


Winter 2008

Improved Constrained Global Optimization for Estimating Molecular Structure From Atomic Distances

Terri Marie Grant
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/mathstat_etds

 Part of the [Bioinformatics Commons](#), [Mathematics Commons](#), and the [Molecular Biology Commons](#)

Recommended Citation

Grant, Terri M. "Improved Constrained Global Optimization for Estimating Molecular Structure From Atomic Distances" (2008). Doctor of Philosophy (PhD), dissertation, Mathematics and Statistics, Old Dominion University, DOI: 10.25777/afxp-2j69 https://digitalcommons.odu.edu/mathstat_etds/15

This Dissertation is brought to you for free and open access by the Mathematics & Statistics at ODU Digital Commons. It has been accepted for inclusion in Mathematics & Statistics Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**IMPROVED CONSTRAINED GLOBAL OPTIMIZATION FOR
ESTIMATING MOLECULAR STRUCTURE FROM ATOMIC
DISTANCES**

by

Terri Marie Grant
M.S. May 2003, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

APPLIED AND COMPUTATIONAL MATHEMATICS

OLD DOMINION UNIVERSITY
December 2008

Approved by:

Glenn Williams (Director)

John Adam (Member)

Gordon Melrose (Member)

Fang Hu (Member)

Lesley Greene (Member)

ABSTRACT

IMPROVED CONSTRAINED GLOBAL OPTIMIZATION FOR ESTIMATING MOLECULAR STRUCTURE FROM ATOMIC DISTANCES

Terri Marie Grant
Old Dominion University, 2008
Director: Dr. Glenn A. Williams

Determination of molecular structure is commonly posed as a nonlinear optimization problem. The objective functions rely on a vast amount of structural data. As a result, the objective functions are most often nonconvex, nonsmooth, and possess many local minima. Furthermore, introduction of additional structural data into the objective function creates barriers in finding the global minimum, causes additional computational issues associated with evaluating the function, and makes physical constraint enforcement intractable. To combat the computational problems associated with standard nonlinear optimization formulations, Williams *et al.* (2001) proposed an atom-based optimization, referred to as GNOMAD, which complements a simple interatomic distance potential with van der Waals (VDW) constraints to provide better quality protein structures. However, the improvement in more detailed structural features such as shape and chirality requires the integration of additional constraint types.

This dissertation builds on the GNOMAD algorithm in using structural data to estimate the three-dimensional structure of a protein. We develop several methods to make GNOMAD capable of effectively and efficiently handling non-distance information including torsional angles and molecular surface data. In specific, we propose a method for using distances to effectively satisfy known torsional information and show that use of this method results in a significant improvement in the quality of α -helices and β -strands within the protein. We also show that molecular surface data in combination with our improved secondary structure estimation method and long-range distance data offer increased accuracy in spatial proximity of α -helices and β -strands within the protein, and thus provide better estimates of tertiary protein structure. Lastly, we show that the enhanced GNOMAD molecular structure estimation framework is effective in predicting protein structures in the context of comparative modeling.

This thesis is dedicated to my parents, *Anthony* and *Tina G. Jordan*.

ACKNOWLEDGMENTS

Completion of this dissertation is a dream comes true. I would like to give thanks to my God, Jehovah, for placing me in the field of mathematics, for giving me the desire to stay committed this research, and for placing so many people in my life who were instrumental in the successful completion of this dissertation. I would like to thank my dad, Anthony Jordan, and mom, Tina Grant Jordan, for their never-ending encouragement and love over the duration of this dissertation. I would like to thank my sisters, Latrice Grant and Sherri Grant, for always helping me to be a bit balanced in my life. I would like to thank my dissertation advisor, Dr. Glenn A. Williams, for the time he spent in helping me to understand protein structure estimation and its relationship to optimization, for his words of encouragement during the valleys of this research, and for his unwavering dedication in helping me complete this dissertation. I would like to thank my committee members, Dr. John Adam, Dr. Gordon Melrose, Dr. Fang Hu, and Dr. Lesley Greene for their interest in my research and for reading and providing helpful suggestions in improving my dissertation. I would like to thank Dr. John Dorrepaal for making it possible for me to receive funding over the duration of my dissertation.

I would like to give a special thanks to Barbara Jeffrey for being a listening ear, for giving me good advice, and for always reminding me to stay focused. I would also like to give a special thanks to Gayle Tarkelsen for making sure that I was well feed and for always encouraging me. I would like to also thank Rose Anne Corbin, Sarah Parrish, Charles Touron, Caleb Adams, and Li Liu for helping me prepare for my oral defense and for listening to me talk for hours about protein structure estimation (although they were unfamiliar with the subject matter). Lastly, I would like to give thanks to all my friends who came into my life during the final stages of my dissertation for providing invaluable guidance and encouragement.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	viii
LIST OF FIGURES	ix
 Chapter	
I. INTRODUCTION.....	1
I.1. PROTEINS AND COMPUTATIONAL MODELING EFFORTS	2
I.2. SCOPE OF DISSERTATION.....	3
I.3. ORGANIZATION OF DISSERTATION.....	7
II. NONLINEAR OPTIMIZATION.....	8
II.1. STANDARD OPTIMIZATION FORMULATION.....	8
II.2. LOCAL OPTIMIZATION	10
II.3. OVERVIEW OF GLOBAL OPTIMIZATION	18
II.4. CONSTRAINED GLOBAL OPTIMIZATION	28
II.5. OVERVIEW OF GNOMAD	29
III. A PRACTICAL DISTANCE-BASED METHOD FOR SATISFYING TORSIONAL ANGLE INFORMATION	43
III.1. INTRODUCTION	43
III.2. BACKBONE TORSIONAL ANGLES.....	44
III.3. METHODOLOGY	46
III.4. VALIDATION	57
III.5. CONCLUSION	69
IV. A COMPUTATIONALLY EFFICIENT METHOD FOR USING MOLECULAR SURFACE CONSTRAINTS	72
IV.1. INTRODUCTION.....	72
IV.2. CONSTRUCTION OF MOLECULAR SURFACES.....	73
IV.3. METHODOLOGY	74
IV.4. VALIDATION	82
IV.5. CONCLUSION	100
V. COMPARATIVE MODELING USING GNOMAD	102
V.1. INTRODUCTION	102
V.2. STATISTICAL MODELING EFFORTS FOR ESTIMATING PROTEIN STRUCTURE	103
V.3. METHODOLOGY	106
V.4. VALIDATION	111

V.5. CONCLUSION	115
VI. CONCLUSIONS AND FUTURE WORK.....	117
VI.1. IMPROVEMENT OF THE VDW CONSTRAINT	117
VI.2. EFFECTIVE USE OF CONTACT INFORMATION IN IMPROVING BETA-STRUCTURES	117
VI.3. SOLVENT ACCESSIBLE CONSTRAINT.....	118
VI.4. CONCLUSION.....	119
REFERENCES	121
VITA.....	129

LIST OF TABLES

Table	Page
1. Inter-atomic distance functions	11
2. Angle potential functions	12
3. Minimum separation distances between all possible combinations of atom types in the model	40
4. Accuracy results for using distances to satisfy torsional information	61
5. Test structures mostly comprised of helices and strands	85
6. Preliminary results for surface constraint enforcement	89

LIST OF FIGURES

Figure	Page
1. Flowchart illustrating the basic principal of computational methods for protein structure estimation	4
2. Example of bond length, bond angle, torsional angle and non-bonded interaction	10
3. Diagram of atom move procedure	32
4. Diagram of physical constraint enforcement	34
5. Example of the chiral forms of an amino acid	36
6. Example of enforcing chirality constraint.....	37
7. Main chain atomic representation consisting of the backbone atoms N, C _α , C, and C _β where C _β atom represents the sidechain	39
8. Outline for GNOMAD algorithm for atomic distances	41
9. Definition of the torsional angle ϕ using main chain atoms	45
10. The geometry of the general polypeptide chain.....	45
11. The placement of <i>l</i> -plane with respect to the bond angle γ and the placement of three dummy atoms – D ₁ , D ₂ , and D ₃ – in the <i>l</i> -plane.....	48
12. A segment of the new atomic representation.....	49
13. Using dummy atoms and bond angles to determine the torsional angle ϕ	50
14. Definition of the torsional angle ϕ in reference to L_1 and L_2	51
15. The outline of the algorithm for numerical uniqueness of a torsional angle	53
16. The starting points of a subsequent optimization grouping	54

17.	Determining the position of the first update atom	55
18.	The initialized position of remaining update atoms in a segment of the polypeptide	55
19.	The outline of the algorithm for finding a set of starting points for remaining atoms in the current optimization grouping	57
20.	Results of relative error for using distances to satisfy torsional angles found in helix region for main chain, main chain with torsional angle constraint, and new atomic representation	62
21.	Results of relative error for using distances to satisfy torsional angles found in strand region for main chain, main chain with torsional angle constraint, and new atomic representation	63
22.	1CTF α -helix and torsional method results	64
23.	1CTF β -strand and torsional method results	65
24.	Illustration of constructing a surface in R^2	76
25.	Outline for molecular dot surface algorithm for atomic distances	77
26.	3D example of constructing a dot surface from an atomic representation	78
27.	Outline for enforcing the surface constraint on an atom	79
28.	Outline for finding the optimal alignment for an original and an experimental set of surface nodes	81
29.	3D illustration of satisfying surface data using translational and rotational alignment	83
30.	Comparison results on the 1ozz protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD	91
31.	Comparison results on the 1ctf protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD	91
32.	Comparison results on the 1aw0 protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained	

	GNOMAD.....	92
33.	Comparison results on the 1fvs protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.....	92
34.	Comparison results on the 1bta protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.....	93
35.	Comparison results on the 1aps protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.....	93
36.	1OZZ protein structure and surface constraint enforcement results.....	95
37.	1AW0 protein structure and surface constraint enforcement results.....	96
38.	1APS protein structure and surface constraint enforcement results.....	97
39.	1BTA protein structure and surface constraint enforcement results.....	98
40.	Flowchart illustrating the basic principal of comparative modeling method by satisfaction of spatial restraint.....	105
41.	A segment of sequence alignment for the <i>E. coli</i> protein and its closely related family members.....	108
42.	Illustration of ϕ and ψ torsional angle distribution computed from known protein structure from PDBselect data bank PDB files.....	110
43.	Illustration of high structural similarity of the family members of 1CTF protein.....	111
45.	Comparison of torsional angle variance, σ^a , for results over a range of distance range tolerance.....	112
46.	Illustrating the effect of decreasing the reliability of torsional information.....	113

CHAPTER I

INTRODUCTION

Could you imagine a world where children are free from the attack of deadly disease such as sickle cell anemia, leukemia, and cystic fibrosis, where people in the prime of their lives are free from viruses that attack the immune system, and where the elderly can cope with diseases that deteriorate their memory such as Alzheimer's, and dementia? What advances would have to take place in order to make this world a near reality? One answer would be simply to design new drugs for better treatment of disease. However, there is nothing simple about the work that must be done behind the scenes in order to design better drugs.

Drug design finds its basis in understanding the functions of proteins. Proteins, although very small in size, are the basic structures of life. They are crucial in ensuring that every part of the body works collectively together to maintain proper functioning of the body as a whole. Because proteins come in many different sizes, shapes, and sequences they can perform a variety of important tasks in the body. For example, hemoglobin transports oxygen throughout our body, antibodies defend our body against bacteria and viruses, and ion channel proteins control brain signaling by allowing small molecules into and out of nerve cells (Schlick 2002)

In just surveying these specific biological functions, what do you think would happen if the proteins were not correctly shaped? In short, there would be a breakdown in processes of the body. As a result, these deadly diseases that we hope will never plague us or anyone in our circle could become a reality in our lives. For example, abnormal hemoglobin proteins cause sickle cell disease. What is more intriguing about this disease is that the incorrect position of only one amino acid in a specific protein affects the transportation of oxygen throughout the body. There are many diseases that are caused from incorrectly folded proteins, including cystic fibrosis, Parkinson's, and Alzheimer's (Irvine *et al.* 2008). In considering these examples, we clearly understand how important

shape is to the study of proteins (Irvine *et al.* 2008).

I.1. Proteins and Computational Modeling Efforts

In the context of molecular modeling, the shape of a protein is the detailed three-dimensional representation of the protein. Experimental methods are the primary methods for determining the three-dimensional structure of protein and are responsible for the determination of many structures found in current databanks. X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy are the most common experimental methods in use today (Andrec *et al.* 2007). In using X-ray crystallography, x-rays are aimed at the solid crystal of the protein, the crystal then scatters the rays on to a detector that captures in three-dimensions how the crystal scatters, and the intensity of each diffracted ray is used in a computer program to calculate the position of each atom in the protein (Knight *et al.* 2008; Ilari and Savino 2008). These methods are limited by the amount of time required to crystallize the protein and the difficulty determining the structure of larger proteins such a multidomain complexes and membrane proteins (Wall *et al.* 1999).

Alternatively, NMR methods are based on solubility behavior of the protein. In the case of soluble proteins, the proteins are submerged into aqueous solution and an analysis of the relationship between the magnetic field and atoms commonly found in proteins such as carbon and nitrogen is used to determine the protein structure (Wüthrich 1990; Wüthrich 2003). However, these methods are limited by the insolubility condition of some proteins. In more recent years, however, NMR researchers have focused on developing solid-state methods for determination of three-dimensional structures for insoluble proteins (Brünger *et al.* 1998; Mehta *et al.* 2008).

To expedite the process of determining the three-dimensional structure, the data extracted from these experimental techniques are employed to estimate the three-dimensional structure through use of a computer program. These programs are most often based on a nonlinear optimization formulation. The optimization formulation typically consists of an objective function that serves as a basis for developing the search space, constraints that model biological conditions, and optimization parameters that are used to design certain features in the model. The goal of the optimization is to find the coordinates for a set of atoms that best satisfies a given set of structural data describing

the protein (Chen *et al.* 1996; More' *et al.* 1999; Sheraga *et al.* 2004).

The basic idea behind these computational approaches, as illustrated in Figure 1, is to generate data from all available sources (*i.e. experimental, statistical, etc.*) and then use that information to estimate a reasonable three-dimensional structure of the protein (Chen 2000; Schwieters *et al.* 2003; Kinjo and Nishikawa 2005; Wu and Wu 2007). One of the current approaches for developing a computer program is to choose a simple objective function based on distances or angles and define physical constraints for any additional data.

In this research, we consider an existing constrained global optimization method, called GNOMAD, that complements a simple distance-based objective function with a van der Waals (VDW) nonbonded interaction constraint and a local domain specific perturbation method to provide better estimates of three-dimensional molecular structures (Williams *et al.* 2001). The choice of a simple objective function reduces the complexity of dealing with large number of interdependent variables and the atom-based approach makes it possible to compute nonlinear iterations in systems of dimension 3 as opposed to a single system of dimension $3n$, where $3n$ is the number of atoms in the molecule. The constraint enforcement procedure ensures that structures satisfy Van der Waals forces, and the domain-specific perturbation methods aid in helping the optimization to move out of local minima (Williams *et al.* 2001). One of the main difficulties associated with a distance-based optimization approach is the large number of low-error local minima that arise, particularly as the amount of distances decreases. Addition of torsional angle and non-distance structural data further complicate the optimization procedure.

I.2. Scope of Dissertation

In this dissertation, we study methods that will improve the quality of estimating the three-dimensional structure of single-domain globular proteins. These proteins are usually spherical or elliptical in nature and are generally comprised of helices and sheets. One central assumption for the folding of these types of proteins is that the protein folds in levels for which the formation of the secondary structure precedes the formation of the tertiary structure (Roder 2004).

In light of this observation, our goal is to provide better quality structures by including structural data that will ensure that the secondary structures are formed

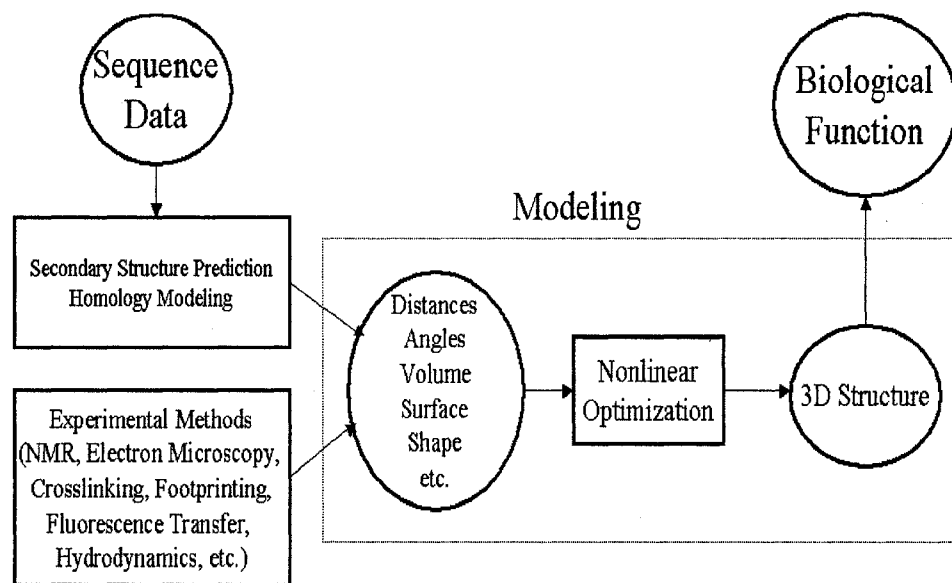


Figure 1. Flowchart illustrating the basic principle of computational methods for protein structure estimation.

correctly and that will aid in forming the tertiary structure. In specific, we investigate the integration of structural data that accurately represent the geometric detail of local secondary structure and that captures some of the effects of the principal forces of folding in globular proteins. These forces include the hydrophobic forces between amino acids and their environment and the local forces that drive the internal architectures in globular proteins such as hydrogen bonding and van der Waals forces (Hue and Dill 1990). We, therefore, propose methods for handling torsional and molecular surface information using the GNOMAD structural estimation framework.

The main objectives of this research is: i) to propose a distance-based method for effectively using torsional angle information and show its value in improving the formation of secondary structures, ii) to demonstrate how readily available surface data such as molecular surfaces can be effectively used to constrain atoms and to improve the estimation of the protein structure, and iii) to investigate the performance of the GNOMAD algorithm when a combination of structural data taken from multiples data sources, including experimental, theoretical, and statistical methods, are employed to estimate the three-dimensional structure of a protein.

I.2.1. Torsional Angles

Torsional angles can provide valuable and “free” information for improving the formation of local substructures within the protein. It is the second most common type of structural data used in many formulations and is gathered from many sources including experimental techniques, comparative and empirical modeling, and secondary structure prediction methods. Torsional angle information can be incorporated directly into the nonlinear objective function or penalty function approaches can be employed to constrain angles within a given range. Torsional angle functions are highly nonlinear as functions of coordinates and therefore significantly complicate the optimization procedure. Penalty function methods also complicate the optimization procedure by adding more barriers to convergence and local minima. It is also difficult to ensure angle constraint satisfaction when using an atom-based optimization approach such as GNOMAD. The inability of GNOMAD to effectively use this readily available torsional information limits the accuracy in the formation of reoccurring substructures that are found in a large number of globular proteins, such as α -helices and β -sheets.

In the first part of this dissertation, our goal is to enhance GNOMAD to make it capable of using torsional information more effectively. The underlying idea is to define a core set of distances that will allow us to constrain atoms to positions that coincide with the known values of backbone torsional angles. In order to identify these distances, we develop a new atomic representation of a polypeptide model. This new atomic representation will allow us to use pseudo-atoms to circumvent the difficulty of satisfying torsional information with distances, to form known torsional angles correctly throughout the optimization process, and to investigate the effectiveness of using distances to significantly improve the formation of secondary structure within the protein.

I.2.2. Surface Constraint

In current research, the external forces that drive the folding of globular proteins are assumed to be the hydrophobic/hydrophilic interaction between amino acids and their environment. The second part of this dissertation is to examine methods for making GMOMAD capable of handling information that is representative of the interaction between atoms and their environment (Wade *et al.* 1996; Cao *et al.* 2002). In this work,

we investigate a computational method for effectively incorporating molecular surface data into GNOMAD.

In designing the surface constraint for the GNOMAD framework, we must first construct a molecular surface within which to constrain atoms. In general, analysis of the molecular surface is a computational problem that is attacked separately from the protein structure estimation problem and provides much insight in many areas of drug design including, identifying clefts and possible drug binding sites in protein surfaces, and screening databases of small molecules for the purpose of identifying molecules of possible pharmaceutical use (Connolly 1983). In the context of protein structure estimation, however, the molecular surface computation provides a way of modeling the interactions between the protein and its environment (Gallicchio and Levy 2000; Felts *et al.* 2002; Kar *et al.* 2006).

In reference to our protein structure estimation algorithm, we intend to use the molecular surface in combination with our improved secondary structure satisfaction method and short-range contact distances¹ to provide better estimates of protein structure. Accordingly, we combine torsional, surface, and contact information to show that our surface constraint method can provide more accurate estimation of molecular structure.

I.2.3. Comparative Modeling Using GNOMAD

Current modeling efforts focus on developing a molecular structure estimator that is effective in using a combination of structural data taken from multiple sources and that does not give the same amount of influence to less precisely measured data (Altman 1985). These approaches include comparative modeling (Sali and Blundell 1993; Furnham *et al.* 2008), empirical or knowledge-based methods (Wall *et al.* 1999), or Bayesian methods (Altman 1985; Chen 2000; Dugan *et al.* 2004).

Many molecular structure estimators are developed based on an unconstrained optimization formulation. Accordingly, some of the computational issues that arise in using these estimators could possibly be attributed to this formulation. In specific, the introduction of additional penalty terms in the objective function creates barriers in finding the global minimum and causes additional computational issues associated with

¹ The phrase “short-range contact distance” refers to a distance between atoms that are found in residues that are far apart in the primary sequence of amino acids and that are less than 6 Å.

evaluating the function (Friesner and Gunn 1996; Williams *et al.* 2001). Alternatively, the weighted nonlinear least squares formulation employed in GNOMAD allows us to combine experimental and statistical data, gives highly precise data more influence than less accurate data in estimating the three-dimensional structure, and possibly avoids some of the pitfalls of unconstrained optimization formulations.

The last objective of this dissertation is to broaden the scope of our work on predicting three-dimensional structure to include comparative modeling. The motivation for this work is to investigate the effectiveness of GNOMAD in estimating the structure of a protein through use of inexact structural information taken from a subset of known protein structures (Sali and Blundell 1993). Starting with a sequence alignment of the target protein and its closely related family members², structural data is generated using standard statistical methods. Then, the GNOMAD structural estimation framework is used to predict the three-dimensional structure of the target protein.

We test the algorithm's ability in using distances and torsional angles derived from the comparative modeling procedure to estimate the protein structure, and compare the generated protein structure with the known crystal structure. The set of distances generated from the comparative modeling procedure includes local bond length and bond angle distances as well as short-range contact distances.

I.3. Organization of Dissertation

The remainder of this dissertation is organized as follows. In Chapter II, we give a survey of standard optimization formulations employed for protein structure prediction, we review the most common nonlinear optimization algorithm adapted in protein modeling and the computational problems existing in these algorithms, and we present an overview of the GNOMAD structural estimation framework. In Chapter III, we describe our practical method for effectively using distances to satisfy torsional information. In Chapter IV, we describe our method for using the molecular surface of the protein to constrain the position of atoms. In Chapter V, we investigate the improved GNOMAD algorithm in using more realistic data taken from comparative modeling methods. In Chapter VI, a discussion of conclusions and future directions will complete this thesis.

² A "family member" of a protein is a known protein structure that has high sequence or structural similarities with the target protein.

CHAPTER II

NONLINEAR OPTIMIZATION

II.1. Standard Optimization Formulation

The objective functions used for estimating three-dimensional protein structure generally rely on energies and forces between atoms that make up the protein. Potential energy, least squares, and knowledge-based formulations are among the most common optimization formulations adapted for protein structure estimation. Although each formulation differs in its underlying assumptions, the objective functions are derived using structural information such as bond lengths, non-bonded distances, bond angles, and torsional angles.

Potential energy minimization is generally employed in many protein structure estimation algorithms. The foundation of this approach is based on the observation that a protein becomes biologically active when it has reached its final folded state, which is considered to be the lowest energy state (Hansmann 2003; Sheraga *et al.* 2004). In light of this work, the basic assumption of this formulation is that the interaction between the atoms and the environment dictate the behavior of proteins to conform into its optimal configuration (Neumaier 1997; Dobson *et al.* 1998; Floudas *et al.* 1999). In order to curtail the number of terms in the potential energy function only the dominant forces are modeled including, hydrophobic effects, hydrogen bonding, and electrostatic interactions (Dill 1990; Dobson *et al.* 1998).

The second most widely used optimization formulation is the least squares approach. The idea of this approach is to minimize the relative errors between calculated and given data. Hence, the objective function represents the measure of how well a particular piece of data is satisfied by the model. The goal is to move atoms so that the model structure closely matches the given input data. This formulation originated from work done on reformulating the molecular distance geometry problem as a global least squares problem (Hendrickson 1991; Wu and Wu 2007). Williams *et al.* (2001) also employed a nonlinear least squares formulation to use bonded and nonbonded distances in estimating the three-dimensional structure of proteins.

Another approach for generating interatomic potentials is to make use of the relationships between atoms in the thousands of known protein structures. These knowledge-based potentials and probability density functions are employed as ideal objective functions. The derivation of these objective functions is based on the idea that small change in the sequence of amino acids results in a small change in the three dimensional structure. Using statistical analysis of collected structural data associated with conserved or low-variance regions of protein, an objective function is derived that gives the probability of occurrence of any combination of structural features (Sali and Blundell 1993; Furnham *et al.* 2008).

Common to either formulation, is the statement of the problem. Formally, the optimization problem is to find the configuration that best satisfies the set of structural data. The optimization problem is written as,

$$\min_x F(x) \quad (2.1)$$

where $x = \{x_i \in R^3, i = 1, \dots, n\}$ represents the molecular configuration, n is the number of atoms in the configuration, and $F(x)$ is the objective function which generally consist of distances, angles, or combination of these parameters. The distance-based functions are generally representative of bond stretching, non-bonded interactions, or L_2 -error and, the angle-based functions are generally representative of angular bending and angular rotation. In Figure 2, we show examples of structural data that are used as parameters in the optimization formulation.

A large amount of research has been devoted to deriving energy, knowledge-based, and probability density function for improving the objective function. The basic assumption here is that the structure of a protein is well determined by a sufficient number of terms modeling structural features and energies (Pearlman *et al.* 1995; Floudas *et al.* 1999; Kinjo and Nishikawa 2005). One of the most widely used terms is one that uses a set of interatomic distances pertaining to a representative set of atoms in the protein. The general form given by the Equation 2.2

$$f = \sum_{i,j \in P} h(r_{ij}) \quad (2.2)$$

where h is the interatomic function and $r_{ij} = \|x_i - x_j\|$ represent the known distance

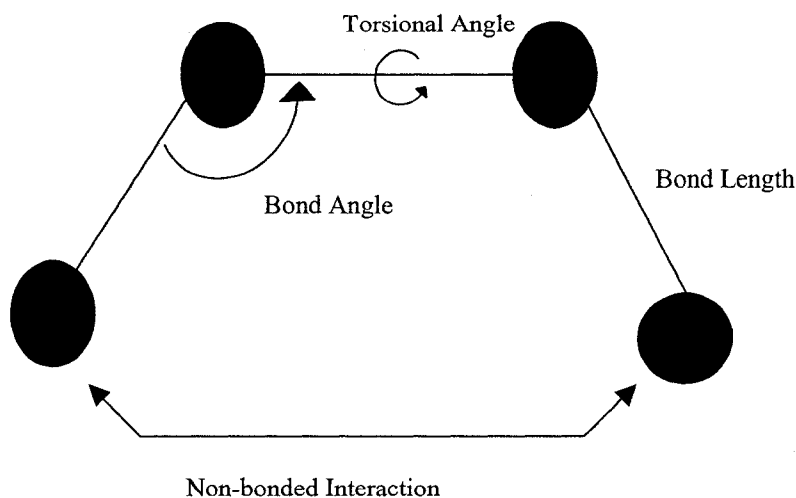


Figure 2. Example of bond length, bond angle, torsional angle and non-bonded interaction.

between atom i and atom j , and P is the set of distances.

These distance-based functions are generally representative of bond stretching, non-bonded interactions, or L_2 -error. Several forms for the h function often employed in optimization formulation are given in Table 1. Included in this table are descriptions of the $h(x)$ functions that define the distance based objective function, parameters for each function, and previous authors who used these objective functions.

Distance information is often supplemented with other types of information in the optimization formulation. Most often, angle-dependent objective functions are used individually or in combination with interatomic potentials to capture the important features that are not captured by distances. The most widely used angle potentials are given in Table 2. Included in this table are descriptions of the energy function, parameters for each function, and previous authors who employed these types of functions.

II.2. Local Optimization

The general form of the objective functions used for estimating the molecular structure is a nonlinear function of input data. Hence, there are a variety of nonlinear optimization methods that have been employed to determine a good estimate of the structure. The basic idea of these nonlinear optimization methods is to perform a local search in the configuration space for the minimum through iteratively or recursively

Table 1. Inter-atomic distance functions.

Survey of Interatomic Distances Used in Optimization Formulation			
Name	Function	Constant of Proportionality	Works
Bond stretching or L_2 -error function	$h(x) = s_{ij}(x - d_{ij})^2$	s_{ij} is the bending constant or variance d_{ij} is the calculated distances	Brooks <i>et al.</i> (1983) Williams <i>et al.</i> (2001)
Stress function	$h(x) = (x^2 - d_{ij}^2)^2$	d_{ij} is the bond lengths or bond angles distances	Hendrickson (1991) More' and Wu (1997)
Lower bond function	$h(x) = \min^2 \left\{ \left(\frac{x^2 - l_{ij}^2}{l_{ij}^2} \right), 0 \right\}$	l_{ij} is lower bond on distances	Crippen and Havel (1988)
Upper bound function	$h(x) = \min^2 \left\{ \left(\frac{x^2 - u_{ij}^2}{u_{ij}^2} \right), 0 \right\}$	u_{ij} is upper bond on distances	Crippen and Havel (1988)
Electrostatic potential	$h(x) = \frac{q_i q_j}{\varepsilon_{ij} x}$	q_i - charge of atom i q_j - charge of atom j ε_{ij} - dielectric constant	Barbosa <i>et al.</i> (2003)
Van der Waals potential	$h(x) = \varepsilon_{ij} \left[\left(\frac{1}{x} \right)^{12} - \left(\frac{1}{x} \right)^6 \right]$	ε_{ij} - dielectric constant	Barbosa <i>et al.</i> (2003)

generating a sequence of iterations from an initial configuration. Each of the iterations consists of an update in the atomic positions of the current configuration by taking into account feasibility, the direction, and the curvature of the function. The resulting configuration is in the neighborhood of the current configuration and replaces the current configuration, if it has lower function value. The algorithm terminates when a configuration is obtained whose function is no worse than any of its neighbors (van Laarhoven and Aarts 1987).

Local minima are defined by the presence of a very small gradient and can occur when the function value is high and the configuration has moved far from the initial configuration (Williams *et al.* 2001). Nonlinear optimization methods perform best if the starting point is within a certain neighborhood of the minimum. Further, the algorithm can stall if the stopping criteria fail to satisfy certain conditions. Hence, these local

Table 2. Angle potential functions.

Angle-based Potentials Used in Optimization Formulation			
Name	Potential	Constants	Works
Bond angle potential	$E_{ba} = \sum_{ijk\text{-bonded}} \frac{c_{ijk}}{2} (\theta_{ijk} - \bar{\theta}_{ijk})^2$	c_{ijk} - bending constant θ_{ijk} - bond angle $\bar{\theta}_{ijk}$ - reference bond angle	Brooks <i>et al.</i> (2003)
Torsional	$E_{ta} = \sum_{ijkl\text{-angle}} c_{ijkl} [1 + \cos(n\theta_{ijkl} - \bar{\theta}_{ijkl})]$	c_{ijkl} - torsional constant θ_{ijkl} - torsional angle $\bar{\theta}_{ijkl}$ - reference torsional angle	Barbosa <i>et al.</i> (2003)

optimization methods alone will not yield acceptable estimates of molecular structure. However, these standard local optimization methods are used as the core of many global optimization methods adapted for protein structure estimation to improve upon the estimates of the global minimum (Torn and Zilinskas 1989).

The purpose of local optimization methods is to find the direction in which a function decreases most rapidly and to determine an acceptable step length to move in that direction. Mathematically, updating the atomic position is represented by the following supposition. Let $x = \{x_i \in R^3, i = 1, \dots, n\}$ represent the molecular configuration with each x_i specifying the spatial position of atom i . A general update in the atomic position is given by

$$x_{new} = x_{old} + \lambda \Delta \vec{x} \quad (2.3)$$

where the direction from the current point for which the objective function decreases initially is given by $\Delta \vec{x}$ and a step length in that direction is given by λ . The underlying idea is to take steps in a direction that lead “downhill” for the objective function f , *i.e.* a descent direction (Dennis and Schnabel 1996). Determination of the step direction and step length is generally approached using the second order truncated Taylor expansion

(TTE) of the objective function or by operating on local quadratic approximations to the objective function (Miller 2000). The advantage of using local quadratic approximations is faster convergence to local minimum than that obtained through using the original function (Williams *et al.* 2001).

II.2.1. Local Optimization Algorithms

Local optimization can be divided into several categories of update methods: (1) gradient-based updates, which use only first derivative information; (2) Newton's Method updates which use first and second derivative information; and (3) Quasi-Newton updates which use first derivative information and approximations to the second derivatives.

Steepest descent (SD) is one of the most commonly used gradient-based methods in molecular structure algorithms because of the ease in implementing the algorithm and the small amount of memory space required at each of the iterations for computing the derivatives. In this method, the step direction is taken to be the negative gradient of the function because the negative gradient is a measure of steepest descent direction. However, this implementation results in poor convergence and is not guaranteed to converge in a finite number of steps. As a result, a slightly different step direction is taken to improve convergence in the local optimization. Instead of taken directions that are orthogonal to the previous direction, the conjugate gradient (CG) method makes use of previous history of minimization steps as well as the current gradient to take directions that are conjugate to the previous direction. Moreover, this method requires fewer evaluations of the objective function and gradient than SD minimization, and thus converges to the minimum in $O(n)$ steps for quadratic function where n is the degrees of freedom (Brooks *et al.* 1983; Williams *et al.* 2001). SD and CG methods perform best when a quadratic approximation of the objective function is a relatively good one. In the context of molecular structure estimation, however, the quadratic approximations of the objective function can be bad and thus cause both methods to be ineffective.

This motivates consideration of Newton's method (NM) as a more practical choice for finding the local minimum. Newton's method has quadratic convergence when the initial guess is close to the solution and the curvature of the function is positive (Dennis and Schnabel 1996). The update of the atomic position is given by

$$\Delta\bar{x} = -[H(\bar{x}_{old})]^{-1} \nabla f(\bar{x}_{old}) \quad (2.4)$$

where H is the Hessian Matrix of f and ∇f is the gradient of f with respect to \bar{x} . Newton methods move to the local minimum in systematic way, but computing the inverse of the Hessian at each step usually requires an excessive amount of computational effort in comparison to other methods. A general way to avoid the computational complexity is to solve Equation 2.4 using the following system of equations:

$$[\hat{H}(\bar{x}_{old})] \Delta\bar{x} = -\nabla f(\bar{x}_{old}) \quad (2.5)$$

NM is known to have good local convergence to a minimum. The success of the NM depends on several factors: (1) the positive-definiteness of the Hessian matrix; (2) a descent direction; and (3) the initial guess must be in the correct neighborhood (Miller 2000; Dennis and Schnabel 1996). The Hessian provides important information about the curvature of a objective function, which accounts for most of the success of Newton's method (Ratchek 1988). However, if the quadratic approximation is poor, the positive definiteness of the Hessian is not always maintained, it can be ill conditioned. Further, the Hessian is sometimes unavailable or too costly to evaluate due in large to the degree of nonlinearity in the function. Furthermore, an initial guess that is not in a sufficient neighborhood of the minimum results in poor convergence (Williams *et al.* 2001).

In order to lessen these problems of positive definiteness and costly derivatives, Quasi-Newton methods are employed. In addition, global convergence is improved by combining existing local optimization with a line search algorithm. Quasi-Newton (QN) methods are developed to handle different kinds of curvatures (Miller 2000). Further, these methods avoid some of the convergence issues associated with Newton's method and reduce the amount of storage at each of the iteration. In these methods, the Hessian is computed by using secant approximations or by solving a nonlinear least squares problem. These methods are developed to search along the gradient line starting with a positive definite Hessian matrix and then using the information to build a quadratic fit to the objective function (Land *et al.* 1960). The update is generally given by Equation 2.5 where the Hessian matrix, H , is replaced by an approximation to the Hessian, \hat{H} . The main advantages of these methods are the efficiency in computing the step direction, numerical stability in the algorithm, and convergence to the local minimum.

There are many variations of the QN method that are successful in meeting this criterion, including Gauss-Newton (GN) and Broyden-Fletcher-Goldfarb-Shanno (BFGS). The Gauss-Newton (GN) methods are ideal for the least square residual approaches because the gradient and Hessian of the function has a simplified form that makes finding the local minimum more efficient. The basic idea of these methods is to linearize the objective function through use of second order TTE. The linear approximation is written as least squares problem and hence, the choice of the step direction is the solution to the linear least squares equation.

By doing this, the second derivative term contains a linear term and a nonlinear term. Further, the resulting Hessian matrices are symmetric, positive definite matrices (Dennis and Schnabel 1996). Depending of the degree of nonlinearity in the second term, the approximation of the Hessian is obtained by dropping this term. Guaranteeing local convergence for such a problem requires second order information about the function. Moreover, the convergence is on order of $O(n)$ for problems that are not nonlinear and have reasonably small residuals. Hence, one of the disadvantages of this update is the dependency on the residual size. That is, convergence is slow for sufficiently nonlinear or reasonably large residuals. Another disadvantage of this method is that when an ill-conditioned Jacobian matrix arises, it causes the search space to become very large (Erikson 1996).

Alternatively, the BFGS is the most successful secant approximation update methods for the Hessian matrix. In this method, the second derivative matrix is defined in terms of first derivative information and thus, causes second derivative to be less costly to construct (Dennis and Schnabel 1996). One of the main differences of this optimization method in comparison to other local optimization methods is that the Hessian matrix is also updated at each of the iterations. These updates produce symmetric positive definite Hessian matrix and thus ensure that the direction is also a descent direction. Moreover, these Hessian updates make it possible to avoid the direct inversion. Williams *et al.* (2001) showed that BFGS method in the context of the molecular structure estimation problem is more accurate and robust than the other variations of QN methods.

II.2.2. Line Search for Improved Global Convergence

In most of the local optimization methods discussed previously, convergence relies on how good the quadratic approximation is to the objective function. If the approximation is good, the optimization usually has good convergence properties because the algorithm tends toward a quadratic as it converges to the minimum of the function. For many nonlinear problem encountered in molecular structure estimation, however, the initial guess is generally not close to the optimal guess and the quadratic approximation is not good enough (Williams *et al.* 2001). Hence, the update produced by these methods is unsatisfactory; that is, there is no decrease in the function value from the old position to the updated position. Global convergence methods for local optimization, methods that converge from remote starting points, are needed (Williams *et al.* 2001).

The underlying concept behind using backtracking line search in the context of this local optimization problem is to combine a globally convergent strategy with an efficient local strategy to obtain better convergence to the local minimum (Dennis and Schnabel 1996). This is done by first taking a full SD, CG, or QN step. Mathematically, backtracking in a certain direction is performed by starting with $\lambda = 1$. If this step length fails to satisfy the criterion used, reduce λ along a line in the direction $\Delta\bar{x}$ from the configuration space until an acceptable update is found. The step length, λ , must satisfy the Wolfe conditions for global convergence (Dennis and Schnabel 1996). These conditions ensure that the step length is not too small and gives sufficient decrease in the objective function.

The step length λ is estimated by interpolation the function $f(\lambda) = f(x_k + \lambda\Delta\bar{x})$ at three points that bracket the minimum. These are defined by using information given about the function, such as the value of the function at $\lambda = 0$, its derivative evaluated at $\lambda = 0$, and the value of the function at $\lambda = 1$. The minimum of this interpolated function is chosen to be the ideal step length. If the next step length does not produce sufficient decrease in the function, the step length is reduced and the process begins again. This process is continued until a step length is chosen that results in a decrease in the function value and satisfaction of the Wolfe conditions. After the first iteration, more information is available about the function so it is advantageous to use a cubic model. Moreover, the cubic model can be accurately used to model the negative curvature of the function when

the quadratic method fails for two positive step lengths and the interpolant has a unique local minimizer under these circumstances (Dennis and Schnabel 1996).

II.2.3. Molecular Distance Geometry

Crippen and Havel (1988) showed that given a full set of distances, the set of atomic coordinates can be determined by creating a distance matrix with rank less than or equal to three, and using its eigenvectors to find spatial position of all the atoms in the molecule. In principal this approach is ideal; however, in practice, using distance geometry techniques for unknown sequences has proven to be a nontrivial task because a full set of distance data is not available from experimental techniques. Much research has gone into developing methods for computing these missing distances.

Hendrickson (1991) first proposed the molecular distance geometry problem as a global least-squares problem. Using a graph theoretic approach to find the optimal configuration for a bounded set of distance data, he showed that a large optimization problem can be reduced to a sequence of smaller ones and thus, lead to a substantial reduction in overall computational effort and a good quality structure. In his method, the molecule is represented by a graph with vertices that correspond to the atoms and an edge connecting two vertices if the distance between the corresponding atoms is known. The graph is then broken into subgraphs to reduce dimensionality and make the problem more tractable. For each subgraph, a local optimization is performed in which the subgraph is deformed so that the atomic positions satisfy a given objective function. The solutions are recombined to determine the optimal configuration.

Many other research groups have approached protein structure estimation through this same top level of Hendrickson's global least squares; however, the mathematical basis has found its home in solving algebraic equations or standard nonlinear optimization methods (More' and Wu 1997; Dong and Wu 2003; Wu and Wu 2007). These methods have provided valuable insight into how to obtain good estimations of some proteins. But, the solution can still be thought of as one solution out of many possibilities. That is, the solution does not definitively solve the distance geometry problem because a wide range of closely related configurations (*e.g. reflection, rotation, translation, etc.*) could be considered as possible solutions. Hence, the resulting

configuration can be considered local minima or starting point for many other optimization algorithms.

II.3. Overview of Global Optimization

Many objective functions adapted in protein structure estimation are generally nonsmooth and/ or nonconvex. Hence, minimizing the objective function is generally a hard task because of the presence of local minima. The complexity of solving the global optimization problem is dependent on the size of the region of attraction of the global minimum in relation to the feasible region, the affordable number of function evaluations, the embedded or isolated global minimizers, and the number of local minimizers (Torn 1989; Dennis and Schnabel 1996).

For nonconvex objective functions, nonconvex constraints, or a combination of both, we cannot be certain that the optimal solution is actually in the set of possible solutions or that the solution can be determined in a finite number of steps (*i.e.* solvability), or that the problem can be solved for higher dimensions. Hence, choosing points, solvability and stopping conditions are very important elements in global optimization. Standard methods for dealing with the unsolvability of a global optimization problem are twofold. One of the basic methods is to turn the problem into a solvable one or at least make it possible to tell for sure that a solution has been found by posing *a priori* conditions on the objective function and the feasible region.

Determining that a nonconvex problem is infeasible, that the objective function is bounded, or that a local minimum is the “global minimum” across all feasible regions can take an exponential amount of time; as a result, the solvability requirement is generally relaxed and an estimate of the global minimum is an acceptable solution (Torn and Zilinskas 1989). Stopping conditions are used to measure the quality of the solution after a given number of function evaluations are made. The stopping criterion is derived from additional information and assumptions about the sampling of the region are based on heuristics about the physical application.

Standard global optimization methods adapted for molecular structure estimation fall into two main categories: heuristic and exact methods. For heuristic methods, the general approach taken is to pick points that cover the feasible region such as uniform sampling and random sampling, successively solve a series of local optimization

problems and, pick the absolutely best optimum from that feasible region based on stopping conditions and solvability conditions. Probabilistic methods, smoothing and continuation methods, and problem-specific heuristic methods are common search methods used in molecular structure estimation. In the first group, simulated annealing and genetic algorithms are among the most commonly used in molecular simulations. These methods are designed to randomly perturb the state variables when trapped in local minima (van Laarhoven and Aarts 1987; Hiroyasu *et al.* 1998; Barbosa *et al.* 2003). Smoothing and continuation methods have been employed in finding the global minimum based on smoothing out as many of the low-error local minimum through transforming the objective function into smooth and, sometimes, convex function (Wu 1997; More' *et al.* 1997).

II.3.1. Random Sampling and Stochastic Methods

Random sampling and perturbation methods are methods that search in the neighborhood of the optimal solution to date and jump to a new solution whenever the algorithm indicates that another configuration has been found that improves upon the optimal solution to date. Simulated annealing (SA) and genetic algorithm (GA) are the most commonly used probabilistic methods in molecular structure estimation. These perturbation methods are based on simulating evolutionary processes found in nature (van Laarhoven Aarts 1987; Hiroyasu *et al.* 1998; Barbosa *et al.* 2003). SA avoids the problem of becoming stuck in local minimum by defining a temperature-dependent jumping function based on statistical theories of cooling a physical system. A solid in a heat bath is heated up to maximum temperature. Thermal equilibrium is reached, *i.e.* the molecules have reached a low energy state upon gradually cooling of the system as the temperature reduces slowly toward zero. However, if the solid is allowed to cool quickly or instantaneously, defects can be frozen into the solid (van Laarhoven and Aarts 1987). Hence, an annealing schedule, which slowly cools the system, is essential to drive the system to thermal equilibrium.

Acceptance of the modified configuration depends on whether thermal equilibrium has been reached. Thermal equilibrium is characterized, mathematically, by the probability of being in state with energy E given by the Boltzmann distribution,

$$P(E) = \frac{1}{Z(T)} \exp\left(-\frac{E}{K_B T}\right) \quad (2.6)$$

where T the temperature, K_B is defined as the Boltzmann constant, and $Z(T)$ is the partition function, depending on the temperature T . This formula states that at temperature T , thermal equilibrium has its energy probabilistically distributed among all different energy states E .

Simulated annealing is a global optimization method that was primary developed to search the space by randomly updating a sequence of initial configuration. However, Li and Scheraga (1997) showed that random perturbations, in particular for SA, perform best when allowing a full local optimization following each random step. Hence, the basic procedure is for SA is to start with an initial configuration obtained from solving a local optimization. The configuration is perturbed by a small amount depending on the attributed error of the current configuration and a random numbers between $[-1,1]$. The local optimization is then rerun using the perturbed configuration. This modified configuration is checked and accepted based on the probability of acceptance. Let $\Delta C_{kj} = C(k) - C(j)$. If $\Delta C_{kj} \leq 0$, then $P_{\text{accept}} = 1$, that is, a lower minimum has been found. However, if $\Delta C_{kj} > 0$, the decision to move or stay at the old position is made with probability of $P_{\text{accept}} = \frac{1}{Q(T)} \exp\left(-\frac{\Delta C_{kj}}{T}\right)$. If the modification is accepted, the new configuration replaces the old configuration; otherwise, the old configuration is kept. The algorithm continues in this process until the Boltzmann distribution converges to the uniform distribution or until some termination criteria has been reached. The temperature is then decreased and the process discussed above starts again with the new temperature. The process continues until a lower temperature bound is reached.

When the temperature is high, the algorithm is allowed to randomly search as much of the configuration as possible. Solutions that do not improve to date have as much of chance of being selected as those that do improve to date. Accordingly, the algorithms can sample more of the local space because uphill as well as downhill jumps are allowed throughout the optimization process. As the search proceeds, the temperature decreases in steps by using an annealing schedule with the system being allowed to

approach equilibrium. As a result, downhill jumps occur less frequently. Eventually the jumping mechanism freezes and SA completes its search like a simple hill climber.

Simulated annealing can be described as placing a probe in an area and allowing it to move randomly out of local minima. However, one of the main disadvantages of SA is that the randomization process allows the probe to move anywhere in the configuration space. Therefore, it is possible that perturbations can occur at times when the optimization is moving toward the global minimum (Williams *et al.* 2001). Further, various *ad hoc* enhancements have been added to make it much faster such as combining SA with other optimization techniques to find better possibilities of configurations because simulated annealing can become exceedingly slow (Neumaier 1997; Pokala and Handel 2001). These improved methods include combination with other optimization methods.

Alternatively, genetic algorithm (GA) is another probabilistic global optimization algorithm commonly used in molecular structure estimation because of its effectiveness in searching a large and complex configuration space. Moving out a local minimum toward the global minimum state requires using perturbation and acceptance strategies based on three genetic operations: selection, crossover, and mutation. Selection is an operation that imitates the survival of the fittest in nature. This operation selects configurations from a population of initial configurations that will be used to produce a better configuration based on their function values. Crossover is an operation that imitates the reproduction of living creatures. The crossover operation combines the atomic positions of the two chosen configurations to produce a better solution. In order to keep from mating configurations that are not ideal, new information about the atomic positions is commonly introduced. This occurs through randomly mutating atomic positions of the offspring.

Genetic algorithms start from a random initial population and select configurations with the best function values. The crossover operation produces offspring of the selected configuration; further alterations are achieved through mutation. Acceptance of modification is based on selection operator. The fitness of each offspring is evaluated, and a subset of offspring is used to replace the parent population. A random process that favors better fitness values selects solutions. The solutions are then mutated

to process better offspring. The process is repeated for a given number of generations or until some stopping criteria is met.

Algorithms that allow stochastic and random perturbations tend to be very effective in searching a wide range of the possible configuration space. Although SA and GA are effective methods for searching a large portion of the configuration space, both require knowledge about the system being modeled to effectively search the space. These methods are limited because the number of independent variables in typical molecular systems can cause the number of minima to become very large and thus, cause inefficiency in the algorithm (Goodfellow 1992).

Instead of developing stochastic methods to search as much of the configuration space as possible, other global optimization methods including smoothing and continuation have been developed which transform the function into a smoother function. Further, local optimization procedures are used to find the global minimum of the transformed function which will be used as a near-optimal solution or to trace the solution of the transformed function back the global minimum of the original function.

II.3.2. Smoothing and Continuation

Instead of accepting the function as a non-smooth function and applying random perturbation methods to effectively search the space, alternative methods for reducing the complexity of the global optimization problem are based on transforming the objective function into a smooth function to ease the task of finding the global minimum. Smoothing and continuation methods are standard methods used to find the global minimum through transforming the objective function into a function with fewer local minima by using an integral transform or differential operator. These operators are used to define a family of functions that is parameterized over a smoother parameter. By varying the smoothing parameters, one can create a series of functions that gradually smoothes the original function. The number of minima is reduced gradually as the objective function becomes smoother. Ideally, the smooth problem will have only one minimizer once the degree of smoothing is sufficiently large and ideally, that minimum will be the absolute minimum (Neumaier 1997; More and Wu 1997)

Several well-known variations of the smoothing technique include using Gaussian transform integral equation, diffusion equation methods, and convex global

underestimator. Currently, integral transform is one of the most commonly used variations of smoothing (Wu 1997; More' and Wu 1997; Neumaier 1997; Pokala and Handle 2001). The transformation of the objective function is given by the following integral transform

$$F_t(X) = \int_{-\infty}^{\infty} K(X, X_0, t) F(X) dX \quad (2.7)$$

where the kernel of the integral equation is defined by the Gaussian transform is given by

$$K(X, X_0, t) = \left(\frac{t}{\pi}\right)^{\frac{1}{2}} \exp[-t(X - X_0)^2] \quad (2.8)$$

where t is the smoothing parameter.

As t approaches zero the Gaussian becomes flat and the result of the integral becomes dependent on the original variable, making the transformed function completely constant and smooth. As t increases, the Gaussian becomes very narrow and the smooth function is identical to the original function. More' and Wu (1997) used the Gaussian smoothing method to estimate the structure of small molecules. Further, using this method gives a good estimate of configuration with less function evaluations. However, one of the main problems associated with this variation of smoothing is that it can yield the wrong global minimum due in large to change in the ordering of the minima as the smoothing parameter varies. Moreover, another problem associated with this smoothing variation is that it is computationally expensive to evaluate integrals.

An alternative form of smoothing is to represent the integral transform as a diffusion differential operator. This method modifies the objective function, by including a time variable, t . The transformed function, $\hat{E}(x, t)$, can be obtained through solution of the diffusion equation given by

$$\hat{E}_{xx}(x, t) = E_t(x, t) \quad (2.9)$$

with the boundary condition, $\hat{E}(x, 0) = E(x)$ which recovers original function. For large of values of t , the surface should provide one minimum. As t is gradually decreased, a sequence of local minima can be traced back to a local minimum of the original function at $t = 0$. The methods are useful mainly when the potential is a sum of univariate functions of distances between atoms (Neumaier 1997). These simplified models are

useful in finding the global minimum of the problem because they give rise to the low-error structures.

Although this method is globally convergent, it may not find the solution if there is a lot of noise in the function. Moreover, the solution only converges to the global minimum if the initial guess is close. Because the smoothing technique is not always reliable, a closely related method is to apply the smoothing procedure to the functions successively, and at each step take the solution for the previous function as the starting point for minimization of the current function. These methods are referred to as continuation methods. Continuation methods are different from smoothing in that the functions are not arbitrarily deformed functions; they are approximations of the original function in the sense that they are coarse estimates (Wu 1997). The kernel is given by the following expression

$$K(X, X_0, t) = \frac{1}{\pi^{\frac{n}{2}} t^n} \exp\left[-\frac{\| (X - X_0) \|^2}{t^2}\right] \quad (2.10)$$

Once the solution to the transformed problem is obtained, an optimization procedure is applied to the functions successively, to trace their solutions back to the original function.

One of the main advantages of this technique is that it could give insight into the problem of how molecules change from arbitrary configurations into the three-dimensional structures that dictates the biological functionality of the protein (Wu 1997). The backtracking procedure is done in several ways: use a general random search procedure to trace the changes of the global solution when the transformed function is gradually changed to the original function, apply only local optimization procedures to each transformed function to trace a set of solution curves and choose the best among all solutions obtained, or solve the initial value problems for a set of solution curves and chose the best solution. This method still contains some of the problems of the smoothing method such as dependence on an initial good guess. Further, the method is designed to trace the stationary points of the function, but the desired stationary point may not be the global minimizer of the function.

II.3.3. Convex Global Underestimator

The most recent smoothing technique for transforming the objective function is the convex global underestimation (CGU) method. This method is designed to find all

known local minima with a convex function, which underestimates all of them, but differs from them by the minimum possible amount in the discrete L_1 norm. This function is used to localize the search in the region of the global minimum. The set of local minima fitted to a convex function are collected from a large number of molecular configurations that are each attained from a random point. The minimum of the CGU function is accepted as a possible global minimum and used to find additional local minima on a reduced space. These additional local minima are added to the original set of local minima and used to create another underestimation on a reduce space and then, another possible global minimum is located. The process is continued until there is no local minimum in a given set that can be chosen as an improved local minimum; the space cannot be reduced any further. The value at the last iteration is used as the global minimum of the objective function (Dill *et al.* 1997).

One of the main advantages to this method is that it does not require as much computation as other methods; *i.e.*, evaluation of the function and its derivatives are reduced. Moreover, the method works well for problems of moderate size protein; *i.e.*, proteins with less than thirty amino acids. However, this method depends heavily on computing local minima quickly and on solving the resulting linear program efficiently to approximate function over the current hyperrectangle domain. Implementation of this method works much better with massively parallel machines because it effectively reduces the time required in computing the large number of local minima (Phillips *et al.* 1995).

Although the main purpose of many of these global optimization methods is to make it easier to obtain low error configurations, transforming objective function has the adverse effect of creating many low-error configurations that do not satisfy physical conditions. Many of the standard global unconstrained optimization methods discussed in this section have been altered by adding constraints on the atomic positions to not only produce low error configuration but to distinguish between them to determine the optimal low error configuration. Hence, the current challenge is to create an efficient global optimization algorithm that distinguishes between the low-error configurations while satisfying physical constraints (Williams *et al.* 2001). This requires developing more

effective perturbation methods and acceptance strategies for moving out of a state of local minimum.

II.3.4. Branch and Bound Method

Branch and bound methods are currently the most commonly used deterministic methods for estimating the three-dimensional structure of a protein (Floudas 1999; Zhou *et al.* 1999). The underlying idea is based on the assumption that the objective function and feasible regions can be transformed into a solvable problem based on *a priori* condition about the function and the constraints and certain biological assumptions about the optimization parameters (Torn 1989; Neumaier 1997; Floudas *et al.* 1999). By adapting these methods, a global optimization is theoretically guaranteed to yield a solution to the protein structure estimation problem with a given accuracy or to show whether a feasible solution can be determined on the specified region (Torn and Zilinskas 1989). The underlying assumption is that the global minimum of a complete structure is a combination of a relatively small number of local minima of configurations on the restricted region. Hence, deterministic methods can then be applied to problem to find the global minimum. The basic idea of these techniques is similar to basic idea of the bisection root finding method. That is, the method reduces the space by continually halving the region until a good lower and upper bound is determined for the global minimum.

The basic idea is to formulate the molecular structure estimation problem as box constrained nonlinear twice-differentiable optimization problem (Klepeis and Floudas 1999). On this box region, the assumption is that the global minimum of a complete structure is a combination of relatively small local minima. Hence, the α BB method brackets the minimum between converging lower and upper bounds. Upper bounds on the global minimum are obtained by local minimizations of the original energy function. Lower bounds belong to the set of solutions of the convex lower bound functions that are constructed from augmenting $f(x)$ with the addition of separable quadratic terms. The general form of the branching function is given below.

$$L(x) = f(x) + \sum_{i=1}^n \alpha_i (x_i^L - x_i)(x_i^U - x_i) \quad (2.11)$$

where x_i^L and x_i^U correspond to the lower and upper bounds on x_i . The variables are chosen to correspond to the dominant variable in the original objective function. The lower and upper bound are refined by successively partitioning the initial rectangular region into smaller ones. As result, a nondecreasing sequence of lower and upper bounds is generated. At each of the iterations, the lower bound of the objective function is the minimum of all the minima in every sub rectangle composing the initial rectangle. Only the subregions that contain the lowest minimum are halved at each iteration. If a single minimum in any of the subrectangles is greater than the current upper bound, the region containing the minimum is ignored because the global minimum cannot be found inside (Floudas *et al.* 1999).

The main advantage of this method has in comparison to other standard optimization methods is that in principal it is guaranteed to find the global minimum of the objective function. The criteria being use are sufficient conditions for global optimality for finding a global minimum of a convex function on a convex domain (Neumaier 1997). The key is that the branching function transforms the objective function into a convex function through defining the values of the control parameters, α_i 's. The measure of the degree of convexity of a function is determined by looking at the most negative value in the Hessian of the branching function. The underlying idea is to transform the Hessian of the original function to a positive semi-definite Hessian by shifting the diagonal elements of the Hessian by the values of α_i 's. The effect of the term is to overpower the nonconvexities of the original function, which ensures that $L(x)$ is a convex. The choice of the control parameters is chosen based on how convex the objective function is.

Further, the lower bound function possesses important properties, which guarantee global convergence. L matches the objective function at each corner and it is convex in the current box constraints (Floudas *et al.* 1999). The properties of the function as discussed with an efficient partitioning scheme make it possible to construct a global optimization algorithm guaranteed to always possess convergence to global minimum through the solution of a series of convex nonlinear optimization problems. Although the method can actually prove optimality of the best local minimizer, the method requires an

exponential amount of work (Zhou and Abagyan 1999). In addition, this method assumes the global minimum of a complete structure is a combination of a relatively small number of local minima of configurations. However, the global minimum of a complete structure may contain fragments far from their local minimum. Lastly, the efficiency of the method depends critically on the effectiveness of the branching and bounding algorithm used (Land *et al.* 1960).

II.4. Constrained Global Optimization

Constrained global optimization seeks to find the values for the optimization parameters that satisfy constraints while optimizing the objective function. A constraint is a limitation on an optimization parameter and is determined by the physical nature of the problem. For example, if the optimization parameters were atomic locations in a molecule, placing minimum separation distances between atoms would make sense because we know from biology that atoms tend to attract or repulse each other based on their polar nature.

In molecular structure estimation, physical constraints are used to move atoms into regions that do not violate known physical conditions. The most common method for incorporating additional information into the model is to introduce physical constraints into the objective function. The atoms are allowed to move around freely in the configuration space. When the atoms are placed in regions that violate physical constraints, a penalty or very large value is assigned to the objective function. In the limit that the penalty is large compared to the rest of the function, the constraints will eventually be satisfied if possible though minimizing the objective function (Sali and Blundell 1993).

The main advantages of this method are that the unconstrained optimization formulation is maintained and the number of variables does not increase with the transformation (Pike 2001). However, some of the drawbacks of defining constraints in this manner are that the derivatives may be difficult to compute as well as store, more barriers are created making it harder to move out of a state of local minimum, and thus results in inefficiency in the algorithm (Ratchek and Rokne 1988; Williams *et al.* 2001).

An alternative approach that avoids some problems associated with penalty function is to turn physical constraints on when they are needed and off when they are not

needed during the course of the optimization process. However, this results in alternating between low error structures that do not satisfy constraints and high error structure that do satisfy constraints, without being able to find the optimal low-error configuration (Williams *et al.* 2001). Further, one successful method for introducing physical constraints into a global optimization formulation has been to build a model consisting of a simple objective function with physical constraints on the atomic locations (Monge *et al.* 1994).

One of the goals of this work is to improve a constrained global optimization method for atomic distances introduced by Williams *et al.* (2001), referred to as GNOMAD. Because of its ease in enforcing physical constraints, GNOMAD is guaranteed to produce configurations that satisfy physical constraints such as van der Waals and chirality. In the next section, we describe a brief overview of the GNOMAD technique and identify modifications that will improve the quality of the estimates of three-dimensional structure.

II.5. Overview of GNOMAD

The GNOMAD computer program is a multi-purpose molecular structure estimation program that implements a novel constrained nonlinear optimization algorithm to find optimal atomic configurations that satisfy given input data and maintain known physical constraints (Williams *et al.* 2001). GNOMAD was originally developed as an efficient and accurate algorithm to provide better quality protein structures using distance information complemented with a van der Waals distance constraint that requires that the distance between atoms be at least 4 angstroms (Å) and at most 6 Å (Finkelstein 2007).

The GNOMAD program has several advantages. First, the atom-based approach ensures that the optimization is always performed in three-dimensional space, which ensures that the algorithm yield structures that satisfy van der Waals (VDW) conditions. Second, the combination of the VDW constraint with the nonrandom perturbation method provides a synergistic effect moving atoms to their locally optimal value and keeping them from getting stuck in local minima. Lastly, the build-up procedure make it possible for the update process to always begin with an optimized subset of atoms (Williams *et al.* 2001).

The remainder of this subsection describes the GNOMAD computer program. Section II.5.1 briefly summarizes the general structure of GNOMAD proposed by Williams *et al.* (2001). Section II.5.2 presents the main chain atomic representation that is used in GNOMAD. Section II.5.3 gives an overview of the previous physical constraints that have been implemented in GNOMAD to improve the quality of protein structures. Section III.5.4 gives an overview of the GNOMAD algorithm and discusses some of the algorithms limitations and some suggestions for improvement.

II.5.1. General Structure of GNOMAD

GNOMAD uses a multi-level optimization process to build up a protein structure. The protein is built by adding one atom, or a group of atoms, in sequence. The process for updating positions of atoms within a group consists of a series of single-atom updates, instead of a simultaneous update of all atoms in the subgroup. At the lowest level of our algorithm, after all the atoms have been arranged so that the objective function is minimized, there is a check for violation of physical constraints.

The general approach to enforcing physical constraints is to define regions where movement into the region will cause violation of biological conditions. Satisfying physical constraints often times results in moving into local minima. To correct this problem, a domain specific nonrandom perturbation method is also enforced during the optimization process. This method involves placing a perturbation sphere around the atom being moved. The radius of the perturbation sphere shrinks and expands proportionally to the function error associated with the atom. The physical constraints are effective in eliminating some of the low-error configurations that result from using a small amount of input data. In addition, the physical constraint together with the local perturbation method work to find the low-error configuration from a set of high error configuration in the cases where is a large amount of distance data (Williams *et al.* 2001).

II.5.2. Atom-based Optimization

At the highest level of GNOMAD, we want to find the optimal configurations of subsets of all the atoms in the molecule. This subset of atoms is referred to as a *group*. In estimating the main chain backbone structure, the algorithm starts by optimizing the first two atoms in the first amino acid. The subsequent optimization groupings (from 3 atoms up to n atoms) are formed by adding the next atom from the residue in the sequence. The

algorithm uses the optimized set of atoms together with a new atom that is positioned in the vicinity of the optimized set of atoms.

Within each of the optimization groups, we perform a series of *cycles*. The total number of cycles is determined by a user-specified limit or termination criteria. Each cycle consists of a series of atom moves, one for each atom in the current group. The atoms are moved based on decreasing order of attributed error and, the order is determined at the beginning of cycle. Within the move atom level, there is a series of iterations. At each iteration, as illustrated in Figure 3, the BFGS/cubic backtracking search local optimization procedure determines the movements of the atoms. The iterations continue until some stopping criteria are met.

For each atom a in the molecule, the iterations are based on a local objective function, f_a , given by

$$f_a = \frac{1}{2} \sum_{i=1}^{m_a} \left(\frac{d_i^c - d_i^e}{\sigma_i} \right)^2 \quad (2.12)$$

where d_i^c is i^{th} the calculated distances, d_i^e is i^{th} the experimental distances, and m_a is the total number of input distances associated with atom a . The 3-dimensional update equation is then given by

$$\vec{x}_{new}^a = \vec{x}_{old}^a + \alpha \Delta \vec{x}^a \quad (2.13)$$

where α is the step length, \vec{x}^a now refers to the state vector consisting of the x, y, and z coordinates of atom a . Equation (2.13) is the reduced dimension form of equation (2.3). The linear system of equations in the QN update, Equation (2.5), is reduced from dimensional $3n$ to 3 and is given by

$$\left[\hat{H}_a(\vec{x}_{old}^a) \right] \Delta \vec{x}^a = -\nabla f_a(\vec{x}_{old}^a) \quad (2.14)$$

where \hat{H}_a is an approximation to the Hessian matrix of f_a and ∇f_a is the gradient of f_a with respect to \vec{x}^a . \hat{H}_a is a 3×3 matrix and ∇f_a is a 3×1 vector.

II.5.3. Physical Constraint Enforcement

Implementation of GNOMAD algorithm rests on the central assumption that the most effective method for improving the resolution of the structure is to develop constraints based on physical information and use them in combination with nonrandom

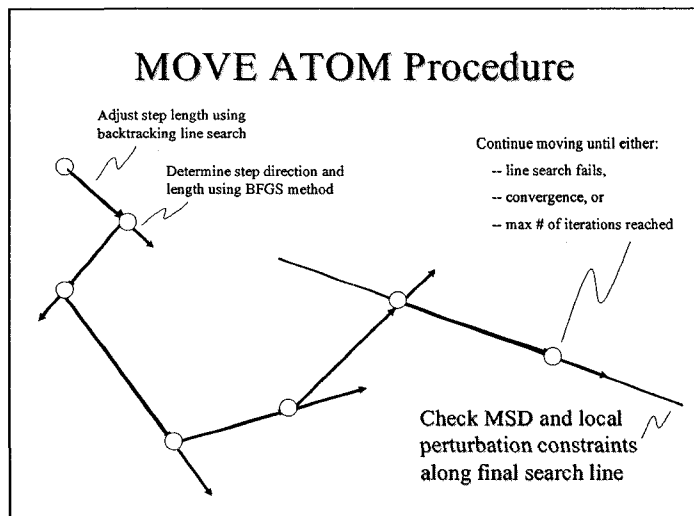


Figure 3. Diagram of atom move procedure. At each iteration, the BFGS/cubic backtracking linear search local optimization procedure determines the movements of the atoms.

perturbation method to drive the optimization to a near global minimum. From this it follows that accuracy of the three-dimensional structure should be reflected in how effective the physical constraints are in enforcing restrictions on the movement of atoms. These physical constraints are enforced in the final iteration of the local optimization algorithm to allow the atoms to sample a large portion of the space during the search and to keep the method from becoming too restrictive on the atomic position.

The advantage of enforcing the constraints in this way is that the best direction as determined by the local optimization is used in determining the location of the violation regions in the configuration space. Accordingly, the basic task of enforcing physical constraints is to determine how far in the optimal direction to move in order to ensure that atoms are not moved into any violation regions. This requires that the step length be updated in such a way that both the physical constraints are satisfied and the function has been minimized with respect to the physical constraints.

The general constraint enforcement procedure begins by placing minimum separation spheres around the specified points after the local minimization procedure has been terminated. We then determine the possible location of the atom by looking for line segments resulting from the intersection of the search line with the spheres. These line segments are merged and a violation check is performed to determine whether the atom is in the merged region. If the atom is in the region, the step length is updated to move the

has atom out of the violation region while maintaining the step direction, as shown in Figure 4.

The general form of the line is given as follows:

$$L : (x, y, z) = (x_0, y_0, z_0) + \lambda(p_0, p_1, p_2) \quad (2.15)$$

where (x_0, y_0, z_0) is the previous atomic coordinate of atom, λ is the current step length, and (p_0, p_1, p_2) is the step direction. We find the line segments resulting from the intersection of the line with all the violation spheres. These spheres are given as follows:

$$S : (x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0 \quad (2.16)$$

where the radius is denoted by r and the center, (x_c, y_c, z_c) , is represented by user defined points³.

If the atom is placed in a violating line segment, the atom is moved out of the violating region by looking for possible updates in the step length that will place atoms in good regions. The possible updates in step lengths are determined by substituting Equation 2.15 into Equation 2.16. This results in creating a quadratic equation with step length, λ , as the independent variable. The resulting quadratic equation is given by

$$a\lambda^2 + b\lambda + c = 0 \quad (2.17)$$

where $a = p_0^2 + p_1^2 + p_2^2$, $b = [p_0\bar{x} + p_1\bar{y} + p_2\bar{z}]$, and $c = [\bar{x}^2 + \bar{y}^2 + \bar{z}^2 - r^2]$. In addition, $(\bar{x}, \bar{y}, \bar{z})$ is defined as the difference between the atomic positions and the centers of the sphere. From this equation, a collection of all the line segments, $[\lambda_i, \lambda_j]$ for $i \neq j$, are computed and considered as possible updates for the original step length. We merge all line segments, including those found from other physical constraint enforcement, together to take care of any overlap regions. We search through these updates to determine whether the original step length, λ , falls in any of the intervals. If this violation occurs, we move the atom by picking either the lower or upper bound step length of the violating segment, whichever produces a lower function error value.

II.5.3.1. Van Der Waals (VDW) Constraint

One of the most commonly modeled biological conditions in protein structure

³ The center varies depending on the type of physical constraint and is given in the explanation of each physical constraint.

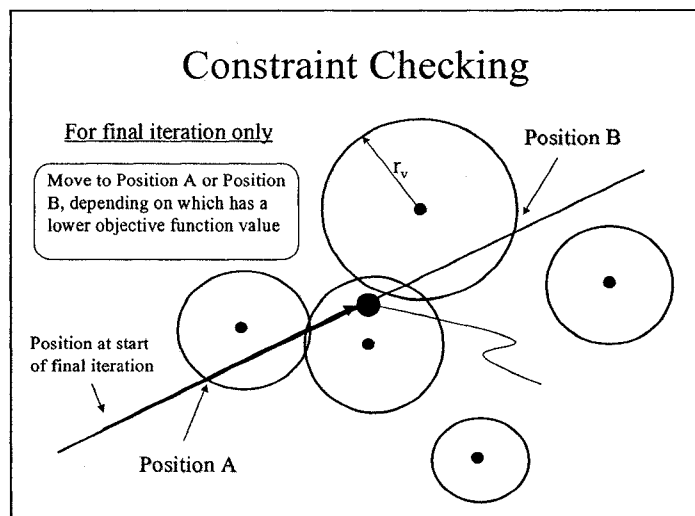


Figure 4. Diagram of physical constraint enforcement. Violation of physical constraints takes place at the end of the final iteration.

estimation is the van der Waals forces. Van der Waals forces are attractive forces that occur between atoms separated by distances of 4 – 6 Å (Finkelstein 2007). In standard optimization formulation, nonbonded interactions are generally modeled by adding a supplementary term based on van der Waals and electrostatic theory to the objective function (Neumaier 1997; Floudas *et al.* 1999; Pokala and Handel 2001). These more complex interatomic potentials can result in a more reliable model for searching of the configuration space (Wall *et al.* 1999). However, enforcing physical constraints on parameters or atomic positions while simultaneously updating the positions of a large number of atoms makes it very difficult to find an optimal configuration (Williams *et al.* 2001).

Alternatively, atom-based approaches have been developed in current research that places restrictions on atomic positions during the optimization process (Williams *et al.* 2001). Since the atoms are moved one at a time, enforcing VDW constraints is made easier. The VDW constraint follows the general constraint enforcement method described previously. An atom is treated as a point in space and spheres are placed around all the other atoms for which VDW constraints are being enforced. The radius is a user-specific value based on physical knowledge of nonbonded distances between atoms in a molecule. As a result, VDW constraints are guaranteed to be satisfied. Further, they serve as added information to reduce the number of low error local minimum (Williams *et al.* 2001).

II.5.3.2. Chirality Constraint

Another physical condition that is commonly modeled is chirality. Chirality is a molecular feature, which describes the left- and right-handedness orientation of substructures within the molecule, and is readily available information used in describing atomic positioning. Modeling this particular feature is important in drug design because the orientation of a molecule yields different behavior for the drugs designed from it. For example, Thalidomide is a chiral molecule used to design a sedative drug for pregnant women. Due to its chiral nature, however, the drug also has been found to cause fetal abnormalities when taken during a specific time of the pregnancy.

A configuration is “chiral” when it can exist in nature in at least two structural forms in which the spatial arrangements of its atoms are non-superimposable mirror images of each other (LeGuennec 1999). These mirror images are referred to as the left-handed and right-handed configuration of the molecules as shown in Figure 5. Most of the amino acids that make up the molecule are chiral. The left-handed and right-handed configuration of these amino acids can be determined by looking at the chirality center of an amino acid. An atom that has four groups bonded to it in such a manner that it has a non-superimposable mirror image characterizes a chirality center.

Several molecular quantities change signs under reflection and thus, are used to model chirality: dihedral angles, improper torsional angles, and volume⁴ (More' *et al.* 1999). Hence, the measure of left-handedness or right-handedness of amino acid is given by positive or negative value on these quantities, respectively. In standard molecular structure algorithms, the most common avenue for incorporation of this information is to add a term to the objective function describing the oriented volume of the four backbone atoms surrounding the C_{α} (More' *et al.* 1999).

The GNOMAD algorithm takes a different approach to enforcing chirality. A figure is chiral if it cannot be mapped to its mirror image by rotations and translations alone. The underlying idea for our chirality constraint is to restrict each atom from taking on incorrect positions that maintain symmetry of distances between atoms in the group; this is the mirror image position. Therefore, the plane of symmetry formed by this

⁴ The oriented volume is the value of the determinant of a matrix formed from a set of four atoms in the molecule.

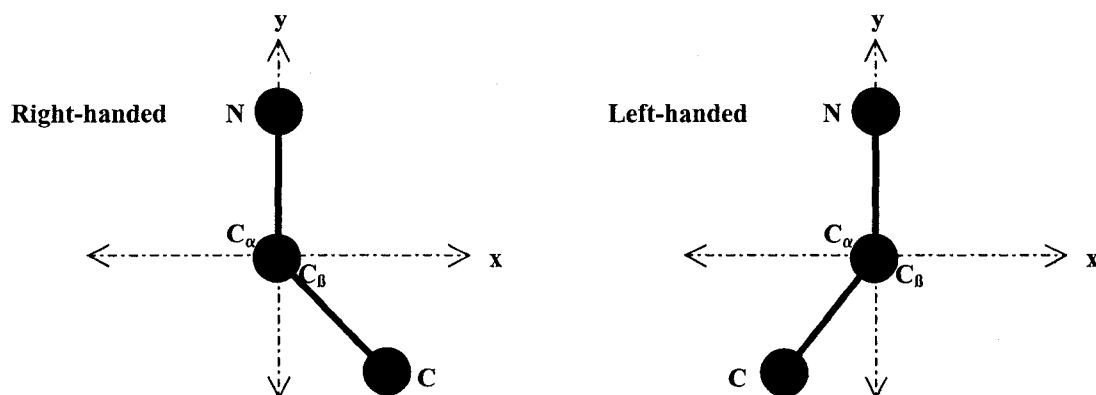


Figure 5. Example of the chiral forms of an amino acid. The C atom can take on two positions and still satisfy distances between the four atoms.

group of atoms aids in satisfying chirality conditions. Maintaining correct chirality is achieved by restricting the movement of N , C_α , C , and C_β atoms to the correct side of the plane of symmetry.

For example, enforcing the chirality constraint when a C atom is being moved involves determining the proper side of the plane to place the C atom, as shown in Figure 6. By looking down the bond vector formed by C_α and C_β atom, we see that the C atom can be placed in two distinct locations that will maintain satisfaction

of the distances between all atoms in the chirality group and the plane of symmetry coincides with the bond vector formed by N and C_α . Assuming we know what side of the plane the atom should be placed relative to the other chiral atoms, we can destroy any attempt for the optimization to constrain the atoms to the mirror configuration. This is done by placing a sphere around the point representing the mirror image of atom and then, restricting the atom from moving into the chirality-violation sphere (Figure 5). As a result, the distances will constrain the atom to the correct side of the plane of symmetry. The method for determining which side of the plane the atom is restricted can be approached in several different ways.

Since many of the amino acids found in nature naturally occur in the left-handed configuration, we can restrict the atom to the side of the plane that will satisfy this criterion (Cronin and Pizzarello 1997). If we are given information about torsional angles

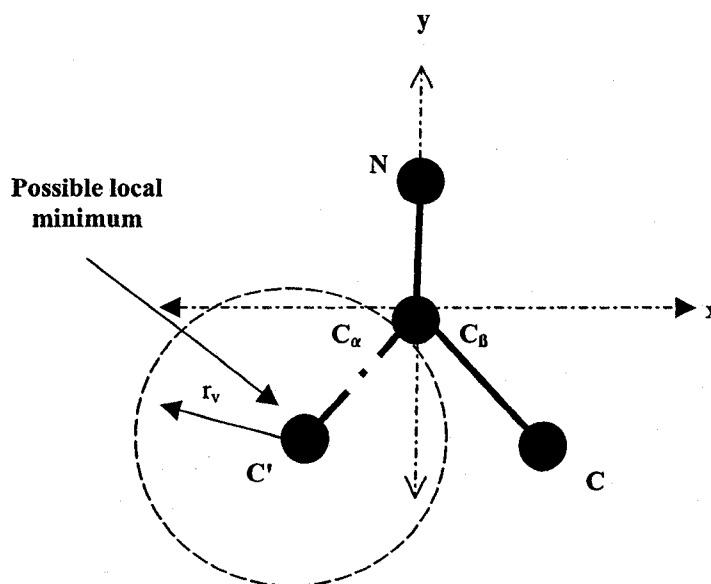


Figure 6. Example of enforcing chirality constraint. A chirality violation region is constructed by placing a sphere around the C' atom (*the mirror image of the C atom*) and the C atom is restricted from movement into this violation region.

and oriented volume, the atom can be positioned so that the sign of these rotational angles is consistent with the sign of torsional angles and oriented volume.

II.5.3.3. Side-Chain and Space Filling Properties

The physical constraints employed in GNOMAD require the presence of N , C_{α} , C and C_{β} atoms. This minimal set of four atom types provides a basis from which to construct an optimal atomic representation. By including these four atom types, we can maintain the constraint satisfaction properties of the GNOMAD algorithm. One of the main difficulties in eliminating atoms from a protein structure model is the adverse affect that it might have on the space-filling properties of the algorithm. By removing atoms from the model, the remaining atoms are then allowed to assume positions that would otherwise be taken by the missing atoms. The result would be a collapse of the overall structure. This type of structure collapse is often seen in molecular structure modeling, particularly in the case of sparse input data that leads to many local minima.

The problem of structure collapse was significantly alleviated by the van der Waals (VDW) constraint enforcement method in the GNOMAD algorithm (Williams *et al.* 2001). This method is able to avoid the many van der Waals illegal local minima and find a low-error solution that places atoms in better space-filling positions. The result is

improved root mean squared deviation (RMSD) for roughly the same optimization error. RMSD is a measure of how closely the model structure matches the true structure.

The VDW constraint enforcement method can be used to force separations between any atoms, whether the separation distances arise from van der Waals theory or from empirical estimates. Therefore, the VDW constraint enforcement method is used in this work to address the problem of structure collapse when using a reduced atomic representation and ensures adequate space-filling properties of the GNOMAD algorithm. This approach will be referred to as VDW constraint satisfaction although in this case it refers more generally to the satisfaction of minimum separation distances between pairs of atoms in the reduced model (not necessarily due to van der Waals forces).

Since the chirality enforcement method requires that most of the main-chain atoms be present (all but the O atom), the primary area where atoms can be removed is in the side-chains of the protein. But it is the side-chains that account for much of the space filling of the three-dimensional structure. Therefore, if side-chain atoms are removed from the model to increase the computational efficiency, then some allowance must be made to ensure that the overall volume of the molecule is roughly maintained. Overall volume of the molecule can be maintained with fewer side-chain atoms by setting minimum separation distances between the side-chain atoms of one residue and the sidechain atoms of other amino acids within the protein. The minimum separation distances could then be maintained using the VDW constraint enforcement method in GNOMAD. This approach would require a set of minimum separation distances that would be dependent upon residue type.

After testing several different reduced atomic representations, it was found that; (1) side-chains can be accurately represented by C_{β} atoms only, given the proper choice of minimum separation distances between C_{β} atoms, and (2) eliminating the O atoms in the main-chain representation does not result in a significant loss of accuracy. This minimal set of atoms – N, C_{α} , C, and C_{β} – forms the basis of the reduced atomic representation that can be used in conjunction with the constraint enforcement method (CEM) as shown in Figure 7. The next step is to construct a set of "optimal" VDW parameters (minimum separation distances) between all of these types of atoms.

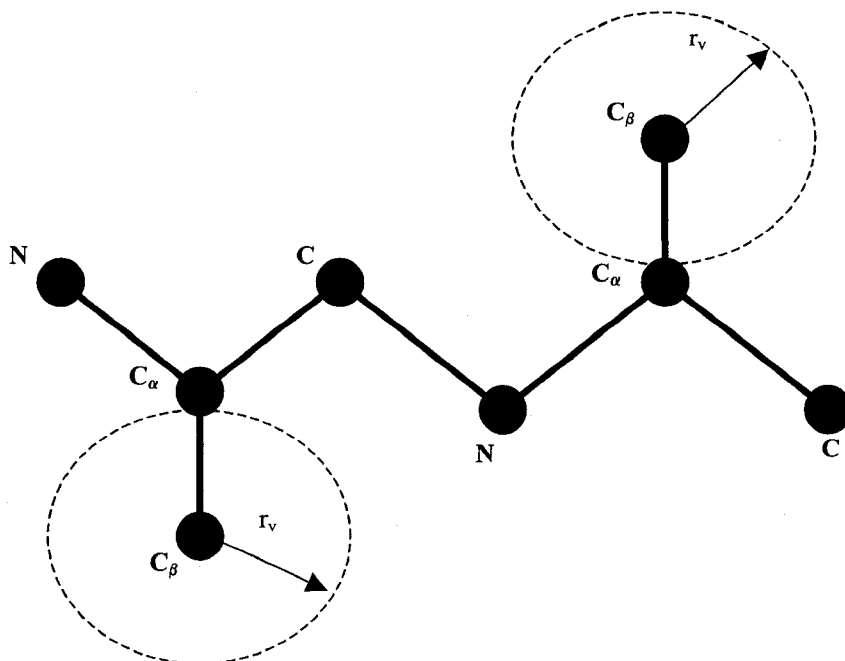


Figure 7. Main chain atomic representation consisting of the backbone atoms N, C_α, C, and C_β where C_β atom represents the sidechain. Spheres are placed around C_β atom to accommodate for deletion of other atoms in the sidechain.

One approach is to determine minimum separation distances between all possible combinations of atom types in the model. This would involve a large number of parameters because there are 22 different types of atoms -- N, C_α, C, and 19 different C_β's. Setting too many VDW constraints, however, will overly restrict the search space and interfere with global convergence. The VDW constraint enforcement scheme used in this work is based on using only a subset of all the possible minimum separation distances. The VDW constraint enforcement scheme used in this work consists of setting several different minimum separation distances, based solely on atom type. Table 3 gives a matrix of minimum separation distances between two atom types. First, a default minimum separation distance between any two atoms in the model is set. The default value used in this work is 1.0 Å. In addition, minimum separation distances are assigned between any two main chain atoms where both atoms are of the same atom type. Finally, minimum separation distances are assigned between any two C_β atoms based on the corresponding residue types. GNOMAD can then easily assign a specific separation

Table 3. Minimum separation distances between all possible combinations of atom types in the model. The initials of each amino acid are given for the corresponding C_β atom.

	N	C_α	C	ALA	ARG	ASN	ASP	CYS	GLN	GLU	HIS	ILE	LEU	LYS	MET	PHE	PRO	SER	THR	THP	TYR	VAL	
N	2.09																						
C_α	1.00	2.71																					
C	1.00	1.00	2.55																				
ALA	1.00	1.00	1.00	2.97																			
ARG	1.00	1.00	1.00	2.92	2.20																		
ASN	1.00	1.00	1.00	2.93	2.72	3.27																	
ASP	1.00	1.00	1.00	2.98	3.21	3.29	3.01																
CYS	1.00	1.00	1.00	3.06	3.19	2.85	3.39	2.89															
GLN	1.00	1.00	1.00	3.19	3.31	3.23	2.96	3.15	3.54														
GLU	1.00	1.00	1.00	2.89	3.15	3.31	2.91	3.33	3.02	2.93													
HIS	1.00	1.00	1.00	3.03	3.49	3.03	2.99	3.50	3.40	3.43	3.57												
ILE	1.00	1.00	1.00	3.21	2.13	2.15	3.34	3.25	3.29	3.37	3.66	3.58											
LEU	1.00	1.00	1.00	3.07	3.20	2.98	3.12	3.04	3.20	3.43	3.24	3.24	3.21										
LYS	1.00	1.00	1.00	2.99	2.52	3.22	3.20	3.19	3.40	3.20	3.33	3.31	3.50	3.14									
MET	1.00	1.00	1.00	2.72	3.02	3.61	3.02	3.44	3.61	3.62	3.65	3.52	2.88	3.28	3.67								
PHE	1.00	1.00	1.00	3.16	3.26	3.30	3.24	2.99	3.43	2.77	3.38	3.35	3.13	3.17	3.64	3.56							
PRO	1.00	1.00	1.00	2.85	3.44	3.05	3.06	3.33	3.45	3.31	3.52	3.57	3.49	3.31	3.50	3.23	3.51						
SER	1.00	1.00	1.00	2.72	2.54	2.42	3.17	2.97	3.32	3.05	3.08	3.29	3.21	2.97	3.39	3.17	3.30	2.89					
THR	1.00	1.00	1.00	3.02	2.58	2.98	3.24	3.38	3.07	2.89	3.43	3.57	3.28	3.37	3.54	3.30	3.01	3.24	3.57				
THP	1.00	1.00	1.00	2.80	3.63	3.54	3.61	3.42	3.29	3.21	3.65	3.55	3.04	3.45	3.49	3.33	3.48	3.31	3.63	3.62			
TYR	1.00	1.00	1.00	2.72	3.30	2.99	3.50	3.42	3.17	3.13	3.49	3.23	3.37	2.86	2.91	3.19	3.38	3.06	3.25	3.53	3.54		
VAL	1.00	1.00	1.00	3.03	3.26	3.03	3.47	3.12	3.02	3.22	3.31	3.72	3.19	3.18	3.35	3.37	3.22	3.08	3.20	3.34	3.32	3.01	

distance between each pair of atoms in the model, based on the type of atoms and the appropriate minimum separation distance value. For example, the minimum separation between any two N atoms in the main chain is 2.09 Å. The minimum separation distance between any two C_α atoms is 2.71 Å, and so on. The specific minimum separation distances represent optimal values as determined empirically through a statistical analysis of the distances between the N, C_α , C, and C_β atoms in all of the known protein structures found in the PDB Select list (Hobohm and Sander 1994).

A final implementation detail of the reduced atomic representation is the order in which atoms are introduced into the optimization. GNOMAD builds a model by optimizing a group of atoms, starting with two atoms and introducing an additional atom at the completion of each sub-optimization.

II.5.4. GNOMAD Algorithm

Figure 8 shows the pseudo-code for the GNOMAD algorithm (Williams *et al.*

GNOMAD Molecular Structure Algorithm

```

for natoms = 2 → total # of atoms in molecule {GROUPS}
  determine starting positions for 1 → natoms
  For cycle = 1 → # of cycles {CYCLES}
    determine order of atoms to move, based on attributed error
    (move atoms in decreasing order of error)
    for a = 1 → natoms (in order determine above) {ATOM MOVES}
      Perform nonlinear iterations to move atoms a {ITERATIONS}
      determine move direction using BFGS quasi-Newton minimization
      determine move length using quadratic/cubic backtracking line search
    perform MSD and Chirality constraint checks
  merge violating segments along search line into nonoverlapping segments
  If necessary, perturb final position of atom

```

Figure 8. Outline for GNOMAD algorithm for atomic distances.

2001). The methodology for GNOMAD gives insight into an alternative method for improving the resolution of three-dimensional structures by refining the model to include more physical constraints in the previously described manner. While GNOMAD produces favorable results for moderate size data sets, this algorithm needs improvement in two areas. This first one is related to the formation of major secondary structure, and the second to improving the capability of GNOMAD in handling non-distance information.

The current approach is to use distances to model hydrogen bonding and angles to characterize the geometry of major secondary structure. In the context of GNOMAD, however, both methods compromise global convergence. Torsional angles are a reliable source to describe the geometry of secondary structure, but are prone to local minima entrapment. Distances derived from regions where hydrogen bonding takes places tend to be short-range contact distances, which tend to be more difficult to satisfy. Improvements in the modeling of secondary structure may be achieved by developing an algorithm that better characterizes the geometry of the major secondary structure elements, such as α -helix and β -strand, and improves the formation of these local regions based on these geometric descriptions.

For the problem of handling non-distance information, we focus on enhancing the capabilities our global optimization algorithm to include surface, torsional angles, and contact data. Accordingly, the complementing of this information with an algorithm that is known to satisfy VDW constraints and chirality constraints will ideally produce a more accurate and reliable estimate of three-dimensional structure.

CHAPTER III

A PRACTICAL DISTANCE-BASED METHOD FOR SATISFYING TORSIONAL ANGLE INFORMATION

III.1. Introduction

GNOMAD is known to produce good quality structures when a large amount of distances are available and the VDW constraints are enforced (Williams *et al.* 2001). In some of these cases, however, the local regions are inaccurate and result in a large error metric (*e.g.* residual error). Usually, these large errors in local structure cause poor overall quality in structure. In order to improve formation of local regions, we look to find other types of readily available structural data describing these local regions. One group of information that is used to geometrically characterize local substructures is the sequence of backbone torsional angles (*also referred to as rotational or dihedral angles*). Although this information cannot be used to construct the whole three-dimensional structure, the backbone torsional angles are powerful in determining the main features of the final geometric shape of the folded protein (Neumaier 1997; Xue *et al.* 2007).

The use of torsional angle information in molecular modeling is of grave importance because it provides a remedy for accurately forming local substructures within the protein. The task of using angles in a global optimization formulation seems to be a trivial problem. It is simple enough to just develop a objective function based on torsional angles and use the function as a term in the objective function to appropriately describe protein structure. In the context of atom-based approaches, however, a major obstacle in using this methodology is that these types of algorithms are contingent upon the process of moving atoms (one at a time) based on attributed function error (Williams *et al.* 2001). Constraining torsional angles while moving atoms based on attributed function error produces much exhaustive work; but does not result in moving any closer to finding the correct optimal configuration.

Alternatively, we could use the approach of determining a representative set of distances from torsional angles and combine these distances with the short-range contact

distances. There has been much work done on using distances to accurately construct local substructure of the protein, including ladder-grouping method by Aszodi and Talyor (1994), helix-packing algorithm by Munenthaler and Braun (1995), and disjunctive constraints by Chen (2000). Whereas distances have been effectively used to form these local regions in their work, these methods are not effective in the context of GNOMAD.

In our work, we set out to answer the question, “How can we ensure that distances can be used to satisfy torsional information?” In the spirit of improving our constrained global optimization approach for distances, we propose a method for using a minimum set of distances to satisfy torsional angles. This requires that we employ a more elaborate representation in specifying a core set of inter-atomic distances for constraining torsional angles.

III.2. Backbone Torsional Angles

A torsional angle is formed from four atoms A-B-C-D (we will refer to this group of atoms as torsional angle group) as shown in Figure 9. It is the angle between the planes formed from atoms A, B, and C and the plane formed from B, C, and D. An alternative and easier definition is to describe the torsional angle in terms of its rotational property. Looking down the bond vector BC, a torsional angle is the clockwise angle made by the vector AB and CD.

In molecular structure estimation, we have at our disposal information that describes the molecular geometry, such as distances and angles, of three-dimensional structure. A very important factor in satisfying torsional angles is the knowledge of what distances are directly controlled by the torsional angle. In Figure 9, we see that changing the torsional angle ϕ changes only the distance between the base atom A and the base atom D. All other inter-atomic distances between the base atoms – A, B, C, and D – are constrained by the chemical bond lengths and bond angles.

In our work, we are interested in the three torsional angles along the main chain of a protein, which are also referred to as backbone torsional angles, denoted by the Greek letters ϕ , ψ , and ω as shown in Figure 10. The angles are defined on the specific atom types defined in torsional angle group. The ω torsional angle group is computed using backbone atoms C_{α} , C, N and C_{α} , from the next consecutive angle and controls the C_{α} -

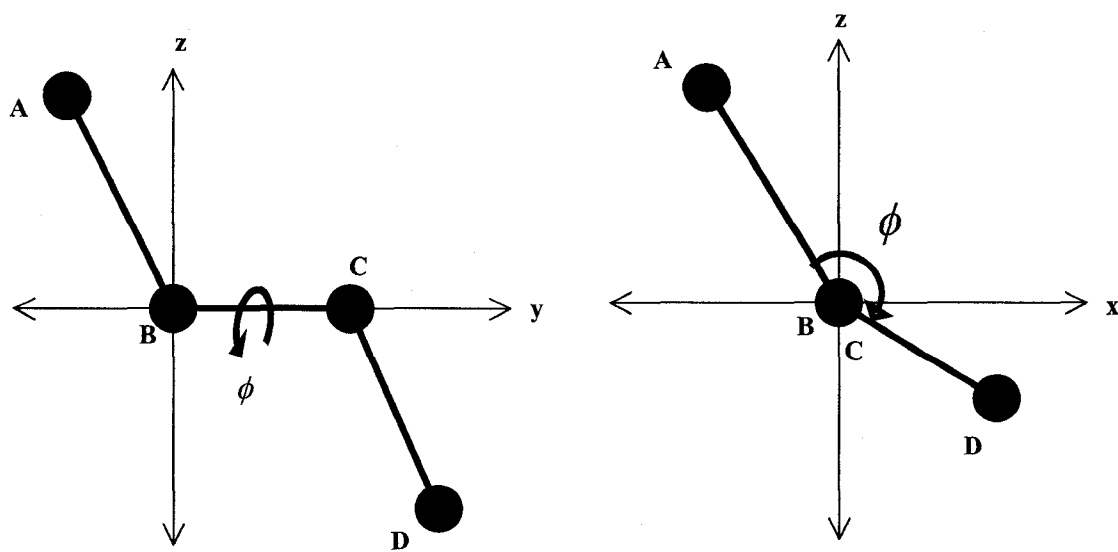


Figure 9. Definition of the torsional angle ϕ using main chain atoms.

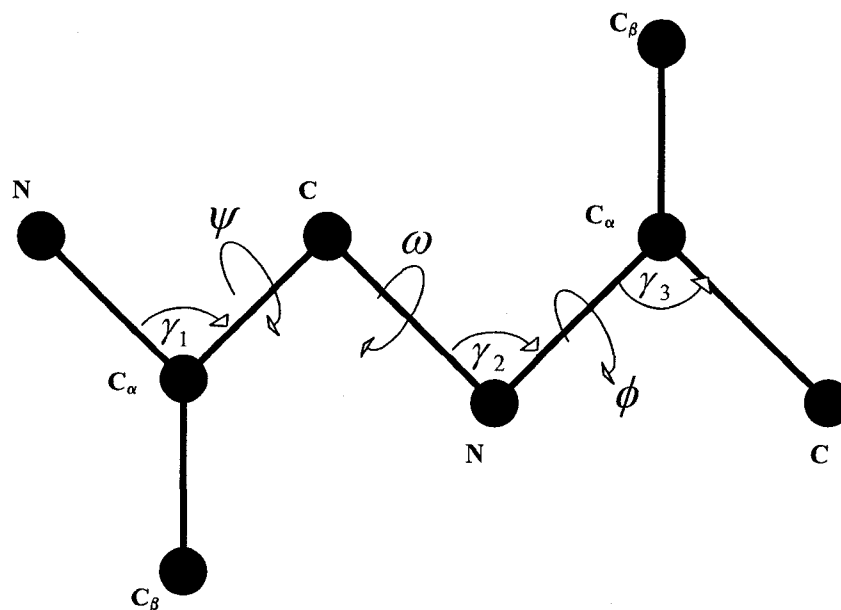


Figure 10. The geometry of the general polypeptide chain. The chain was composed of the backbone atoms N, C_{α} , and C joined by bond angles, γ_i , for $i = 1, 2, 3$.

C_{α} distance. This torsional angle is at the peptide bond and can be described in two possible conformations: the *trans* form for which $\omega=180^{\circ}$ and the *cis* form for which $\omega=0^{\circ}$. For non-proline amino acids, the *trans* conformation is usually observed. Alternative, the *cis* conformation is observed about 5.6% of the time in proline amino acids (Laiter *et al.* 1995). Even so, in modeling effort, the value of ω is usually taken to be 180° . Hence, the flexibility of the protein can almost be described by the remaining backbone torsional angles, ϕ and ψ .

The ϕ torsional angle is computed using the backbone atoms C, N, C_{α} , and the C from the next consecutive residue and controls the C- C distance. The ψ torsional angle is computed using the backbone atoms N, C_{α} , C, and N from the next consecutive residue and controls the N-N distance. The relationship between these angles helps in determining which regions of the protein consist of major secondary structures, such as α -helices and β -strands. These relationships can be seen in the protein's Ramachandran plot and has been generalized through a set of mathematical inequalities (Neumaier 1997). The exact values of the ϕ and ψ can be determined or predicted from experimental and statistical methods (Metha *et al.* 2008; Xue *et al.* 2008).

If we attempt to use inter-atomic distances related to only the main chain atoms in the torsional group in satisfying torsional angle, these distances could possibly result in the occurrence of the mirror configuration. In light of this observation, we develop a method that will eliminate the possibility for the occurrence of mirror images when using distances to satisfy torsional angles. Our working hypothesis is that a set of key distances can be used to constrain the atoms to a position that coincide with the known angles.

III.3. Methodology

In this section, we introduce a method for using inter-atomic distances to satisfy torsional angles. The specific goals of this work is: (1) to develop a new atomic representation for the protein; 2) to specify a minimum set of distances that is needed to satisfy a known torsional angle; and 3) to show that the specified set of distances can be used to determine good starting points for the local optimization.

III.3.1. A New Atomic Representation

The atomic representation used by Williams *et al.* (2001) was made up of an N-

C_α -C backbone and a single C_β atom representing the side chains as shown in Figure 10. In order to successfully use distances to form the angle correctly throughout the optimization, we must solve the problem of determining a torsional angle uniquely from distances. The inter-atomic distances between the four atoms cannot satisfy torsional information; therefore, we have modified the polypeptide chain in Figure 10 to include pseudo-atoms (*also referred to as dummy atoms*) as part of the representative set of atoms in the model protein structure. These dummy atoms have no physical significance and simply act as reference points for insuring that the torsional angles form correctly.

The concept of using nonlinear sequential atoms or pseudo-atoms to circumvent the difficulty associated with directly computing torsional angles has been used in other molecular modeling estimators. These approaches involve computing the torsional angles from consecutive three and four backbone C_α atoms, from consecutive O and C atoms, or from consecutive pseudo-atoms (Laiter *et al.* 1995). More recently, AMPAC semi-empirical quantum mechanical program employs pseudo-atoms in computing the torsional angles when one atom is missing out of the torsional angle group (Dewar *et al.* 2004).

In specific to our work, we will attach three pseudo-atoms to each of the main chain atoms of the polypeptide segment. The coordinates of these pseudo-atoms are determined from the main chain atoms of the polypeptide chain. In order to explain our approach, we will consider only A-B-C of the segment in Figure 9. We will refer to these three atoms – A, B, and C – as base atoms. We convert the base atoms into a new coordinate system as shown in the first plot in Figure 11. The first base atom, A, is placed in the second quadrant at an angle of γ_1 away from the x-axis, the second base atom, B, is placed at the origin of the x-y plane, the third base atom, C, is fixed on the x-axis a distance of fixed distance, d , away from atom B, and, lastly, the side chain atom, C_β , is attached to the B atom. We then create a plane for the pseudo-atoms, the l -plane. The position of the l -plane is determined by bisecting the bond angle γ_1 . Thus the l -plane is from the positive x-axis as shown in the second plot of Figure 11.

Looking along the bond vector A-B, the pseudo-atoms – D_1 , D_2 , and D_3 – are positioned in the l -plane so that an equilateral triangle is formed from the D_1 , D_2 , and D_3 (See Figure 11) The first dummy atom, D_1 , is attached to the B atom at a length of

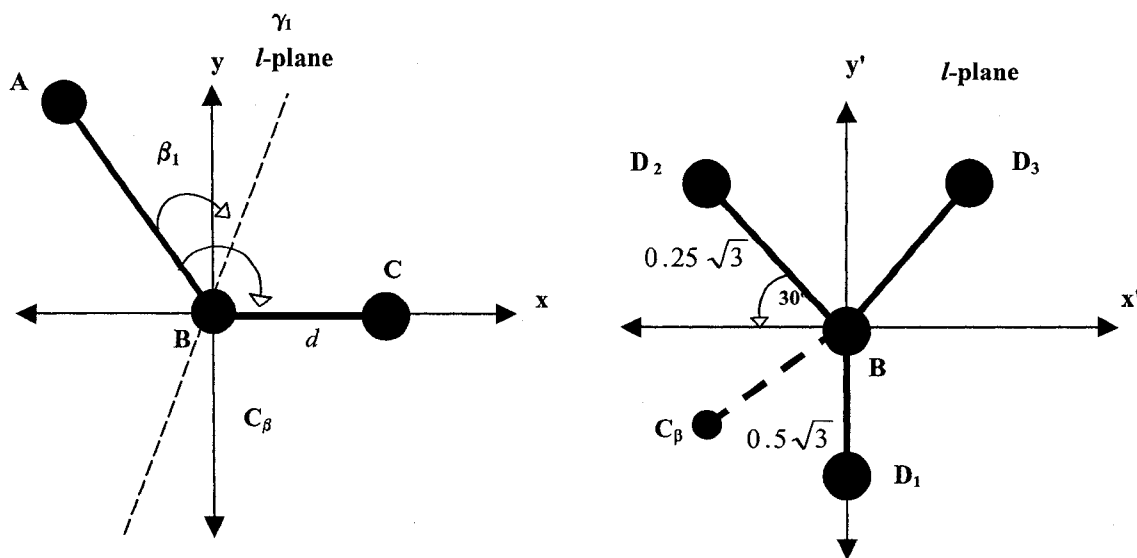


Figure 11. The placement of l -plane with respect to the bond angle γ and the placement of three dummy atoms – D_1 , D_2 , and D_3 – in the l -plane.

$0.5\sqrt{3}$ Å and fixed on the negative y' axis in the l -plane. The second dummy atom, D_2 , is attached to the B atom, at an angle of 30° degrees from the negative x' axis in the l -plane at a length of $0.25\sqrt{3}$ Å. Lastly, the third dummy atom, D_3 , is attached to the B atom at an angle of 30° from the positive x' axis at length of $0.25\sqrt{3}$ Å. Using this technique, three dummy atoms are attached the N, C_α and C atoms. The three dummy atoms that are attached to the N are referred as N_1 , N_2 , and N_3 . $C_{\alpha 1}$, $C_{\alpha 2}$, and $C_{\alpha 3}$ are attached to the C_α atom. Similarly, C_1 , C_2 , and C_3 are attached to the C atom. Once all dummy atoms are allocated for the backbone conformation, we can use information about the new atomic representation, as shown in Figure 12, in improving our build-up algorithm.

It is important to note that some care should be taken in placing the pseudo-atoms because their placement without proper consideration for maintaining exact distances constraints and position of the sidechain can cause unexpected difficulties in the optimization. The placement of the pseudo-atoms should not conflict with true distance constraints and should create ease in using distances to satisfy torsional information. The choice of adding three dummy atoms to each main chain atom was based on studying the performance of GNOMAD in using different atomic representation. In preliminary work, we chose to attach two dummy atoms to each main chain atom to care of the convergence

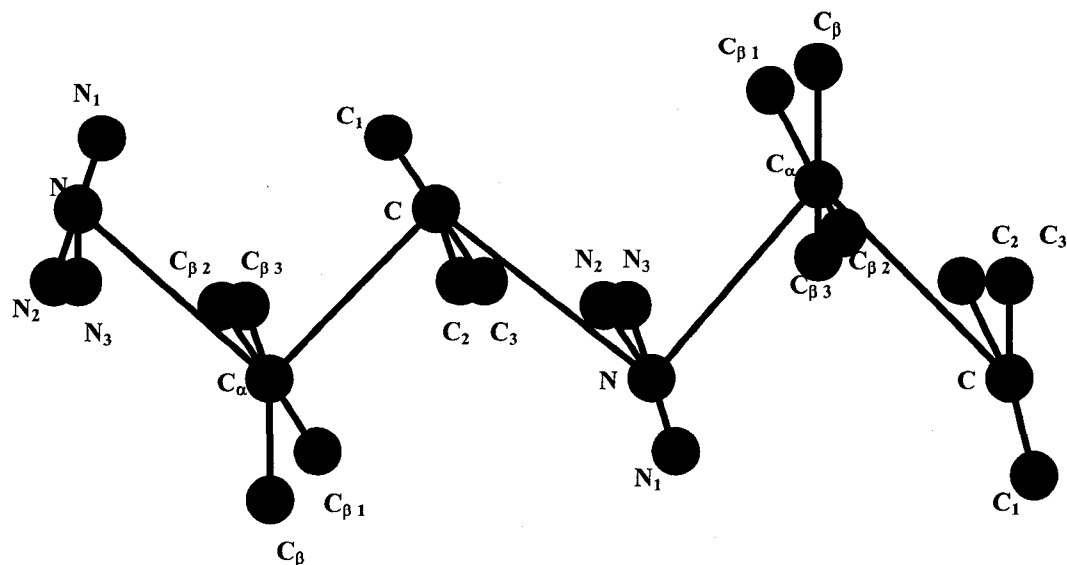


Figure 12. A segment of the new atomic representation. The chain is composed of backbone atoms and dummy atoms joined by virtual bonds.

issues associated with the position of the C_{β} atom. But the distances of the addition of two pseudo-atoms could not satisfy the torsional angle of 180° and resulted in the C_{β} atom oscillating between values that were close to 180° . In experimenting with an increase in the number of dummy atoms attached to each main chain atom, we found that an ideal set consisting of three dummy atoms will permit us to maintain our improvement in the positioning of C_{β} and, also, ensure that there are no violations of distance constraints.

III.3.2. Eliminating Ambiguities in Forming Torsional Angles

In considering our new atomic representation, we can approach the problem of using distances to satisfy a torsional angle from two perspectives: (1) using all the distances between the main chain atoms or (2) using the distances between the pseudo-atoms connected to the main chain atoms. Although these perspectives are equivalent, the second perspective allows us to avoid the mirror image problems associated with distance information because the number of distances controlled by torsional angles is increased. In this work, we will focus on using inter-atomic distances between pseudo-atoms to satisfy torsional information. This requires that we enlarge this idea of rigidity to include the pseudo-atoms that are associated with the bond angle.

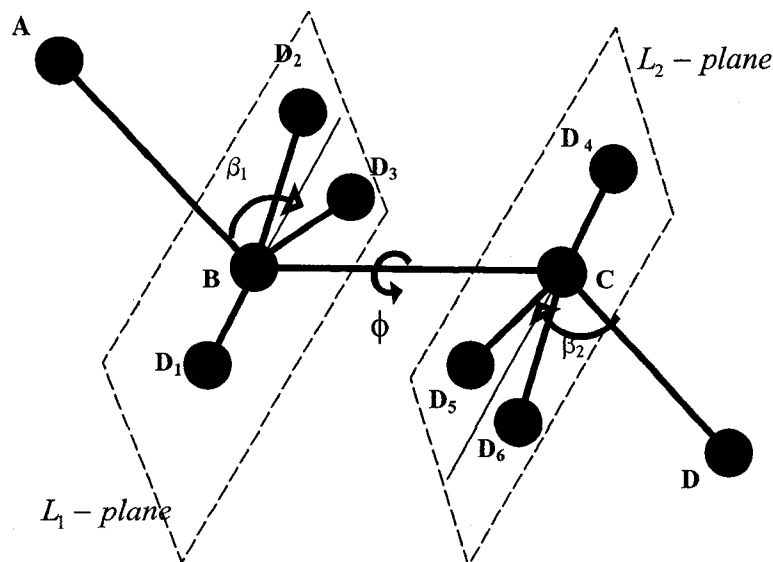


Figure 13. Using dummy atoms and bond angles to determine the torsional angle ϕ . When the torsional angle changes the dummy atom configuration rotate relative to each other.

Consider Figure 13. The pseudo-atoms that are connected to the atom B are fixed in the L_1 plane, which is rigidly connected to the bond vector AB at an angle of $\beta_1 = \frac{\gamma_1}{2}$. Similarly, the pseudo-atoms connected to the atom C are fixed in the L_2 plane, which is rigidly connected to the bond vector CD at an angle of $\beta_2 = \frac{\gamma_2}{2}$. The distance between the pseudo-atoms in L_1 and L_2 are used to constrain the pseudo-atoms to positions that satisfy the torsional angle. Because atom A and D are rigidly connected to the L_1 and L_2 plane, respectively, their atomic positions also satisfy the torsional angle. As the torsional angle ϕ changes, the L_1 and L_2 planes rotate about the bond vector BC. In Figure 14, the L_1 and L_2 planes are parallel to each other when $\phi = 180^\circ$. When $\phi = 0^\circ$, the L_1 and L_2 planes intersect. As the torsional angle, ϕ , changes, the planes will no longer be parallel to each other and each ϕ value produces a unique orientation of the L_1 and L_2 plane with respect to the bond vector BC that is consistent with the correct torsional angle.

The ability to describe a torsional angle in terms of the orientation of two consecutive planes allows us to avoid the problem of obtaining the mirror configuration

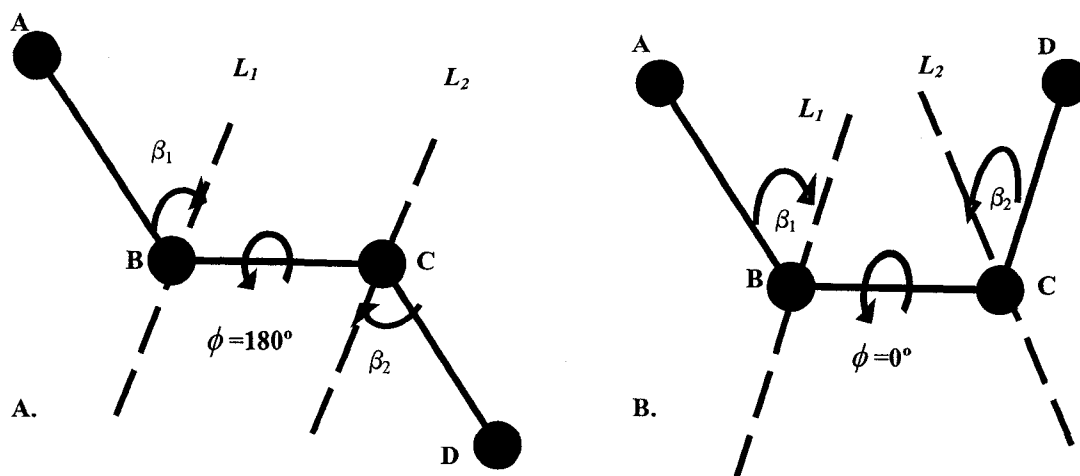


Figure 14. Definition of the torsional angle ϕ in reference to L_1 and L_2 . A) $\phi=180^\circ$ and B) $\phi=0^\circ$

because we have a set of key distances that are effective in constraining the atoms in the torsional group to a position that produced a correct torsional angle value. In light of our new atomic representation, we propose that the minimum set of structural information that can satisfy a torsional angle consist of a set of distances associated pseudo-atoms and/or associated bond angles. The bond angles are used to define a fixed position of the planes with respect to the bond vector BC. Then, the distances between the pseudo-atoms in two consecutive planes are used to constrain the pseudo-atoms to positions that are consistent with the known torsional angle.

The numerical calculation for positioning the pseudo-atoms to coincide with known torsional information is done through minimizing a distance-based L_2 -error function. Because we are using the psuedo-atoms to position the plane with respect to the torsional angle, ϕ , the objective function is defined with respect to only the remaining psuedo-atoms and is represented by,

$$f = \sum_{k=1}^n \sum_{i=1}^{m^k} \left(\frac{d_i^k - d_c^k}{\sigma_i} \right)^2 \quad (3.1)$$

where n is the number of atoms in the current residue, m^k is the number of input distances associated with atom k , d_i^k is the i^{th} input distance, d_c^k is the calculated distance corresponding to the input distance, and, σ_i is the standard deviation of the i^{th} input distance associated with the k^{th} atom.

Figure 15 is an outline for determining numerically unique torsional angles. Note that the problem of determining angles from distances, in general, is not a unique problem. However, our method allows us to determine numerically a near-perfect value of the torsional angle. Thus, for a known torsional angle, we can specify a set of key distances that is effective in producing a correct configuration that accurately reflects the known torsional angle.

III.3.3. Improving the Initial Position Algorithm in GNOMAD

The use of the additional pseudo-atoms in GNOMAD will exhibit the same expenses in computational time as that of working with large proteins because we are almost tripling the number of atoms in our proteins. In order to offset the possible gain in computational time, we use our method of satisfying torsional angles in choosing better starting points for the local minimization procedure used in GNOMAD.

Recall that at the highest level, GNOMAD builds up the protein by first finding the optimal configuration for a subset of atoms, which is also referred to as a “group”. Normally these groups are based on molecular sequence information. In previous work on GNOMAD, an atom is initially placed at a constant and preset distance and direction from the other atoms within the same optimization group (Williams *et al.* 2001). The global convergence strategy of the BFGS-cubic backtracking line search employed in GNOMAD allows us to find the optimal position for a group of atoms that satisfies the given set of distances. However, the initial positions of the atoms are only as accurate as the set of distances being used. While it may be possible to determine the optimal position of an atom when given a sufficient amount of time, we have no way of putting a cap on the time required to find this optimal position.

Our approach to finding better starting positions for atoms is to use our new atomic representation and the minimum set of interatomic distances associated with each torsional angle group to aid in determining a good starting position for atoms when torsional angle are known. In general, we construct an initial configuration for a group of atoms by using the atomic position of the previously optimized set of atoms and picking good starting positions for the remaining atoms in the group. Once the starting points for the current group of atoms have been determined, the GNOMAD algorithm proceeds on a one-atom-at-a-time fashion as described in Chapter III. This allows us to continue

Algorithm for Numerical Uniqueness of Torsional Angles

1. Convert all angles into distances.
 2. Store the set of distances between all dummy atoms and the associated bond angles for torsional angle group.
 3. Initialize the coordinates of each atom in the optimization grouping.
 4. Find the configuration that minimizes the objective function using a modified BFGS local optimization
 5. Use the resulting atomic coordinates to compute the torsional angle correctly.
-

Figure 15. The outline of the algorithm for numerical uniqueness of a torsional angle.

working in the three-dimensions and also maintain the effectiveness of our VDW constraint. The advantage to improving starting point algorithm is the reduction in the computational time spent searching for finding a near global minimum.

Given that we have a previously optimized set of atoms, the choice of the number of atoms that will be added to create a new group is determined based on the torsional angle that is associated with last main chain atom of the previously optimized group of atoms. In Figure 16, for example, the last main chain atom of the previous optimization group is a C atom, and then we add four atoms: the N, N₁, N₂, and N₃ atoms. This will allow us to use distances to ensure that the initial torsional angle ω_0 is in the neighborhood of the known value and thus, that we have good starting points for current optimization group.

In order to understand our method for picking better initial configurations, assume that we have a group of n atoms. The algorithm assumes the minimum of the objective function for $(n-4)$ -atom group has been found. The algorithm then uses the optimized $(n-4)$ -atom group together with four additional atoms to construct an initial n -atom structure, with which a global optimization procedure is started to minimize the objective function of the n -atom group. For this reason, we will focus on determining good starting position for the four remaining atoms in the n -atom group.

We determine the initial position of the first update atom, which is always a main chain atom, by placing it relative to the last three atoms from the previous optimization group. To determine the coordinates of the first update atoms, assume that we have found

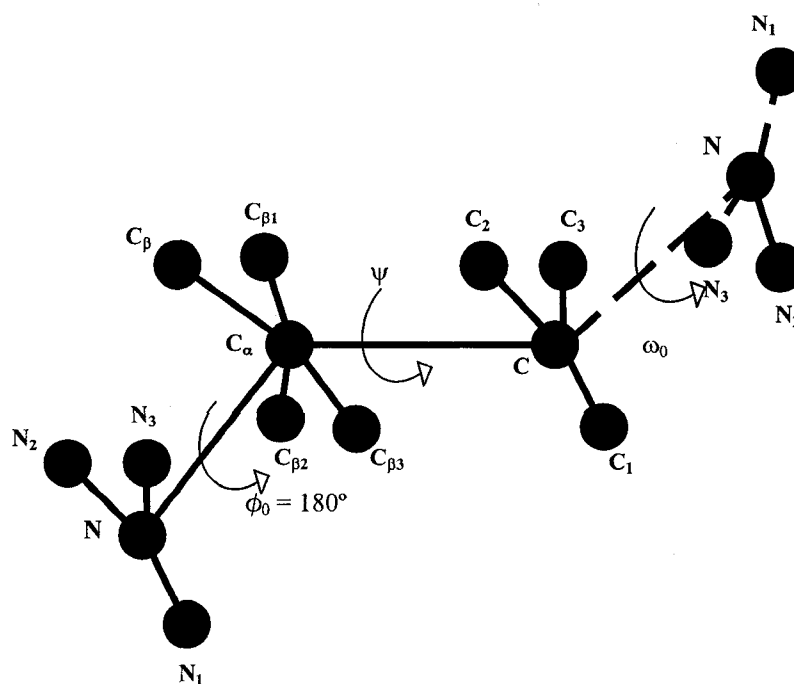


Figure 16. The starting points of a subsequent optimization grouping. These points are made up of the previously optimized of atoms (*indicated by the solid black line*) and additional set of atoms (*indicated by the dashed black line*) whose starting positions are chosen to constrain the torsional angle ω to a value that is in the neighborhood of the known angle.

the coordinates for the last three atoms of the previous optimization group. We will call these our base atoms. Let the coordinates for the three base atoms be denoted by $x_2 = (u_2, v_2, w_2)^T$, and $x_3 = (u_3, v_3, w_3)^T$. Given that we know the coordinates of these atoms, we want to determine the starting coordinates of the next atom, x_4 .

We can find the position for x_4 by using the positions of x_1 and x_2 , the bond angle between x_1 , x_3 , and x_4 , and the bond length between x_3 , and x_4 (*See Figure 17*). Once we have determined the starting point for x_4 , we use the bond angle associated with x_4 to determine the position of the l -plane. As shown in Figure 18, the l -plane is fixed at an angle $\beta_2 = \frac{\gamma_2}{2}$ from the bond vector from atoms x_3 and x_4 . Lastly, the starting position of the last remaining atoms is going to be somewhere in the l -plane. Because we are interested in finding an initial starting position of the last atoms so that

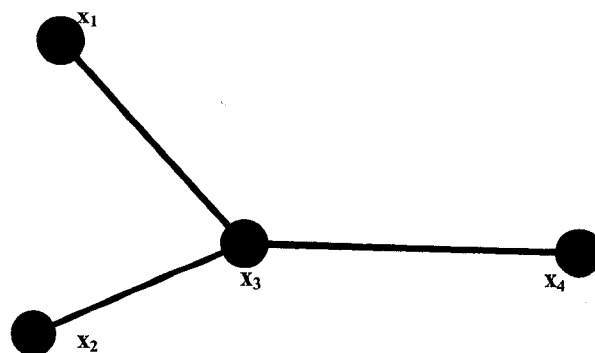


Figure 17. Determining the position of the first update atom. In this diagram x_1 and x_3 are main chain atoms and x_2 is the last dummy atom of the previous optimization grouping.

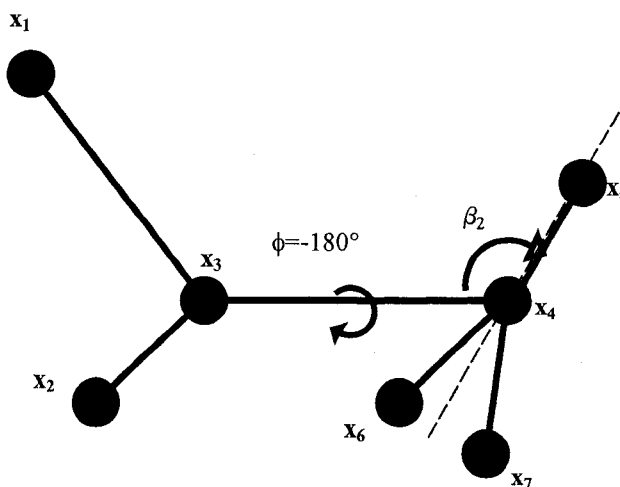


Figure 18. The initialized position of remaining update atoms in a segment of the polypeptide.

corresponding initial torsional angle is in the neighborhood of the correct value, the best starting position for each of remaining update atoms are determined by solving a local search problem in the torsional angle space.

Mathematically speaking, we determine the best starting position of the remaining atoms by finding the initial torsional angle value that produces the lowest function value for the L_2 -function error in Equation 3.1. This requires that we utilize a slightly modified version of the algorithm in Figure 15. For the preliminary setup, we define an error function value threshold cutoff at a high value and partitioning the torsional angle interval $(-\pi, \pi)$ uniformly. We then initialize $\phi = -180^\circ$ so that the l -plane is initially

positioned relative to the starting torsional angle as well as the bond angle as shown in Figure 18.

To find the starting coordinates that yield the optimal torsional angle value, we rotate the torsional angle about the main bond vector. As we rotate the angle, the remaining update-atoms in the l -plane take on different starting position. For each degree increment taken on the unit circle, the torsional angle updates and we compute the new starting positions of each update atom. The update for the new starting position is given by

$$\bar{x}'_i(\theta) = R \cdot \bar{x}_i(\theta) + \bar{x}_4 \quad (3.2)$$

where $\bar{x}'_i(\theta)$ is the optimal starting position of each of the remaining atom added to the group for $i = 5, 6,$ and 7 , R is the rotation matrix relative to the three previously determined atoms, $\bar{x}_i(\theta)$ is the initial atomic position of i^{th} update atom, and \bar{x}_4 is the starting position of the first atom added to the group. For each set of starting positions, we then calculate the set of input distances d_c^k associated with remaining update atom, evaluate the function, and store the error value if it is lower then the threshold error-cutoff. Lastly, we fix the starting coordinates of the update atoms to be equal to the coordinates corresponding to the torsional angle associated with the lowest function value. These initial starting positions are added to the previous optimization group to create an initial configuration, with which our global optimization procedure is started to minimize the function of the n -atom group.

Figure 19 is an outline of the algorithm for finding starting points of a new optimization grouping. Once the optimal position for every atom in current optimization group is determined, we add another group of atoms onto the previously optimized group to create an initial configuration for next optimization group, determine the best starting position of the added atoms, and than perform an optimization to determine the optimal configuration. We continue this procedure until the whole protein structure is built up. One of the major arguments against this approach is that the use of distances to form the angle correctly is unnecessary for determining the good position for atoms. Recently, Wu (2007) introduce an updated and a rigid geometric build up algorithm that yield good estimates of protein structure using a sparse set of inter-atomic distances. The major

An algorithm for finding the initial atomic positions for BFGS local optimization

1. Transform two previous main chain atoms and the first dummy atom into new coordinate system.
 2. Fix the first update (main chain atom) atom in the new coordinate system relative to the previous atoms.
 3. Define the plane containing the psuedo-atoms with respect to the bond angle.
 4. Set an error function value threshold cutoff and partitioning the torsional angle interval $(-\pi, \pi)$ uniformly. Initialize the torsional angle to $-\pi$ degrees.
 5. Fix nsteps = 360. Repeat:
 - For each torsional angle test value;
 - Determine the coordinates of the remaining update atoms.
 - Evaluate the function error;
 - If the function error is less then the minimum threshold,
 - store the coordinates of the three atoms.
 - End
 6. Re-initialize the coordinates of the starting point to the coordinates corresponding to the torsional angle value that generates the lowest function error value.
-

Figure 19. The outline of the algorithm for finding a set of starting points for remaining atoms in the current optimization grouping.

focus of their work is in improving the mechanics of their atom-based approach to effectively produce good quality protein structure when using sparse sets of inter-atomic distances.

Our work is different in that we makeup for the missing distances by making use of available torsional information. Because GNOMAD is known to work well withdistances information, we convert the torsional angles into a set of distances that is able to mirror the variational behavior of known torsional angles. This allows us to use torsional information to determine better starting positions for our atom-based approach and it sets the foundation for being able to use distances to accurately form local structure within the proteins.

III.4. Validation

Optimization experiments were performed to evaluate the method of satisfying torsional information using our new atomic representation. The data used in the

experiments were derived from molecules with known three-dimensional structure that were taken from the Protein Data Bank (Berman *et al.* 2000). In this work, the goal of the experiment was to evaluate the effectiveness in using the modified GNOMAD algorithm and our new atomic representation in satisfying torsional information in secondary structure regions. Therefore, only the GNOMAD code is used in the validation experiments, in order to evaluate effectiveness of the new atom approach compared to using a main chain atom approach.

To evaluate our algorithm, we run our program on secondary structure segments taken from the L7/L12 50S ribosomal protein from *Escherichia coli* (*E. coli*). This protein contains 68 amino acids, three helices and one beta sheet comprised of three strands (Williams *et al.* 2001). The data for the test structure is generated using the 1CTF structure downloaded from the PDB data bank (Berman *et al.* 2000). For the purpose of this work, an optimization experiment consists of an attempt to recreate major secondary structures, α -helices and β -strands, using a minimum set of distances. Since these secondary structures tend to be uniform substructure in proteins, we choose to run experiments on only one α -helix and one β -strand taken from the 1CTF protein. The results will be relatively consistent no matter which helix or strand we pick from the 1CTF protein or from another protein.

Distances are chosen to include only those associated with the atoms that are found in a torsional angle group and that define each secondary structure under consideration. The first secondary structure, the helix, is defined by its series of distances between connected adjacent amino acids: $(i, i + 3), (i + 1, i + 4), (i + 2, i + 5), \dots$ The second structure, the strand, is part of a much larger substructure and thus is not defined by distances between atoms in adjacent amino acids (Dill 1990). For this reason, the minimum set of distances will be made of only distances between all atoms in connected adjacent amino acids. One advantage to this minimum set is that it will eliminate the dependence of satisfying angles on the availability of distances.

The first set of distances include all the fixed “ i to $i+1$ ”, “ i to $i+2$ ”, and “ i to $i+3$ ” interatomic distances taken from the main chain atomic representation⁵, and the second

⁵ The “main chain representation” is an atomic representation including the main chain atoms N, C $_{\alpha}$, C, and C $_{\beta}$ where the C $_{\beta}$ is representative of the sidechain of an amino acid.

set of distances include the same interatomic distances taken from the new atomic representation. We compute three configurations for each secondary structure: first with the main chain distances, then with the addition of a torsional angle constraint, and finally with only the new model distances.

The torsional angle constraint method is designed using the general enforcement procedure discussed in Chapter III. The purpose of this constraint is to keep the optimization from converging to the opposite torsional angle value. Given that we know the correct angle, we can determine the position of the atom being moved that would result in the mirror image configuration and thus the opposite torsional angle. Based on this determination, we construct a torsional angle violation region, for which the atom is restricted from moving into. The region is spherical and can be expanded only to a radius that does not violate any of the true distance constraints. This method of constraining torsional angles becomes less effective as the angle approaches 180°, where the radius of the torsional angle sphere approaches zero.

The results of the optimization experiments were collected and analyzed in terms of accuracy of GNOMAD in recreating the secondary structure and effectiveness of using distances to satisfy torsional angles. Accuracy is measured in terms of the RMSD and the distance residuals (Williams *et al.* 2001). In the absence of a “known” structure – the only error measure available is the distance residuals, which represent how well the optimization satisfies the input distance data. When a “known” structure is available, as is the case in this work, an RMSD can be computed to measure the accuracy of the estimated structure to the known structure.

Effectiveness of our method in using distances to satisfy torsional information is assessed by two measures. The first is the percent of angles that are correct, which is given by:

$$\% \text{ correct} = \frac{\text{number of correct angles}}{\text{total number of angles}} \quad (3.3)$$

This is a measure of the overall effectiveness of each method in satisfying torsional information. Even if there are a large percentage of correct torsional angles, it is difficult to gauge what the actual problem is in using distances to satisfy individual torsional angles.

As a test of satisfaction of individual angles in using distances from the associated torsional angle group, we consider the relative error of each torsional angle. The relative error measures how close an individual angle is to the correct value. Each torsional angle has an associated relative error and is computed by

$$\text{relative error} = \frac{\text{original angle} - \text{calculated angle}}{\text{original angle}} \quad (3.4)$$

The relative error also indicates what type of configuration results from using the minimum set of distances associated with torsional angle groups to satisfy individual torsional angles (*e.g. correct, mirror image, etc.*). This can be seen by defining the original angle by, θ_o , and the calculated angle by, θ_c . Suppose that the distances constrain the atoms to positions that coincide with the additive inverse of θ_o , that is $\theta_c = -\theta_o$. Then, the relative error is $r = 2$. On the other hand, suppose that the distances constrain the atoms to positions that coincide with the correct original angle, θ_o , that is $\theta_c = \theta_o$. Then, the relative error is $r = 0$.

Because the calculated angles are only precise to within a cutoff threshold, we define ranges for relative error that are associated with correct and incorrect configurations. A relative error in the range of 0 to 1 indicates that the distances constrain the torsional angle to the correct configuration associated with a torsional angle. A relative error in the range of 1 to 2 means that the distances constrain the torsional angle to the mirror image of the correct configuration, which yields the additive inverse of correct torsional angle. Lastly, a relative error > 2 indicate that distances do not satisfy the torsional angle, but result in another type of distance constraint violation.

III.4.1. Results

To demonstrate the accuracy and effectiveness of the various methods, the error, the RMSD, and the percentage of correct angles were recorded for each run. Results are presented in Table 4. The first column contains the secondary structure type. The second column contains the atomic models and methods implemented in GNOMAD for satisfying torsional angles and the remaining column list the measures of accuracy for each method for both secondary structures under consideration: maximal distance residual error, the root mean square deviation (RMSD), and the percentages of correct

Table 4. Accuracy results for using distances to satisfy torsional information.

Accuracy and Effectiveness of GNOMAD				
Secondary Structure	Methods	ERROR	RMSD	% correct angles
α -Helix	Main chain atomic representation	0.0195	4.3983	51%
	Main chain w/ Torsional angle constraint	0.0237	2.1487	87%
	New atomic representation	0.0033	0.0812	100%
β -Strand	Main chain atomic representation	0.0089	2.2955	53%
	Main chain w/ Torsional angle constraint	0.0110	3.7475	67%
	New atomic representation	0.0046	0.0171	100%

angles found secondary structure region, respectively.

To understand the effectiveness of using distances to satisfy individual types of torsional angles in the helix and strand region, we also included results for the relative error. Figures 20 and 21 include results for an arbitrary helix and strand found in 1CTF protein in using each method. Each figure displays three plots: one for the main chain atom representation, one for the main chain atom representation with torsional angle constraint, and one for the new atom model representation. Each plot contains three point types, one for representing the “ ϕ ” angle type, one for representing the “ ψ ” angle type, and one representing the “ ω ” angle type. The points are plotted over the entire range of torsional angles for each secondary structure. Each point represents the residue number and the corresponding relative error of each angle in the secondary structure region.

To illustrate the effect of satisfying torsional information with both the main chain and new model, Figures 22 and 23 show qualitative comparisons of both secondary

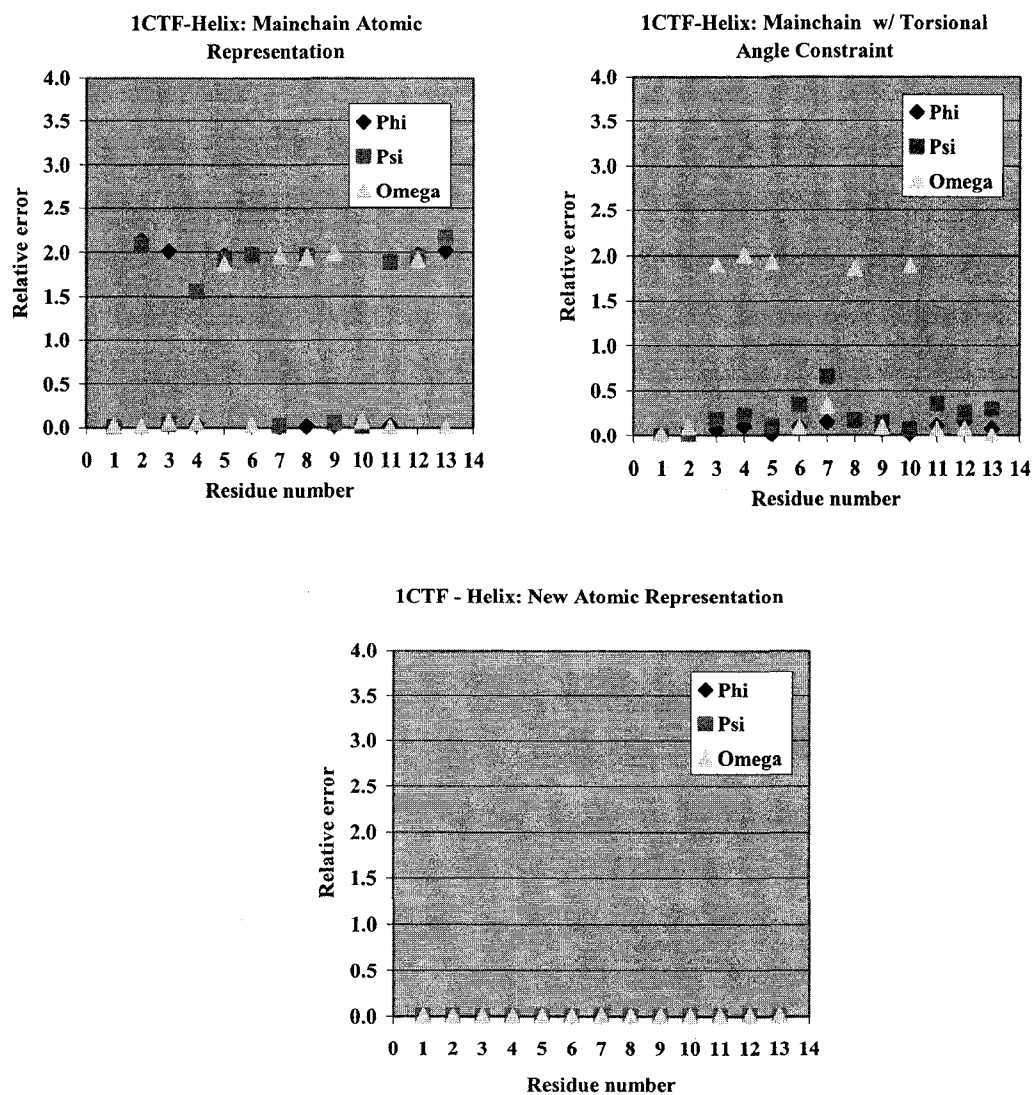


Figure 20. Results of relative error for using distances to satisfy torsional angles found in the helix region for the main chain, main chain with torsional angle constraint, and new atomic representation.

structures resulting from each of the method using GNOMAD. Figures 22 shows a comparison of the helix structure resulting from each of the method implemented in GNOMAD with the corresponding helix structure from the 1CTF protein. Similarly, Figures 23 shows comparisons of the strand structure resulting from each of the method with the corresponding strand structure from the 1CTF protein.

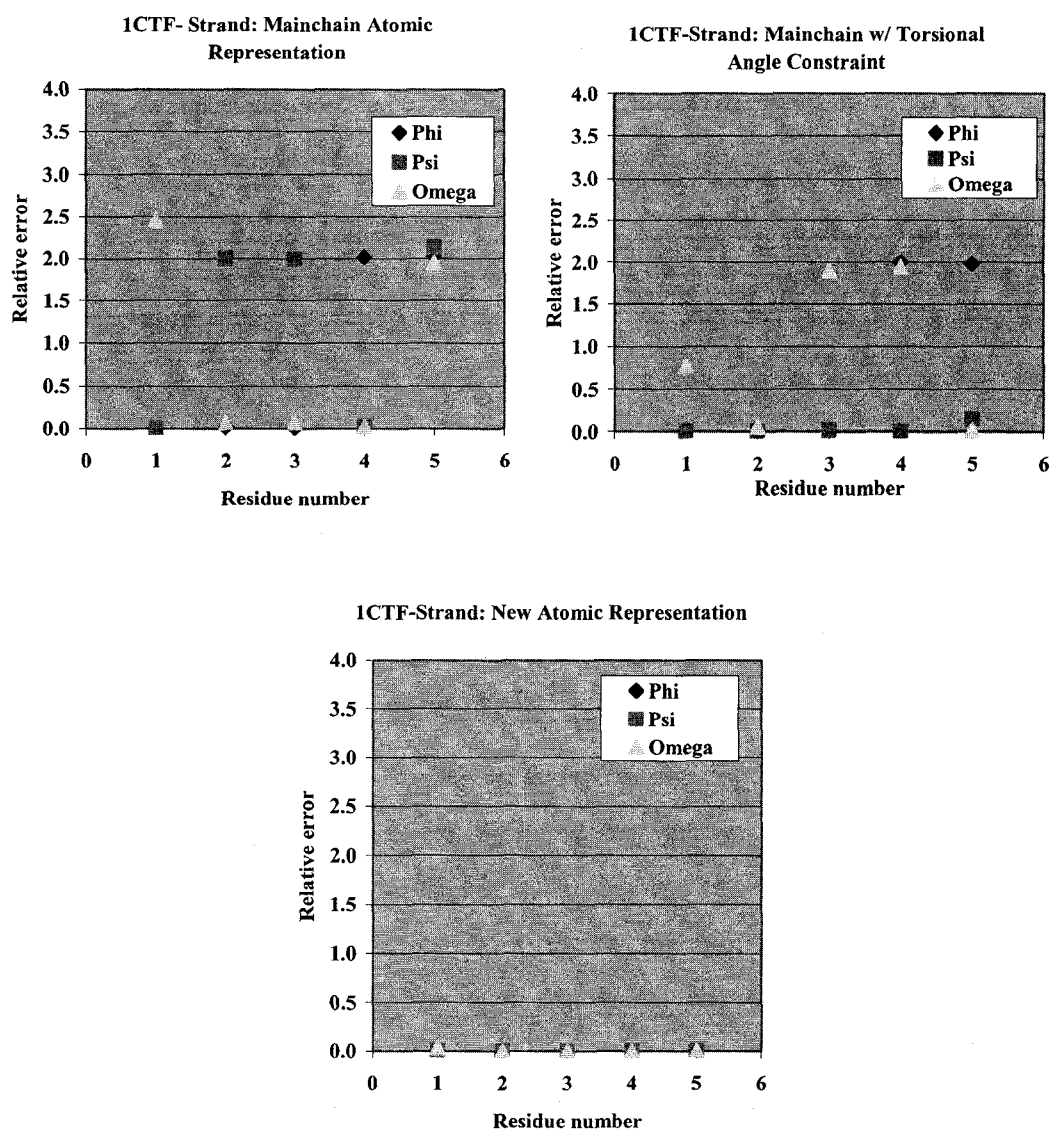


Figure 21. Results of relative error for using distances to satisfy torsional angles found in the strand region for the main chain, main chain with torsional angle constraint, and new atomic representation.

III.4.2. Discussion

Results from using distances to satisfy torsional information – main chain model, main chain with torsional angle constraint, new representation model –revealed several important issues. First, using the main chain atomic representation and our minimum set of distances is not effective in satisfying torsional information in the framework of GNOMAD. In Table 4, we see that the possibility of constraining angles to their correct value in a helix or a strand region through using distances associated with torsional angle

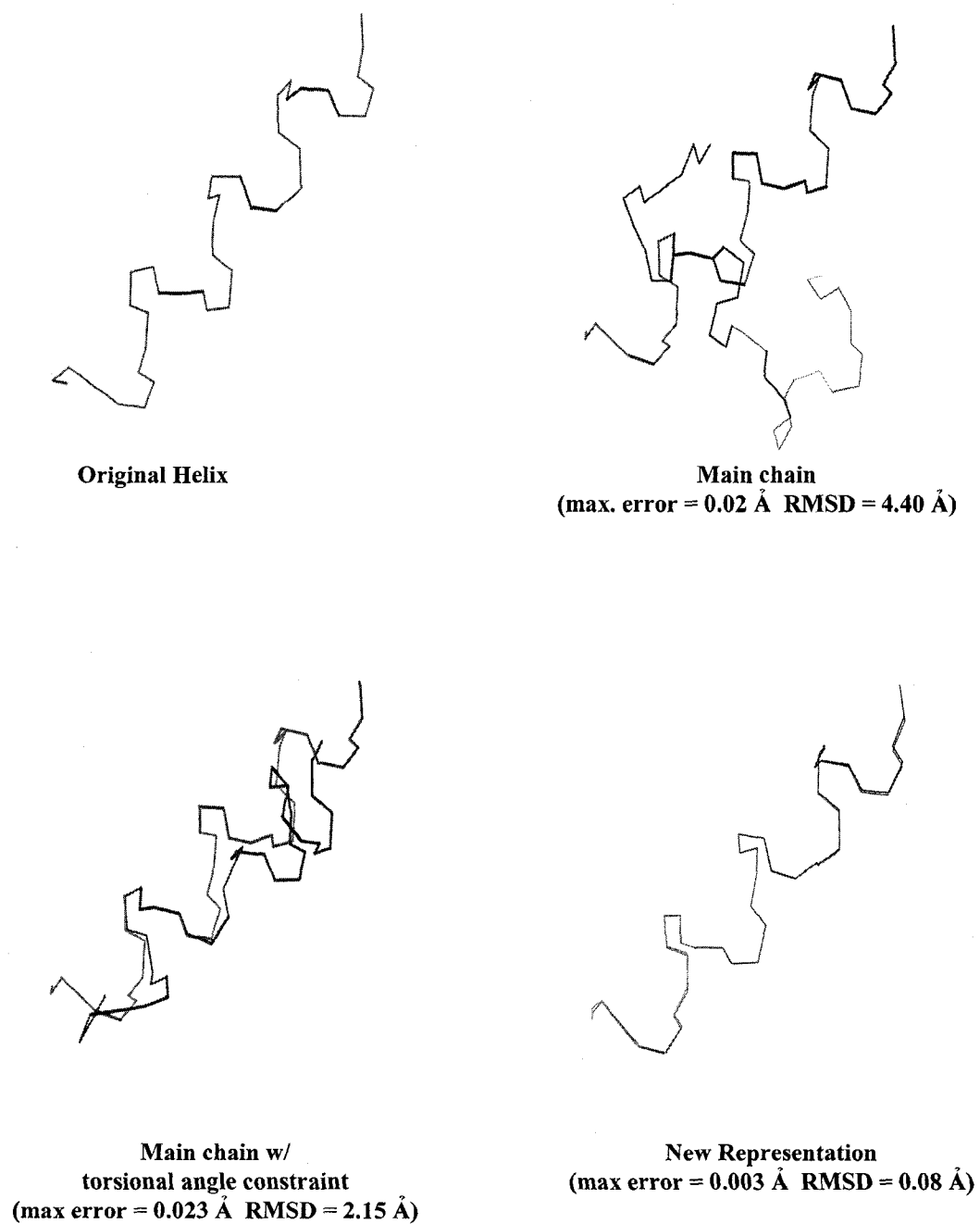
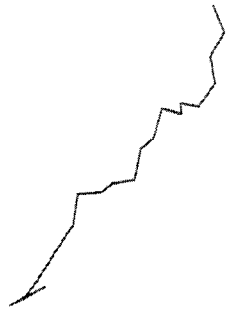
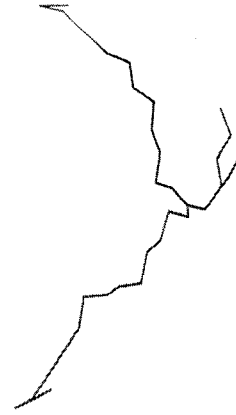


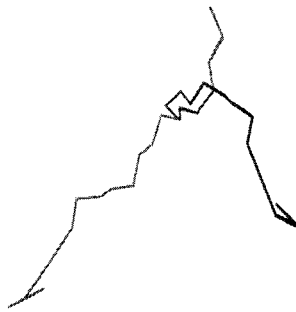
Figure 22. 1CTF α -helix and torsional method results. Comparison of a α -helix taken from the crystal structure of 1CTF with the computed results based on using distances to satisfy torsional angle.



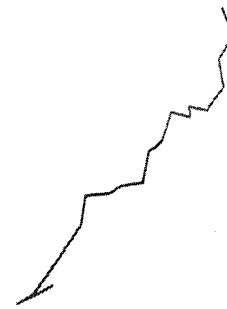
Original Strand



Main chain
(max. error = 0.01 Å RMSD = 2.30 Å)



Main chain
w/torsional angle constraint
(max. error = 0.11 Å RMSD = 3.75 Å)



New Representation
(max. error = 0.01 Å RMSD = 0.02 Å)

Figure 23. 1CTF β -strand and torsional method results. Comparison of a β -strand taken from the crystal structure of 1CTF with the computed results based on using distances to satisfy torsional angle.

group is about 50%. This type of behavior is an inherent problem with distance information and is to be expected because satisfying distance information with an optimization procedure can typically produce two spatial configurations, one that produces the correct torsional angle value and, the other that produces the additive inverse of the torsional angle value (Kinjo and Nishikawa 2006).

In the context of GNOMAD, the occurrence of mirror image problem is related to satisfying individual torsional angles using the distances from the corresponding torsional angle group. The first plot in Figure 20 and Figure 21 supports this observation of mirror image occurrence. A relative error of 2 indicates that a crucial factor in the success of constraining torsional angles with the distances in corresponding torsional angle group is to focus on eliminating the occurrence of mirror image configurations associated with each torsional angle. About half of the angles found in the helix and strand region have a relative value of 2, which indicates that the distances are constraining atoms to positions that coincide with the opposite of the correct torsional angle value.

The torsional angle constraint made it possible to avoid the mirror image problem for 87% of the angles found in the helix region and for about 67% found in the strand region as shown in Table 4. In specific to the helix region, the torsional angle constraint eliminated the occurrence of mirror image configurations associated with all the ϕ and ψ torsional angles as indicated by the second plot in Figure 20. However, this method was unsuccessful in alleviating the problem for some of the ω angles found in same region.

The difficulty in satisfying the ω angles in the helix region is due to the fact that the atomic position associated with the correct ω angle value and the atomic position of its mirror image are so close to each other in the Cartesian plane that the optimization has difficulty using distances and even the torsional constraint to determine which position yields the correct torsional value and thus, the optimization oscillates between the correct and mirror values until the stopping criteria is reached. Because the criterion for the most optimal configuration is the one with the lowest maximal distances residual, it is possible to pick the configuration that contain incorrect omega angles.

For this arbitrary helix region, the 38% increase in the number of correct angles

translates into improved RMSD. Table 4 reveals a drastic improvement in the accuracy of the structure as indicated by its RMSD value of 2.15 Å. However, an increase in the number of correct angles does not always imply improved accuracy in all major secondary structures. In the case of satisfying torsional information in the strand region, the addition of the torsional angle constraint was effective in increasing the number of correct torsional angles; however, the RMSD value worsens with this increase.

In considering the second plot in Figure 21, we see that the problem of mirror images was not isolated to one specific angle. The ψ torsional angles, in addition to the ω angles, are not being satisfied, which is demonstrated by the relative error of the ψ and ω angles found in the tail of the strand. The torsional angle constraint is not as effective in correcting torsional angles in the strand region as it was in the helix region. This observation is not surprising for several reasons. First, the strands are elongated substructures and thus, these substructures are not as tightly packed as helices. Hence, distances or even torsional constraints are not enough to fully determine this secondary structure.

Another reason for the expected relative error results in the strand region is due to the fact that strands are part of a larger substructure in the protein, the β -sheet and thus, the correct formation of the a strand have some dependence on the formation of other strands in the β -sheet (Aszodi and Taylor 1994). Although the formation of one strand may have some dependency on another strand within the protein structure, we want to design a method that yields the best possible strand when only distances are used to satisfy torsional information. Hence, our goal is to design a method that will be able to accurately construct all backbone torsional angles when the information is provided, regardless of the major secondary structure type.

In order to alleviate this mirror image problem completely, we revisited the idea of using distances only to constrain torsional angles with an improved atomic representation. The underlying idea is to increase number distances that are directly affected by the rotation of the torsional angle. In the main chain model, there was only one distance associated with the torsional angle group that is affected by rotation of the angle and this information is not enough to specify a unique angle. We need more distances that are affected by the rotation of torsional angles.

The mirror image problem can be fixed by simply including more distances from other main chain atoms in the protein that are not in the torsional angle group. While this would allow for enough information to correctly form the secondary structure accurately, it would not allow us to use only distances associated with the torsional angle group to satisfy individual torsional angles. Moreover, using arbitrary distances found in the secondary structure region introduces the dependence of satisfying torsional information on the availability of distances and in some cases, the distances may not be sufficient to satisfy torsional angle.

We develop our new atomic representation, so that we would have a minimum set of distances that could always be used to satisfy torsional information when the data is available. Because our new model includes at least six more atoms in the torsional angle group, the number of distances in the torsional angle group that are directly controlled by the rotation of torsional angles increases from one to at least nine distances. This gives us a better chance of eliminating the mirror image problem associated with each torsional angle in the main chain model. By employing this new model and the minimum set of distances associated with each torsional angle group, 100% of the torsional angles are correct in both secondary structure regions. The third plots of Figure 20 and Figure 21 indicate this, where the relative error for all torsional angles is close to zero.

In the context of GNOMAD, good RMSD values do not always correspond to those configurations with low “distance-residual error”. Notice in Table 4 that the “maximum distance-residual error” from both secondary structures is less than 0.005 Å. In light of the two atomic representations, the value could be indicative of two types of configurations. The first being one for which all distances are satisfied, but for which there are many torsional angle constraint violations and the second being one in which all distances and all torsional angle are satisfied. In most cases, an “error” result is around 0.04 Å. This value tells us that some of the data has not been satisfied; however, errors in this range could yield secondary structures with relatively good RMSD values. For the scope of this work, we work with the accuracy metrics: percent of correct angles, relative angle error, maximum distances residual, and RMSD.

The qualitative results presented in Figures 22 and Figure 23 show what typically happens in GNOMAD when the various methods of satisfying torsional information with

distances are applied. For each of the test problems, it is apparent that better secondary structures are found as we improve the method for eliminating occurrence of mirror images. Figure 22 and Figure 23 demonstrate in some worst-case scenarios how the helix structure determined by using the main chain representation can be affected by distances constraining the atoms to the mirror image of the torsional angle configurations. The figures show a helix of the protein 1CTF that was determined, first by using distances taken from the main chain atomic representation and then with the addition of the torsional angle constraint. The first picture in both plots shows that using distances associated with the torsional angle groups taken from mainchain representation results in a secondary structure whose torsional angles disagree with many of the angles in the original secondary structure.

Alternatively, the second picture in Figure 22 shows an overall improvement in the formation of the helix when using the torsional angle constraint and distances taken from the main chain representation and, thus, results in a structure that is consistent with the helix of the original structure. Also, the second picture in Figure 23 shows that using the torsional angle constraint and distances taken from the mainchain representation worsens the formation of the β -strand. The use of a new atomic representation provides significant improvement over the deformed structures that result from using only the main chain atomic representation. We clearly see that employing a new atomic representation and using a minimum set of distances is successful in recreating the α -helix and the β -strand that almost completely agrees with the original structure.

III.5. Conclusion

GNOMAD is effective in satisfying VDW and chirality constraints, has good global convergence properties, and is computationally efficient (Williams *et al.* 2001). But, this algorithm is not designed to effectively use torsional information. The algorithm's inability to effectively use this readily available information limits the accuracy in the formation of reoccurring substructures, such as α -helices and β -sheets, which are found in a large number of proteins. In this section, we proposed a practical method for using distances to satisfy torsional information in secondary structure regions and show that effective use of torsional information in GNOMAD results in accurate formation of these major secondary structures.

In order to do this, we specifically study the secondary structure regions and develop methods for improving those regions when torsional information is available. The idea for the method is based on specifying a minimum set of distances that can be used to give a good estimate of each of torsional angle. The choice of these distances was made so that interatomic distances that were associated with the torsional angle group would be included. A crucial factor in using distances to satisfy torsional information was to include distances associated with specific atoms in the torsional angle group that are directly affected by the rotation of a torsional angle. If we could specify enough distances that are controlled by the change in a torsional angle, we could possibly solve the inverse problem, that is, determining a close estimate of the torsional angle using those distances.

In general, the quantitative and qualitative results indicate that new atomic representation offers an effective approach for satisfying torsional information and increased accuracy in estimating major secondary structures. In all cases, the increased accuracy is due to eliminating the occurrence of mirror image problem inherent in using distance input in nonlinear optimization. Results show that the addition of the torsional angle constraint provides for some improvement over the main chain model alone in the helix region, and the use of the new model yields significantly better RMSD results in both secondary structures.

Several conclusions can be drawn on the results of this work. First, torsional information can provide valuable and “free” information for the estimation for α -helices and β -sheets, but the incorporation of torsional information into nonlinear optimization algorithms is often ineffective. Second, a new atomic representation is found for the protein structure that expands the number atoms in a torsional angle group and thus, the number of distances controlled by an individual torsional angle. This representation, combined with an a minimum set of distances, allows for satisfaction of all torsional information, maintains good global convergence, and results in a significant improvement in the quality of local substructures within the protein. Third, tests with the new atomic representation performed on arbitrary secondary structure from the 1CTF protein yield significant improvements in terms of constraint satisfactions and RMSD.

Since the new atomic representation and the minimum set of distances are sufficient for satisfying torsional angle, it is possible to determine the major secondary structure in protein structure when all torsional information is given in those regions. Our results are novel because they illustrate a practical method for using distances to satisfy torsional information, and provide a mechanism for correcting local substructures within the framework of an atom-based optimization procedure. Given that we have a solid method for constructing some of the major components found in the proteins, we can now examine methods for making GMOMAD capable of handling other types of information, such as surface, solvent accessibility, etc. to help in bringing these major components together and aid in folding of the three-dimensional structure.

CHAPTER IV

A COMPUTATIONALLY EFFICIENT METHOD FOR USING MOLECULAR SURFACE CONSTRAINTS

IV.1. Introduction

Obtaining a better quality three-dimensional protein structure requires integration of a variety of structural data. Distance and angle information are the primary types of data used in optimization procedures for computing protein structures. Due to the significant advancements made in collecting protein structure information, there are other types of structural data that can be used in structure computation such as, surface, shape, volume, etc. Surface and shape information are available from a variety of experimental and computational techniques including solvent accessibility, electron microscopy, sedimentation experiments, and homology modeling (Dugan and Altman 2003). It is difficult, however, to use this data to develop a practical constraint for use during structure computation.

There are several reasons for the integration of surface information into our molecular structure estimation. Primarily, the hydrophobic effect on the folding of proteins can be modeled in the context of molecular structure estimators (Wade 1996; Cao *et al.* 2002). Secondly, Dugan and Altman showed that there is a clear correlation between how well a model conforms to its shape and how close the model is to being the correct model (Dugan and Altman 2003). Lastly, analysis of interactions in specific regions of the molecular surface and the surrounding environment is useful in the context of drug design (Schmidt *et al.* 1998; Pedretti *et al.* 2002).

The standard treatment of this type of data in optimization algorithms involves developing a potential energy function from the solvent accessible surface area and including this potential energy function as a term in the objective function (Lee and Richards 1971; Felts *et al.* 2002; Gallicchio and Levy 2003; Kar *et al.* 2006). These standard methods for integration of surface shape data are effective in numerous optimization algorithms. However, the penalty function methods are not as effective as

the GNOMAD approach, which is to separate constraints from the formulation of the objective function.

In this chapter, we present an algorithm for extracting the molecular surface from an atomic representation. We then introduce a new method for directly using a molecular surface to constrain the position of atoms within the protein using the GNOMAD structural estimation framework. The chapter is organized as follows: in Section IV.2, we provide background theory for molecular surface construction. In Section IV.3, we describe the method for constructing molecular surfaces and discuss how the molecular surface is used as a constraint in protein estimation. Section IV.4 presents some experimental results of implementing surface constraint in GNOMAD algorithm. We conclude the chapter in Section IV.5.

IV.2. Construction of Molecular Surfaces

A molecular surface defines the boundary between the inside and outside of a particular molecule. Primarily, the molecular surface of proteins is used to study the hydrophobicity of atoms within the proteins. In addition, these surfaces yield valuable information that is important to many areas of drug design including, identifying clefts and possible drug binding sites in protein surfaces, studying protein-protein interfaces, and screening databases of small molecules for the purpose of identifying molecules of possible pharmaceutical use (Connolly 1983). In the context of protein structure estimation, the results of the molecular surface computation are used to model the interactions between the protein and its environment.

The key components for constructing a molecular surface are (1) identification of a set of surface atoms and (2) construction of a smooth connected surface. In many of the molecular surface software, the identification of surface atoms is usually done by a computer program that simulates the rolling of a water size probe over the protein structure (Connolly 1983). For high precision surface calculation, a ray casting procedure is used to identify the surface atoms. In this procedure, the surface of the atomic representation is obtained by simulation of the light passing through the protein structure. From this atomic representation, nodes that satisfy a specified accessibility condition are chosen as surface nodes (Torshin 1999).

Once the set of surface atoms is determined, a smooth surface is constructed using a triangularization procedure. Several algorithms have been developed to yield smooth and connected molecular surfaces, including the Connolly method by LeGrand (1993), and Vorobjev (1997), the divide and conquer method by Eisenhaber *et al.* (1995) and Sanner (1996), the spherical harmonic methods by Duncan and Olson (1993) and Wade and Gabdoulina (1996), etc. Many of the standard methods for constructing a molecular surface require a large amount of computations and the number of computations increases with the number of atoms in the molecule

In our work, a molecular surface is constructed many times during the top level of our atom-based optimization. Therefore, enhancing GNOMAD with one of the standard molecular surface algorithms will increase the computational time spent in finding a structure that satisfies surface data. In order to limit the computational time spent in constructing the molecular surface during the optimization, we use a molecular dot surface to constrain atoms.

IV.3. Methodology

Developing a physical constraint that uses a molecular dot surface to constrain the position of atoms during structure computation is complicated by the fact that we are using a surface that is just a discrete set of surface nodes. There are four steps in implementing our surface constraint:

1. Construct a dot molecular surface from the model configuration;
2. Perform the molecular structure optimization and require that all main chain and sidechain atoms stay within a specific distance from the surface nodes;
3. Perform a translational shift so that atoms do not drift too far outside of the surface; and
4. Rotate the model by finding optimal alignment between the original surface and surface constructed around the model configuration.

Although the surface constraint algorithm is made up of these four steps, the algorithm can be explained in two parts: (1) constructing the dot molecular surface and (2) satisfying surface data. Each part requires solving a separate computational problem. For this reason, we discuss the implementation of each part in the next subsections.

IV.3.1. Creating a Molecular Dot Surface from the Atomic Configuration

To represent surface data we construct a molecular dot surface from the atomic configuration. In constructing the molecular dot surface, we use some of the concepts of the marching cubes algorithm. The basic principle for constructing the surface from the marching cubes algorithm is to assign a uniform cubic grid over three-dimensional space. The algorithm then instructs us to ‘march’ through each of the cubes, testing the corner points and replacing the cubes with an appropriate set of polygons. The sum total of all polygons generated will be a surface that approximates the data set described (Lorenson and Cline 1987).

Similarly, our algorithm subdivides the three-dimensional space into a series of grid nodes. Then, the algorithm determines which of the grid points are within a specified distance of an atom in the molecule. All grid points not satisfying the distance constraint are marked as possible surface points. Next, the interior marked grid points are removed. It is easy to remove the interior nodes because we only have to check if there are any other marked nodes outside of a particular marked node. Then, the exterior marked grid points are deleted from the space using a marching cubes approach. If any node within the cube around a node i is not a marked node then that node i is marked as a surface node. This ensures that only the innermost of the marked exterior nodes define the overall dot surface. The remaining marked grid points will be the dot molecular surface of the protein, *i.e.* the boundary of the protein.

Consider the two-dimensional equivalent. Figure 24 illustrates our idea of constructing molecular dot surfaces in R^2 . The first diagram shows a grid of uniform squares equivalent to the cubes from the three dimensional algorithm. A circular object has been inserted into the grid, which is the figure whose shape we are going to approximate using a subset of grid points. In the first step, the grid points near the donut are removed as shown in the second diagram in Figure 24. In the second step, the interior grid nodes are removed as shown in the third diagram in Figure 24. In the next phase, the outermost exterior points are determined and removed. The remaining grid points approximate the shape of the object as shown in the last two diagrams of Figure 24.

The result of the R^2 case easily translates to the R^3 case. The algorithm for constructing a molecular dot surface from the atomic representation is given in Figure 25.

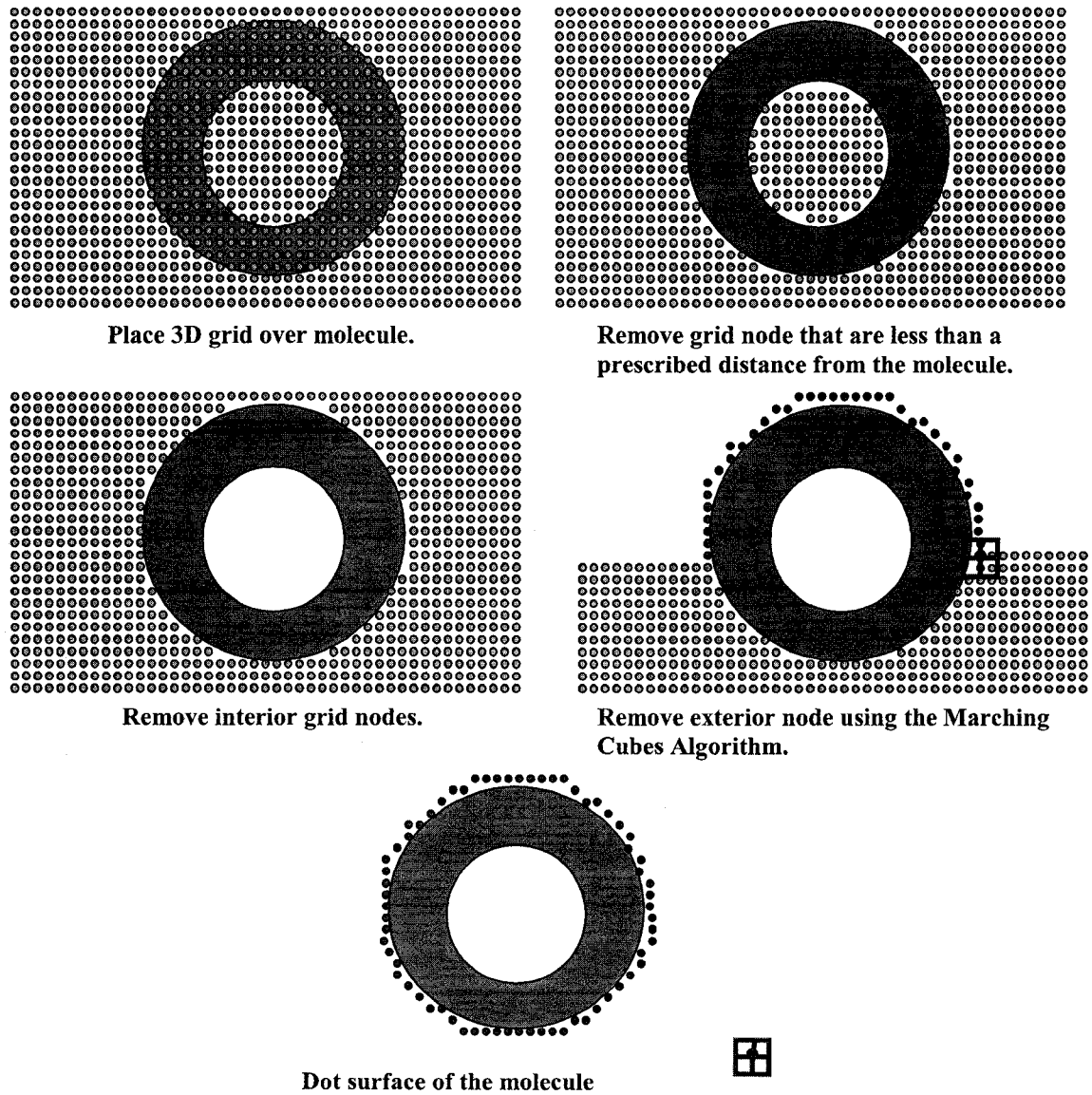


Figure 24. Illustration of constructing a surface in R^2 .

Our molecular surface algorithm has been tested on many proteins and produces a close approximation of the original surface shapes. Furthermore, our method is quite simple to understand and easy to implement.

There are two empirical parameters used in calculating the surface that significantly influence the shape of the molecular surface: (1) grid spacing and (2) distance threshold cutoff. In running preliminary tests, we found the optimal parameters

Algorithm for Constructing Molecular Dot Surface

1. Construct a uniform grid of nodes.
 2. Insert the atomic coordinates.
 3. Remove all grid points that are less than a specified distance from all atoms in the molecule as possible surface nodes.
 4. From the remaining grid points, remove the interior grid points.
 5. Lastly, remove exterior grid points using a marching cubes approach.
-

Figure 25. Outline for molecular dot surface algorithm for atomic distances.

for calculating the surface were grid spacing of 3.5 Å and the distance threshold value of about 2.00 Å. These empirical parameters are set at specific values, but can be changed. The refinement of the grid increases the accuracy of the molecular surface. However, the distance threshold value must be restricted in how much it can be increased. As the distance threshold value increases, the resulting molecular surface will lose accuracy of curvature in intricate regions. The results will be discussed in detail in the section on preliminary parameter analysis.

An example of the molecular dot surface constructed from our method is given in Figure 26. We use the atomic representation of the *E. coli* molecule given in the first picture to construct a corresponding molecular surface as shown in the second picture.

IV.3.2. Satisfying Surface Data

Once the dot molecular surface has been constructed, the next stage of the algorithm is to constrain the position of atoms within the protein. The development of the surface constraint is based on the general constraint enforcement procedure discussed in Section III.5. The surface constraint enforcement procedure begins by placing minimum separation spheres around the molecular surface nodes after the local minimization procedure has terminated. The surface nodes define the center of the spheres. During the optimization, the atoms are restricted from moving into a surface violation region. If the atom is inside one or more surface violation regions, then the position of the atom is not currently satisfying surface data and is pushed outside of the sphere using a low function error criteria.

It is important to note that care should be taken in choosing the atom types that

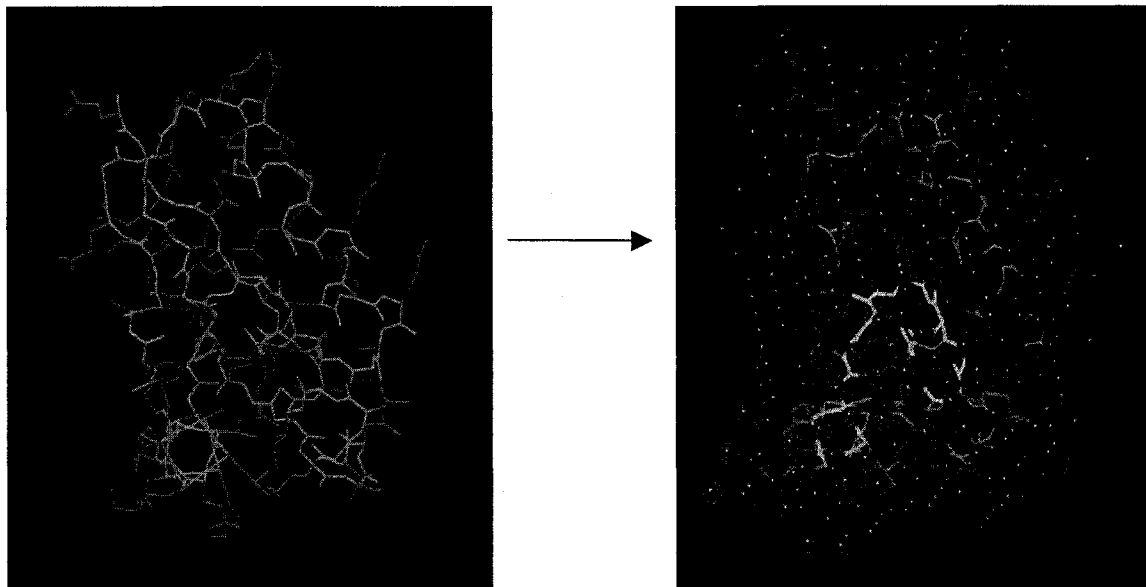


Figure 26. 3D example of constructing a dot surface from an atomic representation.

will be constrained. In preliminary work, we found that enforcing a surface constraint on every atom is too restrictive and results in a collapse of the configuration during the optimization process. For this reason, we chose to constrain only the main chain and side chain atoms of new atomic representation. Since a C_β atom represents the sidechain in our model, we decided to modify the distance cutoff value of these atoms to compensate missing atoms of the sidechains. All atoms that are found inside of surface violation regions are pushed outside the region using the constraint enforcement technique previously described. The algorithm for constraining an arbitrary atom is given in Figure 27.

In light of constraining specific atom types, we also must address the problems of unconstrained atoms shifting outside of the molecular surface periodically throughout the computation. To ensure that atoms do not drift too far outside the surface, the model is shifted after each atom optimization so that the center of mass of the model lies on the center of mass of the original input surface. While this adjustment takes care of atoms going outside of the surface, it does not account for the atomic position satisfying the surface data. Therefore, a rotational alignment is performed to ensure that the unconstrained atoms agree with surface data during the computation.

Algorithm Enforcing Surface Constraint on an Atom

1. Construct the violation sphere for all surface nodes with a user specified radius.
 2. For each surface node. Repeat:
 Compute the endpoints of the intersection of the search line and the violation sphere.
 Compute the distance between the atom and surface node.
 If the distance is less than the radius of violation sphere, store the endpoints as possible step length updates.
 3. Merge all possible step length updates for all surface nodes.
 4. If the atom is inside of any of the violation spheres, move that atom to a position that is outside of all the violation spheres by re-initializing the step length to the possible update that produces the lowest function error value.
-

Figure 27. Outline for enforcing the surface constraint on an atom.

A natural approach to aligning two surfaces is to minimize some error function of misfit such as the squared error, entropy, etc (Besl and McKay 1992). The most common approach is the L_2 -error minimization formulation. In our work, however, the L_2 -error function will produce an alignment that favors regions where there is a higher density of nodes. Another reason L_2 -error function is not employed is because the number of nodes in each set is not equal and we have no defined one-to-one correspondence between the surface nodes in the first set with that of the second set. Therefore, we must choose an error function of misfit that avoids producing an alignment that is biased to the large error in a specific region and that also defines one-to-one spatial coherence between the points in of one set with those of the other set.

We chose to minimize the variance because this will spread the error out evenly around the surface. The error minimization can be described as follows. Given a set of original surface nodes $\{x_i\}$ for $i = 1 \dots n$ and experimental surface nodes $\{p_j\}$ $j = 1 \dots m$, where n and m is the number of points in the original and experimental set, respectively, we find the angles rotational ϕ , θ , and ψ that minimizes the variance function

$$\text{Var}(d_{ij}) = [E(d_{ij})]^2 - E(d_{ij}^2) \quad (4.1)$$

where $E(d_i)$ is the average distance between a node in the original surface and the closest node in the experimental set corresponding to that node.

$$E(d_{ij}) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n d_{ij} \quad (4.2)$$

and each distance, d_i , is defined as

$$d_{ij} = \|x_i - R(\phi, \theta, \psi) p_j\| \quad (4.3)$$

where p_j is the closest node in the experimental set corresponding to the node x_i in the original set, $R(\phi, \theta, \psi)$ ⁶ is standard rotation matrix given by

$$R(\phi, \theta, \psi) = \begin{pmatrix} \cos\phi \cos\theta & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \sin\phi \cos\theta & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi \\ -\sin\theta & \cos\theta \sin\psi & \sin\theta \cos\psi \end{pmatrix} \quad (4.4)$$

We can then find an optimal alignment between the given surfaces by minimizing the function in Equation 4.1 using standard alignment techniques based on principal component analysis (PCA), iterative closest point (ICP), hybrid alignment algorithm, and nonlinear optimization (Besl and McKay 1992; Dugan and Altman 2004).

The advantages of these shape alignment methods are that the convergence is fast and monotonic and the algorithms are theoretically sound. However, the main barrier is that the surface formed from the current configuration does not exactly match the shape or size of the original input surface and thus, these algorithms do not work properly for some molecular shapes (Besl and McKay 1992). As a result, the surface alignment algorithm will fall into local minima traps.

In light of these problems, we employ a non-gradient method for aligning molecular surfaces that does not get hung up in local minima. An outline for our algorithm is given in Figure 28. In implementing this method, there is one parameter that can be used to increase the accuracy of the rotational alignment. The partition threshold number, $n_{threshold}$, is the number chosen to divide each angle interval uniformly to ensure the best reduction in bracket length per step. If the partition threshold number is fixed, the

⁶ The angles ϕ and ψ are rotational angles and are different from the torsional angles discussed in Chapter III.

Algorithm for Rotational Alignment

1. Set an error function value threshold cutoff.
 2. Fix $n_{steps} = n_{threshold}$. Partition the ϕ rotational angle interval $(-\pi, \pi)$ uniformly. Initialize the torsional ϕ to $-\pi$ degrees.
 3. For each ϕ test value, fix $n_{cuts} = n_{threshold}$. Repeat:

Partition the θ rotational angle interval $(-\pi, \pi)$ uniformly. Initialize the torsional θ to $-\pi$ degrees.

For each θ test value, fix $n_{cuts} = n_{threshold}$. Repeat:

Partition the ψ rotational angle interval $(-\pi, \pi)$ uniformly. Initialize the torsional ψ to degrees.

Update the coordinates of the experimental surface nodes by transforming the surface nodes according to the current values of ϕ, θ , and ψ

Find the closest node in the experimental set corresponding to the node in the original set

Evaluate the variance function;

If the function value is less then the minimum threshold,

Store the function value, the corresponding surface nodes, and the corresponding ϕ, θ , and ψ .
 4. Choose the group of ϕ, θ , and ψ values that correspond to the lowest variance function value.
 5. Update the molecular surface by re-initializing the coordinates of the surface nodes to those corresponding to the rotational angle value that generate the lowest function error value.
-

Figure 28. Outline for finding the optimal alignment for an original and an experimental set of surface nodes.

number of operation would $O(n^3)$. To reduce the computational time spent in function evaluation per step, we use an adaptive bracketing method that allows us to reduce the length per step as we get close to a very small function value.

Our method does not compare with the computational efficiency of the standard alignment method such as ICP and advanced ICP. But, the advantage is that it is easily

implemented, avoids local minima entrapment, and produces good alignment for eccentrically shaped protein surfaces. In the future, we can investigate how implementing a better shape-matching algorithm could possibly improve the success of the surface constraint.

Figure 29 shows an example of aligning an original molecular surface and an experimental molecular surface. We use the atomic representation of the *E. coli* molecule to illustrate effectiveness of our approach. The first picture represents the surface nodes of the original protein (red), which is constructed using our molecular surface algorithm above. The second picture shows the experimental molecular surface (yellow) computed from a model configuration. In comparing these pictures, notice that the molecular surface shapes are not exactly the same and there is a possibility for one surface to have more nodes. Our method is still able to produce an optimal alignment for the two dot surfaces while ensuring that the positions of the unconstrained atoms also satisfy surface data.

IV.4. Validation

The goal of the experiments was to evaluate the effectiveness of a molecular dot surface in satisfying surface data. The information used in the experiment was derived from proteins with known three-dimensional structure that were taken from the Protein Data Bank (Berman *et al.* 2000). Therefore, only the GNOMAD code is used in the validation experiments, in order to evaluate effectiveness of the surface constraint compared to angle-constrained optimization. Because the surface constraint requires three empirical parameters in implementing the surface constraint, a section on preliminary results from parameter analysis has been included before the result section. For the purpose of this work, an optimization experiment consists of an attempt to recreate the known crystal structure using distance and surface information. We generate a set of input distances by first including all the fixed " i to $i+1$ " and " i to $i+2$ " interatomic distances (assuming relatively constant bond lengths and bond angles), all torsional angles in the α -helix and β -strand regions (assuming these secondary structures are known), the omega torsional angles in the loop region, and short-range contact distances (SRD). The surface shape information consists of a set of surface nodes generated from the original structure. The choice of this input data set is to include information

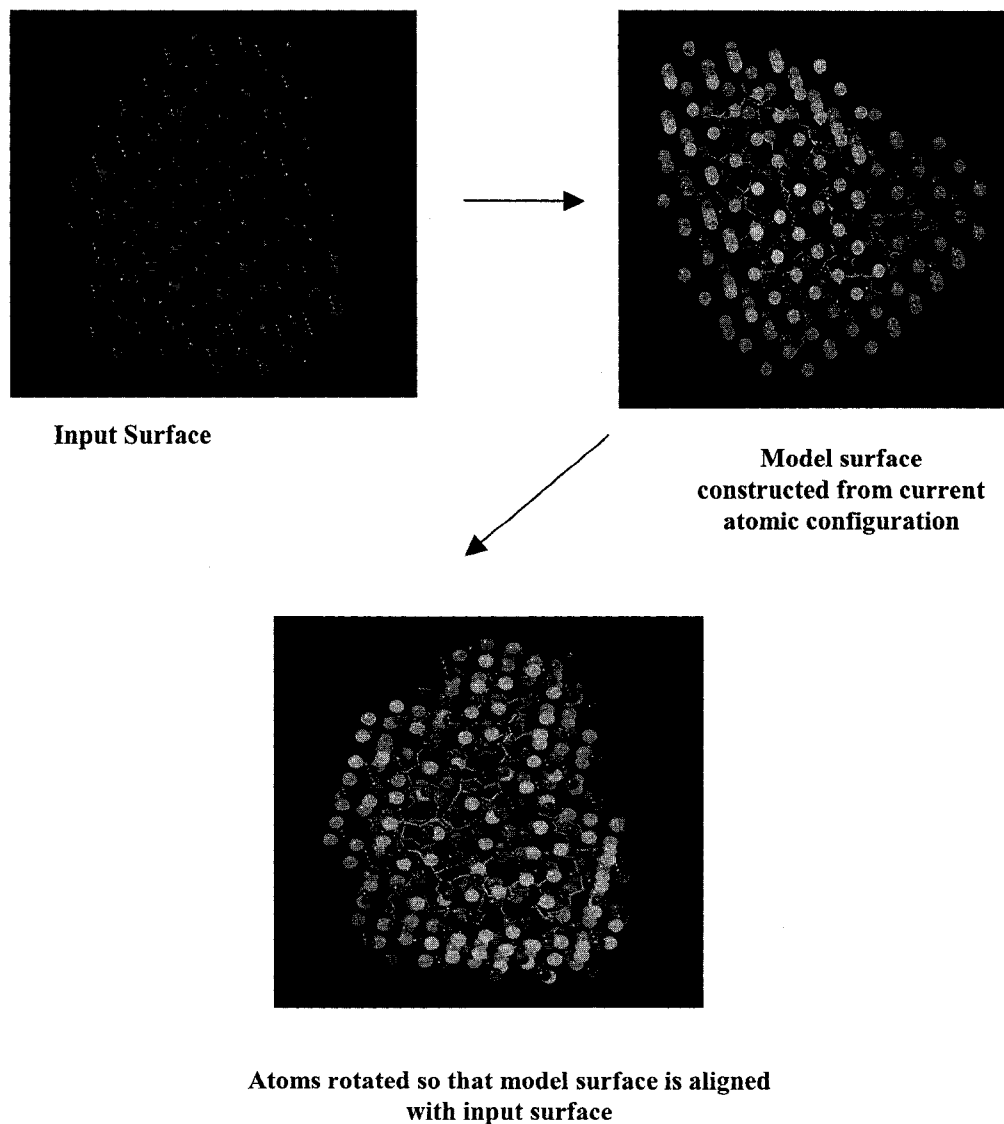


Figure 29. 3D illustration of satisfying surface data using translational and rotational alignment.

associated with the major secondary structures and to provide partial folding information about the protein.

In the preliminary section, we study the relationship between the surface constraints and the quality of the structure as we vary the empirical parameters: grid spacing, distance threshold cutoff, and radius of the violation spheres. Then, we look for the optimal set of parameters that will yield improved structure over a wide range of available input data. In the second section, the main aim of the computational experiments is to show that our surface constraint provides a reliable and effective

approach in bringing major secondary structure together and in improving the folding of the protein. For this reason, a subset of proteins for which at least 40% of the all amino acids are found in helices and strands were chosen as test cases.

The Structural Classification of Proteins (SCOP) (Murzin *et al.* 1995) database provides information regarding classes of proteins based upon the inclusion of structural elements such as α -helices and β -sheets. The classes $\alpha+\beta$ or α/β are two SCOP classes that provide information regarding the inclusion of structural elements such as α -helices and β -sheets. The classes all- β and all- α are two classes based upon the inclusion of all β -sheets or all α -helices, respectively⁷. Six proteins were chosen for testing our surface constraint, all of which belong to either the $\alpha+\beta$ or α/β SCOP classes. The total number of amino acids and nonhydrogen atoms for the various test structures is given in Table 5.

Accuracy is measured in terms of the RMSD and the distance residuals. In the absence of a “known” structure – the only error measure available is the distance residuals, which represent how well the optimization satisfies the input distance data. When a “known” structure is available, as is the case in this work, an RMSD can be measured to compare the accuracy of the estimated structure to the known structure.

IV.4.1. Preliminary Analysis of Empirical Surface Parameters

In this section, we investigate the dependence of the solution structure on the variations of three empirical parameters used in implementing the surface constraint. We find the optimal empirical parameters that aid in the improvement of the quality of an initial test protein, *E. coli* protein (ID code 1CTF). We will use these empirical parameters throughout results section for other test proteins. This is not to say that these values are optimal for every protein in $\alpha+\beta$ or α/β SCOP classes, but is merely a way of gauging the effectiveness of our surface constraint in improving the quality of structure for proteins that are comprised mostly of helices and strands.

In developing the surface constraint, three empirical parameters are crucial to the effectiveness of using a dot molecular surface to improve protein structures: grid spacing, distance threshold cutoff, and radius of the violation spheres. The grid spacing and distance threshold cutoff parameters serve as a way of refining the original and update

⁷ Note that the all- β and all- α proteins were excluded from this investigation because enforcement of the surface constraint will result in structural collapse due to the distance threshold cutoff.

Table 5. Test structures mostly comprised of helices and strands.

Structural Type	PDB ID	# of Residues	# of atoms	
			Mainchain	New
$\alpha + \beta$	1ozz	43	170	572
	1ctf	68	266	884
	1aw0	71	282	936
	1fvs	71	285	936
	1aps	97	384	1274
α / β	1bta	89	351	1157
	1ay7B	89	351	1157

surfaces. The grid spacing helps to tighten and loosen the surface. That is, the closer the grid nodes, the more accurate the curvature of the surface of the protein. The distance threshold cutoff allows us to improve the accuracy of the surface by determining the amount of grid nodes used in creating the surface. Too many grid nodes will increase the computational complexity. On the other hand, an insufficient amount of grid nodes will cause the surface to be too porous and thus result in many atoms moving outside of the surface during the optimization.

The last empirical parameter, radius of the violation sphere, is the means by which we can move the atom relative to surface and aids in producing a structure that agrees with surface data. The choice of the radius of the violation sphere is very important because the radius can be used as a tool for choosing the best position of atoms. The choice of the parameter cannot be too large or too small. If we start with the radius large, then in regions where there is insufficient distance information the atoms will be compressed together, and in regions where there is sufficient information the surface constraint will cause a conflict in satisfying the data set. If we choose the radius too small then the surface constraint will not be effective in moving major components of the protein into agreement with the surface. In both of these cases, the optimization will be unable to yield a good quality structure that agrees with all input data.

In this work, we use one specific value for the radius of all violation spheres, independent of the atom type. This will have the total effect of moving all major

components into agreement with surface and into agreement relative to each other based on other input data. In the section on future work, we discuss a method for choosing the radius to capture the hydrophobic interaction, in which the radius shrinks and expands proportional to the atom's solvent accessibility condition. This involves performing a statistical analysis on the distances between different atom types and water molecules. This information can then be paired with solvent accessibility information to help position atoms to satisfy solvent accessibility conditions and model hydrophobic interactions.

Experiment 1

In our first computational experiment we investigated the effects of changing the grid spacing, g , on the quality of the protein structure with a specific distance threshold cutoff, d , and radius of the violation sphere, v . We varied g over $[2.50, 3.75]$ because any value below this point will be too confining for building a protein structure and result in a collapse in the protein structure before enforcing the surface constraint, and any value above this point will produce a surface that is ineffective in moving the major components of the protein. Results show that the surface constraint algorithm fails to improve the quality of the structure in almost every case except for $g = 3.5 \text{ \AA}$. Based on these observations, grid spacing $g = 3.5 \text{ \AA}$ is the best choice for the surface constraint, when only partial folding information is employed. When varying the grid spacing, the best RMSD we can expect is around 5 \AA because the angles in the loop region are almost rigid and only vary a little from their original position.

Experiment 2

The primary aim of our second experiment was to gauge the utility of the distance threshold cutoff in improving the effectiveness of the surface constraint and in reducing the lower bound RMSD found at $d = 3.5 \text{ \AA}$. The distance threshold cutoff parameter determines the number of grid nodes used in creating the molecular surface and is important in capturing the intricate curvature of the molecular surface. We investigate the results of how small change in d affects the performance of our surface constraint by varying d over $[1.80, 2.20]$ for the problem with $g = 3.5 \text{ \AA}$ and $v = 1.5 \text{ \AA}$.

Results show that varying the distance threshold cutoff does not decrease the lower bound RMSD. More noticeably, the more we move from $d = 2.0 \text{ \AA}$ the worse the

RMSD becomes. This result is to be expected because as d decreases below 2.0 \AA , the amount of surface nodes used in creating the dot surface decreases and accordingly, results in atoms being able to move outside of the surface during the optimization. On the other hand, an increase in d above 2.0 \AA results in more grid nodes farther away from the protein structure being included in computing the dot surface. Consequently, the molecular surface becomes imprecise and unclear and thus, the solution structure satisfies unreliable surface data.

The best choice for the distance threshold cutoff value for grid spacing of $g = 3.5 \text{ \AA}$ is about 2.0 \AA . The results show that that the grid spacing and distance threshold cutoff parameters are dependent upon each other. The general observation that can be made is that change in distance threshold cutoff is only needed when the grid spacing is changed. As the grid spacing decreases, it may be possible to determine another value for d that will yield a slightly improved corresponding RMSD. In general, we cannot expect major improvement in these cases because as the grid spacing decreases the molecular surface becomes too confining and thus, the change in d will be negligible.

Experiment 3

In the previous two experiments, we considered how the exactness of the molecular surface relates to improvement in the quality of the protein. We found that the most reliable molecular surface for our set of partial folding information is constructed from grid spacing, $g = 3.5 \text{ \AA}$ and $d = 2.0 \text{ \AA}$. Choosing this surface ensures that the constraint will be effective enough to allow the structure to fold without the problem of structural collapse and at the same time the surface is compact enough to aid in bringing some the major components together while avoiding collision.

In general, the molecular surface allows us to identify a domain for which we can expect that many structures will satisfy both distance and surface data. But in the case that surface proximity information is available, the additional grid spacing and distance threshold cutoff parameters is enough to aid in moving atoms that are near or far away from the surface. In this experiment, we study the behavior of the structure as the radius of the violations spheres, v , varies over $[1.40, 1.65]$. We choose this interval because choosing the radius to small will not be effective in moving buried atoms to the core of the protein and choosing the radius to large would result in pushing all atoms to the core,

causing a collapse in the structure.

To get a general idea of how effective the parameter is, we first consider the case for which each violation sphere will have the same radius, ν . For $g = 3.5 \text{ \AA}$ and $d = 2.0 \text{ \AA}$, the results indicate that moving outside of $[1.45, 1.6]$ result in the RMSD worsening. The behavior is justified by noting that the position of every atom inside of violation sphere is determined based on the same radius, ν , which could lead to over-expanding or over-compressing the atoms away from or into the core of the structure. In addition, we find that choosing a radius in the interval $[1.45, 1.6]$ tend to yield structures that are similar to the structure corresponding to $\nu = 1.5 \text{ \AA}$.

Using $\nu = 1.45 \text{ \AA}$ results in reducing the RMDS by 1 \AA . This is not a large improvement, but the results are significant because it shows that proper choice of the radius of the violation spheres will result in improvement in the structure beyond the lower bound RMSD for $g = 3.5 \text{ \AA}$. In addition, it gives us motivation that if we adopt a more stringent criterion for choosing the radius of the violation sphere, ν , such as solvent accessibility, surface proximity, etc, the position of atoms in the loop regions would improve and thus, an overall improvement in the RMSD is possible. From this experiment, we conclude that if we are going to use the same ν for each violation sphere the optimal choice is $\nu = 1.5 \text{ \AA}$.

Experiment 4

For the initial test protein, the optimal values of grid spacing, distance threshold cutoff, and radius of the violation spheres, are 3.5 , 2.0 and 1.5 \AA , respectively. In the last experiment, we investigate the effectiveness of the surface on the other test protein structures. For the purposes of the last experiment, we set the empirical parameters to the values found in using the 1ctf test structure. In addition, we considered the case for which the short-range contact distances between connected or unconnected topological neighboring β -strands are available. This will help us to avoid the problem of dangling strands at the beginning or end of the test structures. The choice of these contact distances also provided a minimum amount of folding information. The results are given in Table 6.

We found that the molecular surface is effective in improving only one of the protein structures using this set of empirical parameters. In many cases, the surface

Table 6. Preliminary results for surface constraint enforcement.

PDBID	No of atoms.	RMSD w/angle	RMSD w/angle/surface
1ozz:	572	7.4323	5.9442
1ctf:	884	7.1105	7.5885
1fvs:	936	7.2874	7.3280
1aw0:	936	6.5310	10.8349
1bta:	1157	13.4859	14.4701
1aps:	1274	13.1267	12.1842
1ay7B:	1157	10.4978	12.6107

constraint is unable to improve the position of major secondary structures. The inability of the surface constraint to improve the position of this component is the chief contributing factor to the worse RMSD. The preliminary results indicate that surface is not effective for this set of data at these chosen values of empirical parameters. In light of the observation that the test structures become more complex as the number of amino acids increase, these results are indicative of the observation that surface information paired with only a minimum set of contact information between strands cannot always provide a significant improvement in the quality of the protein.

We could choose another set of parameters that would work for a larger percent of test protein. But the more subtle approach is to investigate the effectiveness of the surface constraint, for the current set parameters, as more contact distances are added. We expect that as we increase the amount of contact information, the surface will become more effective in improving the quality of the structure.

IV.4.2. Results

The purpose of this next set of experiments is to gauge the effectiveness of our surface constraint when more contact information is available. In specific, we want to verify that as the amount of short-range contact information increases, the surface constraint for the chosen value of the empirical parameters maintains or improves its effectiveness in folding the protein and does not result in being too confining for the movement of atoms. This experiment is significant because it provides validation for the observation that as protein structure changes and the corresponding contact information

increases, the chosen values of empirical parameters found in the previous experiments are optimal for any set of contact distances.

We investigate the effect of increasing the amount of contact information between connected and topological neighboring β -strands and α -helices. Therefore, the distance set discussed above is augmented with four sets of short-range contact distance sets. The short-range contact distance sets are described as follows:

1. The first set of contacts is the set of distances between two connected or unconnected topological neighboring β -strands.
2. The second set of contacts is made up of the set of distances between two connected or unconnected topological β -strands or between connected or unconnected topological β -strands and α -helices.
3. The third set of contacts is made up of the set of distances between two connected or unconnected topological β -strands, between connected or unconnected topological β -strands and α -helices, or between two connected or unconnected topological α -helices.
4. The fourth set of contacts is the set of all contact distances between all amino acids in the test structure.

The data sets are designed with exact information to evaluate the ability of the algorithm to use the surface constraint to produce improved structures (as compared to structures computed based on distances alone). We first compute structures based on the distance data sets alone and then compute structures based on these distances with the surface constraint.

Figures 30 – 35 present accuracy results for six of the seven test proteins. Accuracy is reported in terms of RMSD and distance residual error. For each test structure, we conduct the tests with grid spacing of $g = 3.5 \text{ \AA}$, distance cutoff threshold of $d = 2.0 \text{ \AA}$, and radius of violation sphere, $v = 1.5 \text{ \AA}$. Each figure displays two plots. The plot on the left in each figure shows results for the RMSD and the plot on the right show results for the distances residual error. Each plot contains two lines, one for the angle-constrained optimization and one for angle/surface-constrained optimization. Angle-constrained performance refers to the mean deviation from the original structure computed from the distance and angle subsets only. The angle/surface-constrained

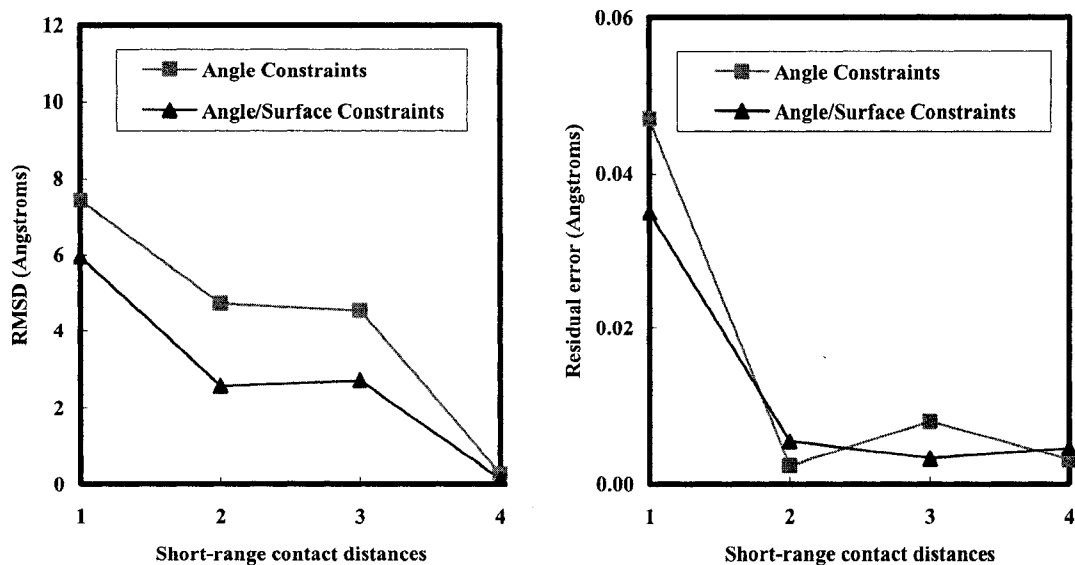


Figure 30. Comparison results on the 1ozz protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

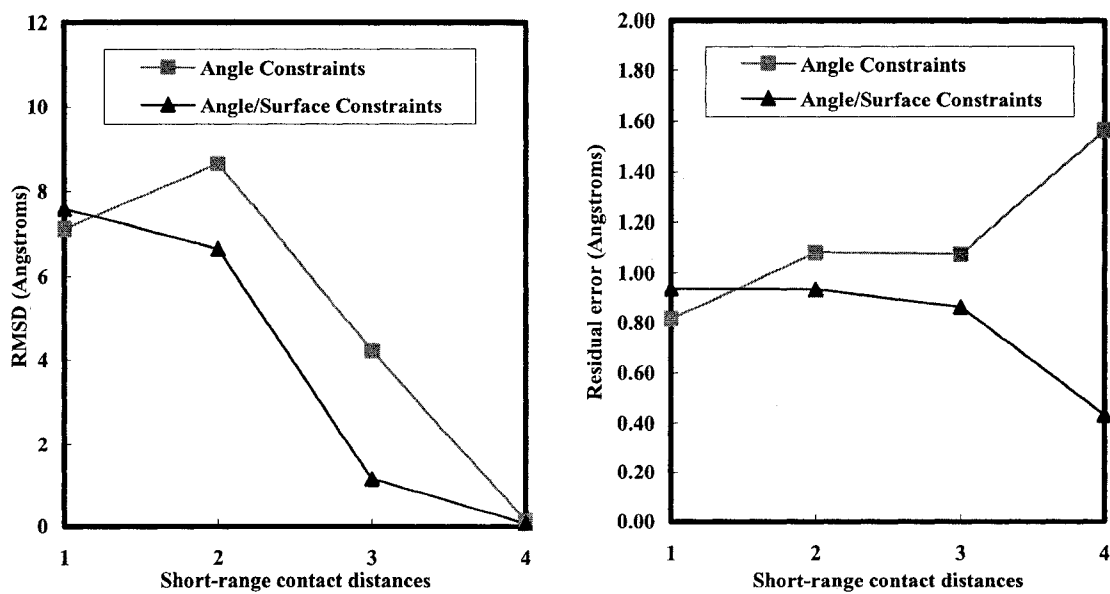


Figure 31. Comparison results on the 1ctf protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

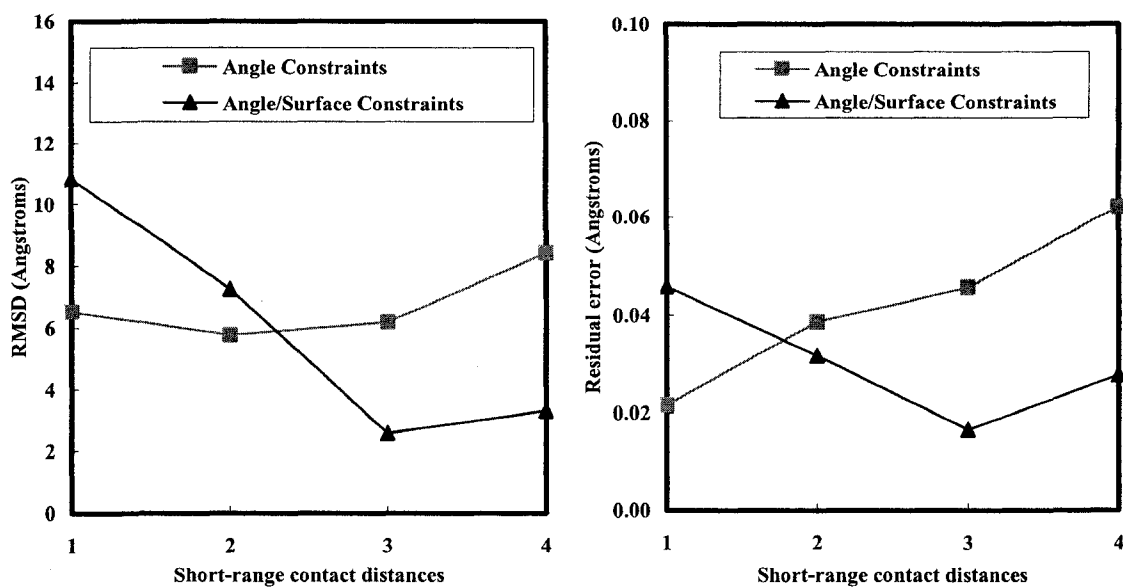


Figure 32. Comparison results on the 1aw0 protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

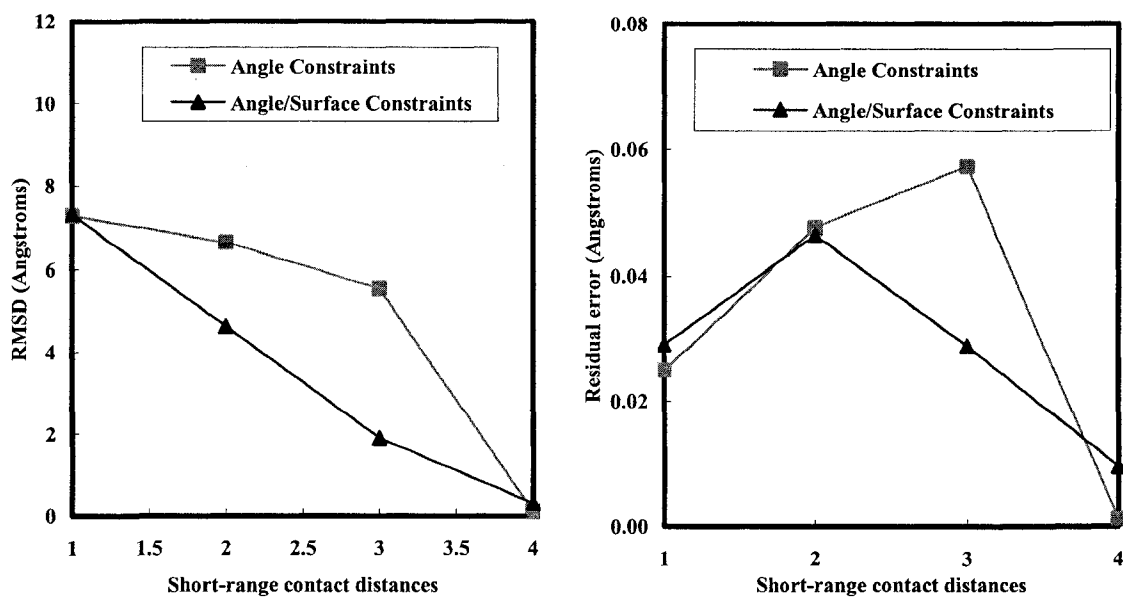


Figure 33. Comparison results on the 1fvs protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

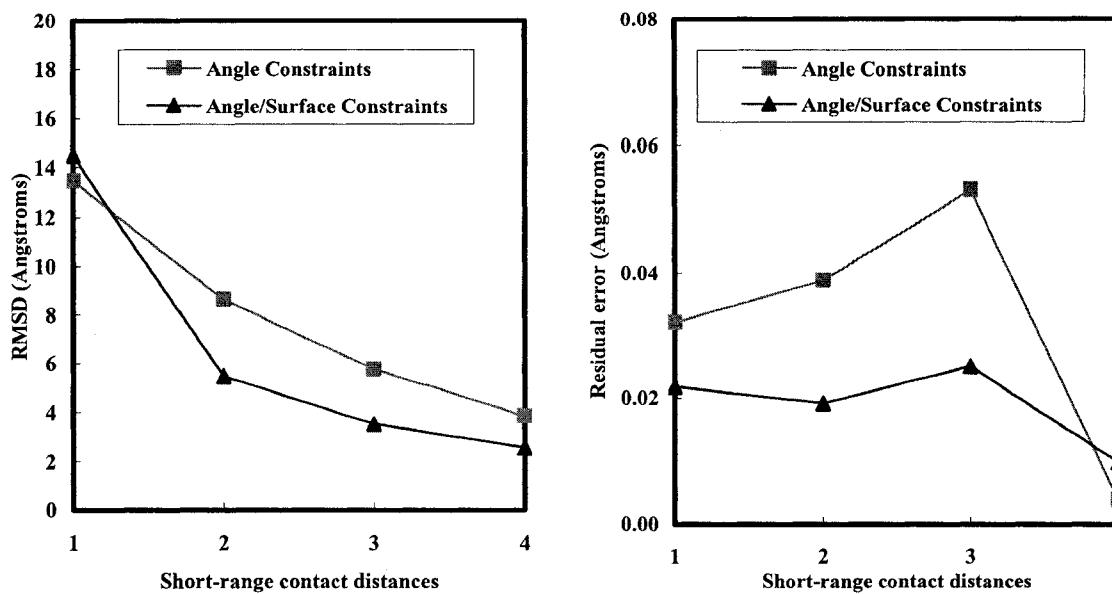


Figure 34. Comparison results on the 1bta protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

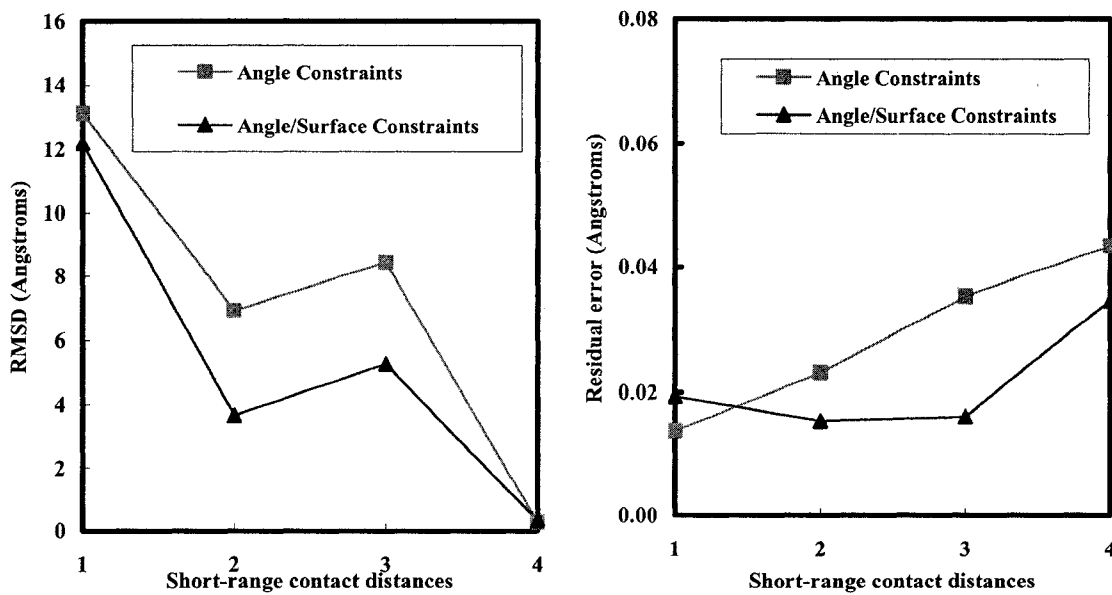


Figure 35. Comparison results on the 1aps protein of RMSD (left) and residual error (right) for constrained angles and angles/surface constrained GNOMAD.

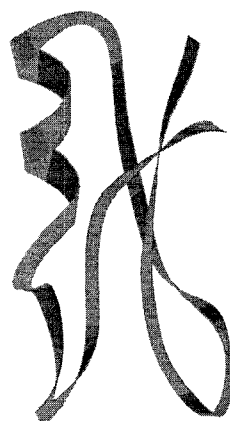
performance refers to the mean deviation from the original structure computed with additional surface data.

To illustrate the effects of enforcing the surface constraint as we increase the amount of contact distances, Figures 36 – 38 show a qualitative comparison of structures resulting from angle constrained and angle/surface constrained GNOMAD runs, and the relative proximity to the crystal structure. These results are for 1ctf, 1aw0, and 1bta protein with using the third set of contact distances.

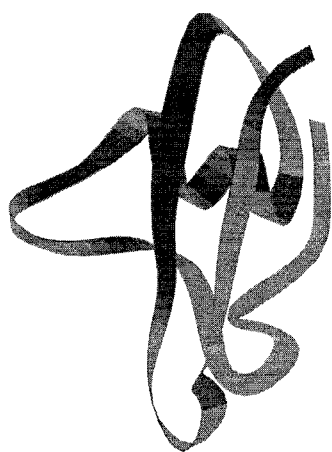
IV.4.3. Discussion

The RMSD results in Figures 30 – 35 show that by increasing the amount of contact distances, the surface constraint method is more effective in improving the accuracy of the protein. For data set 1, the surface constraint only improved the RMSD of one proteins (See preliminary parameter analysis). With the addition of more contact distances between two connected or unconnected topological neighboring β -strands and α -helices (data set 2), however, the surface constraint is able to improve the RMSD in five out of the six test structures. For these test structures, the additional contact distances provided more folding information. However, many of the helices in these test structures are tilted out of correct alignment with their true position, and thus result in a higher RMSD from the true structure. The addition of the surface constraints allows for the correction of most of the tilted helices and also decreases the RMSD by 2 Å.

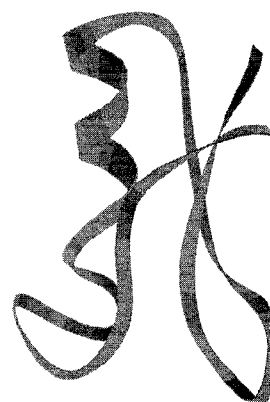
One exception to the effectiveness of the surface constraint, for this set of data, is the 1aw0 test protein. Consider Figure 32. The test structure without the surface constraint enforcement is 5.77 Å. The addition of the surface constraint results in a structure of 7.26 Å. Even so, this structure is close to the correct structure but one helix is significantly titled out of alignment. Because the helix is closer to the end of the protein and the short-range contact distances does not provide enough information to bring the helix close to the other secondary structures, the surface constraint worsens the position of the helix, and consequently, the RMSD increases by 2.5 Å. Increasing the contact distances while enforcing the surface constraint for this structure, results in a decrease in the RMSD of 2.59 Å. Nevertheless, the result of 1aw0 suggests the surface constraint is effective when enough contact distances are available.



Crystal Structure

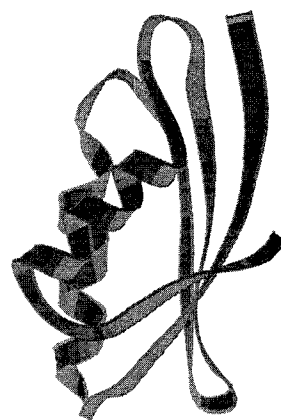


Not Surface Constrained
(max. distance constraint error = 1.07 Å
RMSD = 4.22 Å)

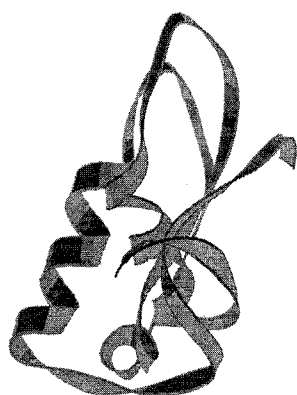


Surface Constrained
(max. distance constraint error = 0.86 Å
RMSD = 1.14 Å)

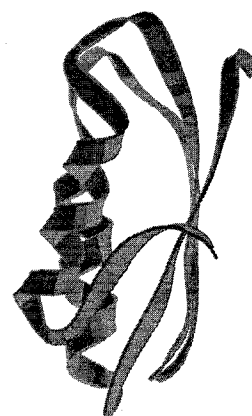
Figure 36. 1OZZ protein structure and surface constraint enforcement results. Qualitative effects surface constraint satisfaction for short-range contact distances between two major secondary structures.



Crystal Structure

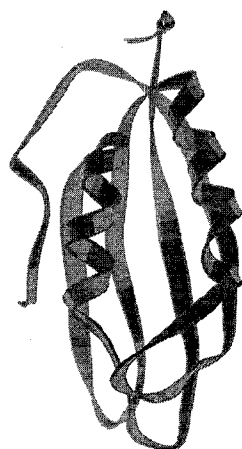


Not Surface Constrained
(max. distance constraint error = 1.85 Å
RMSD = 6.20 Å)



Surface Constrained
(max. distance constraint error = 0.97 Å
RMSD = 2.59 Å)

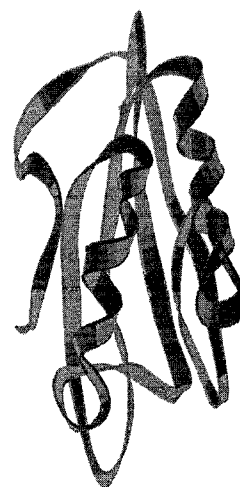
Figure 37. 1AW0 protein structure and surface constraint enforcement results. Qualitative effects surface constraint satisfaction for short-range contact distances between two major secondary structures.



Crystal Structure

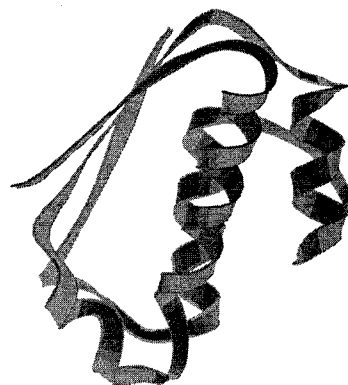


Not Surface Constrained
(max. distance constraint error = 1.61 Å
RMSD = 8.42 Å)

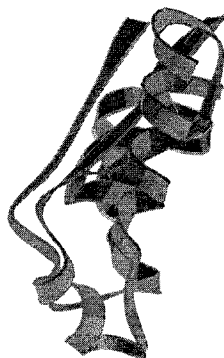


Surface Constrained
(max. distance constraint error = 1.10 Å
RMSD = 5.26 Å)

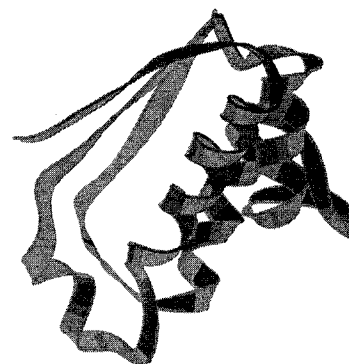
Figure 38. 1APS protein structure and surface constraint enforcement results. Qualitative effects surface constraint satisfaction for short-range contact distances between two major secondary structures.



Crystal Structure



Not Surface Constrained
(max. distance constraint error = 1.41 Å
RMSD = 5.75 Å)



Surface Constrained
(max. distance constraint error = 0.92 Å
RMSD = 3.54 Å)

Figure 39. 1BTA protein structure and surface constraint enforcement results. Qualitative effects surface constraint satisfaction for short-range contact distances between two major secondary structures.

In the data set 3, the surface is able to improve the structure for all six of the test proteins. A reason for this improvement is that the contact distances provide enough information to place most of the helices and strands in the correct spatial proximity with respect to each other although some of the helices may be tilted out of correct alignment with their true position. Hence, the surface is able to make good use of this information in correcting the tilt of helices and results in an improvement in the folding of the protein structure.

Not only is it important to ensure that the surface constraint works for cases where there is limited contact information, it is also significant to make sure the surface constraint does not hinder effectiveness of the algorithm when sufficient contact distances are available. We test this by including enough contact distances to ensure that the test structure would form without the assistance of the surface constraint. When the surface constraint is enforced, there is an improvement in a few of the cases. In the other cases, the surface constraint does not conflict with any of the distance data and the algorithm is still able to maintain its effectiveness in forming a good structure.

The RMSD is a measure of accuracy and effectiveness of the surface constraint when the true structure is known, but in the case where the true structure is unknown, the residual error serves as a measure of accuracy. In addition, the distance residual error serves as a way of measuring the performance of the optimization. For this reason, the distance residual error is also considered in the plots on the right in Figures 30 – 35. The results show that the surface constraint is able to maintain or improve the residual error for more test structures. Interestingly, for the 1aw0 protein, the behavior of the RMSD and residual error curves are so close that if the true test structure were not known the residual error could be used as a measure of how close our structure is to being correct.

Upon considering the other plots, it is noticeable that for the other test structures, the RMSD and residual are not as closely connected. However, the enforcement of the surface constraint results in a considerable decrease in residual error. For the 1ctf, 1bta and 1aps protein, as can be seen in Figures 31, 33, and 34, the surface constraint results in a downward shift of both the RMSD and residual error curves. For the 1ozz and 1fvf, the unchanged behavior of residual error shows that surface constraint maintains the global convergence of the algorithm.

The qualitative results presented in Figures 36 – 38 show what typically happens when additional contact distances are used as input to GNOMAD with the surface constraint enforcement method. These contact distances are between α -helices and connected or unconnected topological β -strands. For each of the test problems, it is apparent that better structures are found when the surface constraints are added. The use of angles provides significant improvement in the formation of the local substructure found in the protein. The addition of the surface constraint provides the additional folding information to improve the position of these local substructures in reference to other local substructures found in the protein.

IV.5. Conclusion

In the previous chapter, we incorporated torsional information to improve the formation of major secondary structures. However, the torsional information in the loop region is not always available. For this reason, additional structural information is needed to give support to improve the folding of the protein. In this section, we propose a sensible method for using a molecular dot surface to improve the quality of protein structures. This method is the beginning of developing constraints, in the context of GNOMAD, from other types of structural information to aid in bringing secondary structures, β -strands and α -helices, together and thus, aid in folding of the three-dimensional structure.

In order to do this, we assume that all torsional angles in the β -strands and α -helices are known and we study the impact that the surface constraint can have on correcting the position of these structures within the protein. The choice of the data sets was made to provide some partial folding information associated with connected or unconnected topological secondary structures. A critical factor in the effectiveness of the surface constraint is the value of three empirical parameters: grid spacing, distance cutoff threshold, and radius of violation spheres. Choosing the optimal empirical parameters will ensure that the constraint will be effective enough to allow the structure to fold while avoiding the problem of structural collapse, and at the same time the surface is compact enough to aid in bringing some of the major components together while avoiding collision.

In general, the quantitative and qualitative results indicate that the surface constraint offers increased accuracy in estimating three-dimensional protein structure. Results also show that when the amount of contact distances are insufficient for properly folding the protein, the surface data provides additional information that aids in the folding of the protein without incurring significant distance residual error. In addition, the surface constraint enforcement method increases in effectiveness with the increase in contact information between two connected or unconnected topological substructures at user-defined empirical parameters. Tests of this new surface constraint enforcement method performed on several different proteins in α/β or $\alpha+\beta$ show that this approach is very robust. The GNOMAD algorithm, with angle and surface constraint enforcement, yields significant improvements in terms of error and RMSD.

This work extends the work in the previous chapter on improving secondary structure formation, provides an algorithm that satisfies a combination of angle, contact and surface data, and has good global convergence properties. This work will serve as a basis for further development of the algorithm to include constraints such as shape, solvent accessibility, and volume.

CHAPTER V

COMPARATIVE MODELING USING GNOMAD

V.1. Introduction

Inexact information can be gathered from many sources and the quality of the data varies for different sources. For example, data taken from X-ray crystallography is generally very precise; however, data taken from NMR has a higher measure of inaccuracy (Chen 2000; Andrec *et al.* 2007). Even for statistical methods, such as comparative modeling, the quality of the structural data may have varying levels of reliability because of the level of structural similarity within other proteins. However, these methods are common sources for much of the information used in structural estimators (Altman 1985; Wu 1996; Chen 2000).

For this reason, current modeling efforts focus on developing a molecular structure estimator that is effective in using a combination of structural data taken from multiples sources (Altman 1985). The purpose of the last part of this dissertation is to show the effectiveness of GNOMAD in estimating protein structures with more realistic structural data. Therefore, we broaden the scope of our work on predicting three-dimensional structure using GNOMAD structural estimation framework to comparative modeling. Proceeding along this avenue is advantageous for two reasons.

Primarily, the weighted nonlinear least squares formulation employed in the GNOMAD structural estimation framework allows us to successfully use combined experimental and statistical data by giving highly precise data more influence than less accurate data in estimating the three-dimensional structure. The second advantage is the use of pseudo-atoms allows for effective use of torsional information in conserved or low-variance regions and, consequently, ensures proper formation of uniform local substructures within the protein and the additional benefit of improved formation in nonuniform regions for protein structures that have closely related family members, (*e.g.* coil). In addition, short-range contact distances between psuedo-atoms replaces long-range constant distances between main chain atoms as input in GNOMAD and thus, aids

in improving the formation of more intricate substructures, such as β -sheets and β - α - β structures.

In this chapter, we discuss our method for collecting and incorporating data taken from reference proteins to estimate the three-dimensional protein structure. Starting with a sequence alignment of the target protein and its closely related family member, low-variance statistical data is generated using standard statistical methods (*e.g.* frequency distribution). The GNOMAD structural estimation framework is used to estimate the three-dimensional structure of the target protein. We test the algorithm using experimental torsional angles to estimate the protein structure and by comparing the generated protein structure with the known crystal structure. In both cases, we include additional short-range distance (SRD) information between atoms that are found in amino acids that are far apart in the amino acid sequence. To demonstrate the effectiveness of our algorithm in using statistical structural data, we estimate the three-dimensional structure of several proteins.

This chapter is organized as follows: in Section V.2, we provide background on statistical methods for adapted for estimating for the three-dimensional protein structure. In Section V.3, we describe our statistical method for taking a sequence alignment and extracting the data we need for GNOMAD run such as, distances and angles (the angles will be used to produce the distances that will satisfy those angles). Section V.4 presents some experimental results of combining experimental and statistical data using GNOMAD structural estimation framework. We conclude the chapter in Section V.5.

V.2. Statistical Modeling Efforts for Estimating Protein Structure

Statistical methods are the most widely used methods for predicting the structure of a protein when experimental methods only contribute a small amount of information about the protein under consideration (Sali and Blundell 1993). The reasoning behind the development of these techniques is that structural data can be collected about the protein under consideration based on sequence and/or structural similarities of known three-dimensional structures. Because these methods provide generalized information about the regions that are common to many proteins, structure estimation frameworks have been designed to make use of the whole or a subset of the body of known protein structures in order to estimate an unknown three-dimensional structure for a protein sequence.

Statistical modeling is separated into two categories: comparative and empirical modeling. In the comparative modeling (or homology modeling) approach, structural data is generated based on the characteristics of closely related family members of the protein under consideration. Predicting the structure of the protein using comparative modeling approach consists of four steps illustrated in Figure 40. The primary and most important step is to find known protein structures, also referred to as reference proteins, whose sequence has at least 70% sequence identity with the sequence of the target protein. Second, multiple alignment of these sequences with the target sequence is performed in order to estimate the correspondence between equivalent amino acids in each structure. Thirdly, the known structures are used to estimate the expected value of structural data. Lastly, a model is built from statistical data taken from conserved regions and assessment of the model is performed based on accuracy and error of sequence alignment.

The most commonly used comparative modeling method is spatial restraint satisfaction (Furnham *et al.* 2008). In using this approach, structural data is collected via a statistical analysis on conserved regions or regions high in structural similarities, a model violation function is developed from this structural data, and the three-dimensional structure is determined by minimizing the violation function using well-known optimization methods such as conjugate gradient. But, one of the major limitations of comparative modeling is that the quality of the structural data is dependent on how closely related (*i.e.* sequence or structural similarities) the reference proteins are to the protein under consideration. That is, the accuracy of the data is directly proportional to the closeness of the target protein and its family members (Sali and Blundell 2003; Furnham *et al.* 2008). Therefore, improvements in comparative modeling methods tend to focus more on the improvement of sequence alignment and refinement of the model violation function phases. This is due to the observation that the accuracy of the structure could possibly be compromised by incorrect alignment or inaccurate representation of geometric features (Wall *et al.* 1999).

In some cases, comparative models are not helpful because some proteins have no closely related family members. In these cases, empirical methods are employed to get a basic idea of the three-dimensional structure. The underlying principle of empirical modeling methods is similar to that of comparative modeling but differs in the fact that

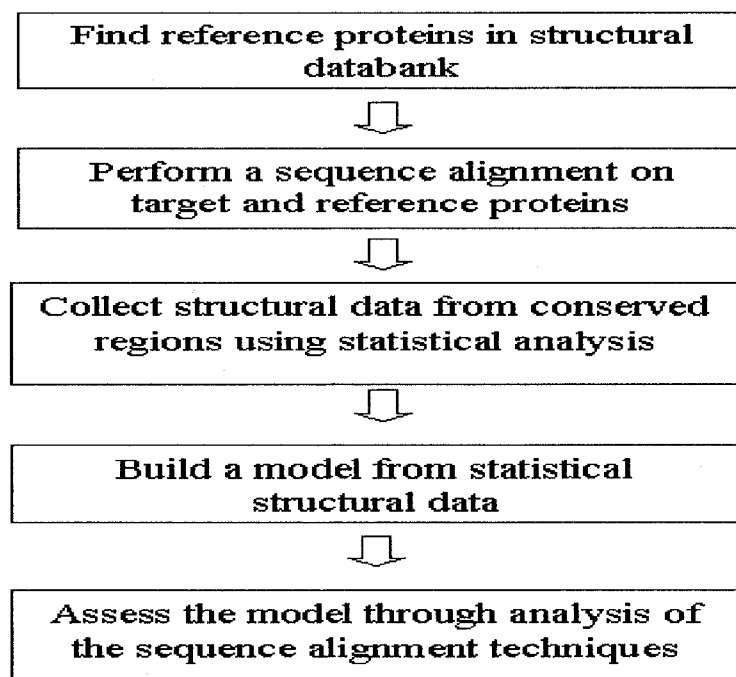


Figure 40. Flowchart illustrating the basic principal of comparative modeling method by satisfaction of spatial restraint.

the criteria of sequence similarities in family members is relaxed and structural similarities of any known protein in the databank becomes the main basis for statistical analysis and thus, the determination the three-dimensional structure. In empirical modeling, a statistical analysis is performed on all protein whose structures are known in a structural data bank. The data is, then, used to determine functions, referred to as knowledge-based functions, for representing the major features and forces that contribute to the fold of the proteins including hydrogen bonding, and van der Waals interactions.

The advantage of using these statistical approaches is that they provide a rich amount of reliable local and global information about the protein structure that is not available through experimental methods including structural data in the coil regions and between side chains of amino acids. In addition, the structural similarity of known three-dimensional structure provides the basis for building a model that can provide a good estimate of the unknown structure of a protein sequence. However, the limitation is that poor quality statistical data is also being included in estimating the structure and thus, the

optimization procedure could spend much time producing an unreliable protein structure.

V.3. Methodology

To generate statistical data using comparative modeling, an alignment of each residue in one structure with the equivalent residue in the reference structures is required. Once the target sequence and the reference sequence has been aligned, the coordinates of the atoms in the known structures are used to estimate the expected value of structural data, such as, inter-atomic distances and torsional angles. The target protein structure is estimated using the GNOMAD structure estimation framework via statistical data taken from the known structures. The various steps in the procedure consist of the following steps:

1. Find known protein structures that have at least 70% sequence similarity with the target protein sequence and align the sequences of the target proteins and its closely related known structures.
2. Convert the Protein Data Bank (PDB) file for known structures into a new format that coincide with our new atomic representation.
3. Estimate inter-atomic distances, torsional angles, etc. by performing a statistical distribution on structural data of the same type taken from the reference protein.
4. Use statistical data (*e.g.* bond lengths, bond angles, etc.) to estimate the protein structure.

Each step is accomplished using separate programs and the structural data generated from these programs result in the necessary input files for estimating structures using GNOMAD.

V.3.1. Sequence Alignment

The basic idea for estimating structural data is to collect data of the same type from known reference structures. In comparative modeling, structural data in a specific region is determined by collecting structural data of the same type from the equivalent region in the reference proteins. For example, if we wanted to determine the distance between a pair of atoms we would calculate the distances of the same type in the equivalent region of the known protein structures and use this information to find the expected value of the distance between the pair of atoms.

In order to develop a one-to-one correspondence between the amino acids in primary sequences of the known structures, we use a sequence alignment. The sequence alignment allows us to arrange the primary sequence to identify regions of similarity between sequences. Figure 41 is an example of a segment of sequence alignment for the *E. coli* protein and its closely related family members. The sequence alignment is represented in rows. The first row consists of PDB ID 1ctf for the target protein and its primary sequence alignment. The second row consists of symbols that indicate the degree to which an amino acid is conserved in the reference sequences. For this row, each symbol indicates conserved or partially conserved regions and each blank indicates that the regions are not conserved.

In order to avoid combining our work using inexact structural information with the question of accuracy of sequence alignments, the target sequence and the reference sequences are aligned using the server for HMM-based Protein Structure Prediction, SAM-T99 (Karplus *et al.* 1999). The server gives many details about the similarity of the reference sequence. We use the server to find protein sequences that are high in sequence similarity and for which the three-dimensional structure has already been determined. Although more sophisticated sequence alignment methods can be employed, we are only seeking to use a method that accurately aligns reference sequences and thus, aids in identifying equivalent amino acids in a family of reference structures.

V.3.2. Converting Format of PDB files

Results in Chapter III showed that we could identify a core set of local distances via our new atomic representation that to satisfy torsional information. For the GNOMAD algorithm, the advantage of using this method is that we did not have to introduce a nonlinear term into the objective function to satisfy torsional information. In context of comparative modeling, the use of distances to satisfy torsional information brings about another advantage in estimating protein structure. That is, more reliable distances can be used in the less reliable torsional information. Hence, this allows us to extend the range of higher variance torsional information and still pick up some rather reliable local distances that can satisfy the torsional information.

In light of this advantage, the second step of the modeling procedure is to convert

```

1CTF__ DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE
      **: **: * **: **: ** ***** *      *.:*.****: :** :** **
1YL3_I DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1YL3_J DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1GIY_J DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1GIY_I DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1DD4_B DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1DD4_A DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1DD3_B DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1DD3_A DVVLKSFGQNKIQVIKVVREITGLGLKEAKDLVEKAGSPDAVIKSGVSKEEAEI KKKLE
1RQU_A DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE
1RQU_B DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE
1RQV_A DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE
1RQV_B DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE
1RQS_A DVILKAAGANKVAVIKAVRGATGLGLKEAKDLVESAP---AALKEGVSKDDAEALKKALE

```

Figure 41. A segment of a sequence alignment for the *E. coli* protein and its closely related family members.

all PDB files into files that also include coordinates for all pseudo-atoms. The PDB files for a known protein structures only contain information associated with real atoms, including atom number, atom type, residue type, residue number, x, y, and z coordinates of real atoms. For all closely related sequences whose structure are known, the files for the atomic coordinates are downloaded from the PDB Databank and converted into a format that contain both real atoms and their corresponding psuedoatoms. Each modified file is similar in format to the original file but differs in that there are more atoms because of the inclusion of pseudo-atoms.

V.3.3. Generating Statistical Data

In this subsection, we describe a standard method for generating low-variance distance information including bond lengths, torsional distances, and short-range contact distances given an alignment of a family of protein structures. This work does not offer insight into improving statistical methods for generating data, but does provide means for identifying reliable structural data, which can be used as input data in our algorithm.

In generating structural data, we need to eliminate any known sources of possible outlier or unreliable information. A few known sources include sequence similarity and gaps in the sequences. For sequence similarity, we use a minimum value of 70%. But for gaps in the sequences, however, more work is required. Gaps are inserted between amino acids in the sequence so that identical or similar amino acids are aligned in successive

column. For the 1ctf sequence in Figure 41, two consecutive amino acids are separated by gaps; but in several reference sequences the two equivalent amino acids are separated by a sequence of amino acids.

In light of this observation, we see that it is possible for two atoms to be close in spatial proximity in one reference structure and farther apart in spatial proximity for another reference structure and thus, the information in this region would not be useful because of the outlier distances. This problem can be alleviated by first identifying the possible amino acids that will cause distance outliers in the reference protein and identifying these as gap outliers. If the gaps are consistent with the target sequence gaps, use the information in this region for estimating structural data. Otherwise, do not use the information in estimating structural data.

V.3.3.1. Estimating Distances

A model distance, D , between atoms, A_1 in residue R_1 and A_2 in residue R_2 , is estimated and chosen using the following steps:

1. Find the corresponding atoms and amino acids for each reference protein using the given sequence alignment.
2. If these amino acids do not cause distance outliers due to gaps in sequence, compute the corresponding distances, D' , between the corresponding atoms A_1' in R_1' and A_2' in R_2' for each reference protein.
3. Compute the mean of all the distances computed in the previous step.
4. If the associated range is smaller than a specified threshold, we use the distance as input into GNOMAD.

In general, distances corresponding to bond lengths are usually of low-variance due to strong chemical bonds between consecutive main chain atoms. Alternatively, the reliability of the torsional and short-range contact distances is reflective of the structural similarities of the reference proteins, and thus, tend to have more variation.

V.3.3.2. Estimating Bond and Torsional Angles

Angle information is the second type of information used in GNOMAD algorithm (Chapter III). We generate both bond and angles and torsional angles. Given a sequence alignment, a model angle, θ , is estimated and chosen using the following steps:

1. Find all the equivalent angles, θ' , for each reference protein using the given

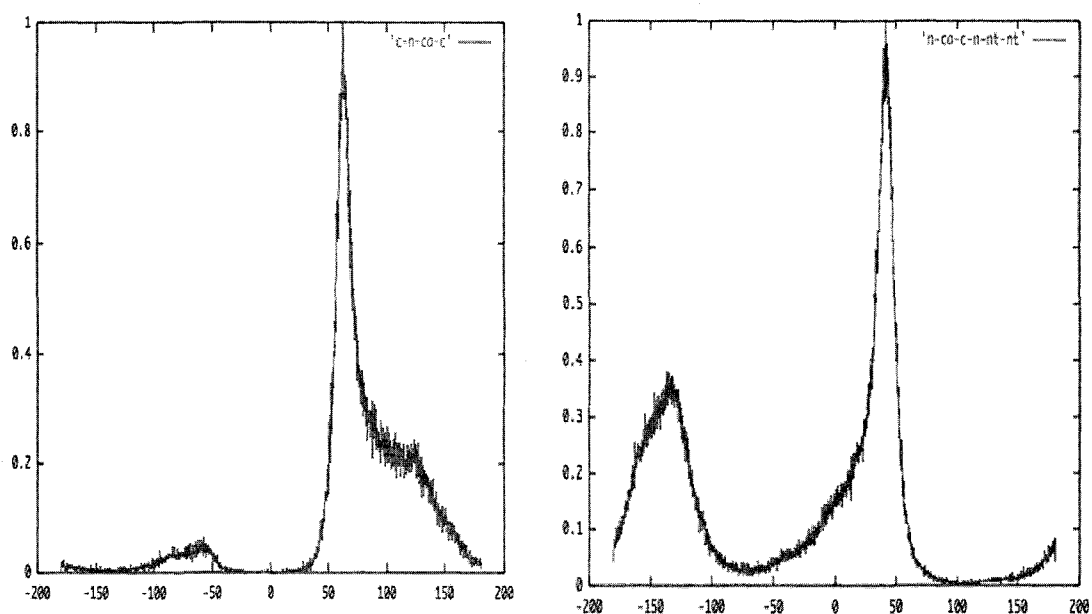


Figure 42. Illustration of ϕ and ψ torsional angle distribution computed from known protein structure from PDBselect data bank PDB files.

sequence alignment

2. Compute the mean angle from all the equivalent angles found in the previous step that are not associated with gap outliers regions.
3. Define the ideal torsional angle values to be the peak values of the torsional angle empirical distribution.
4. If the difference between the mean and the peak value is within one standard deviation of the peak values, we use the angle to generate variational distances as input into GNOMAD.

The bond angles are low variance information and are usually available for all regions of the protein structures. The omega torsional angle is generally close to 180° and is also low variance data.

Alternatively, the preciseness of ϕ and ψ torsional angles depends on the level of structural similarities between the reference proteins. Because of the high level of variation in these angles, we represent these values using an empirical distribution of ϕ and ψ torsional angle generated from the PDBselect database. Figure 42 shows a distribution for each type of angle within a give angle range; say between -200° and 200° .

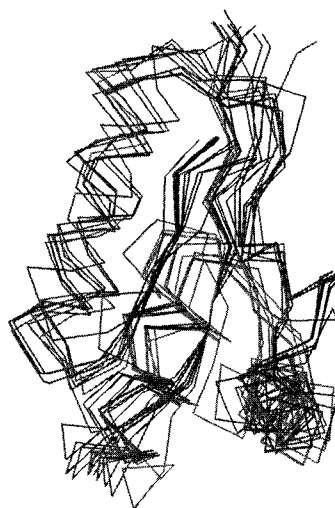


Figure 43. Illustration of high structural similarity of the family members of 1CTF protein.

The distribution is bimodal for both angles and thus we use these two peaks and their associated variance in identifying reliable angles. Given these parameters, we identify reliable angle data by enforcing the criteria that the mean value has to be within one standard deviation of the peak values.

V.4. Validation

Optimization experiments were performed to evaluate the ability of GNOMAD to use structural data taken from reference proteins in estimating the three-dimensional structure of a protein. To evaluate our algorithm, we study the L7/L12 50S ribosomal protein from *E. coli* (PDB entry 1CTF). Given the sequence alignment of 1CTF with its closely related structure (shown in Figure 41), we generate low-variance structural data for estimating the protein structure including bond lengths, bond angles, torsional angles, torsional distances and short-range contact distances. For the 1CTF protein, high sequence similarity translates into high structural similarity. Consider Figure 43, the equivalent segments of the 1CTF and known reference proteins are very close in structural similarity and thus, we can expect a large amount of low-variance structural data for the reference proteins. However, using a large amount of heterogeneous data could lead to over constraining the protein structure. We must find a proper balance in using each type of data to ensure the best quality structure possible.

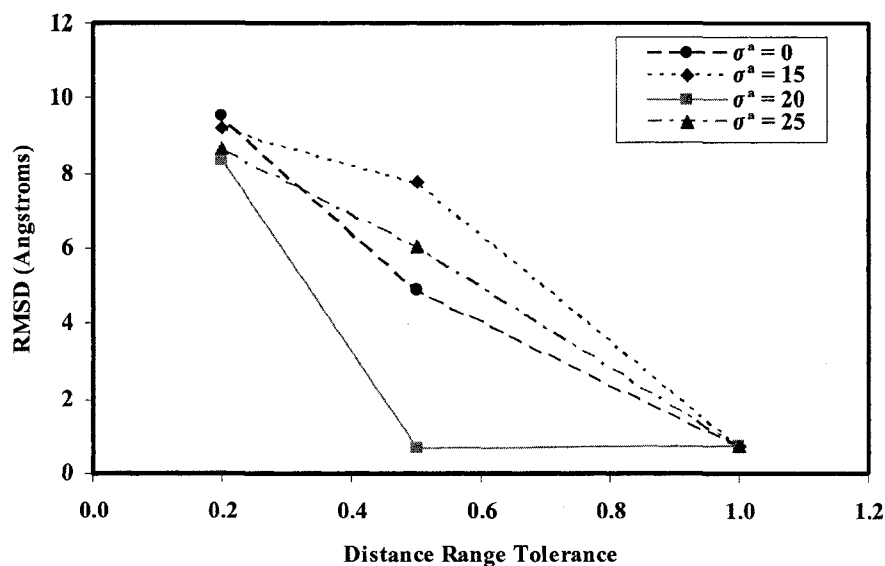
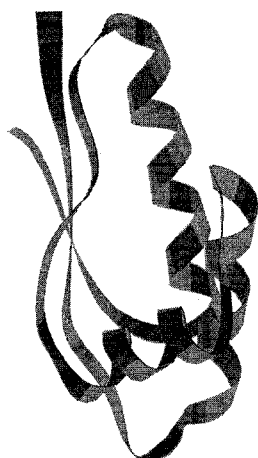


Figure 44. Comparison of torsional angle variance, σ^a , for results over a range of distance range tolerance.

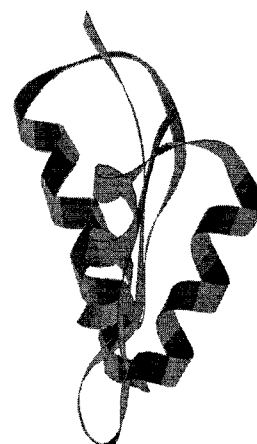
V.4.1. Results

The purpose of this set of experiments is to determine the definition of reliable structural data from reference proteins. We look for the optimal set of tolerance parameters that will yield the best quality structure no matter how much or how little data is available. Figure 44 shows comparison of various torsional angle variances over an interval of distance range tolerance cutoff values, d_c . Different levels of reliable distances and torsional information can have a significant effect on the success of the algorithm in estimating the protein structure. The proper choice of tolerance cutoff parameters would provide a proper balance in using heterogeneous data in GNOMAD.

To illustrate the effects of using a mixture of reliable and unreliable data, Figure 45 shows a qualitative comparison of structures resulting from σ^a values on the interval [15, 25] and their relative proximity to the crystal structure. The results in Figure 45 are based on low-variance distances that are less than 6 Å and whose distance range cutoff value, $d_c = 0.5$ for 1ctf. These plots illustrate the effect of increasing the σ^a values over the specified interval.



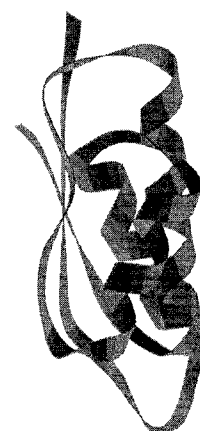
Crystal Structure



Experimental Structure
($\sigma^a = 15$, RMSD = 7.76 Å)



Experimental Structure
($\sigma^a = 20$, RMSD = 0.68 Å)



Experimental Structure
($\sigma^a = 25$, RMSD = 6.06 Å)

Figure 45. Illustrating the effect of decreasing the reliability of torsional information. Crystal structure and structure estimation of 1ctf protein for $d_c = 0.5$ and $\sigma^a = 15$, $\sigma^a = 20$, and $\sigma^a = 25$.

V.4.2. Discussion

In considering the plot in Figure 45, results show that even with low-variance structural data, it is possible to obtain an unfavorable structure. For $d_c = 0.3$, our algorithm is unable to produce good quality structure for all values of σ^a . This suggests that there are not enough distances to provide the necessary amount of folding information to correctly estimate the protein. In slightly increasing the distance range tolerance, we notice that the addition of more distance information increases the possibility of producing a better quality structures. For the case $d_c = 0.5$ and $\sigma^a = 0$, the RMSD is 4.89 Å. This result suggests that low-variance distances, without the inclusion of torsional information, produces an overall improvement in the structure of the protein. However, more structural information is needed to further improve this structure. There are two possible ways for including additional structural data: (1) include low-variance torsional information or (2) relax the distance range tolerance as to include less reliable distances.

In considering the first approach, we investigate the ability of GNOMAD to improve the structure as we vary the torsional angle variance on the interval [10,25]. Immediately, we notice that at $\sigma^a = 20$, RMSD is 0.68 Å. The result is interesting because at this variance value, the torsional angle information is considered unreliable. Because our algorithm allows us to use a core set of distances to satisfy torsional information and distance data tends to be more reliable than angle data, the additional torsional distances are able to effectively satisfy torsional angles. Even so, moving away from this value results in an increase in the RMSD. The reason for this behavior is twofold.

On the lower end of the spectrum (*i.e.* $\sigma^a < 20$), the additional angle information is helpful in improving the fold of the proteins, but this amount of torsional information is not enough to correct the loop regions of the protein structure. On the higher end of the spectrum (*i.e.* $\sigma^a > 20$), more high-variance information results in structural collapse. Figure 45 illustrates the effect of using too little or too much information. In the both cases, the helices are tilted out of alignment with their correct position in the crystal

structure, and thus result in the spatial proximity of the helices being incorrect and an increase in RMSD.

In the considering the second approach, we investigate the behavior of our algorithm as less reliable contact distance information is included as input. In the plot for Figure 45, notice that for $d_c = 1$, the RMSD remains consistently low. An immediate observation is that there is enough distance information available for ensuring that the structure is correctly formed for all values of σ^a . This result is significant because at this value of d_c , the distances are not as uniform but the addition of torsional information does not introduce the problems that we saw in lower values of d_c . Because we are considering the tolerance on the range, the distances at this level could possibly be unreliable.

In our work, we were able to reduce the possibility of generating unreliable data by first requiring that only reference proteins with high structural similarity be used and by eliminating the possibility of obtaining outlier information from gap regions of the sequences. Hence, a distance range tolerance of $d_c = 1$ expectedly results in a good quality structure. But in the case that we cannot eliminate the outlier problems associated with structural similarity or gaps in sequence, choosing this value could result in a structure that is reflective of unreliable data. In light of this observation, the better choice for distance range tolerance is $d_c = 0.5$, because this tolerance suggests that there is higher a possibility for more uniform structural data. At this distance value, a torsional angle variance, $\sigma^a = 20$, produces the best use of both contact distances and torsional information.

V.5. Conclusion

For a family of proteins, the availability of the structural data is proportional to the level of structural similarity of these proteins. Often, much structural data can be gathered and used as input into our algorithm. There are two mainstream approaches in using this data to estimate good quality structures: (1) use all available statistical data and develop more effective computational methods for estimating the structure or (2) use only the most reliable statistical data as input into an existing molecular structure estimator. In this chapter, we consider the latter approach.

Starting with a sequence alignment of a family of proteins, statistical data is generated by using the mean as an estimate of the expected value of a piece of structural data and the associated variance as a measure of reliability on the estimated distances. To avoid using structural data that could have a negative impact of overconstraining the structure, we studied the effectiveness of GNOMAD in estimating structures as we vary the tolerance parameters: distance range tolerance and torsional angle variance. The extension of our problem to comparative modeling shows that the improvements made to GNOMAD are effective still when using a more realistic set of structural data. Choosing the optimal distance and angle tolerance allowed us to find a good balance in using structural information with varying levels of reliability and provided the necessary amount of folding information for obtaining a good estimate of the protein structure.

For this work, however, we only considered the use of distances and angles as input into our algorithm for estimation of protein structure. As we include other types of structural data in GNOMAD, we will be able to consider a more detailed investigation for the problem of extending the GNOMAD structural estimation framework to comparative modeling. In the future, this work can serve as a basis for further development of a comparative modeling package that is based on a constrained global optimization formulation.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In the previous chapters, we have proposed methods for making GNOMAD capable of handling torsional and surface information. In addition, we investigated the effectiveness of GNOMAD in using a more realistic set of structural data generated based on the principals of comparative modeling. At this point our algorithm only uses a small percentage to the structural data available for modeling proteins. In this final chapter, we discuss some future work on improving the GNOMAD molecular structure algorithm. This includes improving the VDW physical constraint enforcement procedure, effective use of contact information to improve the formation of beta-structures regions, and the integration of solvent accessibility information to enhance the use of a surface constraint.

VI.1. Improvement of the VDW Constraint

In future work, we look to improve the VDW constraint so that enforcement of this constraint does not hinder the success of using new structural data types. Currently, the VDW constraint takes into consideration only the minimal set of atoms – N, C_α, C, and C_β – based on the previous atomic representation (as discussed in Chapter II). Hence, the set of "optimal" VDW parameters (minimum separation distances) is constructed between only atoms of these types. In order to completely integrate the VDW constraint with the use of pseudo-atoms for torsional angle satisfaction, we must develop an effective set of VDW parameters that includes the pseudo-atoms.

VI.2. Effective Use of Contact Information in Improving Beta-structures

In Chapter III, we specifically studied satisfaction of torsional information and its consequence in the formation of secondary structure regions. Given that we have an effective method for constructing some of the major secondary structures found in the proteins, we can now examine methods for making GMOMAD capable of handling structural data that provides nonlocal folding information and aid in bringing secondary structures into correct spatial proximity to form beta-structures. The formation of beta-

structures is driven by nonlocal interactions among amino acids that can be far in sequence proximity, but close in spatial proximity (Hue and Dill 1993).

Contact information, is used often as an additional source of structural data in molecular structure estimators for improving the fold of beta-structures because this information describes the nonbonded interaction between amino acids in unconnected secondary structures. The treatment of contact information, in this dissertation, was done in an ad hoc fashion in order to investigate the progression of our surface constraint with the integration of more distance information. In specific, we used short-range contact distances between all atoms and pseudo-atoms in connected and topological neighboring β -strands, and α -helices were used as additional input. The addition of short-range contact distances between all atoms resulted in an improvement of the formation of beta-structures and a reduction in the demands of using only the surface constraint to bring secondary structures together.

In future work, we want to develop more effective methods for improving the formation of beta-structures. This could involve identifying a core set of short-range contact distances and developing a physical constraint, based on contact information, where atoms in one secondary structure are restricted from going into the space of the another secondary structure. The implementation of this type of constraint will help our algorithm to avoid collision between secondary structures, to improve the spatial proximity of secondary structures, and to aid in the formation of beta-structures.

VI.3. Solvent Accessibility Constraint

One of the main avenues for developing protein structure estimation is to use structural data describing the one-dimensional aspects of protein structures. This structural data includes solvent accessibility information, contact numbers, secondary structure predictions, and residue-wise contact order (Chen *et al.* 1996; Kinjo and Nishikawa 2005). One of the goals of future work is to make our algorithm capable of effectively using structural data that describes one-dimensional aspects of protein structure. One type of structural data that will be helpful in using our algorithm to improve structure is solvent accessibility information (Goldman *et al.* 1998).

Solvent accessibility information describes the degree to which amino acids are buried in the core or exposed on the surface of the molecule. The principal idea of solvent

accessibility information is that every amino acid has preferences for certain solvent accessibility states (Richardson and Barlow 1999). Therefore, this one-dimensional structural data can be used to model the hydrophobic effect of protein folding. Furthermore, the combination of solvent accessibility data and surface information would provide further details about the positioning of atoms relative to surface and could be beneficial in improving the accuracy of protein structures.

A number of methods for using surface proximity information have been adapted in modeling proteins to improve the structures when combined with existing structural data (Lee and Richards 1971; Schmidt *et al.* 1998; Chen 2000). However, these methods are difficult to implement in the GNOMAD structural estimation framework. In future work, we want to develop a method that complements both surface shape and solvent accessibility information to aid the optimization of a set of atoms based on knowledge of the degree of burial of the amino acid. However, one important concern to date is whether a good quality estimate can be obtained from flawed surface proximity information.

Much work has been developed to investigate the use of inexact predictions in molecular structure estimation algorithms including Chen *et al.* (2000) and Kinjo and Nishikawa (2005). The results of these methods showed that inexact prediction information produced a marked improvement in the structure of the protein and that the increase in accuracy of prediction methods will result in more accurate structures.

VI.4. Conclusion

In this dissertation, we have investigated using torsional and surface data in GNOMAD. We proposed two methods for integration of this data. First, we developed a new atomic representation that would allow us to use a core set of distances to satisfy torsional information and avoid minimizing a nonlinear objective function. Then, we proposed a method for directly using the molecular surface of protein to move atoms to position that satisfy surface data. The integration of this information has been instrumental in improving the performance of GNOMAD in estimating three-dimensional structures. Finally, we investigate the extension of the GNOMAD structural estimation framework to comparative modeling and found that GNOMAD is effective in a more realistic setting where structural data is generated based on the principal of comparative

modeling.

The improvement to the GNOMAD algorithm discussed in this dissertation may be only for torsional and surface information. However, the improvement represents a significant advance in estimating protein structure using our improved constrained global optimization developed for atomic distances. This work can be extended to include other types of non-distance information and will be pursued in the future.

REFERENCES

- Altman, R.B.: A probabilistic approach to determining biological structure: integrating uncertain data sources. *Inter. J. Human-computer Studies*. **42**, 593-616 (1985)
- Andrec, M., Snyder D.A., Zhou, Z., Young, J., Montelione, G.T., Levy R.M.: A large data set comparison of protein structures determined by crystallography and nmr: statistical test for structural differences and the effect of crystal packing. *Proteins: Structure, Functions, Bioinf.* **69**, 449 – 465 (2007)
- Aszodi, A., Taylor, W.R.: Secondary structure formation in model polypeptide chains. *Protein Engng.* **7**, 633 – 644 (1994)
- Barbosa, H.J.C., Lavor, C., Raupp, F.M.P.: Computational experiments with an adaptive genetic algorithm for global minimization of potential energy functions. In: Floudas, C. A., Pardalos, P.M. (eds.), *Frontiers in Global Optimization*, vol. 74, pp. 71- 82, Kluwer Academic Publisher. New York (2003)
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, Z., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E.: The protein data bank. *Nuc. Acid. Res.* **28**, 235-242 (2000)
- Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. *IEEE Trans. Pat. Anal. Mach. Intell.* **14**, 239 – 256 (1992)
- Brooks, B., Buccoler, R.E., Olafson, B.D., States, D.J. Swaminathan, S., Karplus, M.: CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* **4**, 187 – 217 (1983)
- Brünger, A.T., Adams, P.D., Clore ,G.M., DeLano, W.L. Gros, P., Grosse-Kunstleve, R. W., Jiang, J., Kuszewski, J., Nizges, M., Pannu, N.S. Read, R.J., Rice, L.M., Simonson, T. and Warren, G.L.: *Crystallography & NMR System: A new software suite for macromolecular structure determination.* *Acta. Cryst.* **54**, 905 -921 (1998)
- Cao, J., Pham, D.K., Tonge, L., Nicolau, D.V.: Predicting surface properties of proteins on the Connolly molecular surface. *Smart Mater. Struct.* **11**, 1-6 (2002)
- Chen, C.C., Singh, J.P., Altman, R.B.: Using imperfect secondary structure prediction to improve molecular structure from atomic distances. *Bioinf.* **15**, 53 – 65 (1996)

- Chen, C.C.: Molecular structure computation from multiple data sources. Ph.D. thesis, Stanford University, Palo Alto, CA (2000)
- Crippen, G.M., Havel, T.F.: Distance Geometry and Molecular Conformation. Wiley, New York (1988).
- Connolly, M. L.: Solvent-accessible surfaces of proteins and nucleic acids. *Science*. **221**, 709-713 (1983)
- Cronin, J.R., Pizzarello, S.: Enantiometric excesses in meteoritic amino acids. *Science*, **275**, 951 – 955 (1997)
- Dennis Jr., J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, New Jersey (1983).
- Dewar, M.J.S. Holder, A.J. Dennington II, R.D. Liotard, D.A., Truhlar, D.G. Keith , T.A. and Milliam, J.M.: AMPAC 8 user manual. Sechiam Inc. (2004)
- Dill, K.A.: Perspectives in Biochemistry: Dominant forces in protein folding. *Biochem.* **29**, 7133 – 7155 (1990)
- Dill, K.A., Phillips, A.T., Rosen, J.B.: CGU: an algorithm for molecular structure prediction. In: Biegler, L.T., Coleman, T.H., Conn. A.R., Santosa, F.N. (eds.), Large-scale Optimization with Application Part III, vol. 94, pp. 1-22. Springer-Verlag, New York (1997)
- Dobson, C.M., Sali, A., Karplus, M.: Protein Folding: A perspective from theory and experiment. *Angew. Chem. Int. Ed. Eng.* **37**, 869-893 (1998)
- Dong, Q., Wu, Z.: A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *J. Global Optim.* **26**, 321-333 (2003)
- Dugan, J., Altman, R.: Using surface envelopes for discrimination of molecular models. *Protein Sci.* **13**, 15 -24 (2004)
- Duncan, B.S., Olson, A.J.: Approximation and characterization of molecular models. *Biopoly.* **33**, 219 -299 (1993)
- Eisenhaber, F., Lijnzaad, P., Argos, P., Sander, C., Scharf, M.: The double cubic lattice method: efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *J. Comp. Chem.* **16**, 273 – 284 (1995)
- Erikson, J.: Optimization and regularization of nonlinear least squares problem, Ph.D.

- thesis, Umea University, Sweden (1996)
- Felts, A. K., Gallicchio, E., Wallqvist, A., Levy, R. M.: Distinguishing native conformation of proteins from decoys with an effective free energy estimator based on the OPLS all-atom force field and the surface generalized born solvent model. *Proteins: Structure, Function, and Genetics*. **48**, 404 – 422 (2002)
- Finkelstein, A. V.: Average and extreme multi-atom van der waals interactions: strong coupling of multi-atom van der waals interactions with covalent bonding. *Chem. Cent. J.* **1**, 1 – 9 (2007)
- Floudas, C.A., Pardalos, P.M.: Global optimization approaches in protein folding and peptide docking. In: Farach-Colton, M., Roberts, F.S., Vingron, M., Waterman, M. (eds.) *Mathematical Support of Molecular Biology*, vol. 47, pp. 141 – 171, American Mathematical Society. Providence (1999)
- Friesner, R.A., Gunn, J.R.: Computational studies of protein folding. *Annu. Rev. Biophys. Biomol. Struct.* **25**, 315 – 342 (1996)
- Furnham, N., de Bakker, Paul IW, Core, S., Burke, D.F., Blundell, T.: Comparative modeling by restraint-based conformational sampling. *BMC Struct. Biol.* **8**, 1-15 (2008)
- Gallicchio, E., Levy, R.M.: AGBNP: An analytic implicit solvent model suitable for molecular dynamics simulations and high-resolution modeling. *J. Comput. Chem.* **25**, 479 – 499 (2004)
- Goldman, N, Thorne, J.L., Jones, D.T.: Assessing the impact of secondary structure and solvent accessibility on protein evolution. *Genetics*. **149**, 445-458 (1998)
- Goodfellow, J.M., Moss.: D.S. *Computer Modeling of Biomolecular Processes*. Ellis Horwood, New York (1992).
- Hansman, U.H.E.: Protein folding *in Silico*: An overview. *Computer Simulations*, **3**, 64 – 69 (2003)
- Hendrickson, B.A.: The molecule problem: determining conformation from pairwise distances. Ph. D. thesis. Cornell University, Ithaca, NY (1991)
- Herchlag, D.: RNA chaperones and the RNA folding problem. *J. Bio. Chem.* **270**, 20871 – 20874 (1995)
- Hiroyasu, T., Miki, Ogura, S., Aoi, K., Yoshida, T., Okamoto, Y., Dorgarra, J.: Energy

- minimization of protein tertiary structure by parallel simulated annealing using genetic crossover. *J. Phys. Chem. B.* **102**, 653 -656 (1998)
- Hobohm, U. Scharf M., Schneider, R., Sander, C.: Selection of a representative set of structures from the broken protein data bank, *Protein Sci.* **1**, 409 – 417 (1992)
- Hue, H.S., Dill, K.A.: Origin of structure in globular proteins. *Pro. Natl. Acad. Sci.* **87**, 6388 – 6392 (1990)
- Ilari, A., Savino, C.: Protein structure determination by x-ray crystallography. *Methods Mol. Biol.* **452**, 63 -87 (2008)
- Irvine, G.B., El-Agnaf, O. M., Shankar, G. M., Walsh, D.M.: Protein aggregation in the brain; the molecular basis for alzheimer's and parkinson's diseases. *Mol. Med.* **14**, 451 – 464 (2008)
- Kar P., Wei Y., Hansmann, H.S.: The influence of molecular surface composition on the outcome of poisson boltzman calculations performed to obtain salvation free energies. In: von Neumann, J., Hansmann, U.H.E (eds.), *Computational Biophysics to Systems Biology*, vol 34, pp. 141 – 171, John von Neuman Institute for Computing, Jülich (2006)
- Karplus, K., Barrett, C., Hughey, R.: Hidden markov models for detecting remote protein homologies. *Bioinformatics.* **14**, 846 – 856 (1999)
- Kinjo, A. R., Nishikawa, K.: Recoverable one-dimensional encoding of protein three-dimensional structures. *Bioinf.* **21**, 2167 – 2170 (2005)
- Klepeis, J.L., Floudas, C.A.: Free energy calculations for peptides via deterministic global optimization, *J. Chem. Phy.* **110**, 7491 – 7512 (1999).
- Knight, J.L., Zhou, Z., Gallicchio, E., Himmel, D.M., Friesner, R.A., Arnold, E., Levy R. M.: Exploring structural variability in x-ray crystallographic models using protein local optimization by torsion- angle sampling, **64**, 383 – 396 (2008)
- Laiter, S., Hoffman, D.L. Singh, R.K. Vaisman, I.I, and Tropsha, A.: Pseudotorisonal OCCO backbone angle as a single descriptor of protein secondary structure. *Protein Sci.* **4**, 1633 – 1643 (1995)
- Land, A. H., Doig A.G.: An automatic method for solving discrete programming problems. *Econmentrica.* **28**, 497 – 520 (1960)
- Lee, B., Richards, F.M.: The interpretation of protein structures: estimation of static

- accessibility, *J. Mol. Biol.* **55**, 379 – 400 (1979)
- LeGrand, S.M., Merz, K.J.M.: Rapid Approximation of molecular surface area via the use of Boolean logic and look-up tables. *J. Comp. Chem.* **14**, 349 – 352 (1993)
- LeGuenec, P.: Towards a theory of molecular recognition. *Theor. Chem. Acc.* **101**, 151 – 158 (1971)
- Li, Z., Scheraga, H.A.: Monte carlo-minimization approach to the molecular surface area via the use of boolean logic and look-up tables. *Proc. Nat. Acad. Sci.* **84**, 6611–6615 (1987)
- Lorenson, W.E., Cline, H.E.: Marching cubes: a resolution 3D surface construction algorithm, *Com. Graph.* **21**, 163 – 169 (1987)
- Mehta, M.A., Eddy, M.T., McNeill, S.A., Miles, F.D., Long, J.R.: Determination of peptide backbone torsional angles using double-quantum dipolar recoupling solid-state NMR spectroscopy. *J. Am. Chem. Soc.* **130**, 2202 – 2212 (2008)
- Miller, R.E.: *Optimization: foundations and applications*. Wiley, Canada (2000).
- Monge, A., Friesner, R.A., and Honig, B.: An algorithm to generate low-resolution protein tertiary structures from knowledge of secondary structure. *Biophysics.* **91**, 5027 – 5029 (1994)
- More', J., Wu, Z.: Global continuation for distance geometry problem. *SIAM J. Optim.* **7**, 814 – 836 (1997)
- More', J., Wu, Z.: Distance geometry optimization for protein structures. *J. Global Optim.* **15**, 219 – 234 (1999)
- Mumenthaler, C., Braun, W.: Predicting the helix packing of globular proteins by self-correcting distance geometry. *Protein Sci.* **4**, 863 – 871 (1995)
- Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536 -540 (1995)
- Neumaier, A.: Molecular modeling of proteins and mathematical prediction of protein structure. *SIAM Review*, **3**, 407 – 460 (1997)
- Pearlman, D.A., Case, D.W., Caldwell, J.W., Ross, W.R., Cheatham III, T.E. DeBolt, S., Feguson, D. Seibel, G., Kollman, P.: AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy

- calculations to elucidate the structures and energies of molecules. *Comp. Phy. Comm.* **19**, 1 – 41 (1995)
- Pedretti, A., Villa, L., Vistoli, G.: A versatile program to convert, handle, and visualize molecular structures on windows-based PCs. *J. Mol. Graph. Model.* **21**, 47 – 49 (2002)
- Phillips, A.T., Rosen, J.B., Walke, V.H.: Molecular structure determination by convex global underestimation of local energy minima. *Dimas Series in Discrete Mathematics and Theoretical Computer Science*, **23**, 181 – 198 (1995)
- Pike, R. K.: Optimization for engineering systems. Ph.D. thesis, Louisiana State University, Louisiana (2001)
- Pokala, N. and Handel. T.: Review: Protein design – where we were, where we are, and where we're going. *J. Struct. Biology*, **134**, 269 – 281 (2001)
- Ratchek, H., Rokne, J.: *New Computer Methods for Global Optimization*. Halsted Press, New York (1988).
- Richardson, C.J. Barlow, D.J.: The bottom line for prediction of residue solvent accessibility. *Protein Engng.* **12**, 1051 – 1054 (1999)
- Roder, H.: Stepwise helix formation and chain compaction during protein folding. *PNAS*, **101**, 1793 – 1794 (2004)
- Sali, A., Blundell, T.L.: Comparative modeling by satisfaction of spatial restraints. *J. Mol. Bio.* **234**, 779 – 815 (1993)
- Sanner, M.F., Olson, A.J., Spheer, J.C.: Reduced surface: an efficient way to compute molecular surfaces, *Biopolymers*. **38**, 302 – 305 (1996)
- Scheraga, H.A., Liwo, A., Oldziej, S., Czaplewski, C., Pillardy J., Ripoll, D.R., Vila. J.A., Kazmierkiewicz, R., Saunders J.A., Arnautora, Y.A., Jagielska, A., Chinchio M, Nancias, M.: The protein folding problem: global optimization of the force fields. *Front. Biosci.* **1**, 3296-3323 (2004)
- Schlick, Tamar.: *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer-Verlag, New York (2002).
- Schmidt, J.P., Chen, C.C., Cooper, J.L., Altman, R.B.: A surface measure for probabilistic structural computations. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **6**, 148 – 156 (1998)

- Schwieters, C.D., Kuszewski, J.J., Tjandra, N, Clore, G.M.: The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.* **160**, 65 -70 (2003)
- Torn, A., Zilinskas, A.: *Global Optimization*. Springer-Verlag, Berlin (1989).
- Torshin, I.: Molecular surface sequence analysis of several *E. coli* Enzymes and implication for existence of casein kinase-2 bacterial predecessor, *Frontier in Bioscience.* **4**, 394 -407 (1999)
- van Laarhoven, P.J.M., Aarts, E.H.: *Simulated Annealing: Theory and Applications*. D. Reidel Publishing, Netherlands (1987)
- Vorobev, Y.N.: SIMS: Computation of smooth invariant molecular surfaces. *Biophys. J.* **73**, 722 – 732 (1997)
- Wade, R.C. Gabdouline, R.R.: Analytically defined surfaces to analyze molecular interaction properties. *J. Mol. Graph.* **14**, 341 – 353 (1996)
- Wall, M.E., Subramaniam, S., Phillips Jr., G.N.: Protein structure determination using a database of interatomic distance probabilities. *Protein Sci.* **8**, 2720 – 2727 (1999)
- Williams, G.A., Dugan, J.M., Altman, R. B.: Constrained global optimization for estimating molecular structure from atomic distances. *J. Comp. Bio.* **8**, 523 – 547 (2001)
- Wu, D.: Distance-based protein structure modeling. Ph.D. thesis, Iowa State University, Iowa (2006)
- Wu, D., Wu, Z.: An updated geometric build-up algorithm for solving the molecular distances geometry problem with sparse distance data. *J. Opt.* **37**, 661 – 673 (2007)
- Wu, Z.: The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation. *SIAM J. Optim.* **6**, 748 – 768 (1997)
- Wüthrich, K.: Protein structure determination in solution by NMR spectroscopy. *J. Biol. Chem.*, **265**, 22059 – 22062 (1990)
- Wüthrich, K.: NMR studies of structure and function of biological macromolecules. *J. Biomol. NMR.* **27**, 13 -39 (2003)
- Xue, B., Dor, O., Faraggi, E., Zhou, Y.: Real-value prediction of backbone torsional angles. *Proteins: Structure, Function, and Bioinformatics.* **68**, 76 -81 (2007)
- Zhou, Y., Abagyan, R.: Efficient stochastic global optimization for protein structure prediction. In: Thorpe, M.F. Duxbury, P.M. (eds.) *Rigidity Theory and Applications*,

pp. 345-356. Kluwer Academic/Plenum Publishers, New York (1999)

VITA

Terri Marie Grant
Old Dominion University
Department of Mathematics and Statistics
Norfolk, VA 23529 - 0077

Education

Old Dominion University, Ph.D. Computational & Applied Mathematics, 2008.
Old Dominion University, M.S. Computational & Applied Mathematics, 2003.
Christopher Newport University, B.S. Science, 2001.
Tidewater Community College, A.S. Science, 1999.

Professional Experience

2004-present – Adjunct Instructor, Computational & Applied Mathematics, Old Dominion University, Norfolk, VA.
2006-present – Adjunct Instructor, Tidewater Community College, Suffolk, VA.
2001-2003 – Research Assistant, Computational & Applied Mathematics, Old Dominion University, Norfolk, VA.
2001 – Summer Intern, Booze Allen Hamilton, Norfolk, VA.
1998-2001 – Math Lab Assistant, Tidewater Community College, Suffolk, VA.

Awards and Honors

Old Dominion University Philip R. Wohl Graduate Scholarship Award (Summer, 2008).
Old Dominion University Teaching Assistantship Award (2003- 2004, 2006)
Dominion Dean's Graduate Scholar Fellowship (2001-2003)
Who's Who Among Junior Colleges Award (1999)
Golden Key National Honor Society (1998)
Phi Theta Kappa Honor Society (1998)
National Dean's List Award (1998)
Lions Club Scholarship Award (1998)
Tidewater Community College Provost Leadership Award (1997)