NEW ALGORITHMS FOR PROTEIN STRUCTURE ANALYSIS: FROM NONPARAMETRIC DENSITY ESTIMATION TO CHARACTERIZATION OF MOLECULAR VOLUME SPATIAL DISTRIBUTIONS

by

Jenny Farmer

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computational Biology

Charlotte

2018

Approved by:

_____
Dr. Donald Jacobs

_____
Dr. Irina Nesmelova

_____
Dr. Jun-tao Guo

_____
Dr. Cynthia Gibas

_____
Dr. Srinivas Akella

ABSTRACT

JENNY FARMER. New Algorithms for Protein Structure Analysis: From Nonparametric Density Estimation to Characterization of Molecular Volume Spatial Distributions. (Under the direction of DR. DONALD JACOBS).

The Flexibility and Stability Test (FAST) is a C++ class library designed in the early 2000's to execute free energy decomposition and reconstitution operations applied to protein structures. The library has been substantially expanded to include structural bioinformatics tools useful in the analysis of protein dynamics. Motivated to advance FAST by improving modeling aspects on atomic packing and solvation, two new algorithms have been developed and implemented, enabling high throughput nonparametric probability density estimation and spatial characterization of cavity volume in proteins. In two separate studies involving molecular dynamics trajectories, novel methods were developed for the analysis of statistical significance in the dynamics of beta-lactamase mutants. Additionally, the core methodologies developed through these studies have been validated as critical components of the FAST library, which aims to advance the field of computational biology and structural bioinformatics as a next generation simulation software for protein and drug design.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NMR	Nuclear Magnetic Resonance
PDB	Protein Data Bank
MD	Molecular Dynamics
mDCM	Minimal Distance Constraint Model
FAST	Flexibility And Stability Test
PVA	Protein Void Analyzer
HK	Hoshen-Kopelmann
SASA	Solvent Accessible Surface Area
PDF	Probability Density Function
KDE	Kernel Density Estimation
MEM	Maximum Entropy Method
CDF	Cumulative Density Function
SURD	Sampled Uniform Random Data
CRAN	Comprehensive R Archive Network
FSS	Finite Size Scaling
KL	Kullback-Leibler
KS	Kolmogorov-Smirnov
RSM	Reduced Second Moment
RMSF	Root Mean Square Fluctuations
JS	Jensen-Shannon
PCA	Principal Component Analysis
dPCA	displacement Principal Component Analysis

RMSF	Root Mean Square Fluctuation
OSP	Occluded Surface Packing
BMPG	Bio-Molecular Physics Group

CHAPTER 1: BACKGROUND

1.1 Introduction: History of Protein Dynamics

Named in 1938 after the Greek work *protos*, meaning *first*, proteins have long been recognized as essential to all of known life [1]. In the early 1900's, a few pioneering scientists proposed that the primary structure of proteins is comprised of linearly connected amino acids, and this theory became generally accepted over the following decades. The way in which amino acid sequences fold themselves, however, has proven to be a more elusive mystery. Over a century ago, even before the importance of protein structure was imagined, it was first observed that proteins exhibit the tendency to coagulate when heated, an early demonstration of the response of protein shape according to environment [2]. The fascination with protein folding began with these important discoveries and has continued to this day. In the late 1950's, the structure of myoglobin was published, marking the first complete look at the three-dimensional shape of a folded protein, and the beginnings of an explosion of new information in structural biology [3, 4]. In the years since, crystallography and Nuclear Magnetic Resonance (NMR) technology have provided invaluable insight into the complexity and variety of folding patterns found in all forms of life. To date, over 100,000 structures have been resolved and made available in the Protein Data Bank (PDB), and existing entries are being continuously replaced by structures with improved resolution [5]. In addition to the PDB, other collaborations have established repositories and classification systems for common folds and motifs to organize proteins by shape, function, and evolution [6-8].

While there are many different types of proteins, such as membrane bound, fibril, and intrinsically disordered, globular proteins make up a large class of proteins. Globular proteins span a diverse collection of molecular environments when viewed across all living species on Earth. Despite the immense number of possible sequences and diverse environments, a ballpark number for known distinct globular protein folds recorded to date is only about 1500. Through these classification efforts, it has become clear that the three-dimensional shapes of proteins are not random, but rather intricately evolved systems fine-tuned to satisfy biological function and support life [4, 9-13].

Advances in structure resolution and protein structure classification continue and contribute to the success of the field of structural biology. However, beyond the technical difficulties with experimental methods [14, 15], there are inherent restrictions surrounding the approach of studying proteins as static structures. Proteins are highly dynamic and flexible, with motions ranging from local fluctuations to large-scale conformational transitions that are functionally critical. Differences between protein structures sometimes provide evidence for functionally relevant alternate confirmations, often called ‘hidden states’, but are difficult to sample using experimental methods. Hidden states largely remain elusive because these conformations have relatively high energy and are thus rare and/or transient in nature. These states often may be excluded by the environmental conditions required to conduct the experiment. Recent advances in experimental methods are helping to address these issues [16-19].

1.2 Molecular Dynamics

In parallel with experimental advances, extensive computational methods have been developed to capture the motions found in proteins. The application of Molecular

Dynamics (MD) to proteins is generally considered to have begun in 1977, when the dynamics of bovine pancreatic trypsin were simulated by solving the equations of motion for each of the protein atoms [20]. Although MD simulations have become far more advanced in the years since, the premise remains essentially unchanged: given initial positions and velocities for each atom in the system, all subsequent positions and velocities are calculated at some time interval later, based on known intermolecular forces. Key features of protein stability and packing characteristics are determined by van der Waals interactions, electrostatic interactions, hydrogen bonds, hydrophobic interactions, and torsion angles [4].

Despite the simplicity of the model, the complexity and volume of calculations are overwhelming, even with modern computing power. In principle, an accurate force field would entail a quantum mechanical approach, solving Schrödinger's equation for the entire multi-particle system, at each timestep. Although modern MD algorithms can incorporate some degree of quantum mechanics in highly sensitive areas of the protein, the reality of computational demands forces simulations to remain heavily dependent upon less accurate Newtonian Mechanics for calculating motions. Many different and increasingly complicated force fields have been implemented, but all remain approximations, often crude ones, and are subject to errors and criticisms.

Even with such approximations, it is estimated that for a 100,000-atom system, nearly a billion calculations are required per time step [21]. The greater the elapsed time between position and velocity calculations, the more cumulative error is introduced, so each time step must remain very small, typically no more than a few femtoseconds [21, 22]. Recent MD simulations reported in the literature commonly reach a microsecond of

elapsed time for proteins [22, 23]. This translates to trillions of calculations, and years of simulation time on a single processor. Furthermore, a single MD trajectory represents only one sample out of an incalculable number of combinations. The impracticality of running these simulations has inspired MD developers to become creative in terms of parallel processing, simplifying models, and improved sampling algorithms.

Beyond the many technical challenges with MD, there also remains the more fundamental question of biological relevance. Interesting biological events, such as unfolding, binding, and conformational changes, tend to happen in the relatively long time-scales of microseconds to seconds. Analogous to the problem of repeated experiments with static crystal structures, rare events and conformations are difficult to detect with limited MD simulation times. Despite this well-documented concern, which has become known as the "sampling problem", MD simulations that make use of massive parallel distributed computing on high performance clusters are just now beginning to allow us to sample biological events.

An intrinsic problem with MD simulations is in determining the convergence time. Convergence refers to the initial equilibrium period of the simulation, often starting from a static structure resolved by experimental methods. As computing speed increases and simulations are run for longer time intervals, it is found that this equilibrium state often lasts much longer than initially expected, thus invalidating past conclusions [24, 25]. The problem of convergence is well understood by MD critics and advocates alike, and there have been many methods developed to quantify the problem statistically to increase confidence that a trajectory is in equilibrium [26-28]. Furthermore, there are many new and developing methods for improved sampling that include meta-dynamics, umbrella

sampling, steered MD, and many others, that consider known behaviors of proteins as a means of biasing the potential of the simulation to increase the likelihood of sampling specific states [29-31].

However, as longer time scales are reached on larger systems, the collective results suggest that the natural aging process of many proteins may exclude the prospect of true convergence on a theoretical level, meaning proteins can never reach a true state of thermodynamic equilibrium [32]. Notwithstanding many formidable obstacles, MD simulations have demonstrated predictive power beyond any other approach for studying protein dynamics thus far. The field of computational biology will doubtlessly succeed in obtaining faster and more accurate MD software, furthering the popularity of MD simulation as the primary computational method to glean new insights into protein function.

1.3 Distance Constraint Model

The Distance Constraint Model (DCM) introduced by Jacobs and Dallakayan in 2003, and subsequently applied to proteins in the form of a minimal DCM (mDCM) in 2004 [33-36], is a free energy decomposition method for calculating thermodynamic properties of molecular systems. Fluctuating interactions between atoms, such as hydrogen bonds, are represented as a set of distance constraints. When an interaction forms, there is an associated enthalpy contribution. In addition, depending on the details of the rigidity within the constraint network, an entropy contribution may also be added. Traditional free energy decomposition methods rely on the assumption that there is additivity in both enthalpy and entropy components, but it is known that this assumption is only true for systems that can be divided into independent subsystems [37, 38]. The additivity

assumption yields good approximate estimates for changes in entropy and free energy in small molecular systems that undergo limited conformational changes. However, large errors appear for flexible macromolecules that are stabilized by weak interactions such as hydrogen bonds and atomic packing, like those found in proteins [39-41]. Additivity holds for enthalpy in proteins, but entropy cannot be additive due to the presence of cooperative phenomena such as two-state folding, allostery, and large changes in stability caused by single point mutations [40, 42, 43].

The hallmark of the DCM is to account for non-additivity in entropy during the free energy reconstitution process, thus restoring the utility of the free energy decomposition paradigm for proteins. Molecular interactions are modeled as a distance constraint, or a set of distance constraints, and network rigidity is invoked to identify constraints that are effective in reducing entropy. A graph-algorithm called the pebble game [44] is employed to identify the distance constraints that alter the degrees of freedom of a molecular system, and to distinguish these from those that are redundant [45-47].

A simple two-dimensional representation of this concept is shown in Figure 1, where bonds are modeled as single distance constraints connecting four atoms. The first panel on the left shows the atoms with four bonds between them, represented as points and lines respectively, that have the flexibility to adopt alternative conformations in the two-dimensional plane without breaking any bonds. Notice that changes in conformation imply geometric changes, yet topology is conserved in all these cases. The middle panel shows the same four atoms with an additional fifth constraint, forming a rigid structure. The rightmost panel depicts the addition of a sixth bond which is redundant, in that if any one bond breaks, the cluster remains rigid and entropy does not change. The pebble

game identifies all rigid regions and all redundant constraints such that entropy can be weighted appropriately for each bond, thus allowing the free energy to be summed correctly.

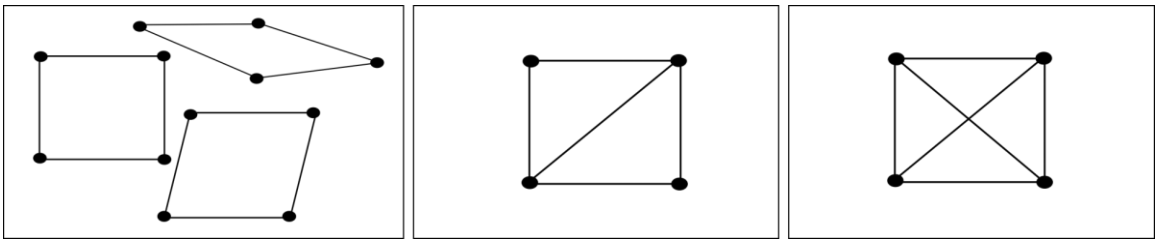

FIGURE 1: Two-dimensional representation of system constraints

The mDCM calculates the free energy landscape of a protein as a function of number of hydrogen bonds, number of constrained torsion angles, and temperature. The hydrogen bonds are identified from the initial input protein structure. Among various molecular interactions, the hydrogen bonds and torsion angle constraints can fluctuate, whereas covalent bonds are fixed. Monte Carlo sampling is used to explore thermodynamic properties such as constraint topologies and free energy, and to calculate characteristics of each constraint topology. The ensemble of topologies reflects fluctuating native contacts (e.g. hydrogen bonds) according to the geometry of the input structure, yet a given topology is associated with multiple geometries wherever there is flexibility. The advantage of the mDCM is that it samples constraint topologies directly without simulating atomic motion. Using relatively little computational time, the mDCM samples a large ensemble of topologies, resulting in accurate estimates of entropy and free energy. Consequently, the mDCM can identify many low-energy states that are separated by high-energy barriers. The disadvantage of the mDCM is that non-native

contacts are not accessible. In contrast, MD simulations explore non-native contacts, but cannot sample long enough to extract thermodynamic properties.

As its name suggests, the mDCM was created as a prototype to demonstrate that constraint theory, facilitated by the pebble game, would be able to reproduce heat capacity as measured by differential scanning calorimetry [35]. The mDCM incorporates three fitting parameters into the free energy decomposition. Although no other computational method has been able to reproduce experimental heat capacity curves in proteins, heat capacity curves have been matched to experimental ones with high accuracy in the mDCM by allowing these fitting parameters to vary. It has been found that these parameters are similar across many sizes and families of proteins, allowing for good predictions of protein thermodynamics even when heat capacity data is not available. Furthermore, it has been shown that by deliberately constructing incorrect heat capacity curves, the mDCM will not match arbitrary data, further indicating that the empirically derived fitting parameters are quasi-transferable and physically significant.

The mDCM has overachieved the original goal of proof-of-concept and has demonstrated unexpected predictive power in the study of protein dynamics [34, 48-51]. However, it remains a crude approximation of its original vision of a complete and accurate free energy decomposition method for the study of protein dynamics. It has long been recognized that understanding of protein behavior cannot be gained without considering the natural environment in which they function. Solvent, temperature, pressure, and chemical environment all effect stability, and thus functionality, and any complete model should address each of these factors.

1.4 FAST

Soon after the mDCM became operational, a much larger scale development project was undertaken [52-56] resulting in the Flexibility and Stability Test (FAST). FAST is designed to account for additional details in the free energy decomposition, such as solvent interactions and atomic packing, to achieve more accurate predictions. To capture a wide range of effects, differences in residue types must be accounted for, necessitating additional parameters and a generalized functional form for the parameterization. This more sophisticated model unfortunately requires estimation and extensive testing to determine the range and degree of transferability of these additional parameters.

Although these improvements are important upgrades to the current mDCM, the high-level design goals for FAST go beyond one-time model refinements. Important software considerations for the future success of FAST are usability and accessibility. The mDCM was originally written mostly in Fortran, a language known for its high performance in computationally intense applications, if not for its readability. However, applications involving large-scale software projects must seek to balance performance with long-term applicability, reusability, error detection, and fast development times. A major design purpose of FAST is to bridge this gap and produce a distributable and adaptable DCM implementation for end-users and experienced developers alike. To pursue the best of all worlds, FAST is written as a C++ class library of highly configurable individual components, which together can be combined to implement a customized DCM. There are over 2,000 classes, and 100,000 lines of code currently in the FAST library, all of which is mostly well-tested at the class level.

The FAST library contains many of the necessary components to develop a next-generation DCM application. Specifically, there are three significant upgrades that are planned for the next version: upgrading the solvation model, refining the model for packing entropy, and accommodating multiple conformations. The latter has already been researched and tested at a prototype level, demonstrating viability as an improved alternative to exploring only native contacts. This was accomplished through a hybrid method that combines the speed and sampling of topologies with geometric methods that efficiently generate new geometries using Monte Carlo moves [57, 58], holding constraint topology fixed. This is called a free energy driven simulation, and an unpublished prototype implementation essentially performs the rigidity analysis in mDCM on consecutive geometric possibilities to allow for very fast calculations sampling alternate conformations.

The upgrades envisioned to solvation and packing entropy are not as well developed and will require in-depth research and simulation before they can be implemented successfully. The mDCM crudely accounts for solvation using a mean field theory, adjusting the free energy according to the average energy of hydrogen bonds that are removed as constraints, and thus allowed to bond with solvent. This has proven to be a good first approximation with results fitting very well to experimental data. For the new upgrade based on the FAST library, a novel implicit solvation model will be based on known properties of water, parameterized from experimental data. In this model, there will be differentiation between free energy contributions from bulk water within the system, versus water molecules that interact directly with residues on or near the surface

of the protein. Bulk water properties are well defined, but surface affects must be parameterized.

Further refinements are also planned for the packing entropy calculations for residues. Conceptually, it is easy to imagine that the residues tightly packed together in the protein interior will have a relatively small contribution to conformational entropy. Conversely, when there is loose packing, either because a residue is on the surface or because it is immediately surrounded by void space, there will be a relatively large contribution to conformational entropy. The correlation between conformational entropy for a residue and its surrounding geometry seems obvious, but the relationship must be quantified to become a direct input to the free energy calculation. A first step towards this end is to study the spatial distributions of the void space surrounding residues in globular proteins. Quantitative insight can be obtained as to how local geometry affects conformational entropy, as well as hydrophobicity, in a more detailed way than previously considered.

1.5 New Contributions

With model extensions and integration of a few key computational methods, it is envisioned that the extensible FAST platform will support a powerful alternative to MD simulation. The contributions presented here are expected to further the development of a distributable generalized DCM using the FAST class library. With this unifying theme, under the umbrella of computational biology and structural bioinformatics, this thesis focuses on algorithm design, implementation, and data analysis needed to investigate spatial characteristics important to the stability of globular proteins.

The next two chapters describe methods implemented in C++ and integrated into the FAST library, following the object-oriented conventions previously established. Chapter

2 introduces the Protein Void Analyzer (PVA) that categorizes various types, quantities, and distributions of void space within a protein structure based on a PDB input structure. Chapter 3 introduces the PDF Estimator that employs a novel nonparametric method of estimating a probability density function for a sample dataset of independent and identically distributed continuous random variables. These algorithms were designed to be applicable to a variety of structural biology problems, and they have been utilized in multiple projects. The opportunity to test and refine these algorithms for practical applications has greatly enhanced their flexibility and robustness and demonstrated their value as stand-alone distributable software.

Chapters 4 through 6 describe three applications of the PVA and PDF Estimator, forming significant contributions independently. In the first application, Chapter 4, percolation theory is applied to globular proteins in a more detailed and complete way than has been done previously. The PVA is used to perform these calculations, employing a grid-based methodology that offers computational advantages over several popular protein volume calculation approaches. The second application, Chapter 5, is a new methodology developed to analyze MD trajectories. The PDF Estimator classes are incorporated into a customized tool, combining MD and Principal Component Analysis (PCA) to study Beta-Lactamase proteins. Finally, the functionality of the PVA and PDF Estimator capabilities are combined to investigate the distribution of void space throughout proteins in Chapter 6, providing high level statistics concerning the entropic nature of residue types.

CHAPTER 2: PROTEIN VOID ANALYZER

2.1 Introduction

The volume of space occupied by and surrounding a protein is an important property that offers insight into normal functionality and behavior. Although computing the van der Waals volume occupied by a collection of atoms is trivial, of practical interest is the volume of a protein in various conformational states. Proteins are known to be densely packed when in their native state [59-61], but even the most well-packed system will include small interior void spaces due to imperfect packing that contribute to the total internal volume. The size and distribution of these empty spaces are known to affect the stability of the protein and can account for pressure unfolding [62-64]. Additionally, small clefts and pockets along the surface are important distinguishing characteristics that often predict catalytic areas and binding sites, and identifying these properties is of interest in areas such as structure prediction and drug design [65-69]. Finally, quantities including protein volume and accessible surface area have been shown to directly correlate with hydrophobic energy transfer energies, which are a driving force in conformational changes [70-72].

Given the importance of these concepts, there have been many computer algorithms developed and implemented over the years to calculate protein volume, surface area, and void space, as well as to identify clefts and channels [65-67, 73-79]. These approaches differ widely in functionality, method, availability, and applicability to a variety of problems and analyses. General analytical methods for computing the volume of

intersecting spheres have been derived using alpha shapes, a term defined by Edelsbrunner in 1983 [80], that is closely related to Delaunay triangulation [81]. This approach is based on the Voronoi diagram, first introduced in 1908 by Georges Voronoi [82], and applied to protein packing by Richards and Finney in the 1970s [60, 83]. The alpha shape theory uses computational geometric methods to assign a volume to each point, or atom, in space, based on its proximity to neighboring atoms. Algorithms employing these basic principles continue to be developed and creatively applied towards applications in molecular biology [59, 61, 79, 84-86]. Although analytical methods in theory can provide highly accurate results for densely packed spheres, problems arise in applications to proteins, specifically in calculations near the surface where solvent interactions occur. To obtain realistic results, Voronoi implementations must make corrections for these surface effects, as well as account for the relative weighting of the bond lengths due to atomic differences. Many of the analytical methods for volumes, surface areas, and cavity areas are based, at least in part, on a Voronoi procedure, however other derivations have been shown to have some advantages in terms of speed and simplicity [87-90].

In contrast, an alternative class of algorithms, collectively referred to as grid-based methods, takes a strictly numerical approach to evaluating protein volume [65, 66, 73, 76, 91]. This is a rather large category that encompasses a variety of methods, the defining characteristic being that the protein is mapped onto a three-dimensional grid of very small discrete cubes that are traversed in some systematic way to evaluate the space interior to and surrounding the protein. Due to the discrete nature of this approach, the volumes calculated are approximations, and generally considered inexact compared to

analytic methods. Furthermore, the somewhat arbitrary decision on the size of the grids in the cube requires a judgment in the tradeoff between computational speed and accuracy. Grid-based algorithms are sensitive to this resolution, as well as the orientation in which the protein is mapped onto the grid [77, 79]. Nevertheless, grid-based implementations have reported competitive accuracy and speed, scaling linearly with protein size even for very large proteins [66, 67, 77, 86]. This is becoming important for the analysis of larger macromolecular structures that can be difficult using analytical methods [73, 86].

Numerical methods, not unlike those using Voronoi-type procedures, offer a great deal of flexibility in implementation. The new method developed and described in this chapter is a grid-based method that uses many traditional concepts for calculating and classifying protein volume but has several key advantages over existing algorithms. Scalability and performance are optimized using a clustering technique that minimizes memory requirements while keeping an accurate tally of void space volume and its distribution throughout the protein. Accuracy is further refined by incorporating a physics-based model of the inter-atomic forces in the protein, averaged over multiple rotations and probe sizes. This statistical approach creates a more realistic representation of protein properties, allowing for the natural flexibility of protein dynamics. Perhaps most importantly, however, is the detailed analysis of the solvent accessible boundary surrounding the protein atoms. The clustering algorithm provides a quantitative means for describing the boundary layer in terms of the local environment that will become important to the future solvation model.

2.2 Definitions of Volume Space

Among the significant advantages of the new method presented here is its ability to define and quantify all aspects of the space a protein encompasses. Although precise definitions vary, common important quantities generally include, at the least, protein volume, enclosed cavity volume, and solvent accessible volume. The distinction between these volumes is clarified by Richards' concept of the molecular surface, described simply as the surface accessible to a probe of some radius R, and has become a standard way of defining protein volume since it was introduced in 1971 [92]. This definition of molecular surface has been incorporated into many algorithms using a technique known as the rolling probe method of volume calculation and has enhanced both analytical and numerical calculations for various types of volume [65, 73, 75]. As the name suggests, the volume of a protein is delineated by a simulated spherical probe rolling around the surface of the protein. This model represents a water molecule surrounding the protein and is a means of determining the solvent-accessible molecular surface.

Richards pointed out several important concepts. First, note that the volume defined by the molecular surface is a function of the probe size, and will approach the van der Waals volume as the probe becomes very small. Second, as the probe becomes infinitely large, the molecular surface will approach a finite limiting value. Richards suggested the diameter of a water molecule, approximately 1.4 Å, as a reasonable probe size, and many have followed this convention when calculating protein volumes [77, 93-97].

With this model, it is easy to see that the volume enclosed by the molecular surface defines solvent accessibility. If there is empty space not occupied by the protein, but within the molecular surface, this is called void space, and can be divided into two

distinct categories: void space that is at least large enough to accommodate a spherically shaped probe (typically called cavity volume), and void space smaller than the probe. In this work, the latter type of space is referred to as *microvoid*. This is volume due to imperfect packing and is typically not directly calculated as a separate quantity of interest. In Voronoi methods, microvoid is generally included as part of the protein volume. However, analyzing microvoid in isolation, distinct from the van der Waals volume, can provide insight into characteristics common to many known proteins. Furthermore, the distribution of clusters of microvoid throughout the protein may have important implications in the dynamics of its function and the entropy allowed by residues in their native states. Figure 2 provides a visual overview of these concepts. Note that van der Waals, cavity, and microvoid volumes together are equal to the volume delineated by the molecular surface area defined by Richards.

An inherent conceptual problem occurs when attempting to identify solvent accessible channels that run through the interior of the protein. According to Richards' definition, solvent accessible volume is not included in the molecular surface area, therefore it becomes difficult to distinguish between an internal channel and the exterior of the protein. This ambiguity is avoided here by defining and calculating a separate category of volume that encompasses all the solvent accessible volume. As a practical matter of convenience, this solvent volume includes a shell four times the length of the radius of the probe size surrounding the protein. This is a default value, chosen to approximate the estimated chemical range of influence of the protein on the solvent, but can be overridden to any value. Any void space that has a path connecting to this shell is added to the solvent accessible volume. The coordinates of the four types of volume

defined (protein, cavity, microvoid, solvent accessible) are stored and summed separately, allowing for analysis and visualization of the locations of each.

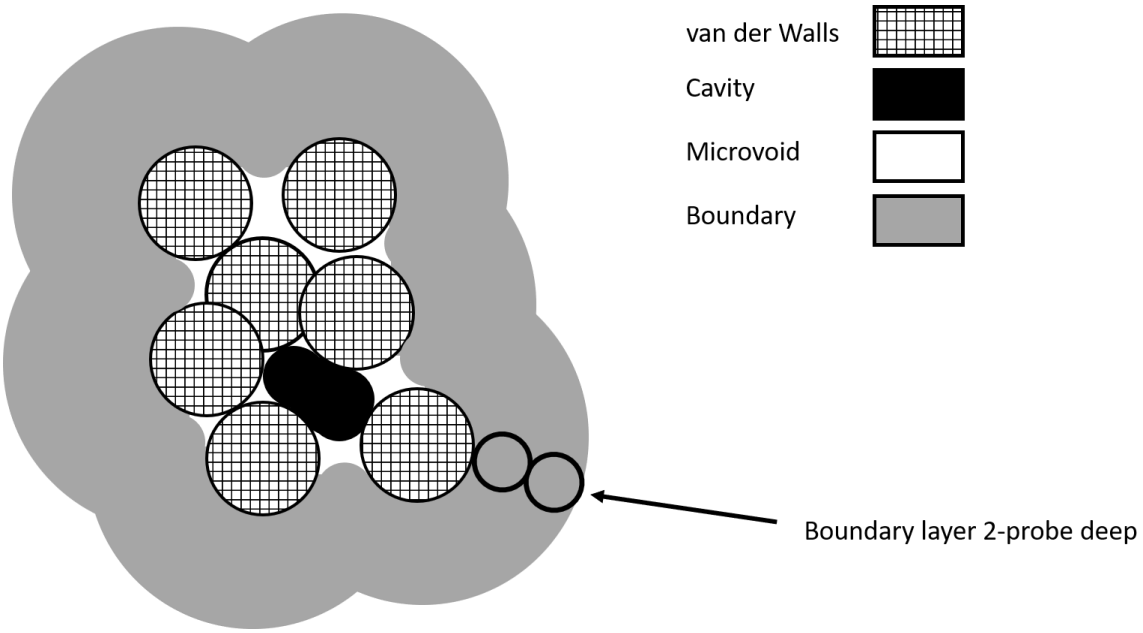

FIGURE 2: Protein volume definitions

2.3 Method

The PVA employs an implementation of the Hoshen-Kopelman (HK) algorithm to traverse through the protein in search of void space [98]. This method is based on a well-known algorithm called the union-find search and was originally developed to study percolation theory. Hoshen and Kopelman devised a computational method for finding the percolation threshold for finite systems by defining a quantity called the *reduced average cluster size* [98]. In their work, the authors also described the algorithm for cluster labeling that has been extended and used in many applications [99-102]. Percolation will be discussed in detail in Chapter 4.

The PVA is a three-dimensional adaptation of the HK algorithm for proteins beginning with the x-ray crystal structure. The three-dimensional spatial coordinates of all the atoms are mapped onto a hash grid and then traversed as consecutive two-dimensional slices, examining each grid point and its surrounding neighbors. The protein volume is calculated, and contiguous clusters of void space are identified and uniquely labeled. Only two slices of the grid and the associated cumulative quantities are stored in memory at any given time, thus greatly minimizing memory requirements. During a single pass thru the entire grid, all void space is detected and classified, and all four relevant volumes are calculated.

Determining whether a grid point is occupied within the van der Waals radius of a protein atom is trivial but classifying void space requires additional analysis. An empty grid point near a protein atom is potentially microvoid, but alternatively may be adjoined to a cavity (Figure 3). Thus, it becomes difficult to determine how to classify a single isolated grid point by strictly geometric means. To solve this problem, a novel spring model is employed which attempts to push a test probe centered at the grid point into a connected region of space which is probe accessible. Figure 4 provides a conceptual visualization of the spring model, where circles within dashed lines represent the probe.

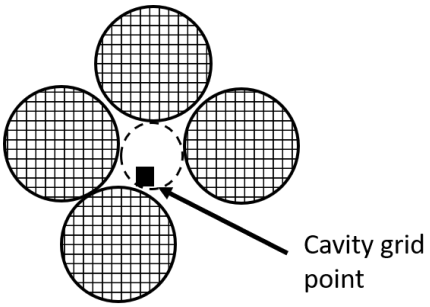

FIGURE 3: Cavity grid point located very near a protein atom

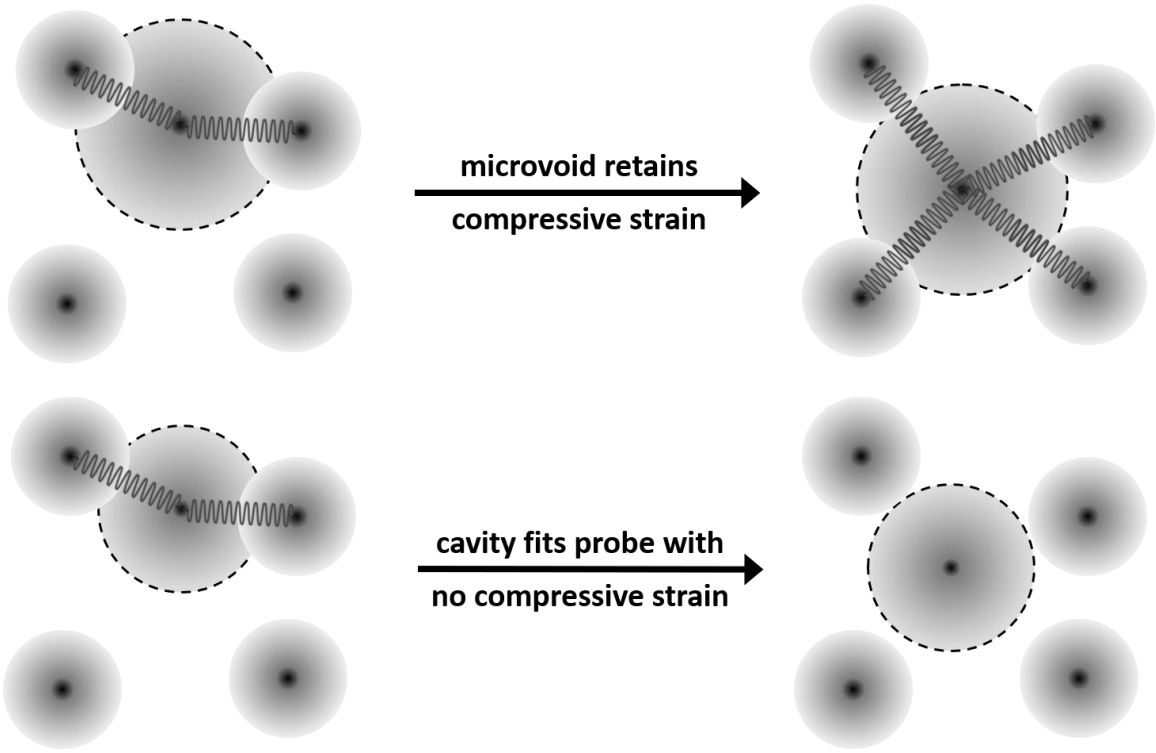

FIGURE 4: Spring model for cavity detection

For each grid point that is a potential microvoid, a surrounding network is constructed by placing springs between a test particle and all nearby atoms. Each spring has a natural length of $R_{vdw} + R_{probe}$, where R_{probe} is the radius of the probe and R_{vdw} corresponds to the van der Waals radius of the protein atom. The net force on the test particle imposed by the springs is calculated based on an assigned spring constant, but is only applied when the spring is compressed, not when it is stretched. The location of the test particle is then updated randomly in small increments through a series of iterations to explore the surrounding space. For each iteration, the potential energy of the spring network on the test particle is recalculated as estimated location of minimal energy is improved. If the resulting minimum energy is greater than a predefined threshold, then some spring compression remains. This means that the test particle could not be pushed

to a region where a spherical probe can fit, thus the grid point is microvoid. If the spring potential is less than this threshold but the final location of the test particle is farther than R_{probe} from the original grid point location, then the grid point is spatially connected to a cavity but lies within a microvoid pocket not accessible to the probe. Otherwise, the grid point is counted as part of the adjoining cavity volume. The flowchart in Figure 5 outlines the major decision points for the determination of void type.

Although the algorithm is efficient and conceptually simple, the implementation requires many practical decisions, including the spring constant, energy threshold, and number of iterations. These quantities have been extensively tested to determine reasonable approximations, and deviations from these choices have minimal impact on the accuracy of the results. Perhaps the most fundamental impact, however, is the ambiguity in the definition of the van der Walls radii for the atoms in the protein itself. Although modeling the atoms as hard spheres is a common method for obtaining realistic results, it remains a crude approximation, and methods for these approximations vary. Unfortunately, the quantitative results for the volume of a protein are dependent on and sensitive to the values used for the van der Waals radii. Bondi van der Waals radii [103] have been chose for this work, and although different definitions yield different protein volumes, the results were shown to be qualitatively the same. The algorithm maintains the flexibility to use any set of radii definitions the user wishes to incorporate.

As with all grid-based methods, the choices for grid and probe sizes also have significant impact on the performance and accuracy of the calculations. Although a probe size of 1.4 Å is nearly optimal for evaluating solvent accessibility in an aqueous environment, this definition is too restrictive for general analysis. Of critical interest in

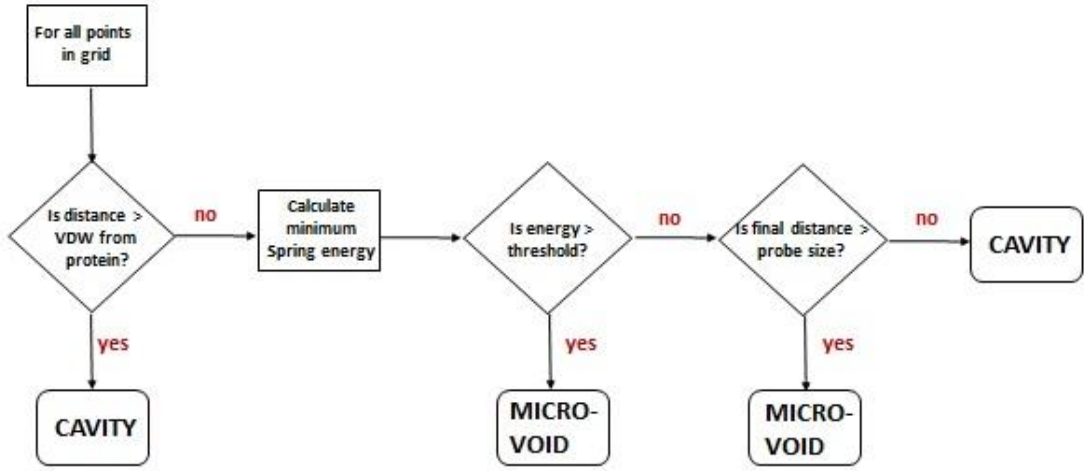

FIGURE 5: Flow chart for assigning void space to either cavity or microvoid

understanding protein-binding behavior are the interactions with other molecules across a range of sizes. Clefts and tunnels that restrict water can be ideal for filtering out larger molecules and enhancing specificity. Therefore, to maximize the applicability of the method, probe size can be set to any arbitrary value, according to the needs of the user. Similarly, grid size is best left as a user-defined value to be adjusted depending upon the desired level of resolution. Smaller grid sizes will produce increasingly accurate results, at the cost of longer compute times. These tradeoffs will be discussed in greater detail in the following sections.

2.4 Probe Averaging

Figure 6 shows a potentially problematic situation that can occur with clustering cavity volume. In this case, there are two adjoining cavities that will be incorrectly assigned to the same cluster despite the passageway that would restrict the probe from entering. This scenario appears to be only of theoretical concern because it is rare in

practice for static protein structures. However, due to dynamic fluctuations in proteins, it is likely that the probe will wiggle through the constriction, in which case considering the two void space partitions as a single cavity is correct.

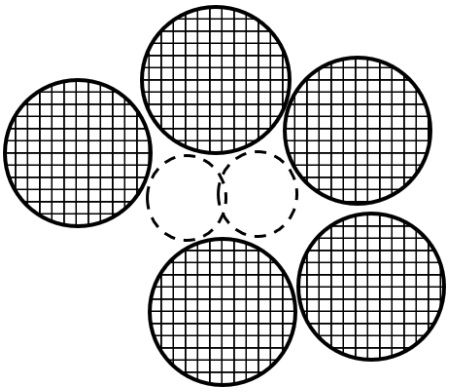

FIGURE 6: Overlapping Cavity Cluster

To model the statistical possibility, in this specific example, of two distinct cavities versus one continuous cavity, a rotational averaging method is applied. This is achieved in two ways. To generate results for analysis, the volume calculator is run 300 times. The protein is randomly rotated along all three axes such that its orientation on the three-dimensional grid is changed. For each orientation, the probe size is also allowed to fluctuate slightly to explore subtle variations in cavity clusters. Specifically, a Gaussian distribution is centered at the probe size with a standard deviation of 0.05 relative to the average probe radius. In this way, dynamic fluctuations of the protein are captured as probe radius fluctuations. All quantities calculated are averaged over rotations at various probe sizes to determine robust likely behavior.

2.5 Results

For initial testing of the protein volume calculations, a dataset of 108 globular proteins were selected. These structures were taken from the Top 500 proteins, originally published by Hobohm and Sander, selected for non-redundancy and structure quality [104]. This list has been continuously updated over the years, and currently contains 8000 structures downloadable from the Richardson Lab website. For the purposes of initial testing, the Top 500 list is further refined to eliminate those with missing residues,

missing chains, incomplete biological units, and fewer than 50 residues. Hydrogen atoms are computationally added to residues. Some preliminary testing was done on all 500 proteins, revealing that missing information at times has a significant effect on the results. This discovery underscores the sensitivity of these calculations. The subset of 108 complete proteins provides adequate representation of structures for testing the volume calculations and highlighting typical characteristics of void distribution.

2.5.1 Accuracy and Convergence

Protein volume and packing have been studied extensively, therefore the first step is to ensure that the results from the new method agree with known measurements. The generally accepted radius of a water molecule is 1.4 Å, therefore most physical and simulated experiments report volumes based on this probe size [77, 93-97]. Values for protein volume are generally qualitatively similar between different methods and software packages and comparing these to our measurements with a probe radius of 1.4 Å has shown good agreement, although exact quantitative comparisons will always be dependent upon the atomic van der Waals radii. Void volumes are much more challenging to compare, mostly due to differences in definitions, but there is qualitative agreement in most cases. A careful analysis of the reasons for many differences in volume calculations will be discussed within the context of packing density in Chapter 6.

To test for internal consistency, calculations were performed across multiple rotations and probes, and across different resolution scales by changing the grid size. In Figures 7(a) and 7(b), the coefficient of variation, defined as the ratio of the standard deviation to the mean, expressed as a percentage, is plotted as a function of grid size for van der Waals, microvoid, and boundary volumes. Figure 7(a) shows the convergence for each

volume type with 300 rotations for a fixed probe size of 1.4 Å. The coefficient of variation decreases as grid resolution increases, demonstrating virtually negligible error with rotation, even for relatively large grid sizes. The convergence of the same quantities using the probe averaging method with a mean radius of 1.4 Å is shown in Figure 7(b). The grid size dependence for the van der Waals coefficient of variation is essentially the same regardless of whether the probe radius is fixed or exhibits variation, indicating van der Waals volume is independent of probe size. Interestingly, microvoid and boundary volumes have nearly constant coefficients of variation across all grid sizes, indicating that the random variation on the test probe radius introduces much more variation than grid size. However, the reason for placing variation on the test probe radius is to better model atomic fluctuations that are present in protein structure. Because the large variations derive from probe size variation, averaging over multiple rotations is not necessary. Nevertheless, while averaging over probe size, randomizing over protein orientation is also performed due to its negligible computational cost.

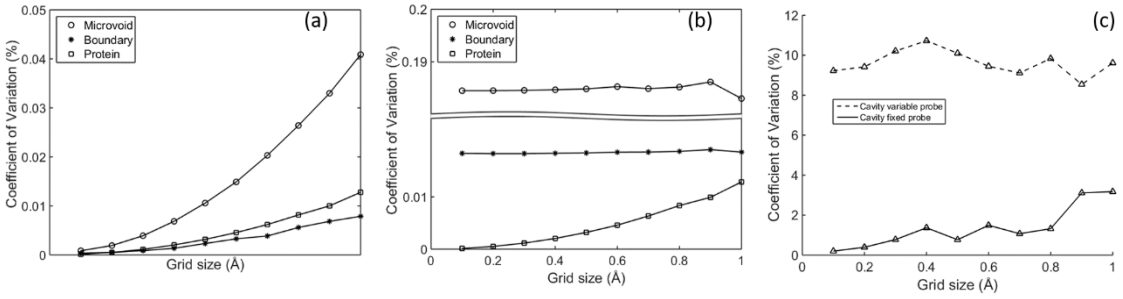

FIGURE 7: Rotational convergence for fixed and variable volume types

The variation of cavity volume due to random protein orientations for fixed and varying probe size is shown in Figure 7(c). As with microvoid and boundary volumes, for

fixed probe size, the variations in cavity volume decrease with decreasing grid size. However, the coefficient of variation is considerably larger than that for other volume types, possibly because cavities can connect to channels that extend to the protein surface. For a sufficiently small probe radius, the channel is open, allowing solvent to penetrate the cavity. In this case, cavity volume transforms into boundary volume. Similarly, two microvoid clusters may merge together to form a larger cavity. As such, when a probe radius is close to the critical threshold for a cavity to dramatically change size, this creates a high degree of sensitivity to protein orientation. These effects are enhanced by varying the probe radius, as shown in Figure 7(c). Again, the variation in probe radius increases the coefficient of variation and appears independent of grid size.

2.5.2 Time Complexities

Figure 8 summarizes how CPU times vary as a function of grid size, probe radius, and protein length. The results shown in Figure 8(a) are for the case of varying grid size, a, with a fixed probe radius, R, of 1.4 Å plotted on a log-log plot with a slope of -2.95, indicating an inverse cubic relationship. The dependence on test probe radius is more complicated. Figure 8(b) shows results for a varying test probe radius and a lattice constant of 0.1 Å. The probe radius is plotted against the log of the CPU times, for an approximately exponential relationship. For both cases, CPU times for each of the 108 proteins were averaged and normalized by the protein's sequence length (e.g. number of residues, N_R).

A linear scaling of CPU time with protein length is seen in Figure 8(c), which shows that the method will remain a viable approach for large molecular systems. Points are plotted for all 108 proteins for a grid size of 0.1 Å and probe radius of 1.4 Å. The

performance of our algorithm has been empirically confirmed by observing the same trends across many different combinations of probe and grid sizes. This performance benchmark reveals that the CPU time to process a protein is given by

$$T_{CPU} \sim N_R \frac{e^{cR}}{a^3}. \quad (1)$$

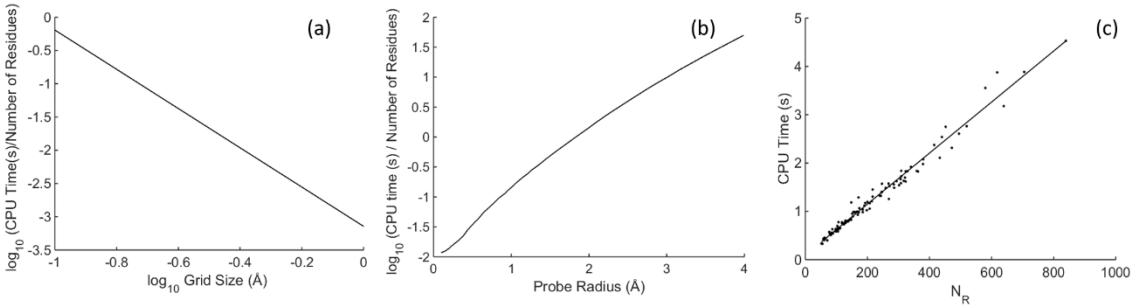

FIGURE 8: Time complexities for Protein Void Analyzer

2.5.3. Linear Scaling by Protein Length

The results thus far have demonstrated good performance and accuracy of the method as a function of grid size, averaged across all proteins of varying lengths. However, for all grid and probe sizes considered, it is observed that some quantities depend strongly on protein length. In Figure 9 the results are summarized for the case that grid size and probe radius are 0.1 Å and 1.4 Å, respectively. In addition, qualitatively similar correlations appear (not shown) when the number of residues (i.e. protein length) is replaced by number of atoms.

In Figure 9(a), it is seen that van der Waals volume and microvoid volume are proportional to protein length. In contrast, cavity volume, Figure 9(b), is shown to have only a general tendency to increase with protein length. These trends have been previously established for cavity and van der Waals volumes using a variety of methods and datasets [59, 77]. However, to the best of our knowledge, microvoid as a separate quantity has not been considered. Figures 9(c) and 9(d), show that the total number of distinct microvoid clusters is proportional to protein length, and the number of distinct cavities (e.g. cavity volume clusters) is approximately linearly correlated. Although it is intuitive to expect various volume type totals to be extensive, it is interesting that the

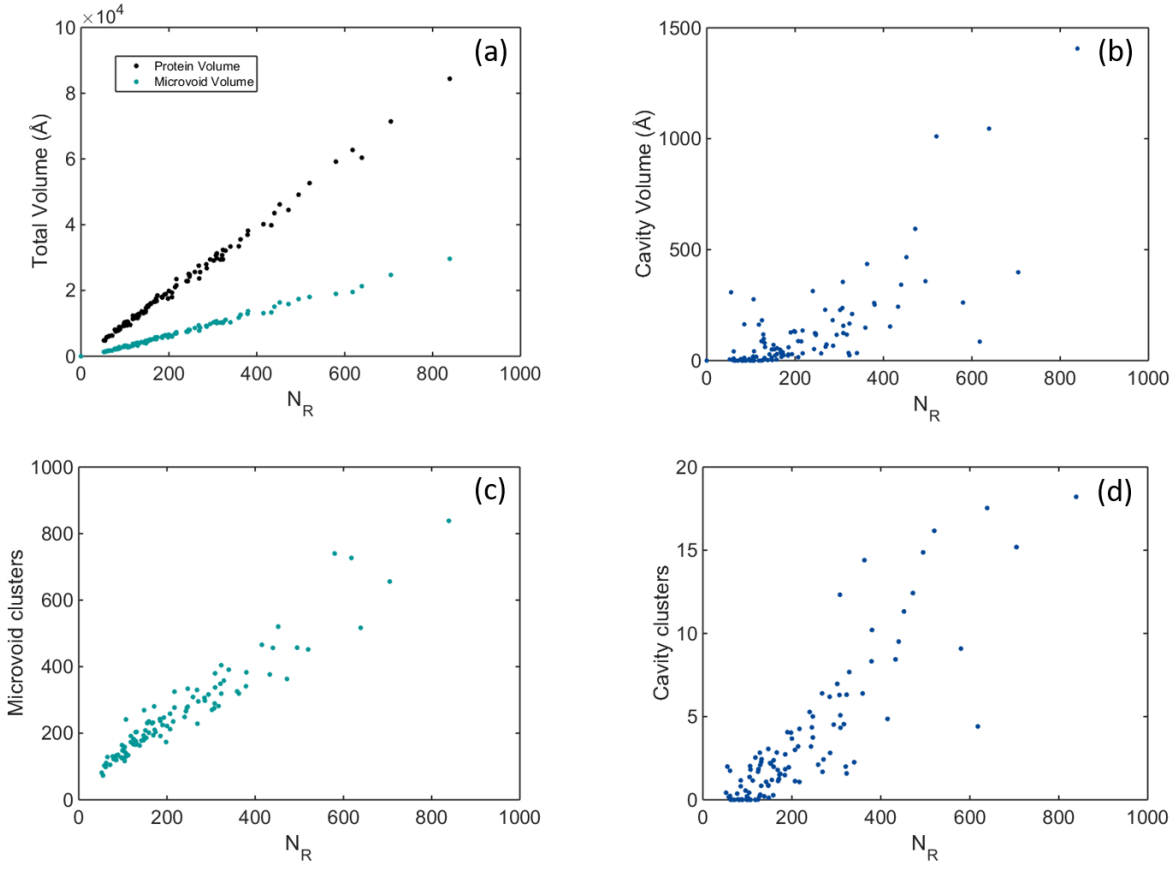

FIGURE 9: Volume characteristics as a function of protein length

extensive property holds for microvoid and cavity clusters. These plots show which quantities scale with protein length, and the scaling behavior of cavities versus microvoid is differentiated. In particular, the microvoid volume dominates the total void space quantitatively. Cavity volume, on the other hand, is more of the exception in void space and its characteristics are erratic by comparison.

2.5.4 Volumes as a Function of Probe Size

Figure 10(a) shows the packing density as a function of probe size. Packing density is defined as the protein volume divided by the total volume, including cavity and microvoid. These results indicate that atomic packing in globular proteins do not vary much from one protein to the next. At a probe radius of 1.4 Å, the packing density is around 75% for all proteins tested, which is consistent with other reports, and is comparable to optimal packing of random spheres confined in a box [59].

Figure 10(b) shows the relative relationships between cavity, boundary, and microvoid volumes as a function of probe size. Specifically, the curves plotted represent the fraction of each volume type over the total molecular volume. The absolute protein volume remains fixed over all probe sizes, but as the size of the probe approaches zero, the volume of the protein defines the entire molecular volume, with no microvoid volume. In this extreme case, the packing density approaches 1, and solvent can access all areas around the protein atoms. This is consistent with Richards' definition of molecular surface area. Richard also noted that for infinitely large probe sizes, these relative fractions should theoretically approach a constant value, and the shape of the curves in Figure 10(b) is suggestive of this behavior as probe size increases.

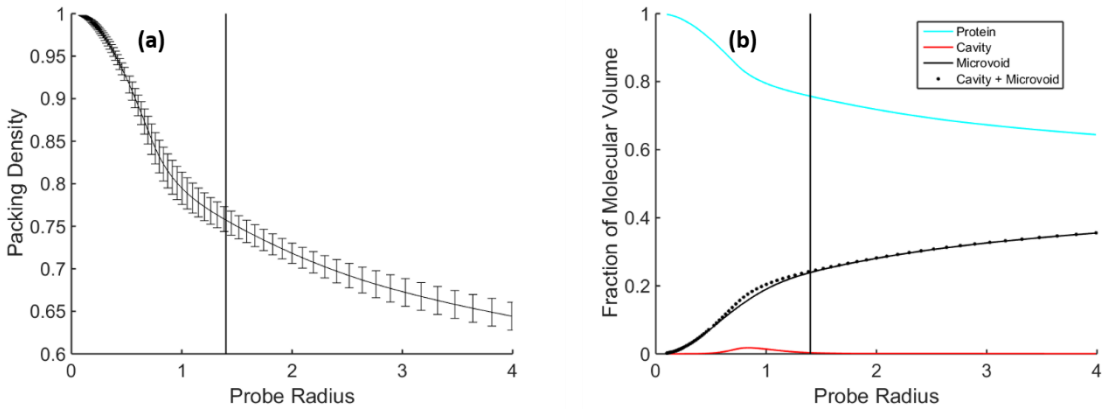

FIGURE 10: Volume fractions as a function of probe radius

Another feature to note in this graph is the very small contribution of cavity volume to the total. At the size of a water molecule, there are typically almost no cavities in the average protein. However, focusing in on the details of cavity volumes highlights the interesting features of individual proteins. Figure 11(a) represents the total cavity volume, as a function of probe size, for 107 of the 108 proteins tested. There is a clear signature of peak cavity volume at a probe radius size of around 0.9 Å, significantly smaller than a water molecule. The remaining 11 proteins, with peaks well above typical probe sizes, are shown in Figure 11(b), and are far more erratic. To understand this behavior, consider the example protein shown in Figure 12, at probe sizes 0.1, 0.8, 1.1, and 2.0 Å from left to right. Spherically shaped clusters are represented by various distinct colors to highlight separate cavities. At very small probe sizes, there are small pockets of cavities that do not amount to a large collective volume, because most of the void space is solvent accessible. As the probe increases, it cannot access these smaller areas in the interior of the protein, so the cavity volume increases.

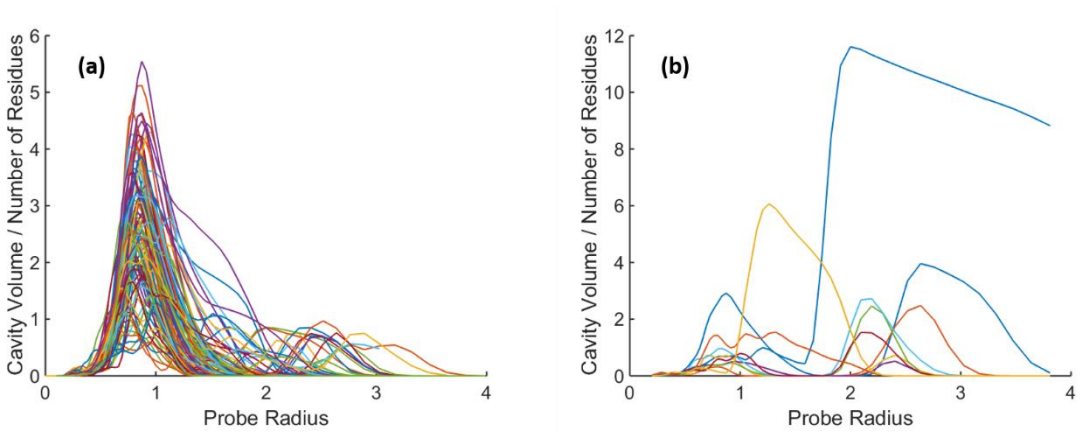

FIGURE 11: Cavity volume as a function of probe radius

However, past a radius of 0.9 Å, fewer clusters of voids are found simply because the proteins themselves are so densely packed. In a small percentage of proteins, such as those found in the example of Figure 12, there can be additional peaks in cavity volume for probe sizes much greater than 0.9 Å. An example is shown in the final panel of Figure 12, where a large cluster of cavity volume appears that was previously solvent accessible. In these situations, there is a large tunnel through part of the protein that will allow water molecules or other smaller particles to pass into but will restrict larger probes. Given the high energy cost and potential instability of a lower packing density, these tunnels often have a critical biological function [62, 105-111].

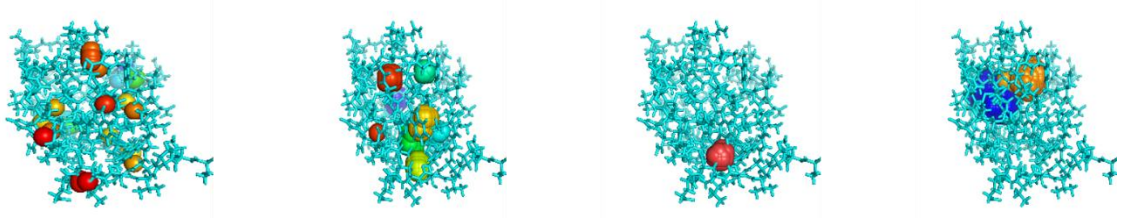

FIGURE 12: Cartoon representation of cavities with increasing probe size, colored by cluster

2.5.5 Partial Volumes

For each of the void types calculated, the total volume is calculated for the entire protein. It is also quite useful to know how these void volumes are spatially distributed. For example, residues with disproportionately larger volumes of boundary solvent volume in their immediate vicinity are likely to be found near the surface of the protein. As the probe size increases, residues buried in the center of the protein will be surrounded by virtually no solvent. To quantify these concepts within the PVA, partial volumes are tracked and recorded for each atom and residue.

Operationally, partial volumes have a simple definition. For each grid point of a given type of void volume, the closest atom is identified, and an associated counter is incremented. After the entire protein has been processed, these counters are tallied per atom to determine the respective number of associated void grid points. Partial volumes of various void types will be used in future DCM enhancements to entropy and solvation free energy calculations. These partial volumes are conceptually similar to solvent accessible surface areas (SASA) computed by popular software packages such as DSSP [112] and NACCESS [113] and have good correlation. Figure 13 shows this correlation for all residues in 108 proteins at a grid size of 0.3 Å and probe radius 1.4 Å.

2.6 Summary of Protein Volume Calculator

This chapter has introduced the method for the Protein Void Analyzer and demonstrated its consistency with other software and known properties of proteins. The speed, low memory requirements, accuracy, and user versatility make the PVA a

competitive alternative to existing methods as a standalone software and will eventually be available for download to the public. The detailed, atomic-level information captured

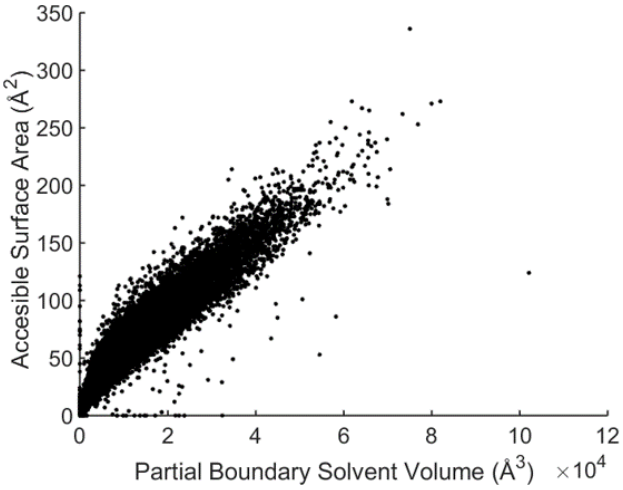

FIGURE 13: Correlation between SASA and partial volumes

provides a powerful analytical tool in the study of protein characteristics and behavior. The combination of features encompasses many other popular packages, including some basic visualization capabilities. Future enhancements, not in the scope of this dissertation, would include a user interface and more comprehensive visualizations, perhaps with a tie-in to another graphics package such as Pymol [114].

CHAPTER 3: PROBABILITY DISTRIBUTION FUNCTION ESTIMATOR

3.1 Introduction

Determining the probability density function (PDF) of a set of sample data is an important and long sought-after goal in the field of statistics, with applications in all disciplines of science, mathematics, economics, or virtually any situation in which predictions are made based on past observations. The problem becomes more challenging when nothing is known about the original mechanism or distribution of the data, and worse still when the sample size is very small. For distributions with long tails or rare events, many observations are needed to adequately sample the population.

For nonparametric density estimation (ie, the form of the distribution is not known) a common approach is kernel density estimation (KDE), available in many mathematical software packages such as MATLAB and R. The premise of KDE is similar conceptually to that of a histogram. However, instead of simple bin counts, the method employs a kernel function, usually a Gaussian, to represent each collection of points in a bin, thus smoothing the distribution. Primary difficulties for KDE include choosing a bin width, and appropriately defining the distribution at the boundaries.

Presented here is a novel nonparametric density estimator that begins with a maximum entropy method (MEM), based on an algorithm first introduced in 2009 [115]. The PDF Estimator deviates from standard methods and forms a unique hybrid algorithm. Specifically, the PDF Estimator combines maximum entropy, maximum likelihood, and order statistics in a new way. Each of these concepts and how they are combined to form

the new algorithm will be briefly explained in the remainder of this chapter. For more detailed descriptions of the method and further examples, see prior publications [116, 117]. A brief overview given here highlights the important features.

3.2 Maximum Entropy

The MEM is a known theoretical parametric method for estimating a PDF [118-120]. Calculus of variations is used to determine the form of the PDF that maximizes entropy, subject to restraints, resulting in the following form [115]:

$$p(v) = exp\left[(\lambda_0 - 1) + \sum_{j=1}^{D} \lambda_j g_j(v)\right] \quad (2)$$

The coefficients, λ, represent D Lagrange multipliers, and $g_j(v)$ are any set of bounded orthogonal functions, in this case chosen to be Chebyshev polynomials. Although the concept is not new, the Lagrange multipliers cannot be determined analytically beyond the first few moments, and numerical solutions become unstable due to errors in statistical measurements for dimensions greater than about six [121, 122]. Furthermore, even when solutions are found, the number of moments is predetermined, thus marking this as a parametric method.

The PDF Estimator takes a different approach to the MEM, using an iterative method to guess the Lagrange multipliers and create a trial PDF using Equation 2. The PDF is then numerically integrated using a second order approximation (Simpson method) with an adaptive, data-driven resolution, *dx*. Then the CDF is evaluated for each of the sample

data points, resulting in a corresponding data set on the interval [0, 1]. If the CDF is the correct representation of the initial data, this transformed data will be sampled uniform random data (SURD). Methods for evaluating SURD characteristics will be discussed in the next section.

Initially, only a single Lagrange multiplier is used, assuming a uniform distribution. If the uniform CDF produces a mapped set of SURD, then the PDF is accepted, otherwise, a new Lagrange multiplier is added. Trial PDF solutions are iteratively tested using a random search method to perturb the Lagrange multipliers, until either a solution is found, or a set amount of trials have occurred without improvement. Additional Lagrange multipliers continue to be added at an accelerated rate until an acceptable solution is found. The user may determine the maximum number of Lagrange multipliers added until giving up. Figure 14 shows the high-level process for the algorithm.

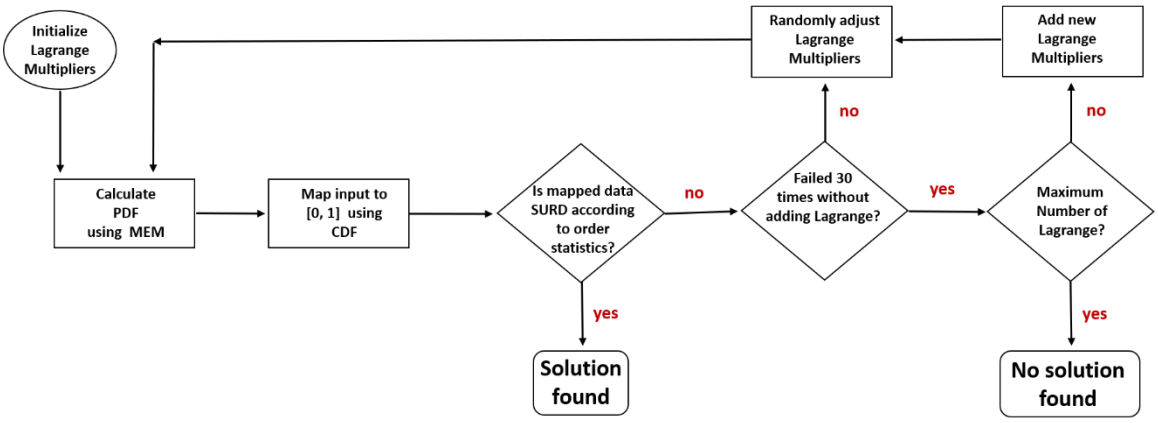

FIGURE 14: Flow chart for MEM

3.3 Order Statistics and Maximum Likelihood

According to order statistics [123], the probability of the finding *u* at position *s* for *N* random samples on the interval [0, 1] can be given by,

$$p_s(u|N) = \frac{N!\,(1-u)^N u^{s-1}}{(N-s)!\,(s-1)!}. \tag{3}$$

Taking the product of all *N* of these probabilities forms a likelihood function, *L*, that will be used to assign a score to the likelihood of a random sample having the characteristics of SURD. Extensive numerical tests were performed on actual SURD across a range of sample sizes using a random number generator [124], and the distribution of the natural log of *L* is shown in Figure 15. The distribution is scaled by the square root of *N*; thus it

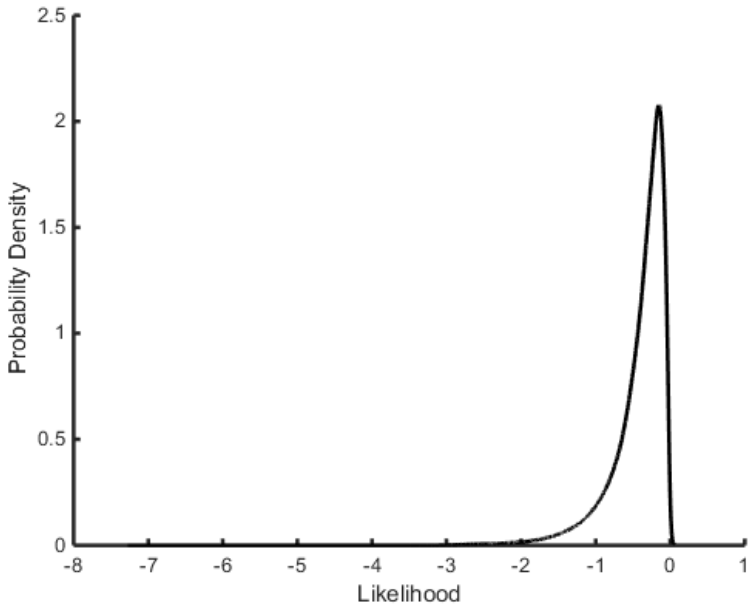

FIGURE 15: Distribution of likelihood function for SURD

is sample-size invariant. By integrating the area under this curve, a confidence level is assigned indicating the likelihood of SURD. This confidence level is assigned to 40% by default but is user-customizable.

3.4 Results

A more in-depth description of the user-options and challenges of the PDF Estimator, as well as many examples, have been published previously [116, 117], but a few results will be highlighted here to demonstrate the advantages of this method, particularly as compared to KDE. Figure 16 is an example demonstrating convergence to the true PDF as sample size increases. This sample data for this distribution was created

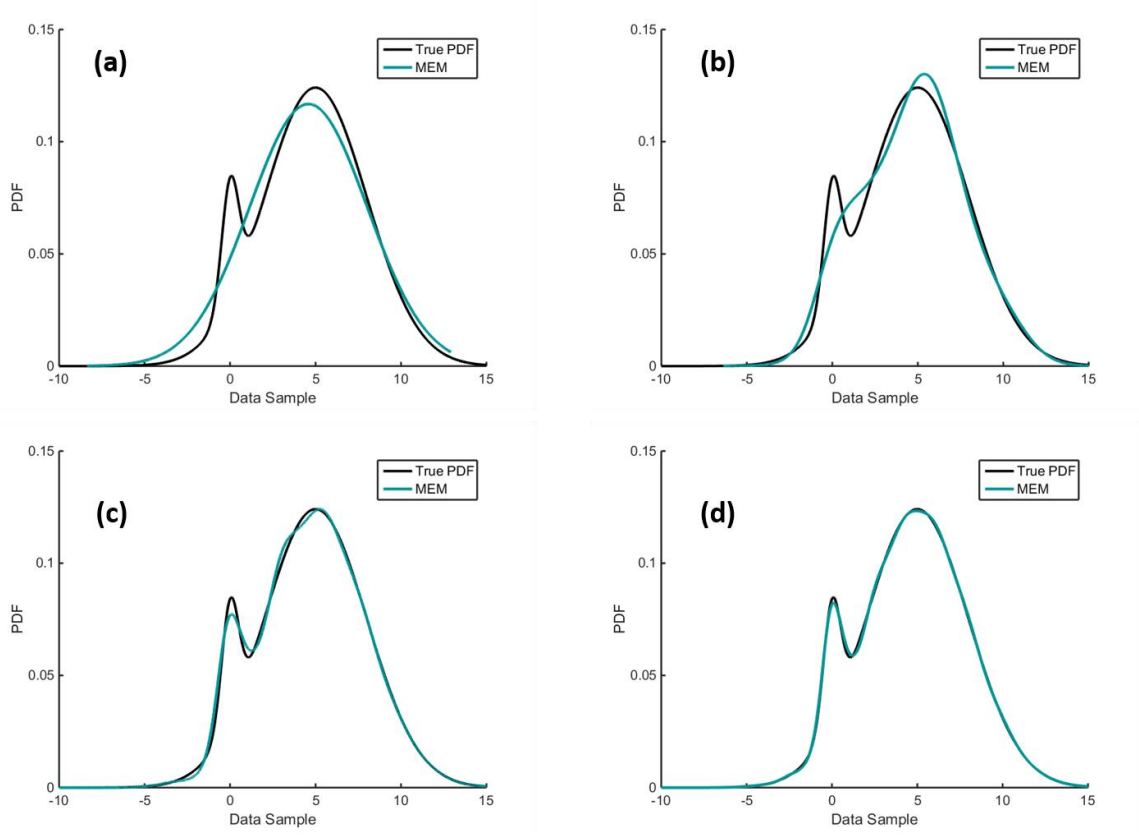

FIGURE 16: Convergence of double Gaussian distribution with increasing sample size

as a weighted sum of two Gaussian distributions. Although detecting the smaller peak is challenging, this is relatively simple for most methods at higher sample sizes, and MEM and KDE perform similarly. For more rigorous testing, a collection of difficult distributions was selected, containing features that are problematic for estimators. These distributions include those with discontinuities, heavy tails, and multi-resolution scales. A brief description of four of these challenging test cases will be summarized here. Figure 17 shows each these distributions for large sample sizes (2^{16}), comparing MEM and KDE performance against the true PDF, using the default settings for both methods. The MATLAB 2014a ksdensity() function was used for KDE results.

3.4.1 Uniform distribution

Although seemingly the simplest distribution of all, the uniform distribution is surprisingly difficult for KDE-based estimators to detect. The results shown in Figure 17(a) demonstrate this difficulty with KDE. Setting appropriate boundary values minimizes the dips at the endpoints somewhat but does not eliminate this problem. The PDF Estimator can fit a uniform distribution near-perfectly in most cases because of the iterative nature of the method. The first iteration begins with a single Lagrange multiplier, reducing Equation 2 to a constant, therefore resulting in a high chance of an immediately successful score.

3.4.2 Cauchy distribution

Heavy tails, representing rare events, present a problem for density estimators, particularly for small sample sizes. A somewhat extreme example of this is the Cauchy distribution, described by the following equation.

$$p(x) = \frac{b}{\pi(x^2 + b^2)} \tag{4}$$

For these examples, *b* was chosen to be 0.5. In the Cauchy distribution, the second moment does not converge, but increases infinitely with sample size, causing it to be nearly impossible to estimate based on a finite sample. The PDF Estimator handles these

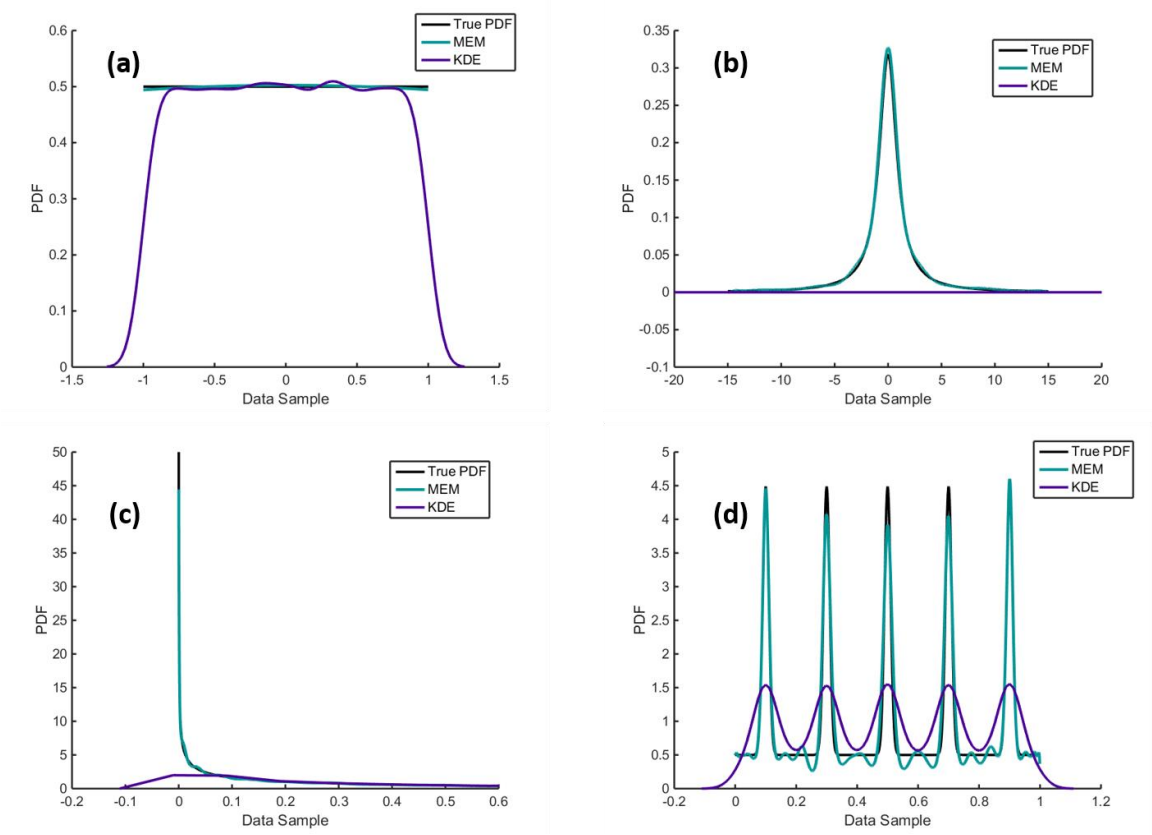

FIGURE 17: Comparison between KDE and PDF Estimator for large sample sizes

situations by automatically detecting extreme outliers. This detection is data-driven, with no intervention from the user. Future versions of the method include a proposal to create

a semiparametric MEM to alter the form of the PDF expansion but will not be addressed here.

Another feature of the PDF Estimator that this distribution demonstrates is the adaptive integration scheme. This is also an automatic data-driven feature that occurs in all cases but is designed specifically for detecting sharp features. The program integrates the PDF using multiple resolution *dx* values, to accommodate precise integration where needed, but will not waste memory and computation time in areas where the data is sparse. Figure 17(b) shows that the KDE method cannot fit to the true distribution using the default settings, for the very reason that the range is too great compared to the sharpness of the peak. Additional tests were done (not shown), demonstrating that ksdensity() can perform similarly to MEM by increasing the discrete points to one million, rather than the default value of one hundred.

3.4.3 Gamma distribution

The gamma distribution is defined as

$$p(x) = \frac{e^{-|x|}}{\sqrt{x}}. \tag{5}$$

This distribution is included in the results as an example of a discontinuity at x=0. The KDE has difficulties at the boundary, as with the uniform case, and misses completely the divergence at x=0, shown in Figure 17(c). Defining boundaries manually and increasing the number of points on the x-axis does not significantly improve KDE performance for the gamma distribution.

3.4.4 Five weighted Gaussian distributions

This is an artificially constructed distribution designed specifically to test the limits of sensitivity of the PDF Estimator. Five equally spaced Gaussians with a small standard deviation are added to a uniform distribution to create the sharp peaks shown in Figure 17(d). The PDF Estimator does a much better job of fitting to the peaks but does not produce a smooth curve as does KDE. The attempt to flatten the lines between peaks results in small fluctuations, but overall represents the true distribution reasonably well.

3.4.5 Independent assessment of results

Many metrics were employed as a means of testing the validity of the log-likelihood scores, including Kullback-Leibler (KL), Kolmogorov-Smirnov (KS), and a least squares method of comparing distributions that was developed and named the Figure of Merit. The results of the p-values using the 1-sample KS test are shown in Figure 18. The PDF Estimator creates an analytical solution for the estimated PDF, using the empirically determined set of Lagrange multipliers, and this solution is used to generate an independent data sample according to this distribution. This sample is compared to the original known distribution using the KS test, which produces a p-value for the null hypothesis that the distribution does not match the data. P-values were generated and collected for a range of sample sizes, and include additional distributions, most of which were constructed to be difficult to estimate. The histogram in Figure 18 shows a somewhat uniform range of p-values, indicating good estimates.

3.5 Summary of PDF Estimator

The PDF Estimator was initially motivated as an extension of the original MEM version almost ten years ago [125]. However, the program has undergone significant

conceptual improvements, two major rewrites, and extensive testing against known distributions with generated data. Distributable versions are currently available in both Java and C++, and the two implementations continue to be tested in parallel as opportunities arise, providing an extraordinarily high level of confidence that

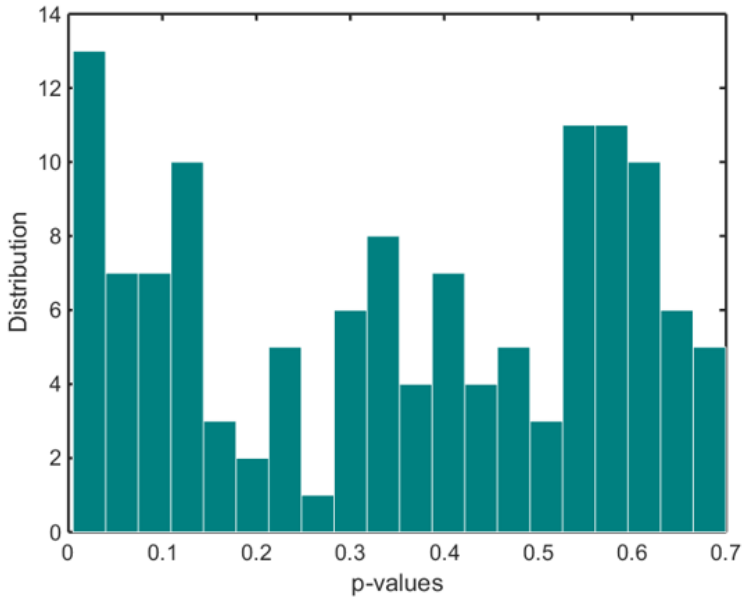

FIGURE 18: P-value distribution of PDF Estimator results using the 1-sample KS test

programming bugs do not exist. The C++ version has been added to the FAST library, and was implemented with computational speed as a priority, as this will become important for the new DCM. Additionally, the C++ code has been uploaded to the Comprehensive R Archive Network (CRAN) and packaged to interface as a function with the R statistical software.

One reason for its importance is when calculating potential of mean force from sampled data. It was noticed about 10 years ago that KDE is simply an untrustworthy

approach for such calculations. When the exact answer is known KDE may be sufficient, provided the correct kernel is selected. However, from a big data, high-throughput analysis point of view, a fully automated robust method is needed, without subjective human intervention for choices of kernel characteristics such as bin width.

In addition to its primary motivating purpose within FAST, the program has been utilized for multiple projects in the lab using real data with unknown distributions. The following chapters provide examples of specific applications for this algorithm, and it has been used throughout this work for data analysis. The class component design and its implementation provide an interface allowing for customizable future development without the need of rewriting or retesting code; the ultimate goal in object-oriented code design.

CHAPTER 4: PERCOLATION

The Hoshen-Kopelman clustering algorithm used to calculate void volumes also provides a means tracking the size, shape, and number of contiguous clusters and study their distributions throughout the protein. As previously mentioned, the HK algorithm was originally developed to study percolating systems. A porous material is said to *percolate* if a liquid can find a path from one end of the material to the other, such as water percolating thru coffee grounds. If the material is too densely packed, the percolating substance will form isolated clusters throughout, but will not form a pathway through the material. Of particular interest is the percolation threshold, defined as the minimum probability of site occupation that allows for percolation. Percolation theory has remained a topic of theoretical interest for many decades and continues to be applied to a range of practical problems [47, 126-131]. Many of the characteristics of the clusters of any percolating system are shown to be universal, regardless of the application.

Percolation has previously been applied to proteins from various perspectives. Mostly commonly, the protein atoms themselves are viewed as the percolating clusters when folded into their native states [59, 132]. Additionally, cavities have been studied for their percolation characteristics [59]. Cavities can never percolate, since they cannot form a pathway through a protein, whereas protein atoms in their native state behave as if they are a percolating system very near the threshold. There are many challenges specific to this application. Most notably, proteins are unique dynamic biological systems with a great range of variability in behavior, relying on a single x-ray crystal structure to capture

their shape. Protein studies therefore cannot hope to obtain the fine-tuned precision of computational experiments where the parameters are well-defined and controlled. For example, the molecular surface defining the internal protein volume is variable and irregular in shape, whereas the most well-studied three-dimensional percolation systems have precise shapes, typical cubic.

Furthermore, percolation theory in its simplest form considers systems with random site probabilities. That is, each site or grid point on the grid lattice has an equal probability of being occupied. Clearly this cannot be the case with proteins, as grid points within the van der Waals radii of atoms will always be occupied by protein volume. Percolation with non-random and correlated probabilities have also been studied [130, 133, 134], but none of these models accurately reflects the unique combination of challenges found in the application to proteins. This chapter will apply percolation theory to proteins in a way not previously considered, by analyzing microvoid clusters and how they merge and connect through the protein with varying probe size. Despite the difficulties mentioned, microvoid has been demonstrated to behave as a percolation system, sharing many common traits among the proteins in the 108-protein dataset.

4.1 Microvoid Percolation Threshold

Figure 10(b) in Chapter 2 shows the fraction of microvoid volume increasing as a function of probe size. Rephrased in the language of percolation theory, it can be said that as molecular surface increases, the probability of a given grid point to be counted as microvoid volume increases with probe size. As this probability increases, there will be a theoretical threshold at which the microvoid volume will percolate. That is, there will exist a continuous pathway of microvoid through the protein. Figure 19 demonstrates

this visually. The gray shading represents the microvoid in the system, with the largest microvoid cluster shown in black, for probe sizes 0.3 Å, 0.4 Å, and 0.5 Å. Prior to percolation, the largest cluster is a very small fraction of the microvoid, but these small clusters merge together until they span the system in at least one dimension. This transition often happens quite suddenly. By the time the probe radius is only 0.5 Å, the largest contiguous cluster represents virtually the entire microvoid volume.

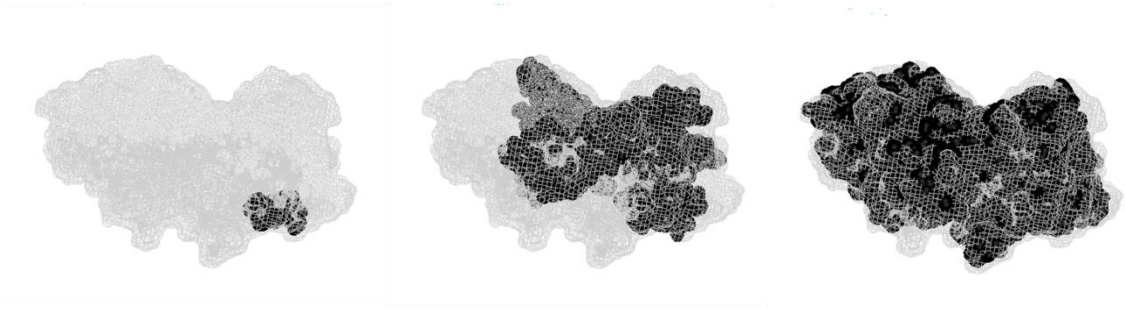

FIGURE 19: Visualization of the largest microvoid cluster near the threshold

Formally quantifying microvoid within the context of percolation theory required specific user-defined parameters to the PVA. First, consider the probability of microvoid volume within a fixed system size. Therefore, it is convenient to keep a constant shell distance surrounding the protein as the probe size increases. Additionally, to obtain precise results to study percolation, the probe radius is not allowed to fluctuate within the 300 rotations for each probe size definition. Neither of these restrictions are necessary to demonstrate percolation but they allow for a clearer mapping to definitions of probability.

The site probability is formally defined as

$$p = \frac{microvoid}{microvoid + cavity + boundary}, \tag{6}$$

where the fixed shell layer for the boundary is set to 0.55 Å for all probe sizes, and the van der Waals volume is omitted from consideration since it can never be occupied by microvoid. The choice of 0.55 Å is somewhat arbitrary and the exact value is not important. However, allowing a much larger boundary shell causes the site probabilities to shrink to zero, making the outcome computationally intractable.

Several other volume sums in the denominator of Equation 6 were considered, including, but not limited to, microvoid as a fraction of total volume and of molecular volume. There was some consistency in overall results for all definitions but this one was chosen primarily as the one with the most reasonable physical interpretation. The site probability, therefore, is defined as the probability of microvoid as a fraction of all void space. This definition maps well to probe size for all proteins, as shown in Figure 20(a).

For finite systems, a good operational definition for the effective percolation threshold, p_c, is the site probability associated with the peak value of the reduced second moment (RSM), defined as [98]

$$RSM = \left(\sum_{s=1}^{\#\,clusters} n_s s^2 \right) \Big/ N - n_{max}^2 / N\,, \tag{7}$$

where s is the cluster size, n_s is the number of clusters of size s, and N is the total number of microvoid volume units (grid points). The percolation threshold for each of the 108 proteins was calculated individually and found to be very similar for all proteins. The

dark circles in Figure 20(a) mark the probe size and site probability of the threshold for each individual protein, and Figure 20(b) shows the RSM as a function of site probability. Although p_c occurs at a range of probe sizes, the associated percolating site probabilities are within a relatively narrow range for all proteins.

The RSM in Figure 20(b) was calculated from the collective cluster size statistics across all 300 rotations, therefore defining average behavior per protein. Figure 20(c) demonstrates more clearly the percolation transition. In this case, the RSM was calculated for each rotation separately, for the same range of site probabilities. The fraction of percolated realizations out of 300 is plotted as a function of the site probability

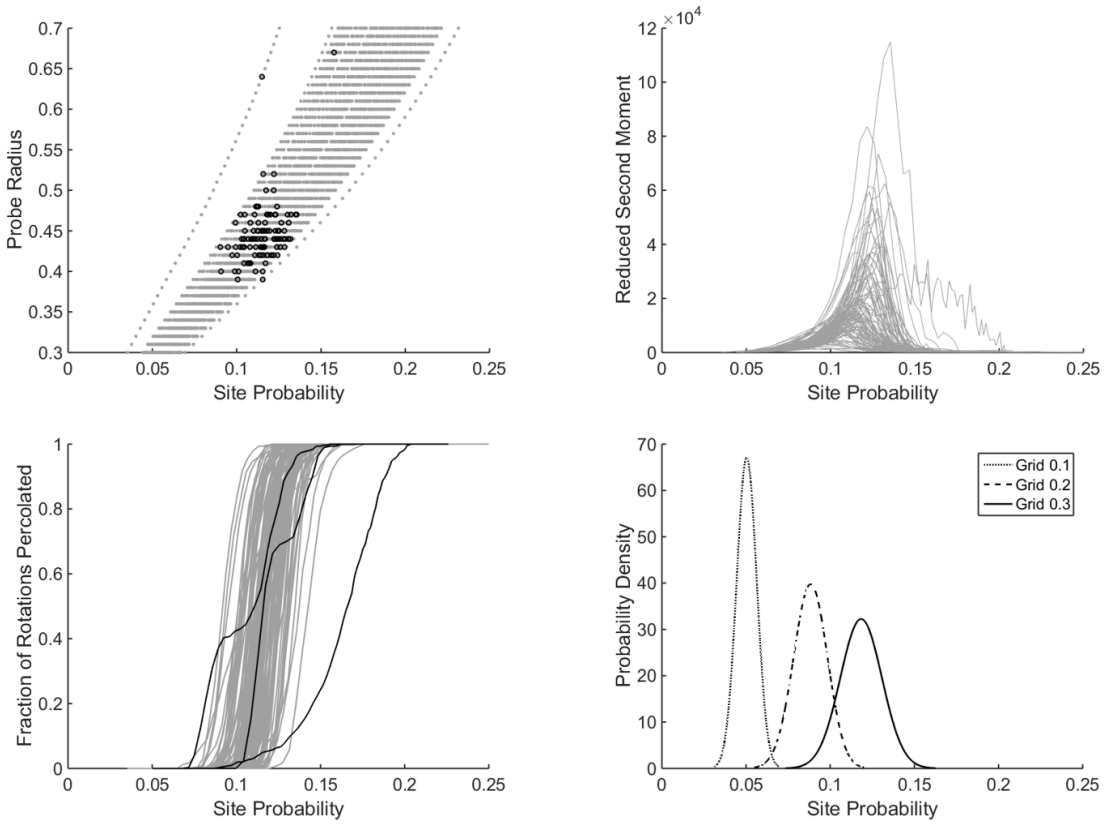

FIGURE 20: Microvoid percolation characteristics by probe and grid size

for each protein. In most cases, the result is the expect sigmoidal shaped curve indicating a sharp transition. Two exceptions, shown in black, are cases where the transition does not occur smoothly. Interestingly, these exceptions represent the largest protein in our dataset (839 residues) and one of the smallest (61 residues). A third exception, the rightmost black line, is an example of a highly non-spherical protein with an unusually long transition. These cases, along with the noise and scatter surrounding the RSM in Figure 20(b) highlight the challenge of applying percolation theory to biological systems. Nonetheless, the percolation characteristics are remarkably similar across a very diverse collection of proteins.

The plots discussed thus far in Figure 20 were all produced using a lattice constant of 0.3Å. Figure 20(d) shows the probability density for the percolation thresholds for three different lattice constants, which highlights a few interesting characteristics. The p_c shifts to the left with decreasing grid size as the spread in the distributions narrow, such that the coefficient of variation is constant. Additional small narrow passageways that can be traversed by the probe as the resolution of the grid is increased are the cause of this shift to smaller percolation thresholds. This shift is therefore an artifact of using a discrete grid. Note that grid sizes greater than 0.4 cannot accurately determine the microvoid percolation threshold because the grid spacing cannot be greater than the probe radius, otherwise the probe size will be smaller than the grid resolution.

The site probabilities in general are significantly smaller than that of a random cubic lattice. Although the percolation threshold for systems between 2 and 6 dimensions are not known exactly, three-dimension cubic lattices have been well-studied computationally with a confirmed p_c of approximately 0.311 [131, 135]. There are

several likely explanations for low microvoid threshold probabilities. Unlike the cubic grid with equal random site probabilities, microvoid can never occupy protein or bulk water grid points, thus it remains a small percentage of the total volume. Furthermore, it is intuitive to expect that the microvoid naturally clusters together in a non-random manner, and the high connectivity represented at relatively small site probabilities confirms this intuition. Non-random correlated probabilities and non-spherical cluster shapes have both independently been shown experimentally to lower the percolation threshold [133, 134, 136-138].

4.2 Microvoid Cluster Dimensionality

A system in a state very near the percolation threshold has certain known theoretical characteristics. Among these traits is the relationship between the volume and surface area of the clusters. Somewhat counter-intuitively, this relationship is expected to be linear in theory, and has verified experimentally for many systems [59, 126, 131, 139]. This relationship holds for probabilities above and below the threshold as well. Figure 21 shows surface area as a function of volume for both cavity and microvoid, at the approximate microvoid percolation threshold. These figures plot every cluster for all 108 proteins. For microvoid, this includes over 500,000 data points, which all fall closely in line with one another. Interestingly, cavity also follows a linear fit moderately well, even though these clusters do not percolate. Similar results were seen for other probe sizes.

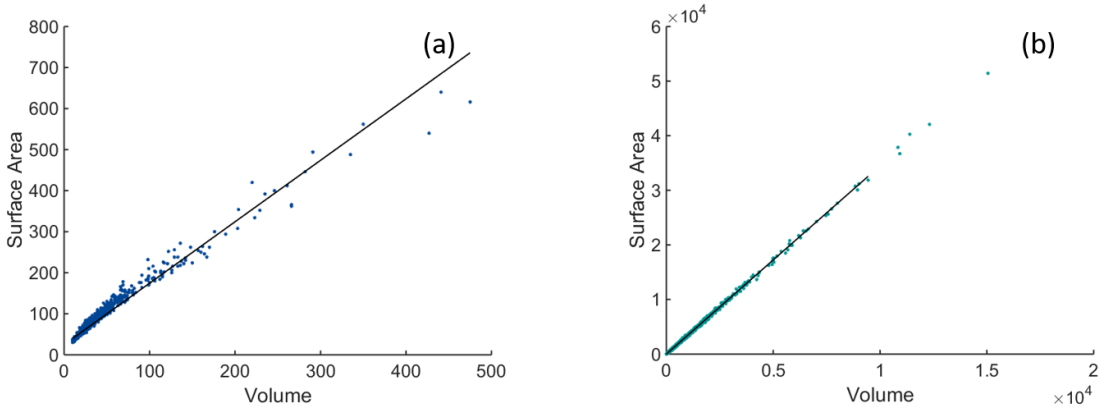

FIGURE 21: Linear relationship between volume and area for (a) cavity (b) microvoid

Another interesting geometric characteristic to consider is sphericity, defined by [140]

$$sphericity = \frac{\pi^{1/3}(6 * volume)^{2/3}}{surface\ area}, \tag{8}$$

where a perfect sphere would give sphericity equal to 1. Plotting sphericity as a function of cluster volume (Figure 22) for cavity and microvoid shows that, while small clusters tend to be somewhat spherical, this trend decreases notably for large clusters, particularly microvoid clusters. Microvoid clusters larger than a handful of grid points are highly non-spherical, confirming the visual trends seen (Figure 19) as microvoid spreading in long narrow channels throughout the protein as the threshold is approached. Cavities by comparison tend to be much more spherical, but become somewhat less so as volume increases, in agreement with other studies of cavity shape [141].

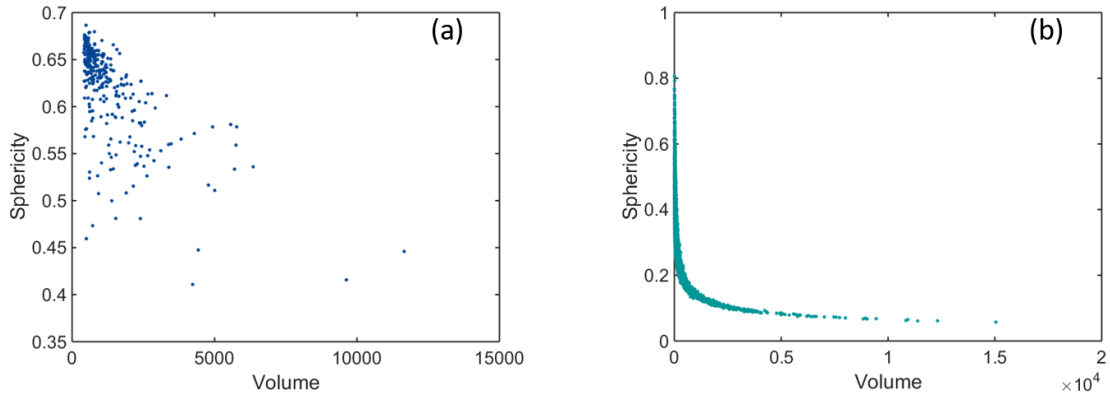

FIGURE 22: Sphericity for (a) cavity and (b) microvoid

The surface area calculations for clusters are not currently part of the PVA implementation. For these results, the surface area was calculated in MATLAB as a post-processing step. A user option in the PVA, previously mentioned as a means of generating PYMOL images such as those shown in Figure 19, is to save the visualization information for void clusters. This produces cluster volume coordinates both in the original coordinate system of the crystal structure, as well as grid points. The latter was traversed in the MATLAB script to determine the number of exposed faces of each grid point, thus calculating the surface area. Incorporating these surface area calculations as a part of the PVA would be a valuable future enhancement, providing much greater efficiency with C++.

4.3 Finite Size Scaling and Standard Percolation Exponents

According to ordinary percolation theory, a set of universal scaling laws describe many characteristics of cluster size and distribution. These laws are defined by scaling exponents, called critical exponents, which are dependent only upon the dimensionality of the system. For dimensions less than three or greater than 7, analytical solutions for critical exponents exist and have been experimentally verified [129-131]. Although these

exponents have been computationally determined for three-dimensional systems with high accuracy, percolation problems involving experimental data from real systems often deviate from the expected values [126, 128, 135, 136, 142, 143]. These scaling laws have been applied to protein microvoid percolation with somewhat better success than expected, given the variability of properties of globular proteins and the limited sample size. Table 1 summarizes the critical exponents that will be discussed throughout this section.

TABLE 1: Expected and estimated critical exponents

Exponent	Description	Expected	Estimated
ν	Finite size scaling	0.87	0.722 ± 0.061
β/ν	Strength of percolating cluster	0.49	0.353 ± 0.061
df	Fractal dimension	2.52	2.651 ± 0.061
τ	Cluster size distribution	2.19	2.134 ± 0.033

The results discussed up to this point have calculated p_c as an empirical value based on the cluster statistics. According to percolation theory, however, the threshold is a function of the linear size of the system [131]. It is intuitive to expect that a small system will percolate with a much lower site probability than that of a larger system. For ordinary percolation problems, the threshold in the theoretical limit can be extracted from the effective probability, $p_c(L)$, determined by the peak RSM, according to the following scaling with the linear dimension of the system as

$$p_c(L) = L^{1/\upsilon} + p_c(\infty). \quad (9)$$

where ν is a critical exponent universally dependent upon dimensionality.

To accurately estimate the theoretical p_c in the limit of an infinite-sized protein system, corrections must be made for finite-size scaling (FSS) effects. The linear dimension of a protein can be calculated as the average of the maximum range of each coordinate axis as follows [59, 139],

$$L = \frac{1}{6}\sum_{j=1}^{3}\left(x_{j,max} - x_{j,min}\right). \tag{10}$$

Figure 23 shows the probability according to Equation 6 at the peak RSM for each protein as a function of $L^{1/\nu}$ for a best linear least square fit. The large amount of scatter across the proteins reflects, in part, the inadequate sampling possible with a finite set of protein sizes, as typical globular proteins occur within a somewhat narrow range of sizes. Ideally, thousands of random realizations for sizes ranging at least several orders of magnitude are employed to precisely extrapolate an accurate threshold [130, 135, 136, 143, 144]. Nonetheless, from this plot, the slope and y-intercept can be found, giving values for ν and $p_c(\infty)$ respectively.

Typically, the next step is to estimate the critical exponent beta. Beta is the scaling exponent associated with the strength of the infinite cluster according to the equation

$$P_\infty \sim |p - p_c|^\beta, \tag{11}$$

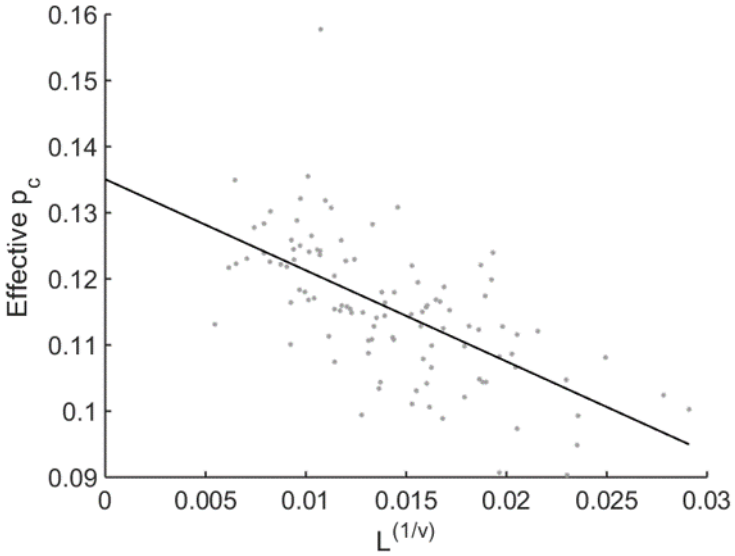

FIGURE 23: Fitted finite size scaling exponent

where P_∞ is the probability that a random site belongs to the infinite cluster. Applying FSS to Equation 11 yields the following equation, which is somewhat less critically sensitive to an exact estimate of p_c [144].

$$P_\infty(L) \sim L^{-\beta/\nu}, \tag{12}$$

The scaling exponent β/ν is calculated by finding the slope of the double log plot of $P_\infty(L)$ against $L^{-\beta/\nu}$, then beta is calculated based on the estimated value of υ from Equation 9. Corresponding values of the fractal dimension df and the critical exponent tau are then calculated according to the following universal scaling relations.

$$df = d - {}^{\beta}/_{\upsilon} \tag{13}$$

$$\tau = {}^{d}/_{df} + 1 \tag{14}$$

This procedure for systematically deriving the exponents $\beta, df,$ and τ are all dependent upon a good estimate of ν. Given the wide scatter in Figure 23, and the subsequent large margin of error in estimating p_c, the rest of the exponents cannot be estimated with high accuracy. Therefore, for this percolation problem, a slightly different approach is taken by reversing the procedure. This is a somewhat novel strategy not previously considered in ordinary percolation problems but applies well to this unique system of diverse proteins. First, Figure 24(a) shows fitted values for ${}^{\beta}/_{\nu}$ as a function of p_c within the margin of error estimated from the y-intercept in Figure 23. This curve is calculated by applying Equation 12 to a range p_c by taking the log of both sides and estimating the slope of the linear fit. The curves for df and τ are then calculated from Equations 13 and 14 for each estimated value of ${}^{\beta}/_{\nu}$.

The precise values for exponents df, τ, and ${}^{\beta}/_{\nu}$ are quite sensitive to the value of p_c, demonstrating why an accurate estimate of ν is essential. However, the fractal dimension can also be derived independently, providing additional information for the appropriate scaling constraints. Figure 24(b) shows the volume of the largest microvoid cluster as a function of the maximum length of this cluster, expected to scale as

$$target\ cluster\ size \sim largest\ cluster\ volume^{\,df}. \tag{15}$$

The slope of the double log plot is 2.6510 ± 0.0606, in close agreement with the value of 2.52 expected for three-dimensional percolation systems. More importantly, df calculated in this manner substantially narrows the range of p_c as shown in Figure 24(a). The three vertical lines in this figure show the mean (center line) and margin of error (flanking lines) of df as calculated from Figure 24(b). These lines mark the corresponding estimates and error margins for the other exponents plotted and are noted in Table 1.

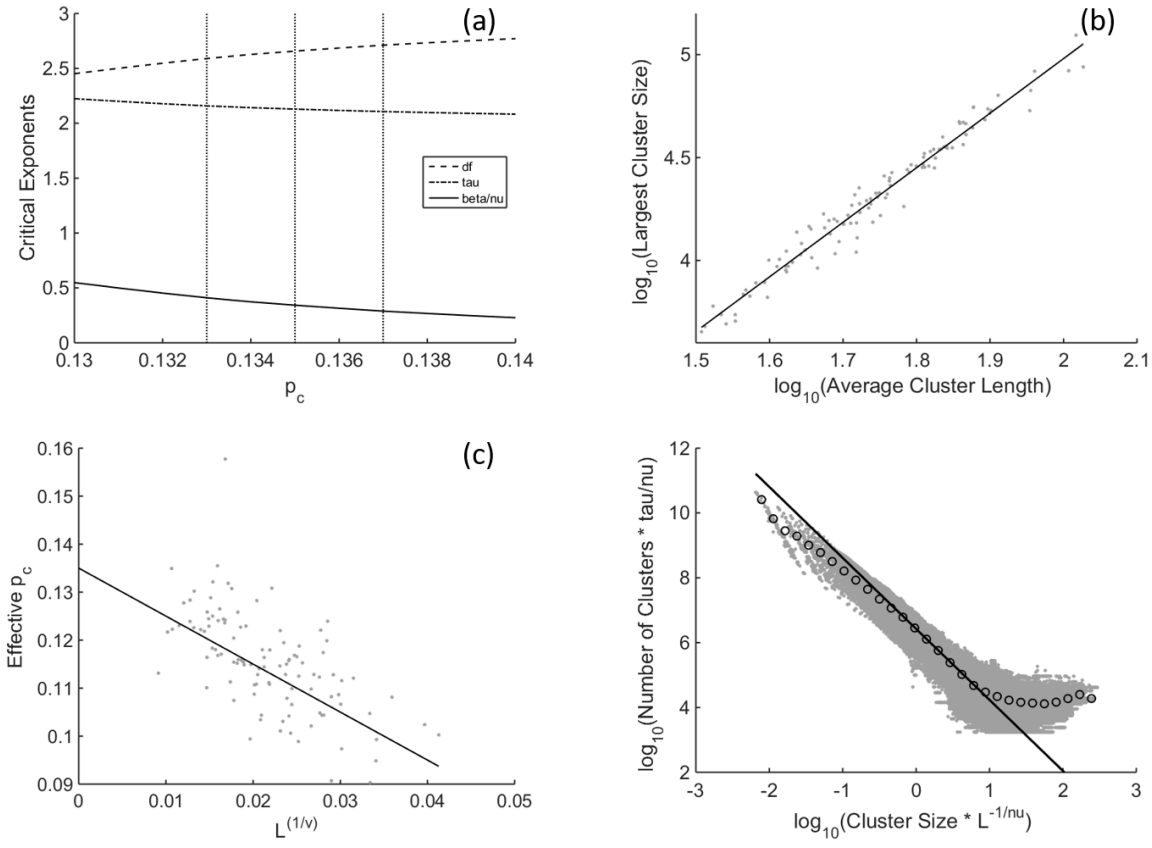

FIGURE 24: Percolation threshold constants

Furthermore, the mean value of p_c can be estimated with this technique as well, and is reported as 0.135 ± 0.002. Again, although this agrees very well with the extrapolated value of p_c from the y-intercept of Figure 23, the more important point is that the error is significantly reduced by accurately determining the fractal dimension first. This, in turn, allows for a more accurate estimation of nu, shown in Figure 24(c). In this case, the y-intercept is fixed for the mean value of p_c, and a least squares fit is computed to estimate the best value of υ, estimated, also reported in Table 1.

Finally, Figure 24(d) shows the cluster size distribution for all microvoid clusters for each protein, at the mean p_c calculated from Figure 24(a). Statistics were collected on all cluster sizes for each of the 300 rotations, binned logarithmically, and plotted on a double-log scale. The circles represent the binned averages, and the gray dots include all the cluster sizes. The cluster size distribution at p_c is known to follow the power law

$$n_s \propto s^{-\tau}, \tag{16}$$

where n_s is the number of clusters of size s. Following FSS, data collapse is achieved with the following length scale corrections,

$$L^{\tau/\nu} * n_s \propto L^{-1/\upsilon} * s^{-\tau}, \tag{17}$$

according the exponent values estimated for this data, and the straight-line fit was forced to have a slope of -τ. The power law in Equations 16 and 17 are valid for large cluster sizes only, and the linear fit shown in Figure 24(d) works very well for finite cluster

volumes greater than 1 Å. The characteristic leveling off at the tail reflects poor sampling in the largest cluster sizes, with only one or two per bin [128, 142, 143].

In summary, the results in this section demonstrate that microvoid clustering can be modeled as a 3-dimensional percolation problem, based on a set of diverse proteins. The critical exponents fall closely in line with expected values and have been confirmed self-consistently. The problems of limited sampling of a narrow range of finite sizes are overcome by working through the normal methods in reverse, starting with the fractal dimension, which is estimated with high accuracy.

4.4 Protein Percolation

A different way of viewing percolation is by looking at the protein residues themselves. The length, according to Equation 10, is plotted for all proteins against the van der Waals volume of the protein on a log-log scale (Figure 25). The slope of the linear fit is 2.56, indicating that proteins are packed similarly to random spheres at the percolation threshold. Similar results have been reported using different protein volume calculators [59].

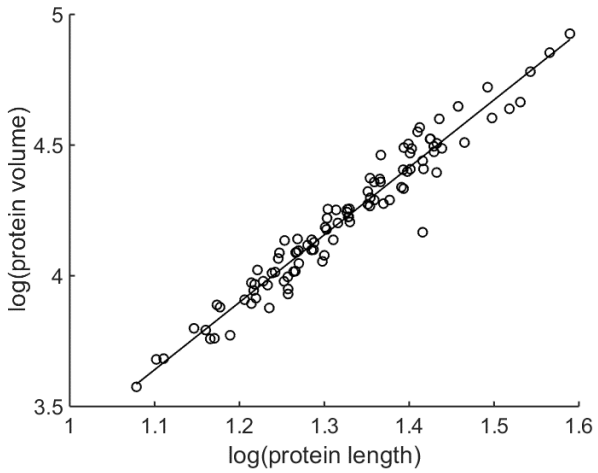

FIGURE 25: Log-log plot of protein volume as a function of protein length

CHAPTER 5: PDF ESTIMATOR APPLIED TO MOLECULAR DYNAMICS

A common method employed when analyzing MD trajectories is to calculate the root mean square fluctuation (RMSF) for each residue in the protein. The RMSF is defined as follows, for each x, y, and z coordinate in the trajectory:

$$RMSF = \frac{1}{\#\,frames} \sum_{i=1}^{\#frames} \sqrt{(\langle x \rangle - x_i)^2 + (\langle y \rangle - y_i)^2 + (\langle z \rangle - z_i)^2} \qquad (18)$$

Plotting the RMSF per residue provides a quick and easy method to spot which areas of the protein are most prone to movement during the simulation. RMSF values can also be compared between different simulations, such as mutations of the same protein, to detect differences in protein dynamics.

Although efficient and effective, RMSF has some limitations in detecting statistically significant dynamics in a MD simulation. RMSF is attractive in its simplicity, in that it provides a single number per residue, reflecting the average fluctuations from an average position. There is some ambiguity in the method for determining the average position, particularly when comparing two or more different simulations, and an appropriate structural superposition is crucial. Furthermore, relevant information can be lost in the

averaging process. The RMSF does not provide details on the distribution of movement around the average position. Perhaps most importantly, however, is the difficulty in discerning between differences in dynamics of functional relevance to those of random fluctuations caused by under sampling.

An alternative to comparing RMSF values is to instead calculate differences in distributions between proteins or simulations. If the exact distribution of the distances from the average position of a residue were known, the impact of fluctuations from a single random sample would be minimized and the noise reduced. Of course, the reality is that these distributions are not known and must be estimated from a single sample. The PDF Estimator introduced in Chapter 3 has been shown to be a valuable tool for this application. As already demonstrated, this approach has advantages over other density estimation methods in that it will not tend to over-fit to a single sample, but instead will produce sample-size appropriate estimates. Additionally, the construction of an analytical solution allows for a powerful means of comparing distributions.

5.1 Probability Density Analysis Applied to Molecular Dynamics Trajectories

There are many known methods for comparing either two distributions or comparing a random sample to a single distribution. Three popular methods will be briefly introduced and investigated to test the comparison of MD trajectories. Additionally, several non-standard adaptations of existing metrics for distribution comparisons will be considered. In all cases, these measurements are applied to two sets of data per residue for comparison. For each sample set, a corresponding distribution is created using the PDF Estimator. These two distributions can then be compared with one another, or with the opposing data sample, to estimate the similarity between the two trajectories. For

demonstration of these techniques, simulations for three TEM-1 and TEM52 Beta-Lactamase wild type structures and corresponding mutations are used as examples: 1erm, 1htz, and 1li9. These initial 100ns simulations were provided by Matthew B. Tsilimigras.

Two commonly used methods for quantifying how two distributions diverge from one another are the Kullback-Liebler (KL) and Jensen-Shannon (JS) methods. The KL divergence is defined as

$$KL[q(z), p(z)] = \int ln\left[\frac{q(z)}{p(z)}\right] q(z)\, dz \tag{19}$$

and interpreted as the divergence of p from q. The probability density q is the expected distribution; therefore, the equation is biased towards q when comparing the two. A symmetrized measure, comparing two distributions equally when neither is known to be correct, is the Jensen-Shannon (JS) divergence, defined in terms of the KL divergence as

$$JS\big(q(z), p(z)\big) = \frac{KL[w(z), p(z)] + KL[q(z), q(z)]}{2}, \tag{20}$$

where *w(z)* is defined as the arithmetic average between *p(z)* and *q(z)*. As an alternate symmetrized KL adaptation, the geometric mean was also explored, as well as minimum and maximum boundaries, each defined respectively as

$$KLavg\big(q(z),p(z)\big) = \int \left| ln\left[\frac{p(z)}{q(z)}\right]\right| \sqrt{p(z)q(z)}\, dz, \tag{21}$$

$$KLmin\big(q(z),p(z)\big) = \int \left| ln\left[\frac{p(z)}{q(z)}\right]\right| \min[p(z),q(z)]\ dz, \tag{22}$$

and

$$KLmax\big(q(z),p(z)\big) = \int \left| ln\left[\frac{p(z)}{q(z)}\right]\right| \max[p(z),q(z)]\ dz. \tag{23}$$

The JS divergence and all versions of the KL divergence are measures of the differences between two distributions. Another commonly used metric, called the one sample Kolmogorov-Smirnov (KS) test, measures the probability that a random sample follows a given distribution. The KS test is defined as

$$KS = sup_z|F_n(z) - F(z)| \tag{24}$$

where $F_n(z)$ is the cumulative distribution function of the known distribution, $F(z)$ is the empirical distribution created from the data sample, and sup_z is the supremum function, measuring the greatest distance. Two adaptations of the KS metric are constructed for this application, which are referred to as KS1 and KS0.

Similar to KLavg, KS1 is the geometric average between two KS tests: the comparison between the distribution of data from one trajectory to the sample data of the

other, and vice versa. In this way, neither trajectory is biased over the other. KS0 uses the definition of KS to compare the two distributions directly. Specifically,

$$KS0 = sup_z|F1(z) - F2(z)|, \tag{25}$$

where F1 and F2 refer to the two estimated cumulative distribution functions. Neither KS1 nor KS0 are typical applications of the KS test, but additional insight into the differences between the distributions can be gained by these comparisons. In all versions of the KS test, the test statistic can be converted into a p-value, providing quantitative measures on the probability of the two samples representing the same distribution. Finally, one additional measure is calculated, which is simply the integrated sum of pointwise differences in the distribution functions as follows:

$$\Delta p\big(q(z), p(z)\big) = \int |p(z) - q(z)|\ dz. \tag{26}$$

For a qualitative comparison between these measurements, see Figure 26. The plot in Figure 26(a) shows each of the measurements, with the exception of KLmin and KLmax, as function of KS1. The key feature of this plot is the wide scatter of RMSF compared to the other methods of comparison, indicating that RMSF does not correlate strongly with any other method. Conversely, the integral sum and KS0 methods are strongly correlated with KS1, Figure 26(c), thus indicating they are somewhat redundant measures. JS, KS,

and KLavg, however, show a more interesting non-linear relationship, seen more clearly in Figure 26(d). KLavg is bounded by KLmax and KLmin, as expected, in Figure 26(b). Measurements that correlate well do not provide new quantitative information, but these figures demonstrate potentially useful orthogonality between RMSF, KLavg, KS1, and JS. These measurements will be used moving forward for the analysis of Beta-lactamase proteins, whereas redundant measurements will be dropped.

Of practical interest is the question of whether comparisons involving the distributions provide specific details about the differences in two trajectories that cannot be detected with the much simpler and easier to calculate RMSF difference. Figure 27(a) provides evidence of one example where comparing two distributions can highlight differences that are hidden from RMSF. Figure shows RMSF difference and KLavg for a small range of residues, comparing 1erm wild type to a single point mutation. Although the two measures show qualitatively similar information, the dotted black line at residue 197 indicates an area where KL shows a relatively large spike compared to RMSF. A visual explanation of this difference can be seen in Figure 27(b), comparing the distributions between the wild type and mutant at residue 197. In this case, the mutant has a slight bimodal nature not seen in the wild type. Although the deviations from the average, captured in a single value with RMSF, are similar, the highly sensitive KL value is detecting the fact that the distributions of these deviations are notably different.

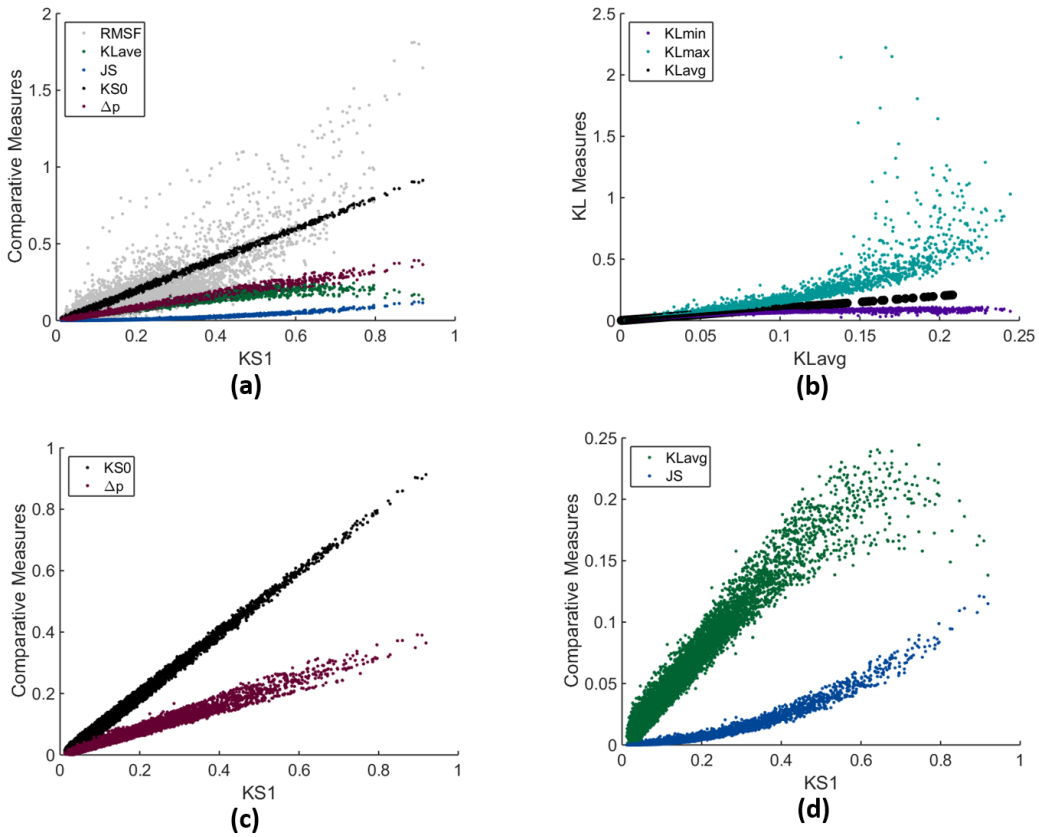

FIGURE 26: Correlations between distribution measurements

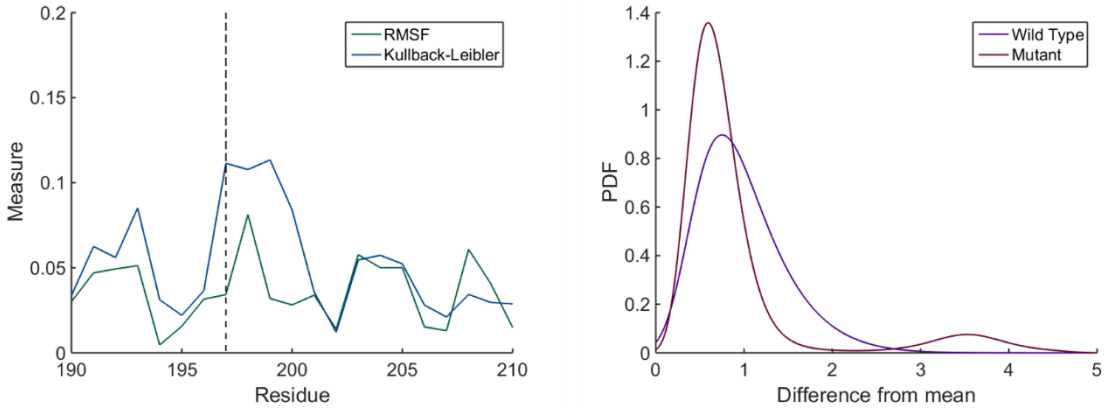

FIGURE 27: Comparison between RMSF and KL for a specific residue

Although these figures provide a valuable visualization of differences along trajectories per residue, it would be helpful to establish a means of defining when a residue or, even more importantly, an entire trajectory, diverges from another in a statistically significant way. Attempts to quantify this significance, using a limited set of trajectories, have been frustrated by the noise of fluctuations and the apparent lack of convergence. To demonstrate the convergence issues for the initial test case, Root Mean Square Deviations (RMSF) for all residues by frame are shown for wild type and mutant trajectories in Figures 28(a) and 28(b) for all three structures for a 100ns simulation. The corresponding distributions of the RMSF for each of these simulations are shown in Figures 28(c) and 28(d). The proteins were equilibrated prior to this production run, and the plots are arguably leveled off visually for the six trajectories, shown in yellow, orange, and blue. The 1erm mutation in figure 28(b) is a striking example of a simulation that appears somewhat converged, but then shows dramatically different behavior as the simulation continues. The fourth purple line, named 1erm2, represents the last 100ns of a 500ns simulation for comparison. Even with the longer 500ns simulation the timescale of biological relevance is not achieved. This conclusion is not atypical, and remains a significant challenge concerning MD analysis [145-147].

Aside from the convergence issues, the other challenge is to determine meaningful differences in residue distributions between two trajectories. The KS1 test defined in this section can be converted to a p-value, making this an ideal candidate for hypothesis

testing. To accomplish this, however, normal fluctuations from one simulation to the next must be benchmarked. Figures 29(a) and 29(c) show examples of

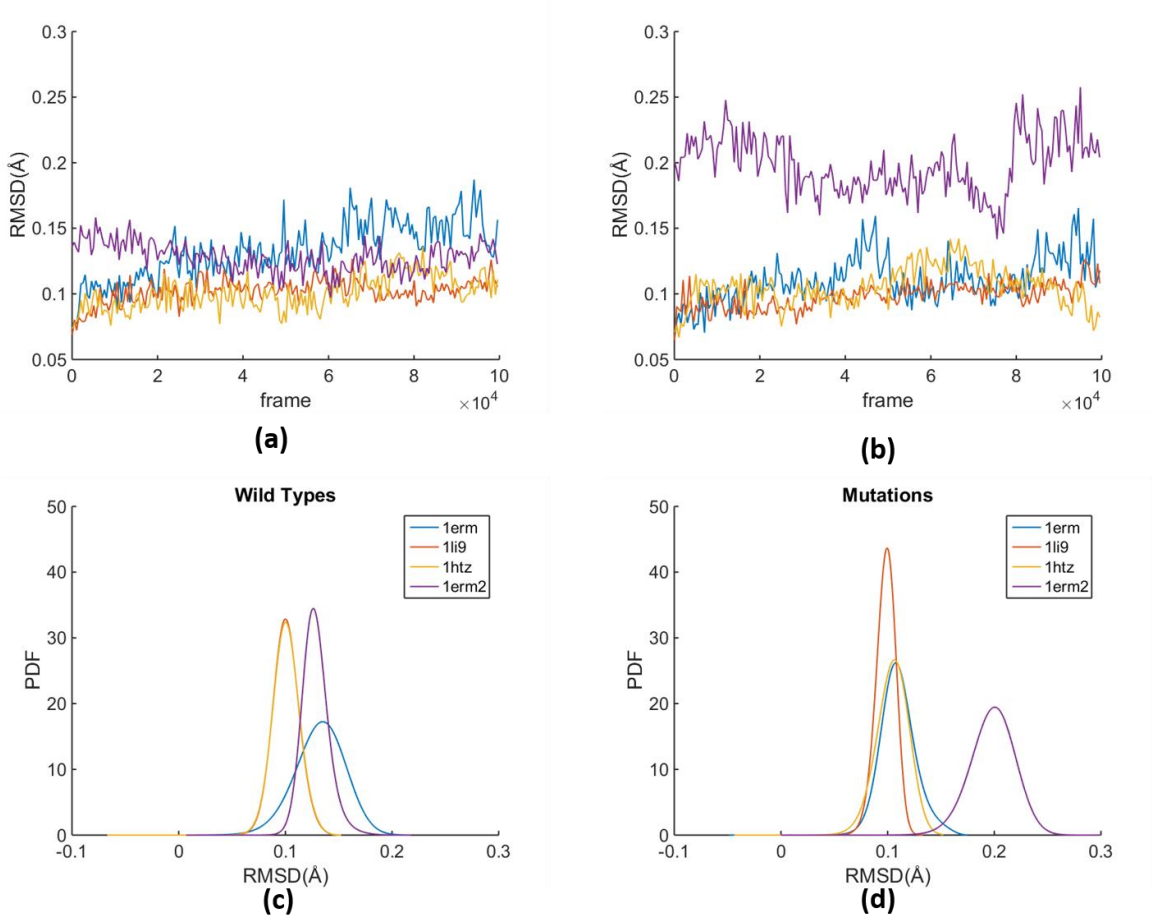

FIGURE 28: Convergence of wild type and mutations in beta-lactamase structures

comparing different wild type structures using RMSF and p-values, respectively, for each residue. There are some qualitative similarities between the measurements, but the p-

values are disproportionately extremely small, indicating a high level of confidence that the simulations for the structures are very different.

For a clear example, see the histogram of p-values in Figure 29(b). All trajectories were first split in two halves in order of simulation time and compared to one another according to the KS1 definition. These test statistics were converted into p-values for each residue and plotted as a histogram for all self-comparisons. The result was a collection of p-values indicating high confidence in rejecting the null hypothesis that these sets of data represent the same distribution. As a control, the same trajectories were then shuffled randomly, and the test repeated. In this case, the resulting p-values failed to reject the null hypothesis, concluding there was no difference in the shuffled trajectories. This, again, provides strong evidence for lack of convergence in the simulations. The final figure, 29(d), plots the average KL values as the blue curve, per residue, for self-comparisons of the first 100ns between the wild type and mutant of 1erm. The orange curve represents the averages, per residue, of all four cross-comparison between the 2 halves of each simulation. The visual difference between self-comparisons and cross-comparisons does not appear strikingly significant. The results of these tests on a very limited data set suggest that the statistical significance of mutation differences is inconclusive, other than to note that the simulations appear not to be converged on these timescales.

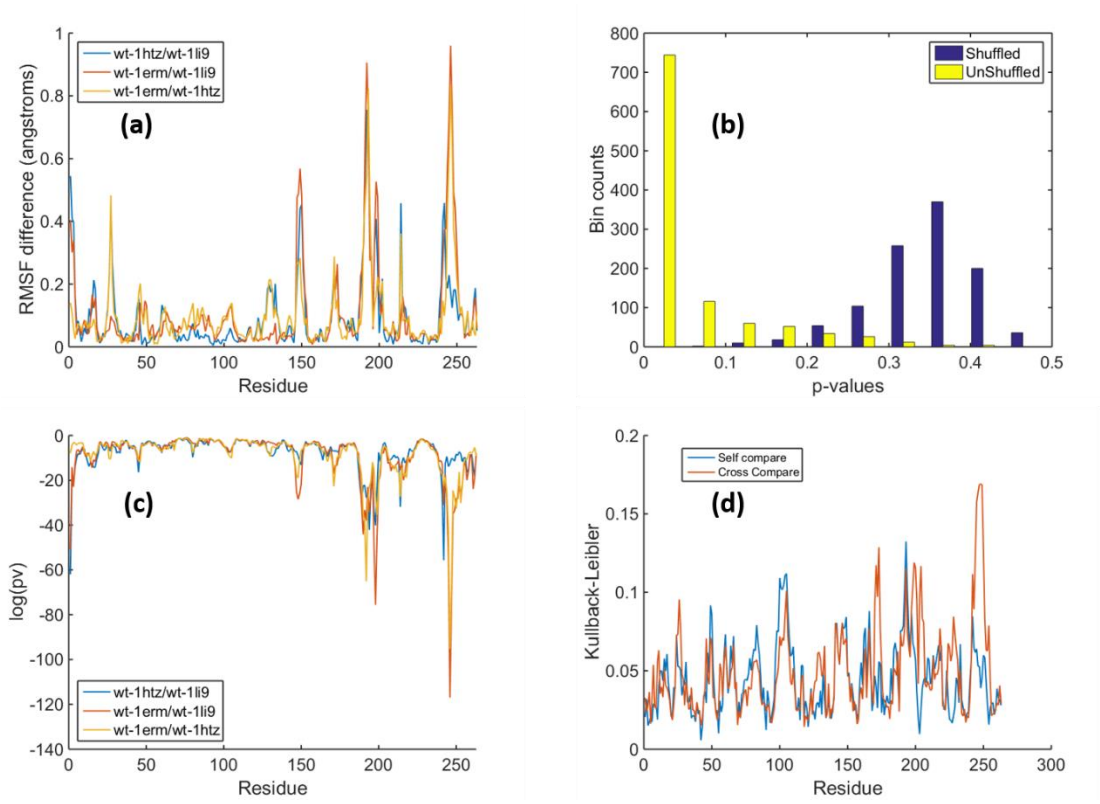

FIGURE 29: Statistical significance of residue fluctuations using distributions

5.2 Principal Component Analysis

One obvious solution for improving convergence is to run longer simulations. However, as Figures 28(a) and (b) demonstrate, there is no guarantee that a longer simulation will reach convergence. Nor is it ever possible to know for certain what may occur just beyond the current simulation time. An alternate explanation for the apparent lack of convergence is the presence of irrelevant noise caused by random fluctuations. Although these small fluctuations may continue to change as the simulation progresses, they do not necessarily represent important functional dynamics. A common approach to filtering out this random noise is through principal component analysis (PCA) [148, 149].

A full description of the method will not be discussed here but, briefly, PCA reduces dimensionality of a system through the diagonalization of a covariance matrix. The statistical technique essentially extracts the collective motions with the highest variability, thus is often referred to as *essential dynamics*. For proteins specifically, the number of degrees of freedom is at least equal to the number of residues, represented by the motions of each carbon alpha atom. As was readily seen in examples with beta-lactamase proteins in the previous section, the large number of degrees of freedom for even a moderately sized protein becomes intractable for statistical analysis. PCA can reduce this dimensionality from 263 residues to just a few components representing the largest sets of correlated motions. The work presented here combines the well-established methods of PCA with the PDF Estimator using the statistical measurements developed in the previous section, demonstrating that the noise can be significantly reduced.

Beta-lactamase proteins remain the focus of this study, with additional structures included for comparison. Furthermore, all trajectories were run for 500ns simulations instead of 100ns, to increase the likelihood of convergence. MD trajectories for this data, as well as PCA analyses, were provided by Chris Avery and the test data is summarized in Table 2. The beta-lactamase TEM1 protein is three residue mutations away from TEM52, and four mutations away from TEM30. There are six representative crystal structures for TEM1, and one each for TEM52 and TEM30. Mutations are all performed computationally and minimized prior to simulation.

TABLE 2: Beta-lactamase proteins simulated for test data

PDB code	Sequence	Protein	Point mutations from Tem1

1erm	Wild type	Tem1	
1erm	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
1ero	Wild type	Tem1	
1ero	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
1erq	Wild type	Tem1	
1erq	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
1xpb	Wild type	Tem1	
1xpb	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
1jwp	Wild type	Tem1	
1jwp	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
3jyi	Wild type	Tem1	
3jyi	Mutation	Tem52	(GLU079LYS) (MET157THR) (GLY213SER)
1htz	Wild type	Tem52	(LYS079GLU) (THR157MET) (SER213GLY)
1htz	Mutation	Tem1	
1lhy	Wild type	Tem30	(ARG244SER)
1lyh	Mutation	Tem1	(ARG244SER) (GLU079LYS) (MET157THR) (GLY213SER)

In ordinary PCA, the collective motions are expressed as eigenvectors that are sort ordered such that the largest variations are represented by the first eigenvectors. A subset of eigenvectors is chosen which, cumulatively, represent a significant reduction in degrees of freedom, while still maintaining a high percentage of the original data. The assumption, although not necessarily accurate, is that the largest motions will represent the functioning essence of the protein dynamics. It is important to remember that PCA will accurately extract the largest correlated conformational fluctuations, which correspond to slow motions, but there is no guarantee that these large-scale motions correspond to biological function. However, as this is often the case, PCA is often successful in extracting relevant information.

Addition of increasing eigenvectors typically follows a law of diminishing returns, and is often visualized with a scree plot, as in Figure 30(a). Scree plots are a valuable tool for choosing an appropriate subspace to work with. Oftentimes, only the first two

eigenvectors are evaluated, reducing the problem to only the very most significant motions. To quickly assess the differences between two proteins, a two-dimensional plot of the first two eigenvectors can provide a visual interpretation of the essential differences. Figure 30(b) shows an example of such a plot for the 1erm structure. Not only are the motions clustered differently between TEM1 and TEM52, but there are several distinct clusters from within the same trajectories. Figures 30(c) and (d) show the associated distributions for eigenvectors 1 and 2, respectively. The solid lines represent the first half of the trajectory and the dotted lines represent the second half. The differences in the scatter plot for Figure 30(b) match the separated peaks in Figures 30(c) and (d), but the distributions will provide a more convenient means of assessing the statistical differences.

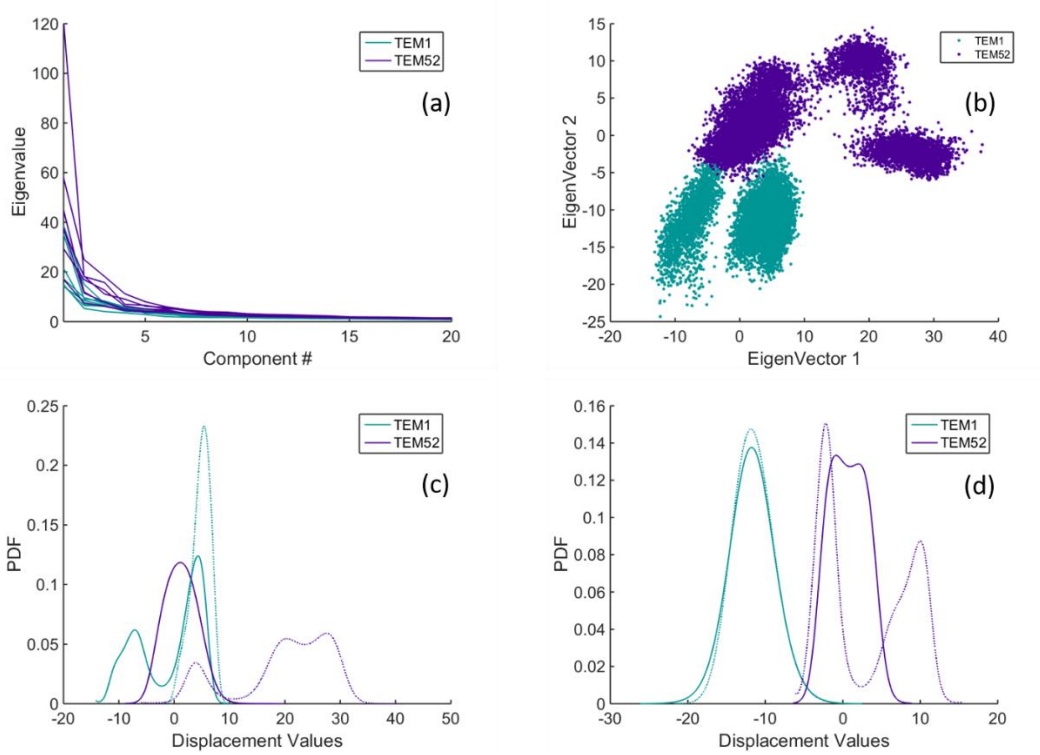

FIGURE 30: Statistical differences in 1ERM mutations for PCA

Unfortunately, traditional PCA analysis has not been able to settle the convergence/noise issues with these trajectories that were seen with the distributions per residue in the previous section. Highly significant differences are seen between wild types and mutations for all eight beta-lactamase structures. However, as can be determined easily by eye in the example shown in Figure 30, the first and second halves of each trajectory are also very different. As with the previous analysis, differences within a single trajectory are as significant as those between structures. Figure 31 more clearly demonstrates the problem. The average p-values were calculated, based on the KS test, for comparisons between wild type and mutated sequence distributions for all eight structures listed in Table 2 and plotted on a log scale including 60 eigenvectors. The cross comparisons are shown in cyan, and the self-comparisons are in black. Even with five times the simulation time, the PCA method is unable to distinguish between sequence mutations and the random fluctuations within a single trajectory.

As a further effort to isolate important differences in mutated beta-lactamase protein motions, more innovative PCA methods have been considered. The method yielding the most success to date is called displacement PCA. Traditional PCA is based on the variations of cartesian coordinates from the crystal structure, typically using the carbon-alpha atom as the reference point for a residue. A critical step in this process is structural superposition of the trajectories onto a common coordinate system. If two frames are misaligned, PCA will determine that their coordinates differ even if they are structurally identical. In displacement PCA, rather than correlating the positions, the displacements of a residues from one timeframe to the next are measured. This is analogous to measuring changes in velocity instead of changes in position.

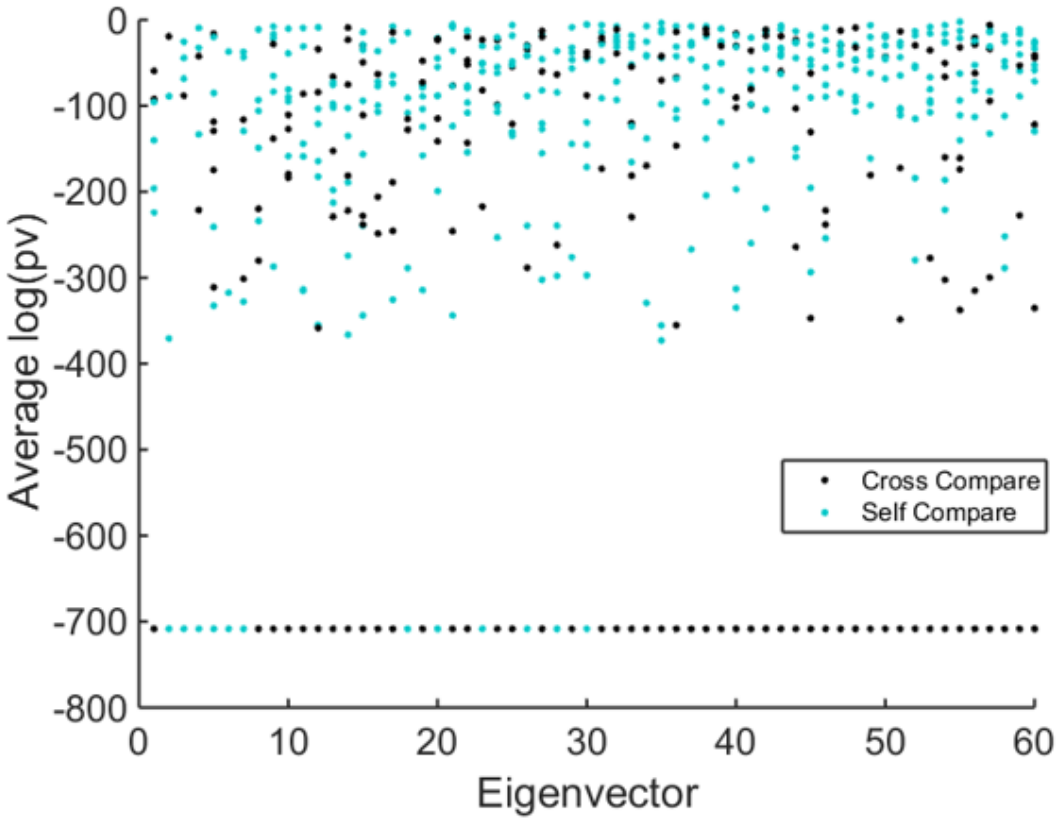

FIGURE 31: Comparison of p-values for all structures

The results from displacement PCA are dramatically different, as shown in Figure 31. First, the scree plot shows a smoother, more gradual significance in eigenvector coverage for all structures. This may be interpreted as being less sensitive to sudden random fluctuations in position. Although more eigenvectors are required to describe the same amount of data, it is likely that this provides a more realistic representation of the important motions rather than just noisy fluctuations. Figure 32(b) depicts the first two eigenvectors, forming single well-defined clusters, with a clear distinction in the mutated structure. More importantly, however, are the distributions of the first two eigenvectors, shown in Figures 32(c) and (d). Unlike the traditional coordinate PCA, the displacement

PCA produces highly similar self-comparisons, indicating the differences in the mutated and wild type distributions are true differences in the dynamics between the two structures.

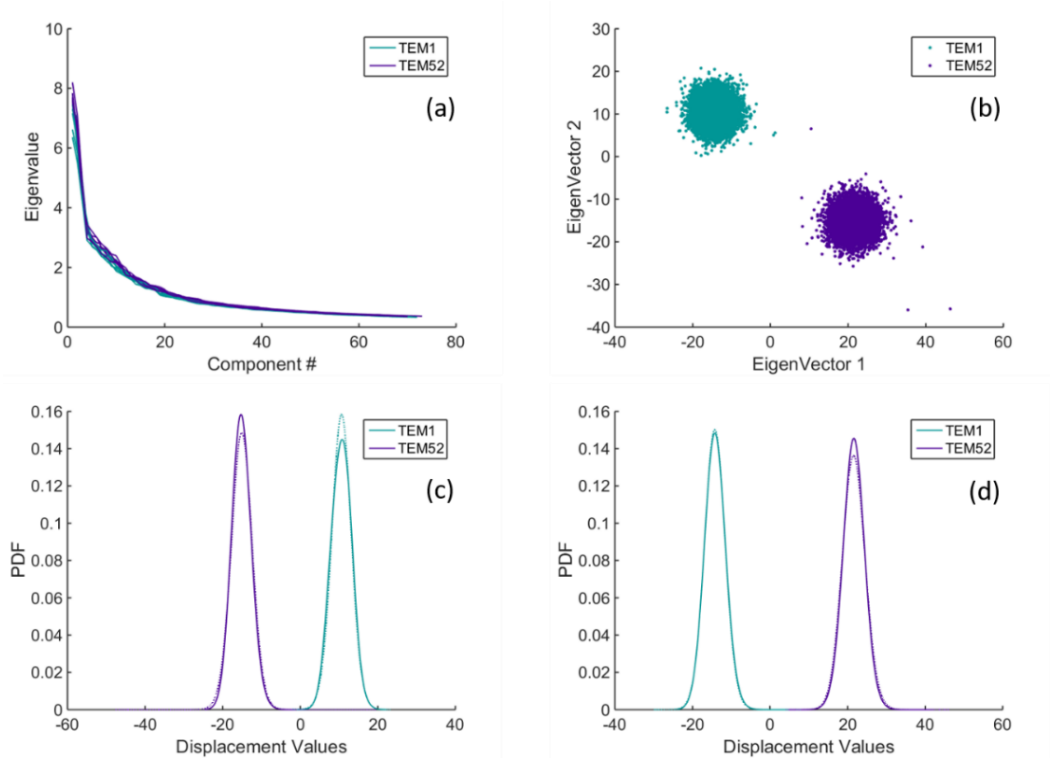

FIGURE 32: Statistical differences in 1ERM mutations for displacement PCA

When all eight structures and 60 eigenvectors are considered, a stark contrast is seen between the two methods. Comparing Figure 31 with Figure 33(a) reveals the significance of this difference. With the exception of three outliers in black, the distributions for eigenvector in the wild type differ from the mutation with the highest possible significance. The self-comparisons, however, indicate that the distributions of motions throughout a single trajectory over time are highly uniform. Figure 33(b) shows only the self-comparisons without the logarithmic scaling. The horizontal vertical line is

the threshold of significance at the 5% level. Points above this line are those that clearly fail to reject the null hypothesis that the motions originate from the same distribution. Although more points fall below the line than above, indicating either some residual noise or lack of convergence, this is not unexpected with any finite MD simulation. The important point is the contrast between self-comparisons and cross-comparisons in Figure 32(a).

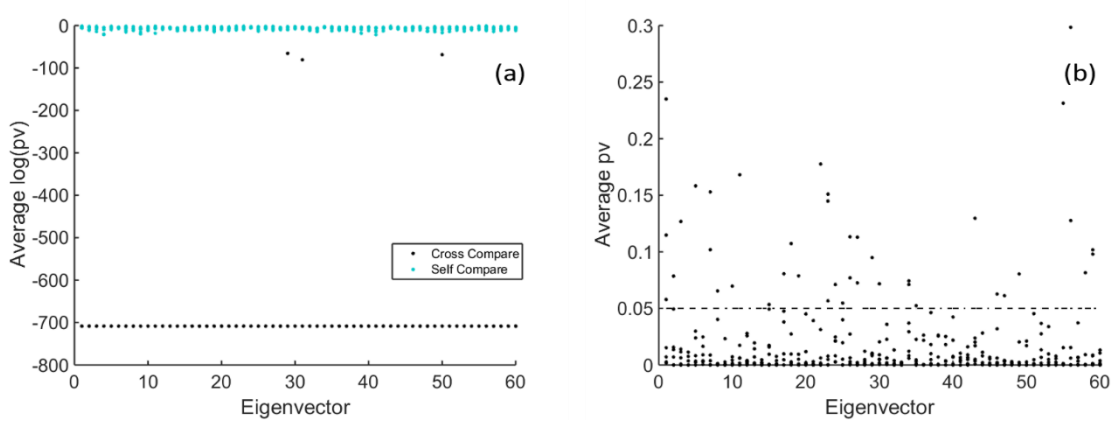

FIGURE 33: P-value comparison for displacement PCA

In summary, with the combination of a novel application of PCA, a high throughput density estimation implementation, and benchmarked statistical measures to quantify differences between distributions, the problems revealed in the previous section have been solved, or at least substantially mitigated. Using this strategy, the known functional differences between mutations of beta-lactamase proteins are clearly demonstrated and quantified in terms of significantly different dynamics. Most importantly, these differences are measured against a control by incorporating a self-comparison component into this procedure. The primary goal for developing this methodology was for the study

of beta-lactamase proteins. However, the technique has been successfully established and generalized, such that it can be applied to any future study involving protein dynamics.

CHAPTER 6: PACKING AND SOLVATION

One of the major design goals for the new DCM using the FAST library is to create a residue-specific set of entropy parameters based on local packing. Specifically, the entropy for a given residue would ideally be a function only of the surrounding quantity of microvoid and boundary solvent. The first step in parameterizing entropy is to evaluate the distributions of void space surrounding each residue type in native crystal structures. This has been accomplished by an automated process of running the PVA against the 108 proteins in the original test data set to calculate the partial void volumes for all residues, then invoking the PDF Estimator to find distributions for each residue type across all structures. By evaluating the location relative to the protein surface of each residue, local packing trends have been established. Before discussing these results, it is useful to briefly consider other approaches for defining and calculating packing, and the challenges involved.

A survey of the literature shows that a variety of methods that calculate local packing density yield inconsistent results, although most methods report at least somewhat higher packing in the core [61, 84, 97, 150-152]. The nature of packing density remains an open problem, in large part due to operational definitions. Conceptually, packing density is a measure of the percentage of the protein volume that is comprised of the van der Waals volume, not including the void volume. Richards originally estimated a mean protein core packing density of near 0.75, and others have verified this estimate across most globular proteins [59, 96, 153]. However, these calculations are critically dependent

upon the definition of the van der Waals radii. In one recent study focused on a careful analysis of appropriately modeling the hydrogen atoms, the packing density restricted to the core of globular proteins was found to be 0.56 [154, 155], substantially lower than expected. These unexpected results highlight the sensitivity of model parameters and variation in definitions. This work uses the Bondi radii[103], and all hydrogen atoms are included explicitly in the calculations.

Voronoi tessellation is the most common method for computing packing density because a local Voronoi cell will quantify the volume immediately surrounding each atom, analogous to the partial volume assigned to each atom in the PVA. Unfortunately, the Voronoi method has difficulties near the surface of a protein when the solvent is not modeled explicitly. In past works on packing density these types of technical problems were avoided by limiting analysis to buried residues within the protein core. Further attempts to overcome limited applicability to surface residues have been handled in a variety of ways, such as imposing boundary conditions to model a solvation shell surrounding the protein [59, 156], excluding certain intractable surface volumes from the calculations [61, 157, 158], or strategically placing water molecules around the protein [159, 160].

A different approach, called occluded surface packing (OSP) [97], computes packing density by extending lines from each atom perpendicular to its atomic surface, until the lines either intersect with another atomic surface or reach a length equal to the diameter of a water molecule. The lengths of these lines are used to determine the packing density. On average, OSP values are considerably smaller than typical Voronoi packing estimates [97]. This difference arises because OSP includes a boundary layer of solvent

surrounding the protein atoms, which lowers the relative density of the protein atoms near interfaces. Yet another approach, by Liang *et. al.*, is to apply a separate definition for surface packing density that considers pockets along the protein surface [59].

Regardless of the method of density calculation, For Voronoi methods, the packing density is generally defined as

$$\frac{VDW\ Volume}{Voronoi\ Volume}, \tag{27}$$

where the Voronoi volume encompasses the van der Waals volume of each atom as well as the void space in immediate proximity. Again, the details of the model can critically affect the accuracy of the density calculations. Another recent study [150] improved upon previous calculations in two important ways. First, water molecules were added explicitly by running a molecular dynamics simulation, thus creating a realistic boundary for the protein surface residues. Second, the Voronoi method was improved through empirically derived weighting parameters to partition space between atoms. Although weighting the atoms unevenly introduces small errors in the Voronoi method, these errors are generally considered to be of little significance, whereas the weighting is critical to a realistic tessellation when dealing with the range of atom sizes found in proteins [60, 150] . With these improvements, residues buried in the core of the protein are found to be approximately the same volume as those on the surface. This result is counter to other research that suggests proteins are packed more densely in the core [59, 84, 97].

Considering this recent work, local packing density will be characterized in terms of microvoid and boundary volume characteristics and elucidate why variations in local

packing occur. Partial volume calculations provide a means for studying the intrinsic packing densities of core residues and surface residues separately. The explicit calculation of microvoid as a separate quantity allows for a consistent definition of packing that ignores the existence of solvent molecules altogether. The following definition of packing density is expressed in terms of previously defined partial volumes.

$$packing\ density = \frac{vdW}{vdW + microvoid + cavity}. \tag{28}$$

Although boundary volume is not included in the packing density formula here, the partial boundary volume is used to rank order how deeply buried a residue is within the protein according the following criteria.

$$fraction\ buried = \frac{microvoid}{microvoid + boundary}. \tag{29}$$

For completely buried residues with no associated boundary volume the fraction will be 1, whereas residues very near the surface with have a fraction approaching 0, as the boundary will dominate the partial volume. All residues are ranked for each protein and split into two equal groups for comparison.

Figure 34 shows distributions for the proteins in the dataset for a probe radius of 1.4Å, a grid size of 0.5Å, and Ls set to 5.6Å, according to this equation, packing distribution among core residues agrees well with Voronoi and OSP estimates. The packing of residues near the surface, however, is slightly higher (c.f. Figure 11a),

contrary to the predictions of many other methods [97, 158]. OSP surface packing densities, are predicted as much lower than core densities, with ranges as low as 0.2 to 0.4. Voronoi method estimates vary depending on how the surface is bounded, but generally find the surface residues to be slightly less packed. To gain insight into the reason for these differences, alternative definitions including boundary volume are considered.

As an opposing extreme to the solvent-excluded density calculation, consider the following definition of packing density,

$$packing\ density = \frac{vdW}{vdW + microvoid + cavity + boundary}, \qquad (30)$$

shown in Figure 34(b). The most buried residues have the same average packing density, but a dramatic shift occurs in the residues near the surface when boundary volume is included. This is unsurprising, due to the 5.6Å layer of boundary surrounding the protein. The dashed line in Figure 34(b) adheres to the definition of Equation 30 as well, but with slightly different parameters. For comparison with OSP, Ls was set to 2.8Å and the distribution shown includes all residues in which each atom has some exposure to solvent. This set of criteria produces surface packing densities in general agreement with OSP calculations, which extend out from the surface to the estimated diameter of a water molecule.

Equation 30 and Figure 34(b) imply an implicit solvent model, where this boundary layer is included in the packing calculation. If water is included explicitly, either as a solvation layer or throughout a simulation box based on molecular dynamics simulation,

the water molecules themselves would not be considered void space, and therefore not be a part of the protein packing density. The question then becomes whether Equation 30 can be modified to account for the microvoid in the boundary layer that should be assigned as partial volume to the surface residues. An empirical approach to this question is to consider a third definition for packing, as follows.

$$packing\ density = \frac{VDW}{VDW + microvoid + cavity + (boundary < cutoff)}. \quad (31)$$

In this calculation, only those boundary grid points within some cutoff distance from the closest atom are counted, as a means of modelling only the microvoid and not the solvent atom. Figure 34(c) demonstrates the relative densities of atoms for a probe radius of 1.4Å, and a cutoff distance of 0.35Å. These parameters result in very similar distributions with equal average densities, such as has been demonstrated with molecular dynamics simulations with a finely-tuned Voronoi tessellation [150].

It is also interesting to plot the cutoff value that produces uniform densities, as a function of probe radius (Figure 34(d)). The peak cutoff values occur very near, and slightly above, the approximate size of a water molecule. The lower cutoff values for smaller probe sizes is somewhat intuitive, as less microvoid is expected for smaller probes. The decreasing cutoff for higher probes is less obvious but suggests a geometric relationship between the relative sizes of the protein residues and the solvent molecules at the surface. Additional simulations were performed with a different definition of vdW radii, as well as fixed atom radii, and similarly-shaped cutoff curves were found in all

cases, albeit with different peak radii. This investigation supports the idea that this cutoff value is a function of geometry, the details dependent upon the model parameters.

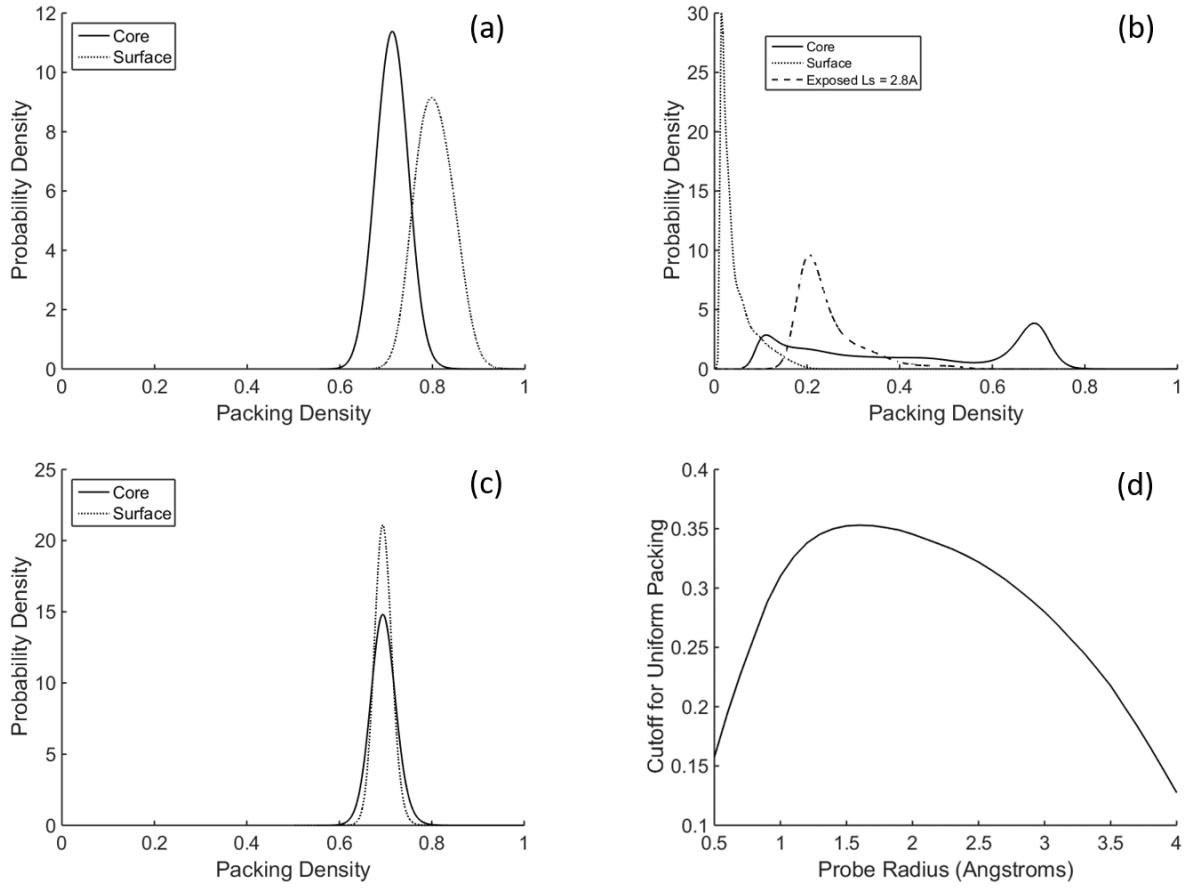

FIGURE 34: Packing density distributions

Related to packing density, Figure 35 shows the relative distributions for van der Waals volumes and microvoid volumes based on their location within the protein. They are both normalized by the average recorded volume per residue, thus ensuring the range of residue volumes are equitably considered. It is important to note that the partial protein volume is different from either molecular volume or van der Waals volume. For bonded or tightly packed atoms, a single grid point may be within the van der Waals

radius of more than one atom but will be only assigned once. Therefore, the distributions in Figure 35(a) are a measurement of the compressibility of the protein, which is shown to be very slightly higher for buried residues. The microvoid volume in Figure 35(b), however, can be much more clearly separated between core and surface residues. Recognizing that some of the microvoid near the surface would be assigned towards solvent molecules if the solvent were modeled explicitly, microvoid along the boundary is somewhat inflated. This is precisely the quantity that is counter-balanced by including a small amount of boundary volume in Equation 31. A larger dataset would be necessary to increase the statistics to determine if these trends hold, but this analysis demonstrates the advantage of the grid-based partial volume method in discerning precise volumes in space.

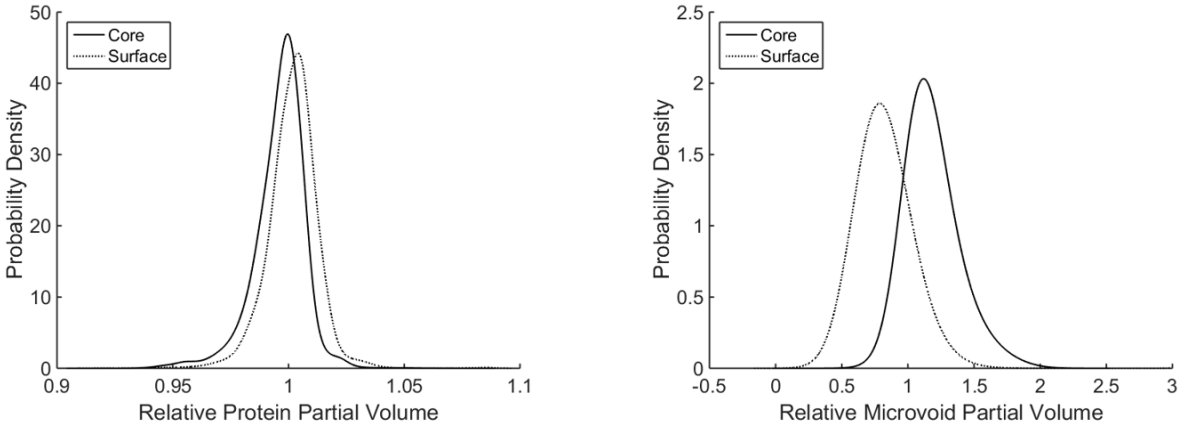

FIGURE 35: Normalized partial volumes distributions for protein and microvoid.

The protein length is another factor affecting packing density that has been demonstrated numerous times in the literature [59, 97, 154, 161]. Interestingly, there remains disagreement as to the nature of this dependence. Applying Equation 28 to the

average total protein volumes for all structures in our data set, packing density is found to be smaller for larger proteins (Figure 36), in agreement with densities reported using Voronoi methods. If the definition includes the boundary volume as in Equation 30, the length dependence reverses, in agreement with OSP. Finally, by applying Equation 31 with a cutoff of 0.35Å, uniform packing density is found across all protein lengths. This difference can be explained by considering the relative increased impact of surface area, where solvent is most prevalent, for small proteins. The packing densities in Figure 36 are averages calculated across 300 rotations per protein using a probe size of 1.4A. All comparisons are qualitative, as exact results are dependent upon the test data set (ie, protein length) and the van der Waals radii for each protein atom.

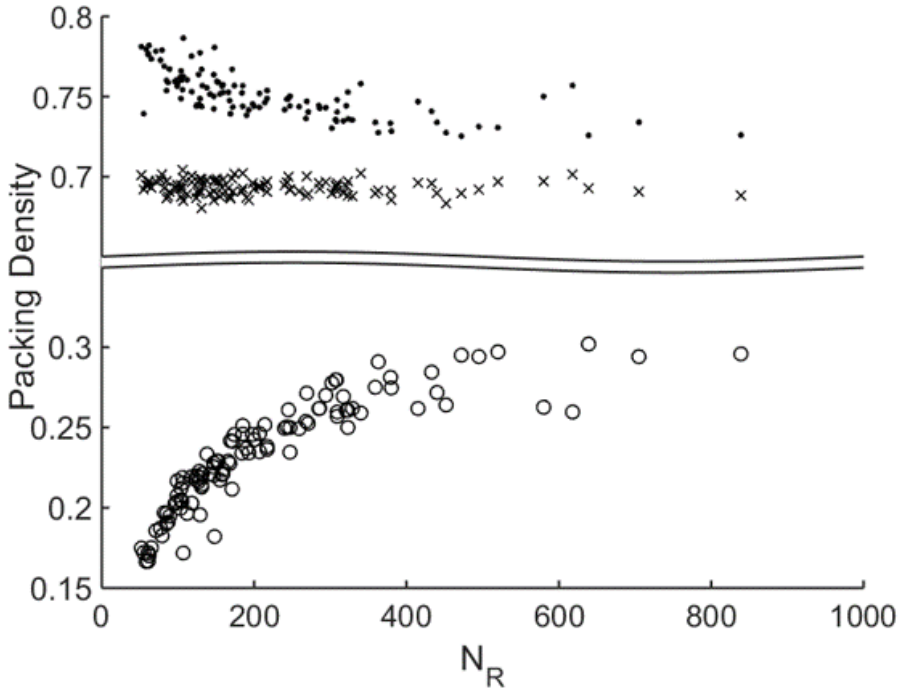

FIGURE 36: Packing densities as a function of number of protein residues

The power of the partial volume calculations in the PVA is that all volume space is mapped out explicitly, providing accuracy and flexibility. This chapter has demonstrated agreement with prominent and conflicting results in the literature, showing that the differences are caused by the ambiguity in the definition of packing density. The parameterization of the cutoff parameter in Equation 31 can be tuned to mimic time-consuming MD simulations extremely quickly. The PVA therefore provides detailed volume information that allows packing to be evaluated according to changing criteria for a better understanding of how void space is distributed.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1 Summary of Conclusions

The PDF Estimator and the Protein Void Analyzer have been independently designed, implemented, and applied to a variety of applications in the field of structural bioinformatics. Both applications demonstrate novel approaches to existing problems and possess unique benefits over alternative solutions. The PDF Estimator out-performs the standard KDE method for many known distributions and does so without requiring advanced user techniques or expertise. The method is computationally efficient and has been optimized in C++ for performance competitive with other density estimators. The code is written in a modern object-oriented style, allowing for customized flexibility while maintaining a single class library. The PDF Estimator has been distributed as a stand-alone product upon request to researchers outside the group and used extensively within the Bio-Molecular Physics Group (BMPG) for many projects. The class library has also been integrated into the popular statistics software, R, and submitted to CRAN as a downloadable package for R users. For a more platform-independent option, an identical Java code is also maintained and distributed.

The Protein Void Analyzer, also written as a C++ class library, is a highly optimized, memory-efficient method for calculating protein volume based on the Hoshen-Kopelman algorithm. In addition to efficiency, several key features of the PVA set it apart from other volume calculation methods. First, the method is easily customizable across a range

of parameters, including van der Waals definitions, boundary layer, probe size, and resolution, allowing seamless comparison to other published results. Second, the averaging over imposed systematic fluctuations of the probe size coupled with grid orientations represents the inherent dynamic nature of proteins as they naturally function. Additionally, the spring model employed to distinguish void types intentionally introduces further ambiguity in the precise positions and boundaries of the protein atoms. The combination of these features models the dynamic motion of a protein without the computational cost of a molecular dynamics simulation.

A third important feature of the PVA, not found in other programs, is the careful mapping of all types of protein volume, including the novel definition of microvoid as a separate calculation. Microvoid volumes are specifically important in the applications of percolation and packing density. The percolation of microvoid through a protein as a function of probe size has been shown to have approximately universal characteristics, consistent with three-dimension percolation experiments. The probability of percolation, which is not strictly dependent on dimensionality according to percolation theory, is the same across a diverse set of 108 globular proteins within a very narrow tolerance, suggesting common packing distributions for all proteins. Exceptions outside these tolerances are found in highly atypical proteins.

Microvoid distribution is also instrumental in the closely related analysis of packing density. Proteins are highly compact systems within their functional native state, and the nature and distribution of this packing has long been considered an important area of study in the field of structural biology. Quantitatively, density calculations are highly dependent on van der Waals radii and probe size. Qualitatively, these same calculations

are also dependent upon definitions of packing. Definitions vary according to how solvation is considered, but an exact calculation requires an explicit solvation model with a molecular dynamics simulation. In most models, including the PVA, solvation is implicit and therefore inexact. However, the precise partial volume calculations in the all-atom model allow the PVA to be parameterized according to the results of careful simulations of explicit solvent. This parameterization has been completed for a probe size of 1.4Å, such that a single cutoff value can model the solvent without the time investment of a long simulation.

Despite the high cost, molecular dynamics remains the most effective way to study protein dynamics. MD is often employed to demonstrate markedly different dynamics between small mutations that control functionality. Two ongoing challenges in the field of molecular dynamics are convergence and statistical significance. In the former, it is extremely difficult, in fact likely impossible, to determine if a protein system has been equilibrated into the low energy state. In the latter, methods are needed to quantify the important differences between two simulations. Both challenges are frustrated by the presence of noise within the fluctuations.

The PDF Estimator, together with principal component analysis, are applied towards both problems in new ways. The class library comprising the PDF Estimator was incorporated into a tool to analyze a pair of trajectories by comparing distributions at a residue level. Distributions were compared using a variety of known statistical methods and applied to beta-lactamase proteins as a test case. These attempts demonstrated that the statistical difference between mutations was indistinguishable from differences within a single trajectory, indicating the simulations may not have reached convergence. A

similar analysis using traditional PCA to isolate important large-scale dynamics confirmed these results. However, a newly developed variation of PCA, called displacement PCA, captured the essential dynamics of the differences in mutations, while reducing the noise characterized by irrelevant fluctuations.

7.2 Peer Reviewed Publications

- Farmer, J., Fareeha Kanwal, Nikita Nikulsin, Matthew C. B. Tsilimigras, and Donald J. Jacobs (2017). "**Statistical Measures to Quantify Similarity between Molecular Dynamics Simulation Trajectories**". entropy **19**(12): 646.
- Farmer, J. and D. Jacobs (2018). "**High throughput nonparametric probability density estimation**". PloS one **13**(5): e0196937.
- Farmer, J., S. Green, and D. Jacobs (2018). "**Distribution of volume, microvoid percolation, and packing density in globular proteins**". (Submitted October 22nd to Physics Review E.)
- Avery, C., J. Farmer, M. Tsilimigras, C. David, D. Livesay, D. Jacobs (2018). "**Characterizing dynamical differences between TEM-1 and TEM-52 beta-lactamases**" (Manuscript in preparation, planned submission to Proteins, December, 2018).

7.3 Additional Contributions

- Green, Sheridan B., Jenny Farmer, and Donald J. Jacobs . "**Universal Scaling of Cavity Volume Pathways in Globular Proteins**". Biophysical Society Annual Meeting, Los Angeles, CA, February 7-11-2016. Presented by Jenny Farmer

- Green, Sheridan B., Jenny Farmer, and Donald J. Jacobs . "**Analysis of Cavity Volumes in Proteins Using Percolation Theory**". American Physical Society, March Meeting, Baltimore, MD, March 9-14-2016. Presented by Sheridan Green.
- Farmer, J. and D. J. Jacobs (2016). "**Nonparametric Maximum Entropy Probability Density Estimation**". arXiv.org: 1606.08861.
- Farmer, J. and D. Jacobs (2018). **PDF Estimator as an integrated package for R statistical software**. Submitted to CRAN October 2018. PDFEstimator::estimatePDF()
- Farmer, J. Sheridan Green, and Donald Jacobs (2016). "**Distribution of volume, microvoid percolation, and packing density in globular proteins**". arXiv.org: 1810.08745
- Avery, C., J. Farmer, M. Tsilimigras, C. David, D. Livesay, D. Jacobs (2019). "**Characterizing dynamical differences between TEM-1 and TEM-52 beta-lactamases**". Biophysical Society Annual Meeting, Baltimore, MD March 2-6, (2016). To be presented by Chris Avery.

7.4 Future and Ongoing Related Work

Tangential to the primary focus on methods and direct applications presented in this thesis, multiple relevant side-projects have been investigated by current and past members in the research lab. Additionally, discussions amongst the group have inspired exciting possibilities in terms of both extensions and improvements to the computational techniques, as well as new opportunities to apply them. This section aims to briefly outline some of the more promising avenues of research pursuit, both future and in progress.

7.4.1 Publications in Progress

- Rigorous comparison of PDF Estimator with other nonparametric methods, with Layton Hall, Layton, Micheal Grabchack, and Donald Jacobs. The comparative methods for this publication, currently in preparation, are kernel density and Akaike/Bayesian criteria. A series of bimodal distributions have been generated and estimated using existing methods, and a protocol of systematic comparison has been developed.
- PDF Estimator parallelization, with Zach Merino, Micheal Grabchack, and Donald Jacobs. This potential publication will include a much broader range of density estimators and distributions for comparison, along with piloted enhancements for parallelization of the PDF Estimator. The R-packaged plugin for the PDF Estimator will be instrumental in this research initially, and the new parallelized version will also be packaged and uploaded as an R plugin.

7.4.2 Enhancements to PVA

The following list is a brief summary, in order of importance, of a few significant improvements that can be made immediately to the PVA for improved performance, functionality, and usability.

- Publication of software method as it currently exists, alongside distributing PVA on public forum such as GitHub.
- Integrated surface area calculations through an extended application of the HK clustering algorithm.
- Spring model enhancements and optimizations for greater accuracy and performance.

- Modify code to handle periodic boundary conditions within a cubic box, applicable to any molecular system. The generalization to other nonprotein materials, including the features of percolation and altering probe radius, is needed for specific ongoing research projects and would have extensive applications in materials science. The required modifications would be relatively simple to accommodate.

7.4.3 Decoy Detection and Structure Prediction

The analysis of 108 proteins with the PVA has highlighted many characteristics that are very closely similar across this data set. Specifically, peak cavity volumes and percolation thresholds as a function of probe size, packing density distribution throughout the protein, and percolation scaling exponents. These results suggest such measurements may be common to most globular proteins and could provide a benchmark for successful structure prediction and protein design. If the basis for these similarities is a function simply of geometry and packing, this is interesting. However, if there is a biological basis dictating these traits, this may be of greater impact in detecting amino acid sequences capable of forming stable structures. If the PVA can distinguish between stable, functional proteins found in nature, and those that have been improperly constructed through protein structure prediction software, then this method could be used as a tool to filter out structures that are considered unviable.

Testing this hypothesis requires a much greater test set against many more known crystal structures to determine if the packing and percolation characteristics found thus far continue to hold for a large data set, and to quantify outliers and exceptions. A test data set has been created by Dr. Azhagiya Singam using a the decoy detection program

3D-Robot [162] containing thousands of know structures for the protein data bank with corresponding decoy structures containing improper folds with the same sequence. Running the PVA against both sets should provide sufficient test for comparison.

7.3.4 PCA Applied to Protein Volume Fluctuations

The same techniques from Chapter 5 can be applied to partial volume changes in an MD trajectory. A preliminary study was conducted by running the PVA against the frames of a 100ns trajectory in the beta-lactamase protein data set and performing PCA analysis on the results. Initially, this did not produce interesting results in terms of extracting meaningful or practical information from this data. However, at the time of this analysis, extensive MD simulation data was not available and only a single structure and its mutant were considered. Currently, 500ms trajectories for eight structures with corresponding mutant structures are available. A similar comparison using displacement PCA will be conducted for all simulations, instead analyzing changes in volume as calculated by the PVA. If results prove interesting, this will be published separately.

7.4.4. Hydrophobicity

The tendency for hydrophobic residues to aggregate in the core of a protein while hydrophilic residues form bonds with solvent on the surface, collectively called the hydrophobic effect, has long been considered the major driving force in protein folding, which occurs in previously unexplained timescales. Ever since this mechanism was proposed, there have been many increasingly refined experimental efforts to quantify this effect by defining a hydrophobicity scale [71, 151, 163-168]. These experiments attempt to measure the transfer free energy change of moving representative molecules from polar to non-polar environments. Computational statistical mechanics approaches for

determining each residues tendency towards water have also been developed by calculating the SASA of residues typical in folded proteins [95, 169-171].

In these computational methods, the transfer free energy from buried to exposed is described as follows:

$$transfer\ free\ energy = -RTln\left[\frac{p(b)}{p(s)}\right] \quad (20)$$

Where p(b) and p(s) are the probabilities for a given residue to be buried or exposed, respectively, and are empirically determined by calculating the solvent accessible surface area in crystal structures. The choice for determining whether a residue is buried is somewhat arbitrary but is often defined as residues with 5% or less surface area exposed. Several such scales have been defined, and correlate well with physical experiments, but subtle differences remain between all hydrophobicity scales, regardless of how they were derived (experimentally and/or computationally). Discrepancies may be due to insufficient sampling, local environmental variations, temperature dependence, and residue length dependence [95, 165, 171] .

Following the statistical mechanical approach for SASA, a similar method can be implemented for the partial volume calculations described in section 2.5.4. Table 1 is a preliminary example from the 120-protein test data set, ranking each residue type according to its associated partial boundary solvent. This table was generated by counting the boundary solvent grid points surrounding each residue for all 120 proteins, as explained in the previous section, and calculating the sum by residue type. The sums were then normalized by the total counts of each residue type found and listed in decreasing order. The result is a relative boundary volume scale that can be used as an alternative to relative accessible surface area.

TABLE 3: Preliminary hydrophobic tendencies based on partial volumes

Residue Type	**Calculated Hydrophobicity**		**Residue Type**	**Calculated Hydrophobicity**
Isoleucine	0.502972652		Threonine	0.193612774
Leucine	0.466221852		Tyrosine	0.192604006
Methionine	0.460264901		Serine	0.15426009
Valine	0.455882353		Proline	0.13018598
Phenylalanine	0.420979021		Asparagine	0.097643098
Cysteine	0.41091954		Aspartate	0.092870544
Alanine	0.331130205		Glutamine	0.072780204
Tryptophan	0.277777778		Arginine	0.057755776
Glycine	0.248666667		Glutamate	0.057432432
Histidine	0.195266272		Lysine	0.012727273

A recent review article on hydrophobicity scales reported correlations between five experimental scales and three different computational scales derived using SASA [95]. Table 2 reports the R^2 values comparing all the experimental scales with one another, and with each of the computational scales. Some correlate quite well, others very poorly. The column and row labeled PVA lists the correlations between all these scales, and the partial boundary volume values from Table 1, showing comparable R^2 values, suggesting that partial volumes are a good indicator of hydrophobicity. It is not known if the partial volume technique will produce a hydrophobicity scale of transfer free energies that is superior to that of SASA calculations, but there has been recent evidence arguing that surface area is not an ideal indicator of hydrophobic free energies [172]. Thus, this alternative method is worth investigation.

TABLE 4: Hydrophobicity scale comparison by pairwise correlation coefficients (R2)

	PVA	**EXP1**	**EXP2**	**EXP3**	**EXP4**	**EXP5**	**SASA1**	**SASA2**	**SASA3**
PVA		0.694	0.7767	0.7095	0.561	0.6861	0.3984	0.8823	0.8463
EXP1	0.694		0.8816	0.4664	0.3796	0.9388	0.6078	0.7597	0.5026
EXP2	0.7767	0.8816		0.7393	0.5955	0.9576	0.4397	0.8081	0.7071
EXP3	0.7095	0.4664	0.7393		0.9015	0.5419	0.2289	0.7233	0.8708
EXP4	0.561	0.3796	0.5955	0.9015		0.4092	0.2313	0.6276	0.7624
EXP5	0.6861	0.9388	0.9576	0.5419	0.4092		0.4704	0.7197	0.5448
SASA1	0.3984	0.6078	0.4397	0.2289	0.2313	0.4704		0.4828	0.2704
SASA2	0.8823	0.7597	0.8081	0.7233	0.6276	0.7197	0.4828		0.8733
SASA3	0.8463	0.5026	0.7071	0.8708	0.7624	0.5448	0.2704	0.8733	
average	**0.694**	**0.6538**	**0.7382**	**0.6477**	**0.5585**	**0.6585**	**0.3912**	**0.7346**	**0.6722**

7.4.5 FAST

The inspiration driving the development of the PDF Estimator and the PVA originated from the design goals for a next-generation DCM. The unanticipated applications for these programs, outside of this original vision, have proven to be valuable contributions on their own, and have spawned their own tangential research goals that are currently being actively pursued. However, the initial requirements for accurate density information and the mechanism for entropy parameterization based on microvoid have been completed and added to FAST library code, allowing for the next step in development. The design and research committed to the FAST library has continued as well and aspire towards even broader impact in the future.

REFERENCES

1. Hartley, H., *Origin of the Word 'Protein'.* Nature, 1951. **168**: p. 244.
2. Chick, H. and C.J. Martin, *On the heat coagulation of proteins : Part IV. The conditions controlling the agglutination of proteins already acted upon by hot water.* TJP The Journal of Physiology, 1912. **45**(4): p. 261-295.
3. Kendrew, J.C., et al., *A three-dimensional model of the myoglobin molecule obtained by x-ray analysis.* Nature, 1958. **181**(4610): p. 662-6.
4. Dill, K.A. and J.L. MacCallum, *The Protein-Folding Problem, 50 Years On.* Science, 2012. **338**(6110): p. 1042-1046.
5. Hosking, J.R.M., *Distributions with maximum entropy subject to constraints on their L-moments or expected order statistics.* Journal of Statistical Planning and Inference, 2007. **137**(9): p. 2870-2891.
6. Lo Conte, L., et al., *SCOP: a Structural Classification of Proteins database.* Nucleic Acids Research, 2000. **28**(1): p. 257-259.
7. Dawson, N.L., et al., *CATH: an expanded resource to predict protein function through structure and sequence.* Nucleic Acids Research, 2017. **45**(Database issue): p. D289-D295.
8. Jaki, T. and R.W. West, *Maximum kernel likelihood estimation.* Journal of Computational and Graphical Statistics, 2008. **17**(4): p. 976-993.
9. Rajpal, A., M.G. Taylor, and J.F. Kirsch, *Quantitative evaluation of the chicken lysozyme epitope in the HyHEL-10 Fab complex: Free energies and kinetics.* Protein Science, 1998. **7**: p. 1868-1874.
10. Bermudez, M., et al., *More than a look into a crystal ball: protein structure elucidation guided by molecular dynamics simulations.* Drug Discovery Today, 2016. **21**(11): p. 1799-1805.
11. McCallum, S.A., et al., *Ligand-induced changes in the structure and dynamics of a human class Mu glutathione S-transferase.* Biochemistry, 2000. **39**(25): p. 7343-56.
12. Zhang, M., T. Tanaka, and M. Ikura, *Calcium-induced conformational transition revealed by the solution structure of apo calmodulin.* Nature structural biology, 1995. **2**(9): p. 758-67.
13. Bertini, I., et al., *Experimentally exploring the conformational space sampled by domain reorientation in calmodulin.* Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(18): p. 6841-6.
14. Sayre, D., *X-Ray Crystallography: The Past and Present of the Phase Problem.* Structural Chemistry Structural Chemistry : Computational and Experimental Studies of Chemical and Biological Systems, 2002. **13**(1): p. 81-96.
15. Davis, A.M., S.J. Teague, and G.J. Kleywegt, *Application and Limitations of X-ray Crystallographic Data in Structure-Based Ligand and Drug Design.* Angewandte Chemie., 2003. **42**(24): p. 2718.
16. van den Bedem, H. and J.S. Fraser, *Integrative, dynamic structural biology at atomic resolution--it's about time.* Nature methods, 2015. **12**(4): p. 307-18.

17. Woldeyes, R.A., D.A. Sivak, and J.S. Fraser, *E pluribus unum, no more: from one crystal, many conformations.* Current Opinion in Structural Biology, 2014. **28**(Supplement C): p. 56-62.
18. Keedy, D.A., J.S. Fraser, and H. van den Bedem, *Exposing Hidden Alternative Backbone Conformations in X-ray Crystallography Using qFit.* PLoS computational biology, 2015. **11**(10).
19. Baldwin, A.J. and L.E. Kay, *NMR spectroscopy brings invisible protein states into focus.* Nature chemical biology, 2009. **5**(11): p. 808-14.
20. McCammon, J.A., B.R. Gelin, and M. Karplus, *Dynamics of folded proteins.* Nature, 1977. **267**(5612): p. 585-90.
21. Dror, R.O., et al., *Biomolecular Simulation: A Computational Microscope for Molecular Biology.* Annual Review of Biophysics, 2012. **41**.
22. Pan, A.C., et al., *Demonstrating an Order-of-Magnitude Sampling Enhancement in Molecular Dynamics Simulations of Complex Protein Systems.* Journal of Chemical Theory and Computation, 2016. **12**(3): p. 1360-1367.
23. Mortier, J., et al., *The impact of molecular dynamics on drug design: applications for the characterization of ligand-macromolecule complexes.* Drug discovery today, 2015. **20**(6): p. 686-702.
24. Zuckerman, D.M., *Equilibrium sampling in biomolecular simulations.* Annual review of biophysics, 2011. **40**: p. 41-62.
25. van Gunsteren, W.F. and A.E. Mark, *Validation of molecular dynamics simulation.* Journal of Chemical Physics, 1998. **108**(15).
26. Wiehe, K. and S. Schmidler, *Monitoring Convergence of Molecular Simulations in the Presence of Kinetic Trapping*. 2017.
27. Sawle, L. and K. Ghosh, *Convergence of Molecular Dynamics Simulation of Protein Native States: Feasibility vs Self-Consistency Dilemma.* Journal of Chemical Theory and Computation, 2016. **12**(2): p. 861-869.
28. Jacobs, D.J., *Ensemble-based methods for describing protein dynamics.* Current Opinion in Pharmacology, 2010. **10**(6): p. 760-769.
29. Barducci, A., M. Bonomi, and M. Parrinello, *Metadynamics.* Wiley Interdisciplinary Reviews: Computational Molecular Science, 2011. **1**(5): p. 826-843.
30. Laio, A., et al., *Assessing the Accuracy of Metadynamics.* The Journal of Physical Chemistry B, 2005. **109**(14): p. 6714-6721.
31. Laio, A. and M. Parrinello, *Escaping free-energy minima.* Proceedings of the National Academy of Sciences of the United States of America, 2002. **99**(20): p. 12562-6.
32. Hu, X., et al., *The dynamics of single protein molecules is non-equilibrium and self-similar over thirteen decades in time.* Nature Physics, 2015. **12**.
33. Jacobs, D.J., et al., *Network rigidity at finite temperature: relationships between thermodynamic stability, the nonadditivity of entropy, and cooperativity in molecular systems.* Physical review. E, Statistical, nonlinear, and soft matter physics, 2003. **68**(6).
34. Jacobs, D.J., et al., *Elucidating Quantitative Stability/Flexibility Relationships Within Thioredoxin and its Fragments Using a Distance Constraint Model.* Journal of Molecular Biology, 2006. **358**(3): p. 882-904.

35. Livesay, D.R., et al., *A flexible approach for understanding protein stability.* FEBS</cja:jid> FEBS Letters, 2004. **576**(3): p. 468-476.
36. Jacobs, D.J. and S. Dallakyan, *Elucidating Protein Thermodynamics from the Three-Dimensional Structure of the Native State Using Network Rigidity.* Biophysical Journal, 2005. **88**(2): p. 903-915.
37. Mark, A.E. and W.F. van Gunsteren, *Decomposition of the Free Energy of a System in Terms of Specific Interactions: Implications for Theoretical and Experimental Studies.* Journal of Molecular Biology, 1994. **240**(2): p. 167-176.
38. Dill, K.A., *Additivity principles in biochemistry.* The Journal of biological chemistry, 1997. **272**(2): p. 701-4.
39. Noskov, S.Y. and C. Lim, *Free energy decomposition of protein-protein interactions.* Biophysical journal, 2001. **81**(2): p. 737-50.
40. Simonson, T. and A.T. Brünger, *Thermodynamics of protein-peptide interactions in the ribonuclease-S system studied by molecular dynamics and free energy calculations.* Biochemistry, 1992. **31**(36): p. 8661-74.
41. Gao, J., et al., *Hidden thermodynamics of mutant proteins: a molecular dynamics analysis.* Science (New York, N.Y.), 1989. **244**(4908): p. 1069-72.
42. Hacisuleyman, A. and B. Erman, *Entropy Transfer between Residue Pairs and Allostery in Proteins: Quantifying Allosteric Communication in Ubiquitin.* PLOS Computational Biology, 2017. **13**(1): p. e1005319.
43. Karolak, A. and A. van der Vaart, *Importance of local interactions for the stability of inhibitory helix 1 in apo Ets-1.* Biophysical Chemistry Biophysical Chemistry, 2012. **165-166**(80): p. 74-78.
44. Jacobs, D.J., et al., *Protein flexibility predictions using graph theory.* PROT Proteins: Structure, Function, and Bioinformatics, 2001. **44**(2): p. 150-165.
45. Jacobs, D.J., *Generic rigidity in three-dimensional bond-bending networks.* Journal of Physics A: Mathematical and General, 1998. **31**(31): p. 6653.
46. Jacobs, D.J. and B. Hendrickson, *An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game.* Journal of Computational Physics, 1997. **137**(2): p. 346-365.
47. Jacobs, D.J. and M.F. Thorpe, *Generic Rigidity Percolation: The Pebble Game.* Phys. Rev. Lett. Physical Review Letters, 1995. **75**(22): p. 4051-4054.
48. Mottonen, J.M., D.J. Jacobs, and D.R. Livesay, *Allosteric Response Is both Conserved and Variable across Three CheY Orthologs.* Biophysical Journal, 2010. **99**(7): p. 2245-2254.
49. Herring, C.A., et al., *Dynamics and thermodynamic properties of CXCL7 chemokine.* PROT Proteins: Structure, Function, and Bioinformatics, 2015. **83**(11): p. 1987-2007.
50. Mottonen, J.M., et al., *Unifying mechanical and thermodynamic descriptions across the thioredoxin protein family.* PROT Proteins: Structure, Function, and Bioinformatics, 2009. **75**(3): p. 610-627.
51. Livesay, D.R., et al., *Hydrogen bond networks determine emergent mechanical and thermodynamic properties across a protein family.* Chemistry Central Journal, 2008. **2**(1): p. 17.
52. J. Jacobs, D. and M.F. Thorpe, *Computer Implemented System for Identifying Rigid and Flexible Regions in Macromolecules*. 2000: US.

53. J. Jacobs, D., *Computer Implemented System for Quantifying Stability and Flexibility Relationships in Macromolecules*. 2012: US.
54. J. Jacobs, D. and D.R. Livesay, *Computer Implemented System for Protein Design utilizing Quantitative Stability/Flexibility Relationships to control functions*. 2013: US.
55. J. Jacobs, D., *Computer Implemented System for Quantifying ensemble averaged properties of rigidity and flexibility in Macromolecules*. 2016: US.
56. Jacobs, D.J., *An Interfacial Thermodynamics Model for Protein Stability*, in *Biophysics*. 2012, InTech.
57. David, C.C. and D.J. Jacobs, *Characterizing protein motions from structure.* Journal of Molecular Graphics and Modelling Journal of Molecular Graphics and Modelling, 2011. **31**(2): p. 41-56.
58. Farrell, D.W., K. Speranskiy, and M.F. Thorpe, *Generating stereochemically acceptable protein pathways.* Proteins, 2010. **78**(14): p. 2908-21.
59. Liang, J. and K.A. Dill, *Are Proteins Well-Packed?* Biophysical Journal, 2001. **81**: p. 751-766.
60. Richards, F.M., *The interpretation of protein structures: Total volume, group volume distributions, and packing density.* J Mol Biol, 1974. **82**: p. 1-14.
61. Tsai, J., et al., *The Packing Density in Proteins: Standard Radii and Volumes.* J Mol Biol, 1999. **290**: p. 253-266.
62. Cioni, P., *Role of protein cavities on unfolding volume change and on internal dynamics under pressure.* Biophys J, 2006. **91**(9): p. 3390-6.
63. Frye, K.J. and C.A. Royer, *Probing the contribution of internal cavities to the volume change of protein unfolding under pressure.* Protein Science, 1998. **7**: p. 2217-2222.
64. Roche, J., et al., *Cavities determine the pressure unfolding of proteins.* Proceedings of the National Academy of Sciences, 2012. **109**(18): p. 6945-6950.
65. Raunest, M. and C. Kandt, *dxTuber: detecting protein cavities, tunnels and clefts based on protein and solvent dynamics.* J Mol Graph Model, 2011. **29**(7): p. 895-905.
66. Weisel, M., E. Proschak, and G. Schneider, *PocketPicker: analysis of ligand binding-sites with shape descriptors.* Chemistry Central Journal, 2007. **1**(1): p. 7.
67. Petřek, M., et al., *CAVER: a new tool to explore routes from protein clefts, pockets and cavities.* BMC Bioinformatics, 2006. **7**(1): p. 1-9.
68. Brezovsky, J., et al., *Software tools for identification, visualization and analysis of protein tunnels and channels.* Biotechnology Advances, 2013. **31**(1): p. 38-49.
69. Pérot, S., et al., *Druggable pockets and binding site centric chemical space: a paradigm shift in drug discovery.* Drug Discovery Today, 2010. **15**(15): p. 656-667.
70. Baldwin, R.L., *The new view of hydrophobic free energy.* FEBS Letters, 2013. **587**: p. 1062-1066.
71. Chandler, D., *Interfaces and the driving force of hydrophobic assembly.* Nature, 2005. **437**(7059): p. 640-7.
72. Reynolds, J.A., D.B. Gilbert, and C. Tanford, *Empirical correlation between hydrophobic Free energy and aqueous cavity surface area.* Proc Natl Acad Sci U S A, 1974. **71**(8): p. 2925-2927.

73. Voss, N.R. and M. Gerstein, *3V: cavity, channel and cleft volume calculator and extractor.* Nucleic Acids Research, 2010. **38**(Web Server): p. W555-W562.
74. Ho, C.M.W. and G.R. Marshall, *Cavity search: An algorithm for the isolation and display of cavity-like binding regions.* Journal of Computer-Aided Molecular Design, 1990. **4**: p. 337-354.
75. Kleywegt, G.J. and T.A. Jones, *Detection, delineation, measurement and display of cavities in macromolecular structures.* Acta Cryst, 1994. **D50**: p. 178-185.
76. Levitt, D.G. and L.J. Banaszak, *POCKET: A computer graphics method for identifying and displaying protein cavities and their surrounding amino acids.* J Mol Graphics, 1992. **10**: p. 229-234.
77. Chen, C.R. and G.I. Makhatadze, *ProteinVolume: calculating molecular van der Waals and void volumes in proteins.* BMC Bioinformatics, 2015. **16**: p. 101.
78. Laskowski, R.A., *SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions.* J Mol Graphics, 1995. **13**: p. 323-330.
79. Liang, J., H. Edelsbrunner, and C. Woodward, *Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design.* Protein Science, 1998. **7**: p. 1884-1897.
80. Edelsbrunner, H., D.G. Kirkpatrick, and R. Seidel, *On the shape of a set of points in the plane.* IEEE Transactions on Information Theory, 1983. **29**(4): p. 551-559.
81. Delaunay, B., *Sur la sphère vide. A la mémoire de Georges Voronoï* Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, 1934. **6**: p. 793-800.
82. Georges, V., *Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs.* Journal für die reine und angewandte Mathematik (Crelle's Journal), 1908. **1908**(134): p. 198-287.
83. Finney, J.L., *Random Packings and the Structure of Simple Liquids. II. The Molecular Geometry of Simple Liquids.* procroyasocilond Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1970. **319**(1539): p. 495-507.
84. Gerstein, M., J. Tsai, and M. Levitt, *The Volume of Atoms on the Protein Surface: Calculated from Simulation, using Voronoi Polyhedra.* J. Mol. Biol., 1995. **249**: p. 955-966.
85. Liang, J., et al., *Analytical shape computation of macromolecules.* Proteins: Structure, Function, and Genetics, 1998. **33**: p. 18-29.
86. Mach, P. and P. Koehl, *Geometric measures of large biomolecules: surface, volume, and pockets.* J Comput Chem, 2011. **32**(14): p. 3023-38.
87. Connolly, M.L., *Computation of molecular volume.* J Am Chem Soc, 1984. **107**(5): p. 1118-1124.
88. Richmond, T.J., *Solvent Accessible Surface Area and Excluded Volume in Proteins.* J. Mol. Biol., 1984. **178**: p. 63-89.
89. Wodak, S.J. and J. Janin, *Analytical appoximation to the accessible surface area of proteins.* Proc Natl Acad Sci U S A, 1980. **77**(4): p. 1736-1740.
90. Cazals, F., H. Kanhere, and S. Loriot, *Computing the volume of a union of balls.* ACM Transactions on Mathematical Software, 2011. **38**(1): p. 1-20.

91. Kleywegt, G.J. and T.A. Jones, *Detection, delineation, measurement and display of cavities in macromolecular structures.* Acta Crystallographica Section D Biological Crystallography, 1994. **50**(2): p. 178-185.
92. Lee, B. and F.M. Richards, *The interpretation of protein structures: Estimation of static accessibility.* Journal of Molecular Biology, 1971. **55**(3): p. 379,IN3-400,IN4.
93. Voloshin, V.P., et al., *An Algorithm for the Calculation of Volume and Surface of Unions of Spheres. Application for Solvation Shells*. 2011, IEEE Publishing. p. 170-176.
94. Gaines, J.C., et al., *Comparing side chain packing in soluble proteins, protein-protein interfaces, and transmembrane proteins.* Proteins: Structure, Function, and Bioinformatics, 2018. **86**(5): p. 581-591.
95. Shaytan, A.K., K.V. Shaitan, and A.R. Khokhlov, *Solvent accessible surface area of amino acid residues in globular proteins: correlation of apparent transfer free energies with experimental hydrophobicity scales.* Biomacromolecules, 2009. **10**(5): p. 1224-37.
96. Richards, F.M., *Areas, Volumes, Packing, and Protein Structure.* Ann. Rev. Biophys., 1977. **6**: p. 151-176.
97. Fleming, P.J. and F.M. Richards, *Protein packing: dependence on protein size, secondary structure and amino acid composition11Edited by F. E. Cohen.* Journal of Molecular Biology, 2000. **299**(2): p. 487-498.
98. Hoshen, J. and R. Kopelman, *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm.* Physical Review B, 1976. **14**(8): p. 3438-3445.
99. Al-Futaisi, A. and T.W. Patzek, *Extension of Hoshen–Kopelman algorithm to non-lattice environments.* Physica A, 2003. **321**(3/4).
100. Teuler, J.M. and J.C. Gimel, *A direct parallel implementation of the Hoshen-Kopelman algorithm for distributed memory architectures.* COMPUTER PHYSICS COMMUNICATIONS, 2000. **130**(1-2): p. 118-129.
101. Frijters, S., T. Krüger, and J. Harting, *Parallelised Hoshen–Kopelman algorithm for lattice-Boltzmann simulations.* Computer Physics Communications, 2015. **189**: p. 92-98.
102. Eddi, F., J. Mariani, and G. Waysand, *Transient synaptic redundancy in the developing cerebellum and isostatic random stacking of hard spheres.* Biol. Cybern. Biological Cybernetics, 1996. **74**(2): p. 139-146.
103. Bondi, A., *van der Waals Volumes and Radii.* J. Phys. Chem. The Journal of Physical Chemistry, 1964. **68**(3): p. 441-451.
104. Hobohm, U. and C. Sander, *Enlarged representative set of protein structures.* Protein Science, 1994. **3**(3): p. 522-524.
105. Fülöp, V., Z. Böcskei, and L. Polgár, *Prolyl oligopeptidase: an unusual beta-propeller domain regulates proteolysis.* Cell, 1998. **94**(2): p. 161-70.
106. Winter, M.B., et al., *Tunnels modulate ligand flux in a heme nitric oxide/oxygen binding (H-NOX) domain.* Proceedings of the National Academy of Sciences of the United States of America, 2011. **108**(43).
107. Hubbard, S.J., K.-H. Gross, and P. Argos, *Intramolecular cavities in globular proteins.* "Protein Engineering, Design and Selection", 1994. **7**(5): p. 613-626.

108. Eriksson, A.E., et al., *Response of a protein structure to cavity-creating mutations and its relation to the hydrophobic effect.* Science, 1992. **255**(5041): p. 178.
109. Spyrakis, F., et al., *Histidine E7 dynamics modulates ligand exchange between distal pocket and solvent in AHb1 from arabidopsis thaliana.(Report).* Journal of Physical Chemistry B, 2011. **115**(14): p. 4138-4146.
110. Voss, N.R., et al., *The Geometry of the Ribosomal Polypeptide Exit Tunnel.* Journal of Molecular Biology, 2006. **360**(4): p. 893-906.
111. Bui, J.M., K. Tai, and J.A. McCammon, *Acetylcholinesterase: enhanced fluctuations and complex alternative routes to the active site in the complex with fasciculin-2.* Journal of the American Chemical Society, 2004. **126**(23): p. 7198-7205.
112. Touw, W.G., et al., *A series of PDB-related databanks for everyday needs.* Nucleic Acids Research, 2015. **43**: p. D364-8.
113. Hubbard, S.J. and J.M. Thornton, *NACCESS*. 1993: Department of Biochemistry and Molecular Biology, University College London
114. Schrodinger, LLC, *The PyMOL Molecular Graphics System, Version 1.8*. 2015.
115. Jacobs, D.J., *Best probability density function from limited sampling.* Entropy, 2008. **11**: p. 1001-1024.
116. Farmer, J. and D.J. Jacobs, *Nonparametric Maximum Entropy Probability Density Estimation.* 2016: p. arXiv.org: 1606.08861.
117. Farmer, J. and D. Jacobs, *High throughput nonparametric probability density estimation.* PLoS One, 2018. **13**(5): p. e0196937.
118. Kapur, J.N., *Maximum Entropy Models in Science and Engineering*. 1989, New York, USA: Wiley.
119. Wu, N., *The maximum entropy method*. 1997, NY, USA: Springer.
120. Golan, A., G.G. Judge, and D. Miller, *Maximum Entropy Econometrics: Robust Estimation with Limited Data*. 1996, New York, NY: Wiley.
121. Abramov, R., *The multidimensional maximum entropy moment problem: a review of numerical methods.* 2010.
122. Ormoneit, D. and H. White, *An efficient algorithm to compute maximum entropy densities.* Econometric Reviews, 1999. **18**(2): p. 127-140.
123. Wilks, S.S., *Order Statistics.* Bull. Amer. Math. Soc., 1948. **54**(1): p. 6-50.
124. Houle, P., *Rngpack: High-quality random numbers for java*. 2003: http://www.honeylocust.com/RngPack.
125. Jacobs, D.J., *Best Probability Density Function for Random Sampled Data.* Entropy (Basel), 2009. **11**(4): p. 1001.
126. Geistlinger, H. and S. Mohammadian, *Capillary trapping mechanism in strongly water wet systems: Comparison between experiment and percolation theory.* Advances in Water Resources, 2015. **79**: p. 35-50.
127. Hassan, M.K. and M.M. Rahman, *Universality class of site and bond percolation on multifractal scale-free planar stochastic lattice.* Physical review. E, 2016. **94**(4-1): p. 4-1.
128. Iglauer, S. and W. Wülling, *The scaling exponent of residual nonwetting phase cluster size distributions in porous media.* Geophysical Research Letters, 2016. **43**(21): p. 11,253-11,260.

129. Huang, W., et al., *Critical percolation clusters in seven dimensions and on a complete graph.* Physical Review E, 2018. **97**(2).
130. Roy, B. and S.B. Santra, *Finite size scaling study of a two parameter percolation model: Constant and correlated growth.* Physica A: Statistical Mechanics and its Applications, 2018. **492**: p. 969-979.
131. Stauffer, D. and A. Bunde, *Introduction to Percolation Theory.* Phys. Today Physics Today, 1987. **40**(10): p. 122.
132. Tang, Q.Y., et al., *Critical Fluctuations in the Native State of Proteins.* Physical review letters, 2017. **118**(8).
133. Duckers, L.J. and R.G. Ross, *Percolation with non-random site occupation.* Physics Letters A, 1974. **49**(5): p. 361-362.
134. Duckers, L.J., *Percolation with nearest neighbour interaction.* Physics Letters A, 1978. **67**(2): p. 93-94.
135. Wang, J., et al., *Bond and site percolation in three dimensions.* Physical review. E, Statistical, nonlinear, and soft matter physics, 2013. **87**(5): p. 052107.
136. Harter, T., *Finite-size scaling analysis of percolation in three-dimensional correlated binary Markov chain random fields.* Physical review. E, Statistical, nonlinear, and soft matter physics, 2005. **72**(2 Pt 2): p. 026120.
137. Müller-Krumbhaar, H., *Percolation in a lattice system with particle interaction.* Physics Letters A, 1974. **50**(1): p. 27-28.
138. Yi, Y.B. and E. Tawerghi, *Geometric percolation thresholds of interpenetrating plates in three-dimensional space.* Physical review. E, Statistical, nonlinear, and soft matter physics, 2009. **79**(4 Pt 1): p. 041134.
139. Lorenz, B., I. Orgzall, and H.O. Heuer, *Universality and cluster structures in continuum models of percolation with two different radius distributions.* Journal of Physics A: Mathematical and General, 1993. **26**(18): p. 4711-4722.
140. Wadell, H., *Volume, Shape, and Roundness of Quartz Particles.* The Journal of Geology, 1935. **43**(3): p. 250-280.
141. Sonavane, S. and P. Chakrabarti, *Cavities and Atomic Packing in Protein Structures and Interfaces (Cavities and Atomic Packing in Proteins).* PLoS Computational Biology, 2008. **4**(9): p. e1000188.
142. Liu, J. and K. Regenauer-Lieb, *Application of percolation theory to microtomography of structured media: Percolation threshold, critical exponents, and upscaling.* Physical Review E, 2011. **83**(1).
143. Ding, B., et al., *Numerical analysis of percolation cluster size distribution in two-dimensional and three-dimensional lattices.* The European Physical Journal B, 2014. **87**(8): p. 1-8.
144. Rintoul, M.D., *Precise determination of the critical threshold and exponents in a three-dimensional continuum percolation model.* Journal of Physics A: Mathematical and General, 1997. **30**(16): p. L585-L592.
145. Amadei, A., M.A. Ceruso, and A. Nola, *On the convergence of the conformational coordinates basis set obtained by the essential dynamics analysis of proteins' molecular dynamics simulations.* Proteins, 1999. **36**.
146. Hess, B., *Convergence of sampling in protein simulations.* Phys Rev E Stat Nonlin Soft Matter Phys, 2002. **65**.

147. Knapp, B., et al., *Is an intuitive convergence definition of molecular dynamics simulations solely based on the root mean square deviation possible?* Journal of computational biology : a journal of computational molecular cell biology, 2011. **18**(8): p. 997-1005.
148. Berendsen, H.J. and S. Hayward, *Collective protein dynamics in relation to function.* Current Opinion in Structural Biology, 2000. **10**(2): p. 165-169.
149. David, C.C., E.R.A. Singam, and D.J. Jacobs, *JED: a Java Essential Dynamics Program for comparative analysis of protein trajectories.* BMC Bioinformatics, 2017. **18**(1): p. 271.
150. Esque, J., C. Oguey, and A.G. de Brevern, *A novel evaluation of residue and protein volumes by means of Laguerre tessellation.* Journal of chemical information and modeling, 2010. **50**(5): p. 947.
151. Dill, K.A., *Dominant forces in protein folding.* Biochemistry, 1990. **29**(31): p. 7133-7155.
152. Rother, K., et al., *Inhomogeneous molecular density: reference packing densities and distribution of cavities within proteins.* Bioinformatics, 2003. **19**(16): p. 2112-2121.
153. Cuff, A.L. and A.C.R. Martin, *Analysis of void volumes in proteins and application to stability of the p53 tumour suppressor protein.* Journal of molecular biology, 2004. **344**(5): p. 1199.
154. Gaines, J.C., et al., *Packing in protein cores.* Journal of Physics: Condensed Matter, 2017. **29**(29): p. 293001.
155. Gaines, J.C., et al., *Random close packing in protein cores.* Physical review. E, 2016. **93**(3): p. 032415.
156. Rycroft, C.H., *VORO++ : A three-dimensional Voronoi cell library in C++.* Chaos: An Interdisciplinary Journal of Nonlinear Science, 2009. **19**(4).
157. Andersson, K. and S. Hovmoller, *The average atomic volume and density of proteins.* Zeitschrift für Kristallographie, 1998. **213**(7): p. 369-373.
158. Gerstein, M. and C. Chothia, *Packing at the protein-water interface.* Proceedings of the National Academy of Sciences of the United States of America, 1996. **93**(19): p. 10167.
159. Chakravarty, S., A. Bhinge, and R. Varadarajan, *A procedure for detection and quantitation of cavity volumes proteins. Application to measure the strength of the hydrophobic driving force in protein folding.* J Biol Chem, 2002. **277**(35): p. 31345-53.
160. Angelov, B., et al., *Nonatomic solvent-driven voronoi tessellation of proteins: An open tool to analyze protein folds.* Proteins: Structure, Function, and Bioinformatics, 2002. **49**(4): p. 446-456.
161. Zhang, J., et al., *Origin of scaling behavior of protein packing density: A sequential Monte Carlo study of compact long chain polymers.* The Journal of Chemical Physics, 2003. **118**(13): p. 6102-6109.
162. Deng, H., Y. Jia, and Y. Zhang, *3DRobot: automated generation of diverse and well-packed protein structure decoys.* Bioinformatics (Oxford, England), 2016. **32**(3): p. 378-87.

163. Baldwin, R.L., *Gas-liquid transfer data used to analyze hydrophobic hydration and find the nature of the Kauzmann-Tanford hydrophobic factor.* Proc Natl Acad Sci U S A, 2012. **109**(19): p. 7310-3.
164. Kauzmann, W., *Some Factors in the Interpretation of Protein Denaturation*. Vol. 14. 1959. 1-63.
165. Huang, D.M. and D. Chandler, *Temperature and length scale dependence of hydrophobic effects and their possible implications for protein folding.* Proceedings of the National Academy of Sciences, 2000. **97**(15): p. 8324-8327.
166. Lum , K., D. Chandler, and J.D. Weeks, *Hydrophobicity at small and large length scales.* J. Phys. Chem, 1999. **103**: p. 4570-4577.
167. Pratt, L.R. and D. Chandler, *Theory of the hydrophobic effect* The Journal of Chemical Physics, 1977. **67**(8): p. 3683-3704.
168. Baldwin, R.L., *Properties of hydrophobic free energy found by gas-liquid transfer.* Proc Natl Acad Sci U S A, 2013. **110**(5): p. 1670-3.
169. Chothia, C., *Hydrophobic bonding and accessible surface area in proteins.* Nature, 1974. **248**(5446): p. 338-339.
170. Durham, E., et al., *Solvent accessible surface area approximations for rapid and accurate protein structure prediction.* J Mol Model, 2009. **15**(9): p. 1093-108.
171. van Dijk, E., A. Hoogeveen, and S. Abeln, *The hydrophobic temperature dependence of amino acids directly calculated from protein structures.* PLoS computational biology, 2015. **11**(5).
172. Baldwin, R.L., *Dynamic hydration shell restores Kauzmann's 1959 explanation of how the hydrophobic factor drives protein folding.* Proceedings of the National Academy of Sciences, 2014. **111**(36): p. 13052-13056.