468-020-00057-9`.

[12] Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12, 2008.

[13] Yiwen R. Chen, Itamar Harel, Param Priya Singh, Inbal Ziv, Eitan Moses, Uri Goshtchevsky, Ben E. Machado, Anne Brunet, and Daniel F. Jarosz. Tissue-specific landscape of protein aggregation and quality control in an aging vertebrate. *bioRxiv*, 2022. doi: 10.1101/2022.02.26.482120.

[14] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.

[15] C Dlugy and A Wolfson. Lipase catalyse glycerolysis for kinetic resolution of racemates. *Bioprocess and biosystems engineering*, 30(5):327–330, 2007.

[16] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022.

[17] Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[18] EMBL-EBI. Biomacromolecular structures. `https://www.ebi.ac.uk/training/online/courses/biomacromolecular-structures/proteins/levels-of-protein-structure-primary/levels-of-protein-structure-secondary/`.

[19] Ernesto Estrada. *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[20] Mark B Frampton, Jacqueline P Séguin, Drew Marquardt, Thad A Harroun, and Paul M Zelisko. Synthesis of polyesters containing disiloxane subunits: Structural characterization, kinetics, and an examination of the thermal tolerance of novozym-435. *Journal of Molecular Catalysis B: Enzymatic*, 85:149–155, 2013.

[21] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

[22] Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32(1): 1–17, 2015.

[23] Mohamad Reza Ganjalikhany, Bijan Ranjbar, Amir Hossein Taghavi, and Tahereh Tohidi Moghadam. Functional motions of candida antarctica lipase b: a survey through open-close conformations. *PloS one*, 7(7):e40327, 2012.

[24] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[25] Allen Hatcher. *Algebraic topology.* FIX THIS, 2005.

[26] Jing Huang, Sarah Rauscher, Grzegorz Nawrocki, Ting Ran, Michael Feig, Bert L De Groot, Helmut Grubmüller, and Alexander D MacKerell. Charmm36m: an improved force field for folded and intrinsically disordered proteins. *Nature methods*, 14(1):71–73, 2017.

[27] Takashi Ichinomiya, Ippei Obayashi, and Yasuaki Hiraoka. Protein-Folding Analysis Using Features Obtained by Persistent Homology. *Biophysical Journal*, 118 (12):2926–2937, 2020. ISSN 00063495. doi: 10.1016/j.bpj.2020.04.032. URL `https://linkinghub.elsevier.com/retrieve/pii/S0006349520303763`.

[28] Matthew Kahle. Topology of random clique complexes. *Discrete mathematics*, 309(6):1658–1671, 2009.

[29] Wael I Karain and Nael I Qaraeen. Weighted protein residue networks based on joint recurrences between residues. *BMC bioinformatics*, 16(1):1–11, 2015.

[30] Mehran Kardar. *Statistical physics of particles.* Cambridge University Press, 2007.

[31] Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

[32] Violeta Kovacev-Nikolic, Peter Bubenik, Dragan Nikolić, and Giseon Heo. Using persistent homology and dynamical distances to analyze protein binding. *Statistical applications in genetics and molecular biology*, 15(1):19–38, 2016.

[33] J Kumaresan, T Kothai, and BS Lakshmi. In silico approaches towards understanding calb using molecular dynamics simulation and docking. *Molecular Simulation*, 37(12):1053–1061, 2011.

[34] Erik Lindahl, Berk Hess, and David Van Der Spoel. Gromacs 3.0: a package for molecular simulation and trajectory analysis. *Molecular modeling annual*, 7(8): 306–317, 2001.

[35] Hosein Masoomy, Behrouz Askari, Samin Tajik, Abbas K Rizi, and G Reza Jafari. Topological analysis of interaction patterns in cancer-specific gene regulatory network: persistent homology approach. *Scientific Reports*, 11(1):1–11, 2021.

[36] William Mendenhall, Robert J Beaver, and Barbara M Beaver. *Introduction to probability and statistics*. Cengage, 2020.

[37] Nora Molkenthin, Steffen Mühle, Antonia SJS Mey, and Marc Timme. Self-organized emergence of folded protein-like network structures from geometric constraints. *Plos one*, 15(2):e0229230, 2020.

[38] Hassan Monhemi, Mohammad Reza Housaindokht, Ali Akbar Moosavi-Movahedi, and Mohammad Reza Bozorgmehr. How a protein can remain stable in a solvent with high content of urea: insights from molecular dynamics simulation of candida antarctica lipase b in urea: choline chloride deep eutectic solvent. *Physical Chemistry Chemical Physics*, 16(28):14882–14893, 2014.

[39] Elizabeth Munch. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61, 2017.

[40] James R Munkres. *Topology*. Prentice Hall, 2000.

[41] Mikio Nakahara. *Geometry, topology and physics*. CRC press, 2003.

[42] Mark Ed Newman, Albert-László Ed Barabási, and Duncan J Watts. *The structure and dynamics of networks*. Princeton university press, 2006.

[43] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL `http://gudhi.gforge.inria.fr/doc/latest/`.

[44] Python Software Foundation. Python package dionysus 2.0.8 - PyPI. URL `https://https://pypi.org/project/dionysus/`.

[45] Gulam Rabbani, Ejaz Ahmad, Mohsin Vahid Khan, Mohd Tashfeen Ashraf, Rajiv Bhat, and Rizwan Hasan Khan. Impact of structural stability of cold adapted candida antarctica lipase b (calb): in relation to ph, chemical and thermal denaturation. *Rsc Advances*, 5(26):20115–20131, 2015.

[46] Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L Ringach. Topological analysis of population activity in visual cortex. *Journal of vision*, 8(8):11–11, 2008.

[47] Ann E Sizemore, Jennifer E Phillips-Cremins, Robert Ghrist, and Danielle S Bassett. The importance of the whole: topological data analysis for the network neuroscientist. *Network Neuroscience*, 3(3):656–673, 2019.

[48] Benjamin Stauch, Stuart J Fisher, and Michele Cianci. Open and closed states of candida antarctica lipase b: protonation and the mechanism of interfacial activation1. *Journal of lipid research*, 56(12):2348–2358, 2015.

[49] Yueru Sun, Shuhui Yin, Yitao Feng, Jie Li, Jiahai Zhou, Changdong Liu, Guang Zhu, and Zhihong Guo. Molecular basis of the general base catalysis of an α/β-hydrolase catalytic triad. *Journal of Biological Chemistry*, 289(22):15867–15879, 2014.

[50] Stefan Thurner, Rudolf Hanel, and Peter Klimek. *Introduction to the theory of complex systems*. Oxford University Press, 2018.

[51] Chad M Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PloS one*, 10(5):e0126383, 2015.

[52] Peter Trodler and Jürgen Pleiss. Modeling structure and flexibility of candida antarctica lipase b in organic solvents. *BMC structural biology*, 8(1):1–10, 2008.

[53] J. Uppenberg and T.A. Jones. The sequence, crystal structure determination and refinement of two crystal forms of lipase b from candida antarctica. `https://www.rcsb.org/structure/1TCA`, 1994.

[54] Menglun Wang, Zixuan Cang, and Guo-Wei Wei. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nature Machine Intelligence*, 2(2):116–123, 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0149-6. URL `http://www.nature.com/articles/s42256-020-0149-6`.

[55] Shmuel Weinberger. What is... persistent homology. *Notices of the AMS*, 58(1): 36–39, 2011.

[56] Adi Wolfson, Eve Yefet, Tomer Alon, Christina Dlugy, and Dorith Tavor. Glycerolysis of esters with candida antarctica lipase b in glycerol. *Journal of Advanced Chemical Engineering*, 1, 2011.

[57] Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

[58] Ningyan Zhang, Wen-Chen Suen, William Windsor, Li Xiao, Vincent Madison, and Aleksey Zaks. Improving tolerance of candida antarctica lipase b towards irreversible thermal inactivation through directed evolution. *Protein engineering*, 16(8):599–605, 2003.

[59] Xiaojin Zhu. Persistent homology: An introduction and a new text representation for natural language processing. In *IJCAI*, pages 1953–1959, 2013.

[60] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.