Development of an Analysis and Simulation Package for Protein Molecules & Improving the Graphics of their Conformational Transition

Hima N. Khoshreza
hima.khoshreza@gmail.com

Recommended Citation
http://digitalcommons.uconn.edu/gs_theses/428

This work is brought to you for free and open access by the University of Connecticut Graduate School at DigitalCommons@UConn. It has been accepted for inclusion in Master's Theses by an authorized administrator of DigitalCommons@UConn. For more information, please contact digitalcommons@uconn.edu.
Development of an Analysis and Simulation Package for Protein Molecules & Improving the Graphics of their Conformational Transition

Hima N. Khoshreza

B.S.E., University of Connecticut, 2009
B.A., University of Connecticut, 2011

A Thesis
Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science
at the University of Connecticut
2013
Master of Science Thesis

Development of an Analysis and Simulation Package for Protein Molecules & Improving the Graphics of their Conformational transition

Presented by

Hima N. Khoshreza, B.S.E., B.A.

Co-Major Advisor

Kazem Kazerounian

Co-Major Advisor

Horea T. Ilies

Associate Advisor

Zbigniew M. Bzymek

University of Connecticut

2013
Sincere thanks to my advisors, Dr. Kazem Kazerounian and Dr. Horea Ilies for their understanding, guidance and tremendous patience. This work wouldn’t be possible without their mentorship and my educational experience would not be completed without their genuine support and inexhaustible diligence.

I would like to also express my gratitude to all my friends and colleagues at the Kinematics Design Laboratory, especially Dr. Zahra Shahbazi, Dr. Hami Golbayani and Pouya Tavousi, for their faithful discussions and help. Thanks to all my friends, who have made it possible to follow my heart and have honored me with their friendship, which has helped me to become the person I am today. My sincere appreciation also goes to the Departments of Engineering and Modern Languages at the University of Connecticut for the support provided by all the exceptional faculty and staff that made it possible for me to experience this unique journey.

I would like to especially extend my heartfelt thanks to my best friend and soul mate Bahador Pazoki for his unconditional love and constant support and encouragement throughout each and every step of my academic life and also for lending his creative talent in designing my webpage to display this work. I would also like to thank my beloved parents, to whom I am grateful for nurturing my sense of curiosity, which has led me to choose the wonderful world of science.
### Table of Contents

**ABSTRACT** ................................................................................................................................................ ix

1. **INTRODUCTION** ........................................................................................................................................... 1

2. **PROTEIN PREDICTION** ............................................................................................................................. 4

3. **PROTEIN STRUCTURE** ............................................................................................................................... 5

4. **KINEMATIC MODELING OF PROTEIN** ...................................................................................................... 7

5. **GUI DESIGN & REQUIREMENTS** ............................................................................................................. 9

6. **PROTOFOLD'S GRAPHICAL USER INTERFACE** .................................................................................... 11
   6.1 Installation Procedure .......................................................................................................................... 12
   6.2 What is the Protofold Software? ........................................................................................................... 13
   6.3 Overview ................................................................................................................................................ 15
   6.4 Input & Output (Files) ............................................................................................................................ 18
   6.5 Menu Description ................................................................................................................................... 21
      6.5.1 Edit ............................................................................................................................................... 21
         □ Force field model .............................................................................................................................. 21
         □ Advanced settings ............................................................................................................................. 22
            o Main Chain Structural constants .............................................................................................. 22
            o Side Chain Structural constants ............................................................................................... 24
      6.5.2 File ............................................................................................................................................... 27
         □ New ............................................................................................................................................... 28
         □ Open local PF ................................................................................................................................. 28
         □ Open local PDB .............................................................................................................................. 28
         □ Download from PDB ..................................................................................................................... 28
         □ Save settings .................................................................................................................................. 28
         □ Upload settings ............................................................................................................................... 29
         □ Print ............................................................................................................................................... 29
         □ Exit ............................................................................................................................................... 29
      6.5.3 Solve ............................................................................................................................................. 29
         □ Run ............................................................................................................................................... 29
         □ Joints Torques & Forces ................................................................................................................... 33
      6.5.4 Display ........................................................................................................................................... 33
         □ Body & unit vectors ........................................................................................................................... 34
Table of Figures

Figure 1: Nanoparticle designed by Dr. Burkhard’s team ................................................................. 4
Figure 2: The condensation of two amino acids to form a peptide bond in proteins ....................... 6
Figure 3: Primary structure of protein .................................................................................................. 7
Figure 4: Protein’s kinematic model ...................................................................................................... 8
Figure 5: Rotamer angles of side chain ................................................................................................. 9
Figure 6: Basic operation of ProtoFold (3) ........................................................................................... 16
Figure 7: Welcome window .................................................................................................................. 17
Figure 8: Guide window ....................................................................................................................... 17
Figure 9: Main interface ....................................................................................................................... 18
Figure 10: ProtoFold formatted input file ............................................................................................ 19
Figure 11: PDB formatted input file (18) ............................................................................................. 19
Figure 12: Individual amino acid (Alanine) ......................................................................................... 20
Figure 13: ProtoFold’s setting file ........................................................................................................ 20
Figure 14: Main menu of ProtoFold ..................................................................................................... 21
Figure 15: Menu > Edit ......................................................................................................................... 21
Figure 16: Main chain structural parameter’s dialog ........................................................................... 24
Figure 17: ............................................................................................................................................... 25
Figure 18: Side chains structural parameters (Directory dialog) ........................................................... 26
Figure 19: Side chains structural parameters (Individual amino acid’s dialog) ..................................... 27
Figure 20: Menu > File ......................................................................................................................... 27
Figure 21: Menu > Solve ....................................................................................................................... 31
Figure 22: Run dialog (Folding Flow dialog) ....................................................................................... 31
Figure 23: ............................................................................................................................................... 32
Figure 24: ............................................................................................................................................... 32
Figure 25: Joints torques and forces ..................................................................................................... 33
Figure 26: Menu > Display ................................................................................................................... 34
Figure 27: Display > Body and unit vectors ......................................................................................... 34
Figure 28: Display > Ribbon ............................................................................................................... 35
Figure 29: Display > Hydrogen Bonds ................................................................................................ 36
Figure 30: Display > Show Torque on Joints ....................................................................................... 36
Figure 31: Display > View in Molviewer .............................................................................................. 37
Figure 32: Display > Label ..................................................................................................................... 37
Figure 33: Menu > Graphs .................................................................................................................... 38
Figure 34: ............................................................................................................................................... 39
Figure 35: Menu > Graphs >Show Dihedral Angles ............................................................................. 39
Figure 36: Menu > Hydrogen Bonds .................................................................................................... 41
Figure 37: Rigidity > Apply Rigidity ..................................................................................................... 41
Figure 38: Rigidity analysis result ......................................................................................................... 42
Figure 39: Menu > Help ......................................................................................................................... 43
Figure 40: Menu > Help > Show Folders ............................................................................................. 44
Figure 41: ............................................................................................................................................... 49
List of Tables

Table 1: Main Chain body vectors .............................................................................................................. 23
Table 2: Result of visual optimization by position method ........................................................................ 56
Table 3: Visual optimization by angle method .......................................................................................... 57
ABSTRACT

Proteins are the building blocks of many living organisms and to study them is to discover many of nature’s processes. Abnormalities of these superb molecules make them also accountable for a wide range of human diseases. Every sixty seconds an American will be diagnosed with Alzheimer’s disease. The mystery behind this disease and many others might well be revealed by novel protein prediction techniques. This thesis investigates developments of two tools to benefit such techniques.

First, a user-friendly graphical interface (GUI) was developed based on a previously perfected program named ProtoFold. ProtoFold is a protein simulation software that examines the structural and nano-kinematic aspects of protein molecules. Prediction of Hydrogen bonds, mobility and rigidity analysis, and structural energy optimization are some of the capabilities of this graphical interface. The ongoing developments in the area of protein prediction pursued by a diverse community of researchers make modern graphical interfaces such as ProtoFold essential to exchange of information on this phenomenon. We used GUIDE Toolbox provided by MATLAB to design this interface.

Second, we suggested a technique to improve the process of capturing video of transitioning protein structures, which originally appeared chaotic. We used kinematic inversion, coordinate transformation, and displacement techniques to facilitate this process.
1. INTRODUCTION

Much effort has been devoted in recent years to analyzing matter in the virtual space. Proteins are one of major focuses of these efforts. Their virtual exploration is by far the most economical and the fastest alternative to majority of experiments pursued in the laboratory. In addition, establishing computer models of proteins in recent years has led to a faster screening of drug candidacy, novel treatments and prevention techniques, and a more efficient production of protein nanoparticles. The ability to predict whether a particular protein is a suitable component for designing a nanoparticle is already the target of many pharmaceutical organizations. The power to manipulate the flexibility, size and 3-D conformation of proteins is now the leading element in development of highly effective anticancer protein based nanoparticles (1). Manmade protein cages and artificial viruses have been proposed for chemical drug delivery and therapeutic purposes (2). Providing novel treatments and prevention techniques is the promise of research and development of “in silico” analysis of protein. Therefore it is absolutely crucial to invest in developing tools and techniques that enhances the knowledge surrounding these versatile molecules. This work is the implementation of a protein analysis software package, named ProtoFold, into a user friendly graphical user interface (GUI) and improvements to the captured visuals of protein’s conformational transition. There are two primary methods commonly used in protein prediction software: template-based modeling, TBM (homology modeling and fold recognition), which primarily uses libraries of available templates to match with a target sequence. Free modeling, FM (ab initio), on the other hand uses physical principals
and select optimization methods to determine structure of a protein possessing a minimum state of energy. *ProtoFold* is a comprehensive approach to protein structure prediction and uses kineto-static compliance method, a method of Free modeling, to predict the three dimensional structure of protein chain (3). This package is a collection of thousands of code lines written in MATLAB\(^1\) programming language and has been under continuous progression since 2003.

Today there is a growing demand for information surrounding protein structure to be met for a diverse community of researchers, most of whom do not favor complex scripts or time consuming user interfaces. This is where Web-severs and user friendly interfaces come to the picture to create much faster and efficient analysis techniques. The following highlights some of these efforts:

*Quark* (4), top ranked the CASP9\(^2\) list in FM servers category, is a computer algorithm for ab initio prediction of protein structure. This online server breaks down the quest sequence into shorter fragments, which are taken from available experimental structures and ultimately reassembled into the original length structure using replica-exchange Monte Carlo simulations led by a composite knowledge-based force field (4). This server is free for academic users and does not have an interface and the result is acquired by uploading a target sequence online. This result is sent to the user’s email via a text file containing atom position and also a link to data gathered on the primary, secondary, and tertiary structure of the quest. There are also some 3-D graphics by *JMolViewer*\(^3\) available for visual comparison that are supplied by the same link. It took this server 36 hours to complete a test sequence with 79 amino acids. Quark is mostly

\(^{1}\) © 2012 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

\(^{2}\) The Critical Assessment of Structure Prediction (CASP), predictioncenter.org

\(^{3}\) An open-source Java viewer for chemical structures in 3D. http://www.jmol.org/
geared towards shorter chains with length below 100 residues and it is one of the preferred choices for sequences without known structures.

**Abalone** (5) is a general-purpose molecular dynamics package for simulation of bio-molecules. It’s primarily designed for simulation of protein folding. A free standalone application can be downloaded from its main webpage and there is no need for installation of any third party programing package. Even though the user is given the option to choose between different modeling methods, the emphasis is on molecular mechanics. Other options include parallel dynamics or replica exchange method and Monte Carlo. The makers of this software claim that they focus on the accuracy of all the above methods the same way; however, this accuracy declines as the number of residues increases. A test case with 14 residues took 31 hours to complete on an average personal computer. The larger the modeled chain is, the longer it takes to reach the final structure. One appealing aspect of this interface was the capability of making arbitrary chains out of elective amino acids right from the interface.

**ProtoFold** too is a promising tool for protein prediction and analysis. The ProtoFold research group has been collaborating with other departments at the University of Connecticut since this software was initiated. Dr. Burkhard, associate professor at Molecular Cell Biology Department in University of Connecticut has been working with his team on developing peptide nanoparticles shown in Figure 1, for which protein links with a specific flexibility is needed to fabricate stable nanoparticles (6). Experimental determination of the perfect amino acid composition for the link segment would be expensive and time consuming. Hence, the ProtoFold team was asked to assess multiple candidate protein molecules to investigate which one would ensure a more stable particle. Such multidisciplinary collaborations necessitate a user friendly format of ProtoFold to be available to researchers with different expertise. Subsequently, they
can apply ProtoFold’s mechanical analysis to their specific application without having to learn tedious programming skills. Therefore it was decided to develop a user friendly graphical interface for ProtoFold. The visual experience of protein folding was also enhanced, which will be explained in details later.

![Nanoparticle designed by Dr. Burkhard’s team](image)

Figure 1: Nanoparticle designed by Dr. Burkhard’s team

2. PROTEIN PREDICTION

Proliferation in protein structure studies has been tremendously accelerated in recent years. Today there are various software and servers available to forecast 3-D structures of proteins. The majority of them work on either of the following premises:

- homology modeling and fold recognition (template-based modeling), or
- ab initio structure prediction (free modeling).

Depending on the specific application, sometimes a combinatory method of the above mentioned is more appropriate. In template-based modeling (TBM) identifying and refining the closest template to the target sequence is critical. Specifically among similar homologous, discerning

---

4 More information on the particle can be found in professor Burkhard’s website: http://www.ims.uconn.edu/~pburkhard/research.htm
closely related structures is a difficult task. On the other hand the challenge in free modeling (FM), where there is a lack of known templates, is to model proteins that reach correct topology (7). The common ground for free modeling uses various optimization methods combined with physical principals to reach a structure with minimum state of energy. Some major approaches to achieve this goal include: minimization of Gibbs free energy and using kinetics or dynamics of protein chain movement (3). Molecular dynamics is minimizing forces applied on each atom by integrating the equation of motion through time. Despite reducing the computational cost of this method by keeping some variables such as bond length and angles constant, the drawback of using molecular dynamics still persists. Complexity of including all aspects of protein interactions in the calculation of force field depresses the popularity of this method. The inherent imprecisions in existing force field models, domination of inter-atomic forces over inertia forces in the motion dynamics of a protein chain, and everlasting quest for computational efficiency has led our team to develop a software package named ProtoFold that uses a successive kineto-static compliance method to predict protein conformation (3). This method uses one of the most widely accepted force field models for protein molecules as the case in many other molecular dynamic packages: AMBER force field (8). Successive kineto-static fold compliance approach is prediction of protein confirmation under the influence of interatomic force field without using molecular dynamics simulation (3). This package models proteins kinematically as explained in two following sections.

3. PROTEIN STRUCTURE

Proteins are large organic compounds, whose primary structures are defined by the exact sequence of amino acids comprising them. Proteins are created when amino acids bond together
by peptide bonds, which are formed between the carboxyl group of one amino acid and the amino group of the next amino acid. There are 20 different amino acids found in proteins which when lined up in a certain sequence, represent individual proteins (9). All amino acids have an identical structure and an altering side chain. The common structure is made out of a central carbon atom with a nitrogen atom covalently bonded to it and also covalently bonded to this same central carbon atom is another carbon atom (N-C-C). As shown in Figure 2, when two amino acids come together, a water molecule gets eliminated and a peptide bond forms between the two. Meanwhile the carboxyl group and the side chain stay disintegrated. This process repeats until a chain of amino acids known as protein’s primary structure assembles in the denatured state (Figure 3).

When exposed to suitable conditions (solvent, PH, and temperature), these strings of amino acids fold into the unique 3D confirmation. The challenge yet remains how to improve proficiency in prediction of folded proteins without conducting the costly experiments.

Figure 2: The condensation of two amino acids to form a peptide bond in proteins
4. KINEMATIC MODELING OF PROTEIN

The formation of successful peptide bonds generates the backbone of the main chain of protein. Figure 4 represents a ball and stick model of a peptide chain, whose backbone repeats an N-Cα-C arrangement. Protein structure resembles a kinematic chain of miniature rigid bodies connected by revolute joints. There are three rotation angles within the main chain of each amino acid. One is around the C-N bond (called ω), one is around the N-Cα bond (called φ), and one is around the Cα–C bond (called ψ). Since the rotation around the C-N bond is very close to 0 or 180 degrees, the Cα–C-N-Cα chain is always in one plane, called a peptide plane. All of the atoms in a peptide plane are fixed in respect to that plane. Bond lengths and angles remain constant in this plane all the way along the chain of amino acids. This plane could be seen as a rigid group of atoms linked into a chain by covalent bonds at the alpha carbons. As a result the degree of freedom (DOF) of the system dramatically reduces and so does the computational complexity. Thus, kinematical modeling of proteins is an economical way of structural protein prediction.
Zero-Position Analysis (10), a robotic method in kinematic studies of open chains, has been proven to fulfill our goals in kinematic studies of proteins (11). The convenient notation in comparison to other popular techniques such as Denavit-Hartenberg (12) has made this method the preferred candidate for our means. Although this method is exhaustively discussed in References (3) and (11), an overview is provided here for convenience.

A serial linkage with \( n + 1 \) solid links (red vectors in Figure 4) connected by \( n \) revolute joints (blue vectors in Figure 4) is assigned to the appropriate bonds and dihedral angles in the backbone of the chain. All vectors of joint axis directions and their locations are defined conveniently in a base coordinate system, which is referred to as the zero-position (13). The zero-position configuration used here is the same as the conventional biological reference position of the chain, in which all peptide planes lie flat in one plane and as a result the main chain’s dihedral angles (\( \varphi, \psi \)) are zero. Side chains are only assigned up to a maximum of 4 DOF.
with a zero DOF indicating that the side chain moves with the main chain. In side chains unit vectors (Rotamer angle axis: $\chi_1 \cdots \chi_4$) and body vectors connecting these unit vectors are similarly allocated (Figure 5).

![Figure 5: Rotamer angles of side chain](image)

5. GUI DESIGN & REQUIREMENTS

Graphical user interfaces (GUIs) were present in our lives before computers invaded our modern routine. Unlike text-based interfaces and typed commands, GUIs are presentation of information and actions by means of graphical icons and visual displays (14). They allow user interaction with a variety of household appliances, portable media players, gaming devices, office equipment, and computer software. They are the medium between users and producers! With today’s rapid technological advances, user interfaces have become integrated in all aspects of our daily lives. Particularly researchers are benefiting from this growing trend. The appreciation for saved time on dealing with behind-the-scenes programing is remarkable among the research community. From our perspective, ProtoFold had to be made available to accelerate the work of researchers like Dr. Burkhard and also assist our research team in advancing our findings.

Since ProtoFold is written in Matlab and Matlab offers high-level script based programming, the GUI was chosen to be developed in Matlab environment. Almost all programming languages
have the capability of developing a graphical user interface to some level. Unlike conventional languages such as C, Matlab scripts do not necessitate compilation, meticulous initiation of variables, or even low-level memory management (15). Consequently the code development process will be much faster and less complicated. High-level script based languages allow developers without intensive programming backgrounds to concentrate on solving their specific problem rather than spending time on constructing a GUI application in a low-level language. GUI development in Matlab enables developers to enjoy the same existing Matlab computational power in their own GUI code (15).

There are two methods available to develop a GUI program in Matlab: 1) Low-level M-file coding or 2) High-level graphical layout. In low-level M-file coding, low-level commands are used to create a GUI. Every single aspect of the interface has to be programmed by UIControl and UIMenu Objects in Matlab from M-file code alone. In fact, this type of programming is called low-level GUI development because it does not rely on any additional tools such as GUIDE, which is used in high-level GUI development (15). GUIDE (Graphical User Interface Development Environment) provides GUI developers with all the built-in GUI functions that are included in Matlab and are somewhat similar to other GUI development environments in other languages.

Using high-level GUI development for complex GUIs has some major advantages, such as simplicity of layout. Complex GUI development is a difficult task in low-level programming since the graphical layout of all the components has to be arranged by specifying coordinates numerically in the code. Overall, much greater work is needed to develop a complex GUI programmatically through low-level M.file coding. GUIDE offers a graphical figure layout which can be populated by clicking and dragging GUI elements off of the component palette.
containing graphical button interfaces. GUIDE also features Menu Editor, through which the menu options can be easily designed. Property inspector can also retrieve properties of different GUI objects. GUIDE also generates automatic M.file which saves a lot of time. High-level GUI development was chosen to generate ProtoFold’s GUI because of the above mentioned reasons.

Since majority of ProtoFold’s programing was developed without a user interface in mind, some major changes had to be applied to various routines to integrate ProtoFold into a user interface. For example, different tasks had to be redefined and reorganized into a consecutive plan of action, in order to reach program’s outputs in multiple stages of the execution. The visual experience of protein folding was also enhanced, which is explained in greater details later. The following is ProtoFold’s user guide:

6. PROTOFOLD’S GRAPHICAL USER INTERFACE
6.1 Installation Procedure

We have made Windows standalone executable files available for both 32 and 64 bit machines. These files are included in a package downloadable from ProtoFold web page. Depending on the availability of MATLAB software on the target machine, you are provided with the option to install MATLAB Compiler Runtime (MCR), which is intended for running the GUI as a stand-alone application outside MATLAB environment.

To install ProtoFold version 1.1:

1. Go to www.himakhoshreza.com and click on “Download ProtoFold” link
2. Uncompress “ProtoFold_gui” folder and run “ProtoFold_pkg.exe”. This will inflate three more files:

- MCRInstaller.exe
- ProtoFold_gui.exe
- readme.txt

If MATLAB is already installed on the target machine proceed to step 3, else install the MATLAB Compiler Runtime (MCR) first. MCR installation should initiate automatically as you run the “ProtoFold_pkg.exe”; if it does not, initiate install it manually by double click on “MCRInstaller.exe”. Specify the directory you wish to install the libraries in, or the installer will automatically choose the default directory. Make note of this directory, as you may need this information when running ProtoFold. To uninstall ProtoFold_gui from your system at any time, simply delete this directory.

3. Double click on “ProtoFold_gui.exe” to run ProtoFold gui

6.2 What is the Protofold Software?

Welcome to the ProtoFold, a protein analysis and simulation package. ProtoFold is designed to predict final conformation of small proteins from denatured configuration and run mobility related analysis on them. Structure based drug design approaches are becoming progressively the reality for treatment of diseases like cancer and Alzheimer’s. The knowledge behind large scaled robots is the driving force for ProtoFold. This knowledge can contribute to the emerging innovative approaches of nanostructured drug design to fight such diseases. We are proud that you chose us to push the limits further and thank you for letting us walk with you on the edge of
science. ProtoFold was initiated in 2006 and funded partially by the National Science Foundation and the University of Connecticut. This package is a collaborative effort of several graduate students, under the supervision of Dr. Kazem Kazerounian and Dr. Horea Ilies.

ProtoFold uses a successive kineto-static fold compliance method (3), in which the conformational changes of the peptide chain are determined by an inter-atomic force field without engaging in Molecular Dynamic Simulation. The chain acts in accordance to the kineto-static effect of the force field, such that each rotatable joint changes by an amount proportional to the effective torque on that joint. Successive iteration of this process continues until all torques imposed on joints have converged to zero and the resultant structure finds itself in minimum state of potential energy. This method has been proven to be superior to the conventional Molecular Dynamics (16). The main steps in successive kineto-static fold compliance are (17):
• Calculate the Cartesian coordinates of all atoms in the protein molecule at a given set of joint angles

• Compute all of the inter-atomic forces in this conformation (using the AMBER force field model discussed in (17))

• Determine the equivalent joint torques using the proportionality between the end effector forces and the joint forces in robotics

• Calculate all corresponding changes to each joint variable, proportional to that joint’s equivalent torque and rotate each joint in accordance

• Repeat step 1 through 4 until all of the joint equivalent torques have converged to zero (within some small prescribed error). Figure 6 shows an overview of basics in ProtoFold.

Next is the secondary tasks performed by ProtoFold:

1. Geometrical determination of Hydrogen bonds

2. Mobility analysis

6.3 Overview

When starting ProtoFold, users are prompted with the following windows:

1. Welcome window: contains brief copyright information

2. Guide window : helps users familiarize themselves with the program by referring them to the appropriate section of ProtoFold’s website/ documentations

3. Main GUI window
Figure 6: Basic operation of ProtoFold (3)

1. Read Sequence of amino acids and \( \varphi, \psi \) (dihedral angles)
2. Main chain/ Side chain body vector creation
3. Change angles from biology notation to kinematic notation (Zero Position Notation)
4. Direct kinematics
   - Determine position of all atoms
5. Apply Amber force field
6. Calculate \( \Delta \theta \)
   - \( \Delta \theta = k[\tau/\tau_{\text{max}}] \) (3)
   - \( \theta_{\text{new}} = \theta_{\text{old}} + \Delta \theta \)
7. Calculate Energy
   - \( E_{\text{new}} \leq E_{\text{old}} \) ?
8. Outputs:
   - Forces, Torques, Hydrogen Bonds, Mobility analysis
9. End
Figure 7: Welcome window

Figure 8: Guide window
6.4 Input & Output (Files)

This part is a guide to files users encounter in ProtoFold package. There are input files containing protein data and files intended for setting the program’s parameters, each of which is stored in separate folders, built by ProtoFold’s executable at initiation time. Input protein files consist of:

**proteinx.pf**: text file containing 3 columns: 1. amino acid abbreviation in the one letter format, 2. phi dihedral angles (in degrees), 3. psi dihedral angles (in degrees)
**Figure 10: ProtoFold formatted input file**

**proteinx.pdb**: standard Protein Data Bank (PDB) formatted file, which contains information on coordinates, amino acids comprising the protein, and other related data.

**Figure 11: PDB formatted input file (18)**

**sample_setting.txt**: txt file enclosing information on program’s structural constants. This file contains data on Main chain and side chain structural parameters. The first part is orientation of the main chain’s body vectors, and the second part is the side chain’s structural constants, which are the positions of all atoms in individual amino acids relative to Nitrogen atom on their main chain (Figure 12). User can set all...
these constants manually or through *Edit* option in the interface, where all the adjustable parameters are visualized and briefly described.

Figure 12: Individual amino acid (Alanine)

Figure 13: ProtoFold’s setting file
6.5 Menu Description

Figure 14: Main menu of ProtoFold

6.5.1 Edit

- Force field model

This option is not active and is reserved for implementation of additional force field model alternatives. In version 1.1 the force field model used in ProtoFold is Amber force field (19), (17). This force field model was developed by Weiner et al. for proteins and nucleic acids and is
further improved by Cornell et al. Equation (1) provides the mathematical description of the total potential energy (3):

$$E_{\text{total}} = E_{\text{bond-length}} + E_{\text{bond-angle}} + E_{\text{torsion}} + E_{\text{van-der-walls}} + E_{\text{electrostatic}}$$  \hspace{1cm} (1)

- $E_{\text{bond-length}}$: Bond energy calculated by Hook’s Law
- $E_{\text{bond-angle}}$: Bond angle (bending) energy calculated by Hook’s Law
- $E_{\text{torsion}}$: Function of the torsion angle (the rotation angle around the bond btw the second and third atom in any serially connected four atoms)
- $E_{\text{van-der-walls}}$: The energy due to nonbonded forces between two atoms
- $E_{\text{electrostatic}}$: Electrostatic energy between two atoms

### Advanced settings

This option provides the opportunity to change user modifiable parameters that construct protein structures. If the user decides to change the default setting, she/ he has to be absolutely confident in their inputs, or else this might lead to errors and inaccuracies. This option furthermore divides into two sub menus:

- **Main Chain Structural constants**

  This is an option to modify the main chain’s structural parameters: orientation of main chain’s body vectors on odd, even, and end peptide planes. The input orientations are in Zero position (11) and are relative to a coordinate system assembled on the first Nitrogen atom of the back bone as the origin (0,0,0). As mentioned before, zero-position configuration is the same as the conventional biological reference position of the chain, in which all peptide planes are coplanar.
For further clarifications please refer to reference (11). In ProtoFold we would need the orientation of six body vectors to geometrically define the entire main chain comprising of all peptide planes in zero-position. Since alternative peptide planes repeat themselves all along the backbone, we need the orientation of two body vectors, \((b_1, b_2)\) and also the end body vector \((b_{1, end})\) in both odd and even planes. The default values for the six main chain body vectors are calculated using the magnitude of bond lengths and the angle between them. The default orientation of main chain body vectors are given in Table 1.

<table>
<thead>
<tr>
<th>Body Vector for Main Chain</th>
<th>Even Peptide Plane</th>
<th>Odd Peptide Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nitrogen to Alpha carbon</td>
<td>[0.69, 1.30, 0]</td>
<td>[1.47, 0, 0]</td>
</tr>
<tr>
<td>Alpha carbon to Nitrogen</td>
<td>[1.81, 1.57, 0]</td>
<td>[2.23, 0.87, 0]</td>
</tr>
<tr>
<td>Alpha carbon to Hydrogen in Carboxyl Group (end residue)</td>
<td>[3.12, 1.69, 0]</td>
<td>[2.95, 1.98, 0]</td>
</tr>
</tbody>
</table>
This option provides a directory of all 20 amino acids, giving one the ability to select individual amino acids for further manipulation (Figure 18). When any of the amino acids are chosen, a dialog box enclosing structural parameters of that specific amino acid’s side chain appears (Figure 19). Only atom positions are editable and side chain’s Rotamer angles ($\chi_1, \chi_2, \chi_3, \chi_4$), body vectors ($b_1, b_2, b_3, b_4$), and unit vectors ($u_1, u_2, u_3, u_4$) are computed as input positions change. The following maps the aforementioned parameters to the appropriate section on this dialog box (Figure 19):
1. Name and three letter format of amino acid, number of degrees of freedom of the side chain (This is also indicated by blue arrows superimposed on the graphics in the dialogs)

2. Atom positions with origin on Nitrogen (indicated by $N(1)$ in blue)

3. Rotamer angles of side chain in radian (each amino acid can have up to four degrees of freedom)

4. Orientation of side chain’s body vectors (shown in blue arrows)

5. Orientation of side chain’s unit vectors (vectors around which body vectors turn), first unit vector is located at the beginning of first body vector and so on

Note that the side chain’s body vectors and unit vectors assume the same direction. In both of the directory and individual amino acid dialog boxes, the default values can be restored by selecting the “Default” or “Reset” button. Unless a legitimate research supports the new input positions, changing the default values is not recommended. When default positions are changed, users are warned before closing these dialogs and if selected to sustain changes, users are asked to save their new settings in a text file (Figure 13).
Figure 18: Side chains structural parameters (Directory dialog)
6.5.2 **File**

Figure 19: Side chains structural parameters (Individual amino acid’s dialog)

Figure 20: Menu > File
- **New**
  Deletes the current protein and refreshes the state of the program to the default for uploading a new file. Note that this won’t restore the values of the program’s structural parameters to the default.

- **Open local PF**
  Gives a selection of locally available input files in ProtoFold format (*.pf) to choose from.

- **Open local PDB**
  Gives a selection of locally available input files in PDB format (*.pdb) to choose from.

- **Download from PDB**
  Provides a dialog for downloading PDB formatted files directly from the Protein Data Bank’s website. This is facilitated by inputting the Pdb ID of a target protein which is a combination of 3-6 alphanumeric characters (ex. 1mp6). The option to save the downloaded protein in ProtoFold format is also provided. Observe that internet connection is necessary for downloading a PDB file from Protein Data Bank’s website.

- **Save settings**
  Saves user determined settings for main chain’s and side chain’s structural parameters in a text file. The option to retrieve a certain setting is also provided in dialogs called in *Edit > Main chain/ side chain structural parameters* and under *File > Upload settings*
- **Upload settings**

  Provides a dialog to upload a custom set of program’s structural parameters

- **Print**

  Prints the current graphics in the main window

- **Exit**

  Quits the program and closes the main window and all associated windows. It also restores all the default values for the program’s structural settings

### 6.5.3 Solve

- **Run**

  Opens a dialog to set program flow parameters (Figure 22):

  There are three parameters determining the conditions, under which ProtoFold solves for protein confirmation with minimum energy. This structural determination is achieved all at once (Figure 22>5) or in steps (Figure 22>6).

  1. **Sensitivity factor “k”** (Figure 22>1):

     - $k$ is a threshold for a small number used to achieve numerical stability in the simulation.
     - It is a proportionality factor used in the successive kineto static fold compliance method.
     - It’s a representation of the correlation between the motion (rotation) at the joints and the equivalent torques imposed on them (3). Extreme caution is advised in regards to changing this value. For more information refer to (3).
2. **Percentage of side chain retraction** (Figure 22>2)

   This determines the amount of spatial retraction for side chains while running the program. It reduces the length of side chain body vectors as main chain’s extrusions from a value of 0% reduction (normal length of side chains) to 100% (no side chains). The closer this value is to 100 the less interaction between side chains develops during the process. The default value is 0.

3. **Step number to pause the execution** (Figure 22>3)

   This determines the number of steps at which simulation pauses to enable the user to check program’s outputs in the middle of the structure transformation.

4. **Default values of parameters** (Figure 22>4)

   Resets all parameters to their default values.

5. **Run step by step** (Figure 22>5)

   Choose when execution of program in multiple stages is desired. Set the number of steps to pause on in (Figure 22>3). The user can check the program’s outputs while execution is paused.

6. **Run** (Figure 22>6)
Choose when uninterrupted execution of the program is desired until the protein structure with minimum energy is achieved.

![Figure 21: Menu > Solve](image1)

Followed by the user’s selection of “Run” (Figure 22>6) or “Run Step by Step” (Figure 22>5), the potential energy axis on the left corner of the main window shows the progress by displaying the energy level of the structure at the most current state (Figure 23). The default scale on the
energy axes is logarithmic. However, if the graphic representing energy fluctuation stays beyond distinction, it can be changed to a linear scale (Figure 24). If “Run Step by Step” is selected, all ProtoFold’s visual as well as data outputs such as forces and torques on joints, Hydrogen bond detection, and rigidity can be redeemed once the simulation pauses.

Figure 23

Energy levels in logarithmic scale

Figure 24

Energy levels in linear scale
- **Joints Torques & Forces**

Displays torques and forces exerted on each joint in a tabulated format, in which force and torque amounts are displayed with the associated joint number as well as the indication of joint’s positions on the main chain or side chain (Figure 25)

![Joints Torques & Forces](image)

**Figure 25: Joints torques and forces**

6.5.4 **Display**

This option offers multiple options for displaying the protein structure and its relevant graphics.
- **Body & unit vectors**

  This option displays the protein molecules in the main chain’s body vectors (white) and unit vectors (magenta) used in our kinematic notation (Figure 4, Figure 27).

- **Ribbon**

  This option switches back the graphic to the ribbon representation of the protein, which is the default representation.
Figure 28: Display > Ribbon

- **Hydrogen Bonds**

This option remains disabled until Hydrogen bonds have been detected under “Hydrogen Bonds > find H bonds”. This option draws Hydrogen bonds in multiple colors, each representing a certain type as indicated in Figure 29. The colors also correspond to colors of shortcut buttons on the toolbar menu.
Figure 29: Display > Hydrogen Bonds

- **Show Torque on Joints**

This option displays torque exerted on each joint in shape of cones. The intensity of the torques is represented in a color bar and the size of the cones. Direction of the cones also denotes torque direction (20).

Figure 30: Display > Show Torque on Joints
- **View in Molviewer**

Displays the input file in *Molviewer*, a three-dimensional molecular visualizer and inspector (21; 22), which is a part of “Bioinformatics Toolbox” in MATLAB. This option is meant for further geometric inspection of the protein.

![Figure 31: Display > View in Molviewer](image)

- **Label**

Prints the one letter formatted amino acid abbreviations on the corresponding amino acids (Figure 32). To delete the labels from the graphics just click on the same option to uncheck it.

![Figure 32: Display > Label](image)
6.5.5 Graphs

Options under Graphs offer to show various outputs on graphs. Forces or torques exerted on all joints or only main chain joints are displayed against their respective joint. The program has to run before this information becomes available. When the option to display all joints is chosen, blue data points are main chain joints and the red ones portray side chain joints. Lines connecting them represent no significance other than following the changes.

- MC Joints # vs. Force
- MC Joints # vs. Torque
- All Joints # vs. Force
- All Joints # vs. Torque

Figure 33: Menu > Graphs
Show Dihedral Angles

Reports main chain’s dihedral angles (Phi, Psi in degree) and side chain’s dihedral angels (Chi1, Chi2, Chi3, Chi4 in radian).
6.5.6 Hydrogen Bonds

Under this option, the geometrical criteria to find Hydrogen bonds as well as their category can be preset. There are six different geometrical criteria that ProtoFold can use to find Hydrogen bonds:

- ProtoFold criteria
- Baker
- Mc Donald
- Dong Xu
- Kotemme
- Fleming

Each of these criteria are extensively discussed in (23). The decision to choose main chain or side chains as either sides of the bond can be set by checking or unchecking the appropriate option (Figure 36-a):

- Main Chain – Main Chain
- Main Chain – Side Chain
- Side Chain – Side Chain

The colors of options match perspective bond representation on the graphics as well as their respective toolbar buttons (Figure 36-b,c).
6.5.7 Rigidity

This option will allow the user to “Apply Rigidity” (Figure 37) to the target protein. It presents the result of rigidity analysis in a dialog box indicated in Figure 38.
Apply Rigidity

Identification of Hydrogen bonds enables the rigidity analysis of protein in ProtoFold. Hydrogen bonds can connect different parts of our previously defined open loop kinematic chain of the protein and make many closed loops within the original chain. Some of these loops can become rigid due to recurring Hydrogen bonds. ProtoFold detects these closed sub-chains by running a preliminary analysis on the topological information of the chain in the form of a connectivity matrix, in which all the links connected to only a single link from either sides of the link dissolves into multiple equivalent links. After this preliminary analysis, rigid loops are determined by employing the Grübler–Kutzbach criterion. Through an iterative process rigid
loops are replaced with equivalent rigid links again (23). The two graphs on the left side of the Rigidity analysis window show the first and last states of the connectivity matrices to make following the progress of link elimination easier (Figure 38- 1, 2). The red nodes represent equivalent links in the each step, while the black connecting lines between them represent whether or not they are joined. The numbers on nodes refer to original link numbers and are just guides for ProtoFold team. The degree of freedom of the structure and the number of links at the initial and final stages are noted beneath each graphs. Figure 38- 3 shows the reduced link representation of the protein, in which solid color portions are links made out of more than two original links and the dotted lines are links made out of a single original link. The sample structure shown, 1w5g, has been reduced to three links (DOF=2): two single original links on both ends and an equivalent large link in the middle of the structure.

Please refer to (23) for detailed guidelines on how to determine the closed loops within the structure and reach a final representation of the protein with far fewer links and joints.

6.5.8 Help

Figure 39: Menu > Help
- **About**

Leads you to the program’s splash screen that contains basic information about ProtoFold group and its webpage.

- **Show Folders**

Using this option, you can reach all papers and documentations on ProtoFold as well as the path on your computer to different folders containing input and setting files. Due to the nature of making an executable file in Matlab, it might be time consuming to manually locate folders inside a package. The path to these three folders can be accessed by choosing the appropriate option in “Help > Show Folders”(Figure 40) :

- PDBfolder : contains PDB formatted input files (*.pdb)
- PF_open    : contains ProtoFold formatted input files (*.pf)
- setting    : contains program settings files (*.txt)

![Find Folder Path](image)

*Figure 40: Menu > Help > Show Folders*
6.6 General Remarks

This version of ProtoFold has been developed in MATLAB by several coders with different coding styles, majority of whom did not have a formal training in programing; hence the program does not carry a consistent programing style throughout. In addition, there wasn’t any intention of creating an interface for ProtoFold during its developmental phase. These made it challenging to integrate the existing structure of the program into the GUI structure. Multiple functions had to be rewritten and the order of their usage had to be reviewed. Since ProtoFold had to be available in a standalone executable application, a few of the previously integrated options had to be omitted from the deployed version of ProtoFold due to the restrictions imposed by “Deployment toolbox” of MATLAB, by which we made the executable version of ProtoFold 1.01. Some aspects of MATLAB toolboxes aren’t deployable such as the Molviewer, a three-dimensional molecular visualizer and inspector (19; 20), which is a part of “Bioinformatics Toolbox”. Although ProtoFold is a state of the art tool to predict the final conformation of proteins, like any other protein prediction software it has some limitations. This version of ProtoFold is not a flawless software package and has not yet reached a mature stage. Despite our every attempt to address possible flaws and accommodate conceivable misuse of the interface, unexpected errors and performance issues are still possible. Some of such misconducts are:

This program cannot work with initial input structures with confirmations far different from the actual final structure. The simulation will be trapped in a local minimum, if the initial state of the structure, specifically the main chain dihedral angles: $\phi$ and $\psi$ are too different than the ones found in nature. This issue arises from the present limitations of the methodology used to reach the minimum state of energy in protein structures (3). The longer the sequence of amino acids in a protein structure is, the slower the performance of the program on personal computers. This
version of ProtoFold performs the best when used for protein structures with less than 300 amino acids.

ProtoFold team is currently working on improving our protein model, program efficiency, accuracy and speed to implement in future version of ProtoFold. Some of these ongoing efforts are:

The next generation of ProtoFold is being rewritten in C++ object oriented programming language to provide more flexibility and efficiency than the current MATLAB version. Also, in the future version of ProtoFold the free energy formulation will be improved and extended. The solvation free energy terms, which contribute predominantly to the folding phenomena, will be added to van der Waals and Columbic interactions and hence our model will be more realistic and accurate. Significant speed-ups will be implemented utilizing more efficient algorithms and data structures, in which the computational complexity will be decreased from $O(n^2)$ to $O(n)$ ($n$ being the number of atoms subjected to computational exploitation). ProtoFold group is also implementing efficient algorithm parallelization techniques to target the Single-Instruction Multiple-Thread (SIMT) architecture of the GPU to furthermore accelerate ProtoFold’s performance. More efficient data structures such as link list and hash table will replace matrices to improve data storage and manipulation.

6.7 Collaborators & Acknowledgments

This Graphical User Interface is made possible by generous support from the following individuals and institutions, whose contributions are gratefully appreciated:
for their leadership, support and academic guidance:

- Dr. Kazem Kazerounian
- Dr. Horea Ilies

following graduate students for code development and other technical contributions:

- Carlos Alvarado
- Peter Bohnenkamp
- Hima Khoshreza
- Khalid Latif
- Chris Madden
- Jesse Parker
- Kimberly Rodriguez
- Zahra Shahbazi
- Pouya Tavousi

for the funds provided:

- National Science Foundation, Grant Nos. 0733107 and 085640

for their financial and technical support:

- University of Connecticut, School of Engineering

for MATLAB software:

- MathWorks
7. IMPROVEMENT IN VISUALISATION OF PROTEIN TRANSFORMATION

7.1 Optimizing Visual Based on Change of Position of the Structure

As we continue to improve ProtoFold, visualizing proteins’ conformational transition becomes an important factor in exploring them. In the transition of protein molecules induced either by our energy minimization process or by simply alternating between two confirmations of the same amino acid sequence, the observed motion of the chain is set relative to the first body vector of the chain. In other words, all coordinate data are described in a local coordinate system attached to the structure. The first body vector, unit vector, and the vector perpendicular to their joint plane with the origin laying on the first Nitrogen atom build this coordinate system, in which all other vectors are defined throughout the transformation. Without an absolute view of the movement in a global coordinate system, what the observer sees is a relative motion of the protein in respect to the first link. There is a degree of chaos in the motion of the linkage as a whole, which makes following the motion distracting and focusing on a particular section of the chain hard. This is due to the fixture of the camera (view point of reference) to part of the structure itself. Imagine a snake moving on the ground with a camera fixed to its tail. What is captured of the snake’s propelling body (local view) cannot accurately describe its movement when filmed by a camera fixed to the ground (global view). Since there isn’t a global view of the motion, we are looking to simulate an optimal view of the transitioning protein that resembles a global view and is as calm as possible.

Consider a simpler model of the protein chain: a two dimensional chain with three links \((b_i)'s\) that is set to motion by adding increments of \(\Delta \theta_i \)'s (
Figure 41-a). This motion is seen differently when observed from coordinate systems attached to each link.

Figure 41-b,c,d). When examined closely, a direct correlation between the intensity of chaos in the captured motion and the swept area by the moving linkage in each coordinate system is evident. When adapting the concept to the three dimensional protein chains, again we noticed this association between the intensity of the captured motion and the sweep volume. The prominence of sweep area or volume is a direct result of position change of all atoms. Figure 42 shows traces of protein 1IYT transforming into 1Z0Q seen from two different local coordinate systems. These two proteins are different confirmations of the same amino acid sequence (42 amino acids, 84 main chain links). Clearly, the motion seen from the coordinate system attached to link 48 looks much calmer than the motion capture in the coordinate system fixed to the first link.

Figure 41
When viewed in a given coordinate system, the higher the overall change in position of all atoms during the transformation, the larger the sweep volume produced by the linkage is and the unrulier the captured motion looks. To represent the amount of chaos captured in the movie, we introduce the variable $Z_j$: a measure of change of position for overall system structure in coordinate system $j$, which is the coordinate system fixed with link $j$ on the chain.

$$Z_j = \sum_{t=0}^{T} \sum_{i=1}^{N} (V_i \Delta T)^2$$  \hspace{1cm} (1)

Where:

$V_i \Delta T$: Displacement of atom $i$ in an increment time of $\Delta T$

$N$: Total number of atoms on main chain

$T$: Total time of the transformation
Due to the nature of our method to define protein structure, a global view of the evolving protein is unknown. The only view known is the view of the evolving structure seen from a camera fixed with the first link. We replicate a better view by choosing an optimal link at each step to mount the camera to. We define the optimum link as the link with the least change in positions of all main chain atoms during the course of transformation, when observed relative to this link.

The following is a brief description of the process undertaken to produce an optimal view of the transformation:

\[
\text{for all steps (i) of adding } \Delta \theta
\]

\[
\begin{align*}
\text{calculate all positions in the original coordinate system } p_0 \text{ (coordinate system mounted on the first link)} \\
\text{for all alternative body vectors } b_j \text{ 's} \\
\text{build the transformation matrix to transform points from the original coordinate system (coordinate system attached to the first link) to the coordinate system attached to } b_j \\
\text{calculate points } p_j \text{ in the new coordinate system for conformations i and i + 1} \\
\text{calculate } z_j \text{ (sum of position change of all atoms seen from camera fixed to } b_j \text{ )in step i} \\
\end{align*}
\]

\[
\text{find } j \text{ associated with minimum } z_j \\
\text{to create a seamless transition: build the Displacement matrix to physically displace the new conformation of } b_j \text{ associated with minimum } z_j \text{ onto the same } b_j \text{ in its old conformation (this is done in the original coordinate system)} \\
\text{apply the Displacement matrix to all the other coordinates in the new conformation} \\
\text{plot the new coordinates of the protein seen as if the optimal link carrying the camera is motionless} \\
\text{the new coordinates are now the old coordinates}
\]

\[
\text{end}
\]

Figure 43, Figure 44, and Figure 45 show the analytical result of our method to reach an optimal video of three sets of protein transformation. Part (a) of these figures displays the improvement
of $Z_j$: the measure of position change versus the link number, relative to which all movements are taking place. Optimized through our method, the horizontal level indicates the lower overall position change during the entire transformation. Graph (b) shows the optimum link throughout the transition. For example, in case of transitioning protein 1cfd to 1cll in Figure 43, we have changed our stationary link 7 times capturing the best movie and this has improved the overall change of position of the structure compared to the movie taken from the point of view of the first link by 54 percent (Table 2: Result of visual optimization by position method.

The visual result is available online at www.himakhoshreza.com.
Figure 43: Transformation of 1cfd to 1cll (142 AA / 284 links)
Figure 44: Transformation of 1IYT to 1Z0Q (42 AA / 84 links)
Figure 45: Transformation of 1fox to 2fow (76 AA / 152 links)
<table>
<thead>
<tr>
<th>Transforming pair</th>
<th>Number of Amino Acids</th>
<th>$Z_j$ of first link (Default)</th>
<th>Minimum $Z_j$</th>
<th>Maximum $Z_j$</th>
<th>Optimum $Z_j$</th>
<th>Percentage of improvement from the default movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>1cfd to 1cll</td>
<td>142 (284 links)</td>
<td>4.30x10³</td>
<td>3.41x10³</td>
<td>4.28x10³</td>
<td>2.428x10³</td>
<td>54 %</td>
</tr>
<tr>
<td>1IYT to 1Z0Q</td>
<td>42 (84 links)</td>
<td>9.22 x10³</td>
<td>1.25 x10³</td>
<td>9.23 x10³</td>
<td>1.24 x10³</td>
<td>87 %</td>
</tr>
<tr>
<td>1fox to 2fow</td>
<td>76 (152 links)</td>
<td>5.98 x10³</td>
<td>2.52 x10³</td>
<td>1.03 x10³</td>
<td>2.01 x10³</td>
<td>66 %</td>
</tr>
</tbody>
</table>

Table 2: Result of visual optimization by position method

7.2 Optimization of Visuals Based on Change in Dihedral Angles

We looked into changes of main chain dihedral angles ($\Delta \theta$) and the magnitude of angular velocity ($\omega$) at each joint over the course of the transformation to see if there is a trend to base the decision of what link to choose on these readily available data verses calculating position. Figure 46, Figure 47 and Figure 48 monitor changes of main chain dihedral angles $\Delta \theta$ (in blue), magnitude of angular velocity $\omega$’s measured in respect to the first default link (in green) and measure of change of position seen from all links $Z_j$ (in red). The angular velocity follows a similar trend as the changes in dihedral angles. The areas with continuous uniform lower changes in dihedral angels throughout the transformation seem to be good options for connecting the camera to, however according to $Z_j$ as a measure of quality; these areas themselves produce movies with substantially different qualities. By arbitrarily choosing a link on each of these areas and comparing their respective $Z_j$, the link with lower $Z_j$ can represent the optimal carrier of the camera that can produce a fairly less chaotic transformation movie. This way we reduce the computational demand of finding the lowest $Z_j$ to a minimum. This is a superior alternative to the position method for avoiding tedious computation, but the position method provides the best possible movie with optimum quality.
<table>
<thead>
<tr>
<th>Transforming pair</th>
<th>Number of Amino Acids</th>
<th>Link with minimum $Z_j$</th>
<th>Link with maximum $Z_j$</th>
<th>Percentage of improvement from the default movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>1cfd to 1cll</td>
<td>142 (284 links)</td>
<td>70</td>
<td>254</td>
<td>43%</td>
</tr>
<tr>
<td>1IYT to 1Z0Q</td>
<td>42 (84 links)</td>
<td>48</td>
<td>2</td>
<td>62%</td>
</tr>
<tr>
<td>1fox to 2fow</td>
<td>76 (152 links)</td>
<td>66</td>
<td>146</td>
<td>47%</td>
</tr>
</tbody>
</table>

Table 3: Visual optimization by angle method
Figure 46: 1cfd to 1ccl

Figure 47: 1IYT to 1Z0Q
8. CONCLUSION

The quest for multidisciplinary studies of proteins as nanoscale biological systems is on the rise. Many of these studies focus on structural flexibility of proteins, which is an important attribute in many diseases, and also a key element in building protein based nanobiodevices. Improving our understanding of three dimensional protein structures and their conformational transitioning is an inevitable part of this revelation. This thesis contributes to the above mentioned by following the development of a user friendly graphical interface for a protein simulation package named ProtoFold and exploring enhancements to improve the visual experience of protein’s conformational changes.
ProtoFold’s GUI performs analysis and design of protein molecules in a mechanical framework. This interface is capable of visualizing the 3D structure in several ways. It predicts Hydrogen bonds, visualizes them and also performs mobility analysis to determine the flexibility of proteins. It uses Successive kinetostatic fold compliance method (3) to find the final confirmation of a given protein. This GUI is proficient to report several geometric properties as well as physical constraints of the structure. This graphical interface accelerates our team’s progress. We are also optimistic that ProtoFold’s GUI can help researchers from different disciplines gain a novel understanding and control of proteins by developing alternative studies of them. This can potentially dispense the need for redundant and costly experimentations by shortlisting proteins that show specific flexibility suitable for different applications. That is the power of ProtoFold; a virtual laboratory for protein analysis and simulations.

In an effort to increase the efficiency of visual analysis of protein transformation, we have introduced two novel approaches to improve the quality of captured movies of conformational transition. The first method calls for accumulation of position change of all atoms over the period of the transition to be the deciding factor in selection of a coordinate system, in which the movie is captured. The less change in positions of all atoms, the calmer the captured movie of the transitioning protein looks. In the second method, we look for areas of the main chain with roughly uniform and lower fluctuation of changes in dihedral angels throughout the entire chain. It seems that areas with similar lower changes move together and will make a good option to be the carrier of the camera. Although the second method looks more efficient, the reliance on choosing random parts of the chain makes it ales reliable method. We recommend the position method over this method in case computational efficiency is not a major withhold. The proposed methods can be further applied to capturing movement of other nano structures.
This Graphical user interface is freely accessible for download under research at

9. REFERENCES


