Computational and Experimental Studies of

Collagen and Related Diseases

by

Chen Yang

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

Aug 16, 2005

Author_____
Department of Electrical Engineering and Computer Science
August 16, 2005

Certified by_____
Collin M. Stultz
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Computational and Experimental Studies of Collagen and Related Diseases
by
Chen Yang

Submitted to the
Department of Electrical Engineering and Computer Science

Aug 16, 2005

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Collagen is the most abundant proteins in mammals, and collagen degradation is a process that may be associated with many diseases. In this research we use collagen-like peptides that model both cleavage and noncleavage sites of collagen type III to study the stability and degradation of collagen as a function of amino acid sequence variation. Molecular Dynamics is used to compute the potential of mean force of these collagen-like peptides and predict their triple helical content. The predicted result is then compared with experimental results from Circular Dichroism spectroscopy. Similar studies are done on mutant peptides of collagen from the Ehlers-Danlos Syndrome type IV.

Thesis Supervisor: Collin M. Stultz MD, PhD
Title: Assistant Professor, Harvard-MIT Division of Health Sciences and Technology &
MIT Department of Electrical Engineering and Computer Science

Acknowledgements

My biggest thanks go to Collin, who made all of this possible. I can't thank him enough for all of his supervising, teaching, funding, and revising—he really was there every step of the way. It certainly has been a heck of a year, but it was an amazing year! I also thank him for being there for me and for giving me advice during the stressful times. I wish him the best for the many future years at MIT, Harvard, and anywhere else he goes. Congratulations to him on starting a wonderful lab, and many more congrats on the wonderful marriage!

My next thanks go to Ramon, who has been my biggest collaborator. Next to Collin, he is the person I turned to and he helped me with everything from CD to MD. I also thank him for being there and listening to me during the difficult times. He truly has been a great friend and teacher, and I am sure he will make his dream come true and become an awesome professor one day.

Many thanks also go to Amelia for the laughs she brought to the lab! She is the person who kept us on top of things. I thank her also for revising my thesis and correcting all the embarrassing errors.

More thanks go to Peter for revising and for being a great friend!

My final thanks go to my girlfriend Yan, my parents, and family, whose help and support have reached me in many more ways than one.

Table of Contents

Chapter 1

Introduction

Collagen is the most abundant protein in mammals and makes up roughly one quarter of all proteins in the human body. It is found in many tissues and organs, especially in connective tissues including cartilage, bone, blood vessels, and skin. Collagen is a major structural protein that gives tensile strength to bones, resilience to skins, and a tough yet flexible quality to vessels and cartilage. Out of the more than twenty types of collagen discovered to date, types I, II, and III are the most abundant. A highly conserved structure is also found in all collagen—the triple helix (Gordon and Olson, 1990; Jacenko *et al*, 1991).

The collagen triple helix is significantly different from the globular structure of most known proteins. The triple helix is an overall rod-like structure (Figure 1.1) composed of three polypeptide chains that coil around each other, where each chain is approximately 1000 residues long. Nearly one third of the amino acids in each polypeptide chain are glycine residues and another 15-30% are prolines or *trans*-4-hydroxyprolines. Other types of hydroxyprolines are also found, but they exist in much smaller quantities. A repeating P-O-G motif is conserved in all chains, where the P is proline, O is hydroxyproline or hydroxylysine, and G is glycine.

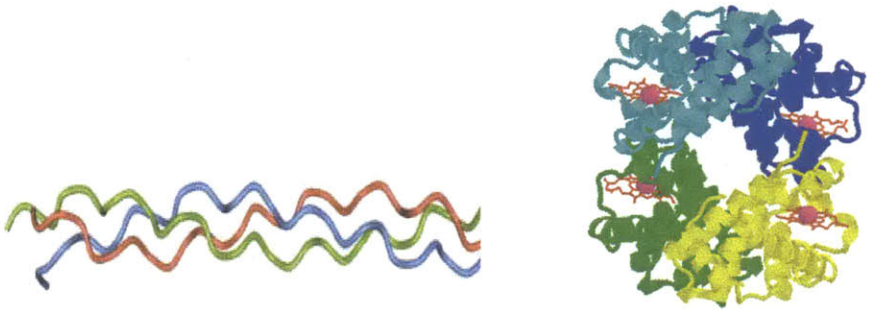


Figure 1.1. The triple helical structure of collagen (left) versus the globular protein structure of hemoglobin (right), adapted from (Hata, 2001).

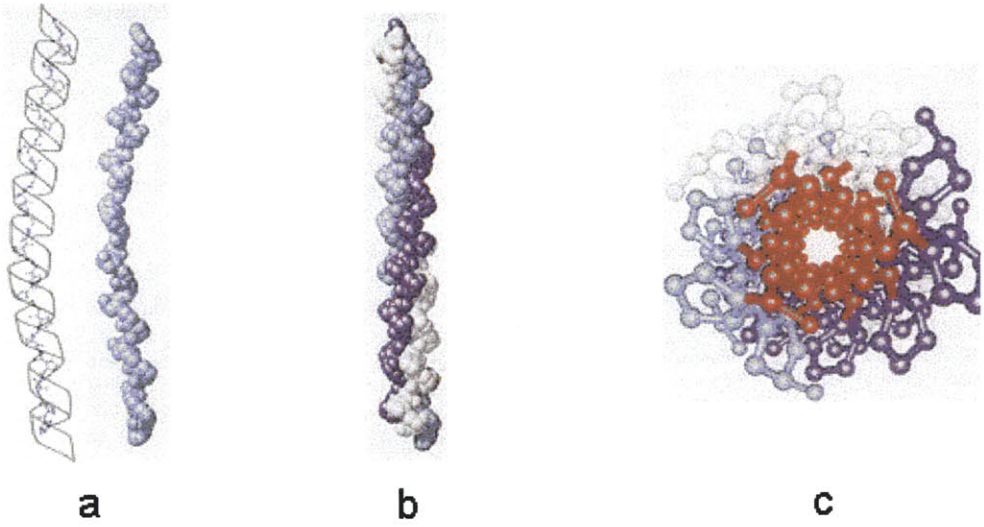


Figure 1.2. (a) The individual collagen polypeptide chains are left handed. (b) The overall triple helical collagen is right handed. (c) Top view of collagen triple helix, demonstrating the glycine residues (red) in the center (Greger, 2001).

While each of the three polypeptide chains occurs in a left handed helix, the three chains are coiled into a right handed triple helix (Figure 1.2). Every 3.3 residues of each chain passes through the center of the triple helix structure. Due to the tight packing only a glycine amino acid, the smallest amino acid with no side chain (or a side chain composed of only a single hydrogen atom), can fit into this site. The bulky and inflexible side chains of proline and hydroxyproline provide a rigid support that further locks the triple helix structure in place. Finally, a number of molecular interactions also place constraints on the triple helix structure, such as the staggered backbone hydrogen bonds between glycine's NH_2^+ group and the proline's CO^- group.

1.1 The Biosynthesis of Collagen

The biosynthesis of collagen is a complicated process involving multiple post-translational modifications which are essential for obtaining the correct collagen structure (Wang, 2002). The post-translational modifications consist of two main stages: intracellular and extracellular modifications. During intracellular modification, the signal peptides are first removed by signal peptidases, enabling the translocation of procollagen from the rough endoplasmic reticulum (RER) membrane to the lumen. In the lumen, hydroxylation occurs under different types of hydroxylases, adding an OH group to select molecules. Selected proline residues are hydroxylated to 3-, 4-, or 5-hydroxyproline, while select lysine residues are hydroxylated to 5-hydroxylysine. The addition of polysaccharide chains to target protein molecules, glycosylation, then takes place on selected hydroxylysines to produce galactosylhydroxylysines, then again on select galactosylhydroxylysines to produce galactosylgalactosylhydroxylysines. At the completion of hydroxylation and glycosylation, the three individual procollagen chains associate using disulfide bonds on the C-terminal end. The chains then fold from the C-terminus to the N-terminus, forming the basic triple helical structure (Wang, 2002). Finally, procollagen is secreted into the extracellular space via the Golgi complex as the last step in intracellular modification.

In the extracellular stage, three more important steps come into play to further stabilize the triple helix structure. First, the loosely coiled peptides at the ends of the triple helix are removed by N-proteinase and C-proteinase, at which time the procollagen becomes mature collagen monomers (Wang, 2002). Second, the collagen triple helix aggregates together in an ordered fashion to form fibrils and other structural patterns depending on application, and strengthens the collagen structure (Figure 1.4). Finally, crosslinks form in numerous additional steps to further stabilize the aggregate structures. A general summary of the post-translational step is depicted in Figure 1.3.

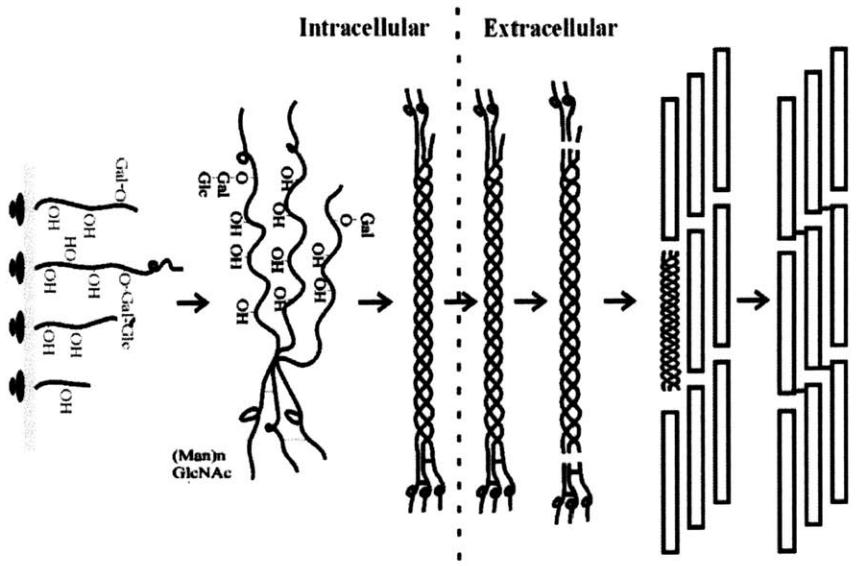


Figure 1.3. A summary of the post-translational stages of biosynthesis (modified from Wang, 2002). From left to right: intracellular hydroxylation, glycosylation, triple helix formation, extracellular removal of end terminals, formation of aggregate structures, and formation of cross links.

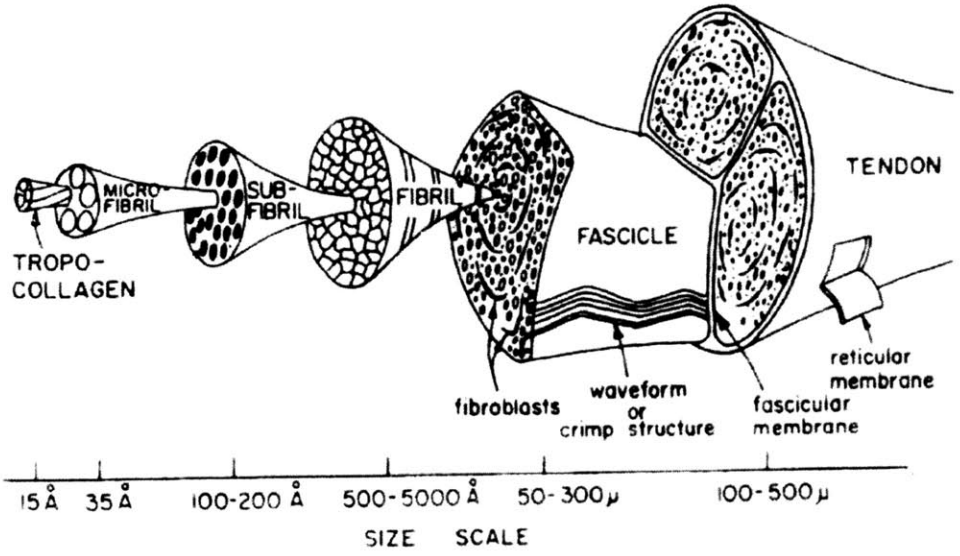


Figure 1.4. The structural hierarchy of a tendon, one form of collagen aggregate (Fratzl *et al*, 1998).

The homeostasis of collagen is maintained by a fine regulation of collagen biosynthesis and degradation in cells and tissues. A significant amount of research has been done on

collagen biosynthesis. In this study, we focus on collagen degradation, which appears to play an important role in many disease processes.

1.2 The Degradation of Collagen

It is known that most proteins have a well defined structure that needs to be maintained for its proper function, and aberrations in a protein's structure can often lead to disastrous results. It is believed that illnesses such as Mad Cow Disease and Alzheimer's Disease, for example, are caused by disruption in proteins' normal structure. Collagen is no exception to this rule.

Collagen degradation has been associated with many diseases including arthritis, tumor metastasis, emphysema, and atherosclerosis (McDonnell *et al*, 1999; Celentano and Frishman, 1997). The cleavage of collagen type III, a homotrimer collagen found in the arterial wall of the extracellular matrix, has been implicated in the pathogenesis of atherosclerotic plaque rupture. When the protective collagen type III layer surrounding the plaque is broken, agents that could potentially cause blood clots are released into the blood stream, leading to acute myocardial infarction and death (Gaziano, 2001). Preventing the degradation of type III collagen may therefore enable us to develop novel therapies for the treatment of heart attacks.

Up until recently, it was believed that the triple helix is a stable structure that is difficult to cleave under normal conditions. Yet cleavage of collagen still occurs regularly in both natural bodily processes and undesirable situations. Matrix Metalloproteinases (MMPs) types 1, 3, and 8, which are the most common mammalian collagenases, target collagen at a unique cleavage site that is characterized by either a glycine-leucine or a glycine-isoleucine bond followed by either an alanine or leucine residue (Fields, 1991; McDonnell *et al*, 1999). However this cleavage site, called the scissile bond, is usually concealed from solvents (Stultz, 2002; Kramer *et al*, 1999) (Figure 1.5). It is therefore not understood how MMPs acquire access to the cleavage site. Furthermore, the width of the MMP active binding site is approximately 0.8nm, whereas the diameter of a triple helical

collagen is about 1.5nm, making it physically unlikely for MMPs to bind to the collagen cleavage site (Figure 1.6).

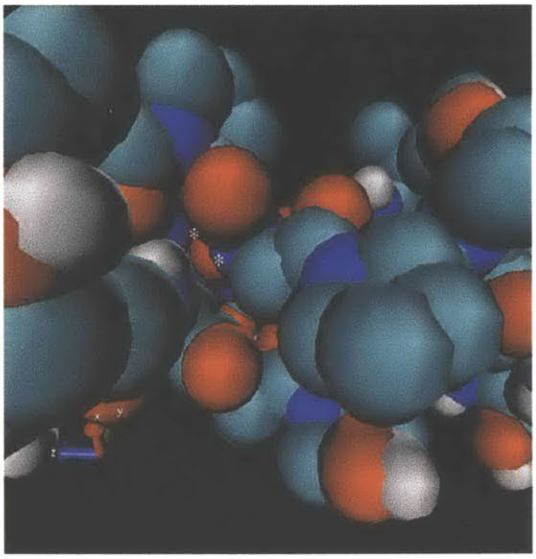


Figure 1.5. The unique cleavage site of collagen (starred) is not accessible to MMP or solvents.

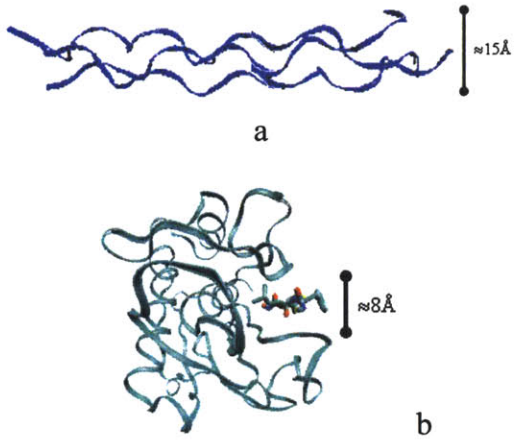
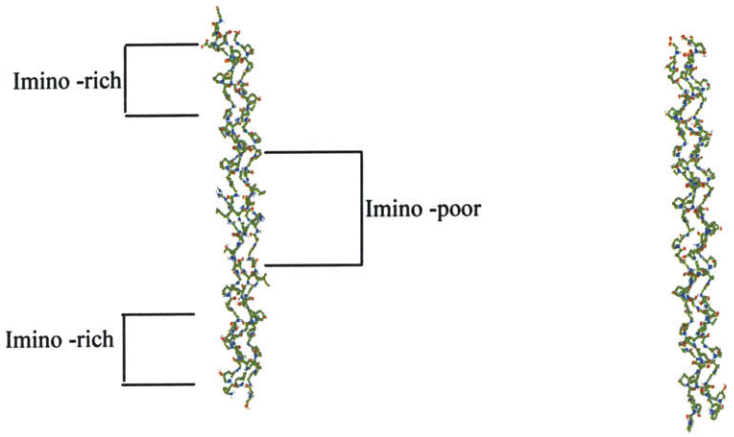


Figure 1.6. (a). The triple helix structure of collagen has a diameter of approximately 15Å (PDB 1BKV). (b). Backbone trace of the structure of the catalytic domain of MMP-16, bound to the inhibitor Batimastat. The opening of the catalytic site spans approximately 8Å (PDB 1RM8).

To solve this paradox, sequence analysis was done to highlight the potential differences in regions at and near the cleavage sites. It was noticed that regions near collagenase cleavage sites have a distinct sequence, with an imino-rich region immediately preceding the cleavage site, an imino-poor region immediately after the cleavage site, and an arginine molecule always located downstream from the cleavage site within the imino-poor region (Fields, 1991). Here *imino acids* are defined as either proline or hydroxyproline. Using these observations, some general rules of when collagen can be cleaved were derived.

X-ray crystallography and NMR studies were done to analyze whether these sequence variations lead to a difference in collagen structure. Different types of collagen-like peptides were synthesized and then compared. Among these peptides the most important ones include the $(POG)_{10}$ peptide, which represents the imino-rich region of collagen, and

a peptide called T3-785. The T3-785 peptide, sequenced $(POG)_3ITGARGLAG(POG)_4$, represents the imino-poor region of collagen and is equivalent to the region near the cleavage site of collagen type III. While the analysis of these two peptides using X-ray crystallography (Figure 1.7) did not show any significant differences in structure, the NMR data showed that the central glycine in T3-785 is much more exposed relative to the central glycine in $(POG)_{10}$ (Fan *et al*, 1993). The region of collagen resembling the T3-785 peptide or near imino-poor regions might therefore be more susceptible to attack by the collagenases.



T3-785 (IP): P-O-G-P-O-*G*P*-O-G-I-T-G-A-R-G-L-A-G-P-O-G-P-O-G-P-O-G-P-O-G
Imino-rich: P-O-G-P-O-G-P-O-G-P-O-G-P-O-A-P-O-G-P-O-G-P-O-G-P-O-G-P-O-G

Figure 1.7. The structure of the T3-785 imino-poor peptide (left) and the imino-rich peptide (right) from X-ray crystallography. The starred bond represents the corresponding position of the cleavage site (Stultz, 2002).

The most recent studies show that in collagen-like peptides, low imino acid content might cause partial unfolding of the triple helix structure in specific localized areas, therefore enabling cleavage of collagen in these areas (Stultz, 2002). In contrast, high imino content such as that of the imino-rich regions may provide a major driving force for the generation and stabilization of the triple helical structure (Bhatnagar *et al*, 1988). Hyperglycemia has been shown to alter the distribution of different collagen conformations, favoring the partially unfolded states and thus increasing cleavage (Stultz and Edelman, 2003). Still, other studies suggest that collagen is even less stable than we previously thought. In fact, individual collagen triple helixes called tropocollagens may have a melting point that is several degrees below body temperature (Leikina *et al*, 2002).

Yet somehow, collagen manages to stay stable by adjusting its hydroxyproline content, being stabilized by a chaperone molecule, or assembling into aggregates.

1.3 Hypothesis and Objectives

To the best of our knowledge, few studies have attempted to understand the different conformational states involved in collagen degradation and the distribution of these states. In this study, we investigate these problems by proposing the following hypothesis and objectives:

Hypothesis

The cleavage of collagen requires localized partial unfolding of the triple helix at specific regions. If no partial unfolding occurs in those areas, collagen can not be cleaved.

Objectives

1. Analyze collagen's conformational stability below body temperature and verify collagen's instability at or near body temperature.
2. Determine the relationship between the collagen sequence variation and structural stability through analysis of collagen peptides extracted from different regions and mutant collagen peptides.
3. Propose a computational method to quantitatively measure the distribution of conformational states in solution. Evaluate this method by comparison with experimental approaches.

Chapter 2

Experimental Studies of Collagen-Like Peptides

In this section we continue the discussion of imino-poor and imino-rich collagen-like peptides and introduce several peptides that model a few forms of mutant collagen. We also introduce some methodologies for studying these peptides including an experimental approach called Circular Dichroism. In chapter 3 we will introduce a computational approach to study collagen by using molecular dynamics and how it relates to the experimental studies.

2.1 Collagen-Like Peptides

While it is generally preferred to conduct studies on full length collagen proteins, it is important to point out that the construction or extraction of such proteins are difficult, due to the large length of the collagen chain. The large number of post-translational modifications such as hydroxylation and glycosylation must also be performed for correct analysis, requiring a large number of enzymes that may or may not be easily available. As a result of these difficulties, collagen-like peptides are studied instead of full length collagen proteins. These collagen-like peptides are much shorter in length and can be used to model specific regions in the protein, such as regions near the collagenase cleavage site.

So far we have introduced two types of collagen-like peptides: the imino-rich peptide and the imino-poor peptide modeling the collagen protein cleavage site. Imino, or imino-acids, indicate the presence of either proline or hydroxyproline residues, which are thought to stabilize the triple helix structure by providing conformational constraints with bulky side chains. The imino-rich peptide, $(POG)_{10}$, is therefore thought to produce a perfect triple helical structure, while the imino-poor or T3-785 peptide, $(POG)_3ITGARGLAG(POG)_4$, models the sequence of collagen type III near the cleavage site.

The goals of our study are to study the relationship between collagen sequence variation and conformational stability near the cleavage region, and to determine why the scissile bond is always unique despite the existence of multiple potential cleavage sites. The scissile bond (Gly-Leu or Gly-Ile) that is cleaved by collagenase is at one unique position in each chain of the collagen triple helix, despite the multiple potential sites that contain the Gly-Leu or Gly-Ile sequence (Figure 2.1).

MMSFVQKGSW LLLALLHPTI ILAQQEAVEG GCSHLGQSYA DRDVWKPEPC
QICVCDSGSV LCDDIICDDQ ELDCPNPEIP FGECCAVCPQ PPTAPTRPPN
GQGPQGPKGD PGPPGIPGRN GDPGIPGQPG SPGSPGPPGI CESCPTGPQN
YSPQYDSYDV KSGVAVGGLA GYPGPAGPPG PPGPPGTSGH PGSPGSPGYQ
GPPGEPGQAG PSGPPGPPGA IGPSGPAGKD GESGRPGRPG ERGLPGPPGI
KGPAGIPGFP GMKGHRGFDG RNGEKGETGA PGLKGENGLP GENGAPGPMG
PRGAPGERGR PGLPGAAGAR GNDGARGSDG QPGPPGPPGT AGFPGSPGAK
GEVGPAGSPG SNGAPGQRGE PGPQGHAGAQ GPPGPPGING SPGGKGEMGP
AGIPGAPGLM GARGPPGPAG ANGAPGLRGG AGEPGKNGAK GEPGPRGERG
EAGIPGVPGA KGEDGKDGSP GEPGANGLPG AAGERGAPGF RGPAGPNGIP
GEKGPAGERG APGPAGPRGA AGEPGRDGVP GGPGMRGMPG SPGGPGSDGK
PGPPGSQGES GRPGPPGPSG PRGQPGVMGF PGPKGNDGAP GKNGERGGPG
GPGPQGPPGK NGETGPQGPP GPTGPGGDKG DTGPPGPQGL QGLPGTGGPP
GENGKPGEPG PKGDAGAPGA PGGKGDAGAP GERGPPGLAG APGLRGGAGP
PGPEGGKGAA GPPGPPGAAG TPGLQGMPGE RGGLGSPGPK GDKGEPGGPG
ADGVPGKDGP RGPTGPIGPP GPAGQPGDKG EGGAPGLPGI AGPRGSPGER
GETGPPGPAG FPGAPGQNGE PGGKGERGAP GEKGEGGPPG VAGPPGGSGP
AGPPGPQGVK GERGSPGGPG AAGFPGARGL PGPPGSNGNP GPPGPSGSPG
KDGPPGPAGN TGAPGSPGVS GPKGDAGQPG EKGSPGAQGP PGAPGPLGIA
GITGARGLAG PPGMPGPRGS PGPQGVKGES GKPGANGLSG ERGPPGPQGL
PGLAGTAGEP GRDGNPGSDG LPGRDGSPGG KGDRGENGSP GAPGAPGHPG
PPGPVGPAGK SGDRGESGPA GPAGAPGPAG SRGAPGPQGP RGDKGETGER
GAAGIKGHRG FPGNPGAPGS PGPAGQQGAI GSPGPAGPRG PVGPSGPPGK

DGTSGHPGPI GPPGPRGNRG ERGSEGSPGH PGQPGPPGPP GAPGPCCGGV
GAAAIAGIGG EKAGGFAPYY GDEPMDFKIN TDEIMTSLKS VNGQIESLIS
PDGSRKNPAR NCRDLKFCHP ELKSGEYWVD PNQGCKLDAI KVFCNMETGE
TCISANPLNV PRKHWWTDSS AEKKHVWFGE SMDGGFQFSY GNPELPEDVL
DVQLAFLRLL SSRASQNITY HCKNSIAYMD QASGNVKKAL KLMGSNEGEF
KAEGNSKFTY TVLEDGCTKH TGEWSKTVFE YRTRKAVRLP IVDIAPYDIG
GPDQEFGVDV GPVCFL

Figure 2.1 The sequence of collagen type III, with all potential cleavage sites (Gly-Leu, Gly-Ile) highlighted. Sequence adapted from Human Protein Reference Database, NP 000081.1. Note the database did not differentiate between proline and hydroxyproline.

In additional to the imino-rich and imino-poor collagen-like peptides, we will also conduct studies on mutant collagen peptides identified to be expressed in Ehlers-Danlos Syndrome. These peptides include the G2S and its modification G2S long mutants, which are discussed below in section 2.2. Furthermore, we will conduct studies on peptides taken from the sequence of some potential cleavage sites that are not near the real cleavage site and determine if there is any significant difference between peptides from different sites.

2.2 Ehlers-Danlos Syndrome and Mutant Collagen

The Ehlers-Danlos Syndrome (EDS) is a group of hereditary diseases characterized by joint hypermobility, skin extensibility, and tissue fragility. The disease is caused by one or more mutations in collagen, which causes defects in connective and other types of tissues. There are nine different types of Ehlers-Danlos Syndrome, categorized by biochemical and clinical variations. Currently no cure is available and the only treatments are directed at managing the symptoms.

Out of the nine types of the Ehlers-Danlos Syndrome, type IV is the most severe form. This type of EDS affects the vascular system so that blood vessels and organs are potentially ruptured, causing severe problems and possibly death. Former studies have indicated that EDS type IV can be caused by a single base mutation near the collagenase cleavage site in procollagen type III, where amino acid at position 790 is changed from a glycine to a serine (Tromp *et al*, 1989). Tromp reported that the mutant collagen type III

was unusually sensitive to proteinases because the single base mutation was near the collagenase cleavage site at amino acid position 781 and also near the trypsin sensitive site at amino acid position 789. Tromp suggested that local unfolding of the triple helix must have happened to expose the arginine residue, which leads to a collagen that is vulnerable to collagenases. Further studies by the same group identified another single base mutation in procollagen type III that leads to EDS type IV: a substitution of valine for glycine at amino acid position 793 (Tromp *et al*, 1995). In the latter study Tromp observed that the substitution of a bulkier amino acid for glycine not only made the type III procollagen less stable, but very little collagen III was found in the cell layer when compared with the control. From these results Tromp proposed that either rapid collagen degradation occurs inside the cell, or the collagen is not being secreted out of cells.

2.3 List of Collagen-Like Peptides Studied

In addition to the imino-rich (IR) and imino-poor (IP) peptides, we will also study an EDS mutant collagen-like peptide which models the glycine to serine mutant (G2S). The mutant peptide is similar to the imino-poor peptide with only one amino acid substituted. This way, the results from the mutant peptide will be directly comparable to the results from the imino-rich and imino-poor collagen-like peptides. We will start with mutant peptide 30 amino acids in length, but may increase the number of POG triplets surrounding the region of interest to make sure a stable triple helical structure can be adopted. Additionally, another imino-poor peptide (IP2) modeled after a region that is not near the cleavage site will be studied to establish why the scissile bond cleavage is unique despite the multiple potential cleavage sites. A list of the collagen-like peptides to be studied is displayed in Figure 2.2. All collagen-like peptides are supplied by Genemed Synthesis. They are synthesized by solid phase synthesis methods, purified to >95% purity, and with hydroxyproline positioned in the desired positions.

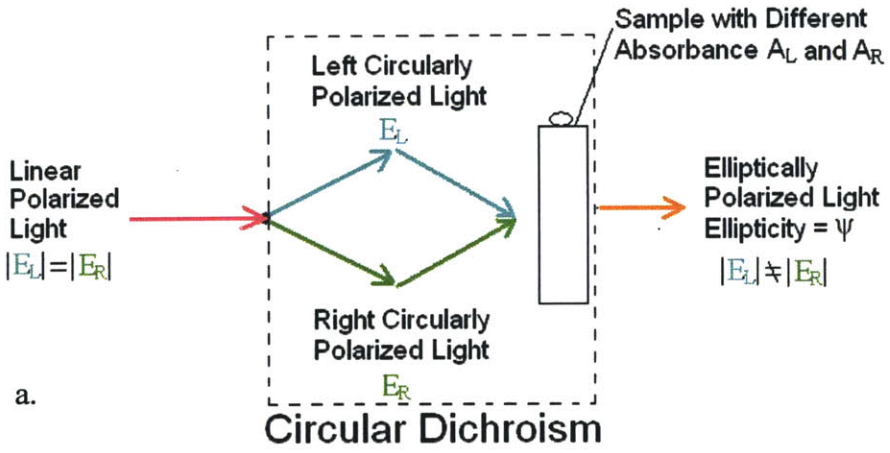
IR: $(POG)_3$-P-O-G-P-O-G-P-O-G-$(POG)_4$
IP: $(POG)_3$-I-T-G-A-R-G-L-A-G-$(POG)_4$
G2S: $(POG)_3$-I-T-G-A-R-S-L-A-G-$(POG)_4$
G2S long: $(POG)_m$-I-T-G-A-R-S-L-A-G-$(POG)_n$

IP2 non-cleavage: $(POG)_3$-A-O-G-L-R-G-G-A-G-$(POG)_n$

Figure 2.2. A list of collagen-like peptides studied. IP and IR denote imino-poor and imino-rich peptides, respectively. G2S denote the mutants of the IP peptide, where a serine substitutes a glycine. G2S long denote longer versions of the G2S peptide, with added POG triplets on the terminals (m>3, n>4). IP2 denotes another imino-poor potential cleavage site, from the non-cleavage region of collagen. All peptides besides imino-rich are taken from a region of collagen type III then surrounded by POG triplets.

2.4 Circular Dichroism Spectroscopy

The major experimental approach we use is called Circular Dichroism (CD). Circular Dichroism is a well established technique that has been continuously improved for many years. CD works by resolving linear polarized light into two circularly polarized components, namely the left- and the right-circularly polarized light (Figure 2.3b, adapted from Yang *et al*, 1986). The two opposite components (E_L and E_R) are equal in magnitude and phase, and their superposition yields a line. When this light passes through a chiroptically active sample with different absorbances A_L and A_R for the two light components, the amplitude of the stronger absorbed component will be smaller than that of the less absorbed component. Circular Dichroism is a measure of such differential absorbance between the two polarized lights, $A_L - A_R$, by the sample. Although this difference is very small, we can indirectly measure the emerging light from the sample medium, which becomes elliptically polarized. Thus an alternative measure of the CD is the ellipticity, ψ.



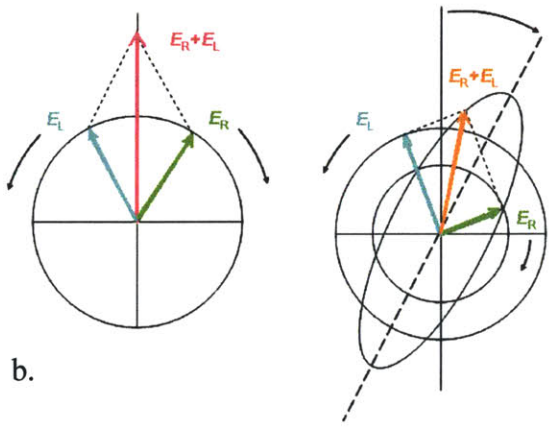


Figure 2.3. (a) The basic principle of Circular Dichroism. Linear polarized light is separated in to left- and right-circularly polarized light, and then passed through a chiroptically active sample with different absorbances for the two components. The resulting superposition of the left- and right-polarized light gives an elliptically polarized light. (b) The separation of linear polarized light into the left- and right-circularly polarized components (left), and the recombination of the left- and right-polarized components into elliptically polarized light (right). Note that all components are color coded in (a) and (b). Figure 2.3b modified from (Yang *et al,* 1986; Rupp, 2005).

The ellipticity of the sample is the ratio of the minor axis to the major axis of the resultant elliptically polarized light. It is related to the differential absorbance by (Yang *et al,* 1986)

Equation 2.1

$$\psi(\lambda) = 33(A_L - A_R)(\lambda)$$

where ψ is the ellipticity in degrees, λ is the wavelength in nm, and A_L-A_R measures the differential absorption. For many occasions it is more convenient to express the data in terms of mean residue ellipticity, $[\theta_{mr}]$:

Equation 2.2

$$[\theta_{mr}] = \frac{100 \cdot \psi(\lambda)}{c \cdot l \cdot n}$$

where $[\theta_{mr}]$ is the mean residue ellipticity in [deg x cm^2 x $decimol^{-1}$], c is the concentration of the medium in [mol / L], l is the length of the medium that the light passes in [cm], and n is the number of residues of the molecule (number of amino acids in case of a protein or polypeptide). Although decimol or mol*10^{-1} is an unconventional unit to use, it is retained because of the wealth of published data that employ this dimension. $[\theta_{mr}]$ is a convenient quantity that can be used to estimate the fractions of various conformations of a protein molecule regardless of its molecular weight or number of

residues. The analysis of the CD spectra can yield important information about the secondary structures of biological molecules, since the CD spectra for distinct molecules (peptides, proteins, or nucleic acids) are different, and each molecule is made of secondary structures whose CD spectra are unique (Figure 2.4),.

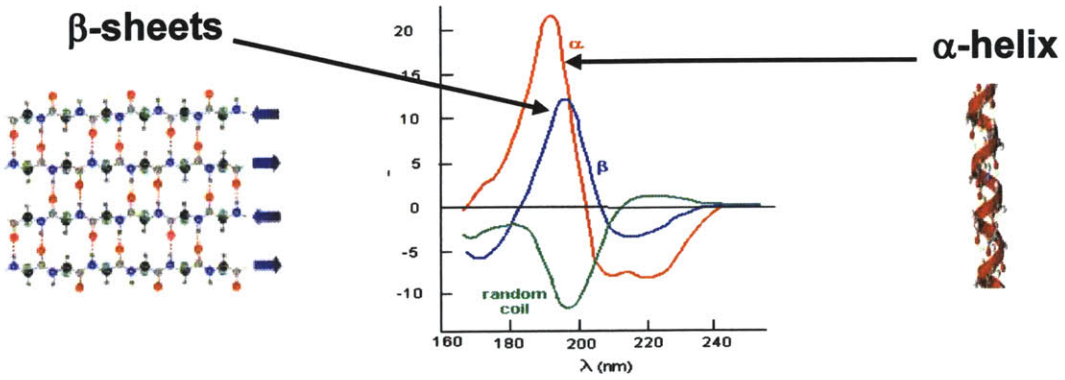


Figure 2.4. The CD spectra for the basic protein structures are different for α-helix, β-sheets, etc (figure modified from Berndt, 1996).

Two types of Circular Dichroism experiments are carried out in our studies. The first is a spectrum measurement test performed under isothermal conditions, used to determine the secondary structure compositions of the collagen and collagen-like peptides. We obtain the mean residue ellipticity from the isothermal spectrum measurements as a function of wavelength and separate the output spectrum into a linear combination of secondary structures. The second type of experiment is a melting point test carried out at the same wavelength but different temperatures. The output of the melting point experiment is the mean residue ellipticity as a function of temperature, which is used to determine the melting temperature and the thermal stability of collagen and collagen-like peptides.

2.4.1 Circular Dichroism Spectrum Measurements

In this study we use CD to determine the secondary structure compositions of collagen and collagen-like peptides. The overall CD spectrum can be approximated by the linear sum of the CD spectrum of individual structures in solution weighted by their

proportionality. Each individual in solution can be classified as either a native structure, a vulnerable structure, or a random coil structure. We define the native as being made of perfectly folded triple helix, the random coil as being completely unfolded, and the vulnerable as being partially unfolded triple helix. Figure 2.5 provides a more apparent definition of the three states. The spectrum in Figure 2.5 represents all conformation space that is possible for the imino-poor peptide, from the perfectly folded state on the very left to the completely unfolded state on the very right. Because the native state and random coil state are strictly defined as perfectly folded or completely unfolded triple helical conformation, they take up a very narrow portion of the conformation spectrum. The vulnerable state is less strictly defined and takes up a wider range of the conformation spectrum. A more quantitative definition of the vulnerable state is given in section 2.5.1.

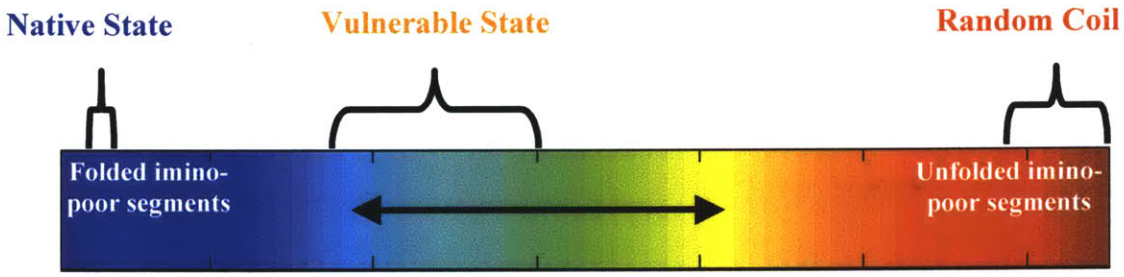


Figure 2.5. The assumed composition of native and vulnerable states of collagen, where the native state consists of only folded triple helix conformation, while the vulnerable state is a linear combination of the native and the unfolded conformations. Figure adapted from (Stultz and Edelman, 2002).

A Jasco J810 spectropolarimeter was used to measure the spectrum of collagen-like peptides from 240nm to 190nm. The solvent used to prepare the sample must be chosen carefully to make sure that it does not absorb light (or absorbs little light) in this region of interest, and therefore does not mask the protein signal. A solvent baseline CD spectrum is measured in the wavelength region of interest to remove any minor contribution of light absorption made by the solvent. The solvent used in our experiments is Phosphate buffer saline (PBS) at pH 7.2, which contains salt and water at neutral pH to simulate bodily conditions. Compressed nitrogen is used to continuously flush out the air in the sample chamber before and during the experiment because air absorbs light and interferes

with the spectrum reading at lower wavelengths. Each sample is prepared in Eppendorf tubes at 100μg/mL and transferred to a 0.1cm thick quartz cell cuvette, then incubated in the chamber for a sufficient amount of time to equilibrate with the experimental temperature before measuring. Each experiment is measured at a rate of 20nm per minute at a constant temperature and nitrogen flow controlled by a thermoelectric temperature unit. Each resultant spectrum is the average of 10 accumulations, with each accumulation being an individual spectrum reading from 240nm to 190nm.

2.4.2 Circular Dichroism Melting Point Measurements

Melting point experiments use the Jasco J810 spectropolarimeter to measure the ellipticity at a fixed wavelength in a continuously varying temperature interval. The bulk of the experimental procedures remain the same as the spectra experiments. Samples are prepared at concentrations of 100μg/mL or 1mg/mL and the experiment is carried out using 0.1cm quartz cell cuvettes. Each sample is measured at a rate of 6°C/hour (0.1°C/min) to ensure the sample equilibrates at each new temperature before a reading is taken. The temperature interval measured also changes depending on the sample and is composed of an interval between 2 to 90 degrees Celsius.

2.5 Computational Analysis of Circular Dichroism Data

In this section we discuss the methods used to decompose the experimental CD spectrum as a sum of different states. We also present a quantitative way of identifying the melting point given a melting curve. The raw CD output measures the ellipticity ψ in mdegrees as a function of wavelength or temperature. This is not the preferred unit of reporting CD results and is converted to the mean residue ellipticity [θ_{mr}] with units of [deg x cm^2 x $decimol^{-1}$], using $[\theta_{mr}] = \frac{\psi(\lambda)}{10 \cdot c \cdot l \cdot n}$. This mean residue ellipticity is the output format used throughout our study.

2.5.1 Analysis of Isothermal Spectrum

The spectrum obtained in isothermal measurements is a linear combination of the spectra of structures in solution weighted by their probabilities. If we are given the CD spectrum of each individual structure in the solution, we can use a simple and direct method of approximating the percentage of each structure in solution: the least squares method.

The least squares method minimizes the sum of the squared difference between the CD spectra curve and the optimized curve at each wavelength by optimizing the linear coefficients of the different structure in solution so that the optimized curve is as close to the measured CD spectrum as possible. This is summarized in Equation 2.3, where F is the value of the squared difference function that we are trying to minimize, Z is a vector containing the experimentally measured CD spectrum, λ is wavelength, and S_i is the individual structure spectrum. The coefficient corresponding to each S_i that we will optimize is C_i, which adds up to 1 in Equation 2.4.

Equation 2.3

$$F = \sum_{\lambda=190}^{240} (Z(\lambda) - \sum_{i=1}^{\#basis} C_i \cdot S_i(\lambda))^2$$

Equation 2.4

$$\sum_{i=1}^{\#basis} C_i = 1$$

The CD spectrum of each individual structure present in solution, S_i, can further be decomposed into the weighted sum of the native and random coil spectra. The native state is the perfect triple helix conformation, which is obtained by measuring the spectrum of the imino-rich peptide at 10 degrees Celsius, a temperature well below the melting point so the peptide is stable. The random coil spectrum is obtained by taking the average spectrum of 50 trials of thermally degraded collagen-like imino-poor peptide at 90 degrees Celsius, a temperature that is greatly above its melting point and favors completely unfolded structures. This is described by Equation 2.5. The coefficients P_{Native} and P_{coil} are obtained from MD simulation, which is discussed in Chapter 3.

Equation 2.5

$$S_i = p_{Native} \cdot Z_{Native} + p_{coil} \cdot Z_{coil}$$

Equation 2.6

$$p_{Native} + p_{coil} = 1$$

2.5.2. Analysis of Melting Point Curves

Assuming a sigmoidal 2-state transition melting curve, the melting point is defined by the point of inflection on the sigmoidal curve. The sigmoidal function is described by Equation 2.7, where a describes the magnitude between the two states, b describes how sharp the transition state falls, c is the inflection point or the melting point between the two states, and d is an offset constant. The least squares algorithm is used to find the best fit curve using Matlab. This provides a more formal method to calculate the melting point rather than by eye, which may contain some human error and bias.

Equation 2.7

$$sigmoidal = \frac{a}{1 + e^{b(t-c)}} + d$$

Chapter 3

Computational Studies of Collagen-Like Peptides

Molecular dynamics simulations are designed to study the structural, dynamical, and thermodynamic properties of biological molecules. Using molecular dynamics (MD), one models the atoms and interactions in a protein, a solvent, or both. The underlying principle used to model the motions of these atoms can be either Classical Mechanics or Quantum Mechanics. Generally classical mechanics are more widely used because they are less computationally expensive. In this study, we use an algorithm called CHARMM (Chemistry at Harvard Molecular Mechanics) to characterize collagen-like peptides' structural stability and properties.

3.1 Basic View of Molecular Dynamics

Imagine an isolated system containing a finite number of particles. If we are given the mass m_i, velocity v_i, and initial position x_i of each particle at time t_0 (Figure 3.1), we can calculate information about any particle in the system at any given time $t \geq t_0$ by applying Newton's equations to model the motions of each particle. This is shown in Equation 3.1, where F_i is the force acting on atom i at time t, m is the mass, a is the acceleration, v is the velocity, and x specifies the position of the atom.

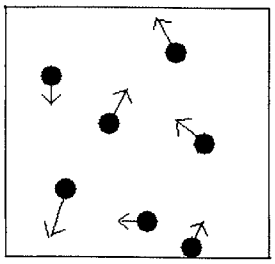


Figure 3.1. An isolated system at time $t>t_0$ can be predicted using Newtonian equations if the initial position, mass, and velocity are given.

$$F_i = m_i a_i$$

$$a_i = \frac{dv_i}{dt} = \frac{d^2 x_i}{dt^2}$$

$$x_i = a_i t^2 + v_{0,i} t + x_{0,i}$$

Equation 3.1. Newtonian equations

The underlying concept of molecular dynamics works in a similar way. The modeled system contains proteins and solvents, which are made of a large number of atoms. Given the initial positions and the forces acting on each atom, we can solve for any future state of the system by applying Newtonian equations. Although special attention must be paid to bonded and nonbonded interactions between the atoms, this should not provide too much difficulty because established approximations of such interactions already exist.

3.2 The Potential Energy Function

The potential energy V is used to give an alternative form to the Newtonian equation, where force is simply related to energy as shown in Equation 3.2. It is important to point out that the different form is actually equivalent to Newtonian equations, so the underlying concept in molecular dynamics does not change.

Equation 3.2

$$F = -\frac{dV}{dx}$$

The existence of bonded and nonbonded interactions in a protein however makes the modeling of potential energy more complicated. The potential energy is a function of bond lengths, bond angles, dihedral angles, electrostatic interactions, and van der Waals interactions, as shown in Equation 3.3.

Equation 3.3

$$V = V_{bonds} + V_{angles} + V_{dihedrals} + V_{electrostatic} + V_{vdw}$$

Accurate models currently exist to approximate these bonded and nonbonded interactions. For example, both the bond stretching potential between atoms and the bond angles potentials term can be modeled by harmonic spring equations. The potential energy is a function of atom positions. A more detailed form of the bonded and nonbonded interactions is described in Equation 3.4, where the first summation describes direct bonding interactions between all atoms, the second describes the bonding angle constraints, the third is the dihedral angle constraints, and the last terms are the van der Waals interactions and electrostatic interactions. The K's are constants obtained through experimental studies, and R is used to describe position.

$$V_{sys}(R) = \sum_{bonds} K_b(b(R) - b_{eq})^2 + \sum_{angles} K_\theta(\theta(R) - \theta_{eq})^2 + \sum_{dihedrals} \frac{K_\phi}{2}(1 + \cos(n\phi(R) - \gamma))$$
$$+ \sum_{nonbonded} \left(\frac{A_{ij}}{r_{ij}(R)^{12}} - \frac{B_{ij}}{r_{ij}(R)^6} + \frac{q_i q_j}{\varepsilon_i \varepsilon_j r_{ij}(R)}\right)$$

Equation 3.4. A more detailed form of the potential energy for the system, as a function of position. The summations account for, in order, bond length, bond angles, dihedral angles, and nonbonded (van der Waals and electrostatic) interactions.

Molecular dynamics can be used to determine the free energy of a system as a function of a predetermined reaction coordinate. This calculation of free energy is obtained through the potential of mean force, which is discussed later in this chapter.

3.3 Boundary Conditions

While the earlier applications of molecular dynamics were done on proteins in vacuum, the importance of solvent on the structure and properties of biomolecules soon became clear. But in order to simplify the complications caused by the introduction of large amount of solvent molecules, specific boundary conditions are often used in appropriate situations. In our studies, we used two types of boundary conditions, stochastic and periodic, to study the structure of collagen-like peptides. A detailed description of the two boundary conditions is listed below. Note that this section describes the traditional

stochastic and periodic boundary conditions; the setup for our slightly different approach is described later in this chapter.

3.3.1 Stochastic Boundary Condition

Stochastic boundary condition is generally applied when the interest is mostly limited to a certain region of the protein-solvent system. In such cases, one can generally reduce the computational resources that go into the regions of non-interest and focus on a localized region.

Molecular dynamics using stochastic boundary conditions is characterized by partitioning the system into several different regions (Figure 3.2). Generally the entire system is divided into a reaction zone and a reservoir region, where the reaction zone

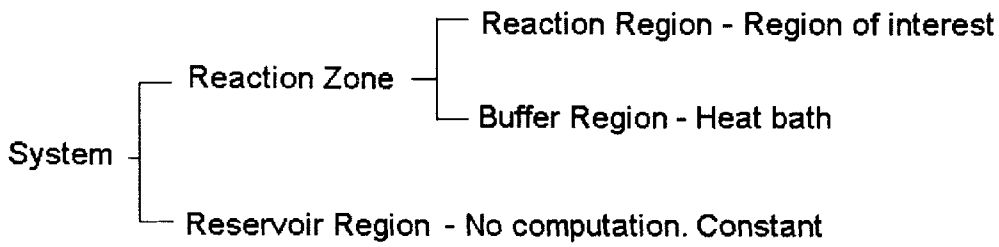


Figure 3.2. Stochastic boundary condition partitions the system into the several regions, based on its function and the spatial relationship to the region of interest.

contains the region of interest and its immediate surroundings, and the reservoir region contains the rest of the system, where no relevant reaction is assumed to take place. The reaction zone is further divided into two parts, the reaction region and the buffer region. The reaction region is the region of interest and is treated by full molecular dynamics simulations. The buffer region acts as a heat bath by applying stochastic forces to the thermal fluctuations that occur in the reaction region, and keeps the temperature inside the reaction region constant (Brooks *et al,* 1988).

Generally the partition for each system is different, but rules exist to generate such partitions. For example, one usually first defines the geometric center of the region of interest, and partitions the system into spherical (or related) regions centered at this point. This separates the system into the reaction zone and the reservoir region. The reaction zone is then further separated into the reaction region and the buffer region. The boundary between the reaction region and the buffer region is dynamic since solvent molecules are allowed to move across this boundary freely, so that the atoms in both regions are updated continuously during molecular dynamics.

The advantage of the stochastic boundary condition is that it significantly reduces the amount of computation needed to model the region of interest. Atoms in the reaction region are treated by standard molecular dynamics, where each atom's motions are governed by Newtonian equations. Atoms in the buffer region go under Langevin dynamics, an approximation for calculating the effects of the neglected degrees of freedom. Atoms in the reservoir region are fixed. The reduction in computation however also leads to the limitations of the stochastic boundary condition. Since this method is limited to a local region only, it neglects the effects from the rest of the system. Long range interactions between the atoms of the reaction region and those outside of the reaction zone, as well as the many solvent-solvent interactions outside the reaction region, are ignored, both of which could affect the behavior in the regions of interest.

3.3.2 Periodic Boundary Condition

Periodic boundary condition is another type of boundary condition that can be applied to molecular dynamics to significantly ease the complications in the study of a solvated system. In periodic boundary conditions, the system of interest (protein plus solvent molecules) is placed into a central cell and then surrounded by periodic images, where each image is a replica of the central cell (Figure 3.3). The particles in the image cells undergo the same motion as those in the central cell through continuous updating, but no dynamic simulation is carried out in the image cells. Furthermore, the simulation carried out in the central cell is in the force field or the presence of the image cells.

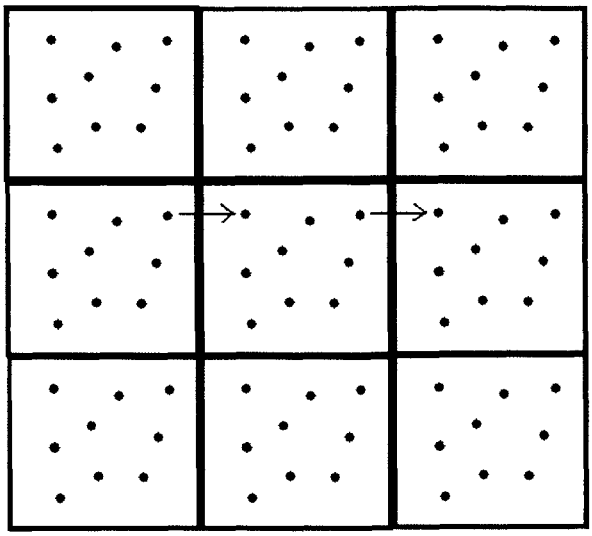


Figure 3.3. 2D periodic boundary conditions, where the central cell is surrounded by its eight nearest neighbor images. The blue arrows describe the egress of one atom from the upper right corner of the central cell, and its reentrance in the upper left corner, thus enforcing periodicity.

The use of periodic boundary conditions provides one way to solve the problem that arises at the boundary of our system. If nothing is done and the boundary simply terminates, the solvent molecules that are close to the boundary will diffuse away from the proteins and decrease the total number of molecules in the system. Periodic boundary is currently the most widely used approach to solve these problems. In periodic boundary conditions, the atoms inside the central cell can interact with those of the neighboring images, and the periodicity property (such as right out, left in, shown in Figure 3.3) itself guarantees the conservation of all molecules in the system. Furthermore, if bonded interactions are established between the atoms at the boundaries of the central cell and those of neighboring images, a system of unusual large size and regular components, such as a polymer or crystal, could be modeled.

3.4 Potential of Mean Force and Umbrella Sampling

In the physical world, reactions usually occur over long time scales that are inaccessible to standard molecular dynamic simulations. Generally, real reactions occur on scales of a microsecond or more, while MD simulations typically occur only in nanoseconds.

Therefore although modeling such processes is possible, simulations would take a huge amount of time. In such cases, we use an indirect method called umbrella sampling to exploit the path independence property of state variables and calculate the key characteristics of the process, such as the free energies differences between structures.

Assume a physical process exists that degrades a protein X from its folded state A to an unfolded state B. Instead of a fast reacting process, this degradation happens slowly and progresses through a series of intermediate states along the way. If we define a reaction coordinate ξ to be a set of coordinates which carry the system from the initial state to the final state, we can use ξ to quantitatively describe each of these intermediate states. There are many such reaction coordinates, such as the width of the molecule, a measure of its shape, etc. Overall, the process can be thought of as consisting of n intermediate stages, each used to sample a unique portion of the predetermined reaction coordinate.

To model such a process directly using molecular dynamics may incur a major obstacle: if the energy barrier for one or more of the intermediate steps is large, the time scale for those intermediate steps would still be too large to be computed by MD; on the other hand, if the energy barrier is too small, the resulting intermediate will be extremely unstable, and therefore forcing a MD with extremely small time steps. In such cases, it is essential to introduce a bias potential energy V_i, to the original physical potential energy V_o, and form a hybrid potential energy V_H at each step of the process (Equation 3.5). This process is called Umbrella Sampling. The resulting hybrid potential may or may not make sense in the physical world. However since the properties we are investigating (such as energy) are state variables, it does not matter if we choose an unphysical path by adding the bias potential and then return to the physical world at the end by getting rid of the bias component.

Equation 3.5

$$V_H = V_o + V_i$$

The bias potential function V_i that we add to the original potential V_o can be thought of as a penalty term that can be used to force the system to stay at a desired intermediate state. The form of the bias potential we use is described in Equation 3.6, where A is a force constant, ξ_i is the reaction coordinate measure of an individual structure at the

intermediate state, and ξ is the desired reaction coordinate measure of that intermediate state. In this way the structures with different reaction coordinates from that of the desired measure will receive higher energy, making them less probable than those that are closer to the desired measure of the intermediate state.

Equation 3.6

$$V_i = A(\xi_i - \xi)^2$$

Given V_o and V_i, V_H can be found using Equation 3.5, and so the biased probability density of a particular structure occurring at a certain intermediate state, $P_i(\xi)$, can be found using Equation 3.7, where $\beta=1/(K_BT)$, and Z_H is the configurational partition function as described by Equation 3.8 (Brooks *et al,* 1988). Note that the bias potential and the probability are inversely related to each other.

Equation 3.7

$$P_i(\xi) = \frac{e^{-\beta V_H(\xi)}}{Z_H}$$

Equation 3.8

$$Z_H = \int e^{-\beta V_H(\xi)} d\xi$$

Similarly, the unbiased probability density of a particular structure occurring at a certain intermediate state, $P(\xi)$, can be described using Equation 3.9. However, the configurational partition function Z_o can not be simply determined; therefore we need to relate the unbiased probability density to the biased probability density.

Equation 3.9

$$P(\xi) = \frac{e^{-\beta V_o(\xi)}}{Z_o}$$

Equation 3.10

$$Z_o = \int e^{-\beta V_o(\xi)} d\xi$$

Starting from Equation 3.9 and working backwards. From Equation 3.5, we have,

$$P(\xi) = \frac{e^{\beta(V_i(\xi) - V_H(\xi))}}{Z_o}$$

Multiply the top and bottom by the right and left side of Equation 3.8, we get

$$P(\xi) = \frac{e^{\beta(V_i(\xi) - V_H(\xi))} \int e^{-\beta V_H(\xi)} d\xi}{Z_o Z_H}$$

Substitute Equation 3.5 for the V_H inside the integral, and Equation 3.10 for Z_o,

$$P(\xi) = \frac{e^{\beta(V_i(\xi)-V_H(\xi))}\int e^{-\beta(V_i(\xi)+V_o(\xi))}d\xi}{Z_H\int e^{-\beta V_o(\xi)}d\xi}$$

rearrange,

$$P(\xi) = (e^{\beta V_i(\xi)})\cdot(\frac{\int e^{-\beta V_i(\xi)}e^{-\beta V_0(\xi)}d\xi}{\int e^{-\beta V_o(\xi)}d\xi})\cdot(\frac{e^{-\beta V_H(\xi)}}{\int e^{-\beta V_H(\xi)}d\xi})$$

Finally simplify using Equation 3.7,

Equation 3.11 $$P(\xi) = (e^{\beta V_i(\xi)})\cdot < e^{-\beta V_i(\xi)} > \cdot P_i(\xi)$$

We represented the unbiased probability density P(ξ) as a function of the biased probability density $P_i(\xi)$, where the quantity in <> is the ensemble average described by Equation 3.12.

Equation 3.12 $$< e^{-\beta V_i(\xi)} >= \frac{\int e^{-\beta V_i(\xi)}e^{-\beta V_0(\xi)}d\xi}{\int e^{-\beta V_o(\xi)}d\xi}$$

Once the actual or unbiased probability density is known, the potential of mean force can be easily calculated using Equation 3.13. The potential of mean force, W(ξ), is defined as the free energy as a function of reaction coordinate.

Equation 3.13 $$W(\xi) = -K_B T\cdot \ln P(\xi)$$

To express the potential of mean force in terms of the biased probability density, we substitute Equation 3.11 for P(ξ):

$$W(\xi) = -K_B T\cdot \ln[(e^{\beta V_i(\xi)})\cdot < e^{-\beta V_i(\xi)} > \cdot P_i(\xi)]$$

Rearrange and simplify,

Equation 3.14 $$W(\xi) = -K_B T\cdot \ln P_i(\xi) - V_i(\xi) + C_i$$

where Ci is a constant specific for each intermediate state,

Equation 3.15

$$Ci = -K_B T \cdot \ln < e^{-\beta V_i(\xi)} >$$

From Equation 3.14, we can evaluate $W(\xi)$-C_i since $P_i(\xi)$ is obtained by molecular simulation, and $V_i(\xi)$ is known. A histogram could then be constructed at each intermediate window as a function of the reaction coordinate. Since the actual process involves a number of intermediate states, each giving a specific potential of mean force function added to a certain constant, an algorithm must be developed to assemble the potential of mean force based on the slope changes. This will be discussed later.

3.5 Molecular Dynamics Simulation for Collagen-Like Peptides

So far in this chapter we have discussed the general concepts in molecular dynamics that were used in our study without going into much detail about the specific procedures used. This section will discuss a more detailed view of the molecular dynamics setups that were used in our study for the collagen-like peptides. An even more detailed explanation of the molecular dynamics commands used and the built in functions of CHARMM can be found in the Appendix section.

The first step in preparing for molecular dynamics simulation involves the loading of the collagen-like peptide 3-D structure. The X-ray crystallographic structure provided by the RCSB Protein Data Bank (PDB 1BVK) were used as the initial structure and coordinates of the collagen-like peptides. Water molecules were then added to the system by using a large equilibrated water cube containing 7177 TIP3 water molecules, removing water molecules that overlapped with protein, and then equilibrated while holding the protein fixed. The process of adding and equilibrating the water molecule was repeated 3 times. Next, boundary conditions were setup to simulate the protein-solvent system and reduce the amount of computation. Both stochastic and periodic boundary conditions were used in this study.

The use of stochastic boundary condition on collagen-like peptides was described (Stultz, 2002). Here we use a slightly different stochastic-like boundary condition. A water sphere of 30A radius was created out of the initial protein-water system by deleting all water molecules outside the 30A radius. The final system includes the entire collagen-like peptide sequence and >3500 water molecules, as shown in Figure 3.4. The reaction region was set to be a sphere with a radius of 30A around the center of mass of the peptide, and the remaining was defined to be the buffer region. The buffer region was fixed while the reaction region underwent full molecular dynamics. A heat bath of 300K was coupled to the reaction region.

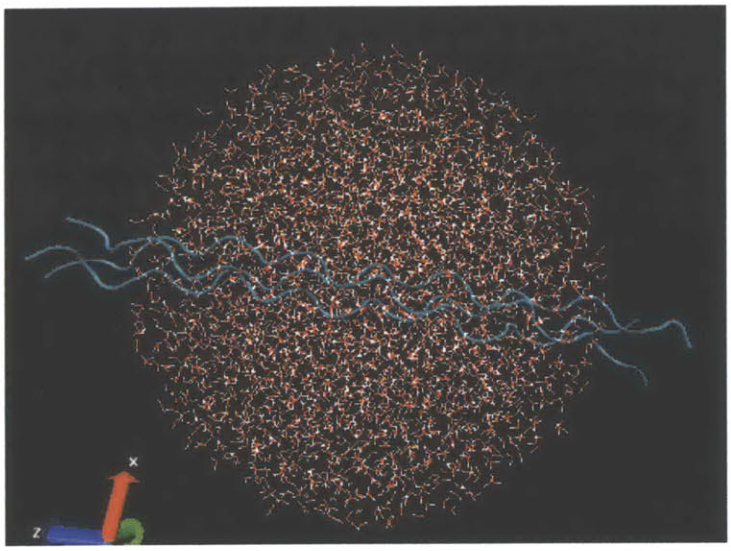


Figure 3.4. Stochastic-like boundary condition setup. The system contains the collagen-like peptide and a water sphere of 30A radius. A total of more than 3500 TIP3 water molecules were included in the system. The reaction region was defined to be anything within 30A of the center of mass of the protein, and the buffer region was defined to be anything outside. Figure generated with VMD.

The periodic boundary condition uses a rectangular box of 42A x 42A x 99A, which contains the equilibrated protein-solvent system. A total of 26 image boxes were then generated around the central cell to simulate the three dimensional environment and fulfill periodicity. A heat bath of 300K was used.

After setting up the appropriate boundary condition, we are ready to calculate the PMF of the system using Umbrella Sampling and find out the most stable (lowest free energy) structure. The unique reaction coordinate used in our studies was the radius of gyration

(RG), defined in Equation 3.16. The radius of gyration can be thought of as a measure of the average width of the peptide structure weighted by the ratio of corresponding atomic mass against the total peptide mass. Thus, the more unfolded the structure is, the higher the radius of gyration.

Equation 3.16

$$RG = \sqrt{\frac{\sum_{i=1}^{N} m_i (R_i - R_0)^2}{\sum_{i=1}^{N} m_i}}$$

Starting from the radius of gyration of the initial conformation, we run molecular dynamics using Umbrella Sampling to investigate the distribution of thermodynamic structures of the peptide in ten continuous plus and minus radius of gyration windows, each ±0.1A away from the previous window. For example, if the radius of gyration of the initial structure is 18.5A, the windows between 17.5A and 19.5A were used. A bias potential is applied to each structure to force the peptide to adopt a desired radius of gyration. At each window of RG, 50 picoseconds of molecular dynamics simulation was run. The first 20 picoseconds were treated as the time to reach equilibrium and discarded. The last 30 picoseconds (or 30000 steps, with 0.001 picosecond per step) were saved with each step containing the coordinates of the structure at that step.

A total of 3000 structures were obtained in each window and the radius of gyration for each one of these structures was calculated. From there, the potential of mean force or the free energy profile were calculated by two algorithms, PMF patching and weighted histogram analysis method (WHAM). The details for these two algorithms are described in Chapter 4. Additionally, the average structure of the collagen-like peptide at each radius of gyration window is also obtained by molecular dynamics, which can be used to hypothesize the mechanism for the unfolding process, or determine the relationship between sequence variations and structural stability by comparing energies.

Chapter 4

Constructing and Processing the PMF

The molecular dynamics simulations using Umbrella Sampling generates individual PMFs (potential of mean force) for each simulation window, where each simulation is responsible for sampling one specific region along the reaction coordinate. Individually, these PMFs only span a narrow range of the reaction coordinate. An algorithm must therefore be devised to construct the global PMF from the individual windows. In our research, we explore two different algorithms for constructing the PMF: PMF patching and the Weighted Histogram Analysis Method (WHAM).

4.1 PMF Patching

The PMF patching program is the first method we explored to piece together the small individual PMF windows from Umbrella Sampling. The program is written in Matlab and is divided into four functions: calc_pmf_general.m, calc_pmf_window.m, calc_pmf_slope.m, and patch_pmf.m. A detailed explanation of the four functions is included in the Appendix section. Briefly, the program takes the temperature in Kelvin and the number of bins used for each simulation window as inputs, and outputs the constructed PMF. Each simulation window is divided into 10 equal bin intervals according to the radius of gyration (RG), and the individual structures from each

simulation are sorted into these bins to generate a histogram for each simulation window. The individual PMFs for each simulation window is calculated from the histograms using the hybrid probabilities and biasing potentials at each discrete bin according to Chapter 3, and the slope is calculated for each PMF. The patching or the process of piecing together the individual PMF windows starts from the leftmost simulation window and progresses towards increasing radius of gyration. Given two windows, the overlap region between the two windows is first identified according to RG and the discrete PMF slopes are then compared to identify the point where the right window is patched to the left (See Appendix for figure illustrations). PMF slopes are used in identifying the patching point instead of actual PMF values because a different constant is added to each individual PMF window as described by Equation 3.14. The patching process repeats until all simulation windows have been pieced together. The final resulting PMF is smoothed using a moving window method with a span of 3, and then normalized by setting its minimum to zero.

While the PMF patching is a very intuitive way of piecing together the individual PMF windows, a number of problems exist. First, the range of RG within each simulation is different, causing the amount of overlap between neighboring windows to be unpredictable: if the overlap is too small, not enough comparison is made between the two windows to determine the correct patching point; if the overlap is too large, new window spacing should be obtained to avoid information loss during each patching, since at each overlap we accept the values from either the former window or the next window and discard the other; still, if no overlap exists between the two neighboring windows, no patching can be made unless additional simulations are done. Second, even if the amount of overlap is appropriate, dividing each simulation into 10 equal bins almost always guarantees that the bins will fall on different RG values in different simulations, causing a mismatch in RG when neighboring windows are compared. Such mismatch can be alleviated by interpolation or extrapolation, however the results of such may be different than if additional simulations were done around that point. Third, although the algorithm guarantees that the patching point is at the point where the slope is *most* similar between two neighboring windows, it does not guarantee the slopes are *truly* similar, since all

slopes are estimated with finite points. In such cases, although patching can be made, a discontinuous point on the PMF is often resulted, so that an additional simulation between the two windows is usually required to get rid of the discontinuous point. Generally speaking, although the PMF patching process is fast and intuitive, many problems arise at the overlap between windows. As a result, we explored an algorithm developed by Kumar *et al* in 1992, the Weighted Histogram Analysis Method.

4.2 Weighted Histogram Analysis Method (WHAM)

The weighted histogram analysis method (WHAM) is a method of constructing PMF that was developed about a decade ago (Kumar *et al,* 1992). The WHAM holds a number of advantages over the previous PMF patching algorithm. The general form of WHAM is carried out using various sets of coupling parameters for the constraint potentials to enhance conformational sampling. WHAM can be easily extended to multiple reaction coordinates, biasing functions, and even temperatures, by the using of appropriate coupling parameters:

Equation 4.1

$$H_{\lambda}(x) = H_0(x) + \sum_{i=1}^{L} \lambda_i V_i(x) = \sum_{i=0}^{L} \lambda_i V_i(x)$$

where H describe the Hamiltonian, H_λ is the modified potential, V_i is the biasing potential, λ's are the coupling parameters that are used to weight or pick out appropriate biasing potentials, and x is the molecular coordinate.

As mentioned before, the Umbrella Sampling simulations yield individual PMFs, each of which contains a unique offset C as described by Equation 3.14. The PMF patching algorithm constructs a PMF by using the principle that a constant offset does not alter the slope of the PMF, but this may be inaccurate depending on the overlap between adjacent distributions. The WHAM method on the other hand, optimizes the links between simulations by using multiple overlaps of probability distributions to obtain better estimates of free energy differences. The general form of the WHAM method originates from Kumar *et al,* 1992, but involves the use of coupling parameters and is confusing in

notation. An equivalent but more comprehensible and specific version of WHAM is taken from (Roux, 1995), summarized by Equations 4.2 and 4.3:

Equation 4.2

$$< p(\xi) >= \frac{\sum_{i=1}^{R} n_i < p(\xi) >_i}{\sum_{j=1}^{R} n_j e^{-(V_j(\xi)-F_j)\beta}}$$

Equation 4.3

$$e^{-F_i\beta} = \int e^{-V_i(\xi)\beta} < p(\xi) > d\xi$$

where R is the number simulations in Umbrella Sampling, $\beta=1/K_BT$, n_i is the number of total structures in the i^{th} simulation, and $<p(\xi)>$ is the ensemble average of the unbiased probability distribution, defined as:

Equation 4.4

$$< p(\xi) >= \frac{\int p(\xi) e^{-\beta V_0(\xi)} d\xi}{\int e^{-\beta V_0(\xi)} d\xi}$$

And F is the free energy constants defined by 5 (Roux, 1995):

Equation 4.5

$$e^{-F_i\beta} =< e^{-V_i(\xi)\beta} >$$

Equations 4.2 and 4.3 are solved iteratively so that F_i can be determined. Generally one can start with an arbitrary set of values for F_i, such as $F_i=0$ for all i, and apply the WHAM equations until convergence is reached. In most cases convergence is reported to be fast.

Generally speaking WHAM uses all information from all simulations of Umbrella Sampling (no information is discarded) and avoids the overlap problem, by constructing an optimal estimate of unbiased distribution function as a weighted sum over the data extracted from all simulations and determining the functional form of the weight factors that minimizes the statistical error. Furthermore, the WHAM equations can also be used to generate PMF and free energies as a function of different reaction coordinates and/or temperature. This is a very useful tool as simulations can now be (Kumar *et al*, 1992) carried out at a range of temperatures to improve conformational sampling and the results extrapolated or interpolated to the desired temperature.

There are some disadvantages to WHAM. First, unlike the PMF patching algorithm, it is not an intuitive algorithm to understand and implement. Second, while the abundance of parameters in WHAM makes it a powerful algorithm with a lot of flexibility, it also makes it difficult to determine the exact parameters needed to construct the ideal PMF. For example, when different bin width is used, the resulting PMF constructed from WHAM will be slightly different. Yet it is hard to decide which should be the correct bin width to use (see Appendix). In another example, while WHAM usually converges fast, it is hard to prove whether different starting F_i values will still result in the same PMFs. Despite these disadvantages however, WHAM is still a powerful and reliable way to construct the PMF, and is therefore implemented in our study.

The WHAM program is implemented in Matlab using three functions, wham_general.m, divide_rg.m, and wham_iteration.m. A more detailed description is provided for each of these functions in the Appendix section. The WHAM program is iterated until both equations converge. From here, there are two methods of constructing the final PMF. The first method uses the last (converged) set of the free energy constants F from the second equation of the WHAM equation pair. Recall Equation 3.15:

Equation 3.15
$$Ci = -K_B T \cdot \ln < e^{-\beta V_i(\xi)} >$$

Rearrange, we get

Equation 4.6
$$e^{-Ci\beta} = < e^{-\beta V_i(\xi)} >$$

Comparing Equation 4.6 with Equation 4.5, we realize that

Equation 4.7
$$F_i = C_i$$

So that we have demonstrated that the free energy constants F are the same as the offset constants in the Umbrella Sampling equation. The PMF is then related to the free energy constants and the biased probability distribution p_i by:

Equation 4.8
$$W(\xi) = -K_B T \cdot \ln p_i(\xi) - V_i(\xi) + F_i$$

The free energy constants F_i are defined however only at points $\xi=\xi_i$, or where each Umbrella Sampling simulation is centered at. If we calculate the PMF $W(\xi)$ at each point $\xi=\xi_i$, we see that from Equation 3.6, the second term in Equation 4.8 drops out:

Equation 3.6 $$V_i = A(\xi_i - \xi)^2$$

Equation 4.9 $$V_i(\xi)=A(\xi-\xi_i)^2=0$$

Since $-K_BT=-1/\beta\approx-0.6$ kcal/mol at 300K, and $0 \le p_i(\xi) \le 1$, where $\xi=\xi_i$ so that $p_i(\xi)$ is usually far away from 0, the first term $-K_BT \cdot \ln p_i(\xi)$ is also usually small. Occasionally however, $p_i(\xi)$ will be further away from 1 so that the first term $-K_BT \cdot \ln p_i(\xi)$ do contribute somewhat to the final PMF. Equation 4.8 therefore becomes:

Equation 4.10 $$W(\xi) = -K_BT \cdot \ln p_i(\xi) + F_i$$

The second method of finding the PMF uses the first equation of WHAM (Equation 4.2) to find the unbiased probability distribution $p(\xi)$, and then constructs the PMF using Equation 3.13:

Equation 3.13 $$W(\xi) = -K_BT \cdot \ln p(\xi)$$

Both methods yield PMF with very similar shape and scale. However, the sizes of the two PMF vectors are dramatically different. The PMF generated using Equation 4.8 contains the same number of points as the free energy constant F_i, where i spans from 1 to the total number of simulations. The PMF generated with the unbiased probability distribution however, contains the same number of points as the number of bins used in the overall reaction coordinate space, which is almost always much larger than the number of simulations. Therefore, the former PMF is usually smoother than the latter since it uses less number of points (the former is a subset of the latter). For this reason, most of the PMFs presented in literature are constructed using Equation 4.8.

4.3 Connecting MD to CD: Predicting CD Spectra from PMF

Once a reaction coordinate is chosen, the generated PMF can give us a lot of information including the radius of gyration of the most stable structures in solution and their relative stability. We can then analyze these structures from the PMF, predict their CD spectra, and then generate the global CD spectra for the sample.

Imagine that the constructed PMF is pictured as in top scheme of Figure 4.1A. This PMF contains two clear minima, each with energy E1 and E2 that is similar to the other. Because the PMF is a plot of free energy as a function of some reaction coordinate, each local minimum indicates a structure that is stable. Since the CD spectrum is a linear average of all structures in the solution and each minimum energy structure makes up a significant percentage of the presenting structures in solution, the overall CD spectrum can be approximated by the weighted sum of the CD spectra of the minimum structures. In this case, two minima are described by the PMF, and their energies are E1 and E2, respectively. Given the free energy of the two structures, we can then find the relative percentage of the two structures in solution according to Boltzman distribution, that is, $p_1 = e^{-\beta E_1}/Z$ and $p_2 = e^{-\beta E_2}/Z$. The predicted overall CD spectrum can then be approximated by $CD_{pred} = p_1 CD_1 + p_2 CD_2$, where CD_1 is the CD spectrum of the first minimum, and CD_2 is the CD spectrum of the second minimum.

To get the individual CD spectrum for the two minima, we use the results from molecular dynamics simulations at the reaction coordinate values corresponding to that of the minima. We obtain the radius of gyration rg_1 and rg_2 for the two minima from the PMF and analyze the results of the simulations centered at rg_1 and rg_2 respectively. Each CD spectrum of a collagen-like peptide structure in solution at a specific rg can be decomposed into the weighted sum of two collagen basis functions, the native or perfect triple helical structure, and the random coil or completely unfolded structure as described in Chapter 2. Recall that the native structure can be described by the spectrum of the imino-rich collagen-like peptide at 10°C, since the imino-rich peptide showed great triple

helical stability at a low temperature; the coil structure can be described by the average spectra of the imino-poor peptide at 90°C, since 90°C is way above the melting temperature of the imino-poor peptide. The decomposition of the second minimum structure is pictured as an example in Figure 4.1B.

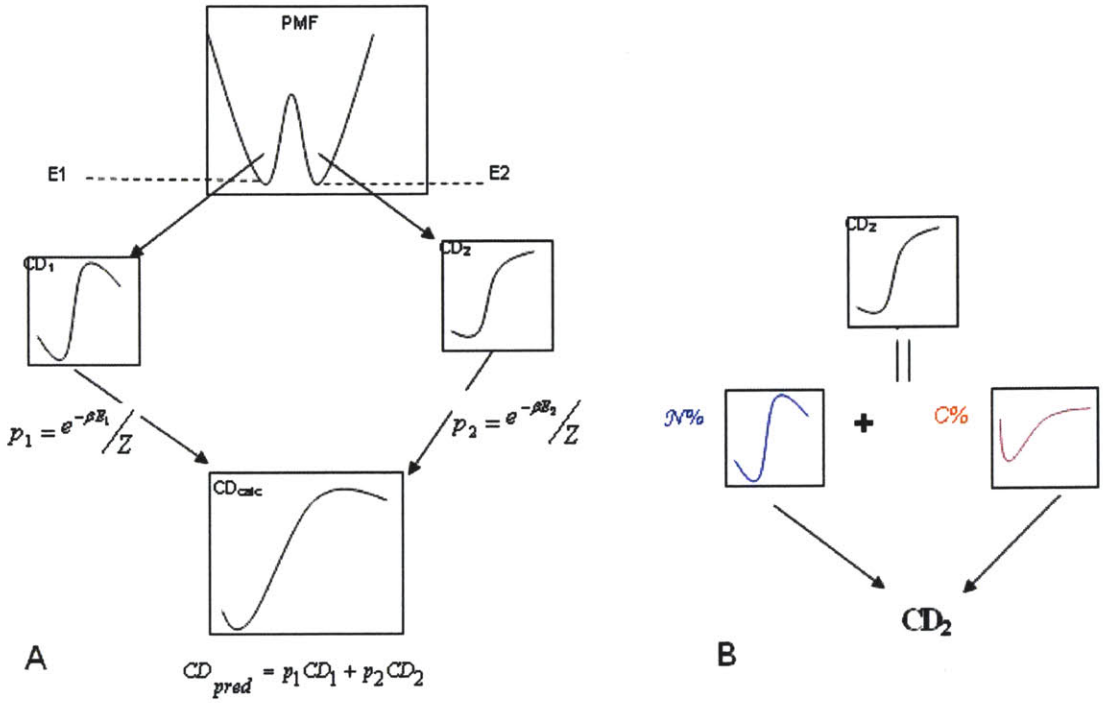


Figure 4.1. (A) Scheme for calculating the CD spectrum of a collagen-like peptide from the PMF. The PMF contains two minima with relative energies, E1 and E2. CD_1 and CD_2 denote the CD spectrum of structures corresponding to minima 1 and 2, respectively. The two CD spectra are then weighted by the corresponding Boltzmann probabilities where $Z \cong e^{-\beta E_1} + e^{-\beta E_2}$ .The calculated CD spectrum, CD_{pred}, is the sum of the weighted spectra. (B) Schematic showing an example of the determination of CD spectra for structure corresponding to minimum E2. The spectrum for state E2 is approximated as a weighted sum of the spectra corresponding to the native structure and the completely unfolded structure. 'N%' represents the percentage of residues in structure corresponding to E2 that have ϕ–φ values similar to the native state and 'C%' represents the percentage of residues that have ϕ–φ values that significantly differ from the native state. The spectrum of the unfolded structure is obtained from the spectrum of a collagen-like peptide at a high temperature well above the melting point. Structures corresponding to the different minima are obtained from the molecular simulations (Salsa *et al*, 2005, submitted).

To determine the percentage of native and coil in the individual structure at a specified radius of gyration, say rg2 for the second minimum structure, we analyze the dihedral angle contents of the structures in the simulation centered at rg2. Only the backbone dihedral angles are analyzed, namely the Φ angle formed between C-N-Cα-C and the Ψ

angle formed between N-Ca-C-N. A detailed description of how this is done is provided in the Appendix section. Briefly, any residue with Φ or Ψ angle that deviates from the average Φ or Ψ angles of the crystallographic structure by more than three standard deviations is classified as contributing to the coil structure, and those within the three standard deviations are classified as contributing to the native structure. The percentage of native and coil in the structure are then approximated as the fraction of native and coil residues to the total number of residues. Since the CD spectra for native and coil are known, the CD spectrum for the structure at this specific radius of gyration can then be approximated by the weighted summation of CD_N and CD_C.

The above procedure for predicting the overall CD spectrum of the peptide can be summarized with the following three equations:

Equation 4.11
$$CD_{pred} = \sum_{i=1}^{\min} p_i CD_i$$

Equation 4.12
$$CD_i = p_{N,i} CD_N + p_{C,i} CD_C$$

Equation 4.13
$$p_i = \frac{e^{-\beta E_i}}{Z}$$

where CD_{pred} is the predicted overall spectrum for the peptide, p_i is the percentage of the i^{th} minimum structure in the PMF calculated by the Boltzman distribution described in Equation 4.13, CD_i is the predicted CD spectrum of the i^{th} minimum structure, $p_{N,i}$ and $p_{C,i}$ are the percentage of native and coil basis functions in the i^{th} minimum structure, CD_N is the native basis structure approximated by the CD of imino-rich peptide at 10C, CD_C is the coil basis structure approximated by the CD of imino-poor peptide at 90C, E_i is the free energy of the ith minimum structure, and Z is the partition function approximated by the summation $Z = \sum_{i=1}^{\min} e^{-\beta E_i}$.

Chapter 5

Results

The results section shows the experimental data obtained using the CD spectroscopy and melting point experiments as well as the computational results from molecular dynamics simulations. The results are grouped together by peptide type, which include imino-poor peptide from cleavage region (IP) at 25C, imino-poor peptide from non-cleavage region (IP2) at 25C, and glycine to serine mutant peptide (G2S) at 25C.

5.1 Basis Spectra

From Equations 4.11-4.13 we see that the predicted CD spectrum of a sample can be expressed as a weighted sum of the spectra of distinct states in solution, where each state is a weighted sum of folded and unfolded triple helical conformations. If CD_N corresponds to the spectrum of an ideal triple-helical structure, we can estimate CD_N by using the CD spectrum of imino-rich (IR) at 10°C. A total of 30 spectra were obtained to approximate the CD spectrum of the native state; the average is shown in Figure 5.1 in blue. The CD spectrum of the random coil, CD_C, can be estimated by measuring the spectrum of T3-785 under conditions that stabilize the unfolded state. As the melting temperature of the T3-785 triple-helical structure is approximately 20°C, spectra of T3-785 at 90°C were obtained and used to approximate the random coil spectra. A total of 50

spectra were obtained and CD_C corresponds to an average over these measurements, as shown in Figure 5.1 in red.

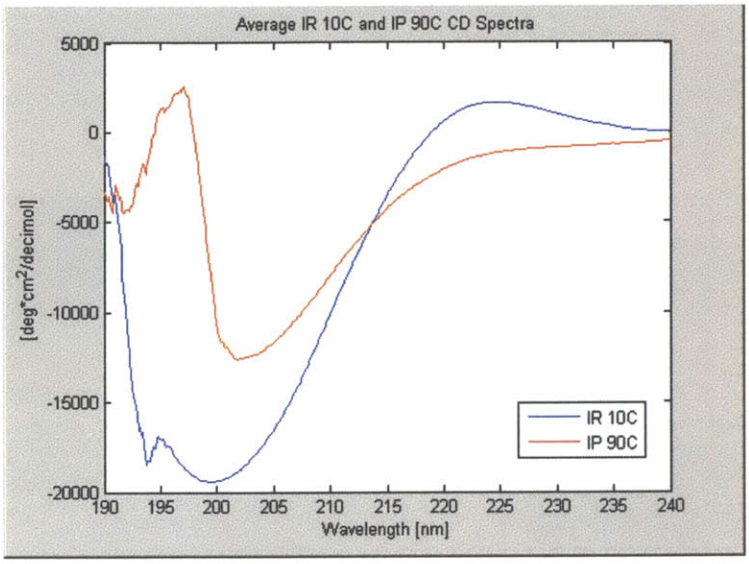


Figure 5.1. Blue: spectrum of imino-rich peptide at 10°C, average of 30 spectra. This approximates the spectrum of a perfectly folded triple helix; Red: spectrum of imino-poor peptide at 90°C, average of 50 spectra. This approximates the spectrum of an unfolded triple helix, or the random coil structure. Plots are generated using Matlab.

Note that both of the CD spectra are smooth between 200-240nm but significant noise was introduced below 200nm. This is due to the absorbance of air in lower wavelength regions. A more detailed discussion of the basis spectra as well as different types of basis spectra is included in the Discussion section.

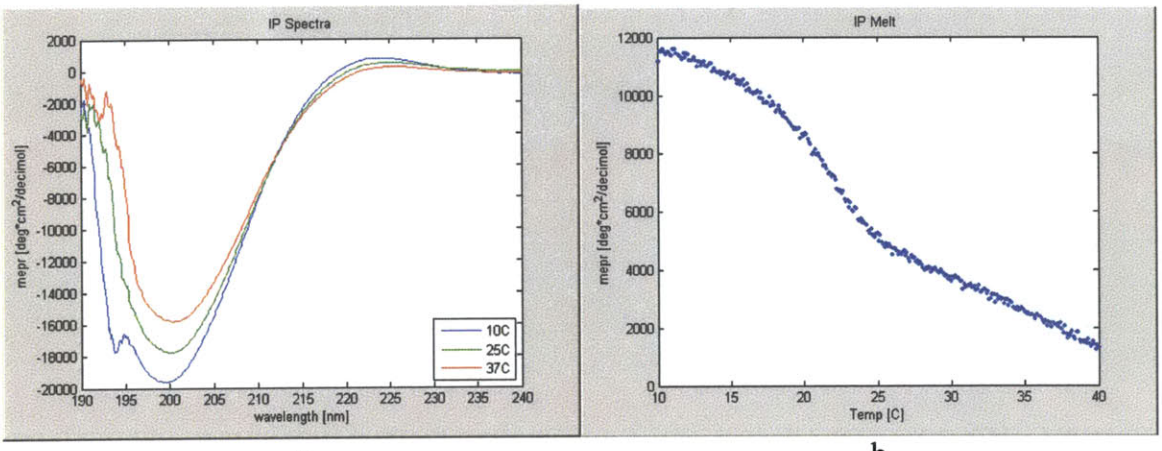
5.2 Additional Spectra and Melting Point Measurements

Previously the most popular way to determine whether a peptide folds into a triple helical collagen-like structure was to verify that the peptide CD spectrum has collagen characteristics: a small positive peak at about 225nm, followed by a crossover from positive ellipticity to negative, and finally a large negative peak at about 200nm. However, in some cases (such as the G2S peptide example below), looking at the CD spectrum alone might not be sufficient. Below we give a summary of the results from the

CD spectra and melting point experiments of IP, IP2, G2S, and G2S long peptides, and analyze whether each peptide forms a triple helical structure.

5.2.1 Imino-Poor Peptide from Cleavage Site (IP)

The CD spectra for the imino-poor peptide were measured at 10°C, 25°C, and 37°C. Figure 5.2a displays the measured average for the three temperatures. All three spectra showed a triple helical, collagen-like, characteristic, with a small positive peak near 225nm and a larger negative peak near 200nm. This triple helical content became more dominant (indicated by larger peak magnitudes) with decreasing temperature, suggesting that lower temperatures stabilize the triple helix formation. The melting curve in Figure 5.2b showed a two-state sigmoidal transition, much like typical proteins. The melting point was determined to be around 22°C using the sigmoidal fitting procedure outlined in Chapter 2, with r^2 value of 0.9935. These data suggests that that the IP peptide indeed folds into triple helixes.

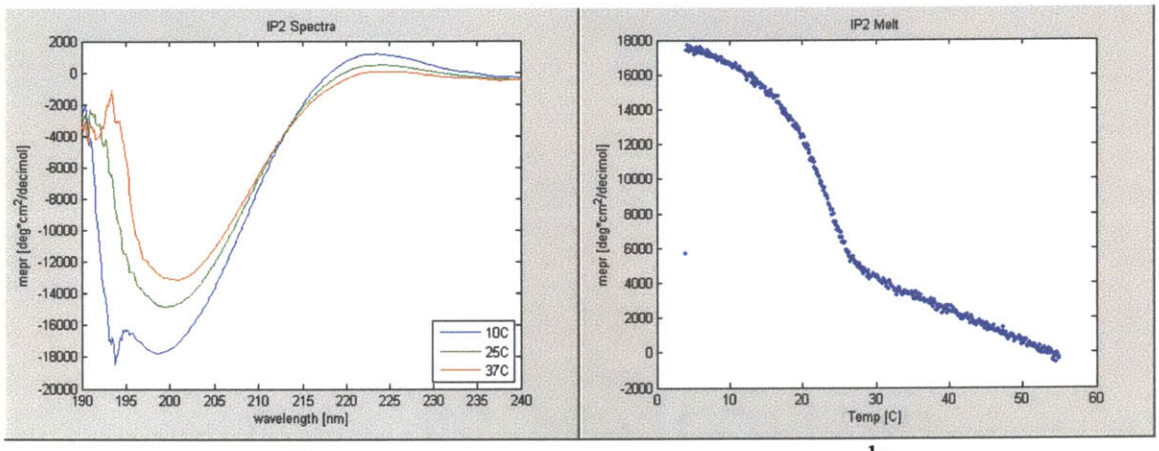


a b

Figure 5.2. (a) The spectra of IP peptide at 10°C, 25°C, 37°C. The spectra are measured with a Jasco J810 spectropolarimeter at 100ug/mL. Each spectrum is the average of 30 or 40 individual spectrum, where all of the spectra are similar in shape and value. (b) The melting curve of IP peptide, exhibiting the two-state sigmoidal transition. The melting curve is measured with the same Jasco J810 at a rate of 6C per hour.

5.2.2 Imino-Poor Peptide from Non-Cleavage Region (IP2)

The average spectra of IP2 peptide at 10°C, 25°C, and 37°C are displayed in Figure 5.3a. Similar to the IP peptide, the IP2 peptide showed a collagen-like CD spectra, and the triple helical characteristic increased with decreasing temperature. The melting curve in Figure 5.3b showed a two-state sigmoidal transition with a computed melting point of about 23C and r^2 value of 0.984. Both the CD spectra and the melting curve suggested that like the IP peptide, the IP2 peptide also forms triple helical structures in solution. Note that the melting point of IP2 is slightly higher than that of IP, indicating that IP2 is more thermally stable.



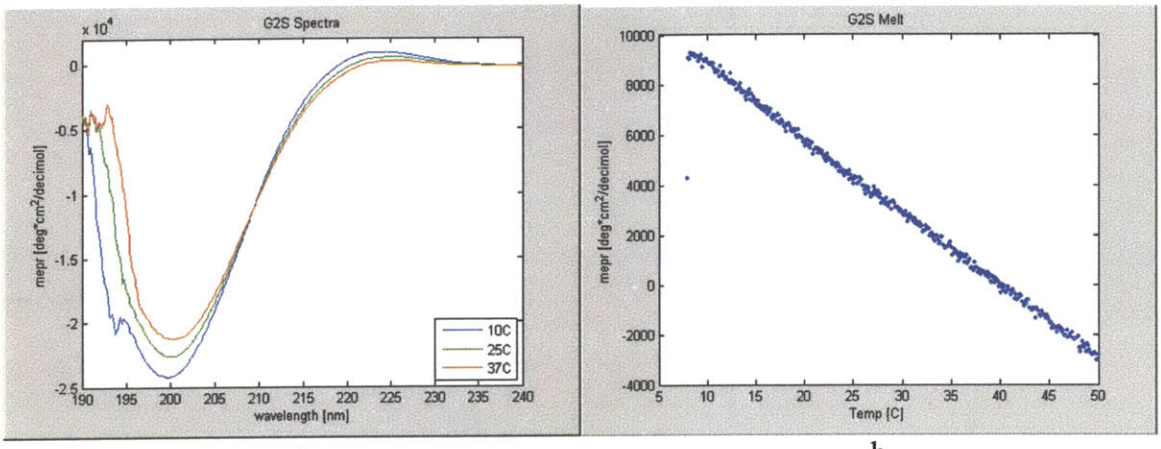
a b

Figure 5.3. (a) The spectra of IP2 peptide at 10°C, 25°C, 37°C. The spectra are measured with a Jasco J810 spectropolarimeter at 100ug/mL. Each spectrum is the average of 30 or 40 individual spectrum, where all of the spectra are similar in shape and value. (b) The melting curve of IP2 peptide, exhibiting the two-state sigmoidal transition. The melting curve is measured with the same Jasco J810 at a rate of 6°C per hour.

5.2.3 Glycine to Serine Mutant Peptide (G2S)

Upon first inspection, the CD spectra for the G2S peptide showed the same trend as those of the IP and IP2 peptides (Figure 5.4a): small positive peak around 225nm, large negative peak around 200nm, and even the relationship between the magnitude of the peaks and temperature also followed the same trend as that of IP and IP2 peptides. However, a more detailed investigation suggests that the G2S behaves very differently. First, the melting curve of the G2S peptide no longer showed a clear two-state sigmoidal transition as those of the IP and IP2. Figure 5.4b shows the melting curve of the G2S

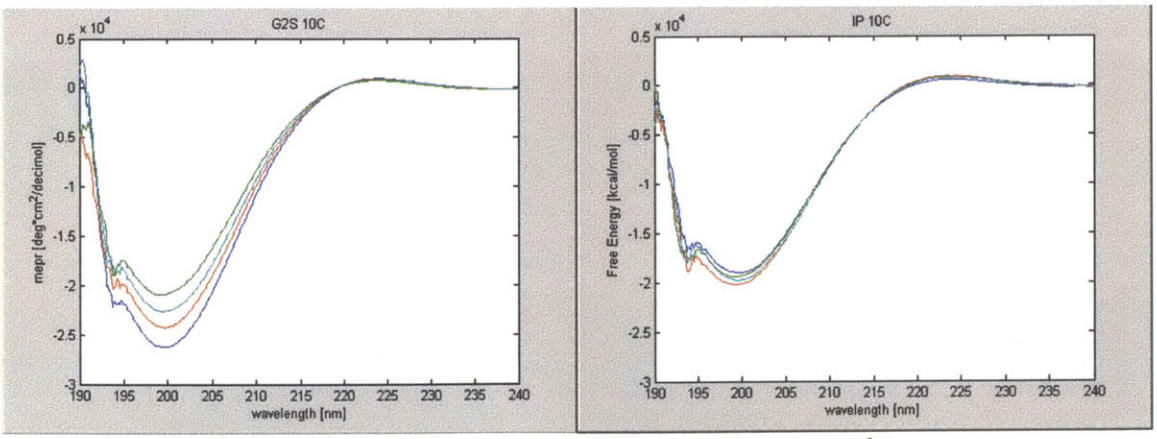
peptide to be a linear transition at 6°C per hour, indicating that the G2S peptide does not behave like a normal protein. Variation of the solution concentration, sample length, and measurement rate did not change the shape of the melting curve.



a b

Figure 5.4. (a) The spectra of G2S peptide at 10°C, 25°C, 37°C. The spectra are measured with a Jasco J810 spectropolarimeter at 100ug/mL. Each spectrum is the average of 30 or 40 individual spectrum. (b) The melting curve of G2S peptide, exhibiting a linear transition. The melting curve is measured with the same Jasco J810 at a rate of 6°C per hour. Varying the sample concentration, cuvette length, and measurement rate did not alter the shape of the melting curve.

A closer investigation reveals that the CD spectrum of the G2S peptide at all temperatures actually "fluctuates" much more severely than the CD spectrum of the other peptides. Although some fluctuation is unavoidable in all CD measurements due to uncertainties in temperature, sample concentration, measurement voltage, and other uncontrollable parameters, a normal fluctuation is usually limited and small. Figure 5.5a shows the fluctuation of four G2S CD spectrum measurements at 10°C, while Figure 5.5b shows the normal fluctuation of four IP CD spectrum measurements at 10°C. The fluctuation in the G2S peptide is noticeably higher than that in the IP peptide at 10°C, and similar findings are also made at 25°C and 37°C.



a b

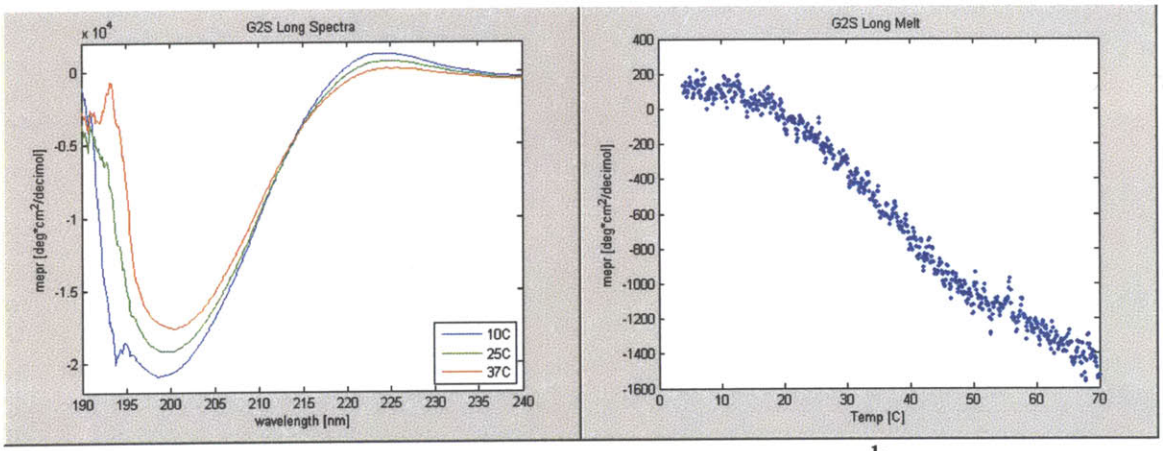
Figure 5.5. (a) Four spectra trials of the G2S peptide at 10°C. (b) Four spectra trials of the IP peptide at 10°C. The G2S peptide showed a much more inconsistent results than the IP peptide. Similar findings were made for spectra at other temperatures.

The above findings suggest that the G2S peptide does not behave like a true protein. The melting curve suggested that the peptide does not have distinguishable folded and unfolded states like a normal protein. The large fluctuation of the CD spectra at all temperatures indicates that the structures of the peptide within each sample changed unpredictably under similar experimental conditions. Therefore to study the G2S mutant, the peptide must be redesigned to ensure a triple helical folding. The redesigned form of G2S mutant is the G2S Long mutant.

5.2.4 G2S Long

The G2S Long peptide is designed out of the concern that the structures in the G2S peptide solution may not fold into triple helixes under experimental conditions. The G2S Long peptide is designed by adding P-O-G triplets to the N and C terminals of the G2S peptide until the structure shows triple helical behavior. One P-O-G triplet was first added to each of the terminals, making the structure $(POG)_4$-I-T-G-A-R-S-L-A-G-$(POG)_5$, but the measured melting curve still showed a linear transition rather than sigmoidal. Three P-O-G triplets were then added to each of the terminals, making the structure $(POG)_6$-I-T-G-A-R-S-L-A-G-$(POG)_7$ with 48 amino acids. This is known as the G2S Long peptide and its CD characteristics are described below.

The CD spectra of the G2S Long peptide at 10°C, 25°C, and 37°C were measured and displayed in Figure 5.6a. All spectra showed the characteristics of collagen CD spectrum and the magnitude of CD spectrum peaks increased with lowering of temperature. The melting curve measured at 6°C per hour still showed a higher fluctuation than the melting curve of the former peptides, but the transition more resembles a two-state sigmoidal instead of the linear transition for G2S. The computed melting point is near 37.5°C, with a r^2 value of 0.977. The melting point might sound surprisingly high compared to that of IP and IP2 peptides, but due to the longer amino acid sequence and the additional P-O-G triplets, the peptide is believed to be more stabilized and therefore the significantly higher melting point is not unreasonable. Given these we conclude that the G2S Long is a better model for the mutant form of collagen.



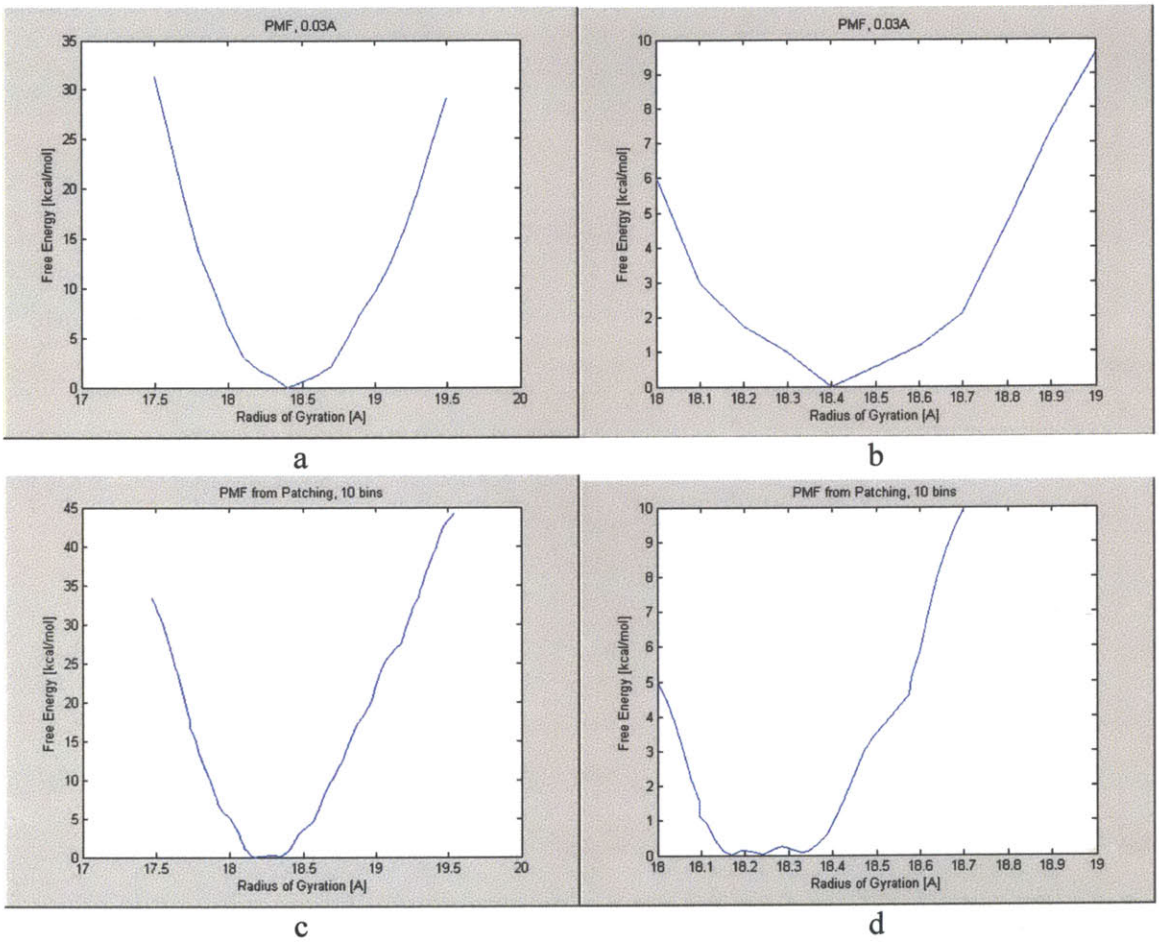
a b

Figure 5.6. (a) The spectra of G2S Long peptide at 10°C, 25°C, 37°C. The spectra are measured with a Jasco J810 spectropolarimeter at 100ug/mL. Each spectrum is the average of 30 or 40 individual spectrum, where all of the spectra are similar in shape and value. (b) The melting curve of G2S peptide, exhibiting the two-state sigmoidal transition. The melting curve is measured with the same Jasco J810 at a rate of 6°C per hour.

5.3 Imino-Poor Peptide Near Cleavage Site (IP)

Both stochastic-like and periodic boundary conditions were used for the simulation of imino-poor peptide from the cleavage region (IP, or T3-785) at 300K. The stochastic-like boundary condition is similar to the stochastic boundary condition used in earlier works of the research group, but the stochastic-like boundary condition has a fixed buffer region

and included many more times water molecules than the earlier stochastic system. The resulting potential of mean force of the stochastic-like boundary condition is generated with both the PMF patching and the weighted histogram analysis method (WHAM), and displayed in Figure 5.7. All simulations of WHAM were done using 10000 iterations unless indicated otherwise. All PMF generated were done using 21 simulation windows unless indicated otherwise.



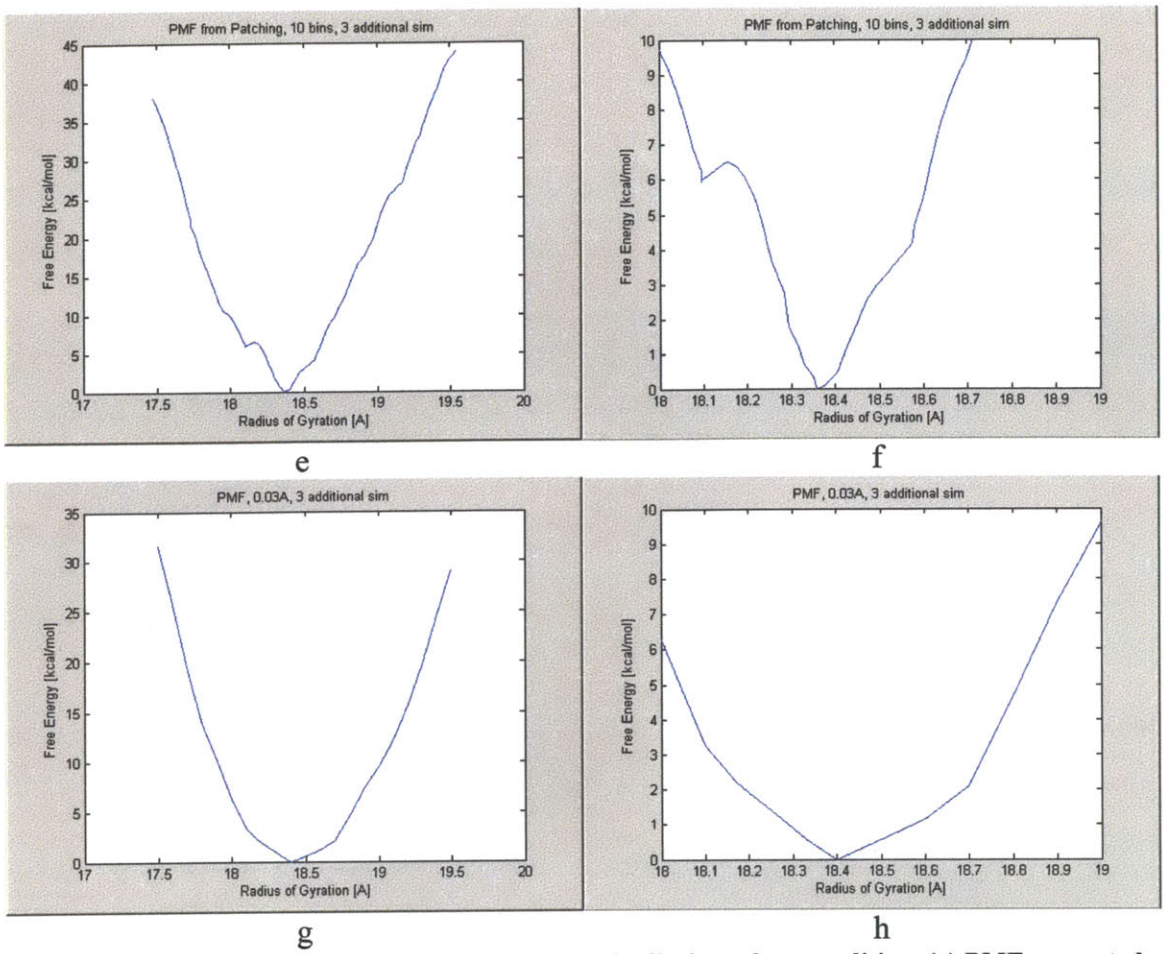


Figure 5.7. PMF of IP peptide at 300K using stochastic-like boundary condition. (a) PMF generated using WHAM (Equation 4.10), with 0.03A bin width. (b) Zoom-in of a. (c) PMF generated using PMF Patching, with 10 bins. (d) Zoom-in of e. (e) PMF generated using PMF Patching, with 10 bins, and three additional simulation windows at the three minima of e. (f) Zoom-in of g. (g) PMF generated using WHAM, with 0.03A bin width, and three additional simulation windows same as g. (h) Zoom-in of i.

All WHAM PMF plots in Figure 5.7 were generated using Equation 4.10. The PMF generated using the WHAM of 0.03A and 0.01A bin width are nearly identical, but WHAM at 0.03A may have smaller statistical errors, since it contains fewer histogram bins with less than 10 elements as shown in Table 5.1. The PMF generated using the patching algorithm indicated at first that three energy minima exist (Figure 5.7d), but another PMF generated with patching with three additional simulation windows (18.17A, 18.24A, and 18.33A, or the three minima in Figures 5.7c and 5.7d) showed only one minimum at 18.4A (Figures 5.7e and 5.7f). The WHAM algorithm with the same three additional simulation windows on the other hand, gave nearly the same result as before as

shown in Figures 5.7g and 5.7h. In this case, we use the PMF generated using WHAM with 0.03A bin width to approximate the PMF of the sample.

Table 1. A statistical view of the histograms used in WHAM. Each row summarizes the histogram i resulted from Umbrella Sampling simulation window i. Columns in order from left to right represent: the total number of nonzero bins in the histograms in simulation i; the total number of nonzero bins with less than 10 elements; the minimum number of elements in all nonzero bins of histogram i; the maximum number of elements in all nonzero bins of histogram i. (a) The histograms generated using WHAM with 0.01A bin width. (b) The histograms generated using WHAM with 0.03A bin width.

Total bin #	# bin < 10	Min bin	Max bin
21	5	3	398
22	5	1	378
20	2	1	336
22	5	1	375
24	6	1	393
20	5	1	407
19	2	3	389
21	3	1	347
20	2	1	381
24	6	1	411
22	5	1	381
23	7	1	426
23	6	1	340
20	3	1	379
23	4	2	366
23	8	1	401
21	5	4	367
21	4	1	379
20	4	1	402
21	4	2	368
24	5	5	344
20	2	3	383
24	6	2	352
20	3	1	384
22	3	2	371
23	6	2	344

a

Total bin #	# bin < 10	Min bin	Max bin
8	1	8	1057
8	1	4	1025
7	0	10	978
8	1	5	1089
9	2	1	1070
7	1	7	1054
7	1	3	1002
8	2	3	936
8	1	1	1118
9	2	6	1042
8	1	1	1025
9	2	1	1152
8	1	7	987
7	1	4	1014
8	1	5	994
9	2	4	1126
8	1	4	1040
8	2	4	1032
7	0	10	1108
8	2	3	1063
9	1	5	1004
8	1	3	1098
9	1	3	1032
7	0	10	1103
8	1	6	1055
8	1	6	977

b

The IP peptide was also simulated using the periodic boundary condition, according to the procedure outlined in Chapter 3 and the Appendix. Although the periodic boundary condition was set up successfully and the peptide simulated using Umbrella Sampling, the resulting PMF did not look smooth, as displayed in Figure 5.8 below.

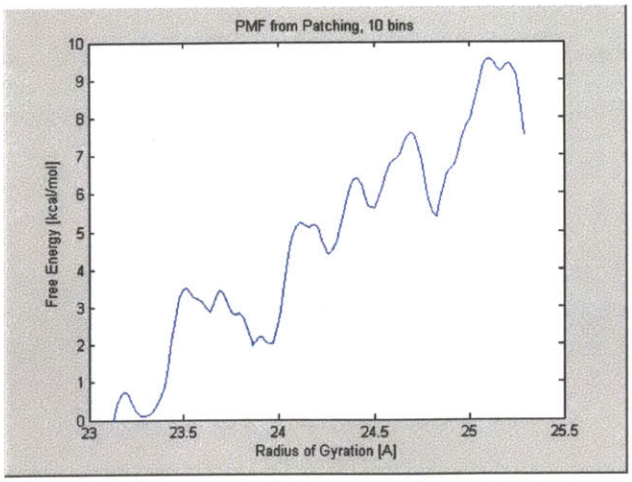


Figure 5.8. The PMF of IP with periodic boundary condition, using the PMF patching algorithm. The result from WHAM did not look smooth. The PMF using WHAM did not converge.

Using the results from stochastic-like boundary condition, we see that the PMF in Figure 5.8 showed a global minimum at 18.4A. This indicates that the structure with the radius of gyration of 18.4A was most stable and therefore the dominant structure. At this radius of gyration, a computation of dihedral angles (see Appendix) showed that 13 out of 27 dihedral angles were found to have Ψ or Φ angle that lies outside of 3 standard deviations of the crystallography Ψ or Φ angle average. Thus 48.15% of the residues at 18.4A are considered coil, while the other 51.85% are considered as folded native. That is, according to Equations 4.11 and 4.12,

Equation 5.1
$$CD_{pred}^{IP,300K} \approx CD_{18.4nm}^{IP,300K}$$

Equation 5.2
$$CD_{18.4nm}^{IP,300K} = 51.85\% CD_{IR}^{10C} + 48.15\% CD_{IP}^{90C}$$

Using Equations 5.1 and 5.2, we then compare the predicted CD spectra of IP at 300K with the measured CD spectra average of IP at 25°C, as shown in Figure 5.9. The CD spectrum of IP at 25°C is measured according to the procedures outlined in Chapter 2.

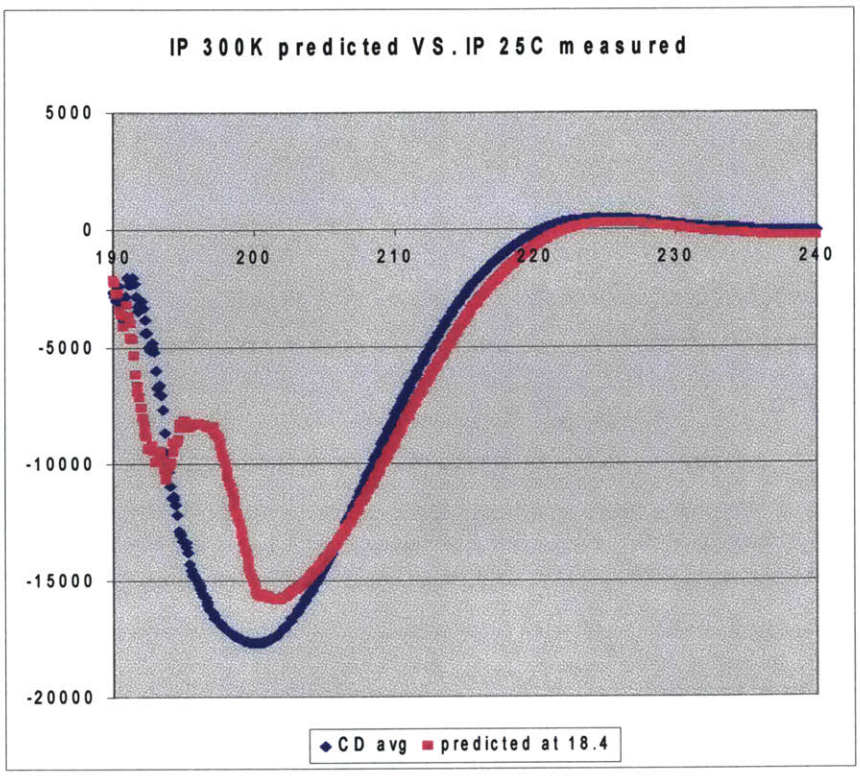


Figure 5.9. A comparison of the predicted IP CD spectrum at 300K and the measured IP CD spectrum at 25C. The predicted CD spectrum is approximated by the dominant structure at 18.4A as computed by the PMF. The measured CD spectrum is the average of 40 CD spectra.

Figure 5.9 showes that the predicted IP CD spectrum at 300K based on the PMF was similar to the average measured CD spectrum of IP at 25C in the higher wavelength range. At lower wavelengths (especially 202nm and below) however, there was significant difference between the predicted and the measured. The reasons for this difference and proposed methods for improving the results are described in Chapter 6.

5.4 Imino-Poor Peptide from Non-Cleavage Region (IP2)

The IP2 peptide was simulated using stochastic-like boundary condition at 300K. The resulting potential of mean force is generated with both the PMF patching and WHAM and displayed in Figure 5.10.

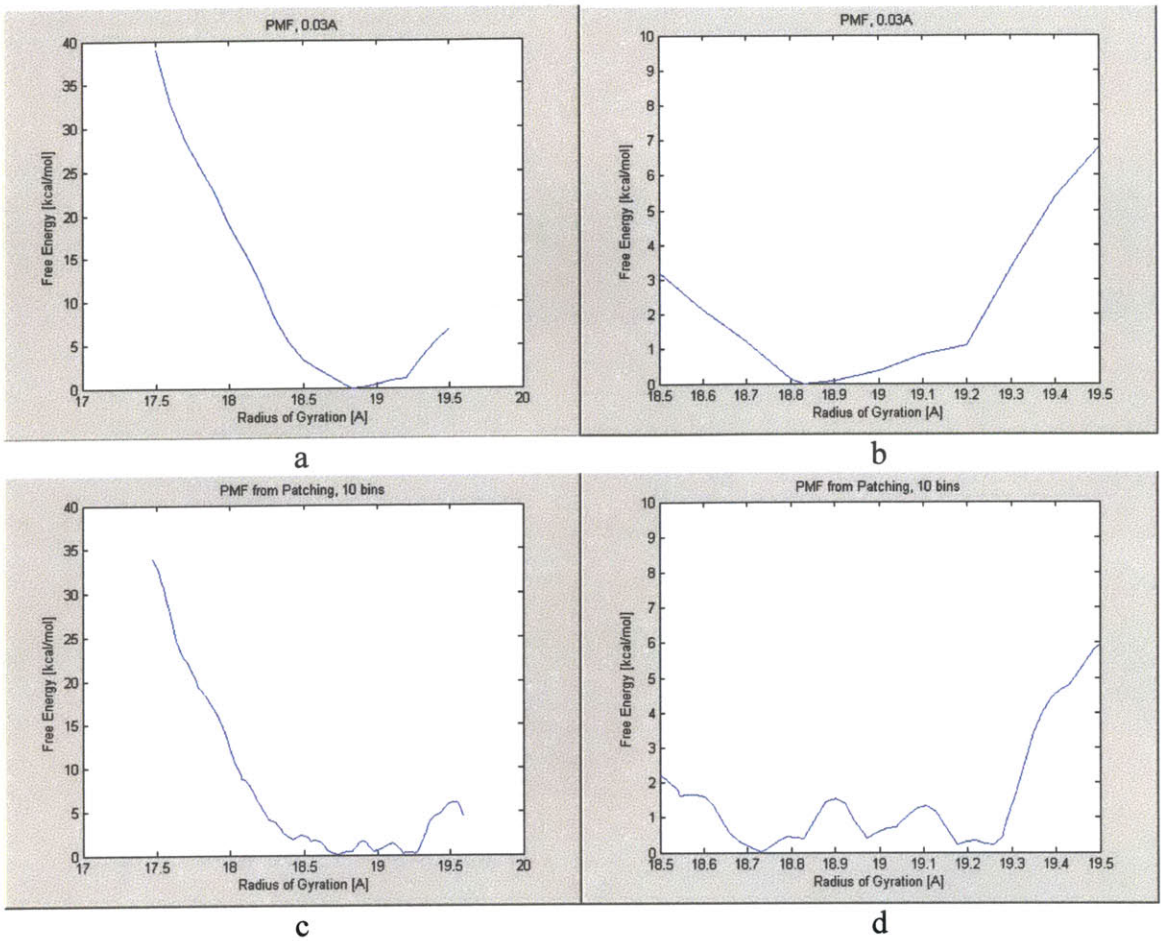


Figure 5.10. PMF of IP2 peptide at 300K, using stochastic-like boundary condition. (a) PMF generated using WHAM, with 0.03A bin width. (b) Zoom-in of a. (c) PMF generated using PMF Patching, with 10 bins. (d) Zoom-in of c.

The PMF generated using WHAM at 0.03A bin width showed one energy minima at 18.83A. The structures at 18.8A and 18.9A almost had the same energy, therefore additional simulations between the two were conducted to find the true minimum. PMF generated using WHAM under different bin width and additional simulations showed similar results and are not included here. The PMF generated using the patching algorithm (Figure 5.10c) seems to have the similar shape but is much less smooth, indicating possible problems in overlap regions. The PMF generated using the patching algorithm with additional simulations also showed similar shape, but the locations of the

minima varied significantly compare to Figure 5.10d. The PMF is therefore approximated using Figure 5.10a, which shows that the structure at 18.83A to be the minimum.

From here, dihedral angle analysis is done on the minimum structure. The analysis shows that at 18.83A, the structure is 55.56% coil and 44.44% native. The overall predicted CD spectrum for the entire structure is then computed and compared with the average measured CD spectrum for the IP2 peptide as shown in Figure 5.11. Although a first glance of the plot indicates that the predicted CD spectrum matches poorly with the measured CD spectrum, a closer look reveals that a large part of the predicted CD spectrum is similar to the measured CD spectrum but shifted by a certain offset. The real difference between the two spectra is again the region near and below 200nm. A hypothesis on why the mismatch occurs between the predicted and measured spectra here is discussed later on in Chapter 6.

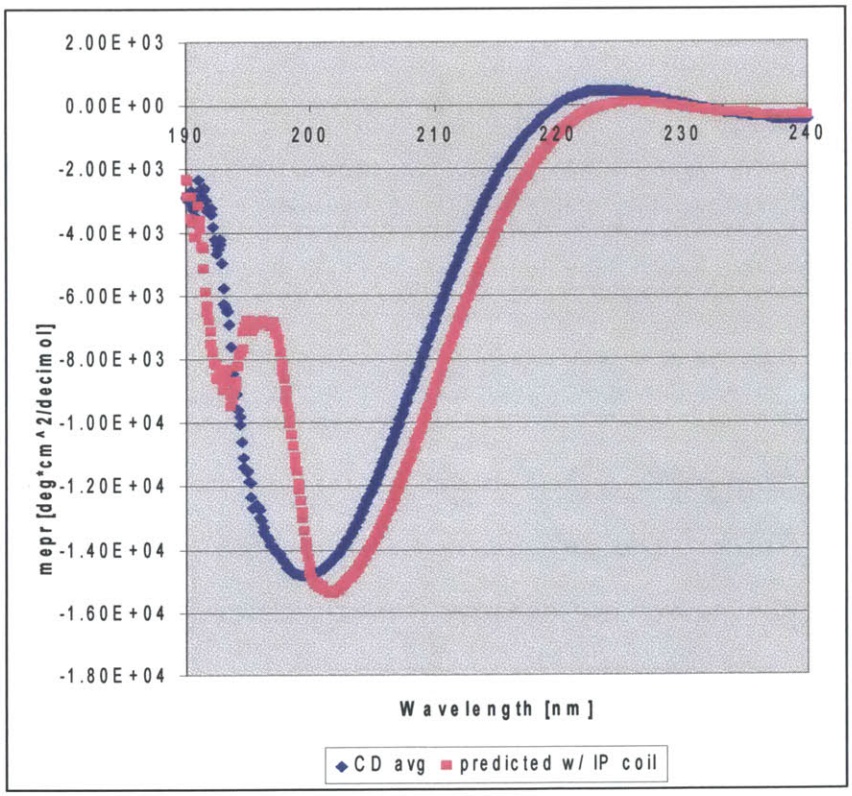


Figure 5.11. A comparison of the predicted IP2 CD spectrum at 300K and the measured IP2 CD spectrum at 25C. The predicted CD spectrum is approximated by dominant structure at 18.83A, as computed by the PMF. The measured CD spectrum is the average of 30 CD spectra.

5.5 Glycine to Serine Mutant Peptide (G2S, G2S Long)

The G2S peptide is a mutant form of the IP peptide in which a glycine has been substituted into a serine residue. Here the computational results of the G2S peptide from molecular simulations are compared with the experimental results for both the G2S and the G2S long peptide. The G2S long peptide is a longer version of the G2S peptide, in which three extra POG triplets were added to each terminal. The reason that the G2S long peptide was included and compared with the predicted G2S spectra is included later in this chapter. Stochastic-like boundary condition was set up at 300K, and the resulting G2S peptide PMF is displayed in Figure 5.12.

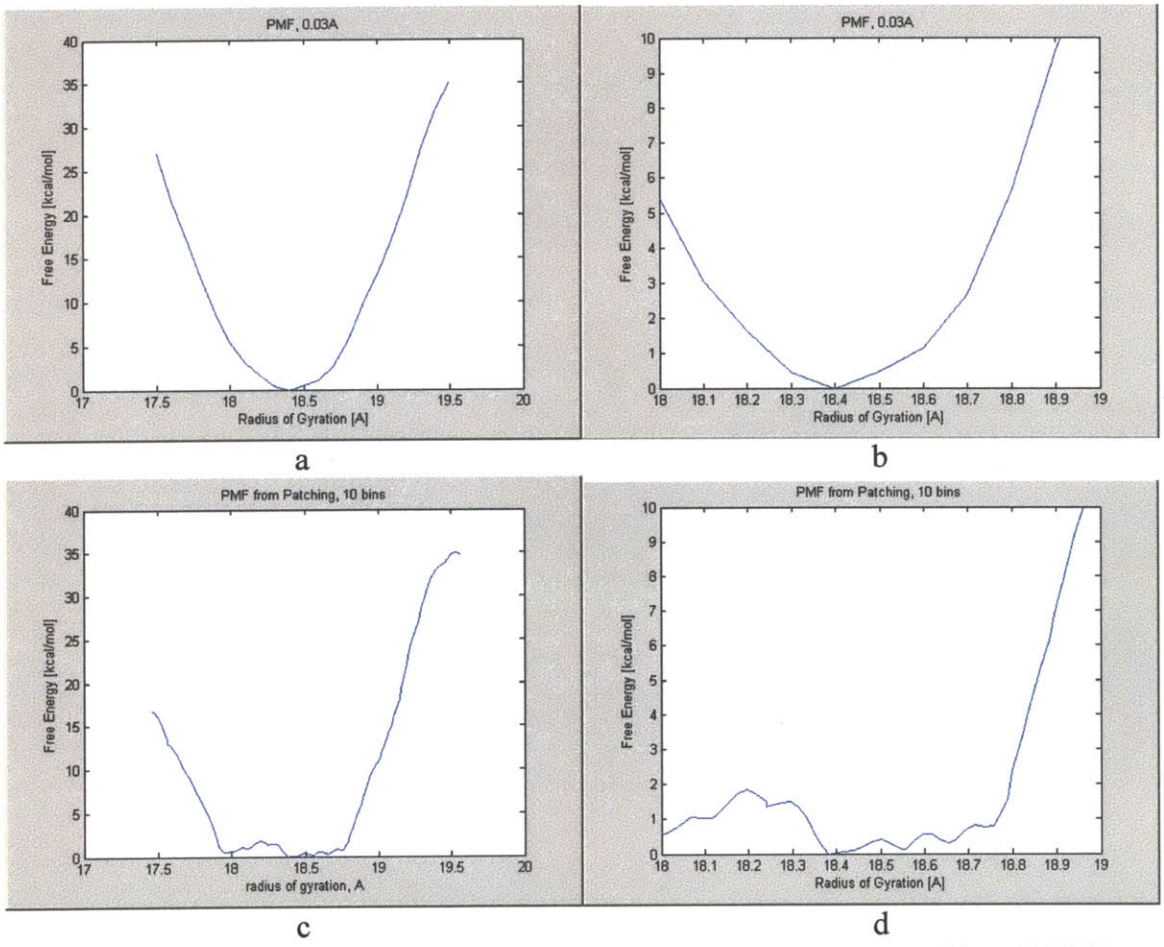


Figure 5.12. PMF of the G2S peptide at 300K, using stochastic-like boundary condition. (a) PMF generated using WHAM, with 0.03A bin width. (b) Zoom-in of a. (c) PMF generated using PMF Patching, with 10 bins. (d) Zoom-in of c.

Once again, while the PMF generated from WHAM showed self consistent results under different bin width and number of simulation windows, the PMF generated using the patching algorithm was not consistent under additional simulation windows. Therefore, the PMF of G2S was approximated by Figure 5.12a. From the resulting PMF, one energy minimum was identified at the radius of gyration 18.4A. Dihedral angle analysis of the structure at 18.4A revealed that the structure is composed of 40.74% coil residues and 59.26% of native residues. Using these data and the IP 90C peptide as the random coil spectrum, the predicted CD spectrum at 300K is constructed in Figure 5.13.

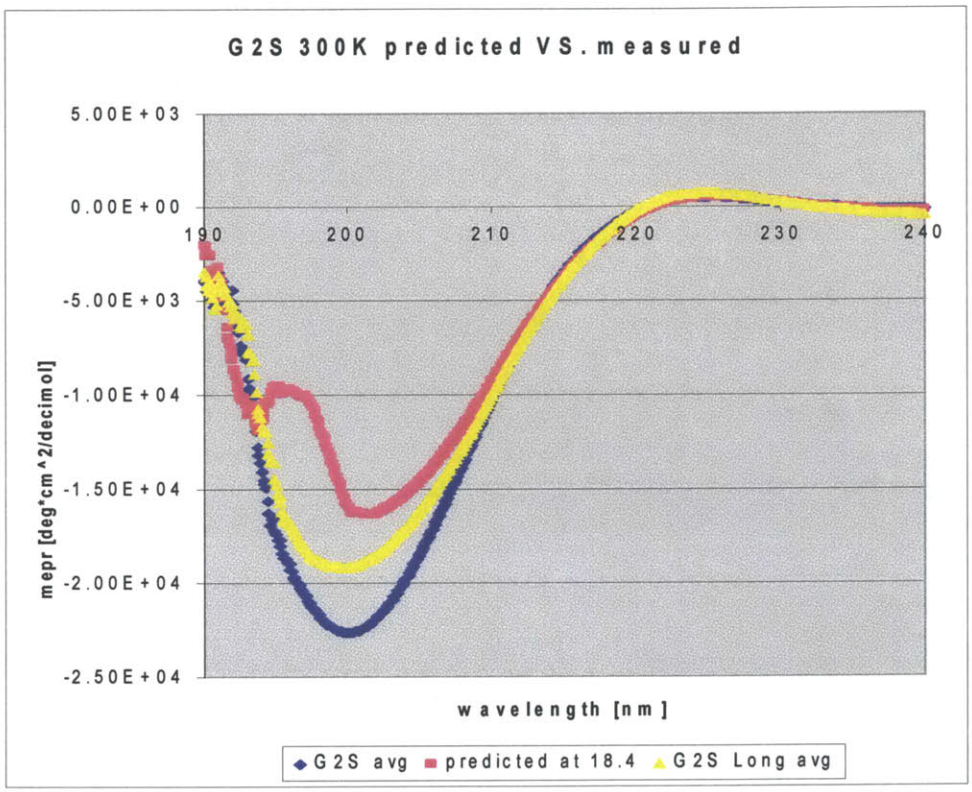


Figure 5.13. A comparison of the predicted G2S CD spectrum at 300K, the measured G2S CD spectrum at 25C, and the measured G2S long CD spectrum at 25C. The predicted CD spectrum is approximated by the dominant structure at 18.4A, as computed by the PMF. The measured CD spectra is the average of 30-40 CD spectra.

The comparison between the predicted (pink) and the measured CD spectra (blue) indicated a good match at wavelengths greater than 210nm, but a poor match at wavelengths less than 210nm. This could be caused by two reasons. First, the random coil spectrum used to generate the predicted CD spectrum is the IP 90C spectrum, which may or may not be appropriate. Second, the measured G2S peptide spectrum may not be

accurate due to high fluctuation; in fact, the measured CD of the G2S long peptide (yellow) was closer to that of the predicted CD spectrum, but differences still exist due to the fact that the G2S long peptide contained 6 extra POG triplets than the simulated G2S peptide. Both of these causes are discussed further later in this chapter or chapter 6.

5.6 Structural Analysis of Collagen-Like Peptides

The structure of the IP, IP2, and G2S peptides at each of their energy minimum conformational states is shown in Figure 5.14. Each structure can be divided into three regions: the two imino-rich segments at the ends of the peptides, and one imino-poor segment (labeled as section between two residues) near the middle. While the imno-rich segments at the ends are similar for all peptides (folded triple helical), the imino-poor segments and their immediate surroundings differ significantly. The location corresponding to the scissile bond is labeled.

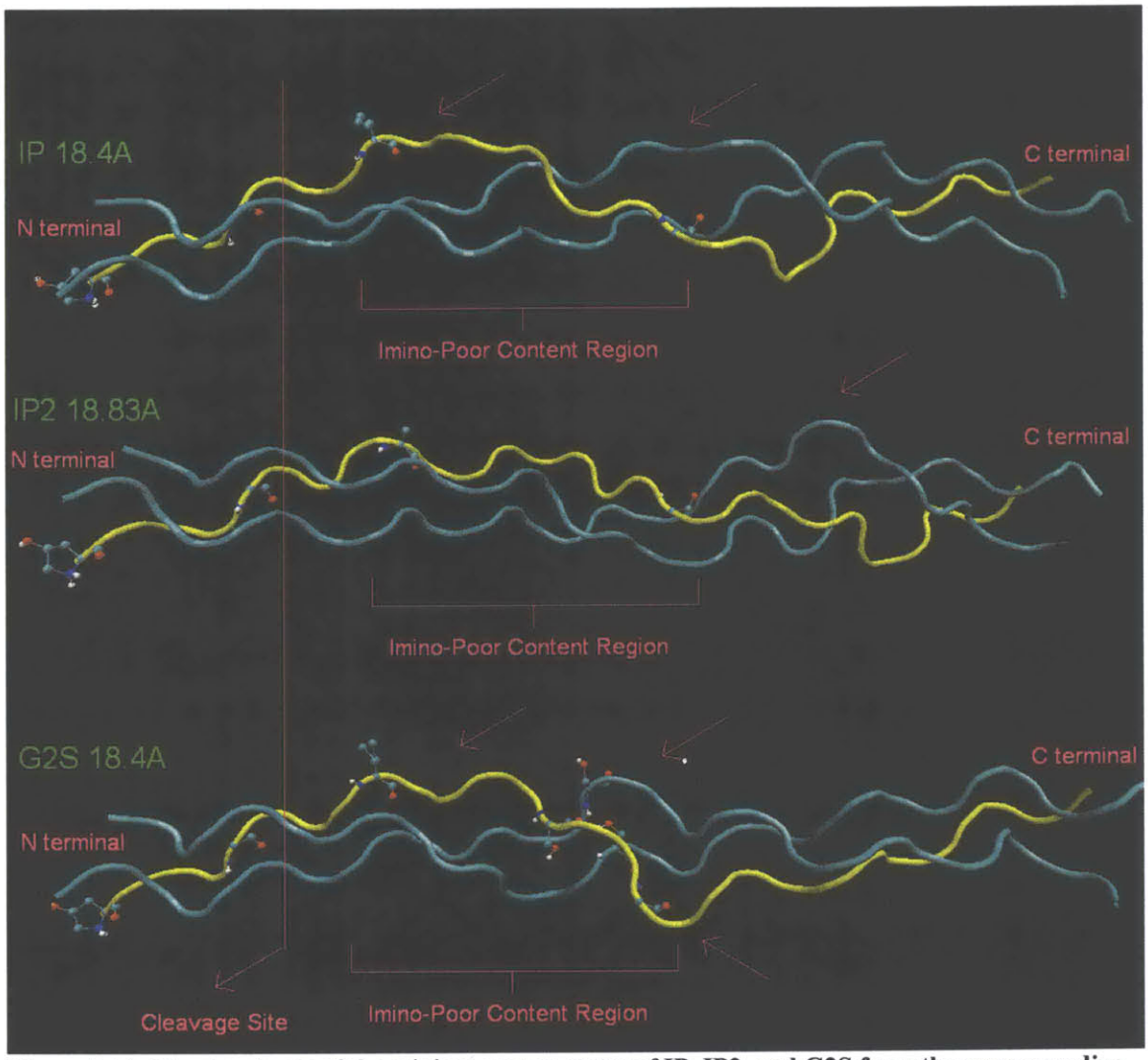


Figure 5.14. The structures of the minima energy states of IP, IP2, and G2S from the corresponding PMFs. All peptides go from the N to the C terminal. The location corresponding to the scissile bond is indicated by a line through all peptides. The imino-poor region of each peptide is labeled as the region between two residues expressed in ball-and-stick form. Chain A of each peptide is indicated by the first amino acid (hydroxyproline) and is expressed in yellow. Plot generated with VMD.

The IP peptide structure at 18.4A radius of gyration, for example, contains an imino-poor region that is characterized by two noticeable open loops in chain A and chain B, each indicating a potential partially unfolded region within the imino-poor segment that have lower stability when compared to the imino-rich segment (Figure 5.14). Here we defined the loop as an opening or a region of the chain that is physically raised up above the rest of chain, as indicated by the 3D structure rotation in VMD. Loops thus provide additional surfaces or areas where solvent or collagenases can access, or provide partially unfolded regions. The loops in the IP peptide are labeled by red arrows, and are to the immediate C

terminal direction of the cleavage site (labeled with a line). It is possible that these loops may be the basis for local unfolding near the cleavage site, which exposes the scissile bond.

The PMF of the IP2 peptide revealed that IP2 had dominating conformational state at 18.83A. Analysis of the structure of IP2 showed that the imino-poor segments of IP2 (labeled between two residues) appeared to be more folded than that of the IP peptide. The imino-poor region of IP2 is fairly tightly packed, yielding no obvious open loops for partial unfolding sites. The areas near the cleavage region are therefore folded, making the scissile bond inaccessible to solvent and collagenases. The only potential open loop on the IP2 peptide is labeled by an arrow near the C terminal end of the imino-poor segment, which is far downstream from the scissile bond. Such evidence could suggest that the imino-poor region of the IP2 peptide is indeed more stable than that of the IP peptide since it contains a more stable imino-poor segment and the potential partially unfolding site is further away from the location of the scissile bond. Note also that this result is consistent with the melting point experiments, where the IP2 peptide was found to be more thermally stable than the IP peptide since IP2 had a higher melting point. In this case, although both the IP and IP2 peptides contain an imino-poor amino acid sequence, only the IP peptide seemed to expose its scissile bond by allowing local unfolding near the cleavage site. This could therefore be an important part of the reasons that the scissile bond is unique near the IP peptide's imino-poor region, but not other imino-poor regions.

An analysis of the IP and IP2 peptides at maximum energy states from the corresponding PMFs further supports the above theory. Figure 5.15 shows the structures of the IP and IP2 peptides at 19.5A and 17.5A radius of gyration, respectively. These radii of gyration are chosen such that the peptides are at their least stable state (highest energy), as revealed by their PMFs. At 19.5A, we see that the open loop of the IP peptide has extended towards the N terminal direction, or the direction of the scissile bond, therefore exposing the cleavage site further by increasing local unfolding near the cleavage site. The IP2 peptide at 17.5A, on the other hand, showed an open loop at the C terminal end

of the imino-poor segment, while the regions near the corresponding scissile bond is folded and not exposed. The protected scissile bond therefore lead to a more stable structure than the IP peptide, since more energy is required to further extend the partial unfolding of IP2 to the N terminal end of the imino-poor segment before the scissile bond is exposed.

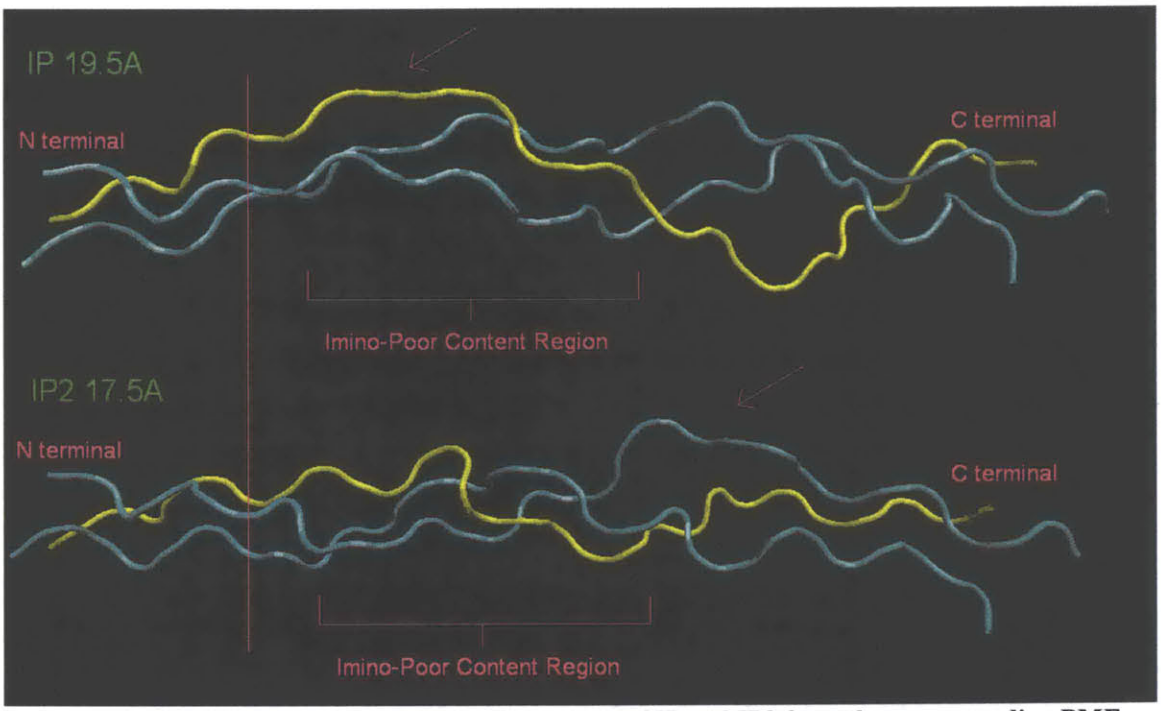


Figure 5.15. The structures of maximum energy states of IP and IP2 from the corresponding PMFs. All peptides go from the N to the C terminal. The location corresponding to the scissile bond is indicated by a line through all peptides. The imino-poor region of each peptide is labeled as the region between two residues expressed in ball-and-stick form. Chain A of each peptide is indicated by the first amino acid (hydroxyproline) and is expressed in yellow. Plot generated with VMD.

The structure of the G2S peptide in Figure 5.14, on the other hand, showed that there are several potential open loops in the imino-poor segment of the G2S peptide (labeled by arrows). Most of these potential partial unfolding sites are near the glycine to serine substitution site, where the hydrogen side chain of glycine is replaced by a bulkier side chain of serine, therefore disrupting the triple helix structure in that region. On the other hand however, a closer investigation in Figure 5.16 reveals that the polar side chain of serine enables the formation of several additional hydrogen bonds, which act together to stabilize the triple helix structure in that region. Furthermore, these hydrogen bonds encourage the side chains of serine to rotate such that the most favorable positions are

adapted to relieve the stress brought about by bigger side chains. As a result, although several potential partial unfolding sites are identified near the serine substitution site, the structure remained fairly stable. In fact, the most stable conformational state of the G2S peptide occurred at 18.4A as revealed by PMF, which is the same radius of gyration as that of the most stable state of the IP peptide.

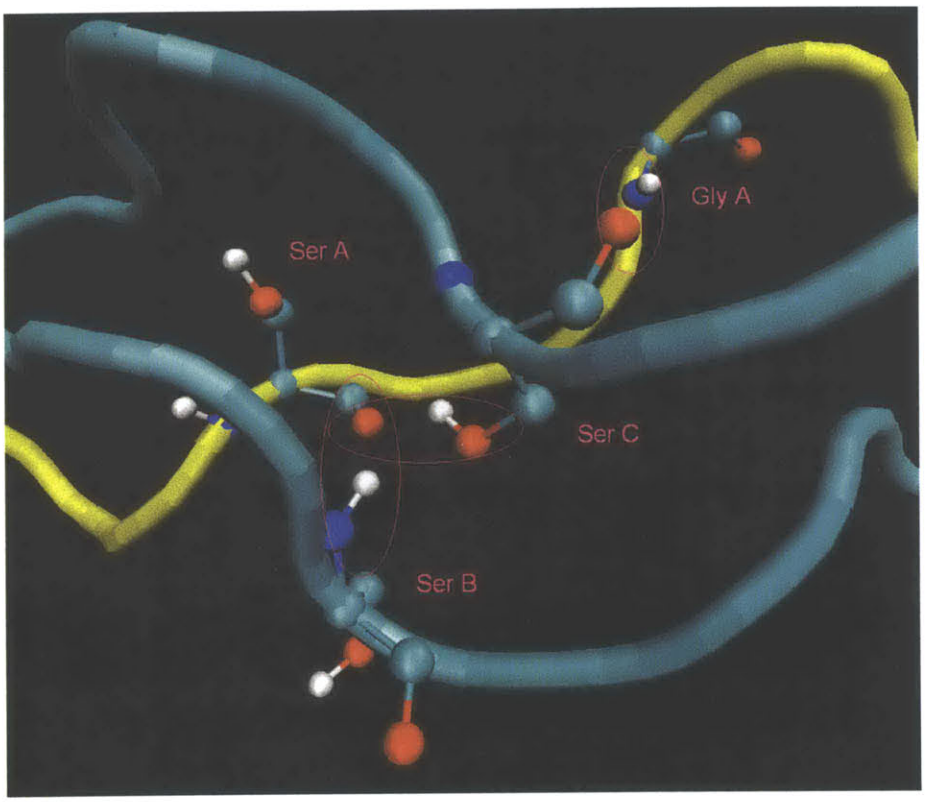


Figure 5.16. A zoom-in of the glycine to serine substitution site. Hydrogen bonds are circled. Chain A is shown in yellow. Three serine residues (one per chain) are shown in ball-and-stick representation. Plot generated with VMD.

Finally, the analysis of the high energy G2S peptide structure at 19.5A radius of gyration shows that the local unfolding of the G2S peptide happens further away from the scissile bond location than that of the IP peptide. This is shown in Figure 5.17. Since the cleavage site is further away from the scissile bond, and if we also assume that the scissile bond is the only possible cleavage site, then it is likely that the G2S peptide have a slower rate of degradation by collagenases than that of the IP wildtype. This would suggest that the Ehlers-Danlos Syndrome type IV caused by the G2S mutant is a collagen degradation related disease where the slower degradation rate of the G2S mutant causes a longer life

of the mutant collagen monomers, which leads to the formation of additional crosslinks, and finally causes the stiffening of the collagen fibers. All these contribute to stiffer arterial walls, which could in time lead to the possible rupture of structures. In this way our studies seem to suggest that the EDS type IV is possibly caused by an over-stabilization of mutant collagen instead of under-stabilization. However, more research is needed to make a conclusion.

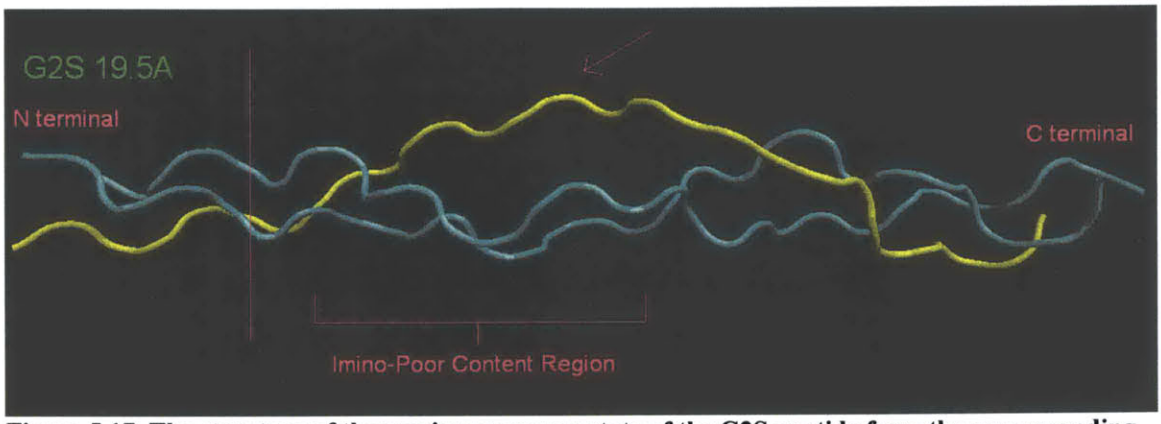


Figure 5.17. The structure of the maximum energy state of the G2S peptide from the corresponding PMF. The peptide goes from the N to the C terminal. The location corresponding to the scissile bond is indicated by a line. The imino-poor region of each peptide is labeled as the region between two residues expressed in ball-and-stick form. Chain A is indicated by the first amino acid (hydroxyproline) and is expressed in yellow. Plot generated with VMD.

Chapter 6

Discussion and Future Studies

In this section, we compare the proposed method of determining the identity and distribution of conformational states in a collagen-like peptide solution with an existing method of measuring the triple helical presence using the Rpn ratio. A number of possible improvements in the experimental studies of our research are suggested including equilibration, melting point extrapolation, basis function modification, etc., all of which can lead to improvements in the proposed method of determining the distribution of conformational states in a collagen-like peptide solution. Finally, a number of computational issues are discussed and possible improvements are suggested for future studies.

6.1 Quantifying Triple Helical Content

The characteristics of a triple helical collagen-like CD spectrum have been observed and known for decades. These include a small positive peak around 220-225nm, one crossover point where the ellipticity goes from positive to negative, followed by a large negative peak around 200nm. Despite these observations, not much research has been done in directly quantifying the triple helical content of a protein or peptide to the best of our knowledge. One of the existing methods computes the ratio between the positive and the negative peak magnitudes (Rpn) of the CD spectrum and compares the resulting Rpn

with the Rpn computed from native collagen or stable collagen-like peptide $(POG)_{10}$ (Feng *et al,* 1996; Usha and Ramasami, 2004). Feng *et al* claimed that the Rpn ratio is a way to establish the presence of triple helix in solution since it successfully detected an increase in presence of triple helical structure with increasing number of POG triplets in the peptide chain. However no known evidence has proven that the Rpn ratio is a correct measurement of the triple helical content. In fact, when Usha and Ramasami used Rpn ratio to measure the effect of urea on the degradation of collagen protein, they noticed an initial sharp increase in the Rpn ratio with the introduction of urea, followed by a sharp decrease in Rpn when urea concentration in solution exceeded some critical value. Physically, it is hard to imagine that urea can first stabilize the peptide, then destabilize it. Such observation could therefore suggest that the Rpn ratio may indeed not be the correct indication of the presence of triple helical structure. Furthermore, the Rpn values do not provide any direct way to measure the actual amount of triple helix presence in solution. As reported by Feng *et al*, the Rpn ratio for a native collagen sequence or the $(POG)_{10}$ peptide was measured to be around 0.17. However, a Rpn ratio of 0.085 (half of 0.17) does not translate directly into a solution with 50% triple helix presence. As a result, the Rpn ratio may potentially be a rough method to rank the amount of triple helix in solution but it can not be used to determine the exact percentage of triple helix content.

In our research, we propose a method to determine the identity and distribution of conformational states in a collagen-like peptide solution. We use the computed PMF from molecular dynamics simulations with Umbrella Sampling to determine the number of stable conformational states in the solution. From there we assign Boltzman weights to each structure according to their free energy and construct a predicted CD spectrum. Each energy minimum structure is decomposed into native triple helical and random coil components, where the percentage of coil is assigned by the number of residues with Ψ or Φ angles that are significantly different from the crystallographic average. Finally by comparing the predicted CD spectrum from computational studies against the measured CD spectrum from experimental studies, we noticed that spectra for all three peptides (IP, IP2, and G2S) showed reasonably good similarity at higher wavelengths, but poorer similarity at lower wavelengths. If the lower wavelength region could somehow be

improved, then this would indicate that the computational model using PMF may indeed be a valuable method in determining the identity and distribution of conformational states in a solution.

A second and yet unexplored method for measuring the triple helical content of a peptide is to measure the intensity of its positive peak on the CD spectrum near 225nm. From past experiences, the negative peak on the spectrum does not always seem to be an indication of the triple helix presence. In some cases such as the G2S peptide, even when the peptide is not triple helical as indicated by all other evidences, the intensity of the negative peak still exceed the intensity of the $(POG)_{10}$ peptide. The positive peak on the other hand, always seems to be closely related to the amount of triple helix in solution. The positive peak of all peptides have intensities equal to or less than that of the $(POG)_{10}$ peptide measured at 10°C, and the positive peak intensity also decreases in an inverse relationship with increasing temperature. Furthermore, for the random coil measurements (IP 90°C, IP2 90°C, IP SDS), the positive peak disappears indicating the non-presence of triple helix under those conditions, while the negative peak could be still very large as in the case of IP SDS (Figure 6.2). It is possible therefore that the intensity of positive peak on the CD spectrum is an indication or measurement of the amount of triple helix, whereas the intensity of the negative peak is an indication of something else—perhaps how exposed a certain region is—so that the intensity of the negative peak is fairly unpredictable. More data and analysis however is needed to make a conclusion.

6.2 Experimental Discussions and Improvements

Several proposed improvements to the experimental aspect of the research have been listed below. Some of these improvements have already been adapted thanks to the work of Ramon Salsas Escat.

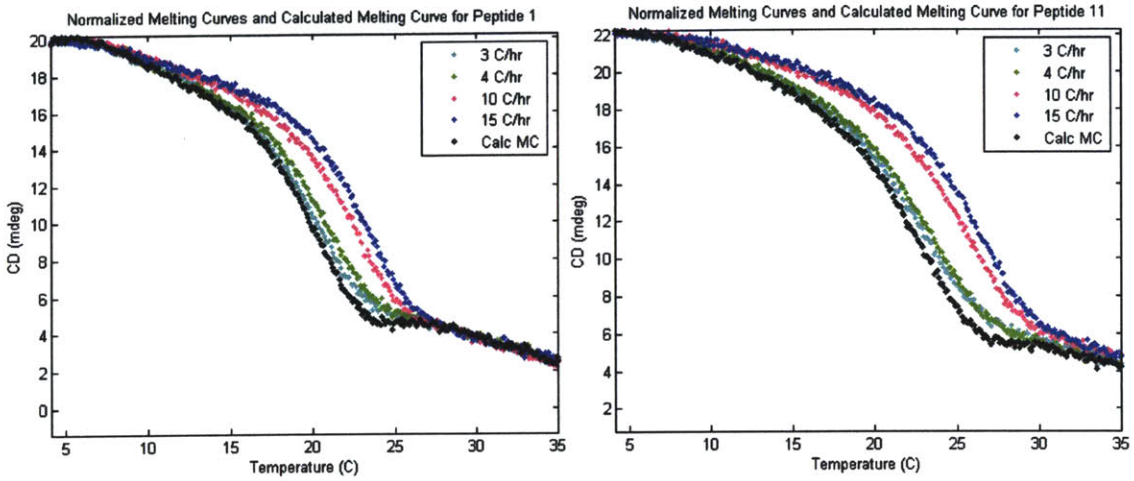
6.2.1 Peptide Equilibration

In the earlier stages of our research we assumed that an equilibration time of 15 minutes is generally good enough for peptides to reach equilibration. This is generally accepted because an incubation of 24 hours was already made before any peptide is used to ensure the formation of triple helix, and the resulting CD spectrum from different samples of the same peptide are usually consistent (for IP, IR, IP2, and G2S long peptides). However, it was noticed that for the G2S peptide, the CD spectrum from different samples done under the same conditions are significantly different from each other. This can be caused by one of two things, assuming the problem is not mechanical in nature: either the G2S peptide is very unstable for certain reasons, or that the equilibration time for the G2S peptide is larger than the currently applied. To be rigorous, the equilibration time curve should be measured for each peptide by measuring the mean ellipticity per residue of the peptide at a fixed wavelength and fixed temperature using CD. This process is still undergoing and is being incorporated into the recent stages of the research.

6.2.2 Extrapolating the Melting Curve

The equilibration of the peptide at a certain temperature also brought up another issue: when measuring the melting curve of a peptide, the resulting melting curve depends on the rate at which the temperature is raised. If the temperature is raised too fast, the peptide is not given enough time to reach equilibration at each temperature, and the reported CD ellipticity is therefore inaccurate, leading to an inaccurate melting point. This is verified when different melting curves for a peptide under different heating rates yielded slightly different melting points. To solve this problem, the melting curve of a peptide was measured for numerous trials, where each trial uses a different heating rate. From here, there are two ways to obtain the melting point. The first is to compute the melting point for each melting curve and establish the relationship between the melting point and the heating rate then extrapolate the melting point to the heating rate of 0C/min. The rate of 0C/min is equivalent to infinitely long equilibration time at each temperature, which gives the ideal melting point solution. The second method is to extrapolate each

point on the melting curves to the heating rate of 0C/min, generate a new melting curve that is equivalent to the melting curve measured at 0C/min then compute its melting point. A typical plot of this process is shown in Figure 6.1 below. Preliminary results for this study done by Ramon Salsas Escat showed that the melting points for the IP and IP2 peptides were 18.96C and 21.37C, respectively.



a b

Figure 6.1. Typical plots of extrapolating the melting curves to the heating rate of 0C/min. (a) The extrapolated melting curve for the IP peptide (peptide 1) is shown in black. (b) The extrapolated melting curve for the IP2 peptide (peptide 11) is shown in black. Notice that since the scale on the Y axis does not affect the melting point calculation, the ellipticity is not normalized to mean ellipticity per residue (mepr). Also note that when higher heating rate is used, a higher melting point is obtained since the melting curve is shifted to the upper right. Data measured by Ramon Salsas Escat.

6.2.3 Improving the Basis Spectra

The collagen native and random coil spectra measured in Figure 5.1 have the following characteristics: at wavelength above 200nm, both spectra appear to be smooth and reasonable; but at wavelength below 200nm, the native (IR 10°C) spectrum showed an unsmooth and small secondary peak at about 195nm that looked like noise, and the random coil (IP 90°C) spectrum showed a huge positive peak around the same wavelength that may or may not be noise. Both of these peaks are not among the characteristics of the typical CD spectrum of a collagen protein or collagen-like peptide. Given the shape and values of the IR 10°C spectrum immediately before and after the secondary peak, and noticing the unsmoothness of that peak compared to other regions of

the spectrum (Figure 6.2), it is likely that the secondary peak on the IR 10°C spectrum is caused by noise rather than the real CD of IR 10°C. Next, we notice that the irregular positive peak on the IP 90°C spectrum is around the same wavelength region, therefore it is possible that this peak is also caused by noise, where the magnitude of this noise is amplified due to the much higher temperature of 90°C.

Such noncollagen-like characteristics in the native and random coil basis functions are the major causes of the differences between the predicted and the measured CD spectrum for the collagen-like peptides. In the case of the IP and G2S peptide, for example, the majority mismatch between $CD_{predicted}$ and $CD_{measured}$ happened at the low wavelength region, whereas the middle and high wavelength regions showed little difference.

A number of changes could be made to potentially improve the accuracy of the basis function spectra. First, additional spectra can be measured to provide a smoother curve by averaging over more samples; this will use statistical averaging to decrease the amount of noise in the spectrum. Second, literatures can be reviewed to determine the cause of interference at about 195nm: if it is air, CD in vacuum or pure nitrogen may be considered; if it is the PBS solvent, another solvent could be tested; or if it is some elements in the cuvette material that absorbs in the measurement wavelength, different cuvette types could be tried. With these above changes, a more accurate spectrum of the IR peptide at 10°C should be obtained. However, the noise in the IR peptide is small compared to the possible noise in the IP 90°C spectrum, since experiences show that increasing the temperature is usually associated with increasing noise intensity. One way to get around this is to measure the random coil spectrum by degrading the peptide using other means than heat, and avoid high temperature completely. For example SDS, urea, n-propanol, or certain enzymes could be used to yield the degradation process instead of a thermal degradation at 90°C. Such methods could give a random coil CD spectrum that is similar to the IP 90°C spectrum at middle and high wavelengths, but different in the lower wavelength regions. Of course, if external chemicals are added, we must also make appropriate changes to the base line of the CD spectrum, for the newly added substance may contribute to the CD itself.

Preliminary results of using SDS to obtain the random coil spectrum have shown that the resulting CD spectrum is similar to the IP 90°C spectrum at wavelengths above 210nm, but significantly different for wavelengths below 210nm (Figure 6.2). Notice that in the figure, the SDS degraded IP peptide spectrum showed a narrower and earlier positive peak at below 195nm. The reduction of this irregular positive peak when compared to that of the IP 90°C spectrum could be caused by the lower measurement temperature at 37°C instead of 90°C. In fact, even if the conditions of degradation remained the same—using thermal degradation at 90°C—but a different peptide is used (such as IP2 90°C), the resulting CD is also somewhat different from that of IP 90°C. It is yet not clear why this is the case—perhaps that the structure and characteristics of the resulting random coil depends on both the starting structure and the type of degradation so that if either changes, the ending random coil structure also changes. Because of this effect, it is not intuitive which random coil spectrum we should use for all peptides, or whether we should use a different random coil spectrum for different peptides.

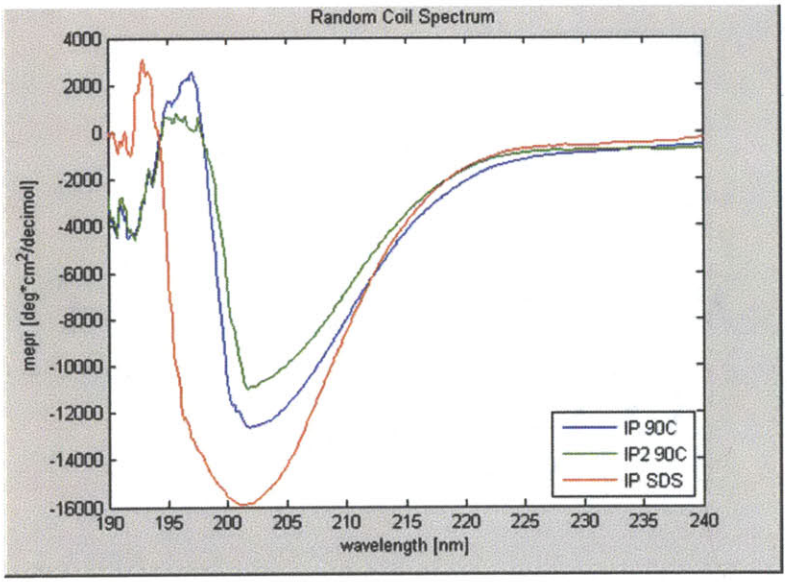


Figure 6.2. The random coil spectrum, measured using IP at 90°C, IP2 at 90°C, and IP degraded by 90°C followed by 2% SDS, and measured at 37°C. All three curves showed similar shape and value at 210nm and above, but the behavior were very different below 210nm. The IP SDS data is measured by Ramon Salsas Escat.

6.3 Computational Studies: Discussions and Improvements

A number of computational issues are discussed in this section, including previous boundary conditions used and the differences in results, etc. A number of modifications are suggested to improve the proposed method for determining the identity and distribution of conformational states in sample solution.

6.3.1 Stochastic VS. Stochastic-Like: A Comparison with Previous Results

Previous research in our group investigated the IP peptide under stochastic boundary condition at 25°C. The stochastic boundary was set up by surrounding a sphere made of water molecules around 20A of the center of mass of the IP peptide, then displacing the water sphere by ±10A along the direction of the triple helix axis, forming a football like water surrounding as shown in Figure 6.3. The reaction region was defined to be anything within 30A of the center of mass, the buffer region was defined to be anything between 30A and 35A of the center of mass, and the reservoir region was defined to be anything outside 35A. The reaction region underwent full molecular dynamics, the buffer region was constrained by harmonic constraints, and the reservoir region was fixed completely. A total of approximately 1000 water molecules were used in the resulting structure.

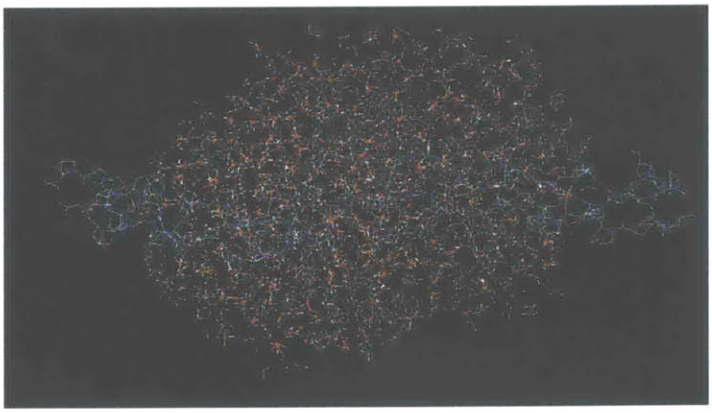


Figure 6.3. The stochastic boundary condition set up for the IP peptide in a previous research (Stultz, 2002).

The resulting PMF from the stochastic boundary study is shown in Figure 6.4. The zoomed-in PMF generated from the PMF patching algorithm shows that two clear energy minima can be identified with about the same amount of energy, namely the native and the vulnerable structures (labeled N and V in Figure 6.4). Using Equations 4.11 and 4.12, we computed the percentage of the N and V structures to be 58% and 42%, respectively. From there, dihedral angles analysis was done to approximate the percentage of native and coil components within each structure. The CD spectrum for the N structure was approximated by that of the $(POG)_{10}$ peptide, whereas the CD spectrum of the V structure was approximated by the 38% native and 62% coil. The resulting predicted CD spectrum, as well as the measured CD spectrum, is shown in Figure 6.5.

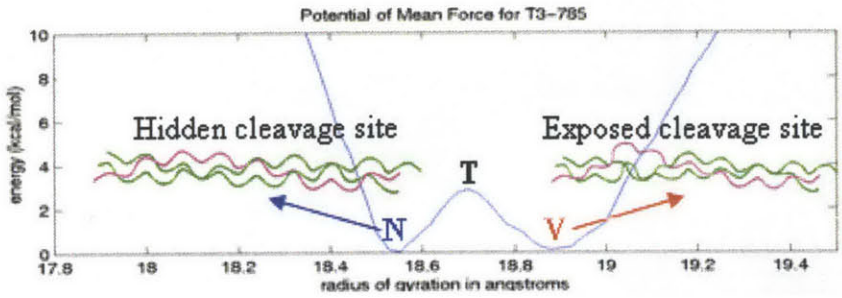


Figure 6.4. The computed PMF from stochastic boundary condition for IP peptide in a previous research. This PMF was generated using the PMF patching algorithm. Modified from (Salsas *et al*, 2005, submitted).

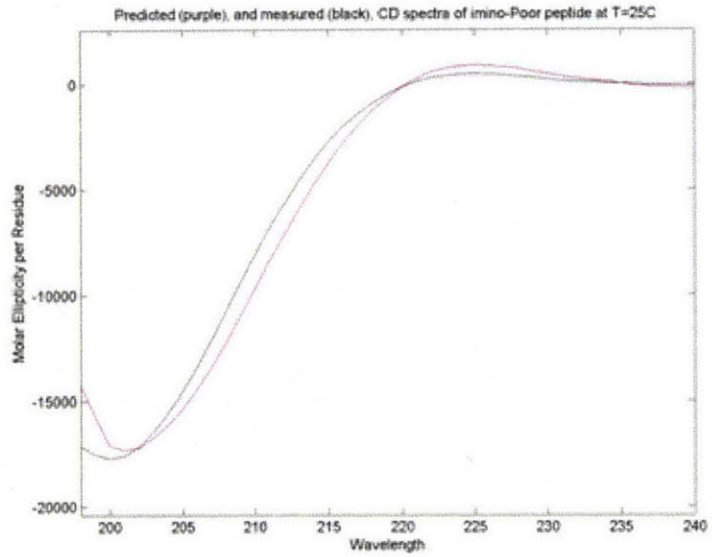


Figure 6.5. The predicted PMF calculated from the PMF (purple) and the measured PMF (black) for the IP peptide at 25C (Salsas *et al*, 2005, submitted).

If we compare Figure 6.5 with Figure 5.4, we see that both of the predicted CD spectra from the previous stochastic boundary condition and that from current stochastic-like boundary condition are similar to the measured CD spectrum of the IP peptide at 25°C. The predicted CD spectrum from the current stochastic-like boundary condition did better at middle and higher wavelengths (205nm and above), whereas the previous stochastic CD spectrum did much better at lower wavelengths (below 205nm). The differences between the two predicted CD spectrum may be caused by the number of water molecules involved in the simulation as well as the slightly different use of the boundary condition in the buffer region (harmonic constrained in stochastic VS. fixed in stochastic-like). In summary, the stochastic boundary condition did better at the negative peak, while the stochastic-like boundary condition did better at the positive peak; since neither of the predicted CD spectra did well at both peaks, it becomes hard to determine which one is the better result.

6.3.2 Troubleshooting Periodic Boundary Condition Simulation

Although the periodic boundary condition is commonly used in molecular dynamics, our simulation using PBC for the imino-poor peptide at 25°C did not yield reasonable results. This could have happened for a number of reasons. For example, periodic boundary condition fixes the collagen-like peptide at only one atom in order to keep the molecule at the same location in the cell. This gives the peptide many more degrees of freedom than the stochastic (or stochastic-like) boundary condition in which the peptide is fixed in the reservoir region, restrained (or fixed) in the buffer region, and only undergoes full molecular dynamic simulation in the reaction region. In the case of collagen and collagen-like peptides, a significant portion of the protein or peptide contains the POG sequence. Experimental studies have shown that these regions form triple helical structure that are very stable and are usually considered fixed and uncleavable even in the presence of proteinase or peptidase. As a result, the imino-poor (low POG content) regions of the collagen protein or collagen-like peptides are usually fixed at their ends by neighboring imino-rich POG triplets, as in the case of our collagen-like peptides such as

$(POG)_3ITGARGLAG(POG)_4$. Given these, stochastic boundary condition would seem to provide a much better approximation of the simulation environment than the periodic boundary condition for the cases of collagen and collagen-like peptides.

There are also few more specific reasons on why the periodic boundary condition did not yield a reasonable result. For example, the nonbonded interaction specified in the periodic boundary condition was smaller than that of the stochastic boundary condition, as pointed out in the Appendix section. The decreased cutoff boundary for nonbonded interactions allows the periodic boundary to be applied and the simulation computed within a reasonable amount of time given its high complexity due to the involvement of images. However at the same time, the decreased cutoff boundary for nonbonded interactions also decrease the accuracy for the simulation, and may have at least partially contributed to the unreasonable results. Finally, it should be pointed out that each resulting individual distribution calculated from each simulation window of Umbrella Sampling was found to be somewhat irregular in PBC instead of the more regular Gaussian-like distributions reported by stochastic boundary molecular dynamics. This directly led to the irregularly shaped PMF, as the final PMF is the weighted sum or physical assembly of the individual distributions. Although it is not obvious why the individual distributions from each simulation became more irregular, it is unlikely that the problem originated from the simulation itself, since the simulation is controlled by MD. It is likely therefore that the problem originated from the periodic boundary condition since this kind of boundary condition may not provide a good simulation environment for collagen and collagen-like peptides.

6.3.3 Improving Computations and Simulations

A number of things could be done to improve the computational portions of this research. First, the stochastic boundary condition should be attempted again instead of the use of a stochastic-like boundary condition. The latter provides an approximation to the former, but the absolute fixing of the buffer region may be an over-estimation of the stability of the POG triplets in the real peptide. Stochastic-like boundary condition was used because

when harmonic constraints were applied to the buffer region, the reaction region also was somehow restrained causing very little change in the entire peptide (see Appendix section). More effort should therefore be made in finding out the cause of this error and finding a correction. However, it is unlikely that stochastic boundary condition with a large number of water molecules will give a much different result than a stochastic-like boundary condition with the same number of water molecules, since the overall movements in the buffer region is very small, as observed through VMD from previous stochastic boundary condition simulations.

After setting up the boundary conditions, another aspect of the simulation that should be investigated is the biasing potential term. Currently, the biasing term is defined to be $A(rg-rg_i)^2$, where A=500, rg is the radius of gyration for a specific structure, and rg_i is the radius of gyration for the current simulation window. The biasing term affects the amount of sampling that is done at each simulation window. For example, if the basing term is high (either A is high or the squared difference is high), the resulting distribution will be a narrow and tall Gaussian-like curve whose peak is near the rg_i of the simulation window. On the other hand, if the biasing term is smaller, the resulting distribution will be wider and shorter, distributing more sample points to the two ends of the distribution. In the second case with the more spread out distribution, we will then have a better chance of getting at least 10 elements per bin for the histograms used by PMF patching and WHAM, and therefore decreasing the statistical errors involved in constructing the PMF. Still, other kinds of biasing function should also be explored to compare the results, such as the use of absolute difference instead of squared difference, $A|rg-rg_i|$, etc.

There are also numerous other ways to potentially improve the construction of the PMF. First, additional WHAM simulation windows could be used to give a higher resolution of the PMF. For example, instead of doing Umbrella Sampling windows centered on radius of gyration 0.1A apart, a simulation of every 0.05A per window near the minima of the energy plot can potentially identify new or additional energy minima which will yield to a better prediction of the CD spectrum. Second, WHAM allows the construction of PMF using any kind of reaction coordinate, or even multiple reaction coordinates. Radius of

gyration was chosen in this research because of its availability in CHARMM and the fact that it has yielded fruitful results in earlier research. But performing simulation based on other reaction coordinates may also provide new insights.

Once the PMF is constructed, the predicted CD spectrum is generated by summing the Boltzman weighted CD spectra of each energy minima structure (Equation 4.11). This process is approximate since it ignores the CD of all other structures in solution. If instead of a discrete summation as in Equation 4.11, a continuous integral is used as in Equation 6.1, the predicted CD spectrum could potentially be improved significantly.

Equation 6.1

$$CD_{pred} = \int p(\xi) CD(\xi) d\xi$$

Here CD(ξ) represents the CD spectrum corresponding to the structure at ξ on the PMF and p(ξ) represents the percentage of that structure in the solution, which is the Boltzman weight. If dihedral angle analysis is used to compute CD(ξ), the CD will only be defined for structures centered at $\xi=\xi_i$. However, if another method is used to approximate the CD of the structure at ξ, such as quantitatively analyze all structures with radius of gyration ξ from all simulations and then compute the average native and coil percentage for the structures, the computed integral will become continuous.

Finally, to reduce the differences between the predicted and the measured CD spectra it is necessary to make the right choice in deciding which random coil spectrum should be used. Figure 6.2 showed that there is a significant difference between different random coil spectra depending on the peptide used and the degradation method. Further studies are recommended to investigate the differences in the structures which will help in deciding how to choose the appropriate coil spectrum.

6.4 Conclusions

While the process of collagen biosynthesis is quite well understood, the process involved in collagen degradation remains largely unknown. It is known that most proteins have a well defined structure that needs to be preserved for its proper function, and aberrations in a protein's structure can often lead to disastrous results such as illnesses like the Mad Cow Disease, Alzheimer's Disease, and many others. Collagen is no exception to this rule. Collagen degradation has been associated with many diseases including arthritis, tumor metastasis, emphysema, and atherosclerosis. The understanding of collagen degradation is therefore of great importance.

Unlike most globular proteins however, the triple helical structure of collagen does not have a clearly marked region or cleavage site to interact with other proteins. In fact, it was believed that the triple helix is a stable structure that is difficult or nearly impossible for a proteinase to bind to and degrade under normal conditions. Fairly recent studies have shown however that under certain circumstances, collagen can be degraded by specific collagenases at specific spots on the triple helical structure. The method in which collagenases bind to and degraded collagen was understood poorly.

In this research we evaluate the stability of collagen-like peptides by using peptides from specific regions of collagen type III. We use molecular dynamics to compute the potential of mean force of the collagen-like peptides, then estimate the triple helical content of the peptide and generate a predicted CD spectrum. The computational result is then compared with the experimental results from Circular Dichroism. Similar studies were done on mutant collagen peptides from the Ehlers-Danlos Syndrome type IV. While the results from this study are unlikely to directly contribute to novel therapies for collagen-related diseases, we hope that this study can be used to provide a better understanding of collagen and collagen-like peptides.

References

Andrade, M. A., 1993, Evaluation of secondary structure of proteins from UV circular dichroism using an unsupervised learning neural network. *Prot. Eng.* **6**:383-390.

Berndt, K. D., 1996, Circular dichroism spectroscopy, World Wide Web, <http://pps98.man.poznan.pl/ppscore/section8/ss_96016.htm>.

Bhatnagar, R. S., Pattabiraman, N., Sorensen, K. R., Langridge, R., MacElroy, R. D., and Renugopalakrishnan, V., 1988, Inter-chain proline:proline contacts contribute to the stability of the triple helical conformation, *J. Biomol. Struct. Dynam.* **6**:223-233.

Brooks, C. L., and Karplus, M., 1983, Deformable stochastic boundaries in molecular dynamics. *J Chem Phys*, **79(12)**:6312-25.

Brooks, C. L., Karplus, M., and Pettitt, B. M., 1983, Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics. *John Wiley & Sons, Inc*, 1988.

Brunger, A., Brooks, C. L., and Karplus, M., 1984, Stochastic boundary conditions for molecular dynamics simulations of ST2 water. *Chem Phys Lett*, **105(5)**:495-500.

Celentano D. C., and Frishman W. H., 1997, Matrix metalloproteinases and coronary artery disease: A novel therapeutic target. *J. Clin. Pharm.* **37(11)**:991-1000.

Fan, P., Li, M., Brodsky, B., and Baum, J., 1993, Backbone dynamics of $(Pro\text{-}Hyp\text{-}Gly)_{10}$ and designed collagen-like triple-helical peptide by 15N NMR relaxation and hydrogen-exchange measurements. *Biochemistry.* **32**:13299-13309.

Feng, Y., Melacini, G., Taulane, J. P., and Goodman, M., 1996, Acetyl-terminated and template-assembled collagen-based polypeptides composed of Gly-Pro-Hyp sequences. 2. Synthesis and conformational analysis by circular dichroism, ultraviolet absorbance, and optical rotation. *J. Am. Chem. Soc.* **118**:10351-10358.

Fields, G. B., 1991, A model for interstitial collagen catabolism by mammalian collagenases." *J. theor. Biol.* **153**:585-602.

Fratzl, P., Misof, K., Zizak, I., Rapp, G., Amenitsch, H., Bernstorff, S., 1998, In-situ synchrotron x-ray scattering of the tensile properties of collagen, *ELLETRA News and ELETTRA Highlights.*

Gaziano, J. M., 2001, Gobal Burden of Cardiovascular Disease. In Heart Disease. Braunwalkd, E., Zipes, D.P., Libby, P., editors. Saunders, Philadelphia. 1-18.

Greger, M., 2001, The official mad cow disease home page, World Wide Web, <http://www.mad-cow.org/00/image_archive.html>.

Gordon, M. K., and Olson, B. R., 1990, The contribution of collagenous proteins to tissue specific matrix assemblies, *Curr. Opin. Cell Biol.* **2**:833-838.

Hata, T. S., 2001, Viewing macromolecules: a Chime tutorial. World Wide Web, <http://www.mybiology.com/chime/>.

Jacenko, O., Olsen, B. R., and LuValle, P., 1991, Organization and regulation of collagen genes, *Crit. Rev. Eularyot. Gene Express.* **1**:327-353.

Kramer, R. Z., Bella, J., Mayville, P., Brodsky, B., and Berman, H. M., 1999, Sequence dependent conformational variations of collagen triple-helical structure. *Nature Struc. Biol.* **6**:454-457.

Kumar, S., Bouzida, D., Swendsen, R., Kollman, P., and Rosenberg, J., M., 1992, The weighted histogram analysis method for free-energy calculations on biomolecules. *J. Comp. Chem.*, **13(8)**:1011-1021.

Kumar, S., Bouzida, D., Swendsen, R., Kollman, P., and Rosenberg, J., M., 1995, Multidimensional free-energy calculations using the weighted histogram analysis method. *J. Comp. Chem,* **16(11)**:1339-1350

Leikina, E., Mertts, M. V., Kuznetsova, N., and Leikin, S., 2002, Type I collagen is thermally unstable at body temperature. *PNAS.* **99(3)**:1314-1318.

McDonnell, S., Morgan, M., and Lynch, C., 1999, Role of matrix metalloproteinases in normal and disease processes. *Biochemical Society Transactions.* **27(4)**:734-740.

Roux, B., 1995, The calculation of the potential of mean force using computer simulations. *Comp. Phys. Comm.*, **91**:275-282

Rupp, B., 2005, Circular dichroism spectroscopy, World Wide Web, <http://www-structure.llnl.gov/cd/cdtutorial.htm>.

Souaille, M., and Roux, B., 2001, Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations. *Comp. Phys. Comm.,* **135**:40-57.

Stultz, C. M., 2002, Localized unfolding of collagen explains collagenase cleavage near imino-poor sites. *J. Mol. Biol.* **319**:997-1003.

Stultz, C. M., and Edelman, E. R., 2003, A structural model that explains the effects of hyperglycemia on collagenolysis. *Biophys. J.* **85**:2198-2204.

Tromp, G., Kuivaniemi, H., Shikata, H., and Prockop, D., 1989, A single base mutation that substitutes serine for glycine 790 of the α1(III) chain of type III procollagen exposes an arginine and causes Ehlers-Danlos Syndrome IV. *J. Bio Chem.* **264(3)**:1349-1352.

Tromp, G., Paepe, A. D., Nuytinck, L., Madhatheri, S., and Kuivaniemi, H., 1995, Substitution of valine for glycine 793 in type III procollagen in Ehlers-Danlos Syndrome type IV. *Human Mutation*, **5**:179-181.

Usha, R., and Ramasami, T., 2004, The effects of urea and n-propanol on collagen degradation: using DSC, circular dichroism and viscosity. *Therm. Acta,* **409:**201-206.

Wang, C. G., 2002, Human lysyl hydroxylase isoforms: multifunctionality of human LH3 and the amino acids important for its collagen glycosyltransferase activities. World Wide Web, <http://herkules.oulu.fi/isbn9514267990/isbn9514267990.pdf>.

Yang, J. T., Wu, C. S. C., and Martinez, H., 1986, Calculation of protein conformation from circular dichroism. *Meth. Enzymology*, **130**:208-268.

Appendix

General MD Procedures and Functions

Reading and Writing Files

At the beginning of every CHARMM script, a number of files are usually loaded into memory before any useful computation can be done. These include the topology file, parameter file, and often the coordinate file and protein structure file. The topology and parameter files are common for all scripts and all proteins. The topology file specifies how atoms interact with each other within a residue, and contains information such as the charge, mass, dihedral angles, improper dihedral for each atom or residue, as well as how atoms are bonded in a residue. The parameter file is a database that specifies values for other parameters required to calculate the potential energy, such as bond length, bond angles, parameters needed to calculate Van der Waals forces, and so on.

Often however the topology and parameter files do not provide enough specific information about the sample. Such information is provided in the coordinate file and protein structure file. The coordinate file contains the 3-D coordinates for each atom of the system under investigation, while the protein structure file describes how and which amino acids are linked in the protein. Below is an example of reading in the topology, parameter, coordinate, and protein structure files.

```
**********************************************************************
!     GET COMMON TOPOLOGY FILE
      open read unit 11 card name -
      /home/fcyang/common/top2.inp
      read rtf unit 11 card
      close unit 11 card

!     GET COMMON PARAMETERS
      open read unit 15 card name -
      /home/fcyang/common/par-ace.inp
      read para unit 15 card
      close unit 15

!     READ SYSTEM PSF
      open unit 1 read card name col3ip2sol3.psf
      read psf card unit 1
      close unit 1

!     READ SYSTEM CRD
      open unit 1 read card name @col3.crd
      read coordinate card unit 1
      close unit 1
```

```
************************************************************************
```

Before reading each files, CHARMM allocates some unit in memory and specifies the type of file (card is a CHARMM file) that is read. For example, before reading the topology file top2.inp, unit 11 was allocated, and top2.inp was read as rtf (read topology file). Similarly, the parameter par-ace.inp was read in as para (parameter file), col3ip2sol3.psf was read in as psf (protein structure file), and @col3.crd was read in as coordinate file where @col3 recalls a variable defined earlier in the script.

The same principle follows when a file is written. File type and name must be specified, and a unit is allocated temporarily:

```
************************************************************************
!     WRITE COORDINATES
      open unit 1 write formatted name @j.crd
      write coordinate card unit 1
*     MD OF COLLAGEN
*
      close unit 1

!     WRITE COORDIATES -PDB
      open unit 1 write formatted name @j.pdb
      write coordinate pdb unit 1
      close unit 1
************************************************************************
```

Non-bonded Specifications

Besides specifying the interactions between bonded interactions in a system, a full molecular dynamics simulation require the specifications of non-bonded interactions such as electrostatic forces and Van der Waals forces. Theoretically speaking however, the interactions between two non-bonded atoms exist as long as the distance between the two atoms is finite. Therefore, the number of real non-bonded interactions is equal to the combination of the number of atoms in the system N of size 2, or $\binom{N}{2} = \frac{N(N-1)}{2}$, which is a huge number since the total number of atoms in the system N is large. The non-bonded specification is therefore needed to specify a cutoff distance for the calculation of non-bonded interactions; if the distance between two atoms is larger than the cutoff, the non-bonded interaction between the two atoms is assumed to be zero, therefore reducing the total number of calculations for molecular dynamic simulations significantly. Below is an example of specifying the cutoff distances for non-bonded interactions in nbond.spec:

```
************************************************************************
NBONd -
     CUTNb 13.0 CTOFnb 12.0 CTONnb 10.0 -
```

```
    VSWItch SHIFt E14Fac 1.0 -
    CDIElectric -
    WMIN 1.0 ! BYCUbe
**********************************************************************
```

In many situations where the system or the number of calculations is large, such as for molecular dynamics using periodic boundary conditions, it is recommended to reduce the cutoff distance even further to decrease computational time. Below is an example from the nbond10.spec script used for PBC non-bonded specification. Note the reduction in the cutoff distances.

```
**********************************************************************
NBONd -
    CUTNb 10.0 CTOFnb 9.0 CTONnb 8.0 -
    VSWItch SHIFt E14Fac 1.0 -
    CDIElectric -
    WMIN 1.0 ! BYCUbe
**********************************************************************
```

Shake

On the atomic scale, there are many degrees of freedom in additional to the three dimensional XYZ translations, such as atomic vibrations, bond stretching, bond rotation, etc. Frequently, the fastest and most frequent motions in the system are not those directly involved in the reaction, but the bond stretching terms, particularly that of bonds to hydrogen. These motions determine the timestep of the simulations—ideally we would want one timestep for every single change in the system. Unfortunately, the motion on the bonds of hydrogen is extremely frequent yet unimportant, making the timestep unnecessarily small. To avoid huge simulation time, the shake command is used prior to every molecular dynamic simulation to limit the high frequency motions of the bonds to hydrogen by applying Holonomic constraints and making them rigid. The bonh flag indicates that the constraint is applied to bonds to hydrogen only.

```
**********************************************************************
!     SETUP SHAKE
      shake bonh tolerance 1.0e-6 parameter
**********************************************************************
```

Minimization Algorithms

In many cases, the structure of the peptide from crystallography is not the global or even local energy minimum. Many of them will contain "bad contacts" such as when the

distance between atoms are too small, leading to a high energy. If molecular dynamics is run starting from these structures, the simulation might be very unstable due to the high energy, and a huge amount of computational time is required to reach equilibrium. A good way to avoid this is using energy minimization algorithms, which are good ways to "relax" a structure before molecular dynamics is done.

There are a number of minimization algorithms available in CHARMM, but the results should be similar. Two minimization algorithms were used in our study: steepest descent and conjugate gradient. Steepest descent is an easy way of minimizing the energy by following the gradient of the potential. However due to its simplicity, the algorithm may get trapped at some local minimum or oscillate endlessly. Conjugate gradient is a more complicated algorithm in which previous steps of minimization is accounted in order to solve such problems. Generally it is enough to run about 100 steps of steepest descent, followed by few hundred steps of conjugate gradient:

```
**********************************************************************
!     MINIMIZE THE NEW COORDINATES
      minimize sd nsteps 100
      minimize conj nsteps 500
**********************************************************************
```

It is also important to keep in mind that minimization algorithms generally can not find the global energy minimum; global energy minimum usually requires full molecular dynamics simulation over a range of reaction coordinate, such as when using umbrella sampling to do MD under different radius of gyration. If too much energy minimization is done, there is some chance that the structure will be trapped in a deep local minimum, which might be hard to get out of. As a result, enough energy minimization should be done to ensure a stable starting structure with little or no bad contacts, but overdoing energy minimization would not be a good thing either.

Residue Patching

Before we run the simulation for any mutants, we need to express to CHARMM the new sequence and initial coordinates of the mutant. This is done through a process named residue patching. Residue patching starts with a raw protein structure file (the imino-poor peptide in col3.psf), and modifies the mutated residues and generates a new protein structure file. Specifically, PRES (patch residue) commands are used in the topology file top2.inp to specify how to change the existing peptide into the mutant peptide by modifying select residues. For example, if the mutation substitutes an hydroxyproline for a threonine, the following changes are needed:

```
**********************************************************************
PRES T2O     0.00000   ! threonine to hydroxyproline
DELETE ATOM H
```

```
DELETE ATOM OG1
DELETE ATOM HG1
DELETE ATOM CG2
GROU
ATOM N   N     -0.20
ATOM CD  CH2E  0.10
ATOM CA  CH1E  0.10
GROU
ATOM CB  CH2E  0.00
ATOM CG  CH1E -0.65
ATOM OG  OH1   0.25
ATOM HG  H     0.40
BOND N  CA     CA  C      C  +N     C  O      N  CD
BOND CA CB     CB  CG     CG CD     CG OG     OG HG
DIHE -C  N  CA  C      N  CA  C  +N     CA  C  +N  +CA
DIHE N  CA  CB  CG     CA  CB  CG  CD     CB  CG  CD  N
DIHE CG  CD  N  CA
DIHE N  CD  CG  OG      CB  CG  OG  HG
IMPH N  CA  CD  -C     C  CA  +N  O      CA  N  C  CB
IC  -C  CA  *N  CD   0.0000  0.00 180.00  0.00  0.0000
IC  -C  N   CA  C    0.0000  0.00 -60.00  0.00  0.0000
IC  N   CA  C   +N   0.0000  0.00 180.00  0.00  0.0000
IC  +N  CA  *C  O    0.0000  0.00 180.00  0.00  0.0000
IC  CA  C   +N  +CA  0.0000  0.00 180.00  0.00  0.0000
IC  C   N   *CA CB   0.0000  0.00 -120.00 0.00  0.0000
IC  N   CA  CB  CG   0.0000  0.00  0.00 108.00  0.0000
IC  N   CD  CG  OG   0.0000  0.00 180.00  0.00  0.0000
IC  CB  CG  OG  HG   0.0000  0.00 180.00  0.00  0.0000
*******************************************************************
```

PRES specifies the current residue is a patch residue named T2O, which can be used to modify threonine to hydroxyproline if the patch command is called later on. DELETE specifies the atoms to be deleted from the threonine template, and select atoms are then added through ATOM. The GROU command groups atoms such that the net charge in each group is 0. BOND, DIHE, IMPH, and IC specifies the detailed interactions between the atoms, including bonds, dihedrals, improper dihedrals, respectively.

However by only specifying the patch residues in the topology file, the protein structure file is unchanged. To generate the mutant protein structure file and its initial coordinates, we use the original protein structure and coordinate files as template to first change (patch) select residues, then run minimization algorithms by fixing everything but the side group of mutant residues to find the initial optimal coordinates for the side groups.

After loading the topology file, parameter file, protein structure file, and coordinate file as usual, the patch command is executed to modify selected residues. For example, in the

case of threonine to hydroxyproline in IP2, the 10^{th} residue on chain A and 11^{th} residue on chain B and C are patched as follows:

```
patch T2O A 10 setup
patch T2O B 11 setup
patch T2O C 11 setup
```

Some IC commands are then used to generate values for missing or undetermined Cartesian coordinates:

```
ic fill preserve    ! fills internal coor table
ic parameters       ! fills in missing values
ic build            ! computes cart coor for all undetermined values
```

The modified residues are next renamed:

```
define hyp1 select segid A .and. resid 11 end
define hyp2 select segid B .and. resid 12 end
define hyp3 select segid C .and. resid 12 end
```

All atoms besides the side chain atoms of the mutant residues are then fixed, and MD is run. Side chain atoms are defined as any atom that is not nitrogen, oxygen, general carbon, or carbon alpha atoms.

```
constrain fix select .not. (mutants .and. sidechain) end
```

Non-bonded specifications are then streamed from nbond.spec, and minimization using Steepest Descent and Conjugate Gradient algorithms are done to minimize the energy by adjusting the coordinates of unfixed atoms. Finally, the new coordinates and protein structure files are written for the mutant peptide, as well as the pdb file format for view in VMD.

The Solvation Process

Solvation is an important process used to simulate the surroundings of the collagen peptides. While the realistic environment and solvent for collagen *in vitro* and *in vivo* are much more complex, water molecules are used *in silico* for simplification purposes. The process starts with the creation of a large densely packed water cube: initial water cubes are overlaid on top of each other, deleting overlaps, and then equilibrated. The process is repeated at least three times to generate a large water cube that is densely packed, which can be used for many applications in different systems. The final water cube created had a width of 60A and contained over 7000 water molecules in total, as shown in the Figure A.1.

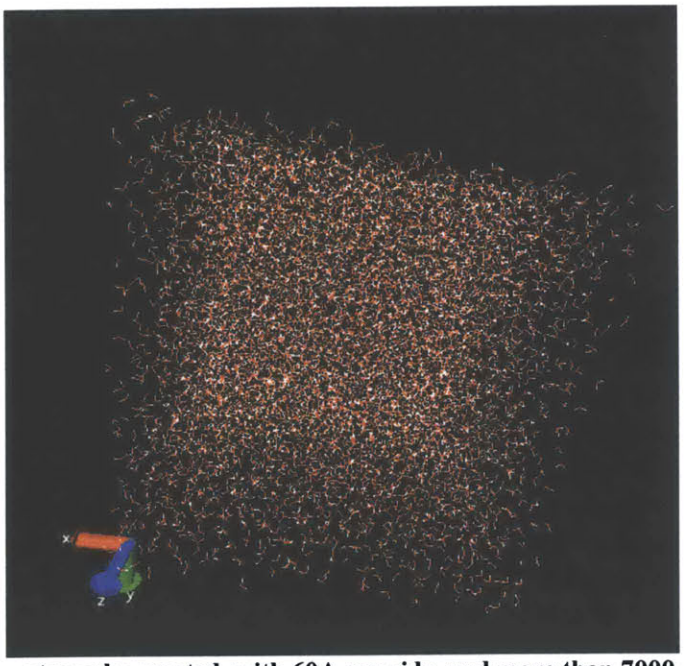


Figure A.1 The final water cube created, with 60A per side, and more than 7000 equilibrated TIP3 water molecules.

Next, the coordinate and protein structure files of the peptides are loaded into memory. For native peptides such as imino-poor, crystallographic data exists and is used here after energy minimization. If the peptide is a mutant, such as IP2, mutant residues will be patched starting from the native structure and energy minimization is done to create the starting structure. The water cube is then loaded into memory, less the water molecules that overlap with the peptide, and equilibrated while holding the peptide fixed. This process is repeated at least three times to ensure that there are enough water molecules in the system, and coordinate and structural files for the final system with peptide solvated in water are saved.

Periodic Boundary Condition Setup

The central cell in PBC is first constructed as an approximately 42A x 42A x 99A box that contains the collagen peptide and filled with water molecules. The PBC requires duplicate images of the central cell being set up in its immediately three dimensional environment. These images are therefore set up by copying and translating the central cell to 26 positions adjacent to each face and corner of the central cell. The images are set up in pairs, each pair containing a new translated image and its inverse, such as the translation pair (42, 0, 0) and its inverse (-42, 0, 0). The positive translations are named using X, Y, and Z, while the negative translations are named using A, B, and C, as shown in the below segment of the CHARMM commands that set up three images and their

inverse images (6 images in total). Note that the inverse images can be defined by either specifying the exact translations, or using the "define inverse" command.

```
*****************************************************************
     IMAGE X
     TRANS   42   0   0
     IMAGE A
     TRANS  -42   0   0

     IMAGE XY
     TRANS   42  42   0
     IMAGE AB
     TRANS  -42 -42   0

     IMAGE Z
     TRANS    0   0  99
     IMAGE C
     define inverse Z

...

     END
*****************************************************************
```

Next, the frequency at which the periodic boundaries are updated is specified using "update imgfrq." The "shake" command is performed normally to specify hydrogen bond tolerances, and nonbonded interaction is specified by streaming the specifications in nbond10.spec. Finally MD is set up to run after fixing one residue near the center of collagen peptide and setting up radius of gyration constraints.

Stochastic Boundary Condition Setup

The stochastic boundary condition is set up using a number of input files, each of which is like a function and performs a certain role. The coordinate and structure files for the solvated system is first read into memory, and then the system is partitioned into reaction region, buffer region, and reservoir region. Each region is identified with a vector, for example, if the residue belongs to the reaction region, vector x will be 1 instead of 0.

```
*****************************************************************
!    PARTITION
     define water select segid bulk end
```

```
      define reac  select .byres. ((point 0. 0. 0. cut 30.0) .or. water) end
      define buff  select .not. (reac) end

      scalar x set 0.0
      scalar y set 0.0
      scalar z set 0.0

      scalar x set 1.0 select ( reac ) end
      scalar y set 1.0 select ( buff ) end
!     scalar z set 1.0 select ( rese ) end

!     FOR USE IN PARTITIONING SYSTEM
      open unit 1 write formatted name @j.reg
      write coordinate card unit 1
*****************************************************************
```

Harmonic constraints are then set up for the system, with evcrything inside the reaction region getting harmonic constraint scaled by 0 and everything outside the reaction region getting scaled by 1.

```
*****************************************************************
!     BEGIN { assign friction coefficients to protein and solvent atoms }
      scalar ycomp set  62.0 select ( solv .and. type oh2 ) end
      scalar ycomp set 250.0 select ( .not. (hydrogen .or. solv) ) end

!     BEGIN { assign average B-factors to atoms }
      define nonh select ( .not. hydrogen ) end
      define back select ( type n .or. type ca .or. type c ) end
      define beta select ( type o .or. (type *B#) ) .and. nonh show end
      define gamm select ( type *G# ) .and. nonh show end
      define delt select ( type *D# ) .and. nonh show end
      define rest select ( .not. ( back .or. beta .or. gamm .or. delt ) ) -
               .and. nonh .and. .not. type OH2 show end

!     ASSIGN AVERAGED B-FACTORS TO ATOMS
      scalar wcomp set  0.0
      scalar wcomp set 14.0 select ( back ) end
      scalar wcomp set 15.0 select ( beta ) end
      scalar wcomp set 16.0 select ( gamm ) end
      scalar wcomp set 17.0 select ( rest .or. delt ) end

!     BEGIN { convert B-factor to force-constant }
!
! notes on the conversion:
! (Kb = Boltzmann's constant, T = temperature, K = force-constant,
!  B = debye-waller factor)
```

```
!
! dr^2        = 3 B / ( 8 pi^2 )
! omega^2     = 3 Kb T / (m dr^2 )
! F(restraint)  = -m omega^2 [ x(t) - xref ]
! =>
!  K         = m omega^2
!            = 3 Kb T / ( dr^2 )          substituting omega^2
!            = (8 pi^2 Kb T) / B          substituting dr^2
!
! and as the CHARMM HARMonic (EXP 2)  restraint term multiplies the assigned
! force-constant by a factor of 2 in the derivative we simply
! halve the value.
! => K         = (4 pi^2 Kb T) / B

      set k ?KBLZ
      multiply k by @t
      set p ?PI
      multiply p BY ?PI
      multiply p BY 4.0
      multiply k BY @p

      scalar wmain = wcomp          ! restore B in Wmain
!     AVOID DIVIDE BY ZERO ON THE RECIPROCAL STATEMENT
      scalar wmain reciprocal select (property wmain .lt. 0.00001) show end
      scalar xcomp = wmain          !
      scalar xcomp multiply @k      ! perform the conversion
!     END { convert B-factor to force-constant }


!     INITIALIZE SCALE FACTORS
      scalar zcomp set 0.0  ! initialize

!     SET SCALE FACTORS FOR EVERYTHING NOT IN REACTION REGION TO
1.0
      scalar zcomp set 1.0 select (rese .or. buff) end

      scalar zcomp store 1
      scalar xcomp *store 1
 ***************************************************************
```

The stochastic potential near the boundary is then streamed using existing specifications depending on the boundary size. An unknown error however occurred when applying the harmonic constraint, in which if harmonic constraint was applied to the buffer region, the same was applied to the reaction region despite of the scaling factor of 0, therefore limiting the movement of atoms within the reaction region. This was dealt with by fixing the buffer region completely and not using harmonic constraint at all, which we call a

"stochastic-like" approximation to the stochastic boundary condition. In normal stochastic boundary condition, the reservoir region is fixed, the buffer region is constrained by harmonic constraints which allows very little change within the buffer region, and the reaction region undergoes full molecular dynamics simulation. In the stochastic-like boundary condition, both the reservoir and buffer regions are fixed, and the reaction region undergoes full molecular dynamics simulation. Despite of the differences, molecular dynamics shows that in most situations the movement within the buffer region under stochastic boundary condition is extremely limited, therefore a fixed buffer region could give a good approximation to the stochastic boundary condition.

Dihedral Angle Calculations

By calculating the dihedral angles for the structure at certain radius of gyration, we can estimate the percentage of the native and coil conformation in the structure, where the percentage of coil conformation is approximated by the percentage of dihedral angles that lies outside 3 standard deviation of the average crystallographic IP structure dihedral. Together with the CD spectra for the native and coil states, we can then estimate the CD spectra for the specific structure at some radius of gyration.

In order to calculate the dihedral angle for a structure at certain radius of gyration, the protein structure and coordinate files for the structure is first loaded into memory, and the dihedral angle for one chain (chain A) is generated like below:

```
**********************************************************************
!     LOAD PROTEIN
      open unit 1 read card name -
      /home/fcyang/stochastic/md/chaina190.crd
      read coordinate card unit 1
      close unit 1

!     PRINT THE IC TABLE
      ic generate select segid A end
      ic fill
      print ic
**********************************************************************
```

The output will be similar to below. It contains information about the atoms involved for each angle, and others.

```
**********************************************************************
  N   I     J     K     L  R(I(J/K)) T(I(JK/KJ)) PHI   T(JKL)  R(KL)
  1   1 CA   1 HT1   1 *N   1 CD    1.4647  110.82  119.53  109.35  1.4525
```

```
  2  1 CD   1 HT1  1 *N   1 HT2   1.4525 109.35 121.19 110.05  1.0451
  3  1 HT1  1 N    1 CA   1 C     1.0381 110.82 -123.63 111.00  1.5290
  4  1 C    1 N    1 *CA  1 CB    1.5290 111.00 -119.10 105.41  1.5188
  5  1 N    1 CA   1 CB   1 CG    1.4647 105.41  27.25 103.14  1.5039
  6  1 CD   1 CB   1 *CG  1 OG    1.5175 100.99 -121.72 114.37  1.4397
  7  1 CB   1 CG   1 OG   1 HG    1.5039 114.37  48.87 116.11  0.9694
  8  1 N    1 CA   1 C    2 N     1.4647 111.00 -178.90 120.04  1.3333
  9  2 N    1 CA   1 *C   1 O     1.3333 120.04 176.76 118.13  1.2297
 10  1 CA   1 C    2 N    2 CA    1.5290 120.04 -172.93 121.69  1.4583
*******************************************************************
```

The above output is simply reformatted to show only the dihedral angles and the atoms involved in each angle:

```
*******************************************************************
1 CA HT1 *N CD 119.53
2 CD HT1 *N HT2 121.19
3 HT1 N CA C -123.63
4 C N *CA CB -119.10
5 N CA CB CG 27.25
6 CD CB *CG OG -121.72
7 CB CG OG HG 48.87
8 N CA C N -178.90
9 N CA *C O 176.76
10 CA C N CA -172.93
*******************************************************************
```

Since we are only interested in the backbone dihedral angles between N-CA-C-N (Ψ angle) and between C-N-CA-C (Φ angle), all other angles are discarded. Finally, the individual Ψ and Φ angles are compared to the average Ψ and Φ angles of the crystallographic structure plus and minus three standard deviations. If either or both of the individual Ψ and Φ angles fall outside the three standard deviations bound, the residue is counted towards percentage of coil, otherwise it is counted towards the percentage of native. In this way, the percentage of native and coil can be estimated to give us the approximate CD spectra of the structure at any radius of gyration.

PMF Patching

PMF patching is one method we used to construct the overall PMF of a sample along some reaction coordinate by piecing together small PMF from each simulation of Umbrella Sampling. A general description of PMF patching is described in Chapter 4. A detailed description of the four functions used in the PMF patching algorithm written in Matlab is included below.

calc_pmf_general

In calc_pmf_general.m, the filenames and their directories are first specified. Then starting from the leftmost radius of gyration simulation window, calc_pmf_window.m and calc_pmf_slope.m are called to calculate the pmf value, radius of gyration (rg) range, and the slope of the first simulation window. Note that the slope vector will contain one less element than the pmf and rg vector, since it takes two points to calculate a slope. For example, if num_bin=10, then calc_pmf_window.m returns a vector of length 10 containing the pmf values, and a vector of length 10 containing the rg values corresponding to each of the pmf values (the middle rg value in each bin is used), while the calc_pmf_slope.m returns a vector of length 9 containing the slopes between each neighboring pair of pmf and rg values. The current pmf, rg, and slope are then moved into the prev_patch, prev_rg, and prev_slope vectors, while the values for a new simulation window is calculated using calc_pmf_window and calc_pmf_slope. Next, the patch_pmf.m subprogram is called to make a patch between prev-values and the values from the new simulation window. The result is stored back into prev-value vectors. The new values for the next simulation window are then calculated, and then patched to the prev-value vectors again. This procedure is repeated until all simulation windows are patched. The final pmf and its corresponding rg values are then reported from calc_pmf_general as outputs.

calc_pmf_window

The calc_pmf_window.m subprogram takes in the rg data from the current simulation window, the temperature in K, the rg of the current simulation window, and the number of bins used, and outputs the pmf values and corresponding rg values for the simulation window. Long type variable are used to enable Matlab with higher degree of accuracy in the calculations. The maximum and the minimum rg value are first determined, then the range between them is divided into num_bin equal-interval bins. The number of structures in each bin is counted, and the probability of each bin is calculated (probability = number of counts in bin / total number in simulation window). This is the constrained probability, which is then converted to the constrained (hybrid) energy function using `hybrid_e=-1*k*T*log(hybrid_prob)`, where k is the Boltzman's constant in kcal/mol/K, T is temperature in K, and log(hybrid_prob) is the natural log of the constrained probability. From here, bias potential is calculated using the form $V=A(RG-RG_i)^2$, where A is a constant (500), RG is the rg of current simulation window, and RG_i is the rg of individual structures, which is approximated using the middle rg value of the bin it falls into. Knowing the bias potential and the hybrid energy, we can then find the unconstrained energy (pmf) with some constant offset by subtracting the bias potential from the hybrid energy. The corresponding rg values for each pmf are approximated using the middle value of each bin.

calc_pmf_slope

This function simply takes in the pmf and rg vectors for a simulation window, and calculates the corresponding slope vector. For example, the first slope element is calculated using: slope(1)=(pmf(1)-pmf(2))/(rg(1)-rg(2)), the second is calculated using: slope(2)=(pmf(2)-pmf(3))/(rg(2)-rg(3)), and the ninth is calculated using: slope(9)=(pmf(9)-pmf(10))/(rg(9)-rg(10)). This way, the function returns a nine element slope vector for a simulation window with bin size of ten.

patch_pmf

This subprogram takes in the patched values so far (stored in prev_patch, prev_rg, and prev_slope), and the current values being patched (stored in pmf_win, rg_win, and pmf_slope), and returns the newly patched vectors of pmf, rg, and slope. The first element of the current rg vector is compared with elements of the prev_rg vector starting from its last element and then the previous values in descending order (leftwards), until the value of the first element of current rg is bounded between two neighboring elements of the prev_rg. The prev_rg region to the right of the left bound value is defined as the overlap (Figure A.2).

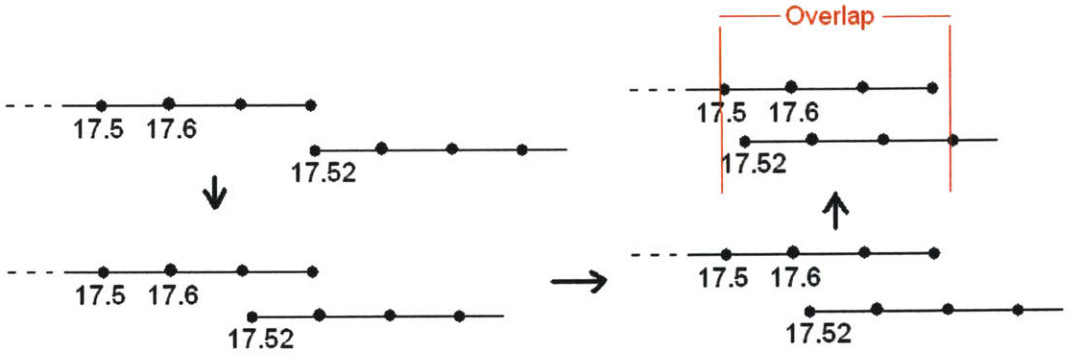


Figure A.2. Method of identifying the overall overlap by moving the windows until they line up.

Once the overlap region is identified, we need to identify at which point in the overlap the patching starts. This is done by comparing either the point slope differences or the average squared slope differences between the possible overlap regions. For the point slope difference, the slopes between each region are compared (see Figure A.3, region 1 is compared with region 1, and so on), and the patching starts from the left point of the region where the slope difference is smallest.

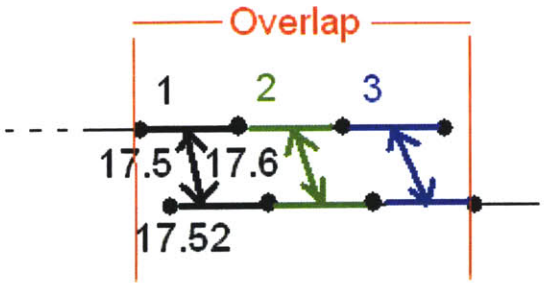


Figure A.3. Comparing the slopes of the overlap region, either point slope difference or average slope difference can be used.

Average overlap patching picks the smallest among A, B, C (calculated below), where if A is picked, the patching starts from the left point of region 1, if B is picked, the patching starts from the left point of region 2, and so on.

$A=\text{average}[(\text{current\_rg\_slope(region 1)} - \text{prev\_rg\_slope(region 1)})^2 + (\text{current\_rg\_slope(region 2)} - \text{prev\_rg\_slope(region 2)})^2 + (\text{current\_rg\_slope(region 3)} - \text{prev\_rg\_slope(region 3)})^2]$

$B=\text{average}[(\text{current\_rg\_slope(region 2)} - \text{prev\_rg\_slope(region 2)})^2 + (\text{current\_rg\_slope(region 3)} - \text{prev\_rg\_slope(region 3)})^2]$

$C=(\text{current\_rg\_slope(region 3)} - \text{prev\_rg\_slope(region 3)})^2$

Unless otherwise stated, the point slope difference method is used. Once we have decided at which point to patch, we need to add to the current pmf values a correct offset, such that the pmf will be continuous at the point of patching. This is done by shifting the current_pmf by -X+Y units (where X is the current pmf value at the point of patching, and Y is the previous pmf value immediately to the left of the point of patching), and then some additional offset value to account for the difference between the rg values of prev_rg and current rg. In the example below (Figure A.4), 17.75 is identified as the point of patching on current simulation window by slope and rg comparison. The current pmf is then shifted down by 5 units (where 5 is value of current pmf at point of patching), up by 2 units (where 2 is the value of prev_pmf at point right before patching), and then finally offset by (17.75-17.7)*slope amount to correct the difference between 17.7 and 17.75. Here the slope used can be either the slope to the right of the patch point in current pmf, or the slope immediately to the left of the patch point on prev_pmf. Depending on the choice of slope used, different final pmf might be produced. Unless otherwise specified, choice 1 is used for consistent results with Ramon Salas Escat's program.

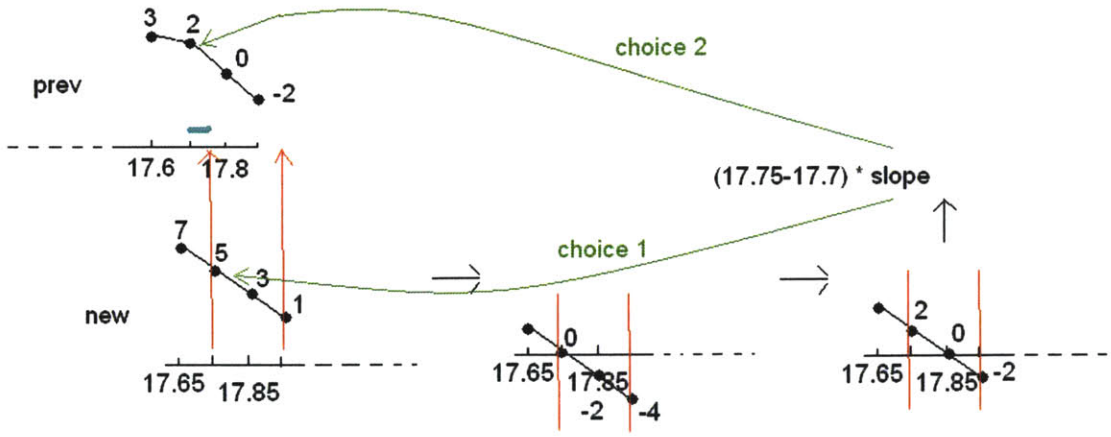


Figure A.4. The patching and interpolating processes.

Finally, the range starting from immediately left of the patching point to the far right end of the previous vectors are deleted, while the range starting from the patching point to the far right of the current vectors are added, as shown in the below figure.

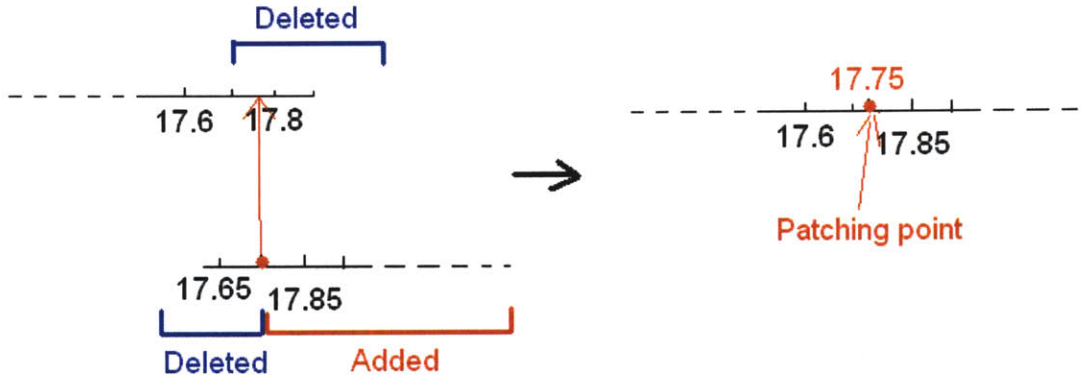


Figure A.5. The final stage of the patching process, where the old segment is partially deleted and the new segmented is added.

calc_pmf_general2

This program serves the same function as calc_pmf_general.m by loading in the radius of gyration data files and calling the calc_pmf_win, calc_slope, and patch_pmf programs to generate the final pmf. The only difference is that calc_pmf_general2 reads in an additional .txt file, which contains the file name for all radius of gyration data files, as well as RG each simulation is done at, instead of generating the filenames using strcat like calc_pmf_general.m. By reading an external .txt file, we now have the ability to easily add additional simulation windows to the patching, and easily load in any kind of filename instead of only filenames with certain pattern. calc_pmf_general2.m is therefore

a more general and useful form of calc_pmf_general.m. The format of the external .txt file is as follows:

filename 1
simulation rg 1
filename 2
simulation rg 2
...

WHAM

The WHAM program is implemented in Matlab using three functions, wham_general.m, divide_rg.m, and wham_iteration.m. The wham_general function is the main WHAM program. It first finds the overall maximum and minimum reaction coordinate for all simulations, and divides the entire reaction coordinate space into bins of fixed width by calling divide_rg.m. Next, the structures within each simulation window are sorted into these bins to generate one histogram per simulation. These histograms are inputted into the WHAM equations, and wham_iteration.m is called to execute each WHAM iteration by using zero as the initial values of F. This continues until convergence is reached in both equations. From here, the PMF is generated through both the free energy constant F and the unbiased probability distribution, as outlined in Chapter 4. Both PMFs are plotted for correct and smooth view as well as high resolution.

Three versions of using WHAM to generate the PMF were explored. The first version uses a fixed bin width from user input to construct histogram for all simulation windows and then applies WHAM. This is the method described above, with user defined bin width. The second version searches for the smallest bin width possible separately for each simulation window, such that each bin contains at least 10 structures. We will refer to this as WHAM with variable bin width below. The third version is similar to the first version in that it uses a fixed bin width to construct the histogram for all simulation windows, but instead of letting user input the bin width, the program searches for the smallest bin width possible such that all bins in all windows have 10 structures or more. This is referred to as WHAM with fixed but optimized bin width.

The second version of WHAM is coded out of the concern that WHAM with user defined bin width may contain statistical errors due to the fact that some bins may contain less than 10 structures. Contrary to the WHAM with user defined bin width which uses a fixed bin width and therefore fixed number of bins for all simulations, this version of WHAM lets the program to search for a bin width separately for each simulation such that each window will use the smallest bin width possible while guaranteeing at least 10 structures falls into each bin. The resulting histogram for each simulation window is therefore likely to have a different number of bins and defined for different sets of ξ.

However, the WHAM equations require the histograms to be defined at the same reaction coordinate ξ for all windows. To get around this, the resulting histogram for each simulation is interpolated to a fixed set of ξ. This however leads to a different problem: the sum of probabilities within each simulation no longer add up to one due to the interpolation, which leads to the non-convergence of the WHAM equations. The non-convergence can be solved by simply renormalizing the probabilities for each simulation window, but this makes the number of structures in each bin to be non-integral, and the resulting PMF still does not resemble a real PMF in terms of both shape and scale. This is most likely because the real histogram (before interpolation) is different for each window in terms of bin width and the number of bins, so that each histogram is truly defined for a different set of ξ, which violates the WHAM equations that require all histograms to be defined at a fixed set of ξ. Although interpolation can help converting all histograms to be defined at a fixed set of ξ, each histogram still practically only contain information from the original set of ξ that it was defined from, not the new set of ξ.

The third version of WHAM also attempts to get rid of any statistical errors contained in version one by guaranteeing each bin of the histograms contains at least 10 structures. Rather than having a variable bin width for each window like version two, however, version three uses the smallest possible fixed bin width for all windows such that at least 10 structures falls into each bin. In another words, the third version of WHAM starts with a large bin width and checks for all bins in all histograms. If all bins contain at least 10 structures, the common bin width is decreased, and the process repeats until the smallest bin width is reached. Although this seems to be a good idea that minimizes statistical error in version one while not violating the WHAM equations like version two, the resulting common bin width that can guarantee all bins to have at least 10 structures was unacceptably large, causing a tremendous loss of resolution.

Statisticians agree that bin width is the most important parameter for histograms. If the bin width used is too small, too much detail or undersmoothing occurs; if the bin width is too large, too little detail or oversmoothing occurs. Over the years, many algorithms for determining the most optimized bin width to use have been developed. Some of the most popular include:

1.
$n=2^{(N-1)}$, where n=number of total samples and N=number of bins;

2. Scott, 1979, "On optimal and data-based histograms"
$w=3.49*\sigma*n^{-1/3}$, where w=bin width and σ is the standard deviation of the distribution.

3. Izenman, 1991, "Recent developments in nonparametric density estimation"
$w=2*IQR*n^{-1/3}$, which is a more general form of the second algorithm, where IQR=75^{th} percentile – 25^{th} percentile.

However from our research, it seems that none of these algorithms (or any other that we know of) actually works consistently. This makes it very difficult to apply any one particular formula or use any particular bin width for all samples, and justify the reason for choosing it. Because of this, it seems that the next best approach would be to determine the bin widths on a case by case basis, depending on the data. Although the histogram (a visualization of the data) and therefore the resulting PMF will be slightly different depending on the bin width used, the bin width should be chosen such that the resulting PMF gives us the best representation based on our understanding. As a result, bin width used for each peptide was slightly different, depending on the resulting PMF.

```
% PMF PATCHING PROGRAM
% THIS FUNCTION USES THE CALC_PMF_WINDOW.M TO PATCH THE PMF FOR ALL
% WINDOWS TOGETHER BY COMPARING THE SLOPES OF THE PMF_WINDOWS AND JOINS
% THEM WITH APPROPRIATE AMOUNT OF OVERLAP.

% THIS FUNCTION IS DIFFERENT FROM CALC_PMF_GENERAL.M IN THAT IT READS
THE
% FILENAMES AND RG_CURRENT FROM A .TXT FILE INSTEAD OF USING STRCAT.
% THEREFORE IT CAN HANDLE ADDITIONAL POINTS AND IRREGULAR FILENAMES.


% 1 = IP pbc chaina1_232.dat to chaina1_252.dat
% 2 = old IP stochastic, chaina_7rgd.dat to chaina_1rgd.dat, and
chaina1rgd.dat to chaina16rgd.dat
% 3 = IP2 stochastic by fixing buffer region, chaina175.dat to
chaina195.dat.
% 4 = IP stochastic by fixing buffer region, chaina175.dat to
chaina195.dat
% 5 = IP3 stochastic by fixing buffer region, chaina175.dat to
chaina195.dat

function [pmf_overall, rg_overall] = calc_pmf_general2(T, num_bin,
sample);

% note: MUST remove path if same filenames are used in different
directories!!!

% path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files'); % 1
% path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic'); % 2

warning off all;

if sample == 2
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 3
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 3
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 3
        fid=fopen('IP2 stochastic-like 300K.txt','r');      % 3
%     fid=fopen('IP2 stochastic-like 300K more.txt','r');      % 3
elseif sample == 1
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
```

```
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
%         fid=fopen('IP stochastic-like 300K.txt','r');       % 4
    fid=fopen('IP stochastic-like 300K more.txt','r');       % 4
elseif sample == 3
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 5
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 5
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 5
    %     fid=fopen('IP3 stochastic-like 300K.txt','r');       % 5
    fid=fopen('IP3 stochastic-like 300K more.txt','r');       % 5
elseif sample == 7
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    fid=fopen('IP stochastic 300k.txt','r');
elseif sample == 8
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    fid=fopen('IP pbc 300k.txt','r');
elseif sample == 9
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\G2S 300k stochastic-like');
    fid=fopen('G2S stochastic-like 300k.txt','r');
```

```
end

format long;        % displays numbers to lots of decimals

itor = 1;
name_line=fgetl(fid);
rg_file_data = load(name_line);
rg_line=fgetl(fid);
rg_current=str2num(rg_line);

[pmf_win(itor,:), rg_win(itor,:)] = calc_pmf_window(rg_file_data, T,
rg_current, num_bin);
clear rg_file_data;
pmf_slope(itor,:) = calc_pmf_slope (pmf_win(itor,:), rg_win(itor,:));
prev_patch=pmf_win(1,:);
prev_slope=pmf_slope(1,:);
prev_rg=rg_win(1,:);

while ~(feof(fid))
    name_line=fgetl(fid);
    rg_file_data = load(name_line);
    rg_line=fgetl(fid);
    rg_current=str2num(rg_line);
    itor=itor+1;

    [pmf_win(itor,:), rg_win(itor,:)] = calc_pmf_window(rg_file_data, T,
rg_current, num_bin);
    clear rg_file_data;
    pmf_slope(itor,:) = calc_pmf_slope (pmf_win(itor,:),
rg_win(itor,:));
    [patched_pmf, patched_rg, patched_slope]=patch_pmf(prev_patch,
pmf_win(itor,:), ...
        prev_slope, pmf_slope(itor,:), prev_rg, rg_win(itor,:));
    prev_patch=patched_pmf;
    prev_slope=patched_slope;
    prev_rg=patched_rg;

end

fclose(fid);

%%%%%% CREATES THE GENERAL PMF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pmf_overall=patched_pmf;
rg_overall=patched_rg;

%%%%%%%%%%%%%%%%%%%% smoothing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pmf_length=length(pmf_overall);
for mtor = 2:pmf_length-1
    pmf_smoothed(mtor)=sum(pmf_overall(mtor-1:mtor+1))/3;
end
pmf_smoothed(1)=pmf_overall(1);
pmf_smoothed(pmf_length)=pmf_overall(pmf_length);

%%%%%%%%%%%%%%%%%%%% normalizing %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
pmf_min=min(pmf_smoothed);
pmf_final=pmf_smoothed-pmf_min;

figure(sample*100+5);
plot(rg_overall, pmf_final);
xlabel('radius of gyration, A');
ylabel('free energy');
title('overall potential of mean force');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% THIS FUNCTION IN TAKES THE .RGD FILE data OUTPUTED FROM CHARMM, THE
% TEMPERATURE OF SIMULATION IN K, AND THE RG OF THE SIMULATION WINDOW,
AND
% RETURNS (THE PMF FOR THE WINDOW + SOME CONSTANT)=corrected_e.
% num_bin divides rg into num_bin # of equal sections in each window


function [pmf_win, rg_win] = calc_pmf_window (rg_file_data, T,
rg_current, num_bin);


format long;        % displays numbers to lots of decimals


%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=3.29766597002e-27 * 6.022e23;  % in kcal/mol/K, converted from kcal/K
A=500;
%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%% READS RG %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
num = length(rg_file_data);
rg = zeros(num,1);

rg = double(rg_file_data);     % radius of gyration as doubles to
preserve digits
% rg = rg_temp(2001:5000);        % get rid of the first 2000 points,
time
% to reach EQ - already done in shell on stultz.
rg_size = length(rg);
%%%%%% READS RG %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%% MAKES HISTOGRAM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% can be replaced by histogram algorithm, but need to get max_rg,
min_rg,
% interval later.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
max_rg = max(rg);
min_rg = min(rg);
num_in_cat = zeros(num_bin,1);          % holds num of structure in each
section
rg_interval = (max_rg - min_rg) / num_bin;

for itor = 1:rg_size
    belong_cat = floor((rg(itor) - min_rg) / rg_interval) +1;
    if belong_cat == num_bin+1          % deals with the max
        belong_cat = belong_cat - 1;
    end
    num_in_cat(belong_cat) = num_in_cat(belong_cat)+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% or use a histogram plot built in from matlab to do the marked above
% figure;
% subplot(2,2,1);
% hist(rg,num_bin);         % plots histogram
% %stem(num_in_cat);
% title('Number of cases in each category');
% xlabel('RG');
% ylabel('# cases');
%%%%%% MAKES HISTOGRAM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%% CALC HYBRID PROBABILITY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hybrid_prob = num_in_cat / rg_size;
% subplot(2,2,2);
% stem(hybrid_prob);
% title('Hybrid probability for window');
% xlabel('Category');
% ylabel('hybrid probability');
%%%%%% CALC HYBRID PROBABILITY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%% CALC HYBRID ENERGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hybrid_e = -1 * k * T * log(hybrid_prob);
% subplot(2,2,3);
% stem(hybrid_e);
% title('Hybrid energy for window');
% xlabel('Category');
% ylabel('hybrid energy');
%%%%%% CALC HYBRID ENERGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%% CALC REAL ENERGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
corrected_e = zeros(num_bin,1);
bias = zeros(num_bin,1);
for ktor = 1:num_bin
    mid_rg = ((min_rg+ktor*rg_interval) + (min_rg + (ktor-
1)*rg_interval))/2;
```

```
                % instead of using individual RG from each structure,
the
                % average RG in each category is used
    bias(ktor) = A*((rg_current - mid_rg)^2);
    corrected_e(ktor)= hybrid_e(ktor) - bias(ktor);
end
% subplot(2,2,4);
% stem(corrected_e);
% title('Corrected energy for window');
% xlabel('Category');
% ylabel('corrected energy');
%%%%%% CALC REAL ENERGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for ltor=1:num_bin
    rg_win(ltor)=((min_rg+(ltor-
1)*rg_interval)+(min_rg+ltor*rg_interval))/2;
end;

pmf_win=corrected_e;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CALCULATES THE SLOPES OF N CONSECUTIVE POINTS GIVEN A PMF VECTOR

function pmf_slope = calc_pmf_slope (pmf_win, rg_win)

format long;        % displays numbers to lots of decimals

max_rg=max(rg_win);
min_rg=min(rg_win);

num_points = length(pmf_win);

for jtor = 1:num_points-1
    pmf_slope(jtor)=(pmf_win(jtor+1)-pmf_win(jtor))/(rg_win(jtor+1)-
rg_win(jtor));
        % note slope vector has 1 less entry than the pmf_win vector
        % currently only takes the neighbor point to calc the slope
end

% figure;
% stem(pmf_slope);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [patched_pmf,patched_rg,patched_slope] = patch_pmf(prev_patch,
current_pmf, ...
```

```
    prev_slope, current_slope, ...
    prev_rg, current_rg);

format long;        % displays numbers to lots of decimals

% CALCULATE THE OVERLAP

start_rg=current_rg(1);
end_index=length(prev_rg);

while start_rg<prev_rg(end_index)
    end_index=end_index-1;
end

% determine if the leftmost rg in the current segment being patched
should
% patched to end_index or end_index+1th rg of the previous patched
segments
% diff1=abs(start_rg-prev_rg(end_index));
% diff2=abs(start_rg-prev_rg(end_index+1));
% if diff2>diff1
%     patch_index=end_index
% else
%     patch_index=end_index+1
% end
patch_index=end_index;

total_amt_overlap=length(prev_rg)-patch_index;     % total amt of
intervals overlaping

% average slope error comparison:
% finds the average of the sse (sum of squared error) for different (1
to
% amt_overlap) amount of overlap. sse(1) is when all of the overlap is
% used. sse(total_amt_overlap) is when only 1 overlap is used.
% for itor=1:total_amt_overlap
%     sse(itor)=sum((prev_slope(patch_index+itor-1:length(prev_slope))-
current_slope(itor:total_amt_overlap)).^2);
%     avg_sse(itor)=sse(itor)/(total_amt_overlap-itor+1);
% end

% single point slope error comparison:
for itor=1:total_amt_overlap
    avg_sse(itor)=abs(prev_slope(patch_index+itor-1)-
current_slope(itor));
end


[best_sse, I]=min(avg_sse);     % I is the index of best_sse. smaller I
represents larger overlap.
%best_amt_overlap=total_amt_overlap-I;       %!! +1
best_overlap_index_start=patch_index+I;  % index of prev_pmf where we
start to replace new pmf !! -1

extension=length(current_rg)-total_amt_overlap;   %!! -1
```

```
patched_pmf=zeros(1,length(prev_rg)+extension);
patched_rg=zeros(1,length(prev_rg)+extension);
patched_slope=zeros(1,length(prev_rg)+extension-1);

patched_pmf(1:length(prev_patch))=prev_patch;
patched_rg(1:length(prev_rg))=prev_rg;
patched_slope(1:length(prev_slope))=prev_slope;


patched_pmf(best_overlap_index_start:length(patched_pmf))=patched_pmf(b
est_overlap_index_start+(-1))+current_pmf(I:length(current_pmf))-
current_pmf(I)...
    +(current_rg(I)-patched_rg(best_overlap_index_start+(-1)))*...
    current_slope(I);       % CMS and RST uses the slope to the right
of Ith rg in current_rg
%    patched_slope(best_overlap_index_start);    % FCY uses the slope
to the left of best_overlap_index_start-th rg in previous_rg
patched_rg(best_overlap_index_start:length(patched_rg))=current_rg(I:le
ngth(current_rg));
patched_slope(best_overlap_index_start:length(patched_slope))=current_s
lope(I:length(current_slope));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% WEIGHTED HISTOGRAM ANALYSIS METHOD (WHAM). need to decide which
bin_width to use, to balance out the two statistical requirements:
having as many bins as possible, but all bins wants to have >10
structures.

function [PMF_smooth,rg_current,PMF,mid_rg,final_P,N] = wham2_general(T,
bin_width,iteration,sample)

warning off all;

if sample == 5
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 3
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 3
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 3
        fid=fopen('IP2 stochastic-like 300K.txt','r');      % 3
%     fid=fopen('IP2 stochastic-like 300K more.txt','r');      % 3
elseif sample == 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 4
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
%         fid=fopen('IP stochastic-like 300K.txt','r');      % 4
    fid=fopen('IP stochastic-like 300K more.txt','r');      % 4
elseif sample == 6
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like'); % 5
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like'); % 5
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like'); % 5
    %     fid=fopen('IP3 stochastic-like 300K.txt','r');      % 5
    fid=fopen('IP3 stochastic-like 300K more.txt','r');      % 5
elseif sample == 7
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
```

```
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    fid=fopen('IP stochastic 300k.txt','r');
elseif sample == 8
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    fid=fopen('IP pbc 300k.txt','r');
elseif sample == 9
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP2 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP3 300k stochastic-like');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 25 stochastic');
    rmpath('C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\IP 300K pbc\RG files\dat files');
    path(path, 'C:\Documents and Settings\Frank\My Documents\Meng\PDB
files\RG\G2S 300k stochastic-like');
    fid=fopen('G2S stochastic-like 300k.txt','r');



end

format long;        % displays numbers to lots of decimals

rg_min=17.5;        % needs to be manually set for every peptide
rg_max=19.5;        % needs to be manually set for every peptide
% rg_min = 17.8;
% rg_max = 20.0;
% rg_min = 23.2;
% rg_max = 25.2;
%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=3.29766597002e-27 * 6.022e23;  % in kcal/mol/K, converted from kcal/K
beta=1/(k*T);
A=500;
%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%% make bins %%%%%%%%%%%%%%%%%%%%%%%%%%
rg_right_bounds = divide_rg (rg_min, rg_max, bin_width);
num_bin = length(rg_right_bounds);
mid_rg = rg_right_bounds - bin_width/2;
%%%%%%%%%%%%%% make bins %%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%% find N %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sim = 0;
while ~(feof(fid))
    sim = sim+1;
    name_line=fgetl(fid);
    rg_file_data = load(name_line);
    rg_line=fgetl(fid);
    rg_current(sim)=str2num(rg_line);
    rg(sim,:) = double(rg_file_data);      % radius of gyration as
doubles to preserve digits
    rg_size = length(rg(sim,:));
    N(sim,:) = zeros(1,num_bin);

    for itor = 1:rg_size
        bin = 1;
        while rg(sim, itor)>rg_right_bounds(bin)
            bin=bin+1;
        end
        N(sim,bin)=N(sim,bin)+1;
    end
end
num_sim = sim;
%%%%%%%%%%%% find N %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%% wham iteration %%%%%%%%%%%%%%%%%%%%%%
F = zeros(num_sim,1);       % initialize f
fig_num = 100*sample;

for iter = 1:iteration
    F_old=F;
    F=zeros(num_sim,1);
    [prob,F]=wham2_iterations(T, num_bin, num_sim, mid_rg, N,
rg_current, F, F_old, rg_size, rg);
    %     figure(fig_num);
    %     plot(rg_current, F);
    %     hold on;
    error(iter) = sum((F_old - F).^2);
end
%%%%%%%%%%%% wham iteration %%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%% print outs %%%%%%%%%%%%%%%%%%%%%%%%%%%
last_F = F;

% figure(fig_num+4);
% plot(rg_current, last_F,'r');
% title('converge of F. last F is in red');
% title('last f (free energy)');
sum_prob=sum(prob);
final_P = prob/sum_prob;
% figure(fig_num+1);
% plot(mid_rg, final_P);
```

xxx

```
% title('unbiased P');

final_PMF = -1/beta*log(final_P);

min_final_PMF = min(final_PMF);
final_PMF = final_PMF - min_final_PMF;
figure(fig_num+2);
plot(mid_rg,final_PMF);
title('free energy = -beta*log(prob/sum(prob))');
% figure(fig_num+3);
% plot(mid_rg,final_PMF);
% title('free energy zoom in');
% axis([17 20 0 5]);


for h=1:num_sim
    k=1;
    while rg_current(h)>mid_rg(k)
        k=k+1;
    end
    slope = (N(h,k)-N(h,k-1))/(mid_rg(k)-mid_rg(k-1));
    N_extrapolated = N(h,k)-slope*(mid_rg(k)-rg_current(h));
    prob_i_zeta(h) = N_extrapolated/rg_size;
end

W=-1/beta*log(prob_i_zeta')+last_F;
W = W-min(W);
figure(fig_num+4);
plot(rg_current,W);
title('free energy, using F');

PMF_smooth = W;
PMF = final_PMF;
% figure(fig_num+4);
% plot([1:iter],error);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function rg_right_bounds = divide_rg (rg_min, rg_max, bin_interval)
%%% use this to divide the overall rg range into "bins".

if bin_interval<=0.009
    extra = 100;
elseif bin_interval<=0.099
    extra = 15;
else
    extra = 2;
end
rg_leftmost = rg_min-extra*bin_interval;
rg_rightmost = rg_max+extra*bin_interval;
```

```
rg_right_bounds(1)=rg_leftmost+bin_interval;

% note the rightmost bound is not counted because its interval is not
the
% same as the others
for itor = 2:(rg_max-rg_min)/bin_interval+(2*extra-1)
    rg_right_bounds(itor)=rg_right_bounds(itor-1)+bin_interval;
end




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [prob,F]=wham2_iterations(T, num_bin, num_win, mid_rg, N,
rg_current, F, F_old, n, rg);

warning off all;

%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k=3.29766597002e-27 * 6.022e23;  % in kcal/mol/K, converted from kcal/K
beta=1/(k*T);
A=500;
%%%%%% CONSTANTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for i=1:num_bin
    num=0;
    denom=0;

    for j=1:num_win
        num = num + N(j,i);
        bias(j) = A*(rg_current(j) - mid_rg(i))^2;
        bf = exp((F_old(j) - bias(j)) * beta);
        denom = denom + n * bf;     % n = 3000;
    end
    prob(i) = num/denom;

    for j=1:num_win
        bf = exp(-1*bias(j)*beta)*prob(i);
        F(j) = F(j) + bf;
        end
end

% for i=1:num_bin
%         for j=1:num_win
%         bf = exp(-1*bias(j,i)*beta)*prob(i);
%         F(j) = F(j) + bf;
%         end
% end
% for j=1:num_win
```

```
%     for i=1:num_bin
%         bf = exp(-1*bias(j,i)*beta)*prob(i);
%         F(j) = F(j) + bf;
%     end
% end

for j = 1:num_win
    F(j) = -1/beta*log(F(j));
end
```