Conformational Ensemble Generation via Constraint-based Rigid-body Dynamics Guided by the Elastic Network Model

by

Krzysztof Borowski

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Krzysztof Borowski

Abstract

Conformational selection is the idea that proteins traverse positions on the conformational space represented by their potential energy landscape, and in particular positions considered as local energy minima. Conformational selection a useful concept in ligand binding studies and in exploring the behavior of protein structures within that energy landscape. Often, research that explores protein function requires the generation of conformational ensembles, or collections of protein conformations from a single structure. We describe a method of conformational ensemble generation that uses joint-constrained rigid-body dynamics (an approach that allows for explicit consideration of rigidity) and the elastic network model (providing structurally derived directional guides for the rigid-body model). We test our model on a selection of unbound proteins and examine the structural validity of the resulting ensembles, as well as the ability of such an approach to generate conformations with structural overlaps close to the ligand-bound versions of the proteins.

Acknowledgements

I would like to thank my graduate supervisor, Forbes Burkowski, for extensive concept discussions, idea gathering, and care throughout the thesis and the entire time I spent at the University of Waterloo. His support of my research endeavors, and patience during the writing of this thesis have been gracious and extremely appreciated. I would like to thank Ming Li and Brendan McConkey for being my thesis readers. I also want to thank my family and friends for helping in various ways while I worked on this research. Finally, thank you to the Programming Languages Group for accepting an interloper into their habitat.

Dedication

This is dedicated to my family.

Table of Contents

List of Tables

List of Figures

List of Important Notation

Chapter 1

Ligand Binding Simulation, and Energy Considerations

Proteins are known to have marginal stability while in solution. What is considered to be the native state of a protein (often assumed to be the energetic minimum) is only 20-60 kJ mol^{-1} more stable than a corresponding denatured state of the same protein [47]. Atoms, due to their kinetic energy and interatomic forces, undergo motion relative to the remainder of the molecule. Consequently, this change in position causes the interatomic forces to change. This relationship between location and the interatomic forces causes constant fluctuations to occur between the forces responsible for forming protein structures; that is, ionic bonding, hydrophobic forces, the hydrogen bonds, and the van der Waals forces acting on individual atoms change constantly with the movement of each atom. This allows the protein structure to assume various different energetically favorable conformations. It also allows a protein structure to assume less energetically favorable conformations transiently, while exploring its nearby conformational space within the potential energy landscape. Under natural physiological conditions, protein functionality may require that these conformational changes occur. It is then necessary for the structural state of a protein to favor either a specific conformational state or a select few low energy conformations: it is the presence of these conformations that allows a protein to achieve functionality.

Due to their marginal stability protein structures may naturally traverse the energy landscape near the global minimum and between various surrounding local minima. At times, passing between neighboring energy valleys, a conformation may even need to traverse over a 'hill' on the energy landscape [69]. In Figure 1.1, a simplified depiction of such a landscape is presented in two dimensions. The various possible conformations

of a protein describe the energy landscape. Three-pronged stars indicate locations of local energy minima in energy valleys. The five-pronged star indicates the location of the global energy minimum. In reality this landscape is highly multi-dimensional due to the numerous degrees of freedom a protein structure contains.

These considerations become important in studying ligand binding. Ligands, often small biomolecules, bind receptor protein to create a protein-ligand complex which may cause changes in the function of the protein or induce a signal cascade inside a cell. This change in function, or the negation of this change, is often sought after in drug design, and thus understanding ligand binding may have important implications on the development of drugs and therapeutics [36].

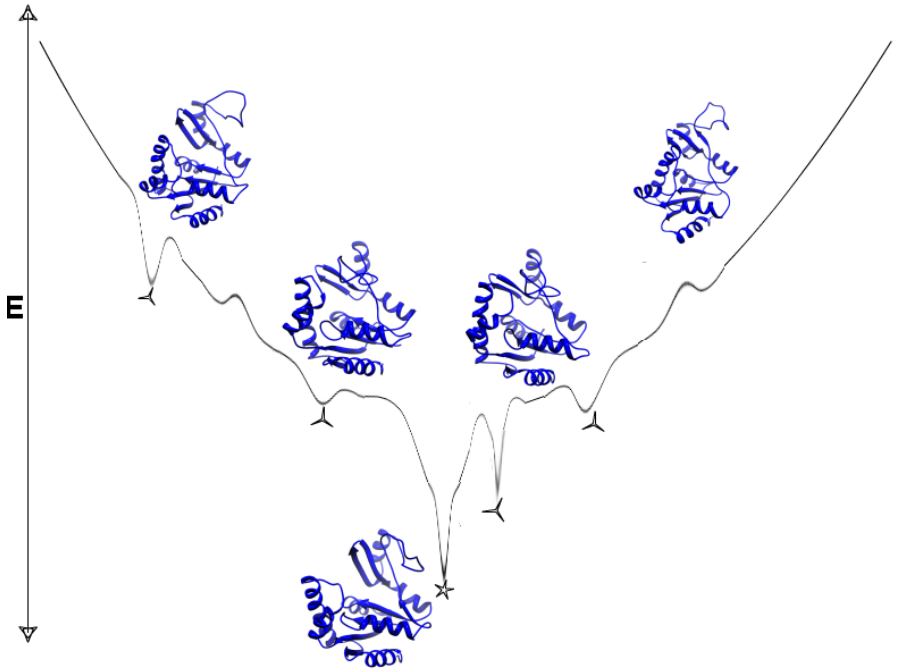


Figure 1.1: A simplified diagram of a theoretical potential energy well. Different conformations of 1EX6 used to illustrate this point. These and all further protein structure images generated using UCSF Chimera [49].

1.1 Ligand binding strategies

A ligand binding to a receptor protein may bind a non-global energy minimum conformation. This non-optimal unbound protein conformation may give rise to a bound protein conformation that has a lower potential energy overall. However, the only path to achieve this low-energy complex may require the unbound protein to reach the alternative, higher-energy conformation near the global minimum first. In certain environmental conditions (such as varying pH levels), specific conformations of a protein may be more stabilized than expected to allow binding of ligands [70]. Some proteins are even known to have local unfolding and folding events while under physiologically native conditions [53]. With all this variability in the behavior of protein structure, and thus in the behavior of ligands binding to protein, various understandings of the ligand binding mechanism have appeared in literature over the years.

The lock and key model

Fischer's description of the lock and key theory of protein and ligand binding (referring to enzymes and substrates at the time) was a simple description of the protein-ligand interaction mechanism [20]. The binding site (or the lock) was believed to be sturdy and immobile, with only one ligand (or the key) being specific enough to enter and activate function of the protein. The specificity of enzymatic catalysis was explained via this perceived mechanism. Before computational studies of this type of mechanism began, the idea of the lock and key complementarity between protein and ligands was challenged by a more realistic view of the ligand binding mechanism.

Induced fit

The concept of 'induced fit' has been prevalent for the majority of past ligand binding research, after Koshland's proposition of the concept in 1958 [39]. In this formulation of ligand binding, ligands are positioned in a receptor structure that is considered to be the predominant conformation in the protein population (often considered to be the structure of lowest free energy, and the global energy minimum). Upon ligand binding, the conformation of the active site in the protein changes in order to minimize the energy of the protein-ligand complex [35]. This method (in general) assumes that the ligand binds the protein while the protein is in its lowest free energy state. This may not be a sufficiently accurate assumption for certain proteins, and an expanded theory of ligand binding is necessary to consider this process in such cases.

Conformational selection

Conformational selection is another concept for explaining ligand binding. It avoids the assumption that only a single protein conformation is initially necessary by finding alternative conformations of the macromolecule. With alternative conformations, docking studies use multiple protein structures when simulating ligand binding. These alternatives are usually energetically near the global minimum in the energy well [69]. The high-dimensional energy landscape around the global energy minimum is rough and rugged, in the form of valleys and peaks. Traversing this energy landscape is the key to finding ensembles suitable for conformational selection. The valleys nearby the global energy minimum may be biologically important conformations of the protein essential to ligand binding and function. A collection of these alternate conformations is known as a conformational ensemble.

Conformational ensembles have uses beyond just protein-ligand interaction studies. Ensemble generation is important in generating protein structures in protein design research [50]. Such research allows for the generation of protein sequences necessary to create a specific conformation. It requires the generation of protein backbones in a way that is efficient and results in biologically plausible conformations. Homology modeling often uses a template structure which is modified based on the amino acid sequence of the target whose conformation is desired [76]. Additionally, pathway generation is another reason for conformational ensemble generation [66]. In pathway generation, stepping along a succession of conformations from one structure to another allows for a deeper understanding of specific protein behavior and functions. However, for such a path to make biological and energetic sense, free energy changes between conformations must be considered.

In the current work we present an approach to conformational ensemble generation that uses constraints as an approximation of protein connectivity and rigidity, and uses classical dynamics and harmonic approximations of protein motion to generate conformational ensembles. We show that such an approach can efficiently generate conformations of protein with sound potential energy profiles. Due to the use of harmonic approximation the approach can generate conformations close to the ligand-bound conformations from unbound conformations alone.

Chapter 2

Conformational Ensemble Generation Methods

2.1 Ensemble generation methods

2.1.1 Conformational ensemble generation via motion planning and rigidity analysis

In robotics, the motion planning problem is the problem of finding a path from one conformational state of an object to another. In terms of protein structures, this would require finding a path between two energetically favorable conformations. It consequently requires the generation of conformations to represent points on the energy landscape.

Early work on the subject involved finding 'saddle points', or high energy conformations around energy valleys. These local energy maximums were used as starting points for minimization techniques in order to find the adiabatic reaction path (the minimum energy path), but the actual generation of new conformations is done by linear interpolation [21].

A recent method of path finding on the energy landscape is the probabilistic roadmap method (PRM). The PRM samples random points in the conformational space of a structure and retains those points where certain feasibility requirements are met. The points that remain are connected to nearby points (nearby in terms of some distance

function; an energy function in the case of protein conformation). Using these points in the conformational space, paths between the points can be extracted.

The motion planning of paths of ligand structures have been explored using an articulated robot-like linkage and motion planning by Singh et al. (1999) [57]. Their research involved using PRM's to study binding pockets in protein, but whole structure applications have recently appeared. As these approaches use rigidity theory, we describe some rigidity related notions before returning back to PRM's for protein structures.

Rigidity theory has been applied to protein structures analysis and ensemble generation. The Floppy Inclusions and Rigid Substructure Topography (FIRST) program is designed to identify rigidity and flexibility in network graphs using an algorithm called the pebble game [34]. The pebble game is a constraint counting algorithm that determines the degrees of freedom based on a two-dimensional representation of a graph. Pebbles represent degrees of freedom relevant to each node. A pebble can be moved to any edge that is adjacent to the node, which makes it necessary for the edge to be covered by a pebble at all future times during the algorithm. After completion of the algorithm, the pebble game defines rigid components of a graph. Since it can be applied to 3D bond-bending networks (a bar-joint model) [33], the pebble game has been previously applied in protein rigidity and flexibility analysis [2, 34].

The usefulness of classifying atoms in rigid clusters, which are the output of FIRST, is evident in the Framework Rigidity Optimized Dynamics Algorithm (FRODA) server [74]. FRODA uses the rigidity results from the pebble game to create rigid clusters. Those clusters then undergo randomized motion in order to create structural conformers. One of the prominent uses of FRODA is targeted dynamics, where both an initial and a final conformation are available and a path of conformations is found between the two conformational states [74]. When no target exists, however, the motions are random and undirected, which can result in smaller deviations from the initial state.

The rigidity considerations present in FIRST have also been used in the PRM methods of Thomas et al. (2007), who also explore the potential energy landscape of proteins using improved PRM's [66]. They model protein folding pathways and use rigidity constraints computed from the pebble game algorithm [34] in order to simulate protein motion. Their iterative sampling of the protein energy landscape involves small Gaussian perturbations, using the pebble game rigidity analysis as a guide for these perturbations [66].

Haspel et al. (2010) use a coarse grained energy function along with a backbone and limited side-chain representation of proteins to study paths of large-scale conformational motions [28]. A conformational sampling method from their previous work was used

to generate conformations. A Monte Carlo simulated annealing algorithm was used, with perturbations of less than two degrees applied to the dihedral angles of the protein backbone. The simulations are launched from conformations that are evaluated at the all atom level [56]. This method requires shifting between the backbone and all atom scale models, but results in pathways consistent with experimental data [28].

2.1.2 Molecular dynamics sampling of the energy landscape

Molecular dynamics (MD) simulations are also capable of producing ensembles [19]. Explicit MD simulations consider most of the interactions between the atoms in a molecule, and calculate bonding and non-bonding interactions between all the atoms. Such a high level of detail in computations of energetics and consequent dynamics of the atoms in a protein facilitates detailed and accurate simulations of protein movement. This allows for the ensemble to be sterically and energetically correct. The extent of computational power and time required for MD simulations to attain this level of detail is large. MD simulations will often have a timestep of femtosecond length, and many simulations can achieve trajectory information for events on the nanosecond scale. However, many protein structural rearrangement events, such as functionally important conformational changes or folding events, take much longer in reality [65]. Reasonable simulations would require trajectories running for microseconds, milliseconds, and sometimes even seconds. With a femtosecond time-step, reaching such simulation lengths becomes a difficult task. Thus, worthwhile MD simulations are very computationally expensive and often require the use of cloud computing and large servers. Alternatives to such simulations employ models that simplify the energy functions and interactions between atoms in molecules.

2.1.3 Protein design to conformation generation

Many methods for ensemble generation were originally acquired through the development of protein design methods. Protein design is the inverse of the protein folding problem: given a known or needed protein structure (what we want to design), protein design attempts to find an amino acid sequence that will fold to this target structure [64]. Generating structural ensembles is important in protein design, as the research requires backbone frameworks that are biologically reasonable [69].

The Baker lab and their Rosetta software created conformational ensembles through backbone flexibility studies in which portions of the protein are spliced into the structure

in question from other known protein structures based on sequence similarity [50]. Such methods require previous knowledge and databases of information about known structures.

Moving large, rigid portions of a protein requires that these portions are reconnected. Loop-closure methods are used to merge such portions, which allows for large scale motions to be modeled [8]. In this approach, large helices and sheets, as well as entire domains of proteins, are translated, and subsequently connected by the modeling of a loop between these pieces. Inverse kinematics, another method from the field of robotics, is the main component of loop-closure methods [13]. Inverse kinematics is the method of solving the problem of finding the angles between joints and bodies necessary for a known, expected final position. In the case of protein design, one knows the endpoints of a loop, and the constraints in the loop (the bond lengths and bond angles), that can be used as input to inverse kinematics algorithms to develop loop-closures. Non-linear programming can be used to solve inverse kinematics problems, and while optimized methods and algorithms exist, a final conformation is necessary in such approaches.

Another method, known as 'dead-end elimination' (DEE), searches for the global minimum energy conformation by pruning the conformations containing rotamers that would not naturally exist in the low-energy conformations [25]. If a path becomes infeasible, as indicated by a scoring function, the algorithm stops generating further paths containing the infeasible structures.

The concept of *backrub motions* in the backbone has generated various methods of creating alternate conformations [14]. In these methods, three amino acids in a protein chain are considered and all the atoms between the first and third alpha carbons are rotated around the axis between the carbons. Following this, the atoms between the first and second amino acids, as well as the atoms between the second and third amino acids are rotated to relieve strain due to the initial rotation. Combining this with DEE algorithms allowed for development of sequences that gave conformations with low energies [24]. Backrub motions were modeled using Monte Carlo (MC) methods by Smith and Kortemme (2008) where the rotations were performed on segments ranging in length from two to twelve amino acids [58]. Both of these methods, however, keep conformation perturbations from becoming excessively large. In many cases, conformational changes during the life of a protein require movements larger than such exploration methods allow [54].

2.1.4 Distance geometry

Distance geometry, or the algorithms that deal with distances between atoms, has been used in conjunction with genetic algorithms to generate conformer ensembles of small ligands. With distance geometry generating the initial structures, the genetic algorithm then changes the torsional angles of the ligand molecules to generate a set of alternate conformations before a force field is used for energy minimization [71].

Distance geometry methods have been used to develop ensembles of protein conformations using the CONCOORD algorithm [15]. The algorithm randomly perturbs distances in a Gaussian manner before it traverses distance restraints randomly to correct those that are outside their allowed interval. Seelinger and de Groot (2010) created ensembles using the tCONCOORD program while using constraints describing the radius of gyration of the conformation. Their dataset contains proteins that undergo large conformation changes during ligand binding. They evaluate their method by generating ensembles from an unbound structure and a ligand bound radius of gyration (the root mean squared distance of the atoms to the center of mass of the protein). However, their method requires that the radius of gyration be experimentally calculated for the unknown bound structures [54]. The authors argue that this additional input is easily acquired through wet-lab means, though this requires some specific knowledge about the final ligand-bound structure in question. Their method also involves a variety of refinement steps, one of which is MD refinement causing an increase in required computation [54].

2.1.5 Normal mode analysis and ensemble generation

The normal mode analysis (NMA) of a protein involves the generation of normal modes, or vectors of preferred motion, from a single protein configuration. The method initially uses a single structure assumed to be at the global energy minimum [16]. Using this structure, a network of inter-atomic connections is built and oscillatory Hookean motions are calculated for the protein. This method can be used in developing an ensemble of conformations as well as understanding large-scale (low frequency) motions of a protein [23]. NMA is known to produce low frequency modes that compare well to real conformational differences seen in crystallography experiments [44]. For example, hemoglobin's change from its T state to its R state contains directions of motion very similar to the second lowest frequency mode, and this change highly affects the efficiency of oxygen binding [48].

NMA has been used in various ways to generate new protein conformations. This approach is often used instead of MD simulations because of the reduced computation requirements while still managing to predict realistic, low frequency large-scale motions within a protein [16]. While faster than MD simulations, NMA methods can be computationally expensive as well due to the size of the $3N \times 3N$ Hessian matrix of second order partial derivatives used in the computations. The potential energy equations used, as well as their second derivatives, can be difficult to calculate in some NMA models [29].

The computational expense of NMA can be avoided by simplifying the model. An elastic network model (ENM) can be described as an abstraction of protein fluctuations at equilibrium based on Hookean spring potentials [68]. An ENM where alpha carbons are the only interacting sites means the model uses only alpha carbon locations in the calculations. Such a model describes the interacting sites as interconnected with springs that fluctuate around their equilibrium position. In an alpha carbon-based ENM, a spring is placed between alpha carbons that are within a given distance threshold from each other. Depending on the threshold, this may give either a dense or sparse model. Abstracting the model beyond alpha carbons, interaction sites could be anything from single atoms to secondary structures. Depending both on the choice of the site and the threshold, various types of harmonic models can be built [44].

Harmonic approximations calculated using the ENM have been found to accurately predict conformational changes within proteins, with the largest changes coinciding with the lowest frequency modes calculated by the model [16]. This qualifies the ENM formulation of protein harmonics for use in the discovery of important conformational changes of the protein structure.

Pathways between known states have been generated using ENM's and geometrical methods. A distance interpolation method developed by Kim et al. (2002) creates intermediate conformations between two local energy minimum structures using ENM's generated from the two different conformations of the same protein. The authors of the work argue that geometric methods are superior to dynamics-based methods for this purpose due to the requirement of small time-steps in dynamics-based methods [37]. Another approach of this type by Zhang et al. (2007), in which a mixed ENM is formulated by combining the potentials between the start and end structures before interpolating across them [77].

He et al. (2003) describe a method of sampling the energy landscape by using collective motions discerned from ENM and amplifying these motions in MD simulations [30]. Their method cycles between stages of relaxation (where MD simulation allows for a local minimum to be reached) and excitation (where the normal modes are coupled

to the MD simulation trajectories). The ENM is used as a method of escaping local minima. Local minima often cause MD simulation-based structure sampling to become inefficient, as the structure remains within these energetic wells for a long time before moving into another low energy state. The collective normal modes of the ENM can be used to avoid this limitation of MD [30].

A method developed by Cavasotto et al. (2005) uses NMA to create new conformations by selecting normal modes that affect specific areas of interest within the protein. Structures are perturbed along a combination of relevant normal modes (i.e. modes that affect a specific region of interest within the protein). This is followed by Monte Carlo methods for side chain optimization before the ensemble is used for docking and virtual screening studies. Using this ensemble has been shown to increase docking scores and enrichment factors [10].

ENM is most often used in a coarse-grained manner, but all-atom ENM models have also been examined in literature. While more computationally intensive, such approaches provide more details about the harmonic, conformational behavior of protein structures. All atom ENM's can be used to acquire the normal modes of specific amino acids, and various smaller atom-specific movements. Rueda et al. (2009) have used such an all-atom approach which yielded cross-docking improvements when using an ensemble generated by this method [51].

Fu et al. (2007) used backbone flexibility considerations in their protein design calculations while also using NMA to sample various backbone conformations [23]. Due to the limited distortions produced by classical normal mode methods and due to the possibility of bad geometries occurring from this method, Yang and Sharp (2009) built upon the typical ENM method by introducing two extra force parameters [75]. These parameters account for interactions between consecutive alpha carbons and the interactions within the same secondary structures. This method requires that the entire backbone be rebuilt around the alpha carbons and that an ENM be newly rebuilt at each step. The Yang and Sharp (2009) method also discards structures based on distance and angle thresholds at each step before deciding to continue with the low frequency mode directions or backtracking to use other modes, in a fashion similar to that of DEE [75].

Song and Jernigan (2006) use an extra force parameter between domain sections of the protein to approximate rigidity considerations [61]. Using a larger harmonic potential for inter-domain contacts, this domain-ENM model takes the rigidity of protein domains into account. Other rigidity constraints are applied to ENM analysis by using the pebble game of Thorpe et al. (2001). The graph-centric algorithm for rigidity classification is used alongside an ENM in work by Ahmed and Golke (2006), in order to define rigid

clusters of the protein with flexible connections between these clusters [2].

2.2 Constraint algorithms

Constraint algorithms can facilitate the calculations of the simulations of protein structure dynamics. A constraint algorithm is a method of employing restrictions on the movement of objects in dynamics simulations. These algorithms have been previously used in MD simulations [5, 62].

Representing the system in coordinates based on the degrees of freedom, also known as internal coordinates, is one method of adding constraints to a protein simulation. In this scenario, protein properties such as the dihedral angles are the coordinates. Such an approach eliminates the need for explicit constraints between bonded atoms. Originally used by Abe et al. (1984) as a method of energy minimization, internal coordinates have since been used in constraint algorithms in MD simulations [5]. The use of these methods requires extending the internal coordinates to explicit atomic coordinates at the end of the simulation. This approach also limits the types of constraints one can impose on a protein structure, as the definition of internal constraints can be difficult. For example, while constraining rigid loops is possible, flexible loops such as the ones caused by disulfide bonding are more difficult to constrain [22].

More often, Lagrange multiplier methods are used to impose constraints on a MD simulation. The SHAKE algorithm [52] and many variations thereof [43], use the Gauss-Seidel method (which we describe in Chapter 3) to approximate a linear system of equations in order to ensure bond geometry constraints remain within bounds.

2.3 Critique of previous methods

The time scales restricting the sampling of the energy landscape when using MD cause these methods to require an immoderate amount of computation, as shown in research described in Section 2.1.2. In avoiding this issue, methods that use NMA (Section 2.1.5) provide a more computationally efficient way of altering conformational states of proteins. However, while coarse-grained NMA based methods conceptualize the protein as a collection of linked bodies, the actual rigidity constraints caused by bonding at the atomic level are not explicitly considered. Work combining rigidity consideration and the ENM is limited to considering only domain rigidity, or coarse inter-residue rigidity.

Separately, the methods that do take rigidity into account without considering low-frequency motions often employ Gaussian perturbations as a form of generating new conformations (Section 2.1.1). This approach causes the exploration of the energy landscape to be more sporadic than the NMA methods, which appropriate the connectivity of potential energy interactions within a protein as a guideline for conformational changes. It also requires larger numbers of generated conformations before structures with a significant difference in structural overlap are generated.

Seeliger and De Groot (2010) argue that the ensemble generation methods that use backrub motions result in conformational changes smaller than those required by certain applications of protein ensembles [54]. Their own method, based in distance geometry, requires additional experimental data for efficient sampling of conformations when sampling with a target structure in mind.

2.4 Problem statement and the proposed solution

Ensemble generation techniques are important tools in the analysis and understanding of protein behavior, motion planning, and ligand binding analyses. Current ensemble generation methods encompass a set of limitations. They are either 1) too computationally expensive, 2) lack a thorough consideration of protein rigidity in the models used or use a coarse-grained approach to rigidity due to bonds, 3) result in conformational changes too small to properly sample the nearby energy landscape or 4) rely on random motion or the availability of previous knowledge of the protein motion.

Techniques that avoid the issues and adhere to the requirements of ensemble generation are necessary to develop a deeper knowledge of the energy landscape of proteins. It is this landscape that regulates motion and functionality of protein structures, and sampling these structures well *in silico* allows for improved computational modeling of various other biological processes, like protein mobility and ligand binding.

We present a technique of ensemble generation that uses kinematic joints as constraints describing rigidity at the backbone atom level in a protein. While maintaining these rigidity considerations, we use the ENM to efficiently guide sampling of nearby conformations from a single protein structure. We use a constraint-based rigid-body dynamics system in order to keep computational requirements low and to allow for intrinsic consideration of rigidity. This ensemble generation approach combines the efficiency of NMA based approaches with the importance of non-coarse rigidity considerations without increasing computational requirements.

Chapter 3

Backbone Rigid-body Model with Normal Mode Guided Iterative Dynamics

In the remainder of this work, we will be using Newtonian kinematics of rigid bodies to model protein motions. Our goal is the efficient production of alternative protein conformations that correspond to large scale movements suggested by the normal modes described by an elastic network model, while constraining the model to the allowable set of states determined by intrinsic distance and angle constraints.

3.1 Introduction

It is important to create a distinction between the terms of *rigid-body* and *rigidity*: while they appear to be similar, the former only describes the types of objects present in a simulation, and the latter describes the connectivity and flexibility restraints of a protein structure. While many physics-based simulations and Newtonian methods have been developed in the area of protein modeling and extended to computationally intensive methods such as molecular dynamics (MD), constraint-based rigid-body models of protein have not appeared in the literature as a vehicle for ensemble generation of protein structures while using rigidity considerations. A constraint-based rigid-body simulation requires much less computing power than a complete MD simulation with full energetic considerations. In this case, parts of the protein as well as some energetic

considerations are abstracted and simplified. Representing the protein in terms of rigid bodies, based on bonds or upper-level structures, may be useful for exploration of conformational space, especially with constraints being included in the simulation to describe rigidity. Indeed, rigidity constraints have already been used in past work on protein motion [2, 34, 61, 66]. However, these works have a coarse-grained approach to protein rigidity.

The examination of other ensemble generation techniques, described in Chapter 2, has elucidated a few key aims necessary in conformational ensemble generation via computational means. To summarize aims in ensemble generation, an ensemble must:

- Contain members that are sufficiently different from one another
- Contain biologically plausible conformations (by not violating bond length and bond angle constraints, and avoiding clashes between non-bonded atoms)
- Contain energetically stable conformations, which means that conformations must not rise too much in energy in comparison to the global minimum
- Be generated efficiently

In this chapter, we briefly describe the steps taken to generate conformations using our rigid-body model. We then explain each step in detail, followed by the presentation of notes on the implementation.

3.1.1 Overall methodology

Our ensemble generation method uses the steps shown in Figure 3.1. Initial normal mode calculated using an ENM are used as a guide for the rigid-body simulation of the backbone structure. This is followed by side-chain repacking and energy minimization.

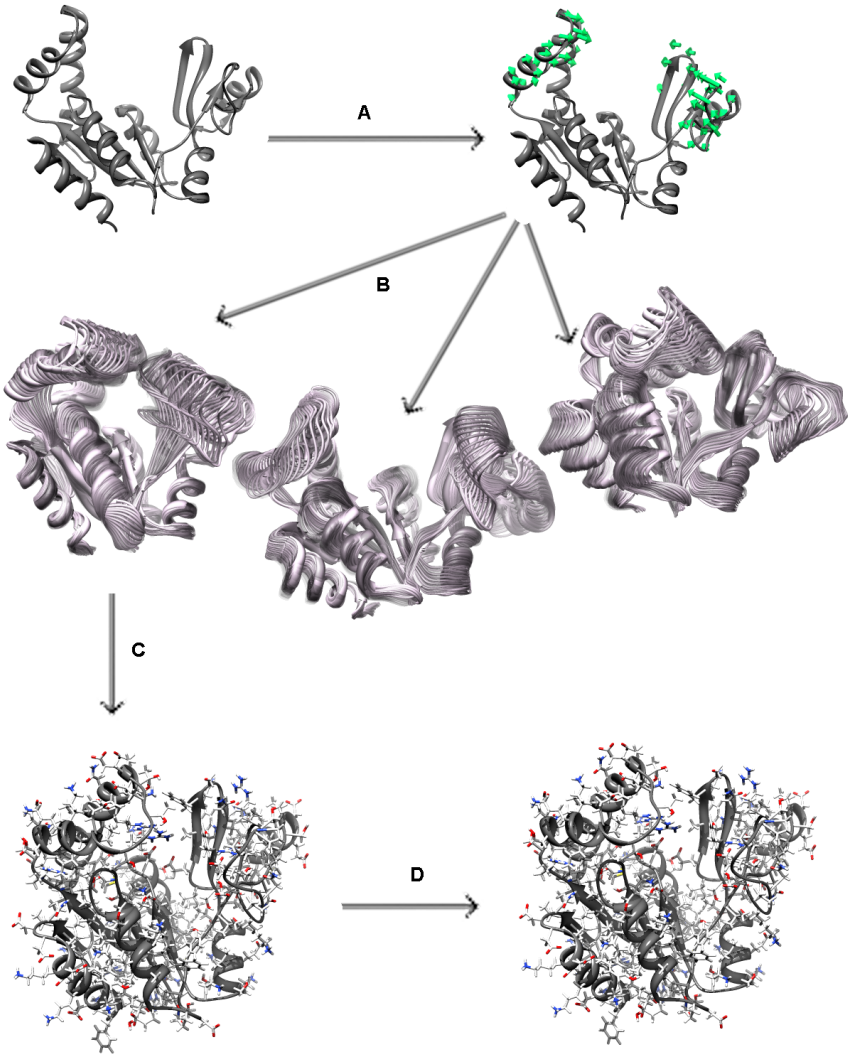


Figure 3.1: The sequence of steps in the ensemble generation method: A) calculating normal modes, B) generating backbone structures via rigid-body simulation, C) side-chain repacking, D) energy minimization.

The rigid-body simulations and the constraint model and time-stepping algorithm are described first. Second, the ENM is described along with its application to the rigid-body model. Finally, implementation details and tools used for the method are covered.

3.2 Rigid-body simulations

Rigid-bodies are solid, non-malleable objects with specific shapes existing in Cartesian space. A rigid-body simulation is the calculation of motions of a collection of rigid bodies that may or may not be connected through a variety of linkages, known as joints. While the solid body experiences no change in form, it may experience a change in position and orientation. When multiple rigid bodies exist, the forces acting upon them may cause a rigid body to change position, thereby affecting the position or orientation of other rigid bodies. This can be thought of as the typical physics scenario found in games of billiards or pool. In these games, rigid balls interact with each other and cause both changes in position and orientation in relation to other rigid balls. Such a scenario can be simulated with simple physical Newtonian equations, describing the state of each ball on the pool table.

It is possible for rigid-body simulations based on Newtonian physics equations to employ restrictions on the motions that occur within the simulation. This can be achieved by creating dependencies between the rigid-bodies, referred to as joints or constraints, defined by constraint equations. The joints found in the human body serve as examples of the types of joints that can be found within a rigid-body simulation. Two rigid bodies that are attached by a joint are limited in the motions they can achieve based on the location and motion of the other body in the pair. The joint thus imposes constraints of distance and mobility of each body.

The typical knee-joint constrains the femur bone above the knee and the tibia and fibula bones below the knee to bend at only one axis, and only to specific maximum and minimum degrees along that axis. Similarly, a constraint equation imposes limitations upon the possible motions of two rigid-bodies in a physics simulation. Various types of joints can be used in a computer model, allowing for simulation of real-world situations such as the motion of a leg or the motions of atoms in a protein structure.

A hinge joint would impose restrictions similar to that of a knee-joint on two connected rigid bodies, as shown in Figure 3.2. A hinge joint (depicted as a cylinder) between two rectangular rigid bodies is shown with two different amounts of rotation, and a rotational axis is indicated with a dashed line. Positioning the rotational axis in a different way, restrictions similar to that of bond rotations become possible, as shown in 3.3. Distance constraints and rotational constraints are just two examples of virtual joints that can exist between two rigid bodies.

It is possible to represent various systems using rigid-body simulations. One of those systems is a general protein structure. We proceed by describing the rigid-bodies and

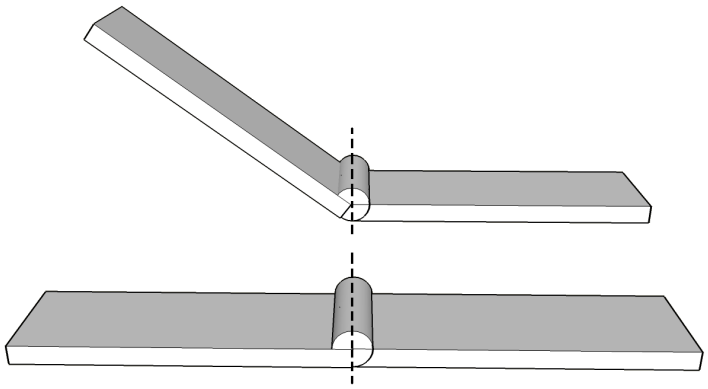


Figure 3.2: A hinge joint (depicted as a cylinder) between two rectangular rigid bodies shown with two different amounts of rotation. The dashed line indicates the rotational axis perpendicular to the length of the rectangles.

joints that connect these bodies.

3.3 The molecular backbone rigid-body model

Proteins are composed of atoms and bonds between the atoms. Many pictorial representations of molecules use a simplified description of the atoms and bonds, the components of molecules, to present molecular structures. Ball-and-stick models show the atoms as spheres and bonds as sticks connecting these atoms. This kind of representation will be useful in conceptualizing the rigid-body model of protein structure.

We start by defining atoms as rigid-body spheres with a center of mass in a 3-dimensional (3D) frame of reference. In a system of n atoms, n respective rigid-body spheres are defined to simulate the atoms. The radius of the spheres is the atomic van der Waals radius. The van der Waals energy spheres of covalently bonded atoms within real protein structures intersect, but unbound atoms experience van der Waals forces that cause attraction and repulsion, disallowing close association between the atoms. In the model, unbound atom radii are considered to be solid and thus inhibit the unbound atoms from getting too close to one another. Figure 3.4 shows an example of these kinds

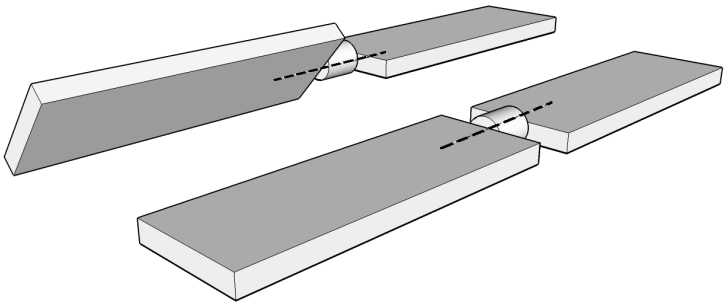


Figure 3.3: A hinge joint with the dashed line indicating the rotational axis parallel to the lengths of the rectangles.

of spheres and their interactions. We show a ball and stick model of the $N_i - Ca_i - C_i - N_{i+1}$ backbone atoms, with the van der Waals spheres superimposed over the model. Van der Waals spheres intersect for bonded atoms and geminal atoms, but atoms with at least three bonds between them have clearly separate van der Waals spheres. In Figure 3.4, this separation is visible between the two N atoms (both depicted as blue spheres). The bound atoms have intersecting van der Waals spheres and even the spheres of the geminally bound atoms interpenetrate, while the unbound atoms do not. This specification for the size of the atom bodies allows the model to avoid energetically impossible conformations during the simulation. While using rigid spheres is a simplification of the actual van der Waals forces, which dominate the limits on proximity between atoms in a molecule, it means that various energy calculations need not be employed to maintain an allowable distance between atoms. In turn, this lessens the number of energy minimization steps often applied at the end of such simulations.

3.3.1 Bonds as Joints

A covalent bond imposes a constraint on the motion of two atoms in nature. This functionality is approximated here by forming a joint between two covalently bound rigid-body representations of atoms (if the atoms are bound in nature). The type of joint

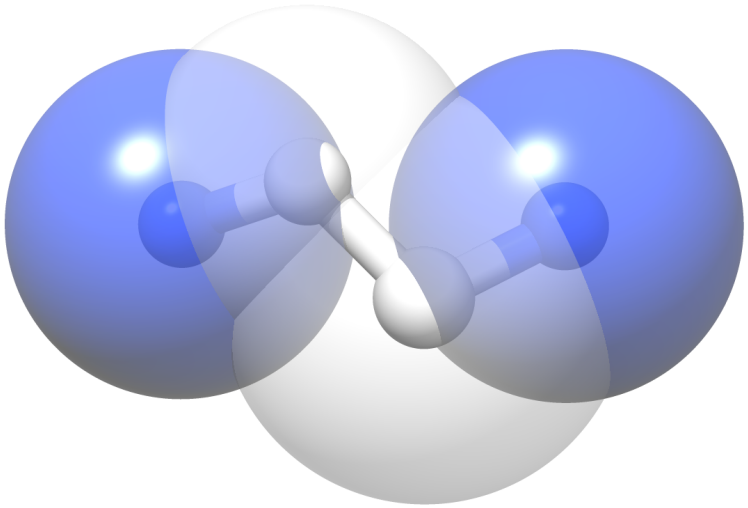


Figure 3.4: The bond joints between atoms in a small portion of the protein backbone, with small spheres representing atoms and large, transparent sphere representing the extent of the van der Waals radius of each atom.

used to represent a covalent bond is a hinge-joint, which allows rotation between two atoms around a specified axis. The hinge joint permits rotation between bonded atoms as shown in Figure 3.5. Bond distances and bond angles remain the same, but dihedral angles experience a change. The hinge joint rotation occurs about the axis running between the center of both atoms. This type of mobility, though simplified in the model, exists within real molecules and is explained in all chemistry and biochemistry texts. Restricting our model to rotate the atom bodies around the bond axis maintains the bond angles between atoms. Since the joint holds the atom bodies rigidly in the rotational axis, the bond distance is kept unchanged. Because the distances and the bond angle remain unchanged, the geminal distances stay the same throughout the simulations. Overall, only the dihedral angles experience changes after application of forces to this model.

In the backbone model, the specific bonds that are taken into rotational consideration are the $N_i - Ca_i$ bond, the $Ca_i - C_i$ bond, and the $C_i - O_i$ bond. Each of these bonds is turned into a hinge-joint connecting two rigid bodies representing the atoms. Figure 3.5 shows a small portion of the backbone before and after a rotation around the $N_i - Ca_i$ bond. We also use joint constraints to model disulfide bonds between the Cb atoms. Because of the chemical nature of the $C_i - N_{i+1}$ peptide bond, where the torsion angle

(known as the Omega angle) remains at approximately 180° or 0° [41], the model does not allow rotations about the bond axis between these atoms. For this bond, a fixed joint is used. A fixed joint merges the motion and rotation of the two bodies involved and disallows rotation between the C_i and N_{i+1} atoms of the backbone model.

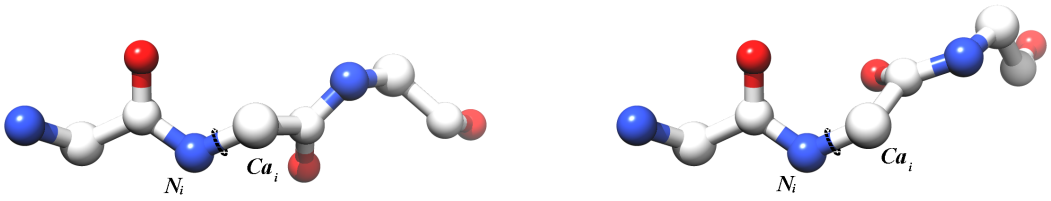


Figure 3.5: An example of a rotation around the $N_i - Ca_i$ bond of a protein backbone.

3.3.2 Hydrogen bond considerations

Finally, an essential consideration in the backbone model is the addition of hydrogen bond (H-bond) constraints. In our approach, each H-bond constrains the two bonded atoms with a hinge joint. Since we do not use side-chains in the simulation, H-bonds are not considered between side-chain and backbone atoms. Instead, we rely on backbone-to-backbone H-bonds only. As the H-bonds in this model experience the limited malleability allowed by hinge joints, rigidity considerations from the existence of H-bonds are not ignored.

Unfortunately, setting hydrogen bonds as permanent constraints in the model causes some regions in the conformational space to be omitted as a source for conformational sampling. Indeed, this may cause some subtle conformational changes to be missed by the model as internal H-bonds are often broken and formed during the lifespan and function of a real protein. Other ensemble generation techniques have encountered this issue [55]. While this is an important limitation, our model does allow for large-scale domain and secondary structure motions. This concern opens a potential new endeavor in ensemble generation: the consideration of structural constraints as non-static entities

may improve the sampling of the structures in the energy valleys surrounding the global minimum. However, as one of the aims of our model is simplicity and efficiency while maintaining the most essential constraints, we avoid dynamic constraints such as changing bond lengths, and assume static initial constraints remain constant throughout the system during simulations.

The existence of H-bonds is predicted based on geometric criteria based on a survey of small protein crystal structures from the Cambridge Structural Database [46]. There are various geometric criteria for predicting H-bonds in a crystal structure, including distance cut-offs and angular range criteria for the donor-acceptor complementary atom pairings that are required for H-bonds. The atoms first selected as donor and acceptor using the criteria and are then connected using a hinge joint.

3.3.3 Importance of constraints

While the high numbers of constraints may seem excessive, they allow for adherence to some essential protein modeling considerations. As the bond distances and bond angles remain unchanged throughout the simulation, the model recognizes short covalent bonding between atoms as invariant. Secondly, the large number of constraints implicitly incorporates rigidity considerations into the protein structure. In physical reality, H-bond's restrict secondary structure elements from becoming unstable during movement, and maintain the secondary structures (alpha helices and beta sheets). They also act as bridges between these secondary structures, and help shape the tertiary structure of a protein. Having H-bonds explicitly constraining motion in a computational model means that biologically insensible structures do not arise during conformational exploration.

This model can be compared to the 3D bond-bending network, or bar-joint model, defined by Jacobs (1998) [33]. The articulated joint-body system we present has been previously used to model ligands and small molecules [57]. But previous modeling of macromolecules with such a model has often been coarse-grained. For example, Thomas et al. (2007) developed an ensemble generation method that had to be guided by rigidity constraints using the tertiary pebble game algorithm [66]. The use of the pebble game algorithm has predominated rigidity considerations in protein constraint modeling [2, 34].

The model presented here innately considers rigidity constraints due to the inclusion of disulfide and H-bond linkages between atoms bodies. That is, by explicitly considering these linkages as joints between atom bodies, the structural stability of the

rigid portions of the system is maintained. For example, the atoms of an alpha helix secondary structure within the protein may be modeled as rigid with respect to one another. In other models, violation of such restrictions would cause the removal of a conformation from the final ensemble only after the generation of the improperly formed conformation. In the current model, however, we assume that such conformations are never generated. Post-generational assessment and scoring would need to be explicitly planned and executed, but with the inclusion of H-bonds between the atoms in the backbone of the helix, rigidity is maintained naturally through the constraint of the H-bonds themselves.

We continue by describing joint constraints and the iterative algorithm used for quick solutions of dynamics using such constraints.

3.4 Backbone rigid-body dynamics

3.4.1 Basic Kinematics

An excellent introduction and description of basic equations required for the simulation of rigid body motions has been compiled by Baraff (1997) [6]. For a complete description of basic rigid-body dynamics, along with implementation details and samples, we refer the reader to [6].

We consider a rigid body (or atom) as a body i with a center of mass in space defined by the function $x_i(t)$ where t is the current time in the simulation. The particle has a linear velocity defined by the function $v_i(t)$. The angular velocity is defined by $\omega_i(t)$. The mass of atom i will be m_i. Since velocity is a derivative of position, we can relate linear velocity of atom i to the position of atom i as follows:

$$v_i(t) = \frac{d}{dt}x_i(t) = \dot{x}_i(t) \tag{3.1}$$

where the overhead dot specifies differentiation, and will be henceforth used as such.

3.4.2 Joint Constraints

We describe the mathematical formulation of joints as described by Smith (2004) [59]. A joint constrains two bodies by specifying velocity constraints between them. A velocity

constraint on a rigid body specifies the velocity values that are allowable. The overall velocity of body i must contain its linear and angular velocity components and so the overall velocity vector is

$$\vec{v_i} = \begin{bmatrix} v_i & \omega_i \end{bmatrix}^T = \begin{bmatrix} v_{ix} & v_{iy} & v_{iz} & \omega_{ix} & \omega_{iy} & \omega_{iz} \end{bmatrix}^T \tag{3.2}$$

where the linear velocity of body i is described by three v components of velocity along the three axis in $\mathbb{R}^3$, and its angular velocities are described by the ω notation.

The Jacobian is the matrix of first order partial derivatives of a vector with respect to another vector (often two vectors of different dimensionality). Used in the calculations of constraint dynamics, the Jacobian is the coefficient matrix of the velocity matrix $\mathbf{V}$, which stores the $\vec{v_i}$ vectors of the system. The vector $\vec{c_i}$ holds the velocity constraints. In the case of a protein backbone chain, these constraints will disable certain types of movement between bound atom bodies, while allowing other types of motion. We describe the mathematical formulation of the Jacobian constraint matrix, presented by Smith [59], as $\mathbf{J}_i\vec{v_i} = \vec{c}$, with $\mathbf{J}_i$ representing the Jacobian of body i, $\vec{v_i}$ its velocity vector, and $\vec{c}$ the velocity constraint vector.

The constraint equation for body i with s_i constraints is then expressed as:

$$\mathbf{J}_i\vec{v_i} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ J_{s_i1} & J_{s_i2} & J_{s_i3} & J_{s_i4} & J_{s_i5} & J_{s_i6} \end{bmatrix} \vec{v_i} = \begin{bmatrix} c_1 \\ \vdots \\ c_{s_i} \end{bmatrix} \tag{3.3}$$

where s_i is the order of the constraint. As each constraint removes degrees of freedom from the system, s_i can also be considered as the number of degrees of freedom removed from body i. We present two examples: a simple constraint of disallowing movement along the z axis, and the constraint of limiting rotation about a specified axis (this second case to be used in the protein backbone model).

To inhibit movement (or translation) of the body i on the axis of vector $z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, one simply specifies the constraint:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \vec{v_i} = \begin{bmatrix} 0 \end{bmatrix} \tag{3.4}$$

Using equation 3.2 as the definition of the velocity matrix $\vec{v}$, the equation 3.4 holds if $(z)(\vec{v_i}[0:3]) = \begin{bmatrix} 0 \end{bmatrix}$, where $\vec{v_i}[0:3]$ represents a truncated version of $\vec{v_i}$, containing only the linear velocity components. The only way for $(z)(\vec{v_i}[0:3]) = \begin{bmatrix} 0 \end{bmatrix}$ to hold, and thus

for equation 3.4 to hold, is for the linear velocity on the axis of z to be 0. The body can then only move on the plane defined by the x and y axes. Figure 3.6 shows this scenario, where a spherical object can move along the x and y axes but shows the object as crossed out if motion occurs with a z component.

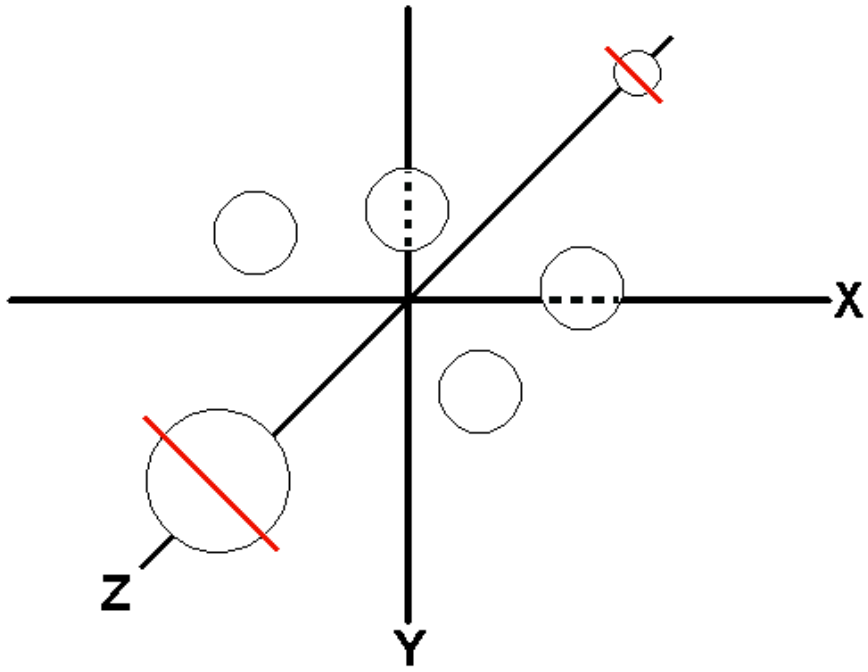


Figure 3.6: Constraining linear velocity, or motion of the rigid object, to the $x - y$ plane

In order to limit rotation about a specific axis, we must first consider the constraint of *preventing* the rotation of the body around an axis. To prevent the rotation of body i around the axis specified by vector $\vec{q} = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}$, we set the constraint:

$$\begin{bmatrix} 0 & 0 & 0 & q_x & q_y & q_z \end{bmatrix} \vec{v}_i = \begin{bmatrix} 0 \end{bmatrix}$$

This restrains the rotation of the body only along an arbitrary axis orthogonal to $\vec{q}$. Alternatively, it restricts the rotation of the body to be possible only around an arbitrary axis in the plane with a normal parallel to $\vec{q}$. This effectively flattens the space of directions in which rotation can occur. Figure 3.7 shows the possible rotations of two rectangles constrained in this way. Rotations around two arbitrary axes orthogonal to $\vec{q}$ are possible but rotations around the $\vec{q}$ defined axis itself are not.

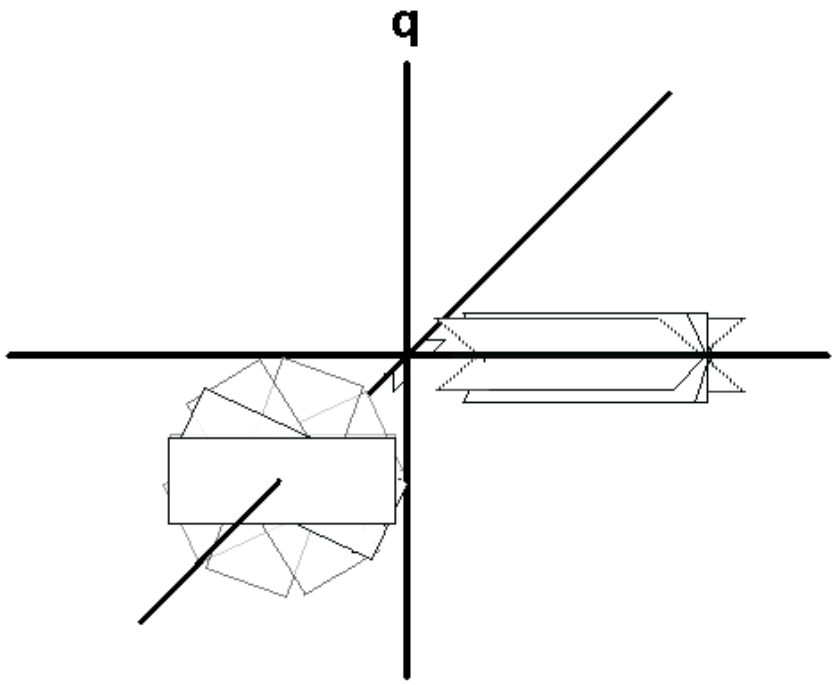


Figure 3.7: Constraining rotation of objects to axes orthogonal to the axis defined by arbitrary vector $\vec{q}$

Knowing how to prevent rotation around an axis (or to limit rotations to a plane specified by its normal), we can limit rotations to only one axis. If using one constraint as in Figure 3.7 limits rotation within a specified plane, then we can think of using two constraints to limit rotation around the intersection of two planes. Thus, if we can acquire two vectors, $\vec{o}$ and $\vec{h}$, orthogonal to our desired rotational vector (which is a simple matter covered in basic linear algebra texts), we can set two constraints to handle this requirement:

$$\begin{bmatrix} 0 & 0 & 0 & o_x & o_y & o_z \\ 0 & 0 & 0 & h_x & h_y & h_z \end{bmatrix} \vec{v}_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This constraint is required for hinge motions of two bodies in relation to each other. Using the bond between two atom bodies, we must set the hinge to be parallel to, and located at the center of, the bond. In this case, $\vec{o}$ and $\vec{h}$ are two vectors, both orthogonal to the bond axis as well as each other. Figure 3.8 shows this type of rotational constraint.

This restriction represents the motion of dihedral angles, and so this constraint is used to define bonds between atoms. A covalently bound atom is allowed to rotate only on the bond axis with respect to the atom(s) it is bound to, as shown in Figure 3.5.

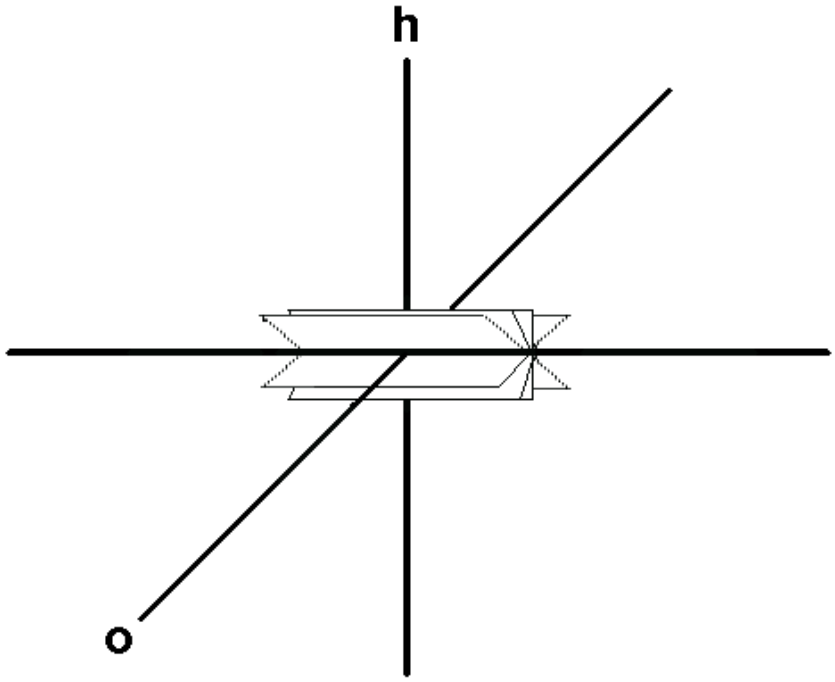


Figure 3.8: Constraining rotation of objects to only one axis orthogonal to both $\vec{o}$ and $\vec{h}$

For further description of the joint constraints, and joint constraint formulas of various other types of joints, we refer the reader to Smith (2004) [60].

3.5 Iterative dynamics

The velocity constraint model of rigid-body motion equations allows for an iterative time stepping scheme as developed by Catto (2005) [9]. This algorithm requires only linear time and space, with an O(n + s) run-time where n is the number of atoms in a simulation and s the number of constraints present in the system. We describe the general concept behind this procedure, as described by Catto (2005) [9], in the current section.

We relate the velocity vector and the Jacobian to the motion constraints (**C**) as:

$$\dot{\mathbf{C}} = \mathbf{J}\mathbf{V} = \vec{c} \tag{3.5}$$

where **J** is the s-by-$6n$ (constraints-by-velocities per body) Jacobian that must be calculated, and **V** is the corresponding matrix of velocity vectors. The velocity constraint vector, $\vec{c}$ is the vector described in Section 3.4.2.

To ensure that constraints are adhered to, internal constraint forces can be applied to the atoms in the simulation. The matrix of constraint forces $\mathbf{F}_c$ is related to the Jacobian by:

$$\mathbf{F}_c = \mathbf{J}^T\lambda \tag{3.6}$$

where λ is a vector of s multipliers that represent the signed magnitudes of the constraint forces. Atoms adhere to constraints specified on each atom by experiencing an internal force that maintains the position and orientation of each atom within an allowable state. These internal forces are applied when necessary, and so the Jacobian is evaluated at each time-step.

In general, computing the Jacobian requires that the constraint equation in **C** be developed as a function of rigid-body positions and rotations. This is followed by differentiation of the constraint equation with respect to time. Finally, the coefficient matrix of **V** must be identified (as this matrix is actually **J**).

By considering the constraints at the velocity level, as in equation 3.5, the constraint equation formulation and the differentiation can both be omitted. This is why the derivatives of the constraint functions are used as coefficients of the velocity vector in Section 3.4.2. Using velocity constraints instead of explicit constraint equations is possible due to equation 3.5, where the derivative of the constraints equations in **C** (which are functions of position and orientation) are the velocity constraints in $\vec{c}$ (both linear and rotational velocities).

Computing the new velocities of the system, $\mathbf{V}^2$, is done via a projected Gauss-Seidel (PGS) algorithm. The PGS algorithm is an iterative way of solving the linear equation $Ax = b$, which in this case is

$$\mathbf{J}\mathbf{B}\lambda = \eta \tag{3.7}$$

where

$$\mathbf{B} = \mathbf{M}^{-1}\mathbf{J}^T \tag{3.8}$$

$$\eta = \frac{1}{\Delta t}\mathbf{c} - \mathbf{J}(\frac{1}{\Delta t}\mathbf{V}^1 + \mathbf{M}^{-1}\mathbf{F}_{ext}) \tag{3.9}$$

and $\mathbf{F}_{ext}$ is the matrix of external forces applied, Δt is the time-step used in the simulation and $\mathbf{M}$ is a matrix collecting masses and rotational inertia tensors $\mathbf{I}$ (described in Appendix A) for each body along the diagonal to coincide with the linear and rotational velocities, specified by

$$\mathbf{M} = \begin{pmatrix} m_1 D & 0 & \dots & 0 & 0 \\ 0 & \mathbf{I}_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & m_n D & 0 \\ 0 & 0 & \dots & 0 & \mathbf{I}_n \end{pmatrix}$$

where D is a 3-by-3 identity matrix.

After computing λ, equation 3.10 is used to acquire the post time-step velocities of the system $\mathbf{V}^2$:

$$\mathbf{M}(\mathbf{V}^2 - \mathbf{V}^1) = \Delta t(\mathbf{J}^T\lambda + \mathbf{F}_{ext}) \tag{3.10}$$

where $\mathbf{V}^1$ are the initial velocities. The positions of all atoms are then updated using these new velocities acquired from the time-step. Catto (2005) describes the $O(s)$ running time and $O(s + n)$ space PGS algorithm, together with implementation details [9]. Due to the nature of the velocity constraint definitions in the system, contacts and collisions between atoms are treated as new constraint equations, which causes s to be a dynamic variable. However, due to the linear speed of the algorithm and the tight packing of protein structures, this fluctuation does not cause the computational requirements to become intractable.

Having the ability to run efficient dynamics simulations on this approximation of a protein structure, we can apply a variety of external forces to analyze the structural behavior of the bond network without concern over energetic clashes between atom bodies. In the current work, we use NMA, and specifically the basic ENM [68], to guide these motions.

3.6 Normal mode guided movement

NMA has previously been used in the generation of new conformations, with the modes being used as a rail on which atom movement was restricted [51]. Often, to generate new conformations of the protein, a model would relocate atoms along the normal modes of motion and rebuild the remainder of the protein around these perturbations.

The idea behind NMA is the expansion of a Taylor series of a potential energy function E about the mass-weighted coordinates of the global energy minimum $\mathbf{q}^0$:

$$E(\mathbf{q}) = E(\mathbf{q}_0) + \sum_i \frac{\partial E}{\partial q_i^0}(q_i - q_i^0) + \frac{1}{2}\sum_{i,j} \frac{\partial^2 E}{\partial q_i^0 \partial q_j^0}(q_i - q_i^0)(q_j - q_j^0) + ... \tag{3.11}$$

The first term of the expansion is set to zero under the assumption that the structure is at the global energy minimum. The first derivative term, the second term of the series, is also zero at any local or global minimum of the potential energy function, and thus is also omitted. NMA ignores the third and higher order derivatives leaving only the second order derivative. This approximation of the power series appears to be a limiting factor in the use of NMA because of the rugged nature of the potential energy landscape of protein structures. Traversing energy barriers is not immediately possible using only classical NMA and detail may be lost in calculations. However, correlation of normal modes with experimental data has been consistently shown and has made this approximation method useful, though not exact, in examining protein motions [29].

Normal mode calculations require the diagonalization of the Hessian matrix $\mathbf{H}$. The Hessian is a matrix of the second derivative of the potential energy function. With only the second order term remaining, the potential energy becomes a sum of pairwise potentials:

$$E(\mathbf{q}) = \frac{1}{2}\sum_{i,j} \frac{\partial^2 E}{\partial q_i^0 \partial q_j^0}(q_i - q_i^0)(q_j - q_j^0) = \frac{1}{2}(\mathbf{q} - \mathbf{q}_0)\mathbf{H}_{i,j}(\mathbf{q} - \mathbf{q}_0) \tag{3.12}$$

In the case of a system with n connecting nodes, $\mathbf{H}$ has $n \times n$ sub-matrices H_{ij} of partial derivatives:

$$H_{ij} = \frac{\partial^2 E}{\partial q_i^0 \partial q_j^0} = \begin{pmatrix} \partial^2 E/\partial x_i \partial x_j & \partial^2 E/\partial x_i \partial y_j & \partial^2 E/\partial x_i \partial z_j \\ \partial^2 E/\partial y_i \partial x_j & \partial^2 E/\partial y_i \partial y_j & \partial^2 E/\partial y_i \partial z_j \\ \partial^2 E/\partial z_i \partial x_j & \partial^2 E/\partial z_i \partial y_j & \partial^2 E/\partial z_i \partial z_j \end{pmatrix} \tag{3.13}$$

where each element describes the energetic contribution of the components of the displacement between atoms i and j. The symmetric **H** can be diagonalized to produce eigenvectors and eigenvalues that are the normal modes and their respective frequencies. However, the derivation of each H_{ij} component of the sub-matrix requires a lot of computation, especially when the potential energy function is as explicit as an all-atom potential. Classical NMA uses a detailed all-atom potential, such as the one used in AMBER [12]. To alleviate this, simplified models of NMA have been developed, such as the ENM [68].

3.6.1 The basic elastic network model

To guide our conformation generation method, we use a coarse-grain ENM model with a single parameter elastic network potential that uses alpha carbons as the interaction sites. This is a simplified potential employed by Tirion in 1996 [68], and later extended by Atilgan et al. (2001) [4]. The ENM has been previously used to develop alternate protein conformations [75]. In this model, contact points are the alpha carbons of the protein and the interactions between these contact points are replaced with harmonic springs with a single spring force constant.

Given a single force constant k between alpha-carbon atoms i and j within a distance threshold (R_c = 12Å), the Hookean pairwise spring potential is:

$$E(i,j) = \frac{k}{2}(d_{ij} - d_{ij}^0)^2 = \frac{k}{2}(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} - d_{ij}^0)^2 \tag{3.14}$$

where d_{ab} is the distance between the atoms i and j, d_{ij}^0 is the original distance between the atoms assumed to be in equilibrium within the given crystal structure, and the x, y, z values are the components of the displacement vectors describing the fluctuation of the positions of atoms i and j. Since the coarse-grain model in this description uses alpha carbons as interacting sites, only the interactions between alpha carbons within a distance threshold will be used in calculating fluctuations. This coarse-graining approach has been shown to generate information on motions that coincides with experimentally determined protein motions [29]. Because of the coarse-graining, it is possible to consider large backbone motions and for proteins of moderate size to be analyzed. Conversely, calculations using classical NMA methods may become intractable with too many interacting sites.

The total energy for the entire molecule becomes:

$$E_{ENM} = \frac{1}{2}k \sum_{i,j} E(i,j)\theta(R_c - |d_{ij}|) \tag{3.15}$$

where θ is the binary Heaviside function requiring the distance between the interacting sites to be within the cutoff distance R_c.

As a consequence of using Equation 3.14 as the energy function, the elements of the Hessian matrix are simple to calculate [4]. The first derivative of E with respect to the components of the equilibrium position vector of atom i is:

$$\frac{\partial E}{\partial x_i} = -\frac{\partial E}{\partial x_j} = -k(x_j - x_i)(1 - \frac{d_{ij}^0}{d_{ij}}) \tag{3.16}$$

$$\frac{\partial^2 E}{\partial x_i^2} = -\frac{\partial^2 E}{\partial x_j^2} = k(1 + d_{ij}^0(x_j - x_i)^2/d_{ij}^3 - \frac{d_{ij}^0}{d_{ij}}) \tag{3.17}$$

with similar expressions holding for the y and z components of the equilibrium position vectors. At equilibrium $d_{ij}^0 = d_{ij}$ and the Equations 3.16 and 3.17 become:

$$\frac{\partial E}{\partial x_i} = 0 \tag{3.18}$$

$$\frac{\partial^2 E}{\partial x_i^2} = k(x_j - x_i)^2/d_{ij}^2 \tag{3.19}$$

The cross derivatives and elements of H_{ij} then become the simplified:

$$\frac{\partial^2 E}{\partial x_i \partial y_j} = -\frac{\partial^2 E}{\partial x_j \partial y_i} = -k\frac{(x_j - x_i)(y_j - y_i)}{d_{ij}^2} \tag{3.20}$$

After the Hessian is constructed, diagonalization is used to solve for the normal modes of motion. **H** is diagonalized by

$$\mathbf{H} = \mathbf{E}_m \mathbf{U}_m \mathbf{E}_m^T \tag{3.21}$$

where H is the Hessian, $\mathbf{E}_m$ are the eigenvectors containing normal modes, and $\mathbf{U}_m$ are the frequencies of the modes. Equation 3.21 provides the easy acquisition of normal modes and frequencies through the eigendecomposition of the Hessian matrix. The six

lowest frequencies and corresponding modes describe translations and rotations in three dimensions and are thus non-essential modes of motion, and therefore ignored. The lowest frequency modes following the non-essential modes are the important modes of motion which, as previously explained, correlate well with experimentally determined large protein motions.

3.6.2 Normal modes as external forces

We use the normal mode directions provided by the ENM to direct motion in the rigid-body model of the molecule. A force equivalent to:

$$f_i = \alpha e_i \tag{3.22}$$

is applied to the relevant atoms of the molecule. In equation 3.22, e_i is the eigenvector (or the mode) of atom i used in the ENM, gained from the diagonalization of the Hessian matrix. α is an experimentally defined multiplier value tunable by the user. We found that using the length of the protein (number of residues = α) appears to work well in this context, though the value itself is arbitrary. When using the basic ENM, this force is applied to the alpha carbons, in the direction provided by the normal mode e_i. An example of normal mode directions is shown in Figure 3.9.

In order to explore the conformational space specified by the normal modes, the model is modified by such forces for a few non-trivial modes. In our application, we use the first three non-trivial modes with the lowest related frequencies. Low frequency motions are the motions describing large collective motions of the protein [29]. If the normal modes are ordered based on their frequencies, from low to high, this means we use modes 7, 8, and 9. Normal modes 1-6 are trivial (they describe rotations and translations in 3-dimensional space) and are thus omitted during this exploration [44]. While it is possible to use normal modes beyond 9, these modes begin to exhibit higher frequency values and will eventually fail to describe large scale motions. This is important when using an ENM, as we do not take side-chain motions into account with a coarse-grained ENM.

For each of the non-trivial modes selected, the force is applied to each alpha carbon in the positive and negative direction of the normal mode specified. Because normal modes are harmonic in nature, normal mode directionality requires that both available choices (positive and negative directions of normal modes) are used to guide motion simulations. According to Thorpe (2007), the symmetric properties of the ENM potential allow the assumption that nearby atoms move together either in the positive or negative directions

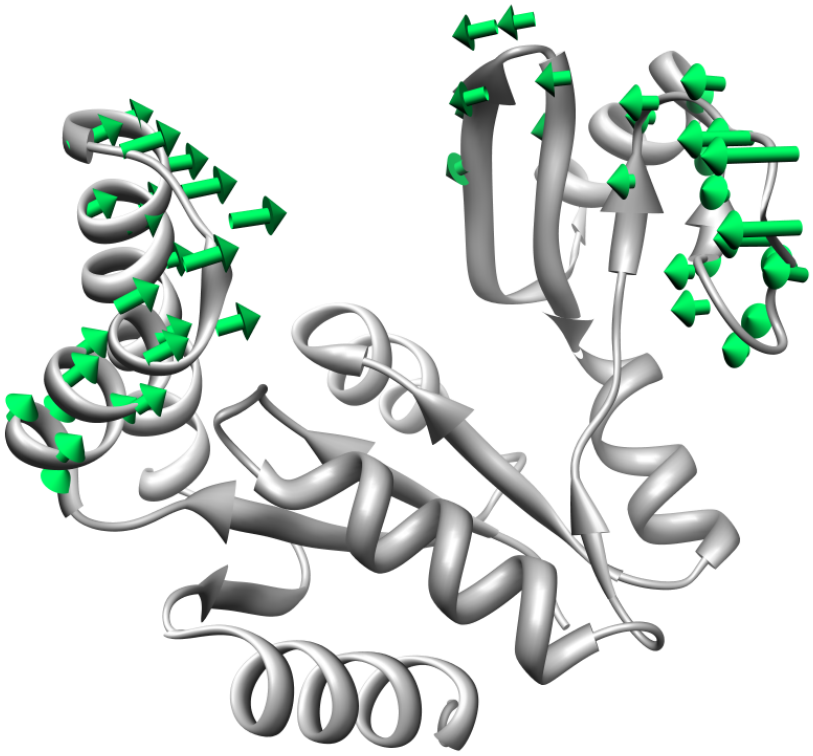


Figure 3.9: The base conformation of Guanylate kinase (PDB ID: 1EX6), with arrows indicating the direction of the first non-trivial normal mode of the alpha carbons. Only the modes with largest magnitude are shown.

of a given normal mode [67]. This is due to using only the displacements of atoms in the potential function of equation 3.15. As the function uses only the displacements, and because of the quadratic nature of the potential function, translational invariance is inherent within the ENM. The sum of all Hookean potentials is zero when the motion is translational at the macromolecular scale (when the entire molecule is translated as one rigid body). This can be extended into a discussion of acoustic modes that describe the motion of neighboring atoms moving in approximately the same collective direction describing a low-frequency behavior of the protein [67, 3]. Thorpe stipulates the assumption that nearby atoms move in the same direction, but while experiencing small shifts in directionality of their vibrations along the normal mode [67]. Furthermore, as the low-frequency normal modes represent directions of slow, collective motions of the protein, we can assume these directionality shifts are not detrimental to the

overall collective motion of the atoms in one vibratory direction. Thus, we can avoid a combinatorial explosion caused by selecting all combinations of positive and negative directions for all interacting sites and assume nearby interacting sites experience motion in similar directions.

The application of this force is a single event at the beginning of the simulation (not an impulse), and so acts as the initialization of the motion of the structure. Even though only the alpha carbons that experience this force, due to the connectivity of the model and the joint constraints, the remainder of the backbone is also modified. If side-chains are placed in the model as well, they also experience this motion due to the connectivity of bonds. However, using side-chains would cause clashes between side-chain atoms which would invalidate the analysis on the intended time scale in such a simulation. As we use the ENM to gather directions of slow collective motions of the protein, collisions between side-chains would inhibit the proper exploration of alternative backbone conformations based on those collective motions.

3.7 Side-chain addition and energy minimization

The newly created backbone conformations lack side-chains. Side-chains are added using SCWRL4 [40]. This is followed by energy minimization using the Molecular Modeling Toolkit (MMTK) [31] suite included in UCSF Chimera [49] in order to optimize the backbone and side-chain geometries. This energy minimization method is based on the conjugate gradient method of minimization, in which steps are taken along the downward gradient direction in order to find the lowest point of an energy well. Few minimization steps are necessary to achieve a low energy structure from the members of the generated ensembles. In many cases of generated structures, negative MMTK energy scores are generated after less than 100 steps in the MMTK conjugate gradient minimization algorithm [31]. The energy calculations use the AMBER forcefield and all energy values are in kJ/mol [12].

3.8 Implementation details: summary of tools used

The requirements of rigid body simulations include a dynamics engine with a stable integrator. We use the Open Dynamics Engine (ODE) by Russell Smith as a base for our rigid body simulations. ODE facilitates modeling of rigid bodies and joint contacts

and uses a stable integrator to step through the simulation [60]. While other software packages for physics simulations exist, ODE is considered one of the most popular and has been used in commerical and research contexts, often in robotics simulations [17].

A comparison of ODE to other available physics engines is presented in Boeing and Bräunl (2007) [7]. While achieving average result stability, ODE outperformed other tested packages on constraint error tests when configured for simulation with the included Euler integrator. ODE includes the projected Gauss Seidel algorithm, with a $O(s)$ run time for a modeled system where s is the number of constraints, or joints, at a given timestep in the system.

The PyODE Python bindings available for the ODE [32] ease scripting with the physics engine, and the UCSF Chimera Molecular Graphics program which allows use of the bindings with protein structure files [49].

The elastic normal modes were calculated by the ModeHunter package [63] with the R_c cut-off of 12.0 Å.

As the steps described within this chapter generate backbone structures of proteins, we must re-pack amino acid side-chains for a full atom model to exist. We use the SCWRL4 side-chain packing program due to its speed and side-chain prediction capabilities [40]. A leading tool in the area of side-chain prediction, SCWRL4 uses a tree decomposition algorithm as well as potential functions and averaging to select rotamers for side-chain placement [40].

The MMTK package is used for energy minimization of the generated all-atom structures [31]. The conjugate gradient minimization algorithm is applied, using the AMBER 94 force field [12].

Chapter 4

Results and Discussion

4.1 Ensemble generation from unbound protein structures

Alternative conformations of protein structures are useful in computational studies of ligand binding. Docking studies benefit from the use of conformations with a structural overlap close to that of the bound structure. Studies have shown that virtual screening using only an unbound structure results in poorer ability to predict binders and non-binders in a set of ligands when compared to the use of bound receptor structures [45, 73]. In this section, we show the rigid-body backbone model can be used to generate alternate backbone conformations from apo (unbound) structures. These new conformers are then compared to their holo (bound) counterparts and their structural likelihood is analyzed.

4.1.1 Data set

The protein structures are sampled from the dataset used by Seelinger & Groot (2010) [54]. These include a selection of proteins that undergo large conformational changes upon ligand binding. The families of the receptors used, and the Protein Data Bank (PDB) ID's of the unbound structures and their ligand-bound target structures are:

- Periplasmic-binding protein, which have a wide range of fundamental functions
  - D-Allose binding protein (ALLO), apo: 1GUD, holo: 1RPJ
  - Osmo-protection protein (OSMO), apo: 1SW5, holo: 1SW2

  - D-Ribose binding protein (RIB), apo: 1URP, holo: 2DRI

- Actin-like ATPase domain
  - DNA Beta-Glucosyl-transferase (GLUCO), apo: 1JEJ, holo: 1JG6
  - Hexokinase (HEXO), apo: 2E2N, holo: 2E2O

- P-loop containing nucleoside triphosphate hydrolase
  - Guanylate kinase (GUAN), apo: 1EX6, holo: 1EX7

4.1.2 Qualitative exploration of approximating bound structures through ensembles

A small ensemble was created using recent crystallizations of the Guanylate kinase unbound (apo) form structure. The backbone model of the apo-form of GUAN was built using the method described in Chapter 3. The model was then modified by application of forces along the directions of the normal modes, which were calculated for the crystallized conformation using the original ENM described by Tirion [68] with alpha carbons as interacting sites. This force application caused a continuous conformational change in the protein backbone. The backbone positions were recorded at consecutive, arbitrary, time-step intervals during this process. The first 3 non-trivial, low-frequency normal modes were used to guide the force application and new conformations were recorded for each normal mode setting. The resulting conformations were visually compared to the holo forms of the protein. Figure 4.1 shows the holo and apo forms and the generated structures. The progression of the generated structures from the apo toward the holo form is especially apparent near the top of the protein in Figure 4.1, where two of the outer arms of the protein are closing down over the active site. The conformations generated using the first non-trivial normal mode appear to move their outer arms beyond the holo-form of the structure after passing close to the target conformation.

4.1.3 Beginning a quantitative comparison of ensembles to target structures

Root mean square deviation (RMSD) is a function used as a comparison to measure the difference between two conformations of the same protein. The RMSD is a measure of

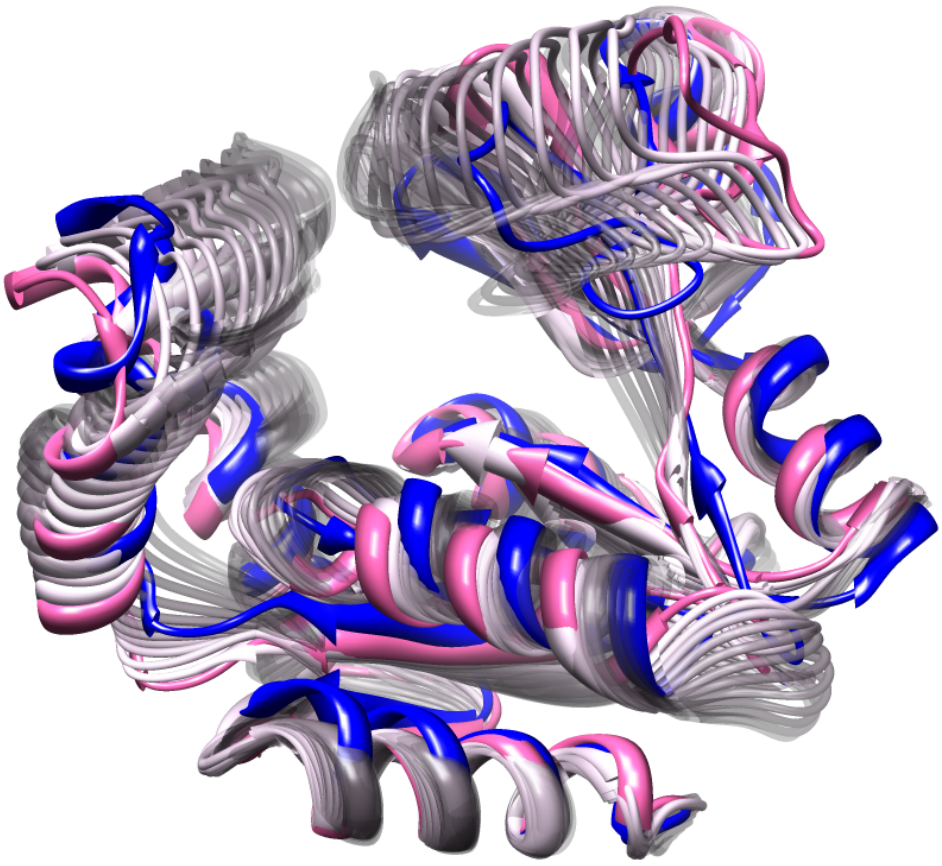


Figure 4.1: GUAN apo form (pink) and holo form (blue). Structures generated from the first non-trivial normal mode are shown in faded grey.

structural overlap between two sets of atoms. For two protein structures, the RMSD can be calculated as:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}||t_i - w_i||^2} \tag{4.1}$$

where n is the number of alpha carbons (or the atoms used in the analysis) and t_i and w_i are the vector positions of the atoms in the protein chains being compared (for example, t_1 as the position of the alpha carbon of the first residue of the apo-form protein and w_1 as the position of the alpha carbon of the first residue of the holo-form of the

same protein).

A small conformational ensemble (nine structures, with three generated using each of the first three non-trivial normal modes) of Guanylate kinase was compared to the holo-structure with Equation 4.1. As shown in Table 4.1, the procedure finds backbone conformations with a structural overlap closer to the holo-form than a structural overlap between the holo-form and the apo-form. Conformations that result in an RMSD lower or equal to that of the original pair of the 1EX6 apo-conformation and the 1EX7 holo-conformation are noted with an asterisk (*).

While the RMSD between the apo and holo conformations of Guanylate kinase is 3.640 Å, our model manages to generate a backbone conformation with an RMSD to the holo target structure of approximately 2.2 Å. We extend the test to generate more structures from the apo-form (PDB ID: 1EX6) and compare them to the holo-form bound to GMP ligand (PDB ID: 1EX7), in Section 4.2.1, alongside other apo and holo protein pairs.

Table 4.1: RMSD values between different conformations of 1EX6 (Guanylate Kinase apo-form) and 1EX7 (Guanylate Kinase bound to GMP ligand).

Mode	Conformation	RMSD (Å)
1	1*	3.597
1	2*	3.447
1	3*	3.195
2	1	3.642
2	2	3.659
2	3	3.710
3	1	3.651
3	2	3.696
3	3	3.791
-	Original 1EX6	3.640

4.2 Numerical results

Backbone structures were generated using each of the apo protein listed in Section 4.1.1. Each simulation encompassed the generation of 26 conformations for each of the first

three non-trivial normal modes, for a total of 78 structures. This was followed by side-chain repacking for each generated protein conformation and energy minimization for each conformation as described in 3.7.

4.2.1 Energy and structural similarity to bound structures

The structures in ensembles generated from apo structures had the RMSD from the holo structure calculated using the alpha carbons of the protein backbone of each model. The all-atom potential energy of each structure in the ensemble was calculated using the AMBER94 forcefield[12] as described in Section 3.7.

Figure 4.2 shows graphs of the values of structural overlap of each member of the generated ensembles to the target holo structures on the x-coordinate, with the potential energy along the y-coordinate. All of the resultant ensembles produce a somewhat linear relationship between RMSD values and potential energy values. The weak, positive association is visible due to the conformational changes causing the structure to move further away from the energetically stable conformation used as the base conformation (the apo structure). The further the new structures are from the original apo structure (and consequently the target holo structure), the more unlikely the existence of the conformation and thus the minimization steps do not produce the same amount of energetic stability in the structure.

As seen in Figure 4.2, structures with outlying values of potential energy exist in some of the ensembles: this is due to energetic and steric clashes after the addition of the side-chains to the backbone. Often, sidechain repacking may result in a small amount of steric clashes that are eliminated via small movements in the energy minimization step. In some cases, as in the outliers visible in Figure 4.2, 100 steps of the conjugate gradient minimization algorithm are not enough to decrease the energy value even below zero. In Figure 4.2 this is most apparent in the Osmo-protection protein (OSMO) graph. Increasing the number of minimization steps removes this concern. The maximum of 100 was used in this experiment in order to show the efficiency of the model in creating few geometrically unrealistic situations, and for many of the models in the ensembles, fewer than 30 minimization steps are necessary to gain a negative energy value. In past research, conjugate gradient minimization is employed with steps numbering in the thousands [38], whereas our method does not require such a high number of steps to achieve a structure with a low potential energy value.

The results of structural overlap of the ensemble members and the target structure depend on the protein under analysis. Some of the conformations achieve a lowest

Figure 4.2: The all-atom energy values vs. the RMSD from target structure for the ensembles generated by the apo form of the protein structures. Ensemble conformers appear in blue, original apo-form in purple, and original holo-form in green.

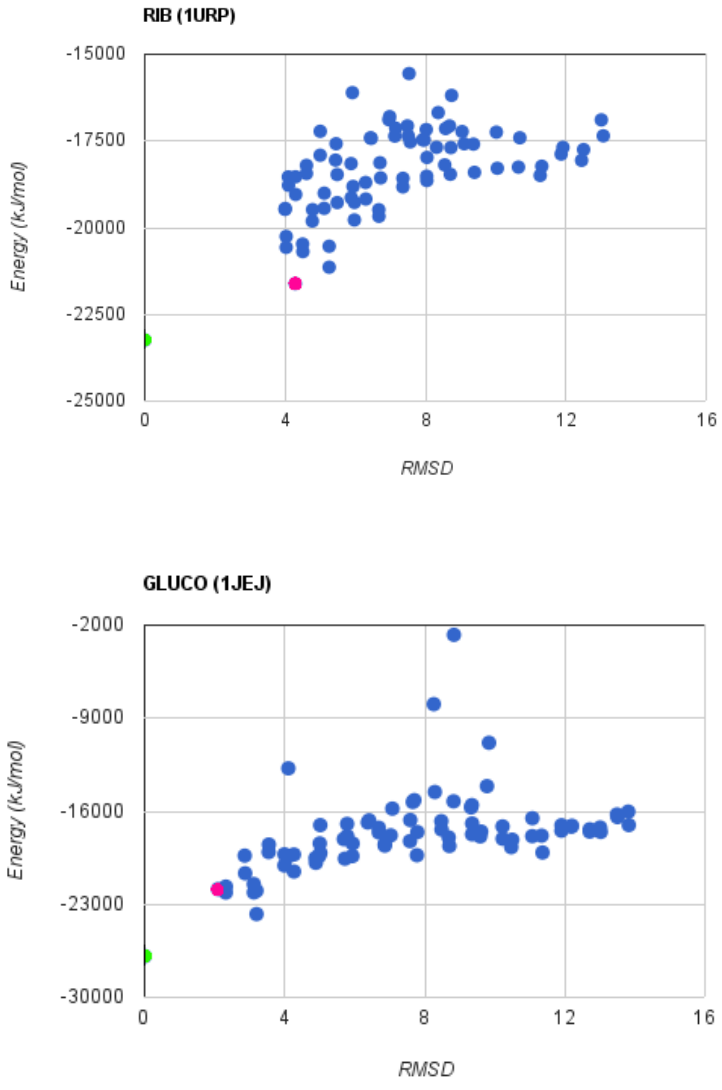


Figure 4.2

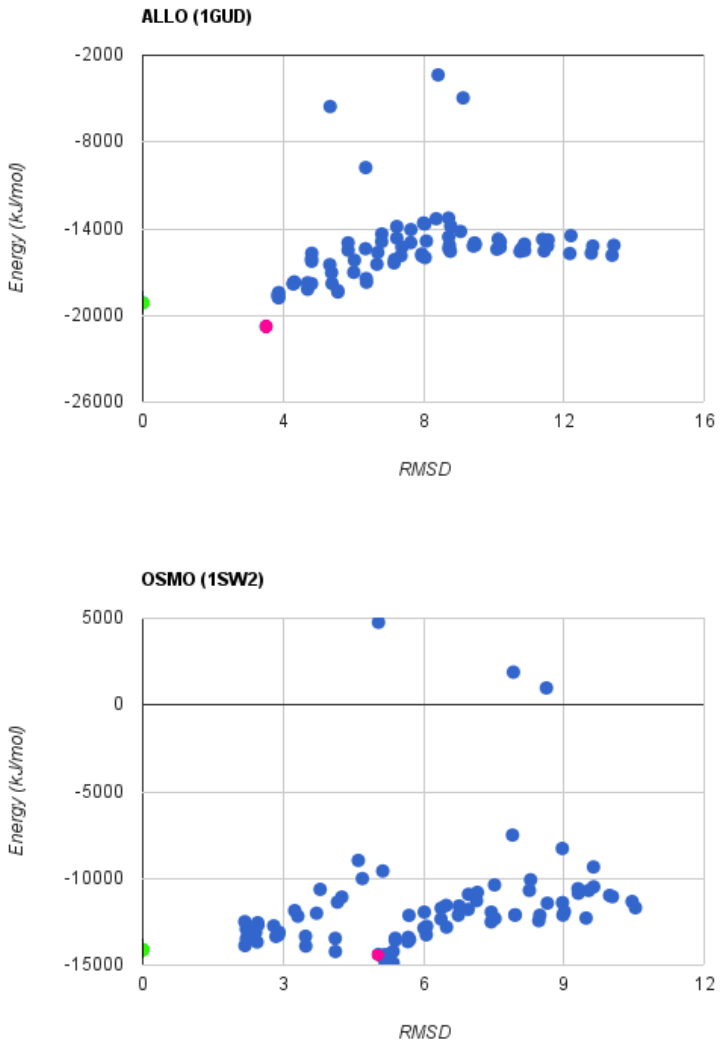

ALLO (1GUD)
-2000
-8000
-14000
-20000
-26000
Energy (kJ/mol)
0
4
8
12
16
RMSD
OSMO (1SW2)
5000
0
-5000
-10000
-15000
Energy (kJ/mol)
0
3
6
9
12
RMSD

Figure 4.2

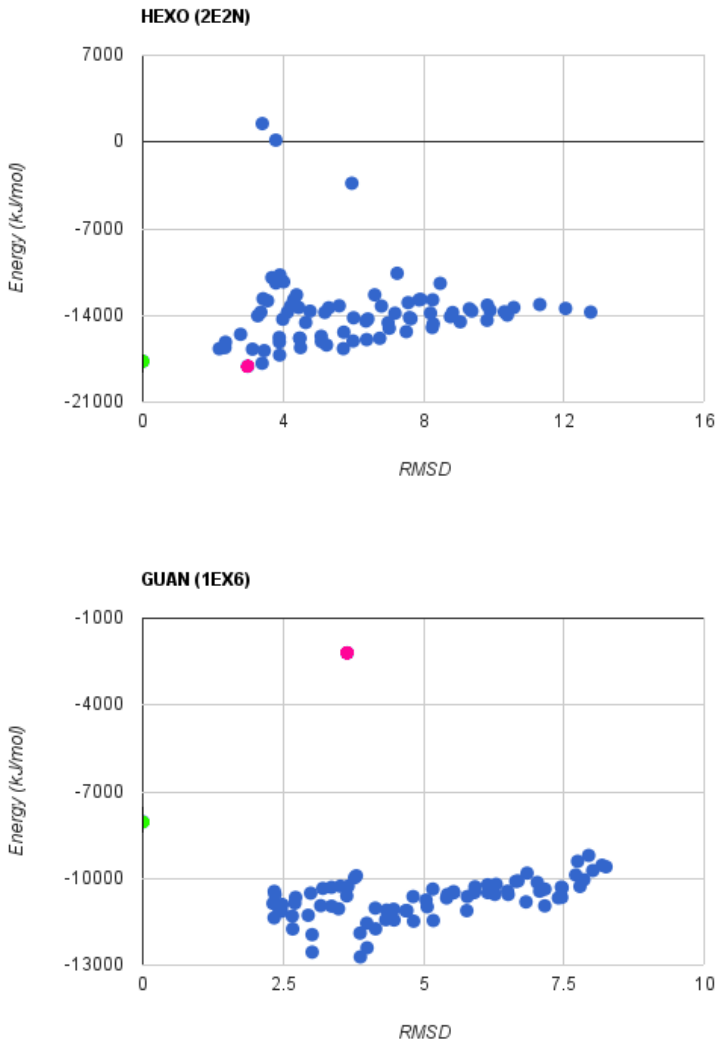

HEXO (2E2N)
7000
0
-7000
-14000
-21000
Energy (kJ/mol)
0
4
8
12
16
RMSD
GUAN (1EX6)
-1000
-4000
-7000
-10000
-13000
Energy (kJ/mol)
0
2.5
5
7.5
10
RMSD

Figure 4.3

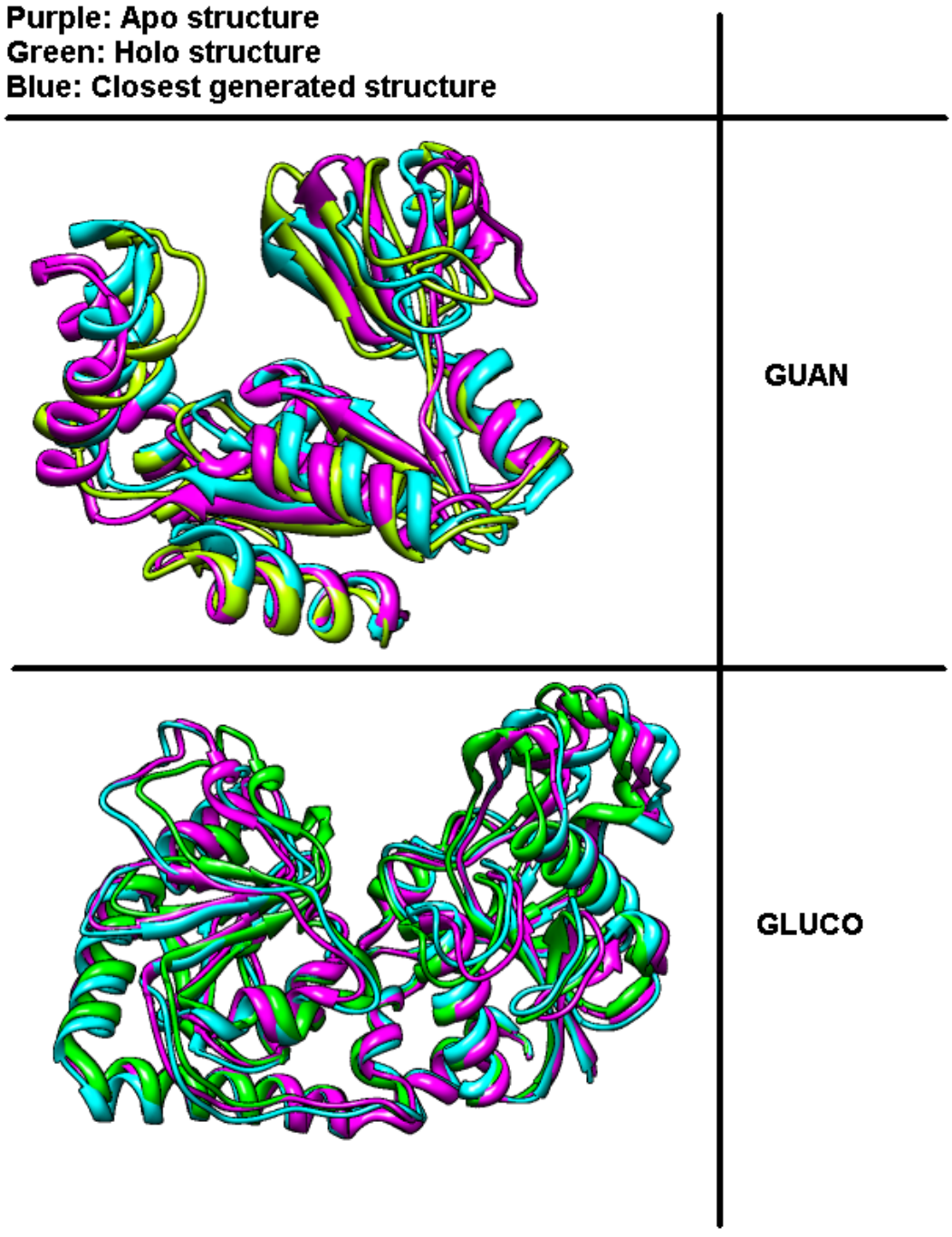

Purple: Apo structure
Green: Holo structure
Blue: Closest generated structure
GUAN
GLUCO

Figure 4.3

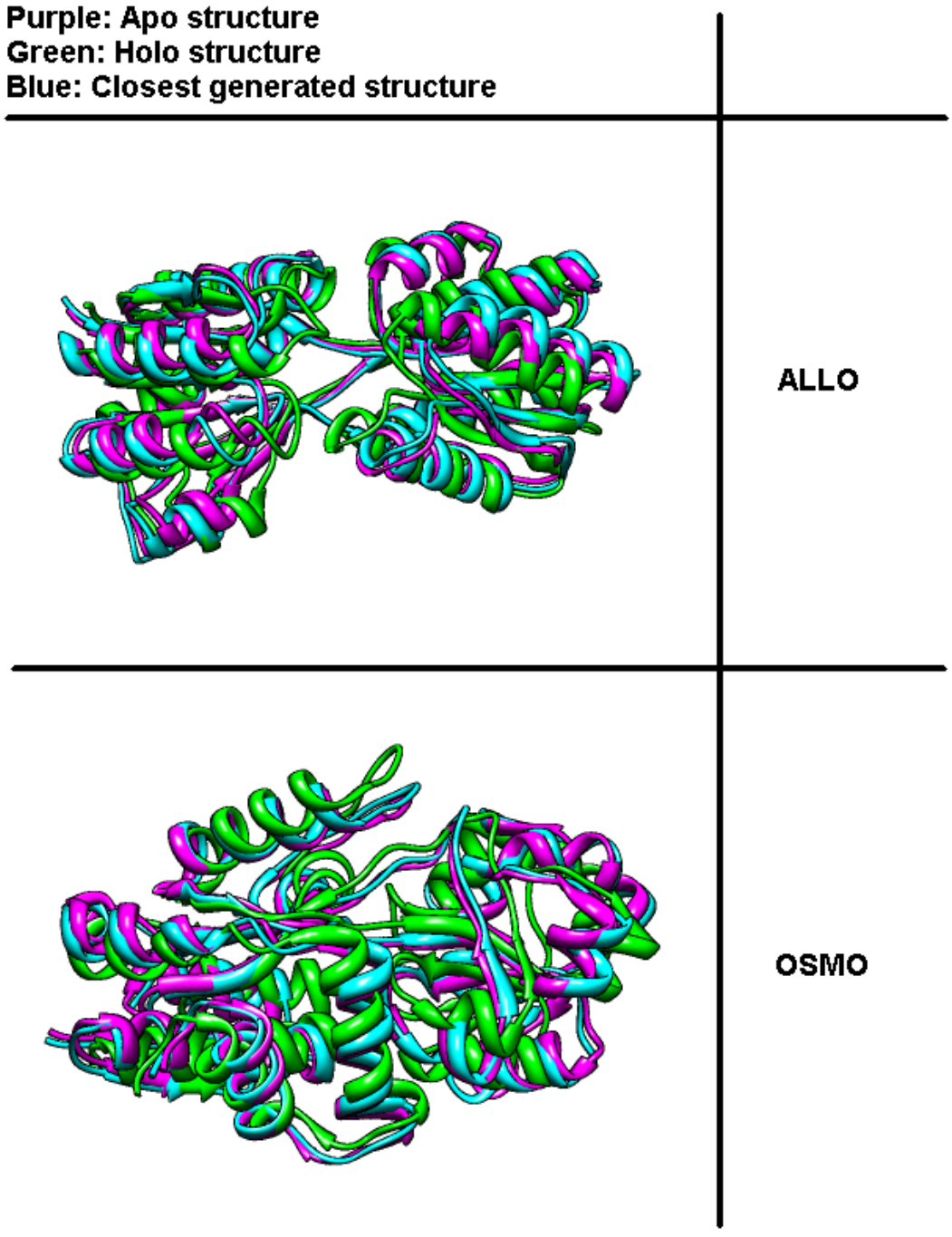
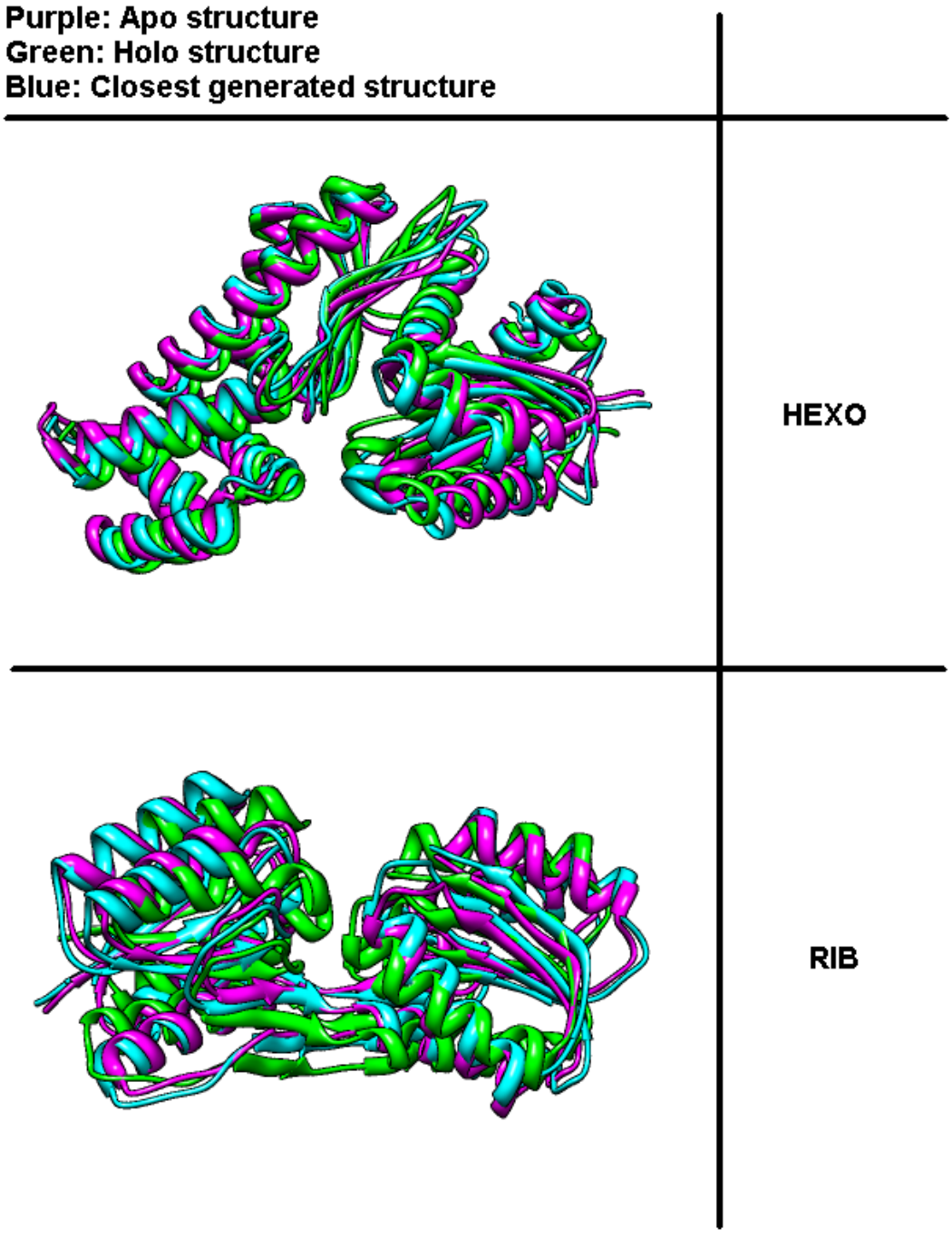

Purple: Apo structure
Green: Holo structure
Blue: Closest generated structure
ALLO
OSMO

Figure 4.3


Purple: Apo structure
Green: Holo structure
Blue: Closest generated structure
HEXO
RIB

RMSD (LRMSD) closer to the holo-form than their apo-form origin structure. Specifically, GUAN, RIB, HEXO, and OSMO, all form ensembles with LRMSD's lower than their apo-holo form RMSD's. These apo-form to holo-form RMSD's are 3.6 Å, 4.3 Å, 3.0 Å, and 5.0 Å, respectively. The apo structure of the Osmo-protection protein was used to generate a structure resulting in the largest improvement in structural overlap with the holo structure. The LRMSD for OSMO was 2.2 Å. This shows an improvement of 2.8 Å from the original unbound to bound conformation structural overlap. The Guanylate kinase ensemble yields the LRMSD of 2.3 Å, with an overlap improvement of 1.3 Å from the original value of 3.6 Å. The improvement in overlap between the Hexokinase bound structure and the closest member of its respective ensemble was 0.8 Å. D-Ribose binding protein had a lower improvement in structural overlap, with the LRMSD being 4.0 Å which is a gain of approximately 0.3 Å.

The other ensembles did not contain structures with structural overlaps closer to their target holo structures. However, all ensembles hold structurally diverse (as suggested by their RMSD values) and energetically viable (as suggested by the energy values) conformations.

Table 4.2: Lowest RMSD's to holo target yield from generated ensemble

Name	APO PDB	HOLO PDB	Residue Count	Orig. RMSD (Å)	LRMSD (Å)	LRMSD Energy (kJ/mol)
HEXO	2E2N	2E2O	298	2.989	2.180	-16682.06
GLUCO	1JEJ	1JG6	351	2.099	2.318	-22111.87
OSMO	1SW5	1SW2	270	5.035	2.181	-12490.26
ALLO	1GUD	1RPJ	288	3.515	3.852	-18547.03
RIB	1URP	2DRI	271	4.288	3.998	-19470.69
GUAN	1EX6	1EX7	186	3.640	2.317	-10863.86

Compared to Seelinger and De Groot's (2010) ensemble generation via the tCOONCORD program (Table 4.3), the LRMSD yields generated by our rigid-body backbone model are higher, making the change in structural overlap inferior. The tCOONCORD work managed structural overlap results below 2.0 Å in the ensembles generated [54]. While some of the ensembles we generated using our method approach the 2.0 Å LRMSD value, most do not reach this magnitude of improvement of structural overlap between apo and holo protein structure pairs, and none of the ensembles contain members which achieve an RMSD of below 2.0 Å.

We used FRODA to generate 500 conformers from the apo structures. FRODA is a ensemble generation software which uses the pebble game in order to define rigid clusters within a protein structure, followed by randomized Monte Carlo dynamics to change the conformation [74]. Table 4.3 shows the LRMSD's calculated for each ensemble by our approach, tCONCOORD, and FRODA. In some of the cases, our approach results in ensembles with an LRMSD lower than that of FRODA (Hexokinase, Osmo-protection protein, and Guanylate kinase), but in others FRODA manages better results, though still not comparable with tCONCOORD.

Table 4.3: Lowest RMSD's to holo target yield from generated ensemble, comparison to other methods

Name	LRMSD (Å)	tCOONCORD	FRODA
HEXO	2.18	1.42	2.90
GLUCO	2.32	1.38	2.12
OSMO	2.19	1.27	4.54
ALLO	3.852	1.07	3.37
RIB	3.998	0.98	2.52
GUAN	2.317	1.45	3.64

While our method generates some conformers that have a structural overlap closer to the target holo structure than the original apo structure, in some cases the results are not competitive with previous methods. The reasons for this may include the obvious limitations of our experimental procedure: only the first three non-trivial normal modes are used, and combinations of normal modes are not considered. Additionally, the normal modes may not contain information on conformational changes between unbound and bound structures as various modes of binding may exist. If conformational changes between holo and apo structures are achieved by passing through more than one low energy state, the protein structure may transiently exist in a local energetic minimum different from that of the original structure. In such a situation NMA approaches such as the one used presently may require further refinement such as the rebuilding of the ENM at some steps during the simulation. Additionally, the constraints set by our model do not change over time, which limits the conformational space accessible by our model. H-bonds are not dynamic in our model of the backbone, and so conformational changes where the H-bonds change would not be explored.

As the timesteps of the simulation are arbitrary due to NMA not being a time-based indicator of motion, it is difficult to decide how often structural snapshots for the ensemble should be taken. Additionally, if the model is being used to find a target structure, as it was in the apo and holo structure comparison studies shown both in the current work and previously [54], it is difficult to decide on the duration of the simulation. For example, we allow FRODA to generate 500 conformations instead of the 78 generated by our method. The results from the tCONCOORD test are based on an experiment that generated 1000 conformations before reducing the ensemble size after energy minimization and constraint adherence methods [54]. Though our generated conformers are fewer in number, some of the ensembles contain conformers with a good structural overlap to the target holo structure.

The RMSD results indicate a large displacement of the backbone when compared to the displacement seen in FRODA results. FRODA dynamics seem to constrain the mobility of the backbone to approximately 1 Å from the apo structure, whereas our method manages maximum RMSD's of approximately 7-10 Å. Though tCONCOORD generates structures with good structural overlaps to the holo target structures, their use of the radius of gyration of the holo structure as input has some impact on the effectiveness of the method, and consequently on their results. Both FRODA and our approach use only the apo conformation as input in the ensembles we generated.

4.2.2 Time requirements

In the implementation of the method presented in this work, the minimization step requires the majority of the time taken for structure generation. All ensembles generated in this work were constructed using a 2.91 Ghz AMD Athlon dual core machine with 4 GB of RAM. Each conformation requires approximately 10 seconds to be generated without minimization (this includes generating the extraneous backbone conformation and skipping conformations which are too similar to one another in terms of structural overlap) and an extra 2 minutes with minimization. Reducing the need for minimization steps is important and using geometrical and steric constraints present in the rigid-body model calculations in this work alleviates the need for extensive minimization.

When compared with the tCONCOORD method, which requires 4 days per case on a 50 node cluster [54], our approach is more tractable. The multiple steps and MD refinement used by the tCOONCORD work cause a high computational and time requirement. While the actual conformational sampling by tCONCOORD may take much less time, the MD requirements of their proposed workflow cause the full method to suf-

fer from high time expenditures. Each of the cases in our work requires approximately 3 hours to generate 78 structures on average. The majority of this time was used for the 100 conjugate gradient energy minimization steps for each generated protein structure.

4.2.3 G-factor scores

We use the G-factor as one of the indicators of goodness of the generated structural ensembles. The G-factor is calculated as the log-odds score based on stereochemical properties [42]. It is calculated based on the observed distribution of these stereochemical properties from a set of 163 high resolution crystal structures (solved by X-ray crystallography to a resolution below 2.0 Å) [18]. For each residue, the G-factor is the combination of the log-odds ratio of the torsion angle properties of the given residue (the phi and psi angles, chi angles when applicable, and the omega angle) and the log-odds ratio of the covalent geometry properties (backbone bond lengths and backbone bond angles) of the given residue. A G-factor below -1.0 indicates an unreliable residue score and suggests that the properties do not coincide with the distribution seen in the high resolution set of structures used to develop the scoring function.

The program PROCHECK-NMR [41, 42] is used to calculate the G-factor for each residue in the generated models. PROCHECK-NMR is designed to verify the structures of NMR ensembles and is therefore a good candidate for the quantification of protein ensemble quality.

Table 4.4 shows the resulting average G-factors in the generated protein structures. G-factors averaged over residues for each generated ensemble are shown in the first column and percentages of residues with average G-factors below the -1.0 threshold are shown in the second column. All the ensembles have average G-factor scores above this threshold. However, approximately 10% of the average residue G-factors are below this threshold in all the generated ensembles. Nonetheless, some of the ensembles have an average G-factor very near that of the original apo structure (notably OSMO, ALLO, and GUAN).

4.2.4 Ramachandran plots

The Ramachandran plot is a distribution of allowable phi/psi angle pairs in a protein backbone. This two dimensional plot shows areas where the phi/psi torsional angle pairs

Table 4.4: G-factors of generated ensembles.

Name	Apo structure average G-factor	Ensemble average G-factor	% < -1.0
HEXO (2E2N)	0.42	0.04	10%
GLUCO (1JEJ)	0.32	-0.05	8%
OSMO (1SW5)	0.10	0.05	9%
ALLO (1GUD)	0.17	0.11	8%
RIB (1URP)	0.24	0.13	9%
GUAN (1EX6)	0.00	-0.04	12%

are statistically likely for specific residue types or for the entire protein backbone chain. A data point on the plot represents a phi and psi combination for a sampled residue. The phi dihedral angle is defined by the horizontal axis (ranging from [-180,+180]) and the psi dihedral is situated on the vertical axis with the same range. An area on the plot is defined as ”core”, ”allowed”, ”generous”, or ”disallowed” based on the distribution of residues in known structures having specific phi/psi combinations qualifying in the area. Protein conformations are considered to be reliable when 90% of the non-glycine and non-proline residues are within the ”core” sections.

We use PROCHECK-NMR [42] for calculation of the plots for the structures in each ensemble. The different permissive regions are originally defined within PROCHECK-NMR based on an analysis of 118 structures of resolution less than or equal to 2.0 Å.

The results are shown in Table 4.5. The majority of residues reside in the ”core”, or the most favored regions of the Ramachandran plot. The GLUCO ensemble is the only generated ensemble without at least 90% of phi/psi pairs in these ”core” regions. The additionally and generously allowed regions contain approximately 10% of residues in each ensemble. None of the ensembles contain more than 2% of residue dihedral angles within disallowed regions. The ALLO and RIB ensembles both have 1.1% of residues in the disallowed regions.

In the rigid-body joint constraint backbone model, it is only the dihedral angles that experience changes due to the application of forces to the atoms. It is important that these changes avoid creating unlikely dihedral angles in residues of the generated protein structure. While the results of the ensemble generation presented in Table 4.5 are reasonable, some of the dihedral angles in the ensembles are in disallowed regions of the Ramachandran plot. From visual inspection of the Ramachandran plots, the majority of

Table 4.5: Ramachandran plot location percentages of the generated ensembles.

Name	Method	Core	Allowed	Generous	Disallowed
HEXO (2E2N)	Joint	91.4%	8.0%	0.5%	0.1%
	FRODA	89.8%	9.3%	0.5%	0.3%
GLUCO (1JEJ)	Joint	87.5%	11.4%	0.7%	0.4%
	FRODA	83.7%	15.0%	1.0%	0.3%
OSMO (1SW5)	Joint	90.8%	8.1%	0.9%	0.1%
	FRODA	86.6%	12.8%	0.5%	0.1%
ALLO (1GUD)	Joint	91.4%	7.4%	0.1%	1.1%
	FRODA	87.5%	10.9%	0.7%	0.9%
RIB (1URP)	Joint	90.9%	7.6%	0.4%	1.1%
	FRODA	86.4%	12.1%	0.6%	1.0%
GUAN (1EX6)	Joint	90.9%	8.2%	0.6%	0.3%
	FRODA	85.9%	13.8%	0.3%	0.0%

these outliers are from late-stage models, i.e. structures that have experienced motion in the directions of a given normal mode over many time steps. This implies that permitting the simulation to achieve extensive structural change in the direction of a single normal mode will result in unlikely structures. Permitting this extent of structural changes in the conformation may also be responsible for the G-factor results shown in Section 4.2.3 as the G-factor scores are also dependent on the phi and psi angles of the residues.

The Ramachandran plots of the ensembles generated by FRODA were analyzed using PROCHECK-NMR in order to compare the results of our model to an already established method that strongly considers rigidity. We compare the location percentages in Table 4.5 and in all cases, our method produces a better Ramachandran plot distribution for each ensemble than the FRODA method, which has residues in the core regions consistently under 90% in all ensembles it generated. This implies that our approach produces more geometrically likely structures, at least in terms of phi and psi angle pairs.

4.2.5 Bond length and bond angle stability

The PGS algorithm acquires the new velocities of objects in order to update the new positions of the objects [9]. Such an approach, in which the new velocities are used to update positions, is referred to as a semi-implicit, or a symplectic Euler integration scheme. It has been shown that this scheme has stability and energy conservation properties that are comparable to the Verlet integration scheme [27]. The Verlet integration scheme is the most commonly used integration scheme in MD simulations [1]. However, the PGS

algorithm is an iterative algorithm and may suffer from errors. To discern whether these errors affect the results, we check that constraints specified by the rigid-body model are held in the newly generated conformations.

Distances between bound atoms and bond angles are used to evaluate stability of the algorithm. Since the model defines these as static, the maintenance of bond lengths and angles will imply stability of the algorithm.

The number of iterations of the iterative PGS algorithm can be varied between simulations. Continuing the example of 1EX6, with 20 iterations of the PGS algorithm per time-step, bond lengths experience modifications that generate errors over 0.05 Å, with some errors becoming larger than 0.1 Å. However, increasing the number of iterations to 200 completely eliminates errors over 0.05 Å. After the short energy minimization procedure bond lengths can be assumed to be within acceptable ranges.

While an analysis of the backbone bond lengths and angles is included within the G-factor score (Section 4.2.3), a closer examination of these properties is done in the current section. The WHATIF program [72] calculates the root mean square (RMS) Z-scores and standard deviations for both bond lengths and bond angles in the ensembles. Z-scores that indicate good agreement with likely values are close to 1.0. As shown in Table 4.6, all ensembles contain reasonable bond angles and bond lengths. The Z-scores of the Omega angles were also calculated and are all close to 1.0. All of the backbone angles were classified as acceptable by WHATIF.

Because the constraints imposed on the model were the bond lengths and bond angles, these results imply the constraints were obeyed during the generation of alternate protein conformations. The Omega angle was constrained with a non-rotatable joint, which means that the $C_{i-1} - N_i$ inter-residue bonds were kept from rotation about the bond axis, keeping these atoms immobile in relation to one another throughout the simulation.

Table 4.6: Bond length and angle Z-scores as calculated by WHATIF.

Name	Bond length z-score	Bond angle z-score	Omega angle z-score
HEXO (2E2N)	1.382 ± 0.132	0.981 ± 0.063	1.216 ± 0.062
GLUCO (1JEJ)	1.365 ± 0.089	1.029 ± 0.071	1.250 ± 0.070
OSMO (1SW5)	1.355 ± 0.067	1.010 ± 0.065	1.492 ± 0.065
ALLO (1GUD)	1.373 ± 0.059	1.020 ± 0.070	1.385 ± 0.056
RIB (1URP)	1.339 ± 0.012	1.001 ± 0.027	1.205 ± 0.041
GUAN (1EX6)	1.338 ± 0.016	0.976 ± 0.023	1.337 ± 0.043

4.3 Concept discussion

4.3.1 Non-coarse rigid-body simulation

The method presented in Chapter 3 uses a spherical rigid-body per backbone atom during the simulations, and links these rigid-body atoms using joints. This introduces many bodies, joints, and contact points for a given motion simulation. Thus, efficient dynamics simulation techniques are essential when simulating these many-body systems. The use of Catto's (2005) linear algorithm to solve dynamics equations enable the simulation of motions of medium sized protein backbones with very reasonable time and computation expenditures. However, if a coarse-grained rigid body model was employed, faster simulations may become possible.

The reasons why excessive coarse-graining is not essential when generating structural samples from NMA guided motions in our approach are two-fold. First, the ENM used as a directional guide in our model is already coarse-grained in that it takes only alpha carbons into consideration. Abstracting the protein further may incur loss of information about the finer changes in the low-frequency motions of the backbone. Second, using only the backbone atoms in the simulation removes the need to rebuild the entire backbone from the alpha carbon locations. When an ENM is analyzed using only alpha carbons, the positions of the remainder of the atoms in the protein must be extrapolated from the alpha carbon positions. For example, Yang and Sharp (2009) apply likely angles and bond lengths found via PROCHECK between consecutive alpha carbons. They superimpose these bond locations onto the alpha carbon positions and follow this with energy minimization [75]. These types of rebuilding steps are unnecessary if all backbone atoms are explicitly involved in the generation of the backbone, as is the case in the model presented in this work.

Using SCWRL4 [40] to repack side-chains has two benefits. The obvious benefit is the decreased amount of computation required to simulate only a backbone model, though the actual time required to run a full-atom mechanics simulation using the present model is not excessive due to the linear nature of the integrator and time stepping scheme by Catto (2005) [9]. More importantly, the removal of side-chains for the simulation is, conceptually, an essential part of generating backbone ensembles. We use NMA to gather information about the low-frequency motions of the protein, which are long-term motions and take place over large time scales. Side-chain motions are high-frequency motions; that is, the motions of the side-chains are much faster than backbone motions. Low and high-frequency motions of protein structures occur at different time scales in nature and efficiently simulating one type of motion (low-frequency motion) may

not allow for an easy way to efficiently simulate the other (high-frequency motion) concurrently. Thus, we limit ourselves to low-frequency motions which are important in the function of proteins [11, 26]. Tests of all-atom models with simulated motion initiated by forces constructed from the normal modes were undertaken, but due to the presence of side-chains, very limited mobility was present in the backbone. This crowding of atoms in an all-atom model makes motions in a long time-scale difficult to simulate efficiently.

4.3.2 Normal modes as motion guidelines and rigidity considerations

Previous use of NMA methods to guide structural sampling involved the translation of atoms along the normal modes [75]. Our method considers what effect the connectivity of the entire protein backbone has on the low-frequency motions described by NMA. By taking the backbone connectivity into account, along with the H-bonds and disulfide bonds, we recognize that low-frequency motions of the atoms may not continue in the original normal mode directions.

The theory of ENM approximates the protein as a network of bodies connected by springs with an equal spring potential [44]. The normal modes are thus the initial direction of harmonic motion of the atoms used as interacting sites. However, once these atoms move along the normal modes the original network may become obsolete due to the cut-off threshold and the assumptions of harmonic motion. This is why NMA methods are used on structures considered to be within an energetic minimum and why large conformational deviation from this structure is avoided in the original NMA model [44].

Because we do not mix the normal modes, and because we only limit our exploration of the conformational space to the directions indicated by the first three non-trivial normal modes, we lack a thorough sampling based on other normal modes. While an obvious limitation, the effectiveness of using such a small amount of information as directional guidance supports the previously experienced usability of the ENM to describe low-frequency protein motions [29, 44]. The ENM used in our work is a single-potential ENM, and

In our simulation, the motion of the backbone is initiated by the normal modes, but constrained by backbone connectivity. Backbone connectivity, together with the energy considerations of interatomic forces, results in motion constraints. These constrained motions are modeled after real protein structures due to the natural rigidity that this

connectivity provides. This joint-based approach to applying rigidity constraints is a conceptual mirror of the covalent bonds and H-bonds present in real protein structures.

Rigidity analysis has been used in protein motion considerations before and has played and important role in generating ensembles and paths between conformations [34, 66]. While an ENM alone has some inherent rigidity considerations, the connectivity in an ENM is based on a distance cut-off due to the connectivity of the springs that the ENM simulates. To properly understand the motion a protein experiences due to harmonics, we must take care to incorporate bond-connectivity. This is accomplished in the current model by constraining ENM guided movement with the rigidity of both covalent and hydrogen bonds in the protein backbone. Rigidity theory algorithms have been used with NMA methods before, but the actual sampling involved coarse graining and was essentially based on adding spring potentials based on the rigid or non-rigid distinction to portions of the ENM [2]. Rigidity considerations must be employed when studying protein motion due to the effect bond connectivity has on protein structure. Our method approximates the physical reality of this connectivity by explicitly defining all the bonding constraints in a protein backbone model. Because of this connectivity, the conformers generated by our rigid-body, joint-constrained backbone are of good stereo-chemical and energetic quality.

Chapter 5

Conclusions

We have presented an application of a constraint based dynamics technique and the elastic network model to explore conformational changes in protein structures. The constraint based method provides quick integration due to the linear algorithm used for time-stepping of rigid-bodies. Using the rigid-body approach for atom modeling facilitates the abstraction and simplification of electrochemical inter-atomic interactions through rigid-body collisions. These approximations make backbone structure computations tractable for larger proteins. The ability to inherently constrain certain aspects of the simulation as part of the model provides a useful option to reduce unlikely conformations in efficient ensemble generation. Rigidity considerations are explicitly defined due to the joint constraints and their computational tractability. The use of a rigid-body model also allows for application of directional information, such as the information generated by an ENM, as is done in this work.

The method was tested with a few unbound protein structures: D-Allose binding protein, Osmo-protection protein, D-Ribose binding protein, DNA Beta-Glucosyltransferase, Hexokinase, and Guanylate kinase. We show that the method is able to generate conformations structurally closer to a biological ligand-bound target conformation than the unbound conformation in most of these cases. The structures in these generated ensembles are shown to achieve low potential energies with only few clean-up steps.

While the use of basic ENM derived directions does not generate conformers structurally closer to the holo structures than those generated by previous methods [54], we are able to generate conformers with a better dihedral angle distribution than previous ensemble generation techniques that consider rigidity [74]. This suggests that such an

explicit consideration of connectivity and thus of rigidity within the protein backbone can be a beneficial tool in ensemble generation techniques.

5.1 Future Work

With constraint based methods being useful for very basic descriptions of inter-atomic interactions, it will be worthwhile to explore the possibilities of increasing the biological relevance of such constraints. Joints *insilico* can be abstracted in various ways. Thus, creating joints that encompass the Lennard-Jones potential or other interactions between nearby atoms would allow for improved motion studies of protein structure, especially if the speed of using constraint algorithms like the linear time stepping method presented in Chapter 3 is not hindered by such additions. Molecular dynamics simulations with an iterative integrator and constraints which describe atomic interactions could be useful in understanding protein motions. Applying fluidity to the connectivity modeled by the rigid-body backbone, such as enabling the breaking and forming of H-bonds during the simulation, can open up a larger conformational space when sampling conformers.

Exploring how directional force application of normal modes can affect protein structures further may allow for new possibilities within NMA research. NMA is not the only way to describe expected protein motions, and the use of other methods such as principal component analysis would be beneficial to improving the quantity of structures generated. As there was no mixing of normal modes in the analysis of the protein structures in this work, the generated ensembles may be missing important conformations as well. Combining normal modes for use as directional guides may improve results of this study.

While we only explored conformational relations between protein structures using structural overlap calculations, motion planning and path discovery methods could be applied to the ensembles generated by our rigid-body backbone model. This may shed new light on the interaction of normal modes between protein structures and important functional behavior of protein.

Appendix A

Rotational Inertia

The concept of rotational inertia of a rigid body i with mass m is necessary in the mathematics of rigid-body dynamics. The inertial tensor matrix I is used in the calculation of the rotational momentum $L_i(t)$ at time t,

$$L_i(t) = I\omega(t)$$

with I being a 3×3 inertial tensor matrix and $\omega(t)$ the rotational velocity at time t.

The inertial tensor matrix is a descriptor of the distribution of mass within the body around the center of mass of the body. Since this does not normally change, an inertial matrix is usually computed for each body at the beginning of a simulation and stored throughout the simulation to ease the computational requirements. I is defined by

$$I = \sum_{\forall i} \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

where

$$
\begin{aligned}
I_{xx} &= \sum_{\forall i} m_i(r_{iy}^{'2} + r_{iz}^{'2}) \\
I_{yy} &= \sum_{\forall i} m_i(r_{ix}^{'2} + r_{iz}^{'2}) \\
I_{zz} &= \sum_{\forall i} m_i(r_{ix}^{'2} + r_{iy}^{'2}) \\
I_{xy} = I_{yx} &= -\sum_{\forall i} m_i(r_{ix}^{'} r_{iy}^{'}) \\
I_{xz} = I_{zx} &= -\sum_{\forall i} m_i(r_{ix}^{'} r_{iz}^{'}) \\
I_{yz} = I_{zy} &= -\sum_{\forall i} m_i(r_{iz}^{'} r_{iy}^{'})
\end{aligned}
$$

with $r_i^{'}$ being the displacement, from the origin, of a point i on the rigid body. The $r_{ix}^{'}, r_{iy}^{'}$ and $r_{iz}^{'}$ values are the respective components of the displacement vector. During implementation, the finite sums are converted to integrals over the body volume and the mass becomes a density function: summing over all points of a rigid body shape would otherwise be a complicated task. To make this calculation feasible, the body-coordinate system allows us to define a body specific inertial tensor I_b, where the origin is at the center of mass of the body. Conversion between the body inertial tensor matrix I_b and the global coordinate system inertial tensor matrix I is simple:

$$
I = R(t) I_b R(t)^T \tag{A.1}
$$

where $R(t)$ is the orientation matrix, or a matrix indicating the orientation of the body in comparison to the global coordinate system. At time t, the columns of $R(t)$ are global coordinate directions that coincide with the axes of the rigid-body. For a deeper explanation of the rotational inertia matrices, we refer the reader to Baraff (1997) [6].

Bibliography

[1] S.A. Adcock and J.A. McCammon. Molecular dynamics: survey of methods for simulating the activity of proteins. *Chemical reviews*, 106(5):1589–1615, 2006.

[2] A. Ahmed and H. Gohlke. Multiscale modeling of macromolecular conformational changes combining concepts from rigidity and elastic network theory. *PROTEINS-NEW YORK-*, 63(4):1038, 2006.

[3] N. W. Ashcroft and N. D. Mermin. *Solid State Physics* . Brooks Cole, 1976.

[4] AR Atilgan, SR Durell, RL Jernigan, MC Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophysical Journal*, 80(1):505–515, 2001.

[5] D.S. Bae. A recursive formulation for constrained mechanical system dynamics. *Dissertation Abstracts International Part B: Science and Engineering[DISS. ABST. INT. PT. B- SCI. & ENG.],*, 47(12), 1987.

[6] D. Baraff. An Introduction to Physically Based Modeling: Rigid Body Simulation I-Unconstrained Rigid Body Dynamics and IINonpenetration Constraints. *An Introduction to Physically Based Modelling, SIGGRAPH'97 Course Notes*, 1997.

[7] A. Boeing and T. Braunl. Evaluation of real-time physics simulation systems. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 281–288. ACM, 2007.

[8] A.A. Canutescu and R.L. Dunbrack Jr. Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Science*, 12(5):963–972, 2003.

[9] E. Catto. Iterative dynamics with temporal coherence. In *Game Developer Conference*, 2005.

[10] C.N. Cavasotto, J.A. Kovacs, and R.A. Abagyan. Representing receptor flexibility in ligand docking through relevant normal modes. *J. Am. Chem. Soc*, 127(26):9632–9640, 2005.

[11] K.C. Chou. Low-frequency collective motion in biomacromolecules and its biological functions. *Biophysical Chemistry*, 30(1):3–48, 1988.

[12] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, and P.A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[13] E. A. Coutsias, C. Seok, M. P. Jacobson, and K. A. Dill. A kinematic view of loop closure. *J Comput Chem*, 25:510–528, Mar 2004.

[14] I. W. Davis, W. B. Arendall, D. C. Richardson, and J. S. Richardson. The backrub motion: how protein backbone shrugs when a sidechain dances. *Structure*, 14:265–274, Feb 2006.

[15] B. L. de Groot, D. M. van Aalten, R. M. Scheek, A. Amadei, G. Vriend, and H. J. Berendsen. Prediction of protein conformational freedom from distance constraints. *Proteins*, 29:240–251, Oct 1997.

[16] S.E. Dobbins, V.I. Lesk, and M.J.E. Sternberg. Insights into protein flexibility: The relationship between normal modes and conformational change upon protein–protein docking. *Proceedings of the National Academy of Sciences*, 105(30):10390, 2008.

[17] E. Drumwright, J. Hsu, N. Koenig, and D. Shell. Extending Open Dynamics Engine for Robotics Simulation. *Simulation, Modeling, and Programming for Autonomous Robots*, pages 38–50, 2010.

[18] R.A. Engh and R. Huber. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(4):392–400, 1991.

[19] P. Ferrara, J. Apostolakis, and A. Caflisch. Computer simulations of protein folding by targeted molecular dynamics. *Proteins*, 39:252–260, May 2000.

[20] E. Fischer. Einfluss der Configuration auf die Wirkung der Enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27(3):2985–2993, 1894.

[21] S. Fischer and M. Karplus. Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom. *Chemical physics letters*, 194(3):252–261, 1992.

[22] T.R. Forester and W. Smith. Shake, rattle, and roll: efficient constraint algorithms for linked rigid bodies. *Journal of computational chemistry*, 19(1):102–111, 1998.

[23] X. Fu, J. R. Apgar, and A. E. Keating. Modeling backbone flexibility to achieve sequence diversity: the design of novel alpha-helical ligands for Bcl-xL. *J. Mol. Biol.*, 371:1099–1117, Aug 2007.

[24] I. Georgiev, D. Keedy, J. S. Richardson, D. C. Richardson, and B. R. Donald. Algorithm for backrub motions in protein design. *Bioinformatics*, 24:196–204, Jul 2008.

[25] I. Georgiev, R. H. Lilien, and B. R. Donald. The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles. *J Comput Chem*, 29:1527–1542, Jul 2008.

[26] N. Go, T. Noguti, and T. Nishikawa. Dynamics of a small globular protein in terms of low-frequency vibrational modes. *Proceedings of the National Academy of Sciences of the United States of America*, 80(12):3696, 1983.

[27] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration, volume 31 of Springer Series in Computational Mathematics, 2002.

[28] N. Haspel, M. Moll, M. Baker, W. Chiu, and L. Kavraki. Tracing conformational changes in proteins. *BMC Structural Biology*, 10(Suppl 1):S1, 2010.

[29] S. Hayward. Normal mode analysis of biological molecules. *Computational biochemistry and biophysics*, pages 153–168, 2001.

[30] J. He, Z. Zhang, Y. Shi, and H. Liu. Efficiently explore the energy landscape of proteins in molecular dynamics simulations by amplifying collective motions. *The Journal of chemical physics*, 119:4005, 2003.

[31] K. Hinsen. The Molecular Modeling Toolkit: A New Approach to Molecular Simulations. *J. Comp. Chem.*, 21:79–85, Jan 2000.

[32] http://pyode.sourceforge.net/. PyODE: Python bindings for the Open Dynamics Engine, 2010.

[33] DJ Jacobs. Generic rigidity in three-dimensional bond-bending networks. *Journal of Physics A: Mathematical and General*, 31:6653, 1998.

[34] D.J. Jacobs, AJ Rader, L.A. Kuhn, and MF Thorpe. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Bioinformatics*, 44(2):150–165, 2001.

[35] Koshland DE Jr. The active site and enzyme action. *Adv Enzymol Relat Subj Biochem.*, 22:45 – 97, 1960.

[36] T.P. Kenakin. *A pharmacology primer: theory, applications, and methods*. Academic Pr, 2006.

[37] M.K. Kim, R.L. Jernigan, and G.S. Chirikjian. Efficient generation of feasible pathways for protein conformational transitions. *Biophysical Journal*, 83(3):1620–1630, 2002.

[38] C.A.K. Koppisetty, W. Nasir, F. Strino, G.E. Rydell, G. Larson, and P.G. Nyholm. Computational studies on the interaction of ABO-active saccharides with the norovirus VA387 capsid protein can explain experimental binding data. *Journal of computer-aided molecular design*, 24(5):423–431, 2010.

[39] DE Koshland Jr. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 44(2):98, 1958.

[40] G.G. Krivov, M.V. Shapovalov, and R.L. Dunbrack Jr. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, 2009.

[41] R.A. Laskowski, M.W. MacArthur, D.S. Moss, and J.M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26(2):283–291, 1993.

[42] R.A. Laskowski, J.A.C. Rullmann, M.W. MacArthur, R. Kaptein, and J.M. Thornton. AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR. *Journal of biomolecular NMR*, 8(4):477–486, 1996.

[43] B.J. Leimkuhler and R.D. Skeel. Symplectic numerical integrators in constrained Hamiltonian systems. *Journal of Computational Physics*, 112(1):117–125, 1994.

[44] J. Ma. Usefulness and limitations of normal mode analysis in modeling dynamics of biomolecular complexes. *Structure*, 13(3):373–380, 2005.

[45] S.L. McGovern and B.K. Shoichet. Information decay in molecular docking screens against holo, apo, and modeled conformations of enzymes. *Journal of medicinal chemistry*, 46(14):2895–2907, 2003.

[46] JEJ Mills and P.M. Dean. Three-dimensional hydrogen-bond geometry and probability information from a crystal survey. *Journal of computer-aided molecular design*, 10(6):607–622, 1996.

[47] C. Nick Pace. Measuring and increasing protein stability. *Trends in Biotechnology*, 8:93 – 98, 1990.

[48] D. Perahia and L. Mouawad. Computation of low-frequency normal modes in macromolecules: improvements to the method of diagonalization in a mixed basis and application to hemoglobin. *Comput. Chem.*, 19:241–246, Sep 1995.

[49] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin. UCSF Chimera-a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[50] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker. Protein structure prediction using Rosetta. *Meth. Enzymol.*, 383:66–93, 2004.

[51] M. Rueda, G. Bottegoni, and R. Abagyan. Consistent improvement of cross-docking results using binding site ensembles generated with elastic network normal modes. *Journal of chemical information and modeling*, 49(3):716–725, 2009.

[52] J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen. Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics*, 23(3):327–341, 1977.

[53] M. Sadqi, S. Casares, M. A. Abril, O. Lopez-Mayorga, F. Conejero-Lara, and E. Freire. The native state conformational ensemble of the SH3 domain from alpha-spectrin. *Biochemistry*, 38:8899–8906, Jul 1999.

[54] D. Seeliger and B.L. De Groot. Conformational Transitions upon Ligand Binding: Holo-Structure Prediction from Apo Conformations. *PLoS Comput Biol*, 6(1):e1000634, 2010.

[55] D. Seeliger, J. Haas, and B.L. de Groot. Geometry-based sampling of conformational transitions in proteins. *Structure*, 15(11):1482–1492, 2007.

[56] A. Shehu, L.E. Kavraki, and C. Clementi. Multiscale characterization of protein conformational ensembles. *Proteins: Structure, Function, and Bioinformatics*, 76(4):837–851, 2009.

[57] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[58] C. A. Smith and T. Kortemme. Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. *J. Mol. Biol.*, 380:742–756, Jul 2008.

[59] R. Smith. Constraints in rigid body dynamics. *Game Programming Gems*, 4:241–251, 2004.

[60] R. Smith et al. Open dynamics engine, 2005.

[61] G. Song and R.L. Jernigan. An enhanced elastic network model to represent the motions of domain-swapped proteins. *Proteins: Structure, Function, and Bioinformatics*, 63(1):197–209, 2006.

[62] M. Sprik and G. Ciccotti. Free energy from constrained molecular dynamics. *The Journal of chemical physics*, 109:7737, 1998.

[63] J.N. Stember and W. Wriggers. Bend-twist-stretch model for coarse elastic network simulation of biomolecular motion. *The Journal of chemical physics*, 131:074112, 2009.

[64] A. G. Street and S. L. Mayo. Computational protein design. *Structure*, 7:R105–109, May 1999.

[65] H. Taketomi, Y. Ueda, and N. Gō. Studies on protein folding, unfolding and fluctuations by computer simulation. *International journal of peptide and protein research*, 7(6):445–459, 1975.

[66] S. Thomas, X. Tang, L. Tapia, and N. M. Amato. Simulating protein motions with rigidity analysis. *J. Comput. Biol.*, 14:839–855, 2007.

[67] MF Thorpe. Comment on elastic network models and proteins. *Physical Biology*, 4:60, 2007.

[68] M. M. Tirion. Large Amplitude Elastic Motions in Proteins from a Single-Parameter, Atomic Analysis. *Phys. Rev. Lett.*, 77:1905–1908, Aug 1996.

[69] C. J. Tsai, B. Ma, Y. Y. Sham, S. Kumar, and R. Nussinov. Structured disorder and conformational selection. *Proteins*, 44:418–427, Sep 2001.

[70] N. Vaidehi and T. Kenakin. The role of conformational ensembles of seven transmembrane receptors in functional selectivity. *Curr Opin Pharmacol*, 10:775–781, Dec 2010.

[71] M.J. Vainio and M.S. Johnson. Generating conformer ensembles using a multiobjective genetic algorithm. *Journal of chemical information and modeling*, 47(6):2462–2474, 2007.

[72] G. Vriend. What if: a molecular modeling and drug design program. *Journal of Molecular Graphics*, 8(1):52–56, 1990.

[73] G.L. Warren, C.W. Andrews, A.M. Capelli, B. Clarke, J. LaLonde, M.H. Lambert, M. Lindvall, N. Nevins, S.F. Semus, S. Senger, et al. A critical assessment of docking programs and scoring functions. *Journal of medicinal chemistry*, 49(20):5912–5931, 2006.

[74] S. Wells, S. Menor, B. Hespenheide, and MF Thorpe. Constrained geometric simulation of diffusive motion in proteins. *Physical Biology*, 2:S127, 2005.

[75] Q. Yang and K. A. Sharp. Building alternate protein structures using the elastic network model. *Proteins*, 74:682–700, Feb 2009.

[76] Y. Zhang. Progress and challenges in protein structure prediction. *Current opinion in structural biology*, 18(3):342–348, 2008.

[77] W. Zheng, B.R. Brooks, and G. Hummer. Protein conformational transitions explored by mixed elastic network models. *Proteins: Structure, Function, and Bioinformatics*, 69(1):43–57, 2007.