The Pennsylvania State University

The Graduate School

Department of Materials Science and Engineering

IMMUNOGLOBULIN G: SOLUTION DYNAMICS, CARBOHYDRATE STRUCTURE, AND SELF-ASSOCIATION FROM ATOMISTIC AND COARSE-GRAINED SIMULATIONS

A Thesis in

Materials Science and Engineering

by

Michael E. Fortunato

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2015

The thesis of Michael E. Fortunato was reviewed and approved* by the following:

Coray M. Colina
Associate Professor of Materials Science and Engineering
Thesis Adviser

Ralph Colby
Associate Professor of Materials Science and Engineering

Scott Showalter
Associate Professor of Chemistry

Suzanne Mohney
Professor of Materials Science and Engineering and
Electrical Engineering
Chair, Intercollege Graduate Degree Program in
Materials Science and Engineering

*Signatures on file in the Graduate School.

Abstract

Immunoglobulin molecules are extremely effective at providing protection from foreign molecules or viruses; however, in certain cases the naturally occurring immune system cannot provide adequate protection. Monoclonal antibodies are designed to fill these gaps by engineering antigen binding regions capable of targeting and eliminating dangerous molecules. The monoclonal antibodies can function when isolated and able to travel through the blood stream. However, immunoglobulin efficiency decreases upon self-association for two reasons. First, the immune system may recognize the associated molecules and eliminate them depending on the size of the aggregates. Second, if the molecules associate in such a way that the antigen binding regions are no longer accessible they will be unable to function.

This thesis discusses molecular simulation techniques that can be used to study the structural changes that occur due to intrinsic molecular flexibility in immunoglobulin molecules as well as the structure of small aggregates that form through self-association. One solvated immunoglobulin molecule was studied by explicitly representing every atom however coarse-graining techniques were required in order to study multiple molecules in the same system.

The role of terminal galactose residues in the carbohydrate attached to the F_c domain in an atomisitc model of an antibody molecule was studied in this work. Carbohydrate mobility as well as the protein–carbohydrate hydrogen bonding interactions were compared between simulations with and without terminal galactose residues. It was shown that one of the two biantennary terminal galactose residues preferentially interacts with the protein when both were present; however, when both were removed the carbohydrate structure changed such that new protein–carbohydrate interactions formed. This change in solvent accessible protein surfaces will be an important area of study to both increase effector functions and decrease self-association involving the F_c domain.

Understanding through which residues immunoglobulin molecules interact will lead to better designed monoclonal antibodies with increased aggregation resistance. To aid in this understanding, a previously developed residue level coarse-grained model was employed using the same model molecule from the previous atomistic studies. The residues most often involved in self-association of the antibody were individually mutated to alanine and the resulting association frequencies were compared to the "native" immunoglobulin results to determine the residues most important in self-association. Because of the asymmetry in the three-dimensional structure of the model immunoglobulin molecule used in this work, differences in aggregation behavior between two domains with identical amino acid sequence indicated that molecular structure can play a significant role in self-association.

These simulation techniques were shown to be effective at studying molecular flexibility and protein–carbohydrate interactions with great chemical detail as well as the self-association with resolution at the residue level. The future combination of results from these types of studies will increase the understanding of the solution behavior of immunoglobulin molecules in order to develop more effective monoclonal antibodies.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would first like to acknowledge MedImmune for providing partial financial support for the research performed for this work, and Jai A. Pathak and Ralph H. Colby for establishing the collaboration between MedImmune LLC and Penn State University. I would also like to thanks Melissa Damschroder at MedImmune for her help in reading and improving the quality of this work. Computational resources for this research were provided by the Materials Simulation Center of the Materials Research Institute, the Research Computing and Cyberinfrastructure unit of Penn State Information Technology Services, and the Penn State Center for Nanoscale Science. I would also like to acknowledge Ping Lin and Maria Monica Castellanos for their assistance in homology modeling and initial structure generation for the atomistic simulations, and Ping Lin for the development of the molecular dynamics potential used in coarse-grained simulations.

Chapter 1

Introduction

1.1 Background

When the scientific fields of chemistry and biology were combined and used to study the chemistry of life, a new science was born. An immediately obvious application of knowledge in this area was health care and ways to fight disease in order to prolong life. An increased understanding of biological processes with a basis in fundamental chemical structures allowed for the development of small molecules that can manipulate the equilibria of these chemical reactions for a specific desired result. These drugs are now a many billion dollar industry in the United States alone and the focus of much research effort in both academic and industrial settings. A similar renaissance occurred when advances in computational power were applied to the study of biological systems using the understanding gained from the biochemistry field. As far back as the 1950s and 1960s computers had been used to model human speech[1] and large scale systems such as bus terminal operations[2] and traffic signals[3], however beginning in the late 1960s and early 1970s the first works of biological simulations began. These simulations modeled a significantly smaller scale, attempting to represent natural phenomena on an atomistic or molecular scale. Ambitiously, one of the first problems tackled was protein folding by Levitt and Warshel in which a simple representation of pancreatic trypsin inhibitor was subjected to successive energy minimizations and allowed to undergo thermal fluctuations.[4] Although not perfect, these simulations were a first indication that simulation without prior knowledge of the native structure could be used to predict protein folding. A few years later, Sternberg and Thornton asserted that the three dimensional structure of a

protein should be able to be determined based on the linear amino acid sequence, and discussed various methods of determining native structures using computer simulation.[5] Work at Harvard by McCammon, Gelin, and Karplus based on the initial work of Levitt and Warshel led to the first simulations of protein dynamics in which the first fundamental molecular dynamics computer code was developed.[6] This code went on to be the basis of the now well known molecular dynamics code in the CHARMM and AMBER software suites and initiated great developments in both the underlying chemical force fields and computational techniques used to study structural biology.[7]

Experimental results that define well structured proteins can be used to form the basis of a database for knowledge-based structure matching, aided by use of modern computational power. This idea was suggested by Blundell et. al. in 1987 in which the structure of a protein of interest can be predicted by searching for similarities in amino acid sequence within a large database of already known structures.[8] This procedure showed promise 27 years ago and the number and accessibility of known protein amino acid sequences and structures has been and is still growing rapidly. Computational powers today would allow for not only similarities between primary through quaternary structures to be matched but also protein function, active sites and the organism(s) of origin to enhance the matching procedure. Although the accuracy of predicting native protein structures was reasonable with a simple model and has been increasing with more complex prediction methodologies, protein folding is still a somewhat intractable problem today for large molecules. A free energy landscape is shown in one dimension in Figure 1.1 to help demonstrate this point. The native state for a given protein could exist at either (a) or (b). Although (a) is lower in energy than (b), energy barrier(s) may limit the existence of the protein at (a) in nature. This phenomenon was suggested in 1990 by Honeycutt and Thirumalai[9] and called the metastability hypothesis, in which multiple local minima in the free energy exist with similar structural characteristics. In this case the absolute lowest energy conformation obtained from simulation may not be related to the most

relevant structure of the protein. Conversely, if there is a unique folding process through which a protein can reach a final native state (a), there may be an energy barrier that limits access to (a) from a starting point (b) using molecular simulation.

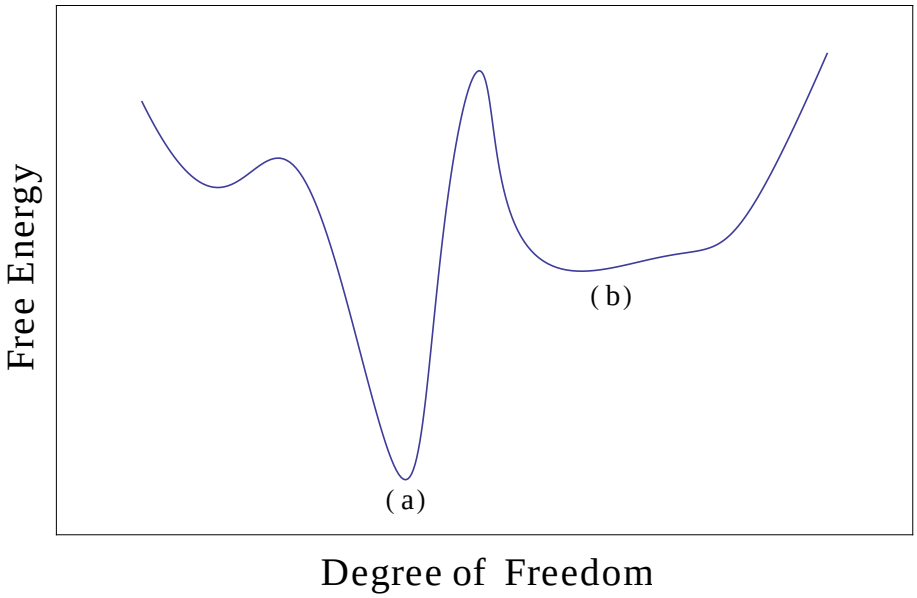

Fig. 1.1 Schematic free energy landscape showing two possible protein native states (a) and (b). Given a starting point not in close proximity to (a), molecular simulations may not be able to overcome energy barriers to result in the lowest energy conformation. The energy barrier may be overcome in nature through a unique folding process. It may also be the opposite case, where the energy barrier cannot be overcome naturally but a molecular simulation technique could be employed to arrive at (a). These two cases demonstrate the need for at least some knowledge of the native state of a protein in order to study it *in silico*.

Experimental methods such as x-ray crystallography are able to capture a snapshot of the conformations in which a molecule exists in nature and are useful starting points for simulation. In fact, because these experimental methods often only capture static representations, use of molecular dynamics simulations can offer further insight into the biophysical behavior of molecules. Although some information about dynamic behavior is available from the "blur" of electron density arising from thermal fluctuations in diffraction patterns, these results are usually obtained

at non-physiological conditions in order to aid crystallization. Furthermore, some molecules pose difficulties to crystallize for this type of characterization; however fragments of these molecules can be isolated and crystallized and then recombined *in silico* to study their full three dimensional structure. For example, Brandt et. al. used individual fragments of an immunoglobulin molecule, which will be discussed at greater length throughout this work, obtained from x-ray crystallography and reconstructed a three-dimensional structure representative of an intact antibody.[10] Insight from experimental work was used as input to generate an ensemble of structures which was then validated using a hydrodynamic model and compared to experimental results. Confirmation that the simulations were recreating the correct hydrodynamics validated that the structures resulting from simulation could in fact be found in reality but had not previously been observed.

1.2 Immunoglobulins

Immunoglobulins are a class of proteins that act as antibodies. Antibodies function in various organisms in the immune system by selectively targeting and eliminating or neutralizing foreign molecules. Antigen binding regions of the protein attach to specific regions on the foreign molecules. Other regions on the protein are recognized by various cells in the immune system that assist in elimination or destruction of antigens that may cause a risk to the organism. Immunoglobulins can be categorized into five different classes, IgA, IgD, IgE, IgG, and IgM, with varying structures and functions. IgD, IgE and IgG are found as a monomeric units representing a singular immunoglobulin molecule while IgA is found a dimer and IgM is found a pentamer of these monomeric repeat units. IgG is by far the most abundant serum immunoglobulin and the subject of most research regarding immunoglobulins.

Four protein chains in an immunoglobulin molecule fold into three globular domains that form a "Y" shape. Two arms of the Y-shape are referred to as F_{ab} domains, named as such

because these two domains contain the antigen binding regions. The third domain is called the F_c domain, named as such because it was the first domain to be found crystallizable when isolated from the rest of the protein.[11] All three domains consist of four subdomains which contain a familiar recurring immunoglobulin fold structure. This characteristic immunoglobulin fold is shown in Figure 1.2. This visualization is made using the Visual Molecular Dynamics software program[12] (VMD) specifically from the CH2 subdomain from chain 1 using the atomic coordinates from X-ray crystallography experiments reported in the 1HZH entry in the Protein Data Bank. (See later for more information about IgG crystal structures) The secondary structure appearing in each immunoglobulin fold is primarily made of antiparallel β-sheets. Two β-sheets, one containing three β-strands (blue in Figure 1.2) and one containing four β-strands (red in Figure 1.2), lie on top of each other with a disulfide bond holding them in place (yellow in Figure 1.2).[13] The loops that fold between β-strands form the complementarity determining region (CDR) in the F_{ab} domains that are responsible for binding to antigens. A total of twelve disulfide bonds (one in each of four subdomains in three domains) help maintain the structure of subdomains and, along with non-bonded interactions, consequently maintain stable domain structure. In IgG1, two disulfide bonds connect the two heavy chains in the hinge region and one disulfide bond connects CH1 and CL subdomains in each F_{ab} domain which help maintain structure and govern flexibility. The hinge region disulfide bonding differs among IgG classes.

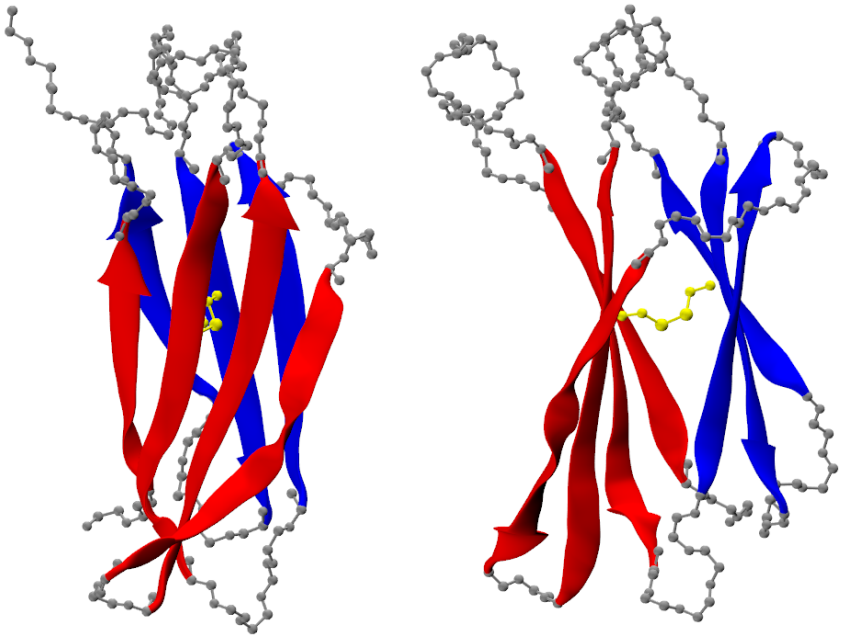

Fig. 1.2 Characteristic immunoglobulin fold in IgG subdomains visualized using VMD from the atomic coordinates captured in X-ray crystallography experiments and reported in the 1HZH entry in the Protein Data Bank. Specifically, the CH2 subdomain in 1HZH is shown here. Two β-sheets (red and blue) connected by a disulfide bond (yellow) are responsible for the stability of IgG subdomains.

The structure of a whole immunoglobulin molecule is shown in Figure 1.3. This representation visualizes the atomic coordinates using VMD captured from X-ray crystallography experiments and reported in the 1HZH entry in the Protein Data Bank. (See later for more information about IgG crystal structures) Each of the two longest of the four chains in IgG, often referred to as "heavy" chains, begin in one domain and end in a second domain, while making a flexible hinge region that connects domains. The two heavy chains begin in separate domains and meet in the third common domain to form the finally folded F_c domain. The other end of the heavy chain folds with one of two shorter, or "light", chains to form the quaternary structure of the F_{ab} domains. Heavy chains fold into four distinct subdomains, of which three are referred to as "constant" (CH1, CH2, CH3), and one is referred to as "variable" (VH). The three constant domains are largely

conserved within an immunoglobulin class (i.e. - CHX subdomains across all IgG antibodies show significant similarity in linear amino acid sequence), while VH shows more variability.[13] Sections of the VH subdomain reside in the antigen binding region and the differences in sequence within an immunoglobulin class contributes to the ability for different immunoglobulin molecules within the same class to bind to many different antigens. The light chains contain one constant subdomain, CL, and one variable subdomain, VL. As with the constant heavy chain subdomains, the CL subdomains remain reasonably conserved within an immunoglobulin class, while the VL subdomains, along with VH subdomains, determine to which antigens the immunoglobulin can bind.

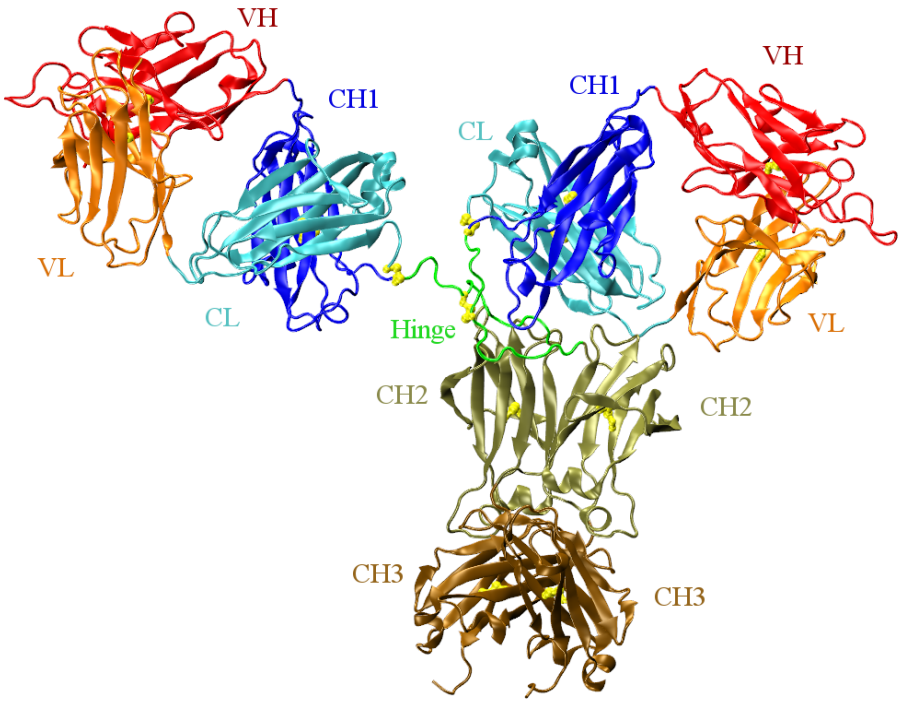

Fig. 1.3 Structure of IgG visualized using VMD from the atomic coordinates captured in X-ray crystallography experiments and reported in the 1HZH entry in the Protein Data Bank. Subdomains with "H" (but not Hinge) form from the folding of IgG heavy chains. The N-terminus begins in the VH subdomain and the chain travels through CH1, the hinge, CH2, and CH3 subdomains. Subdomains with "L" form from the folding of IgG light chains. The N-terminus begins in VL and ends in CL.

Specifically in the constant subdomains, heavy chains within each of the five classes of immunoglobulins have been observed to be very similar in amino acid sequence. They have been named α, γ, δ, ϵ, and μ for heavy chains in IgA, IgG, IgD, IgE, and IgM respectively.[13] Light chains in IgG, having been the most studied immunoglobulin, have been grouped into two different types, κ and λ, based on the CL subdomain amino acid sequence. Amino acid sequence in variable subdomains, however, are different among IgG molecules and contribute to the ability for IgG to bind to a variety of antigens.

Based on the prevalence of IgG1 in humans, it has been selected as the target for engineering monoclonal antibodies for use in supplementing the naturally occurring immune system. For many diseases, antigen binding sites can be engineered to specifically target sites on viruses or cancer cells to provide protection against antigens a human cannot normally defend. For example, engineering antibody molecules with γ type heavy chains and either κ or λ type light chains can be presumed to behave similarly to IgG in the body. By changing the amino acid sequence in VH and VL subdomains such that the newly engineered antigen binding regions specifically target harmful foreign molecules, deficiencies in the naturally occurring immune system can be remedied. The self-association of these molecules in solution limits how well they can perform their desired task of targeting antigens. Therefore, the causes of self-association must be understood in order to increase the efficiency of these monoclonal antibodies.

1.3 IgG Crystal Structures

The structure and dynamics of IgG molecules are of great interest to study in order to understand how the molecules behave in solution. Immunoglobulins, however, are too large to be studied using experimental techniques such as solution NMR, but characterization techniques such as X-ray crystallography are capable of capturing snapshots of some conformations. Only a few of

these snapshots have been captured using X-ray crystallography and released publicly[14;15;16] and they show significantly different conformations available to IgG in solution.

The 1HZH IgG crystal structure[16] was obtained from antibody b12 at a resolution of 2.7 Å. This IgG1 antibody was shown to recognize a specific site on the human immunodeficiency virus (HIV) believed to limit the virus' ability to enter new cells. While the amino acid sequences of each heavy and light chain were identical to the other, the three-dimensional structure of the conformation captured from X-ray crystallography was very asymmetrical. The F_c domain was located in close proximity to what was designated as the $F_{ab}2$ domain, while the $F_{ab}1$ domain was more isolated from the other two domains. The presumed flexibility in the hinge region manifested itself as a difference in dihedral torsional angles between the two chains. There were two possible hinge inter-chain disulfide bonds, however only one was found to be connected. The authors explained that it may fluctuate between connected and disconnected states naturally or could have been disconnected due to radiation during experiment, but indicated that only one of the two disulfide bonds needed to be connected for normal biological function. Carbohydrates were identified in the F_c domain and characterized as G1F and G2 glycans. Because antibody b12 is believed to have potential for the neutralization of HIV this crystal structure is often used as a model antibody for use in molecular simulations.

The remaining two crystal structures available in the Protein Data Bank are 1IGY and 1IGT. The 1IGY crystal structure[15] was obtained at 3.2 Å from a murine IgG1 molecule. The monoclonal antibody was designed to target the drug phenobarbital. The angle between the main axis of the F_c domain and the common plane of the F_{ab} domains was much more acute than in 1HZH and was nearly 90 degrees. The 1IGT crystal structure[14] was obtained at 2.8 Å from an IgG2 monoclonal antibody designed to target canine lymphoma tumors. The angle between main F_{ab} axes is nearly 180 degrees, while the F_c domain was found to be near perpendicular to their common axis as was

found in the 1IGY crystal structure. These two crystal structures, while informative with respect to the ensemble of conformations available to IgG molecules in solution, represent alternative but not ideal structures for modeling IgG using molecular simulations.

1.4 Thesis Overview

Building on the background information provided above, solution dynamics and small-aggregate structure of IgG molecules were studied using molecular simulation techniques. These techniques are described briefly in the following sections of Chapter 1. Chapter 2 describes all-atom molecular dynamic simulations of a single IgG molecule and highlights specific changes in protein-carbohydrate interactions in the F_c domain depending on the presence of a terminal galactose residue in the glycan. All-atom computer simulations of one solvated IgG molecule push the limit of current computational power, however are capable of describing solution dynamics unable to be captured by experimental methods. In order to study interactions between multiple IgG molecules using computational methods, a coarse-grained model is employed to study the structure of small aggregates and is described in Chapter 3. This coarse-grained model is used to highlight the residues that were shown to be important in the formation of these small aggregates. Chapter 4 discusses conclusions drawn from both works and suggests possible future work to further the understanding of IgG solution structure and dynamics to improve efficiency of therapeutic monoclonal antibodies.

1.5 Molecular Dynamics Simulations

Molecular dynamics (MD) simulations can provide access to detailed information at length and time scales that are either impossible or difficult to obtain experimentally. Classical MD simulations, derived from Newton's laws of motion, can be used to sample an ensemble of configurations

and reveal the dynamics of equilibrium structures. The applicability of simulation ranges from *ab initio* systems in which the electronic state of matter can be studied, to atomistic systems with every atom explicitly represented, to coarse-grained systems where atoms or molecules are grouped together on to continuum systems. Given this large range of length scales available to probe, there is balance between the accessibility of chemical detail and time scale based on current computational power. In order to study long time scales, one must sacrifice chemical detail; conversely, to study small, detailed systems, one is limited to short time scales. This work will discuss simulations of atomistic and coarse-grained systems containing immunoglobulin molecules. These techniques are described in the following two subsections.

1.5.1 Atomistic Molecular Dynamics

As mentioned earlier, classical MD simulations follow the laws of classical mechanics. As such, each atom in a system can be defined by a position in space and a mass. Given the initial configuration defined by a set of coordinates and a force field that describes atomic interactions, Newton's laws of motion can be numerically integrated to predict the movements of each atom through time. The selection of the force field parameters that govern interactions and consequently atomic motions is not trivial; however, much work has been done since the first MD simulations and various "well developed" force fields exist today. These force fields contain parameters for use in potential energy functions that describe pairwise interactions between different atom types. Atom types can be relatively general, such as sp3 hybridized carbon atoms or very specific, such as aromatic sp2 hybridized carbon atoms between 2 nitrogen atoms in a 5-membered ring (this occurs specifically in the histidine amino acid residue).

The set of atom types and their corresponding parameters along with the functional form of the equation that represents pairwise potential energies define the various force fields. The

atomistic simulations described in this work used the 99SB AMBER force field, which was developed specifically for simulations containing biologically relevant matter. The functional form used for pairwise interactions in the AMBER force fields is

$$V(r) = \sum_{bonds} k_b(b - b_0)^2 \quad + \sum_{angles} K_\theta(\theta - \theta_0)^2 \quad + \sum_{dihedrals} (V_n/2)(1 + cos[n\phi - \delta]) + \sum_{nonbij} (A_{ij}/r_{ij}^{12}) - (B_{ij}/r_{ij}^{6}) + (q_i q_j/r_{ij}), \quad (1.1)$$

where the potential energy, V, is dependent only on the separation distance, r, between particles i and j. Bonded parameters are required for bond force constant, k_b, equilibrium bond length, b_0, angle bending force constant, K_θ, equilibrium angle, θ_0, torsion rotation force constant, V_n, torsion symmetry, n, and equilibrium dihedral angle, δ. Nonbonded parameters are required for Lennard-Jones constants, A_{ij} and B_{ij}, and partial charges, q_i and q_j, for particles i and j. Given an instantaneous configuration, the forces acting on all atoms (and corresponding accelerations) can be obtained through the derivative of the potential energy function. Numerical integration allows prediction of atomic positions after a finite time step. Different algorithms exist to perform this propagation of atomic positions through time in which the atomic positions, velocities and accelerations are approximated by Taylor series expansion. The velocity-verlet algorithm implemented within the AMBER software package was used in the simulations described in this work.

The optimization procedures used to obtain parameters for the different atom types were performed in efforts to reproduce experimentally observed properties of macromolecules, specifically proteins. Beginning with parameters derived from *ab initio* studies, refinement of parameters in order to better replicate known structures provides a set of always improving force field parameters. Since the atomistic work presented herein was performed, a newer version of the AMBER all-atom force field, 12SB, has been released. Efforts were placed on better reproducing experimentally

observed secondary structures. As will be described in Chapter 2, the immunoglobulin domains and their secondary structures remained stable on the time-scale studied using the 99SB force field. An important note made by the forefathers of biologically oriented MD simulation is that these parameters provided in simulation software packages are approximate and can be adapted for different systems in order to better study properties of interest.[17] In simulation work where protein folding and unfolding is of great interest, the newer 12SB force field could be more applicable.

1.5.2 Coarse-Grained Molecular Dynamics

With modern computer performance, relatively large systems (∼1 million atoms) can be simulated with atomistic details for reasonable time scales (∼hundreds of nanoseconds). However, there are many systems of interest to study that are beyond these boundaries. At this scale, the number of calculations per time step with all atoms represented explicitly, even when limited to pairwise interactions, are too high to perform simulations within a reasonable time frame. In order to limit the number of these calculations, coarse-graining techniques can be employed in which the degrees of freedom in a system is decreased which in turn decreases the calculations required per time step. This is usually accomplished by grouping multiple atoms together and treating them as a single point in space or a coarse-grained (CG) bead. The motions of individual atoms within a CG bead can no longer be studied, however if these motions can be considered insignificant in order to study a property of interest, this sacrifice can prove to be beneficial. These CG beads interact through potentials that are mathematically defined in order to maintain known, important system properties such as structural motifs, etc. These potentials cannot be developed in the same way as atomistic potentials, however may still contain similarities with regards to bond, angle, and dihedral energetic terms. Successful coarse-grained systems can yield insights into systems larger

than those accessible to atomistic study and can also provide a valuable stepping stone from taking data from atomistic systems for continuum studies.

Any number of atoms can be coarse-grained into one bead. This choice depends on the system and properties of interest, and could range from grouping hydrogen atoms with heavy atoms (united-atom models) to representing molecules with thousands of amino acid residues with just a handful of CG beads. One common method to reduce the number of degrees of freedom in protein simulations is by using an Elastic Network Model (ENM), which has shown to accurately reproduce thermal fluctuations around an equilibrium structure.[18;19] Using this technique, coarse-grained beads within a cutoff distance are connected by an "elastic spring" based on separation distance. The elastic spring is mathematically represented with a harmonic energetic potential with equilibrium distance set based on an equilibrium structure and force constant adjusted to maintain protein structure. By implementing this ENM, the structure can be maintained without representing all atoms explicitly. An ENM allows for structural fluctuations and therefore allows for more efficient simulation of an equilibrium structure. An ENM can be sufficient to study longer time scales for an equilibrium structure, however to study molecular interactions an intermolecular potential is required. Different intermolecular potentials have been suggested for different degrees of coarse-graining such as the MARTINI potential[20] which uses multiple beads per protein residue. Other choices involve knowledge-based energetic potentials, as will be discussed in Chapter 3.

Whereas atomistic simulations allow specific atom-atom interactions in molecules such as hydrogen bonding or interactions between atoms with opposite partial charges, coarse-graining grants access to much longer time and length scales. Many biological phenomena occur on the microsecond to millisecond time scale which is not possible to simulate fully atomistically today. Coarse-grained simulations have been used to help solve problems on these time scales such as membrane protein assembly[21] and and RNA folding[22].

Chapter 2

Effects of Galactosylation in Immunoglobulin G from All-Atom Molecular Dynamics Simulations*

2.1 Introduction

Immunoglobulins (Ig) are a class of glycoproteins that act as antibodies in the immune system. Antibodies target and neutralize or eliminate foreign molecules to provide protection to the host organism. Of the five classes of human immunoglobulins (IgA, IgD, IgE, IgG, IgM), IgG is the most abundant and well studied. Four subclasses of IgG have been identified (IgG1–4), with IgG1 being the most prevalent serum IgG antibody.[23] Two heavy chains and two light chains fold to form three domains connected by a hinge region, which introduces molecular flexibility. Two domains contain antigen binding sites (F_{ab} domains), while the third domain contains binding sites for effector functions (F_c domain).

The F_c domain was the first domain to be isolated and crystallized by Porter in 1958.[11] Since then, the structure of many F_{ab} and F_c domains has been experimentally determined and reported in the Protein Data Bank (PDB); however, to date only three intact structures of a 150 kDa glycoprotein have been determined by X-ray crystallography. The 1IGT crystal structure[14] presents an IgG2 murine monoclonal antibody. However, as previously mentioned, we are interested in studying IgG1, because IgG1 is the most prevalent IgG subclass and of most interest for applications such as monoclonal antibodies. The 1IGY crystal structure[15] captured the structure of a murine IgG1 molecule, however the 1HZH crystal structure[16] captured the structure of a human IgG1 antibody and was therefore chosen as a model molecule to study in this work. Although these

*Reprinted with permission from Fortunato M. E., Colina C. M. *J. Phys. Chem. B* 2014, 118 (33), 9844–9851.

three structures show distinctly different IgG configurations, they by no means represent a complete sampling of all conformations available to a solvated IgG molecule. Because of the large size of the antibody, the dynamic motions of IgG in more physiologically relevant conditions cannot be studied by solution NMR. However, molecular simulations can be used to sample the ensemble of conformations available to a solvated IgG molecule by using previously determined crystal structures as starting points. The motivation of these computational studies is the need for a model IgG molecule with increased aggregation resistance to be used as monoclonal antibodies for therapeutic treatments to aid the immune system. Moreover, in recent years, much attention has been given to the role that glycosylation can play in all biomolecules and specifically IgG for the aforementioned applications driving the development of therapeutics as well as providing novel targets for drug design. These are highlighted in the recent review published in Science by Dalziel et al.[24] on glycosylation in biomolecules.

Chennamsetty et al.[25] studied an ensemble of IgG1 conformations collected from 30 ns atomistic molecular dynamic (MD) simulations, with explicit water molecules, to locate regions of the molecule prone to aggregation. Calculating the average spatial aggregation propensity (SAP) parameter for each residue allowed for the discovery of groups of residues that showed increased risk for aggregation in solution. SAP parameters were also determined from implicit water models and from the crystal structure; however, the accuracy of the results from these models decreased when compared with all atom explicit water models. These results corroborated the importance of modeling water molecules in an explicit fashion to correctly capture the behavior of IgG molecules in solution.

Also from the Trout group, Voynov et al.[26] used explicit water atomistic simulations to study protein–carbohydrate interactions in IgG1 with G0-type glycosylation. (See later for description of glycosylation types) By calculating the solvent-accessible surface area (SASA) of certain protein

residues near the carbohydrate in the C_H 2 subdomain, they were able to determine which residues were exposed with G0 glycosylation. Subsequently, by removing the carbohydrate during SASA calculations, they compared the solvent exposure of C_H2 protein residues with and without glycosylation. These results showed that certain residues initially covered by the carbohydrate could dynamically become exposed within a matter of nanoseconds. Moreover, the aromatic side chains of many of these residues were believed to be important for the protein–carbohydrate interactions and resistance to aggregation. The authors supported this by performing mutations of phenylalanine residues to nonaromatic serine residues and showed an increase in aggregation when compared with native IgG or a mutation to tryptophan, which also contains an aromatic side chain.

Recently, Wang et al.[27] performed MD simulations on a murine IgG2 molecule to study destabilization upon deglycosylation and heat stress. Their results showed that at 298 K the C_H2 subdomain was destabilized by deglycosylation evidenced by an increase in root-mean-square deviations (RMSd) of the α-carbons from the crystal structure; however, other domains were not significantly destabilized. At 400 K, destabilization was observed with deglycosylation as an increase in α-carbon RMSd throughout the whole molecule, with the hinge region and F_{ab} variable domains showing the largest destabilization in addition to the C_H2 subdomain.

Kortkhonjia et al.[28] used atomistic and explicit water MD simulations to study dynamic motions of IgG1 in solution with G2 glycosylation. By using various constraints on bond lengths and hydrogen motions, a time step of 4 fs provided increased computational efficiency and the ability to collect an ensemble of conformations representing the microsecond time scale. Supported by fluorescence anisotropy experiments, these MD simulations demonstrated the flexibility of an IgG molecule and its ability to adopt a variety of conformations.

2.1.1 Galactosylation in Immunoglobulin G

It is envisioned that antiviral therapy will have a robust glycan factor in the forthcoming years. The development of therapeutic glycoproteins has been greatly inspired by biosynthetic technologies that can produce specific glycoforms. A clear example of this approach is the development of monoclonal antibodies with engineered glycosylation, which enhanced in vivo properties, including its direct modification. Thus, it is key to understand the conformational freedom available to the various glycoforms found in IgG. Advances in this area of study, which this work aims to address, will prove to be essential in the search for therapeutic monoclonal antibodies with increased performance.

IgG contains two conserved glycosylation sites in the F_c domain. Glycosylation is heterogeneous across IgG sub classes and can also be heterogeneous within the same molecule. A biantennary carbohydrate is N-linked at a specific asparagine residue on each heavy chain in the C_H2 subdomain. As shown in Figure 1, the core, branched structure of the carbohydrate is constant, while terminal residues characterize specific glycans. These glycans are often named according to the number of biantennary terminal galactose residues (i.e., G0, G1, G2) and the presence of a core fucose residue (i.e., G0F, G1F, G2F). Additionally, a bisecting N-acetyl glucosamine residue and biantennary terminal sialic acid residues can be present. Any combination of these additional residues is possible, and the distribution of different glycans in human populations has been well studied.[29;30] These studies showed that the vast majority of IgG glycans contain a core fucose residue and one or more terminal galactose residue. Core fucose residues have been shown to be important for binding to $F_c\gamma$RIIIa, a receptor for antibody-dependent cellular cytotoxicity.[31;32] The lack of terminal galactose residues has been associated with the autoimmune disease rheumatoid arthritis,[33;34] and sialyation has been linked to anti-inflammatory responses.[35;36] While the

molecular mechanisms connecting these residues to the biological roles they play are still unclear, it is without question the glycosylation of IgG plays a large role in its biological activity.

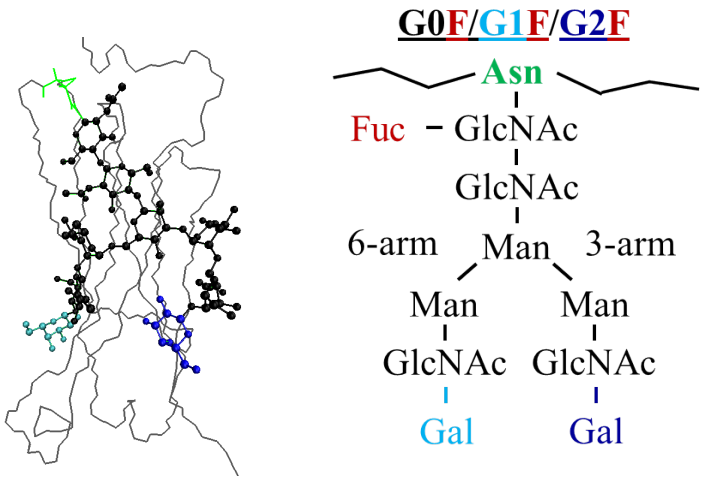

Fig. 2.1 Carbohydrate structures in the F_c C_H2 subdomain in IgG. (Left) One heavy chain Fc fragment showing the glycosylation site and carbohydrate structure. (Right) Cartoon representation of possible glycans showing the conserved, branched core structure with variable terminal residues.

In G1 glycans, the terminal galactose residue is preferentially located on the six-arm of the branched carbohydrate. (See Figure 2.1) Specific interactions of this six-arm galactose residue with the F_c domain are suggested from X-ray crystallography studies, while (if present) the three-arm galactose residue is found to be a further distance from the protein.[16;37;38] Solution NMR studies of isolated, glycosylated F_c fragments[39] have shown longer relaxation times for six-arm galactose residues in comparison to three-arm galactose residues, supporting previous findings from static X-ray crystallography snapshots. In this work, atomistic MDs simulations are used to study the structure and mobility of the carbohydrate in two different IgG1 glycoforms to understand the conformational freedom available to them. This understanding can prove to be essential in the search for therapeutic monoclonal antibodies with increased performance.

2.2 Methods

Two independent 100 ns atomistic MD simulations of a full immunoglobulin G1 (IgG1) glycoprotein were performed using the AMBER 12 software suite[40] with the AMBER ff99SB[41] and GLYCAM 06h[42] force fields. The two simulations used initial coordinates of non-hydrogen atoms obtained from X-ray diffraction experiments on IgG1 reported in the pdb entry 1HZH.[16] Thirteen missing residues in heavy chain 2 reported in the pdb entry were reconstructed using heavy chain 1 as a model. The two independent simulations differed in the glycosylation in the F_c domain. One simulation contained G1F/G2 glycosylation (IgG-G3) and the other simulation contained G0/G0 glycosylation (IgG-G0). Amber's LEaP program was used to (1) add hydrogen atoms to the heavy atoms present in the pdb entry, (2) add 22 Cl^- counterions to neutralize the charge of the protein, (3) connect disulfide bonds, (4) construct two N-linked oligosaccharides present in the F_c domain, and (5) solvate the glycoprotein with TIP3P water molecules in a truncated octahedron with a minimum distance of 10 Å between the glycoprotein and any edge of the simulation box. The initial structures contained ∼330 000 atoms (of which 20 626 atoms belonged to the protein). Periodic boundary conditions were used in all optimization and simulation procedures. Energy minimization on the initial structures was performed using the SANDER program in two steps. First, the glycoprotein atoms were restrained to their initial positions allowing the solvent and specifically the protein/water interface to relax first. Ten thousand steps of steepest descent minimization were performed, followed by 10 000 steps of conjugate gradient minimization. The restraint was then removed for another cycle of 10 000 steepest descent and 10 000 conjugate gradient minimization steps. Energy calculations for nonbonded interactions were made with a cutoff of 12 Å during minimization as well as throughout the MD simulations. The PMEMD program was used to heat the system using MD for 40 ps from 0 to 300 K in the canonical ensemble (NVT). The glycoprotein atoms were restrained during heating. The SHAKE algorithm was applied with a tolerance of

0.00001 Å to constrain bonds involving hydrogen during all MD simulations, allowing a time step of 2 fs. Particle mesh Ewald summations were used to calculate electrostatic interactions beyond the nonbonded cutoff distance in the system during all MD runs. The GPU-accelerated PMEMD program written in CUDA [43] was used for production runs. MD simulations were performed at 300 K and 1 bar in the isobaric–isothermal ensemble (NPT). System properties (energies, temperature, pressure, density) and atom coordinates were recorded every 2 ps during the simulations. Every fifth frame of the trajectory was used for analysis resulting in data points every 10 ps. Analyses of root-mean-square deviations, radii of gyration, atomic fluctuations, and hydrogen bonding were performed using CPPTRAJ in the AmberTools 12 software suite. [40]

The RMSd of backbone atoms (–C–C–N–), using the crystal structure as a reference, was mass weighted and averaged over each frame considered for analysis. To obtain domain RMSd values, only root mean square fitting and RMSd calculations for the domains of interest were averaged. The domains were defined as follows: $F_{ab}1$, residues 1–230 of heavy chain 1 and resides 1–215 of light chain 1; $F_{ab}2$, residues 1–230 of heavy chain 2 and residues 1-2-15 of light chain 2; F_c, residues 246–457 of heavy chain 1 and heavy chain 2; Hinge, residues 231-245 of heavy chain 1 and heavy chain 2.

To analyze domain orientations, we determined two vectors for each domain (see Figure 2.2) using α-carbon atom positions. The first vector defined the directional orientation of the domain (directional vector), and the second vector defined the rotational orientation of the domain (rotational vector). Each domain contained two variable subdomains of which the centers of mass were determined (points A and B). Two additional points were determined as the center of mass of the two constant subdomains (point D) and the two variable subdomains (point C). The directional vector was defined as DC, and the rotational vector was defined by the result from the cross product $DAXDB$, which represents the vector normal to the plane containing points A, B, and D. This

procedure was used to obtain two characteristic vectors for each domain. Angles of interest were determined by calculating the dot product between normalized unit vectors.

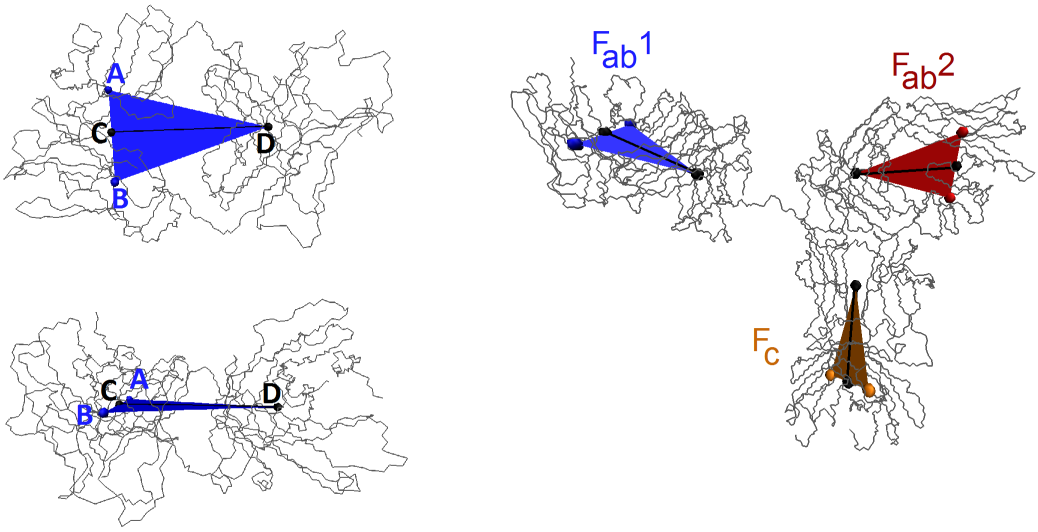

Fig. 2.2 IgG representation showing centers of mass (COM) used to determine domain orientations. A trace of the backbone is shown in gray lines. (Left) One F_{ab} domain showing the COM for the two variable subdomains (points A and B) and their COM (point C) and the COM of the two constant subdomains (point D) viewed from the top(top) and side (bottom). The plane used to represent the orientation of the domain is shown, which contains points A, B, and D. (Right) Characteristic planes shown for all three IgG domains.

Carbohydrate mobility was measured on a per residue basis by calculating the atomic fluctuations of carbon and oxygen atoms after performing a root-mean-square (RMS) fit of the F_c domain backbone coordinates. This RMS fit was preferred to fitting only the carbohydrate residues as it was of interest to study how the carbohydrate moves within the cavity between C_H2 subdomains in the F_c domain and not strictly deviations in the carbohydrate structure. Interatomic interactions were considered hydrogen bonds if the distance between the donor and acceptor heavy atoms was <3.0 Å and the angle between heavy atom–hydrogen–heavy atom was >135°. Heavy atoms

considered able to form hydrogen bonds were oxygen and nitrogen atoms. Every 10 ps, the number of hydrogen bonds between residues of interest was calculated. The fractional occupancy of hydrogen bonding was determined by breaking the trajectory into 100 1 ns blocks and calculating the percentage of frames in each block where a hydrogen bond was present.

2.3 Results and Discussion

IgG has been chosen as a model antibody to engineer monoclonal antibodies for therapeutic treatments to aid the immune system. Much effort has been put toward better understanding the structure and dynamics of the molecule in solution to decrease aggregation propensity in commercial solutions. The structure of the ∼150 kDa glycoprotein has been determined by X-ray crystallography; however, because of the large size of the antibody, the ensemble of conformations IgG can adopt under more physiologically relevant conditions cannot be currently determined by experimental techniques such as solution NMR. Thus, atomistic molecular simulations are used in this work to complement the experimental efforts, sampling the ensemble of conformations available to a solvated IgG molecule by using the experimentally determined crystal structures as starting points.

2.3.1 RMSd

RMSd calculations for individual domains and the hinge region, shown in Figure 2.3 a–d, showed that the smaller hinge region (30 total residues) displayed higher backbone deviation than the larger IgG domains (400+ residues each). Usually, larger groups of residues will have larger backbone deviations, contrary to these results. This indicated significant flexibility in the hinge region that connects relatively stable domains that allow IgG molecules to adopt a wide range of available conformations in solution.

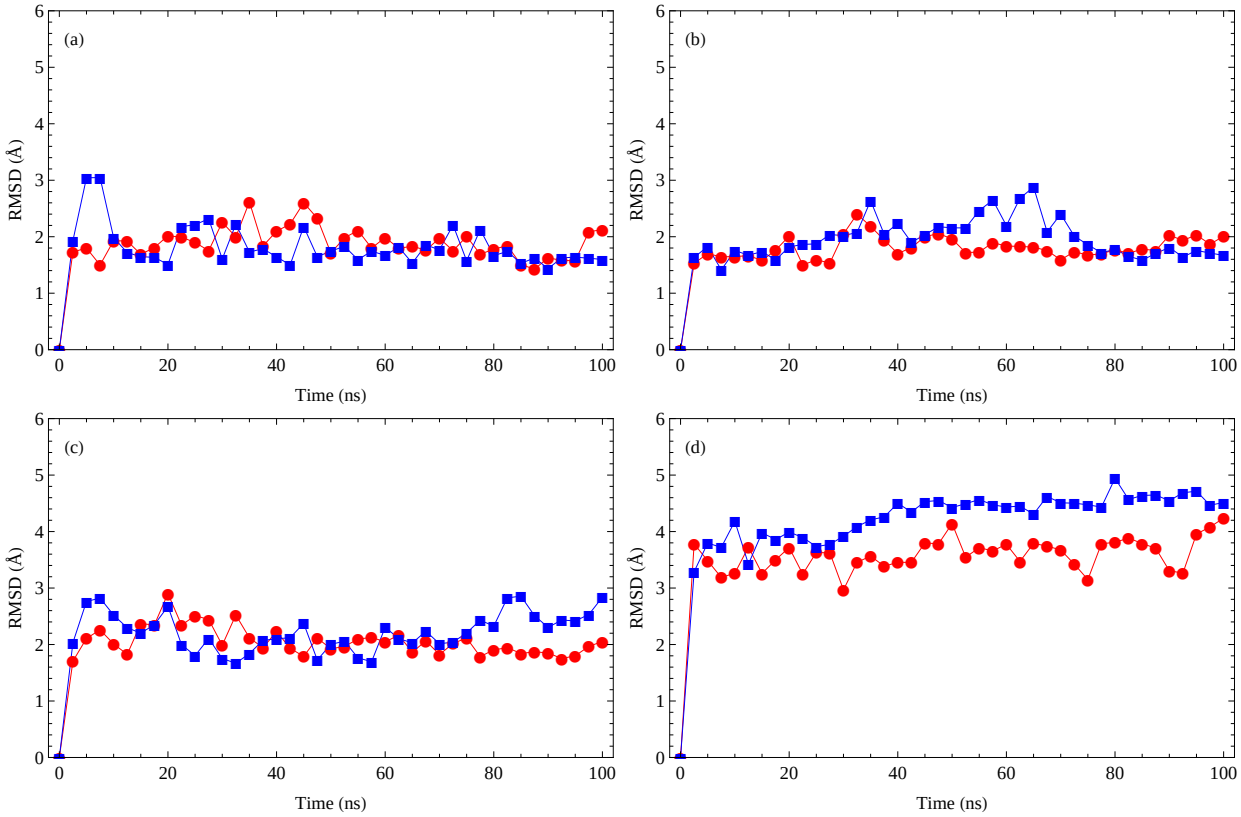

Fig. 2.3 IgG-G3 (red circles) and IgG-G0 (blue squares) domain RMSD values for backbone atoms: (a) $F_{ab}1$ domain, (b) $F_{ab}2$ domain, (c) F_c domain, and (d) hinge region.

The protein backbone RMSd values for the whole IgG molecules (IgG-G3: red circles and IgG-G0: blue squares) as a function of simulation time are shown in Figure 2.4. Fluctuations are shown in dark colored lines in the background behind data points plotted every 2.5 ns. (Fluctuations were omitted from all other plots for clarity) Large RMSd values are obtained when averaging over all backbone atoms in an IgG molecule due to segmental motions as a result of hinge flexibility and not due to domain distortions.

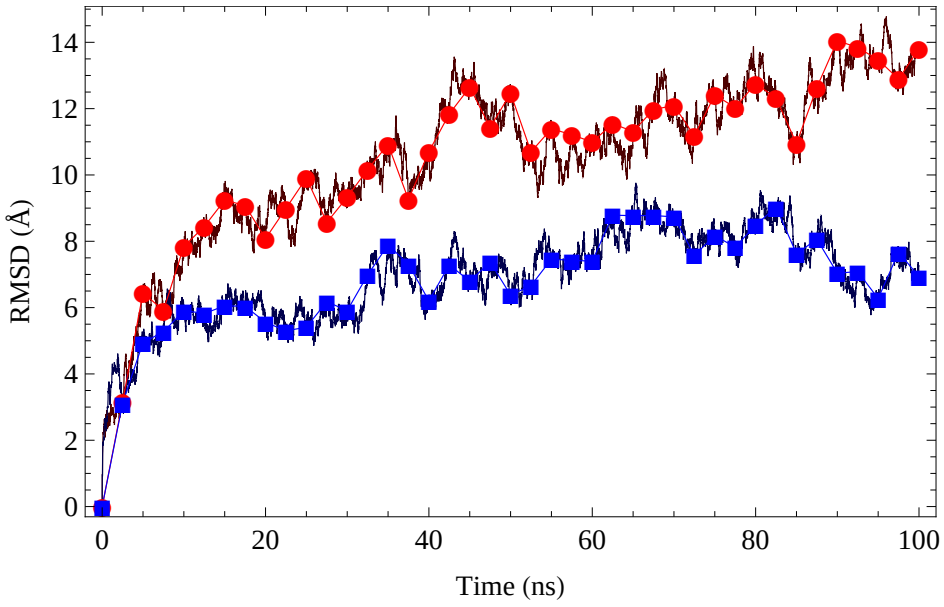

Fig. 2.4 Backbone RMSd values for IgG-G3 (red circles) and IgG-G0 (blue squares). Data points are shown every 2.5 ns, with fluctuations shown in the background.

2.3.2 Segmental Motions

The segmental motions of IgG domains contribute to the efficiency of antigen binding by allowing the domains to adopt a variety of orientations. Because of the stability observed within domains, geometrical definitions can be used to characterize these domain orientations during MD simulations. These geometrical definitions, defined in the Methods section, provided directional and rotational vectors to describe the angle between directions the domains point and the rotation between domains, respectively. In some cases, domains were free to translate or rotate freely with respect to other domains, and in other cases pairs of domains moved in tandem. The $F_{ab}1$ domain began isolated from the $F_{ab}2$ and F_c domains (see Figure 2.2) and presumably had more degrees of freedom to adopt a wider variety of conformations. This claim is supported in Figure 2.5, which shows the angle between the directional vectors of the $F_{ab}1$ and $F_{ab}2$ domains. Both IgG-G3 (left)

and IgG-G0 (right) displayed a wide range of possible angles between directional vectors of $F_{ab}1$ and $F_{ab}2$.

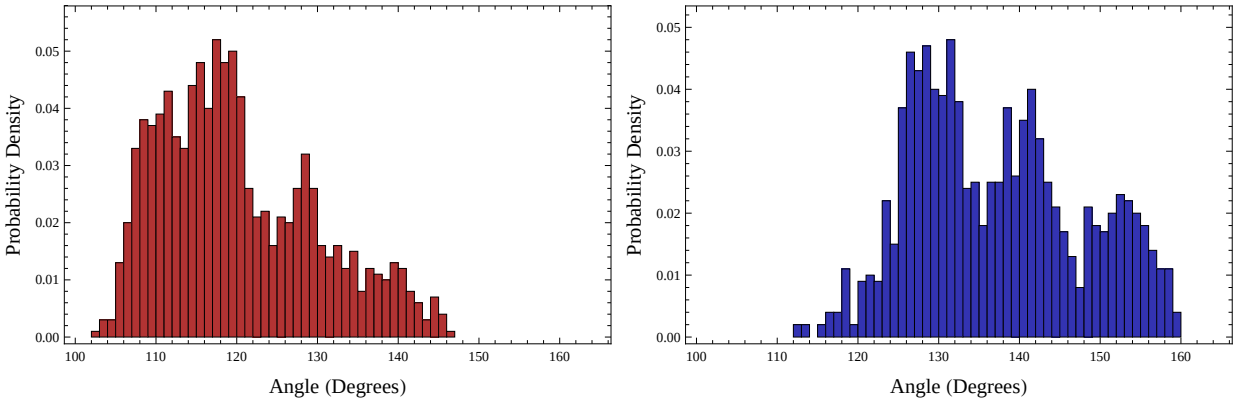

Fig. 2.5 Distributions of angles between directional vectors of $F_{ab}1$ and $F_{ab}2$ domains in the IgG-G3 (left) and IgG-G0 (right) simulations.

In contrast with the overlapping peaks seen in Figure 2.5, much sharper and more discrete distributions of favorable orientations were observed when analyzing the distributions of rotational orientations between $F_{ab}1$ and $F_{ab}2$, shown in Figure 2.6. In both simulations, the directional vectors were observed to fall roughly within the range of 105–160°without any apparent favorable directional orientations. However, three distinct distributions of favorable rotational orientations were observed at 60, 100, and 135°. Evidence of F_{ab} domains orienting at specific angles has also been seen previously by Brandt et al. in atomistic MD simulations.[10]

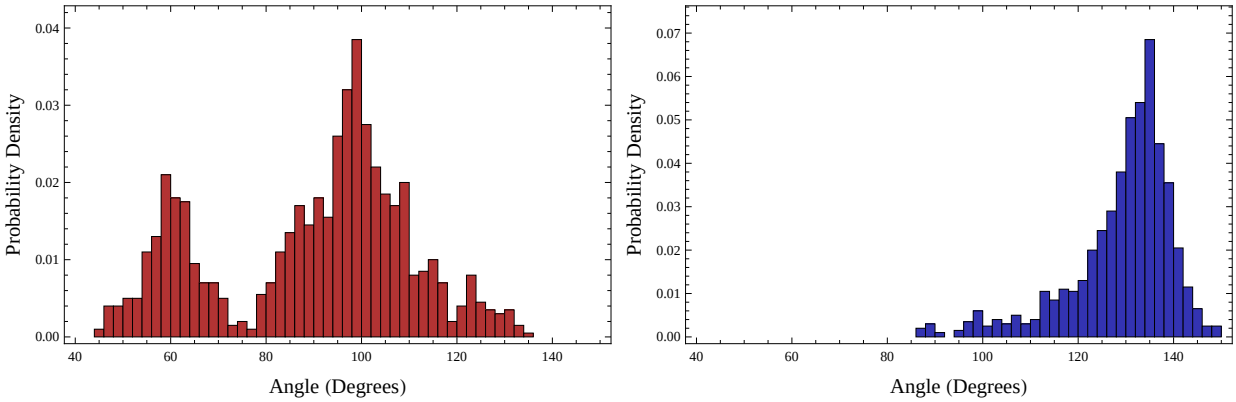

Fig. 2.6 Distributions of angles between rotational vectors of $F_{ab}1$ and $F_{ab}2$ domains in the IgG-G3 (left) and IgG-G0 (right) simulations.

2.3.3 Radius of Gyration

Along with RMSd measurements, radius of gyration (R_g) calculations provide a first indication of the structural changes occurring during the simulations and also provide an opportunity to compare with experimental results. R_g distributions of IgG-G3 (left) and IgG-G0 (right) are shown in Figure 2.7. In both simulations, the IgG molecules sample conformations that resulted in a R_g between 49 and52 Å during most of the simulation. However, the distribution of R_g observed for IgG-G3 showed three distinct peaks, while IgG-G0 showed only two peaks. The main peak in both distributions was centered at 49.5 Å and a smaller peak was centered higher between 51 and 52 Å. IgG-G3 also sampled a conformation with a smaller R_g, shown by the peak centered at 47.5 Å. This indicated that IgG-G3 was able to sample three different conformations during the 100 ns simulation while IgG-G0 sampled only two. Furthermore, in both simulations, the majority of data are found below the R_g for the initial configuration in the crystal structure of 51.8 Å, indicating that IgG adopts a more compact structure in solution than what is captured in the 1HZH crystal structure. This is likely a consequence of the environmental differences between X-ray crystallography conditions and simulations conditions.

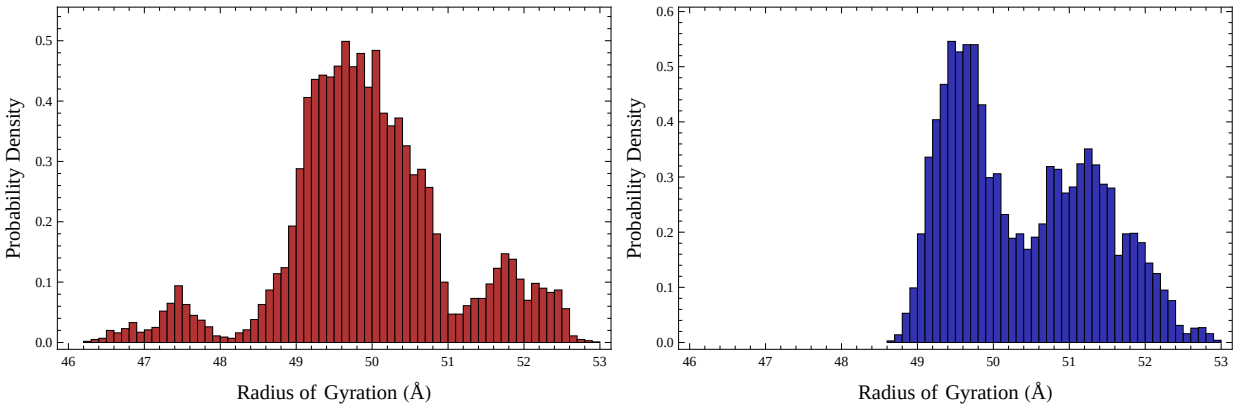

Fig. 2.7 Distribution of radii of gyration in the IgG-G3 (left) and IgG-G0 (right) simulations.

The results from the simulation here presented are within the range of results observed from X-ray and neutron scattering experiments. Monte Carlo and MD simulations in the work of Clark et al.[44] showed the R_g of IgG can range from 39 to 55 Å using free-energy constraints. Rayner et al.[45] used X-ray and neutron scattering to calculate the R_g at various concentrations and by extrapolating to an infinitely dilute solution obtained an Rg of 50.1 Å. Pilz et al.[46] also used X-ray scattering to determine the R_g of rabbit IgG to be 51.7 Å. The agreement of R_g obtained in this work with the experimental results available supports the results here presented.

2.3.4 Carbohydrate Mobility

The mobility of each carbohydrate residue (measured as described in the Methods section) is shown in Figure 2.8. Each IgG molecule contained two carbohydrates, shown on the left (carbohydrate one) and right (carbohydrate two) sides of the plot separated by a solid black line, and consisted of a core group of residues and three-arm and six-arm groups that branch from the core. The difference between IgG-G0 and IgG-G3 can easily be seen here as the absence of terminal galactose (Gal) residues in the three-arm and six-arm groups of carbohydrate one and the six-arm group of carbohydrate two as well as the absence of the fucose (Fuc) residue in carbohydrate two.

The core carbohydrate residues in both simulations showed similar behavior with reasonably constant mobility, with slightly higher overall mobility in carbohydrate two than in carbohydrate one. However, the behavior of terminal residues in the three-arm and six-arm groups showed much more interesting trends. The six-arm groups in both carbohydrates in IgG-G3 contained terminal Gal residues, which showed the lowest mobility of all residues. Furthermore, there was an apparent trend of decreasing mobility along the six-arm group from the mannose (Man) residue, which is connected to the core group, to the N-acetyl glucosamine (GlcNAc) residue, and finally to the terminal Gal residue. This indicated a specific interaction between the carbohydrate and the protein pinning the six-arm to the protein. In contrast, whether there was a terminal Gal residue in the three-arm group (carbohydrate one) or not (carbohydrate two), there was a trend of increasing mobility moving from the Man residue out toward the terminal residue, which indicated the absence of this protein–carbohydrate interaction. Similarly, in IgG-G0, where there were no terminal Gal residues, there was no trend of significantly decreasing atomic fluctuations in either the three-arm or six-arm branches. Our results agree with previous studies of IgG F_c oligosaccharides using NMR to determine terminal Gal residue mobility and specific interactions between the protein and carbohydrate.[30]

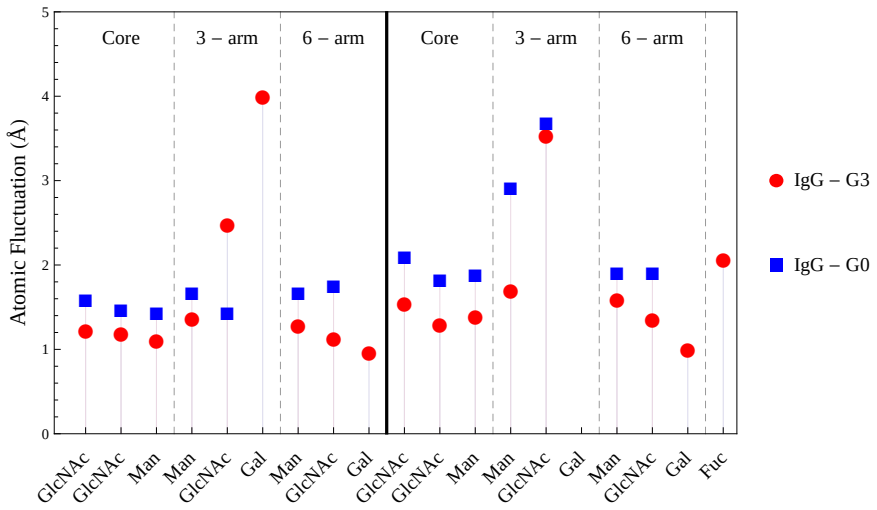

Fig. 2.8 Atomic fluctuations for carbohydrate residues. Carbohydrate one (left half) is connected to the heavy chain originating in $F_{ab}1$ and carbohydrate two (right half) is connected to the heavy chain originating in $F_{ab}2$.

2.3.5 Protein–Carbohydrate Interface

Hydrogen-bonding analyses of terminal residues in IgG-G3 and IgG-G0 were compared to study the differences in specific interactions at the protein–carbohydrate interface in the F_c domain. The fractional occupancies of hydrogen bonding between carbohydrate terminal residues and F_c domain protein residues in IgG-G3 (left) and IgG-G0 (right) are shown in Figure 2.9. Data from IgG-G3 simulations are represented in red/orange, and data from IgG-G0 simulations are represented in blue/cyan. In each plot, carbohydrate one and carbohydrate two are distinguished by dark/light colors, while galactose and GlcNAc residues are distinguished by solid/dashed lines.

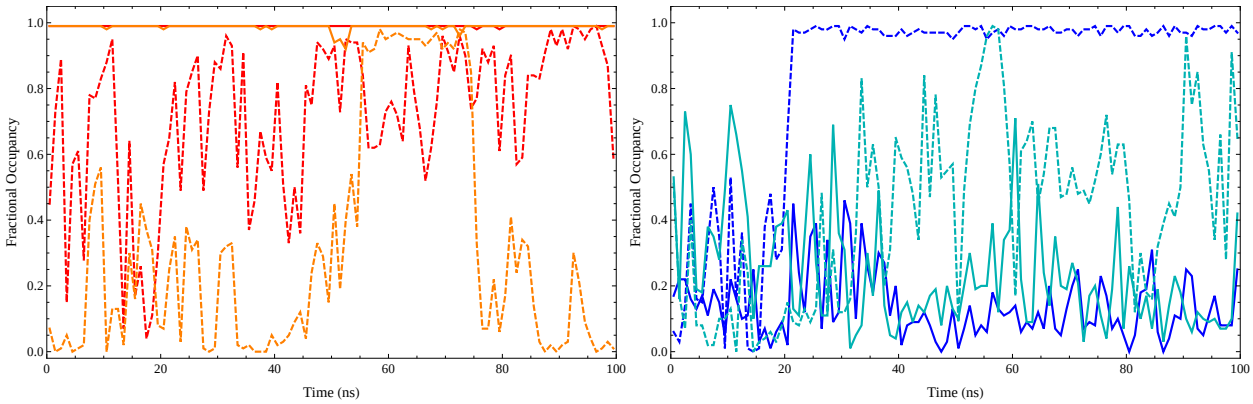

Fig. 2.9 Fractional occupancy of hydrogen bonding between terminal carbohydrate residues and the protein in IgG-G3 (left) and IgG-G0 (right) simulations. Six-arm terminal residues are represented by solid lines, and three-arm terminal residues are represented by dashed lines. Carbohydrate one is colored red (left plot) and dark blue(right plot), and carbohydrate two is colored orange(left plot) and light blue(right plot).

For both carbohydrates in IgG-G3, the six-arm terminal galactose residues (solid lines) showed nearly permanent hydrogen-bonding with a fractional occupancy >0.99 throughout the whole 100 ns trajectory. The terminal galactose residue on the three- arm of carbohydrate one (red dashed line) and terminal GlcNAc residue on the three-arm of carbohydrate two (orange dashed line) showed fewer and more dynamic interactions with the protein. The strong specific interactions of the six-arm terminal galactose residues with the protein agreed well with the low mobility determined for these residues. The three- arm terminal residues, which showed high atomic fluctuations, showed dynamic hydrogen bonding agreeing well with increased mobility.

In IgG-G0, the more permanent hydrogen-bonding interactions between the carbohydrate and the protein were observed on the three-arm branches of the carbohydrates (dashed lines). Specifically, a very strong interaction between the terminal GlcNAc residue on the three-arm of carbohydrate one and the protein was observed to form after 20 ns and remained throughout the rest of the simulation. This agreed with the slight decrease in mobility seen in the three-arm GlcNAc residue in comparison to the three-arm Man residue in carbohydrate one, although this

was not obvious from the mobility analysis alone. The very low fractional occupancy of six-arm terminal GlcNAc residues in IgG-G0 compared to the very high fractional occupancy of six-arm Gal terminal residues in IgG-G3 demonstrates the importance of terminal six-arm Gal residues to maintain specific interactions that stabilize glycan structure. If removed, as in IgG-G0, a significant structural change is observed in the glycan, leading to different specific interactions at the protein–carbohydrate interface.

2.4 Conclusions

Fully atomistic MD simulations of IgG were performed on two glycoforms (∼330 000 atoms each) for a total simulation time of 200 ns. Each independent simulation showed segmental motions of IgG domains resulting from the hinge region flexibility that allowed a variety of conformations to be adopted. Domains that interacted through specific hydrogen bonding that limited their relative motions were identified as well as other pairs of domains that were free to move but maintained specific rotational orientations relative to each other. These conformations sampled do not nearly represent the full set of configurations available to IgG in solution but do expand upon the set and knowledge available from crystallographic orientations. Carbohydrate structures between two simulations with different glycoforms were compared and found to support previous studies where terminal six-arm galactose residues are important in protein–carbohydrate interactions. Additionally, the simulations show that in the absence of terminal six-arm galactose residues, the three-arm branch of the carbohydrate can form specific hydrogen bonds at the protein–carbohydrate interface. Glycan structure has been suggested to be important for IgG effector functions upon binding to F_c receptors[47] as well as self-association through governing available protein–protein interactions.[48] Furthermore, intermolecular interactions involving the F_c domain have been shown to be important specifically in IgG self-association.[49] However, resolving glycan structure from electron

density maps from crystallographic experiments can be difficult due to their small size and relative flexibility. This work demonstrates the ability of atomistic simulations to capture glycan structural changes in the F_c domain that govern the accessibility of specific protein regions for either receptor recognition or self-association and will lead to a more thorough understanding of the role glycosylation plays in these phenomena.

Chapter 3

Coarse-Grained Simulations of Immunoglobulin G

3.1 Introduction

Due to the large size of immunoglobulin molecules, fully atomistic simulations studying intermolecular interactions on the microsecond timescale are not possible with current computing power. Studying the specific interactions responsible for the initial aggregation of IgG molecules experimentally can prove to be difficult due to problems in isolating small aggregates and the ensemble average nature of many experimental techniques. Coarse-grained models that allow access to extended length and time scales can be used to study the structure of small IgG aggregates and reveal the interactions that play a crucial role in their initial formation.

A coarse-grained model was recently developed to study electrostatic based interactions between domains of IgG.[49] Two different degrees of coarse-graining were used: a 12-bead and a 26-bead model. Both of these models greatly reduced the degrees of freedom in the system by reducing the 1000+ amino acid residues in IgG in each molecule to 12 or 26 beads. The 12-bead model placed one coarse-grained site at the center of mass of each immunoglobulin subdomain while the 26-bead model added more refinement in the CDR regions of the F_{ab} domains and the hinge region. Masses and partial charges of the coarse-grained beads were obtained from the atomistic representations of the residues that made up the coarse-grained regions. Intramolecular potential energy terms were developed from previous atomistic simulations, while Lennard-Jones and Coulombic potentials were used for intermolecular interactions. Using these coarse models 1000 antibody molecules were simulated in one system and the differences between interacting domains

of two different antibody molecules were observed. One of the two molecules showed that both F_{ab}-F_{ab} and F_{ab}-F_c interactions were highly probable, while the second molecule showed significantly more F_{ab}-F_c interactions. This work highlighted the ability of coarse-grained models to elucidate the structures formed upon initial aggregation of large immunoglobulin molecules using a very coarse CG model.

There are numerous, practically unlimited, ways to coarse-grain molecular coordinates to reduce the number of degrees of freedom. The choice for the degree of coarse-graining depends greatly on the phenomena one wishes to study. In this work the self-association of immunoglobulin molecules was studied. Although explicitly representing all atoms was out of computational reach, residue based coarse-graining allowed for the identification of specific residues that cause and/or play an important role in self-association. Additionally this allowed for the systematic mutation of these select residues to observe the self-associating behavior when this residue has been substituted. Other coarse-grained models have been developed in which finer coarse-graining is applied to proteins, such as the AMBER united-atom potential[50] which groups hydrogen atoms with heavy atoms, and the MARTINI potential[20] which uses a mapping of four heavy atoms to one coarse-grained bead and results in few, but multiple, beads per residue. These models, while more computationally expensive than a one–bead–per–residue mapping, can capture more detailed structural changes which are believed to be very important in dictating aggregation potential.[51] Joubert et al. showed that a variety of monoclonal antibodies across all IgG subclasses showed aggregation behaviors dependent on the type of stress applied.[52] This suggested that because the aggregation pathway was found to be dependent on stress and independent of IgG subclass, the destabilization and/or changes in structure (domain orientations) likely plays a more significant role than amino acid sequence. This leads to two practical techniques to study coarse-grained IgG molecules: one in which one structure (domain orientations) is simulated to study how molecules with this structure

associate and then systematically alter the domain orientations, and another method in which the molecular domains are allowed to reorient during the simulation. The first method is less computationally expensive and still allows for the study of how molecules with differently oriented domains associate and is therefore employed here.

The work presented herein uses a coarse-grained Elastic Network Model in which each amino acid residue is represented by one bead. The model was originally presented by Lin and Colina[53]. The choice of spatial position for each coarse-grained bead was chosen to be the position of each alpha carbon atom in the protein crystal structures. An Elastic Network Model was used for intramolecular bead-bead interactions in which all beads within a cutoff distance of 11.5 Å were connected with an elastic bond based on separation distance. Spring force constants of 20, 5, and 1 kcal/mol were applied to beads separated by 4, 7, and 11.5 Å respectively. A pairwise Buckingham potential was used for intermolecular bead-bead interactions. Parameters for each unique bead-bead interaction were determined on a knowledge basis using a database of 384 protein domains and a scoring function based on complexes formed between these domains. It is important to note that these parameters developed for the intermolecular potential were determined in the presence of explicit solvent, and as such the coarse-grained potentials implicitly included solvent effects. In other words, the strength of interactions between bead types was related to their probability of interacting in solution. With this taken into account, solvent molecules are not explicitly represented in the simulations presented in this work. The methodology for structure generation and simulation are presented first, followed by a description of the different systems studied in this work. The selection process for mutations and calculation procedures for analytical results are then discussed. Results from the various systems are presented before providing conclusions gathered from all simulations.

3.2 Methodology

The initial structures used to perform coarse-grained simulations of immunoglobulin G were obtained using spatial coordinates from crystal structure entries in the Protein Data Bank (see introduction chapter for in depth description of 1HZH and 1IGY crystal structures). Coarse-grained bead coordinates for each amino acid residue were defined by the atomic coordinates of the alpha carbon atoms in the crystal structures. The effective potentials developed in the Lin model described all pairwise interactions in the system and were used to determine the forces acting on all coarse-grained beads. Intramolecular potentials were modeled with using simple harmonic energy wells centered at the "equilibrium" separation distance (*i.e.*, the separation distance between beads in the starting structure). Intermolecular potentials were modeled using the Buckingham potential

$$E = Ae^{-r/\rho} - \frac{C}{r^6}, \tag{3.1}$$

where the parameters A, ρ, and C were previously determined for each residue–residue interaction using protein docking. A scoring function for residue interactions was created using a set of protein complexes. Residue sizes were determined to define the effective separation distances between each pair of residues. For those pairs of residues determined to be "attractive" based on the scoring function the potential was set to zero at $\frac{3}{4}$ of the effective separation distance, while the potential well depth was set to 1.5 times the scoring function. For those pairs of residues determined to be "repulsive" based on the scoring function the C parameter is set to zero and the potential is set to correspond to the scoring function at the effective separation distance and 100 times the scoring function at zero separation.

A structure generation program was developed to create systems containing coarse-grained molecules using Lin's model. For each residue in the PDB file provided to the program, the residue

type, and α-carbon x, y, and z coordinates were used to determine all bonds required for the Elastic Network Model. The desired number of molecules were inserted with random orientation onto a cubic lattice and simulation box dimensions were determined in order to result in a system with the desired concentration. The program then output all ENM bond coefficients, CG bead types (residue types) and positions, and pairs of bonded CG beads formatted as a data file compatible with the LAMMPS (Large-scale Atomic/Molecules Massively Parallel Simulator) software package. See the Appendix for the python script used to generate the LAMMPS input data file.

Molecular dynamics simulations were performed using LAMMPS in the canonical ensemble (NVT) at 300 K using a time-step of 50 fs. Walls reflected beads that moved outside the simulation box by placing them back inside the system proportional to the distance they traveled outside the system and inverting the velocity component in the direction it exited the system. Thermodynamic information about the system was recorded every 2.5 ps and atomic positions were recorded every 50 ps. Simulations were performed for at least 60 million time steps (3 μs). Initially, systems containing two native 1HZH IgG molecules were simulated at 10, 20, and 40 mg/ml. The concentration is set by the simulation box size, such that the lower concentration systems have more space to explore and high concentration systems feel more confinement from simulation box walls. For example, to obtain a system at 10 mg/ml, two IgG molecules were confined in a cubic simulation box with a dimension of 368 Å. Similarly, to obtain a system at 40 mg/ml, two IgG molecules were confined in a cubic simulation box with a dimension of 232 Å. Systems containing two molecules with single site mutations were then simulated, as well as larger systems containing 8 native or mutated IgG molecules. Residues for mutation were selected based on the frequency with which they were involved in associations in simulations containing native molecules and surrounding residues with nonzero solvent accessibility. Lastly double-site mutations were simulated in which the residue in the other corresponding heavy or light chain was also mutated. Table 3.1 summarizes the

systems studied using this coarse-grained model. Systems are given an identification name for future reference. The first letter(s) describe the mutations present (N-native; SM-single mutation; DM-double mutation). Residue numbers are absolute residue numbers such that they range from 1 to 1344. The last letter before punctuation describes the largest aggregate that can be studied in the system, or the system size (D-dimer; O-octamer). The integers following punctuation either describe the concentration being studied (in the case of native IgG molecules), or the mutation(s) present in the system. Columns 2-5 in Table 3.1 contain information about the system size, concentration, mutations, and reference structures used to obtain the force constants and equilibrium separations for the intramolecular Elastic Network Model.

Table 3.1 Summary of coarse-grained systems.

Sim. ID	**System Size (# IgG)**	**Conc. (mg/ml)**	**Mutations**
ND.10	2	10	None
ND.20	2	20	None
ND.40	2	40	None
SMD.487	2	20	S487A
SMD.489	2	20	F489A
SMD.508	2	20	I508A
SMD.Mut#1	2	20	Mut#1
SMD.Mut#2	2	20	Mut#2
SMD.702	2	20	L702A
NO.10	8	10	None
NO.40	8	40	None
DMO.30	8	40	S30A/S487A
DMO.Mut#1	8	40	Mut#1

Molecular associations were determined by calculating the total non-bonded interaction energy between two molecules. Two molecules with an interaction energy less than -12 kcal/mol were considered to be involved in a reversible association based on previous work with this model.[53] The frequency of these associations was monitored to determine equilibration of the systems with respect to molecular associations. Average association frequencies were obtained by performing averages over successive blocks of 1000 snapshots. Each snapshot is separated by 50 ps, and therefore

averages were taken over 50 ns of simulation time. Equilibration was asserted after a plateau in the running averages of these values was observed. The occurrences of domain-domain interactions were determined based on both molecular pairwise interactions energies and bead-bead separation distances. For each snapshot in which a molecular association was present (molecular pairwise interactions energy less than -12 kcal/mol), all pairs of CG beads separated by less than 6.5 Å in that snapshot were recorded. For each pair of domains, the total number of occurrences of CG bead interactions was used to create distributions of domain-domain associations.

3.3 Results and Discussion

3.3.1 Native IgG Dimers

Systems containing two coarse-grained native 1HZH IgG molecules were simulated at 10 (ND.10), 20 (ND.20), and 40 (ND.40) mg/ml to study the formation of native IgG dimers. The simulation time required to obtain equilibration of association frequency at these different concentrations was determined which helped to guide the selection of concentration for subsequent simulations of larger systems. Equilibration of average association frequency was a requirement in order to adequately sample the conformational space available to the small aggregates in the system. Plots of average association frequency as a function of simulation time for one independent simulation at each concentration are shown in Figures 3.1-3.3 for concentrations of 10, 20 and 40 mg/ml respectively. The number of molecular associations per 1000 snapshots are shown in blue, while the running average of these values is shown in pink. As the concentration increased, the frequency of associations and average association frequency value both increased.

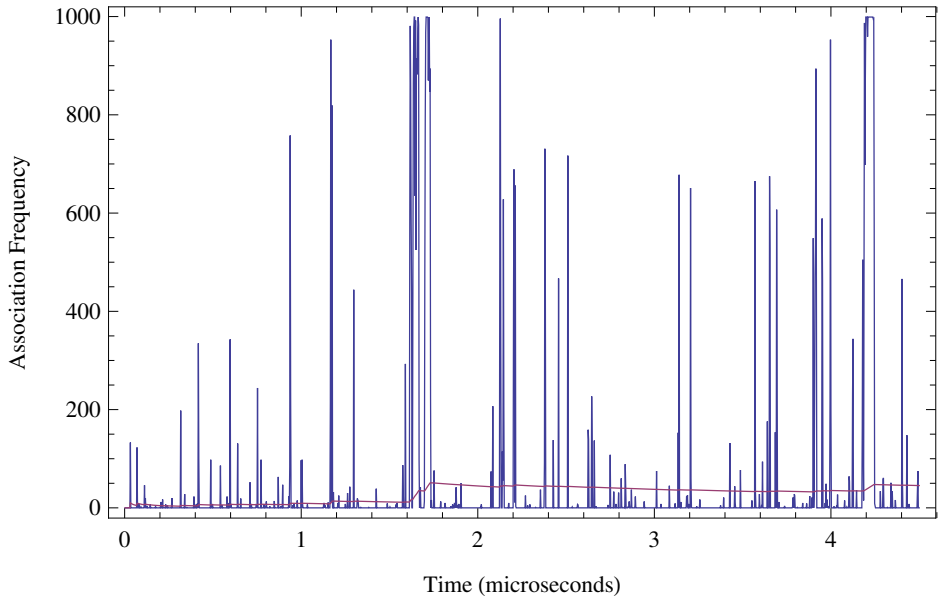

Fig. 3.1 Association frequency (blue) and running average of association frequency (pink) of IgG molecules at 10 mg/ml.

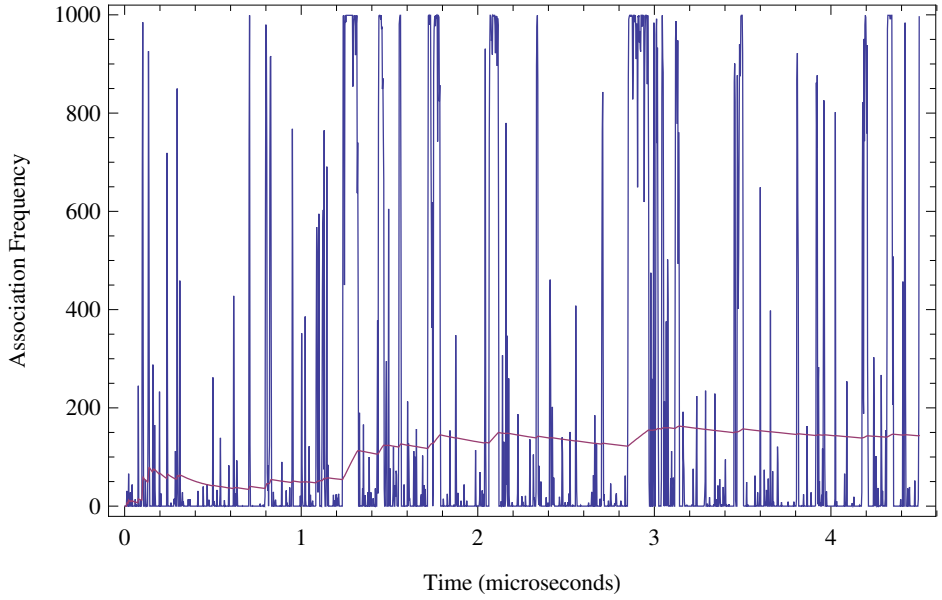

Fig. 3.2 Association frequency (blue) and running average of association frequency (pink) of IgG molecules at 20 mg/ml.

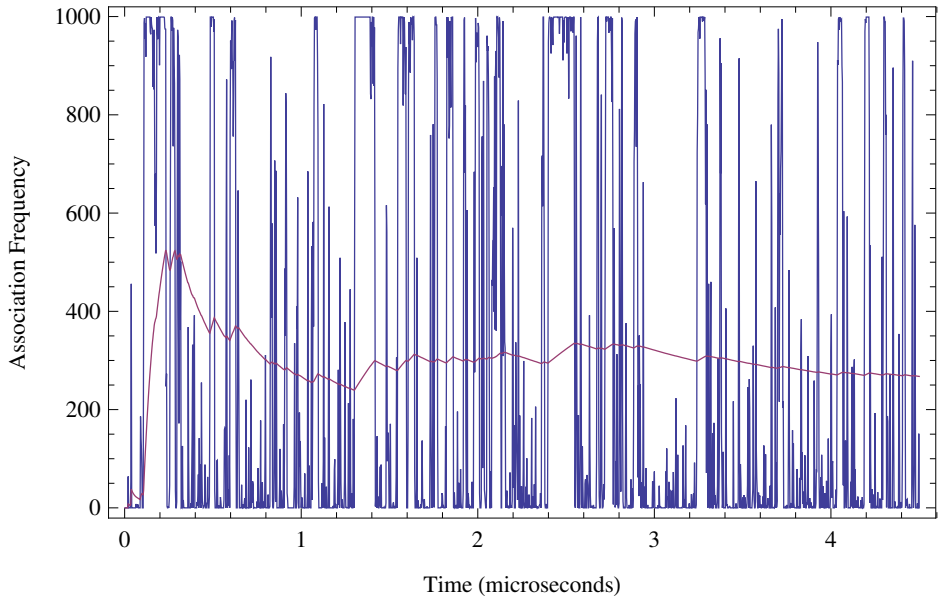

Fig. 3.3 Association frequency (blue) and running average of association frequency (pink) of IgG molecules at 40 mg/ml.

The average association frequency (pink lines from Figures 3.1-3.3) for each of three independent simulations at each concentration are compared in Figures 3.4-3.6, showing the degree of reproducibility between independent simulations. Three independent simulations of ND.20 and ND.40 systems showed sufficient reproducibility of average association frequency and evidence of equilibration after at least two microseconds. The ND.10 systems showed many fewer associations as expected at a lower concentration, and did not show adequate equilibration within four microseconds. Due to the increased simulation time required to achieve equilibration at 10 mg/ml, dimer systems at this concentration were not studied further when introducing mutations; however, this concentration was not eliminated from consideration when studying systems with more molecules. Both increasing simulation time and increasing the number of molecules in each system accomplish the same goal of increasing the total number of IgG interactions, which is the ultimate goal to ensure adequate sampling. Using the plateau value of the average association frequency from each

simulation, the averages and standard deviations of average association frequency at each concentration were obtained and plotted in Figure 3.7. The results at 10 mg/ml are included, however are represented with an open symbol to indicate equilibration was not achieved. As seen previously, the average association frequency increased as the concentration increased, however the deviation in this value also increased.

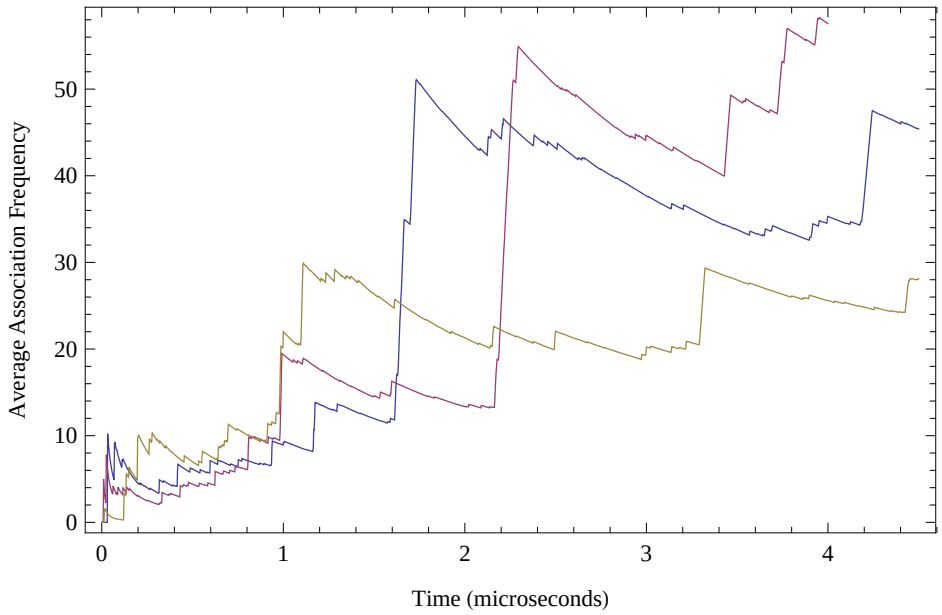

Fig. 3.4 Average association frequency of IgG molecules in dimer system ND.10 at 10 mg/ml for three independent simulations.

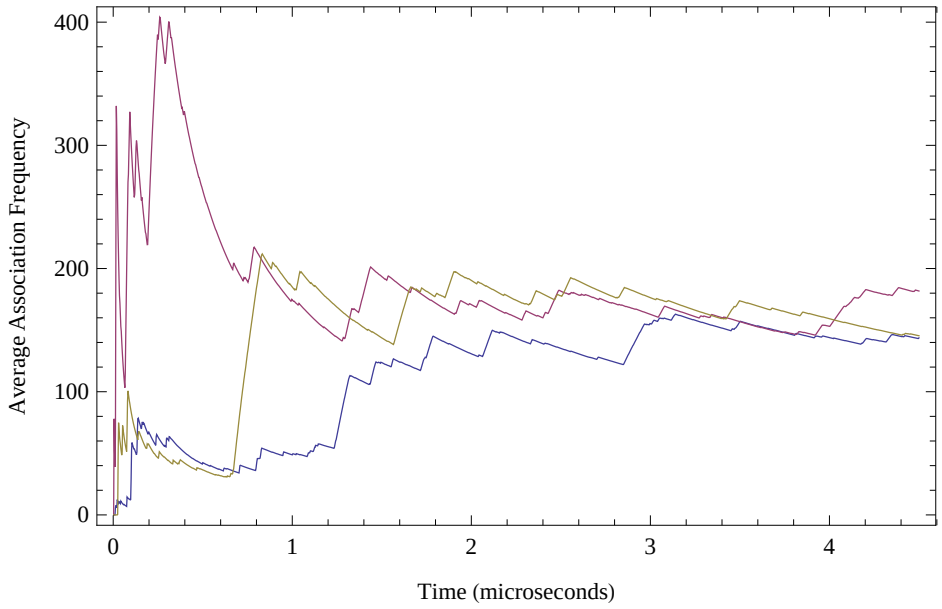

Fig. 3.5 Average association frequency of IgG molecules in dimer system ND.20 at 20 mg/ml for three independent simulations.

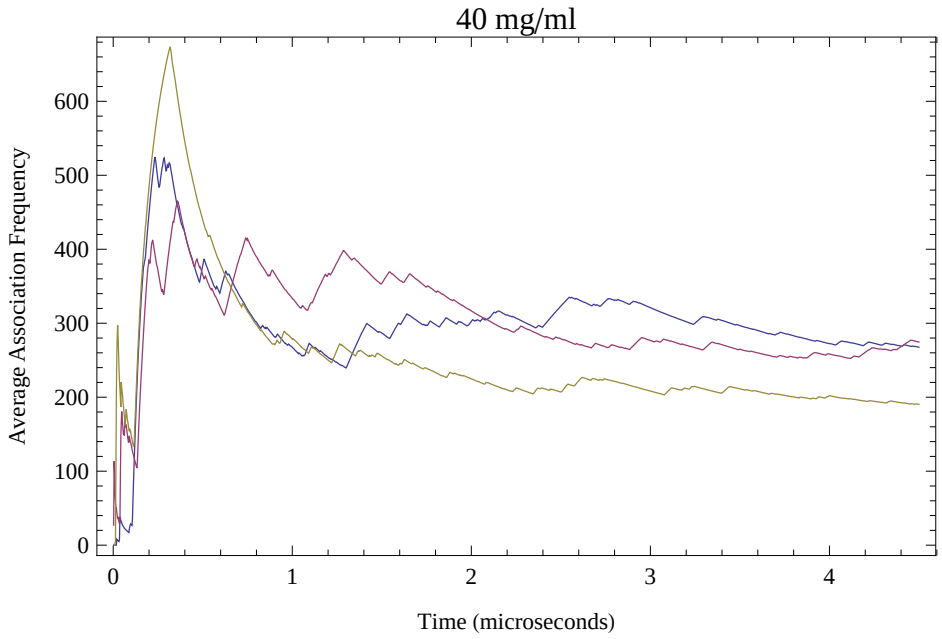

Fig. 3.6 Average association frequency of IgG molecules in dimer system ND.40 at 40 mg/ml for three independent simulations.

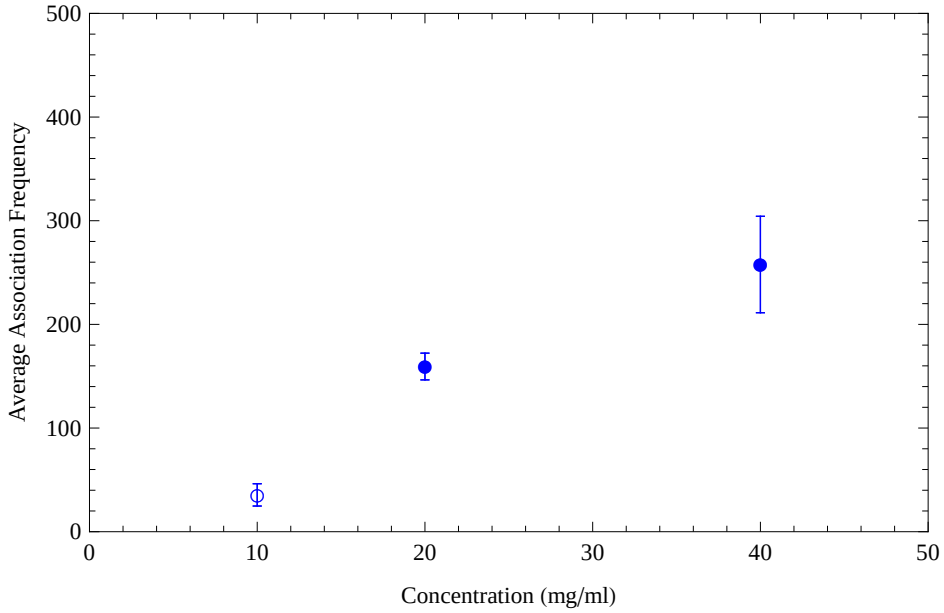

Fig. 3.7 Mean and standard deviation of average association frequency values showing degree of reproducibility over three independent simulations at each concentration. Data point at 10 mg/ml is shown with open symbol indicating equilibration was not achieved.

3.3.2 Small Native IgG Aggregates

Once the model was shown to be capable of capturing reversible molecular associations in the formation of IgG dimers, larger systems were simulated in order to study the formation of small IgG aggregates. It is important to note that almost exclusively reversible associations were observed, such that molecules can separate and re-associate. Only under these conditions can many different conformations of small aggregates can be studied. Here, small IgG aggregates are considered to be groups of three or more IgG molecules that are considered associated using the -12 kcal/mol pairwise intermolecular interaction energy threshold. Eight native 1HZH IgG molecules were placed in a simulation box such that their centers of geometry were located on a simple cubic lattice. Each molecule was randomly rotated to remove initial orientational bias and the simulations were performed following the same procedure as with previous systems. Only systems at 10 mg/ml (NO.10) and 40 mg/ml (NO.40) were simulated. It is important to note that dimer systems at

10 mg/ml which showed poor sampling of associations within a few microseconds displayed better sampling on the same time scale due to the increased number of opportunities for association (*i.e.*- increased number of molecules in the system). In this regard, simulating on an extended time-scale or increasing the number of molecules at the same concentration both result in improved sampling of molecular associations.

The number of occurrences of aggregates of different sizes on a logarithmic scale is shown in Figure 3.8 with NO.10 shown in blue and NO.40 shown in red. As expected, there were more occurrences of larger aggregates at a higher concentration indicating the probability of forming larger aggregates increased with concentration. Figure 3.8 also shows the relative probability of aggregation growth at these two concentrations. At 10 mg/ml (blue) there was a higher probability for intermediately sized aggregates (4-5 molecules) to decrease in size rather than increase in size, evidenced by the larger number of smaller sized aggregates. At 40 mg/ml intermediately sized aggregates had similar probability to increase or decrease in size. This can be considered a relative increase in the probability of aggregation growth at 40 mg/ml. This model is best suited to study the structure of small IgG aggregates as opposed to the dynamics of their formation, and as such studying systems with a higher concentration that increase the rate at which aggregates form allowed for more sampling of small IgG aggregates.

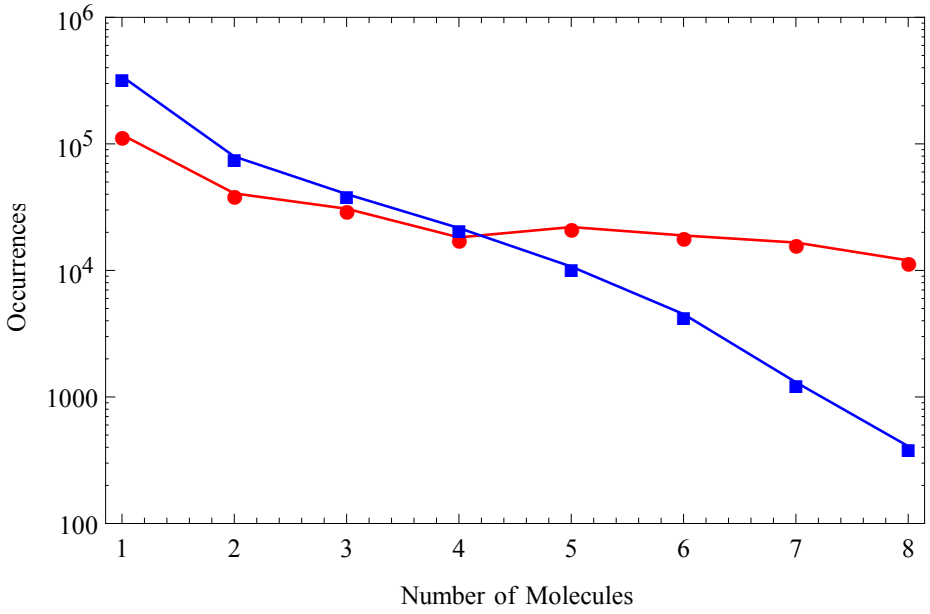

Fig. 3.8 Distribution of differently sized 1HZH IgG aggregates in a system containing eight molecules at 10 mg/ml (NO.10; blue) and 40 mg/ml (NO.40; red).

To determine which of the domains were most frequently involved in the self-association of IgG in these simulations the number of occurrences of each domain-domain interaction was determined. For each association (defined by pairwise molecular interaction energy less than -12 kcal/mol) pairs of residues that were within 6.5 Å were recorded. These residue pairs were categorized into six different domain-domain interactions: three like-domain associations (Fab1-Fab1, Fab2-Fab2, Fc-Fc) and three mixed-domain associations (Fab1-Fab2, Fab1-Fc, Fab2-Fc). Figure 3.9 shows the number of occurrences for each of these domain-domain associations. Each set of data points (connected by lines as a guide for the eye) in grey represents a different molecule in the system. Data points and error bars in red are means and standard deviations obtained from all molecules in the system. Two important observations were made from this plot. First, the mixed-domain associations occurred much more frequently than the like-domain associations. Secondly, the most probable association that formed was between the $F_{ab}2$ and F_c domains. This association

occurred with a higher probability than the association between $F_{ab}1$ and F_c even though the linear sequence of amino acids is identical in $F_{ab}1$ and $F_{ab}2$ domains in the 1HZH molecule. These two observations indicated that differences in association probability between pairs of domains originated due to differences in domain orientations. The 1HZH crystal structure is asymmetrical with the $F_{ab}2$ and F_c domains in close contact while the $F_{ab}1$ domain is more isolated. It must then be concluded that changes in IgG domain orientation could alter the association probability between domains and could play an important part in IgG self-association.

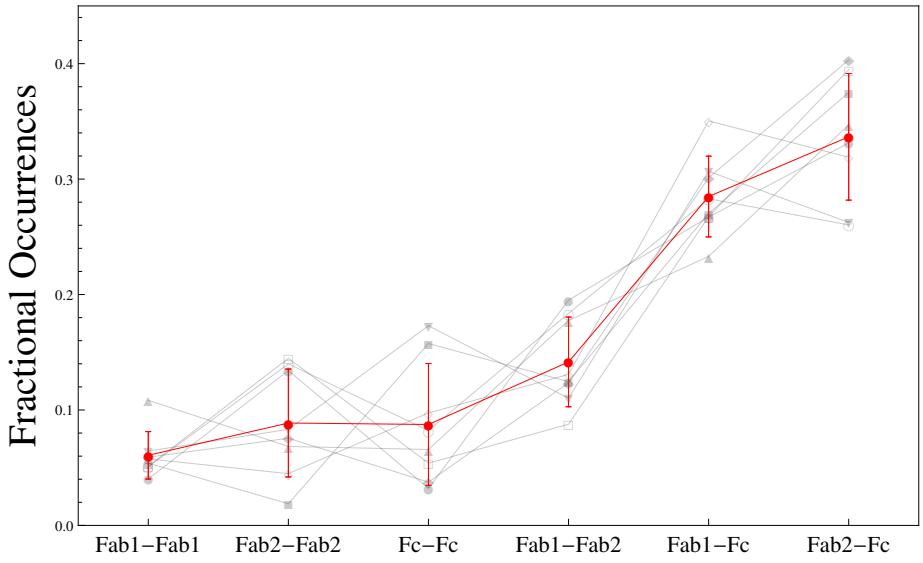

Fig. 3.9 Number of occurrences for each domain-domain associations in native 1HZH IgG. Each set of data points (connected by lines as a guide for the eye) in grey represents a different molecule in the system. Data points and error bars in red are means and standard deviations obtained from all molecules in the system.

3.3.3 Mutations and Their Effect on Domain-domain Interactions

Dimer Systems

Dimer systems at 20 mg/ml were chosen as a model system to confirm this CG model can be used to study the effect of mutation of specific residues on IgG association. It was shown above

that these systems had relatively small standard deviation in average association frequency and therefore the statistical significance of differences observed in this value upon mutation can be taken into consideration. Dimer systems (as opposed to larger systems) were simulated first to isolate important residues for IgG association. These residues/mutations could then be studied further by simulating larger, more computationally demanding systems.

Simulations with single site mutations were performed to study the influence of individual protein residues on self-association. Mutation sites were chosen in regions believed to be important for IgG self-association based on the frequency of involvement in associations of native IgG from previous simulation. These mutations were S487A, F489A, I508A, Mut#1, Mut#2, and L702A and were all located in the $F_{ab}2$ domain. Systems with two 1HZH IgG molecules with these mutations were created at 20 mg/ml and were simulated using the same procedure as the native dimer systems. Figure 3.10 shows the average association frequency values obtained from these simulations with single site mutations (SMD.487 - brown, SMD.489 - magenta, SMD.508 - red, SMD.Mut#1 - teal, SMD.Mut#2 -green, and SMD.702 - blue). For comparison, the average association frequency value of native IgG was 159 $\pm$ 13 associations per 1000 snapshots. Both Mut#1 and Mut#2 resulted in a decrease of average association frequency of statistical significance, indicating these residues may be important in IgG self-association.

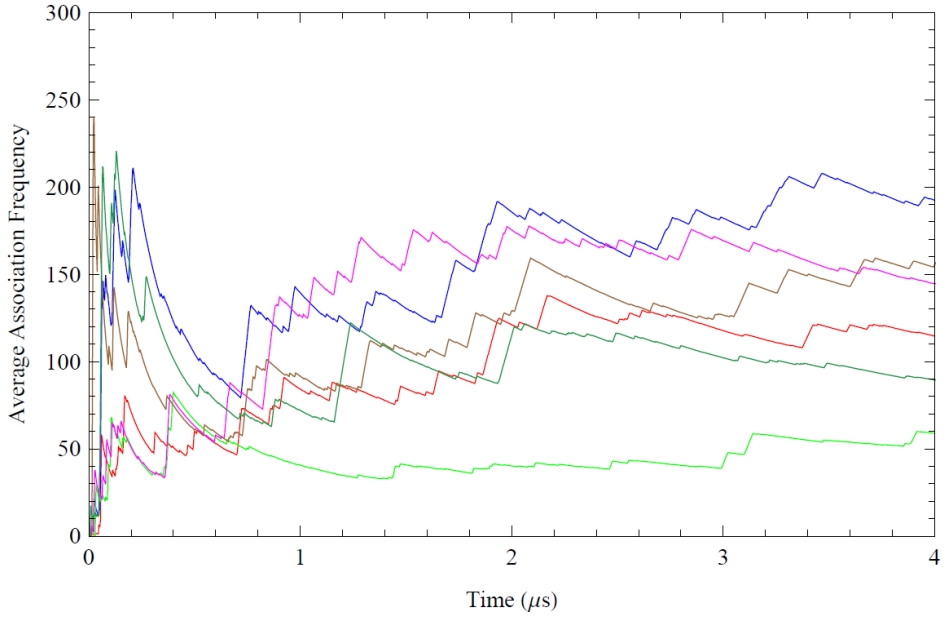

Fig. 3.10 Average association frequency values obtained from simulations with single site mutations (SMD.487 - brown, SMD.489 - magenta, SMD.508 - red, SMD.Mut#1 - teal, SMD.Mut#2 -green, and SMD.702 - blue). The average association frequency value of native IgG was 159 ± 13 associations per 1000 snapshots.

Figure 3.11 shows the number of occurrences for domain-domain interactions in three of these single site mutation dimer simulations. The S487A mutation showed native-like behavior, as expected from the similarity in average association frequency with the native IgG simulations. Surprisingly, the F489A mutation, which also showed similarity in average association frequency with the native IgG simulations, showed a significant decrease in $F_{ab}2$-F_c interaction occurrences, however showed an increase in $F_{ab}1$-F_c interactions, which resulted in an overall average association frequency similar to that of native IgG. The F489A mutation was effective at decreasing the interactions between the $F_{ab}2$ and F_c domains, as residue 489 is located in the $F_{ab}2$ domain, however an increase in $F_{ab}1$-F_c interactions compensated for this decrease. The end result was little change in overall average association frequency. The Mut#1 mutation also showed a significant decrease in $F_{ab}2$-F_c interactions, as well as decreases in $F_{ab}2$-$F_{ab}2$, $F_{ab}1$-$F_{ab}2$, and F_c-F_c interactions. In this

case, there was not a compensation for the loss of associations through increased $F_{ab}1$-F_c interactions. In other words, associations involving this residue are not only solely governed by residue type but domain orientation may also play a role in association probability.

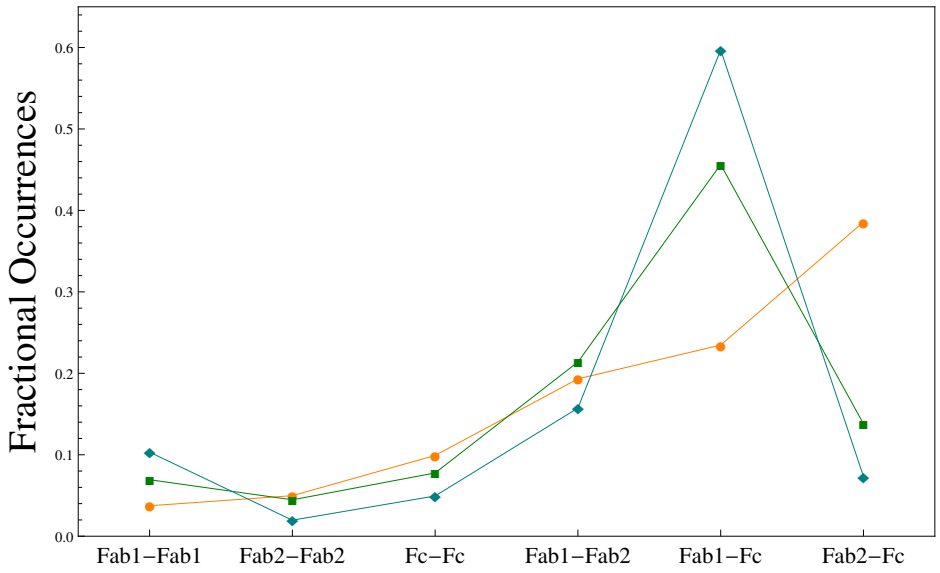

Fig. 3.11 Number of occurrences for domain-domain associations in SMD.487 (orange), SMD.489 (green), and SMD.Mut#1 (cyan).

Octamer Systems

S487A and Mut#1 mutations were further studied by simulating systems containing eight of these mutated IgG molecules at 40 mg/ml. Additionally, mutations of molecules in these systems were performed in both the $F_{ab}1$ and $F_{ab}2$ domains. These systems were refferred to as DMO.30 (S30A/S487A) and DMO.Mut#1. Please refer to Table 3.1 for the full description of these simulations. In each simulation, the number of domain-domain interactions for each molecule were averaged and the standard deviation was calculated. The results for DMO.30 are shown in Figure 3.12. Standard deviations were relatively low, indicating consistent association behavior for all molecules in the system. Compared to native IgG (see Figure 3.9), DMO.30 showed a decrease

in the probability of F_{ab}2-F_c associations, although F_{ab}1-F_c associations were still present. This again reinforces the importance of IgG structure on the self-association probability. The results for DMO.Mut#1 are shown in Figure 3.13. A low number of occurrences of most domain-domain interactions were observed in comparison to both native IgG and DMO.30; however, a very large standard deviation was obtained for F_{ab}2-F_c interactions. For 6 of the 8 molecules in the system, a decrease in F_{ab}2-F_c interactions was observed. In 4 of these 6 cases the number of interactions was below 200,000, and in the other 2 cases, the number of F_{ab}2-F_c interactions was below 400,000 (mean value from DMO.30). In the remaining 2 cases (*i.e.*-2 of the 8 molecules in the system) the number of F_{ab}2-F_c interactions was over 1,000,000 contributing to the large standard deviation. This indicated that for the majority of the molecules in the system, the DMO.Mut#1 mutation significantly decreased the association probability resulting from all of the domain-domain interactions. However, in a few cases (2 of 8 molecules) F_{ab}2-F_c interactions remained prevalent in the system. If IgG aggregation occurs in two steps, first a weak reversible association followed by formation of strong irreversible associations, it is possible that the mutations in DMO.Mut#1 decreased the probability of forming strong associations through limiting the initial formation of weak reversible associations. However, if these strong associations did form, the mutations present in DMO.Mut#1 played no role in causing molecular separation. It logically follows then that either or both residues in DMO.Mut#1 are important in forming the initial weak reversible associations however are not crucial to the formation of strong irreversible associations.

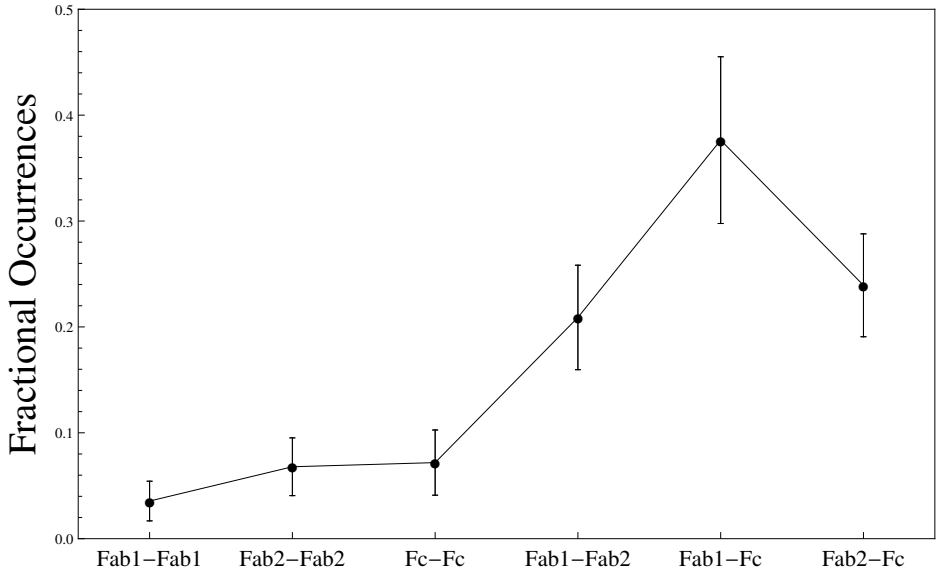

Fig. 3.12 Number of occurrences for domain-domain interactions in DMO.30. Averages and Standard deviations are calculated from all eight molecules in the system.

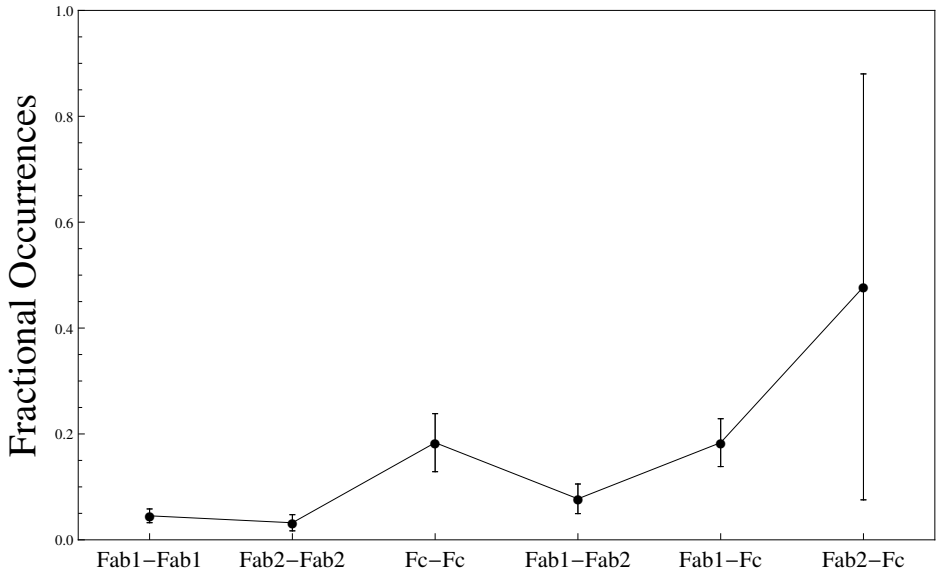

Fig. 3.13 Number of occurrences for domain-domain interactions in DMO.Mut#1. Averages and Standard deviations are calculated from all eight molecules in the system.

3.3.4 Conclusions

A one–bead–per–residue coarse-grain model was used to study the structure of small IgG aggregates. The model implicitly included solvent effects through modeling intermolecular interactions (residue–residue interactions) based on a scoring function of docking simulations from many protein complexes. Molecular structure was maintained using an ENM with distance dependent force constants. Equilibration of the association frequency during simulation was confirmed for 20 and 40 mg/ml, however a concentration of 10 mg/ml showed poor equilibration within 3 μs due to fewer molecular contacts. Simulations of IgG using the structure captured from X-ray experiments and reported in the 1HZH entry in the Protein Data Bank showed that "native" 1HZH IgG associated most frequently through F_c-F_{ab} interactions. Although the amino acid sequence is identical in both F_{ab} domains, the domain orientations were not symmetrical and this resulted in slightly higher association between $F_{ab}2$ and F_c domains, where $F_{ab}2$ is the F_{ab} domain in close contact with the F_c domain in the crystal structure. For *in silico* mutagenesis studies, residues were selected based on high frequency of involvement in associations in the native 1HZH IgG. Mutation of some of these residues showed statistically significant changes in association occurrences while mutation of other select residues caused little change in association behavior. Mut#1 in $F_{ab}2$ and the corresponding residue in $F_{ab}1$ showed promising results with regards to decreasing IgG self-association by limiting weak reversible associations, however strong associations still formed which the mutations studied herein did not eliminate. The methodology presented in this work shows the viability of studying the important residues for self-association for IgG through mutation of specific residues. Moreover, the model is general and can be applied to any protein or mixture of proteins. This provides a relatively fast way to screen through many mutations to identify problematic regions of proteins for aggregation as well as study the structure of small aggregates.

Chapter 4

Conclusions and Future Work

4.1 Atomistic Scale

The ability to explicitly represent every atom in a system allows for great chemical detail to study protein solution behavior. Immunoglobulin molecules are of great interest to study in this manner to better understand their solution behavior and increase their efficiency for use in therapeutic treatments as monoclonal antibodies. The work presented in this thesis discussed the segmental motions resulting from hinge region flexibility on the 100 ns time scale and carbohydrate mobility as a function of the presence of terminal galactose residues in the F_c carbohydrate. It is believed that both of these phenomena can play a role in IgG self-association and/or effector functions. Glycosylation has specifically received much attention recently as a "newer" method to control IgG behavior. The simulations presented here demonstrated the ability to capture changes in carbohydrate mobility and protein–carbohydrate interactions by removing one residue at the terminal ends of the biantennary carbohydrates. The results agreed well with the previously suggested mobility of agalactosylated carbohydrates. New carbohydrate structures - ones in which the three-arm carbohydrate branch interacted strongly with the protein through hydrogen bonding - were presented which can have a large impact in the design of future monoclonal antibodies, specifically regarding the choice of glycosylation, to decrease IgG self-association.

Various other interesting phenomena were observed during these simulations as well. Although IgG1 molecules normally have two disulfide bonds connecting the hinge region, the structure captured in the 1HZH crystal structure (which is IgG1) showed that only one of the two possible

disulfide bonds were connected. In fact, the second set of CYS residues were separated by more than 15 Å. The authors[16] indicated the disulfide bond could be dynamic, or could have been damaged during X-ray experiments. During the atomistic simulations presented in Chapter 2, the second, unconnected set of CYS residues were observed to approach each other and remain within a reasonable distance that could allow a dynamic disulfide bond to exist. Preliminary results are presented below followed by suggested future work based on these results.

4.1.1 Disulfide Bonds in IgG1

Disulfide bonds in the hinge region of IgG connect the two heavy chains and play a part in governing the segmental motions and intrinsic flexibility of the domains. Different IgG classes have different disulfide bond connectivity in the hinge region depending on the location of CYS residues in the amino acid sequence. In IgG1 there are two pairs of CYS residues in the hinge region that can be connected in disulfide bonds. They are separated by three residues in each heavy chain and is it generally believed that both are connected although only one is required for normal effector functionality.[16] In the 1HZH crystal structure[16], it was found that only one pair of CYS residues was connected by a disulfide bond. This conformation is shown in Figure 4.1 visualized using VMD. The direction in which the domains can be found are indicated as well for reference. It can be seen here that the sulfur atoms (yellow) in the initial configuration in the unconnected CYS residues are separated by more than 15 Å. During simulation (specifically discussed here for simulation IgG-G3 from Chapter 2, however the same phenomenon was observed in IgG-G0 simulations), these unconnected CYS residues approach to a distance likely able to accommodate a dynamic disulfide bond. The separation distance at 9 ns can be seen in Figure 4.2.

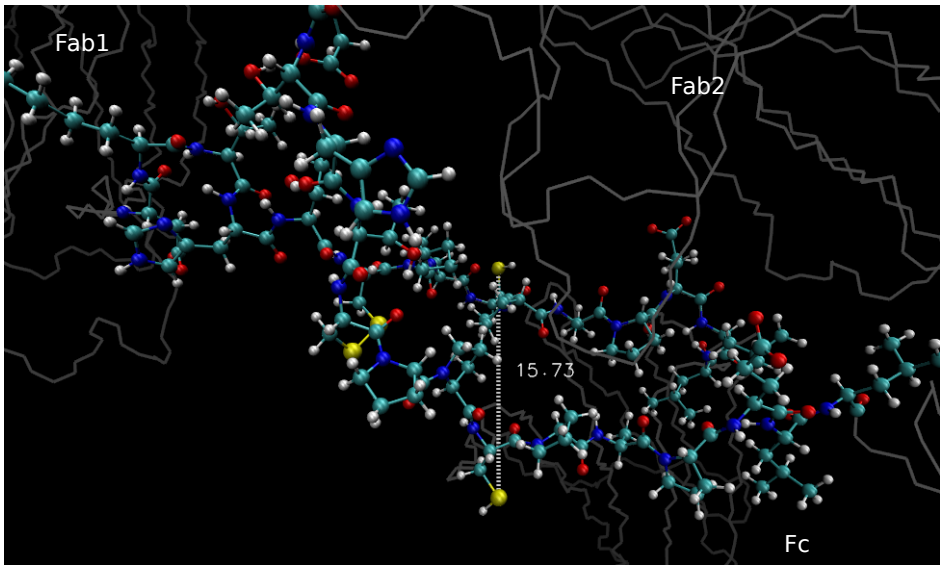

Fig. 4.1 Hinge region in 1HZH IgG at 0 ns in the IgG-G3 simulation from Chapter 2. Sulfur atoms (shown in yellow) from two pairs of CYS residues can be seen. One pair of CYS residues is connected in a disulfide bond while the other pair begin separated by >15 Å.

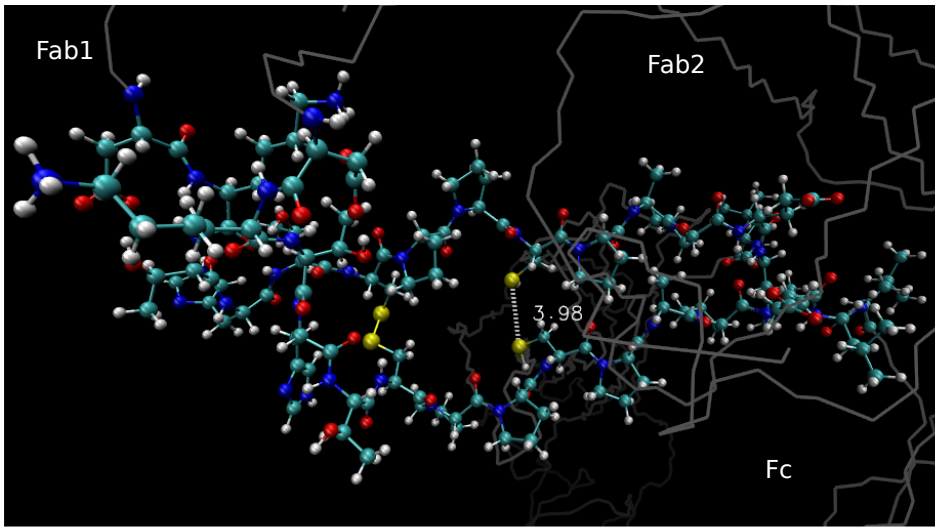

Fig. 4.2 Hinge region in 1HZH IgG at 9 ns in the IgG-G3 simulation from Chapter 2. Sulfur atoms (shown in yellow) from two pairs of CYS residues can be seen. One pair of CYS residues is connected in a disulfide bond while the other pair has now approached to a separation distance <4 Å.

A plot of separation distance between the two pairs of sulfur atoms in corresponding CYS residues in the hinge region can be seen in Figure 4.3. Here it can be seen that relatively quickly (within 10 ns) the sulfur atoms approach in close proximity, going from a separation distance >15 Å to <4 Å. The separation distance between sulfur atoms connected in the disulfide bond are also shown for reference. This separation distance will never be reached for nonbonded sulfur atoms.

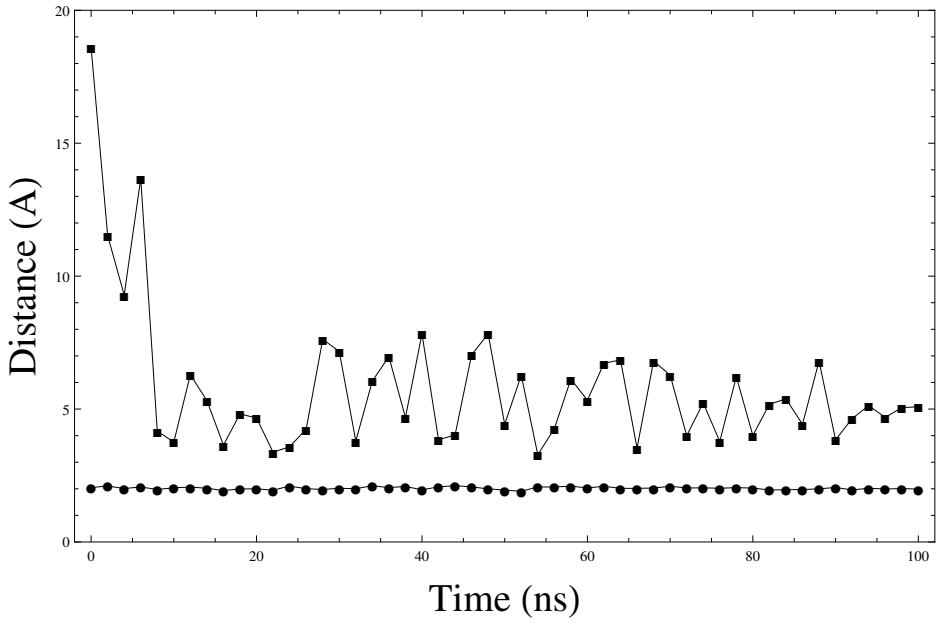

Fig. 4.3 Separation distance between sulfur atoms in two pairs of CYS residues in the hinge region of 1HZH IgG as a function of simulaion time in the IgG-G3 simulation from Chapter 2. One pair of sulfur atoms are connected in a disulfide bond and their separation distance is held constant at the equilibrium bond distance governed by the AMBER force field. The second pair of sulfur atoms begin separated by >15 Å by relatively quickly approach <4 Å and fluctuate near that distance.

4.1.2 Future Work

Future simulations studying the effects of hinge region disulfide connectivity on segmental motions could lead to insights crucial to engineer antibodies with controlled flexibility. Following the methodology presented in Chapter 2 for fully-atomistic simulations of IgG, independent simulations with different hinge region disulfide connectivity will demonstrate how disulfide bonds in the hinge

region govern flexibility. In these new simulations, instead of beginning the simulations using the 1HZH crystal structure, a snapshot during the simulations presented in Chapter 2 would be used as the initial configuration. This snapshot should be chosen such that the stress introduced by connecting the previously unconnected pair of CYS residues in the hinge region would be minimized. This could be achieved by selecting a snapshot in which, at minimum, two parameters are optimized: sulfur–sulfur atom separation distance and α-carbon–α-carbon separation distance for the CYS residues in consideration. Additionally parameters that define the angle(s) between α-carbon–β-carbon–sulfur atom and associated dihedral angles could be optimized. These parameters would be optimized such that they closely match the parameters for a connected pair of CYS residues (i.e., the pair of CYS residues connected in the hinge region in the crystal structure). Beginning from this initial configuration, simulations in which both CYS residue pairs are connected, neither of the CYS residue pairs are connected, and the opposite CYS residue pair is connected. These simulations, in addition to the simulations already performed with the hinge region disulfide bond connectivity in the crystal structure, represent all the permutations of disulfide bond connectivity available in the hinge region from this starting structure. Using the crystal structure initial configuration, another simulation should be performed in which both hinge region disulfide bonds are disconnected. This would be used to confirm that minimally one disulfide bond is required to maintain the close proximity observed between pairs of CYS residues. Comparisons of R_g and segmental motions (like those described in Chapter 2) between these simulations will help elucidate the role the disulfide bonds play in governing domain flexibility. It is hypothesized that disconnecting the disulfide bond between residues closer to the N-terminus of the heavy chains (closer to F_{ab} domains) would increase the magnitude of segmental motions and connecting the disulfide bond between residues closer to the C-terminus of the heavy chains (closer to the F_c domain) would decrease carbohydrate mobility and/or decrease separation distance between CH2 subdomains.

4.2 Coarse-Grained Scale

By sacrificing atomistic detail and coarse-graining out degrees of freedom that prove insignificant in the scope of phenomena of interest allows simulation time and length scales to reach longer and further than those achievable in fully-atomistic simulations. This technique was used to study the self-association of IgG molecules using a one–bead–per–residue mapping, Elastic Network model (ENM) for intramolecular potentials, and an intermolecular knowledge based Buckingham potential. The ENM allowed for thermal fluctuations in the structure while keeping computational costs relatively low. The one–bead–per–residue mapping with unique potentials for pairs of all 20 amino acids allowed for relatively high resolution of interacting sites on IgG molecules. After demonstrating equilibration of association frequency within 3 μs, the distribution of associating domains showed that mixed-domain interactions involving the F_c domain were the most likely associations. Additionally, interactions between $F_{ab}2$ and F_c domains occurred with higher probability than $F_{ab}1$-F_c interactions. Because the linear amino acid sequence of $F_{ab}1$ and $F_{ab}2$ are identical, these results indicated that the structure and domain orientations of IgG plays an important role in self-association. Furthermore, drawing on results from the previous atomistic studies, changes in IgG structure or domain orientations, which can be large due to high hinge region flexibility, must be considered to fully understand IgG self-association. This can be accomplished by allowing the coarse-grained IgG molecules to change structure by replacing the ENM with a different intramolecular potential. An alternative approach is to continue to use the same ENM but begin simulations with different structures. Using the ENM, it was shown that important residues in self-association could be identified through systematic mutation and also has the added benefit of maintaining a low level of computational cost. These different structures can be obtained from the two other intact IgG crystal structure entries in the Protein Data Bank or can be extracted from fully-atomistic simulations similar to those presented in Chapter 2. Those simulations were able to

capture a wide range of domain orientations which provided solvent accessibility to a different set of residues. Selection of these snapshots should be performed such that the full range of domain conformations observed atomistically can be studied in the new coarse-grained systems. PDB files obtained from atomistic simulations can be used with the structure generation script presented in the Appendix in the same way the crystal structure PDB file was used. The structure generation script takes care of generating the parameters for the intramolecular ENM potential. The same simulation methodology described in Chapter 3 is still applicable to these systems with new molecular structures, as well as the analysis techniques. In order to capture the effect of glycosylation and/or changes in carbohydrate structure, the intermolecular potential for those residues "covered" by the carbohydrate can be altered to represent their decreased accessibility or changes in surface hydrophobicity. It will be through the combination of knowledge gained from atomistic and coarse-grained simulations that accurate solution behavior at high concentrations will be understood.

Appendix

Coarse-Grained Structure Generation Python Script

The python script written for Python 2.7 used to generate the coarse-grained systems used in Chapter 3 is provided here. Minimally this script reads in a PDB file containing atoms named 'CA' (for α-carbon) and generates all bonds in ENM described in Chapter 3. Based on the input of the number of molecules, n, a system of n, molecules are placed on a cubic lattice with random orientation at the desired input concentration. The atom and bond information is then output formatted to be read by LAMMPS to the designated file name. The code presented here is v1.0 while future versions can be obtained from the Colina research group web page: `http://www2.matse.psu.edu/colinagroup/`. In v1.0, important functions used in the structure generation of these systems are provided in `functions.py` while the main functionality is provided in `structureGeneration.py`. As described in the code the usage syntax for the program is

```
python structureGeneration.py -p <pdb> [-n <number>] [-c <conc>] [-m <mut>] [-o <out>],
```

where the only required command line option is the input pdb file and default values of 2 molecules, 40 mg/ml, no mutations, and `data.lmps` are used for the rest of the input parameters unless otherwise specified. While this code works generally for any pdb file, future revisions are envisioned to accept multiple pdb files to generate heterogenous systems.

structureGeneration.py:

```
# imports
import sys
import math
import time
from operator import itemgetter
from copy import deepcopy
from functions import read_options
from functions import translate_molecule
from functions import rotate_molecule
from functions import read_file
from functions import make_bonds
```

```
from functions import get_dimensions

start = time.time()

res_codes = {
        'ALA': {'number': 1, 'mass': 71.0788},
        'ARG': {'number': 2, 'mass': 156.1876},
        'ASN': {'number': 3, 'mass': 114.1039},
        'ASP': {'number': 4, 'mass': 115.0886},
        'CYS': {'number': 5, 'mass': 103.1448},
        'GLN': {'number': 6, 'mass': 128.1308},
        'GLU': {'number': 7, 'mass': 129.1155},
        'GLY': {'number': 8, 'mass': 57.05200},
        'HIS': {'number': 9, 'mass': 137.1412},
        'ILE': {'number': 10, 'mass': 113.1595},
        'LEU': {'number': 11, 'mass': 135.1595},
        'LYS': {'number': 12, 'mass': 128.1742},
        'MET': {'number': 13, 'mass': 131.1986},
        'PHE': {'number': 14, 'mass': 147.1766},
        'PRO': {'number': 15, 'mass': 97.11670},
        'SER': {'number': 16, 'mass': 87.0782},
        'THR': {'number': 17, 'mass': 101.1051},
        'TRP': {'number': 18, 'mass': 186.2133},
        'TYR': {'number': 19, 'mass': 163.1760},
        'VAL': {'number': 20, 'mass': 99.1326}
}

cutoff = 11.5  # cutoff distance for EBN
system = list() # initialize list to hold all molecules

# reads command line arguments and returns parameters
input_pdb, num_molecules, concentration, mutations, output_file = read_options(sys.argv)

# reads input pdb file and returns list of alpha carbon atoms
atoms = read_file(input_pdb, mutations)

# reads list of atoms and returns all ENM bonds
bonds = make_bonds(atoms, cutoff)

# add list of atoms to system : this is molecules # 1
system.append(atoms)
temp_num_mol = 0 # temporary counter for number of molecules in the system

# create simple cubic lattice
for k in range(0, int(math.ceil(int(num_molecules) ** (1 / 3.0)))):
        for j in range(0, int(math.ceil(int(num_molecules) ** (1 / 3.0)))):
                for i in range(0, int(math.ceil(int(num_molecules) ** (1 / 3.0)))):
                        # if this is not the first molecule create copy then translate
                        # and rotate molecule, then add to system
                        if temp_num_mol != 0:
                                # output so it looks like something is happening
                                print('Attempting to add molecule ' +
                                str(tempNumMol + 1) + ' to simulation box.'
                                + str(i) + ' ' + str(j) + ' ' + str(k))
                                temp_atoms = deepcopy(atoms)
                                temp_atoms = translate_molecule(temp_atoms, i, j, k)
                                temp_atoms = rotate_molecule(temp_atoms, system)
                                system.append(temp_atoms)
                        # update temporary molecule counter
                        temp_num_mol += 1
                        # break out of loops if we have added
                        # the correct number of molecules
                        if temp_num_mol >= num_molecules:
                                break
                if temp_num_mol >= num_molecules:
```

```
                        break
        if temp_num_mol >= num_molecules:
                break

# find box dimensions based on desired concentration
xlo, xhi, ylo, yhi, zlo, zhi = get_dimensions(system, concentration)

# number of unique bond types
nbtypes = len(bonds)
# number of bonds is multiplied by number of molecules
nbonds = nbtypes * num_molecules
# number of beads is residues * number of molecules
natoms = len(atoms) * num_molecules

# write lammps data
output = open(output_file, 'w+')
output.write(' LAMMPS DATA FILE\n\n')
output.write(' %12d atoms\n' % natoms)
output.write(' %12d bonds\n' % nbonds)
output.write(' %12d angles\n' % 0)
output.write(' %12d dihedrals\n' % 0)
output.write(' %12d impropers\n' % 0)
output.write('\n')
output.write(' %12d atom types\n' % 20)
output.write(' %12d bond types\n' % nbtypes)
output.write(' %12d angle types\n' % 0)
output.write(' %12d dihedral types\n' % 0)
output.write(' %12d improper types\n' % 0)
output.write('\n')
output.write(' %16.6f  %16.6f    xlo xhi\n' % (xlo, xhi))
output.write(' %16.6f  %16.6f    ylo yhi\n' % (ylo, yhi))
output.write(' %16.6f  %16.6f    zlo zhi\n' % (zlo, zhi))
output.write('\n')
output.write(' Masses\n')
output.write('\n')
for key in sorted(res_codes):
        output.write(' %6d    %12.6f\n' % (res_codes[key]['number'],
        res_codes[key]['mass']))
output.write('\n')
output.write(' Bond Coeffs\n')
output.write('\n')
for bond in sorted(bonds, key=itemgetter('type')):
        output.write(' %6d %12.6f %12.6f\n' % (bond['type'], bond['k'],
        bond['dd']))
output.write('\n')
output.write(' Atoms\n')
output.write('\n')

for n in range(0, num_molecules):
        for atom in sorted(system[n], key=itemgetter('number')):
                output.write(' %8d %6d %6d %9.4f %9.4f %9.4f\n' % (
                (atom['number']) + n * len(atoms), n + 1,
                res_codes[atom['type']]['number'],
                atom['x'], atom['y'], atom['z']))
output.write('\n')
output.write(' Bonds\n')
output.write('\n')
for n in range(0, num_molecules):
        for bond in sorted(bonds, key=itemgetter('type')):
                output.write(' %8d %6d %6d %6d\n' % ((bond['type']) + n * nbtypes,
                bond['type'], bond['a1'] + n * len(atoms),
                bond['a2'] + n * len(atoms)))
output.write('\n')
```

```
output.write('\n')

output.close()

end = time.time()

print end - start

sys.exit(1)
```

functions.py:

```
import random
import math
import re
import sys
from operator import itemgetter

res_codes = {
        'ALA': {'number': 1, 'mass': 71.0788},
        'ARG': {'number': 2, 'mass': 156.1876},
        'ASN': {'number': 3, 'mass': 114.1039},
        'ASP': {'number': 4, 'mass': 115.0886},
        'CYS': {'number': 5, 'mass': 103.1448},
        'GLN': {'number': 6, 'mass': 128.1308},
        'GLU': {'number': 7, 'mass': 129.1155},
        'GLY': {'number': 8, 'mass': 57.05200},
        'HIS': {'number': 9, 'mass': 137.1412},
        'ILE': {'number': 10, 'mass': 113.1595},
        'LEU': {'number': 11, 'mass': 135.1595},
        'LYS': {'number': 12, 'mass': 128.1742},
        'MET': {'number': 13, 'mass': 131.1986},
        'PHE': {'number': 14, 'mass': 147.1766},
        'PRO': {'number': 15, 'mass': 97.11670},
        'SER': {'number': 16, 'mass': 87.0782},
        'THR': {'number': 17, 'mass': 101.1051},
        'TRP': {'number': 18, 'mass': 186.2133},
        'TYR': {'number': 19, 'mass': 163.1760},
        'VAL': {'number': 20, 'mass': 99.1326}
}

pi = math.acos(-1)
random.seed(5)


def read_options(args):

        pdb = ''
        num = '2'
        conc = '40'
        mut = ''
        out = 'data.lmps'

        my_args = list(args)
        flag = my_args.pop(0)

        while len(my_args) > 0:
                flag = my_args.pop(0)
                if flag == '-p':
                        pdb = my_args.pop(0)
                elif flag == '-n':
                        num = my_args.pop(0)
                        try:
                                num = int(num)
                        except ValueError:
                                print('\nPlease enter a valid integer for
```

```
                                the number of molecules\n')
                                exit(1)
                elif flag == '-c':
                        conc = my_args.pop(0)
                        try:
                                conc = int(conc)
                        except ValueError:
                                print('\nPlease enter a valid integer for
                                the concentration\n')
                                exit(1)
                elif flag == '-m':
                        mut = my_args.pop(0)
                        mut = mut.split(',')
                elif flag == '-o':
                        out = my_args.pop(0)
                else:
                        print('\nCoarse-Grained Structure Generation Script\n')
                        print('Usage: python structureGeneration.py -p <pdb file>
                        [-n <number of molecules>] [-c <concentration>]
                        [-m <mutations>] [-o <out file>]')
                        print('Default values: -n 2 -c 40 -o data.lmps')
                        exit(1)

        if pdb != '':
                print('\npdb: ' + pdb)
                print('number of molecules: ' + str(num))
                print('concentration: ' + str(conc) + ' mg/ml')
                print('mutations: ' + ', '.join(mut))
                print('output file: ' + out)
                return pdb, int(num), conc, mut, out
        else:
                print('\npdb file must be provided\n')
                exit(1)


def overlap(atoms, system):
        for m in range(0, len(system)):
                for i in range(0, len(atoms)):
                        for n in range(0, len(system[m])):
                                dd = math.sqrt(
                                (atoms[i]['x']-system[m][n]['x']) ** 2 +
                                (atoms[i]['y']-system[m][n]['y']) ** 2 +
                                (atoms[i]['z']-system[m][n]['z']) ** 2)
                                if dd <= 12:
                                        return True
        return False


# function that translates molecule
# by 100 units in each i,j,k direction
def translate_molecule(atoms, i, j, k):
        for n in range(0, len(atoms)):
                atoms[n]['x'] += i * 100
                atoms[n]['y'] += j * 100
                atoms[n]['z'] += k * 100
        return atoms


def rotate_molecule(atoms, system):

        min_x = sys.float_info.max
        max_x = sys.float_info.min
        min_y = sys.float_info.max
        max_y = sys.float_info.min
        min_z = sys.float_info.max
        max_z = sys.float_info.min
```

```
        for i in range(0, len(atoms)):
                min_x = atoms[i]['x'] if atoms[i]['x'] < min_x else min_x
                max_x = atoms[i]['x'] if atoms[i]['x'] > max_x else max_x
                min_y = atoms[i]['y'] if atoms[i]['y'] < min_y else min_y
                max_y = atoms[i]['y'] if atoms[i]['y'] > max_y else max_y
                min_z = atoms[i]['z'] if atoms[i]['z'] < min_z else min_z
                max_z = atoms[i]['z'] if atoms[i]['z'] < max_z else max_z

        dx = max_x - min_x
        dy = max_y - min_y
        dz = max_z - min_z

        avg_x = (min_x + max_x) / 2
        avg_y = (min_y + max_y) / 2
        avg_z = (min_z + max_z) / 2

        for i in range(0, len(atoms)):
                atoms[i]['x'] -= avg_x
                atoms[i]['y'] -= avg_y
                atoms[i]['z'] -= avg_z

        # generate random value for rotation from 0 to 2pi
        ran_x_twist = random.random() * 2 * pi
        ran_y_twist = random.random() * 2 * pi
        ran_z_twist = random.random() * 2 * pi

        # apply rotation matrix
        for m in range(0, len(atoms)):

                atoms[m]['x'], atoms[m]['y'], atoms[m]['z'] = (
                 (atoms[m]['x'] * math.cos(ran_y_twist) * math.cos(ran_z_twist) +
                  atoms[m]['z'] * math.sin(ran_y_twist) - atoms[m]['y'] *
                  math.cos(ran_y_twist) * math.sin(ran_z_twist)),
                 (-1 * atoms[m]['z'] * math.cos(ran_y_twist) * math.sin(ran_x_twist) +
                  atoms[m]['x'] * (math.cos(ran_z_twist) * math.sin(ran_x_twist) *
                  math.sin(ran_y_twist) + math.cos(ran_x_twist) * math.sin(ran_z_twist)) +
                  atoms[m]['y'] * (math.cos(ran_x_twist) * math.cos(ran_z_twist) -
                  math.sin(ran_x_twist) * math.sin(ran_y_twist) * math.sin(ran_z_twist))),
                 (atoms[m]['z'] * math.cos(ran_x_twist) * math.cos(ran_y_twist) +
                  atoms[m]['x'] * (-1 * math.cos(ran_x_twist) * math.cos(ran_z_twist) *
                  math.sin(ran_y_twist) + math.sin(ran_x_twist) * math.sin(ran_z_twist)) +
                  atoms[m]['y'] * (math.cos(ran_z_twist) * math.sin(ran_x_twist) +
                  math.cos(ran_x_twist) * math.sin(ran_y_twist) * math.sin(ran_z_twist))))

        for i in range(0, len(atoms)):
                atoms[i]['x'] += avg_x
                atoms[i]['y'] += avg_y
                atoms[i]['z'] += avg_z

        if overlap(atoms, system):
                print 'Overlap: attempting rotation again'
                rotate_molecule(atoms, system)
        else:
                pass

        return atoms


def read_file(my_file, mut):

        try:
                f = open(my_file)
        except IOError:
                print(my_file + ' does not exist or is not readable')
```

```
                exit(1)

        atoms = list()
        num_atoms = 0

        line = f.readline().strip().split()
        x_column = ''
        while line:
                if line[0].upper() == 'ATOM':
                        if x_column == '':
                                for i in range(0, len(line)):
                                        if re.search('^[0-9]+\.[0-9]+$',
                                        line[i]) is not None:
                                                x_column = i
                                                break
                        if line[2].upper() == 'CA':
                                num_atoms += 1
                                if line[3].upper() == 'CYX':
                                        line[3] = 'CYS'
                                if str(num_atoms) in mut:
                                        print('Mutation: ' +
                                        str(num_atoms) + ' to ALA')
                                        atoms.append({'number': num_atoms,
                                                'type': 'ALA',
                                                'x': float(line[x_column]),
                                                'y': float(line[x_column + 1]),
                                                'z': float(line[x_column + 2])})
                                else:
                                        atoms.append({'number': num_atoms,
                                                'type': line[3].upper(),
                                                'x': float(line[x_column]),
                                                'y': float(line[x_column + 1]),
                                                'z': float(line[x_column + 2])})
                line = f.readline().strip().split()

        return atoms


def make_bonds(atoms, cutoff):

        bonds = list()
        num_bonds = 0

        atoms = sorted(atoms, key=itemgetter('number'))

        for a1 in range(0, len(atoms)):
                for a2 in range(a1 + 1, len(atoms)):
                        dd = math.sqrt(
                        (atoms[a1]['x'] - atoms[a2]['x']) ** 2 +
                        (atoms[a1]['y'] - atoms[a2]['y']) ** 2 +
                        (atoms[a1]['z'] - atoms[a2]['z']) ** 2)
                        if dd <= cutoff:
                                num_bonds += 1
                                if dd < 4.0:
                                        k = 20.0
                                elif dd < 7.0:
                                        k = 5.0
                                else:
                                        k = 1.0
                                bonds.append({'type': num_bonds,
                                                'a1': atoms[a1]['number'],
                                                'a2': atoms[a2]['number'],
                                                'k': k,
                                                'dd': dd})

        return bonds
```

```
def get_dimensions(system, concentration):

        mass = 0
        for i in range(0, len(system[0])):
                mass += res_codes[system[0][i]['type']]['mass']

        min_x = sys.float_info.max
        max_x = sys.float_info.min
        min_y = sys.float_info.max
        max_y = sys.float_info.min
        min_z = sys.float_info.max
        max_z = sys.float_info.min

        for m in range(0, len(system)):
                for n in range(0, len(system[0])):
                        min_x = system[m][n]['x'] if system[m][n]['x'] <
                        min_x else min_x
                        max_x = system[m][n]['x'] if system[m][n]['x'] >
                        max_x else max_x
                        min_y = system[m][n]['y'] if system[m][n]['y'] <
                        min_y else min_y
                        max_y = system[m][n]['y'] if system[m][n]['y'] >
                        max_y else max_y
                        min_z = system[m][n]['z'] if system[m][n]['z'] <
                        min_z else min_z
                        max_z = system[m][n]['z'] if system[m][n]['z'] >
                        max_z else max_z

        dx = max_x - min_x
        dy = max_y - min_y
        dz = max_z - min_z

        target_dimension = (1 / float(concentration) * len(system) /
        6.02 / 10 ** 23 * mass * 10 ** 27) ** (1 / 3.0)

        if target_dimension < dx or target_dimension < dy or
        target_dimension < dz:
                print('Error: molecules cannot fit in simulation box in
                this orientation at this concentration. Try again or
                decrese target concentration.')
                sys.exit(1)
        else:
                        pm_x = (target_dimension - dx) / 2
                        pm_y = (target_dimension - dy) / 2
                        pm_z = (target_dimension - dz) / 2
                        xlo = min_x - pm_x
                        xhi = max_x + pm_x
                        ylo = min_y - pm_y
                        yhi = max_y + pm_y
                        zlo = min_z - pm_z
                        zhi = max_z + pm_z

        return xlo, xhi, ylo, yhi, zlo, zhi
```

References

[1] E. E. David, J.; Mathews, M. V.; McDonald, H. S. *J. Acoust. Soc. Am.* **1959,** *31,* D2.

[2] Jennings, N. H.; Dickins, J. H. *Manage. Sci.* **1958,** *5,* 106–120.

[3] Fox, P.; Lehman, F. *Traffic Quarterly* **1967,** *21,* 53–66.

[4] Levitt, M.; Warshel, A. *Nature* **1975,** *235,* 694–3696.

[5] Sternberg, M.; Thornton, J. *Nature* **1978,** *271,* 15–20.

[6] McCammon, J.; Gelin, B.; Karplus, M. *Nature* **1977,** *267,* 585–590.

[7] Levitt, M. *Nature Structural Biology* **2001,** *8,* 392–393.

[8] Blundell, T.; Sibanda, B.; Sternberg, M.; Thornton, J. *Nature* **1987,** *326,* 347–352.

[9] Honeycutt, J. D.; Thirumalai, D. *Prod. Natl. Acad. Sci. USA* **1990,** *87,* 3526–3529.

[10] Brandt, J. P.; Patapoff, T. W.; Aragon, S. R. *Biophysical Journal* **2010,** *99,* 905–913.

[11] Porter, R. *Nature* **1958,** *182,* 670–671.

[12] Humphrey, W.; Dalke, A.; Schulten, K. *J. Molec. Graphics* **1996,** *14,* 33–38.

[13] Amzel, L.; Poljak, R. *Ann. Rev. Biochem.* **1979,** *48,* 961–997.

[14] Harris, L. J.; Larson, S. B.; Hasel, K. W.; McPherson, A. *Biochemistry* **1997,** *36,* 1581–1597.

[15] Harris, L. J.; Skaletsky, E.; McPherson, A. *Journal of Molecular Biology* **1998,** *275,* 861–872.

[16] Saphire, E.; Parren, P.; Pantophlet, R.; Zwick, M.; Morris, G.; Rudd, P.; Dwek, R.; Stanfield, R.; Burton, D.; Wilson, I. *Science* **2001,** *293,* 1155–1159.

[17] Karplus, M.; McCammon, J. *Nature Structural Biology* **2002,** *9,* 646–652.

[18] Bahar, I.; Atilgan, A. R.; Erman, B. *Folding and Design* **1997,** *2,* 173–181.

[19] Atilgan, A. R.; Durell, S. R.; Jernigan, R. L.; Demirel, M. C.; Keskin, O.; ; Bahar, I. *Biophysical Journal* **2001,** *80,* 505–515.

[20] Monticelli, L.; Kandasamy, S. K.; Periole, X.; Larson, R. G.; Tieleman, D. P.; Marrink, S. *J. Chem. Theory and Comput.* **2008,** *4,* 819–834.

[21] Bond, P.; Sansom, M. S. P. *J. Am. Chem. Soc.* **2006,** *128,* 2697–2704.

[22] Russel, R.; Millettt, I. S.; Tate, M. W.; Kwok, L. W.; Nakatani, B.; Gruner, S. M.; Mochrie, S. G. J.; Pande, V.; Doniach, S.; Herschlag, D.; Pollack, L. *Prod. Natl. Acad. Sci. USA* **2002,** *99,* 4266–4271.

[23] Bonilla, F.; Geha, R. Blood: Principles and Practice of Hematology. In , 2nd ed.; Handin, R.; S.E. Lux, T. S., Eds.; Lippincott Williams & Wilkins: Philadelphia, PA, 2003; Chapter 22: Primary Immunodeficiency Diseases and Immunoproteins.

[24] Dalziel, M.; Crispin, M.; Scanlan, C.; Zitzmann, N.; Dwek, R. *Science* **2014,** *343,* 37–45.

[25] Chennamsetty, N.; Helk, B.; Voynov, V.; Kayser, V.; Trout, B. L. *Journal of Molecular Biology* **2009,** *391,* 404–413.

[26] Voynov, V.; Chennamsetty, N.; Kayser, V.; Helk, B.; Forrer, K.; Zhang, H.; Fritsch, C.; Heine, H.; Trout, B. *PLoS ONE* **2009,** *4,* e8425.

[27] Wang, X.; Kumar, S.; Buck, P.; Singh, S. *Proteins* **2013,** *81,* 443–460.

[28] Kortkhonjia, E.; Brandman, R.; Zhou, J.; Voelz, V.; Chorny, I.; Kabakoff, B.; Patapoff, T.; Dill, K.; Swartz, T. *mAbs* **2013,** *5,* 306–322.

[29] Pucic, M. *et al. Molecular and Cellular Proteomics* **2011,** *10,* M111.010090.

[30] Wormald, M.; Rudd, P.; Harvey, D.; Chang, S.; Scragg, I.; Dwek, R. *Biochemistry* **1997,** *36,* 1370–1380.

[31] Iida, S.; Misaka, H.; Inoue, M.; Shibata, M.; Nakano, R.; Yamane-Ohnuki, N.; Wakitani, M.; Yano, K.; Shitara, K.; Satoh, M. *Clinical Cancer Research* **2006,** *12,* 2879–2887.

[32] Kanda, Y.; Yamada, T.; Mori, K.; Okazaki, A.; Inoue, M.; Kitajima-Miyama, K.; Kuni-Kamochi, R.; Nakano, R.; Yano, K.; Kakita, S.; Shitara, K.; Satoh, M. *Glycobiology* **2006,** *17,* 104–118.

[33] Parekh, R.; Dwek, R.; Sutton, B.; Fernandes, D.L., L. A.; Stanworth, D.; Rademacher, T. *Nature* **1985,** *316,* 452–457.

[34] Malhotra, R.; Wormald, M.; Rudd, P.; Fischer, P.; Dwek, R.; Sim, R. *Nature Medicine* **1995,** *1,* 237–243.

[35] Anthony, R.; Nimmerjahn, F.; Ashline, D.; Reinhold, V.; Paulson, J.; Ravetch, J. *Science* **2008,** *320,* 373–376.

[36] Collin, M.; Ehlers, M. *Experimental Dermatology* **2013,** *22,* 511–514.

[37] DeLano, W.; Ultsch, M.; de Vos, A.; Wells, J. *Science* **2000,** *287,* 1279–1283.

[38] Idusogie, E.; Presta, L.; Gazzano-Santoro, H.; Totpal, K.; Wong, P.; Ultsch, M.; Meng, Y.; Mulkerrin, M. *J Immunol* **2000,** *164,* 4178–4184.

[39] Barb, A.; Prestegard, J. *Nature Chemical Biology* **2011,** *7,* 147–153.

[40] Case, D. *et al.* **2012,** *University of California, San Francisco,*.

[41] Hornak, V.; Abel, R.; Okur, A.; Strockbrine, B.; Roitberg, A.; Simmerling, C. *Proteins* **2006,** *65,* 712–725.

[42] Kirschner, K.; Yongye, A.; Tschampel, S.; Daniels, C.; Foley, B.; Woods, R. *J. Comput. Chem.* **2008,** *29,* 622–655.

[43] Salomon-Ferrer, R.; Goetz, A.; Poole, D.; Le Grand, S.; Walker, R. *J. Chem. Theory Comput.* **2013,** *9,* 3878-3888.

[44] Clark, N.; Zhang, H.; Krueger, S.; Lee, H.; Ketchem, R.; Kerwin, B.; Kanapuram, S.; Treuheit, M.; McAuley, A.; Curtis, J. *J. Phys. Chem. B* **2013,** *117,* 14029–14038.

[45] Rayner, L.; Kadkhodayi-Kholghi, N.; Heenan, R.; Gor, J.; Dalby, P.; Perkins, S. *J. Mol. Biol.* **2013,** *425,* 506–523.

[46] Pilz, I.; Schwarz, E.; Durchschein, W.; Licht, A.; Sela, M. *Proc. Natl. Acad. Sci. USA* **1980,** *77,* 117–121.

[47] Borrok, M. J.; Jung, S. T.; Kang, T. H.; Monzingo, A. F.; Georgiou, G. *ACS Chem. Biol.* **2012,** *7,* 1596–1602.

[48] Jo, S.; Lee, H. S.; Skolnick, J.; Im, W. *PLoS Comput. Biol.* **2013,** *9,* e1002946.

[49] Chaudhri, A.; Zarraga, I. E.; Kamerzell, T. J.; Brandt, J. P.; Patapoff, T. W.; Shire, S. J.; Voth, G. A. *J. Phys. Chem. B* **2012,** *116,* 8045–8057.

[50] Yang, L.; Tan, C.; Hsieh, M.; Wang, J.; Duan, Y.; Cieplak, P.; Caldwell, J.; Kollman, P. A.; Luo, R. *J. Phys. Chem. B* **2006,** *110,* 13166–13176.

[51] Agrawal, N. J.; Kumar, S.; Wang, X.; Helk, B.; Singh, S. K.; Trout, B. L. *J. Pharm. Sci.* **2011,** *100,* 5081–5095.

[52] Joubert, M. K.; Luo, Q.; Nashed-Samuel, Y.; Wypych, J.; Narhi, L. O. *J. Biol. Chem.* **2011,** *286,* 25118–25133.

[53] Lin, P.; Colina, C. M. *in preparation,*.