FlexSADRA: Flexible Structural Alignment using a Dimensionality Reduction Approach

by

Shirley Hui

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

Waterloo, Ontario, Canada, 2005

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

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

Abstract

A topic of research that is frequently studied in Structural Biology is the problem of determining the degree of similarity between two protein structures. The most common solution is to perform a three dimensional structural alignment on the two structures. Rigid structural alignment algorithms have been developed in the past to accomplish this but treat the protein molecules as immutable structures. Since protein structures can bend and flex, rigid algorithms do not yield accurate results and as a result, flexible structural alignment algorithms have been developed. The problem with these algorithms is that the protein structures are represented using thousands of atomic coordinate variables. This results in a great computational burden due to the large number of degrees of freedom required to account for the flexibility. Past research in dimensionality reduction techniques has shown that a linear dimensionality reduction technique called Principal Component Analysis (PCA) is well suited for high dimensionality reduction. This thesis introduces a new flexible structural alignment algorithm called FlexSADRA, which uses PCA to perform flexible structural alignments. Test results show that FlexSADRA determines better alignments than rigid structural alignment algorithms. Unlike existing rigid and flexible algorithms, FlexSADRA addresses the problem in a significantly lower dimensionality problem space and assesses not only the structural fit but the structural feasibility of the final alignment.

Acknowledgements

I would like to gratefully acknowledge the supervision of Dr. Forbes J. Burkowski throughout each stage of this work.

I am also grateful to my friends, family and Kevin for their support, understanding and encouragement when it was most needed.

This thesis is dedicated in Loving Memory to Steven Hui.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The maturity of the Human Immunodeficiency Virus (HIV) into the Acquired Immunodeficiency Syndrome (AIDS) virus relies on the cleaving of long chains of polypeptide units into smaller units by an enzyme called HIV protease. The cleaving is performed by the enzyme's ability to exhibit scissor-like motions with the flap parts of its structure. Without the ability to execute this motion, the HIV virus would not mature into AIDS. It is this cleaving motion that researchers have tried to inhibit in order to fight further HIV infection [1].

This is only one example that illustrates how protein structural flexibility is fundamentally involved in a biological process. In reality, proteins are involved in almost all biological processes including immune protection, enzymatic catalysis, and control of cell growth. A protein molecule's ability to carry out its biological function is based largely on its ability to move and flex. Therefore, it is important to understand protein structural flexibility since it determines how a protein will interact with other molecules and decides what the outcome of a biological process will be.

1.1 The Structural Alignment Problem

An important question that arises in Molecular Biology is: How *similar* is one protein molecule to another? The answer is not straightforward since the definition of similarity can mean different things to different people. To the evolutionary biologist, the meaning of similarity might pertain to the protein amino acid sequences and how conserved the sequences are from one protein chain to another. To a drug discovery scientist, the meaning of similarity will pertain to the protein's structure and how geometrically alike one is to another. In this thesis, the structural similarity problem is posed as finding the best three-dimensional alignment between two three dimensional protein structures.

Although easy to state, the problem is hard to solve since proteins are flexible structures. These structures are usually represented by three dimensional atomic coordinates of the protein atoms and consist of hundreds of amino acids in long chains folded into complex structures. They can undergo a range of motions from simple side-chain rotations to movements of entire domains. As a result, there is an immense number of degrees of freedom that must be taken into account when modeling proteins as moving structures resulting in an enormous computational burden. For this reason, many structural alignment algorithms do not model proteins as flexible structures. In fact, most algorithms treat the structure as rigid and static objects that fit together like a lock and key. This simplified version of the problem can lead to incorrect and unrealistic results since proteins in reality can undergo a wide range of conformational changes.

1.2 Previous Work

Numerous rigid structural alignment algorithms have been developed over the years. Such algorithms align two structures without considering the flexibility of the structures. Since protein structures are flexible, rigid alignments may not produce the best results. This has resulted in the development of flexible structural alignment algorithms. These algorithms perform the flexible alignment by breaking the structure into rigid fragments and finding the best alignment with the smallest number of rigid fragments. The gaps between the rigid fragments are considered to be flexible hinge areas. This approach may be problematic since the direction and magnitude of the flexing motion required to obtain the alignment is not considered. Although protein structures may bend, not all structural changes are valid. While the actual alignment may be very good, without assessing the structural feasibility of the flexed structure, these algorithms produce alignments that are impossible to obtain. This highlights the need for an alignment approach that takes into consideration not only protein flexibility but also the feasibility of the final alignment.

1.3 Research Description

A characteristic of all existing alignment algorithms is that they perform the structural alignment in the original high dimensional problem space. As a result, these algorithms are usually quite intricate and complex. However, past research has shown that dimensionality reduction techniques are effective in transforming highly complex problems into problems with much less complexity [2, 3, 4]. A recent application is the use of a dimensionality reduction approach to model high dimensional protein flexibility motion using a significantly reduced number of degrees of freedom [5]. A novel algorithm called FlexSADRA (Flexible Structural Alignment using a Dimensionality Reduction Approach) is introduced in this thesis. FlexSADRA performs a flexible structural alignment by executing a set of straightforward algebraic operations in a much lower dimensional problem space. Unlike FlexProt and other existing flexible structural alignment algorithms, the feasibility of the final flexed structure is assessed using an energy score and the alignment is performed with much less complexity and significantly fewer degrees of freedom.

1.4 Thesis Overview

This thesis summarizes the findings in the fields of protein flexible structural alignment and dimensionality reduction. It then presents the design and test results of the FlexSADRA algorithm. Chapter 1 presents the motivation of this research project and highlights the latest approaches related to the flexible protein structural alignment problem. Chapter 2 covers the background materials pertinent to the design and analysis of the approach. In particular, a literature survey on past research in the areas of both protein structural alignment and dimensionality reduction is presented. Chapter 3 describes the technical design of the approach. The test results are presented and discussed in Chapter 4. Finally, conclusions and recommendations on related topics for future research are given in Chapter 5. Additional information relating to software and hardware used in this research are included in the appendices.

Chapter 2

Background

2.1 Overview

Protein structure and protein structural flexibility play important roles in the regulation of biological processes. This is evident in the example of the role of the HIV protease molecule in the maturation of HIV into AIDS from the previous chapter. There are also many other examples of how structure and flexibility allow a protein to fulfill its biological function. These examples range from simple calcium-regulating proteins that wrap around calcium ions to more complex antibody proteins that protect the body by binding to foreign invaders. As a result, protein structural flexibility is a relevant and important area of research in Structural Biology. In particular, researchers are often interested in determining the degree of similarity between two protein structures.

This chapter discusses past and current research in the areas of protein structure, protein structural flexibility, and protein structural similarity. It also introduces the topic of dimensionality reduction and its different areas of application.

2.2 Protein Structure

Proteins are molecules that are composed of chains of amino acids joined by peptide bonds and folded into three dimensional structures. The structure of proteins can be described in the following levels of detail:

Primary Structure - This is the amino acid sequence. There are twenty different amino acids referred to as residues that may compose a protein chain.

```
ABQLTEEQIAEFKEAFSLFBKBGBGTITTKE
LGTVMRSLGQNPTEAELQBMINEVBABGNGT
IBFPEFLTMMARKMKBTBSEEEIREAFRVFB
KBGNGYISAAELRHVMTNLGEKLTBEEVBEM
   IREANIBGBGQVNYEEFVQMMTAK
```

Figure 2.1: Calmodulin amino acid primary structure

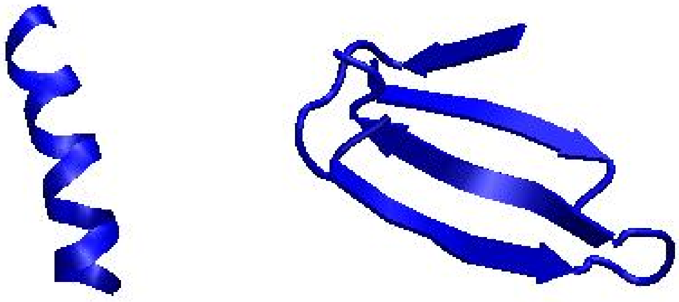
Each amino acid has a backbone, which consists of four atoms: nitrogen (N), carbon (C), α-carbon (Cα) and oxygen (O). The backbone is decorated with different types of side chains and connected to the next amino acid through peptide bonds.

Secondary Structure - These are locally defined substructures like the alpha helix, beta sheets and looser coil formations. Within a single protein molecule there typically are a few to many secondary structure elements.

Tertiary Structure - This is the three dimensional structure that a protein assumes in order to carry out its specific functions. A major area of study is how to predict a protein's tertiary structure given its primary structure. Techniques like X-ray crystallography and nuclear magnetic resonance spectroscopy are used to experimentally deduce the tertiary structure.

Quaternary Structure - This is the protein complex structure that is formed through the union of more than one amino acid chain.

Tertiary or quaternary structures are referred to as conformations. While the protein is



(a) Alpha helix structure

(b) Beta sheet structure

Figure 2.2: Alpha helix and beta sheet structures

performing its biological function, it may shift between several similar conformations. Such transitions are known as conformational changes.

2.2.1 Conformational Space

If atomic coordinates are used to represent a protein's structure, the number of different assignments of values to the coordinates represents the different conformations the protein can assume. The set of all possible conformations is known as the conformational space and for big systems this space is extremely large. However, proteins are made of atoms that have specific chemical and structural properties that cause the protein to prefer certain conformations to others. Therefore, although a protein can assume many different conformations, not all conformations are assumed with equal probability.

2.2.2 Potential Energy Surface

Whether or not a protein molecule adopts a certain conformation depends on the potential energy associated with that conformation. Potential energy is a measurement of the forces acting upon the molecular structure in a given conformation. As a result, a protein's energy is a function of its atomic coordinates. A potential energy surface may be constructed that

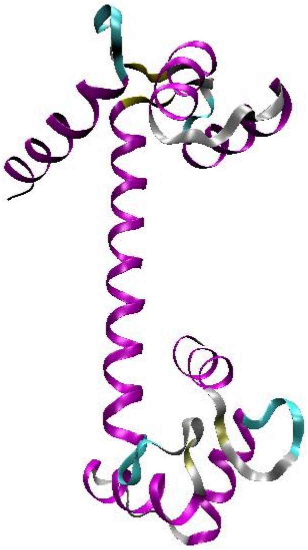


Figure 2.3: Calmodulin tertiary structure

associates a protein conformation with its potential energy. A simplistic concept of the potential energy surface for a protein molecule can be thought of as resembling a funnel with similar conformations close to each other on the surface. The funnel's global minimum represents the conformation that has the lowest potential energy. Local minimums may also exist if the funnel is not smooth, however the general direction for a protein to move toward its globally energy minimized state.

2.3 Protein Flexibility

In the early days of protein structure and flexibility investigations it was widely held that proteins, in particular enzymes, were rigid immutable structures. The notion that a substrate would fit into an enzyme's rigid cavity was likened to a lock and key effect [6]. The development and use of X-ray crystallography allowed the first glimpse of the yeast triose phosphate isomerase protein undergoing very slight conformational changes [7]. The change was only a few Angstroms in an isolated loop region of the molecule, however it was the

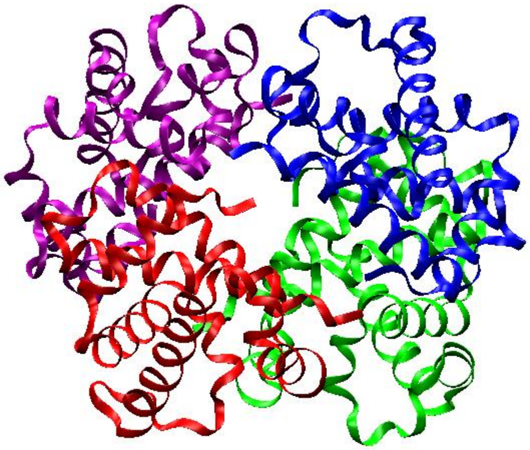


Figure 2.4: Hemoglobin quaternary structure

beginning of evidence suggesting that proteins were not static rigid structures. Today the progress of research and technology has shown that proteins are actually dynamic flexible structures that can undergo large-scale conformational changes (i.e. muscle fiber proteins).

Proteins are built of smaller substructures or domains. The motions associated with these domains are what provide the mechanisms for the wide range of protein flexibility motions observed. Although there are many motions that can be demonstrated, the overall structural changes can be summarized by two basic mechanisms: hinge and shear movements.

2.3.1 Hinge Motion

The protein structures that display hinge motions consist of two domains connected by a hinge area. Lactoferrin is a transport protein that uses hinge movements to recognize and trap small molecules. This protein is composed of two similar lobes. Each lobe has two domains with a hinge between them. Upon binding with iron, the two domains move together rotating as rigid bodies [7].

Another example is the Calmodulin molecule,which is a dumbbell shaped structure that

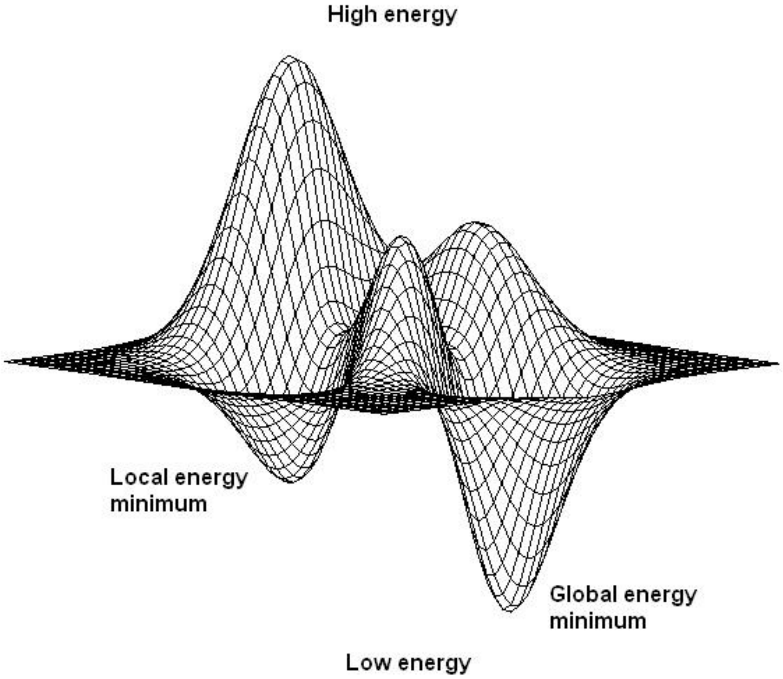
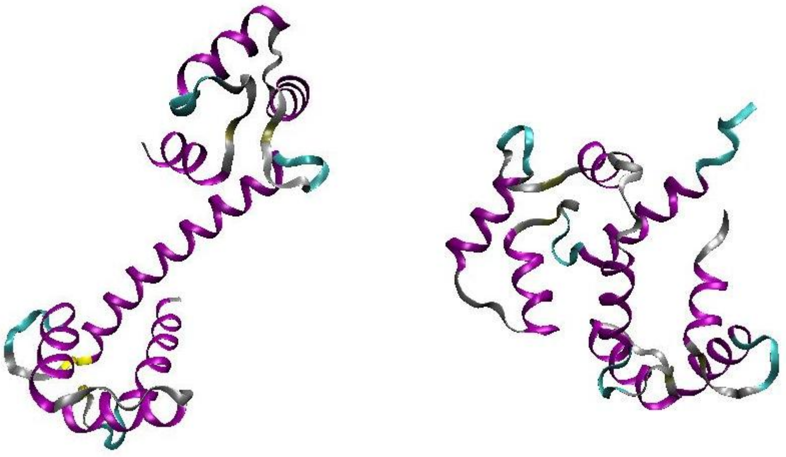


Figure 2.5: Example of a potential energy surface

bends in the middle to enclose a peptide ligand. The movement is described as a set of outstretched arms coming to clasp both hands around the ligand in a secure embrace [7]. In the absence of peptide, the two domains do not interact to a large extent, but the section between the two domains is flexible.

2.3.2 Shear Motion

Shear motions can be described as a stack of blocks with each block sliding slightly to make the stack lean in a certain direction. Individual shear motions are not very large motions, however a number of shear motions can move together to produce a large motion. Structures that exhibit shear motions have layered architectures with one layer or interface sliding over the other. The most common shear interfaces are found at helix-helix interfaces, however sheet-helix, loop-sheet and loop-helix are also possible.



(a) Open (b) Bound

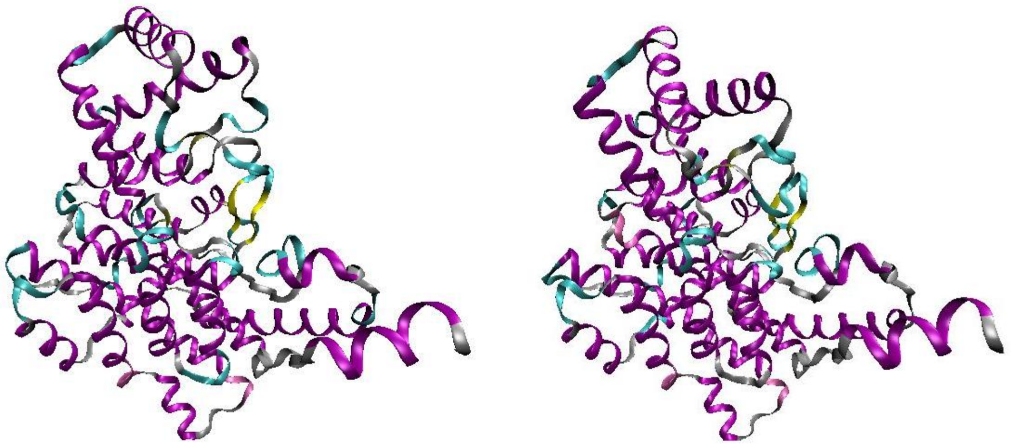
Figure 2.6: Calmodulin open and bound conformations

Citrate synthase is an enzyme involved in the catabolic pathway for the oxidation of fuel molecules in animals. The molecule is a dimer with each monomer consisting of a large fifteen helix domain and a smaller five helix domain. This molecule exhibits shear motion with the smaller domain closing over the larger domain. This is possible by the cumulative shear motions between pairs of packed helices in the small domain.

2.4 Protein Conformational Data

If it could be collected, flexible protein conformational data would consist of the location of the protein's atomic movements over time. Obtaining such data would ideally come from an accurate experimental technique that could capture a large number of movements in a short time interval (i.e. picoseconds or nanoseconds). Unfortunately, such a technique does not exist.

Two commonly used experimental techniques are X-ray crystallography and nuclear magnetic resonance spectroscopy. Although these methods are relatively accurate, they



(a) Open

(b) Closed

Figure 2.7: Citrate synthase open and closed conformations
The shear motion is seen by the movement of the alpha-helices in the domain at the top of the molecule.

are costly to perform and yield only a few structures. Another approach is to perform a molecular dynamics simulation. These simulations are capable of generating large sets of conformational data, but are less accurate than the experimental approaches. These three methods are now discussed in more detail.

2.4.1 X-Ray Crystallography

This technique is based on the fact that X-rays diffract when beamed at a crystal structure. The beam is applied to crystallized protein structures and produces an electron density map. This map is the diffraction pattern caused by the electrons in the crystal diffracting the X-ray. It shows the contour lines of electron density. An example of such a map is shown in Figure 2.8 [8]. Using this map, the locations of atoms can be deduced. Further refinement is required to achieve the final structure.

X-ray crystallography produces highly accurate results, however it is sensitive to the experimental conditions under which it was performed and is an expensive and time-consuming

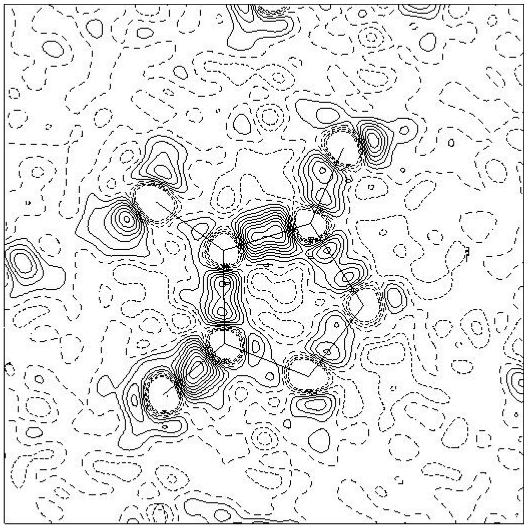


Figure 2.8: Example of an X-ray density plot

process. Furthermore, producing a suitable protein crystal is often difficult since the process of doing so is more of an art than a science. The accurate locations of hydrogen atoms are also difficult to determine since hydrogen atoms only have one electron and therefore do not diffract as much of the X-rays as atoms that have many electrons. Therefore, the electron density around the hydrogen atoms are usually very weak or even too weak to detect [8].

2.4.2 Nuclear Magnetic Resonance

Nuclear magnetic resonance (NMR) is a physical phenomenon that occurs with atoms whose nuclei possess a property called spin. In the absence of a magnetic field, the nuclei will spin in random orientations. However in the presence of a magnetic field, the nuclei will either spin aligned with the field, or opposing the field.

If the strength of the field is increased, the alignments will flip from the low energy aligned state to the high energy opposing state creating a resonance. Different nuclei in a molecule resonate at slightly different frequencies depending on their local environment. By understanding the different resonance signals, each signal may be assigned to an atom or group of atoms in the molecule being studied.

The result of NMR analysis is a data set of distances between atoms, which is used to generate a set of configurations consistent with the data. As a result, NMR produced structures are not as detailed or accurate as those obtained using X-ray crystallography [9]. However, NMR offers some advantages over X-ray crystallography because it is suitable for small proteins, which are not readily crystallized and it is able to determine the positions of the hydrogen atoms.

2.4.3 Molecular Dynamics

Molecular dynamics (MD) simulations supplement or provide an alternative to the costly and time-consuming experimental approaches. The first MD simulation was conducted in the late 1950's using a hard sphere model [10]. Out of this initial work, emerged simulations for liquid argon in 1964 by Rahman [11]. The first realistic simulations were carried out for liquid water in 1974 [12], with the first protein simulations being performed in 1977 on the bovine pancreatic trypsin inhibitor molecule [13]. Today, MD simulations are performed on solvated proteins, protein-DNA complexes or lipid systems to study a wide range of issues like protein folding or ligand docking.

The simulation operates on a set of parameters contained in a forcefield file, which is used to approximate the potential energy surface of a protein using information about bond distances, bond angles, torsion angles, van der Waals and electrostatics. The contributions of these factors on the atoms in a simulated molecule are determined by adjusting the simulation parameters so that the molecule displays characteristics that are consistent with experimental results or first principle calculations. The result is a trajectory of the positions and velocities of the particles in the system and how they vary over time.

2.5 Protein Similarity

Determining the degree of similarity between two protein structures is a frequent question asked by scientists and researchers. As mentioned previously, proteins can be described through their primary structure, which the a sequence of amino acid residues or their tertiary structure, which can be complicated three dimensional structures. Due to the different ways of describing protein structure, there are also different ways to analyze the similarity between proteins. One approach is to find the alignments of proteins represented by their sequences and the other is to find alignments based on their three dimensional structures.

2.5.1 Sequence Similarity

Proteins consist of chains of amino acids and can be described as a string of characters representing the amino acids. The best alignment between two character strings is solved using an algorithm called dynamic programming. This technique was first applied to align protein sequences by Needleman and Wunsch [14]. The method searches and scores all possibilities of alignments to find the optimal alignment. If there are gaps in the alignment where a residue does not align with a residue in the other protein, a penalty is assigned to the score. At the end of the algorithm, the best global alignment is the one with the highest score. A major disadvantage in using dynamic programming is that it requires an increasing number of computations to find an alignment as the number of sequences to align and the lengths of the sequences increase. However, variations of the algorithm working with different scoring schemes and incorporating additional searching constraints have improved the original algorithm to make it more efficient [15]. Today there are numerous algorithms that have been developed to align protein sequences. However, almost all are based on the original Needleman and Wunsch algorithm in varying degrees.

2.5.2 Structural Similarity

The structure and movement of a protein molecule is directly related to its biological function. This is evident in the example of calcium-modulating proteins in the body, which are built and move in a way to wrap around calcium ions. This suggests that similar protein structures can carry out similar biological functions. As a result, researchers are interested in determining how structurally similar two protein molecules are. Although protein sequences may be compared, it is not sufficient to determine the similarity between two protein molecules by simply comparing the protein sequences. This is because over the course of the evolution of the protein molecule, certain changes may have occurred to the sequence. However, the structure of the molecule will not have changed much because changes in the structure would not have survived natural selection. As a result, proteins with very dissimilar sequences may have similar functions.

In order to assess the amount of similarity between two structures, a structural alignment can be performed. In the most basic case, a structural alignment involves determining the best superposition of one three dimensional structure over another three dimensional structure. A structural alignment algorithm that finds such a superposition will often try to minimize the distances between the protein Cα atoms.

2.5.2.1 Rigid Structural Alignment

Simplistic algorithms assume that the protein structures are represented using three dimensional atomic coordinates, the structures are rigid and the molecules are equal in the number of residues being aligned. In this case, the best superposition involves a translation and rotation of one structure to another to minimize the distance between the two structures. A linear algebra method proposed by Kabsch [16] accomplishes this and is one of the most widely used methods when seeking to align two rigid structures of equal length.

More sophisticated algorithms assume that the structures can have different lengths. A search algorithm is used to find the best alignment by scoring different alignments introducing gaps in the alignment if necessary. Although most rigid structural alignment programs follow this general approach, there are many variations that exist. These algorithms differ in the similarity measures, structure representation and search algorithms employed. There have been several algorithms developed in the past but only a few are mentioned below.

A distance matrix alignment algorithm (DALI) was developed by Holm and Sanders [17]. In this algorithm, two dimensional distance matrices of residue-residue (Cα-Cα) distances describe the three dimensional structures. The measure of similarity is based on the assumption that similar structures have similar residue-residue distances. Alignments are formed by searching for the best alignment of the distance matrices using the residue-residue distances to measure the level of similarity between the two matrices. In the original algorithm a simulated annealing search algorithm was employed, but later versions used Monte Carlo methods.

MINAREA was developed by Falicov and Cohen [18]. This algorithm finds alignments between structures using dynamic programming to search for a minimal area triangulation between the two structures. The process is described as trying to minimize a soap bubble surface between the two structures.

Combinatorial Extension (CE) was developed by Shindyalov and Bourne [19] and involves building a global alignment by breaking the protein chains into fragments. These fragments are aligned to produce aligned fragment pairs. Alignments are extended by combinatorially starting from each fragment pair defining a set of paths joining the fragments with gaps if needed. Dynamic programming is used to find the optimal alignment.

2.5.2.2 Flexible Structural Alignment

Since proteins can bend and flex, rigid alignment methods do not provide accurate solutions to the protein structural similarity problem. If the one of the molecules is simply hinge-bent with respect to the other molecule, rigidly aligning the two molecules will not give accurate results. Therefore, algorithms have been developed to incorporate flexibility into the search for the best alignment. Currently, only a small number of flexible structural alignment algorithms have been developed. In these algorithms, one protein is considered flexible while the other is rigid.

FlexProt was developed by Shatsky and Nussinov [20] and tries to find the largest flexible alignment between two protein chains by decomposing them into a minimal number of rigid fragment pairs having similar structure. As in the classic rigid alignment approach, two fragments are similar if they have the same number of atoms and there exists a three dimensional rotation and translation that best superimposes the corresponding atoms. Aligned fragments pairs are consistent with their order on the protein chain and between these fragments are proposed flexible hinge areas.

FATCAT was developed by Ye and Godzik [21]. The algorithm begins by identifying a list of aligned fragment pairs in the two proteins to be compared. The alignment is produced by chaining the aligned fragment pairs, allowing flexibility in connecting them. A rotation/translation (twist) can be introduced between two consecutive fragment pairs if it results in a substantially better superposition of the structures. Dynamic programming is used to find the alignment and detect hinges according to a scoring function introducing gaps as necessary.

The flexed structure in the final alignments produced by these algorithms are composed of disjoint rigid fragments. The gaps in the fragments represent areas that the molecule can bend or twist. Since flexible structural alignments rely on the flexible molecule bending in a particular way, a structural energy calculation should be made in order to assess the structural feasibility of the flexed structure. However, both algorithm do not do so. If the flexed structure was a complete structure, it could be submitted to an energy calculation application by the user. However energy calculation applications require as input complete structures with all atomic coordinates present. Therefore, it is difficult to determine the feasibility of the alignments produced by these algorithms.

2.6 Dimensionality Reduction

Dimensionality reduction is an important topic in statistical machine learning, data analysis and scientific visualization. These techniques try to discover the true dimensionality of high dimensional data.

The concept of a manifold is central to the understanding of dimensionality reduction. A manifold is a topological space, which is locally Euclidean. Using the Earth as an example, when looking at the Earth from outer space, it appears to be round. However, when looking at the Earth's horizon on the corner of a street, the Earth appears to be flat. In general, any object like the Earth's surface, which is nearly flat on small scales is a manifold. Manifolds like many other surfaces can be globally linear or nonlinear.

It is often the case that the high dimensional data actually reside in a much lower dimensional space or manifold. Discovering the true dimensionality of data is facilitated by discovering the lower dimensional manifold on which the data actually lie.

Dimensionality reduction techniques can be very broadly described as being linear or nonlinear techniques.

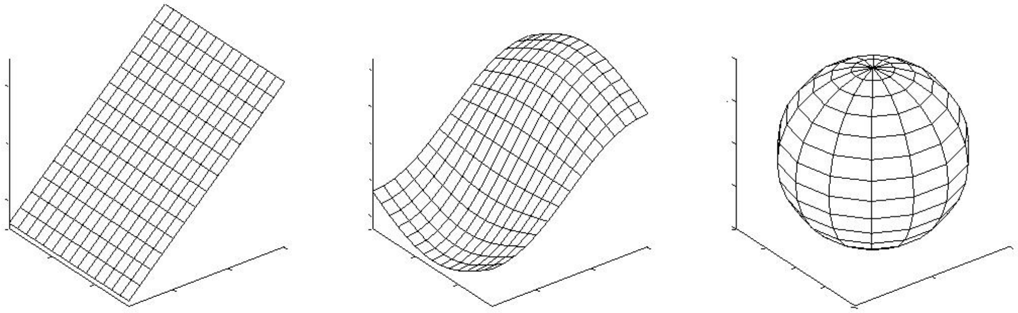


Figure 2.9: Examples of different types of 3-D manifolds

2.6.1 Linear Techniques

Linear dimensionality reduction techniques seek to find a transformation that maps points in high dimensional space to a set of corresponding points in a lower dimensional space. The dimensionality reduced data is simply a transformation of the original data onto an underlying linear manifold.

Linear techniques assume that the high dimensional data resides on a lower dimensional linear manifold or subspace, and can be represented as a linear function of its parameters. The main drawback of linear approaches is their inherent limitations as linear methods. If the data actually lies on a nonlinear manifold, using a linear method will not accurately capture relationships in the data. In fact, linear methods will represent the true reduced manifold in a subspace higher than necessary in order to capture the nonlinearities in the data. In this case, linear methods may also violate the geodesic distances between points by mapping faraway points to nearby points in the reduced space.

Despite these shortcomings, linear methods are popular techniques for dimensionality reduction because they are fast and straightforward to implement, do not require many free parameters and do not run into convergence problems like many of the nonlinear methods do [5].

2.6.2 Applications

Principal component analysis (PCA) is one of the oldest but most popular techniques developed for the analysis of multivariate data. It was first introduced by Pearson [22] in 1901 and independently developed by Hotelling [23] in the 1930's. Even though the PCA technique has been around since the early 1900's, it still remains a popular choice for dimensionality reduction today and has been applied to many problems in a wide variety of different fields.

2.6.2.1 Face Recognition

Faces can be represented as a matrix of pixels yielding an image. Given a new face image, face recognition systems should be able to decide whether or not the new face has been seen before or even if it is a face at all. Intuitive methods try to compare distinguishable facial features like noses, eyes, or mouths. Goldstein, Harmon and Lesk [24] made the first attempts using this approach in the 1970's however, this proved to be extremely hard to do. The matrix representation of a face involved many variables making the problem solution extremely complex. Also, the measurements made for the facial features were subjective because they were made by hand.

Kirby and Sirovich first applied PCA to the face recognition problem in the 1980's [25]. Although their work has since undergone additions or improvements by other researchers, the basic idea is to use PCA to decompose the original variables used to describe a face into a smaller number of new variables. These new variables represent new facial features. Although they do not correspond to any of the familiar facial features like a nose, eyes or mouth, they are better at describing the original data. These new variables are known as eigenfaces and can be combined in different proportions to represent a given face. A new face can be compared to these combinations of eigenfaces to determine whether or not it matches any face previously seen or if it is a face at all.

2.6.2.2 Microarray Gene Expression Analysis

Microarrays are powerful tools by which the expression levels of thousands of genes can be monitored simultaneously. The array consists of a collection of DNA spots attached to a silicon chip. Each spot represents the complement of a transcript that might be expressed in a gene and can hybridize with cDNA from different types of cells, which are tagged using different fluorescent colours. When combined, some of the tagged cDNA will bind to certain spots on the array in varying amounts. The tags will fluoresce under a laser and the intensities of the colours seen indicate the level of expression for a gene.

Principal component analysis has been applied to the data represented by the arrays as a pattern analysis and clustering tool. In separates studies, Alter and Holter applied PCA to decompose cell-cycle gene expression data into eigengenes [26, 27]. Both studies found that cyclic patterns were observed for the first two eigengenes. This suggested that cell-cycle gene regulation might be more of a continuous process than implied by previous studies. Holter et al. also applied PCA to yeast sporulation time series data to produce a set of eigengenes that indicated a strong separation of gene groups in the data. The experiment showed the usefulness of PCA as a preprocessing tool for classification studies on gene expression data.

2.6.2.3 Protein Structure Flexibility

Teodoro et al., applied principal component analysis to model protein flexibility motion [5]. The motion was originally modelled using the coordinate positions of the various atoms movements of the protein molecule in three dimensional space. The PCA method was used to decompose the motion into a set of new variables called collective modes of motion. Together, these modes describe the protein's motion and form a significantly reduced basis of representation.

Their work showed that these collective modes could model the opening and closing motion of the flap regions of the HIV protease molecule. Instead of the several thousand atomic coordinates used to represent the protein's movement, the motion could be represented using only a few hundred collective modes.

2.7 Chapter Summary

Proteins are flexible structures that are involved in many important biological processes. The way a protein is shaped and the way it moves is associated with its biological function. As a result, protein structure similarity and more specifically protein structural alignment are interesting areas of research. The majority of research in these areas has produced algorithms that align rigid structures. However, proteins are flexible and therefore protein flexibility should be incorporated into these algorithms.

This chapter has presented various rigid and flexible protein structural alignment algorithms. Most of these algorithms are derived from the original dynamic programming approach and model the protein structure in high dimensional atomic coordinates.

Dimensionality reduction techniques were also introduced specifically a linear technique called principal component analysis. Different examples of the use of this technique were presented showing that it is a powerful method for dimensionality reduction with diverse and relevant applications.

Chapter 3

Design and Analysis

3.1 Introduction

Past research has demonstrated the effectiveness of using a linear dimensionality reduction technique called principal component analysis to model protein flexibility using significantly fewer degrees of freedom than existing approaches. The PCA technique has also been used in a number of other areas of research including face recognition systems. In fact, the problem of flexibly aligning two structures is very similar to the problem of recognizing a face. As a result, the dimensionality reduction approach to flexible structural alignment draws ideas from past research in modelling protein flexibility as well as previous work in face recognition systems. This chapter describes a new algorithm called FlexSADRA (Flexible Structural Alignment using a Dimensionality Reduction Approach). This algorithm shows that it is possible to not only address a complex problem like flexible structural alignment, but that it is possible to do it in a significantly reduced dimensionality problem space.

3.2 A Dimensionality Reduction Approach

3.2.1 Principal Component Analysis

One of the most common linear dimensionality reduction techniques is principal component analysis (PCA). This technique has been the areas of image compression, face recognition, and microarray data analysis [4, 3, 2].

Dimensionality reduction using PCA is achieved by transforming the original variables describing the data, to a set of new variables called principal components. The PCA algorithm is described as the following series of steps.

Let the i-th observation in a set of observations be represented by n variables in the form of an n-dimensional vector:

$$X_i = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{n,i} \end{bmatrix} \tag{3.1}$$

A set of m observations may be represented as a $n \times m$ matrix. In the case of high dimensional data, the number of variables n is usually much larger than the number of observations m. Before PCA is applied to the matrix, it must be mean-centered so that the n rows of the matrix sum to be zero. The result is a mean-centered $n \times m$ matrix X, where $n \geq m$:

$$X = [X_1, X_2, \ldots, X_m] = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,m} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{n,1} & \cdots & \cdots & x_{n,m} \end{bmatrix} \tag{3.2}$$

The first step of PCA is to find a linear combination $U_1^T X$ which have maximum variance and where U_1 is a vector of n constants:

$$U_1 = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ \vdots \\ u_{n,1} \end{bmatrix} \tag{3.3}$$

The resulting linear combination is:

$$U_1^T X = \begin{bmatrix} u_{1,1}x_{1,1} + u_{2,1}x_{2,1} + \cdots + u_{n,1}x_{1,m} \\ u_{1,1}x_{2,1} + u_{2,1}x_{2,2} + \cdots + u_{n,1}x_{2,m} \\ \vdots \\ u_{1,1}x_{n,1} + u_{2,1}x_{n,2} + \cdots + u_{n,1}x_{n,m} \end{bmatrix} = \begin{bmatrix} \sum_{j=1\ldots n} u_{j,1}x_{j,1} \\ \sum_{j=1\ldots n} u_{j,1}x_{j,2} \\ \vdots \\ \sum_{j=1\ldots n} u_{j,1}x_{j,m} \end{bmatrix} \tag{3.4}$$

The next step is to find another linear combination $U_2^T X$ of the elements of X, which have maximum variance and are uncorrelated with $U_1^T X$. At the d-th step, a linear combination $U_d^T X$ of the elements of X is found, which have maximum variance and are uncorrelated with $U_1^T X$, $U_2^T X$,..., $U_{d-1}^T X$. Up to n of these linear combinations may be found and each of the linear combinations is a principal component. The resulting matrix is:

$$U = [U_1, U_2, \ldots, U_n] = \begin{bmatrix} u_{1,1} & \cdots & \cdots & u_{1,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ u_{n,1} & \cdots & \cdots & u_{n,n} \end{bmatrix} \tag{3.5}$$

The first component specifies the most variation in the data, while the d-th component specifies the least amount of variation. Dimensionality reduction is achieved by using the smallest set of components that account for the most amount of variation in the data. This truncated basis of representation consists of only the significant degrees of freedom that describe the data.

The set of principal components forms a matrix that transforms the original data X to points Y in the new lower dimensional space. Since each component specifies decreasing amounts of variance in the data, dimensionality reduction is achieved by using the d components that explain the largest amount of variance in the original data as follows:

$$Y = {U_d}^T X \tag{3.6}$$

where

- X $n \times m$ matrix
- U_d $n \times d$ matrix the first d orthogonal principal components
- Y $n \times d$ matrix where $d << n$

Y is the orthogonal projection of the n-dimensional data X onto a d-dimensional subspace using the transformation matrix U_d.

The original data may also be approximately reconstructed using Y and the transformation matrix U_d as follows:

$$X_d = ({U_d}^T)^+ Y \tag{3.7}$$

where $({U_d}^T)^+$ is known as the pseudoinverse[1] of U^T [28]. The approximation X_d is the least squares estimate of X.

Since only d principal component are used, there is a certain amount of information loss if $d \neq n$. The reconstruction error is minimized in a least squares sense according to:

$$Error = ||X - X_d|| \tag{3.8}$$

[1] The pseudoinverse of a matrix U is calculated by $U^+ = (U^T U)^{-1} U^T$

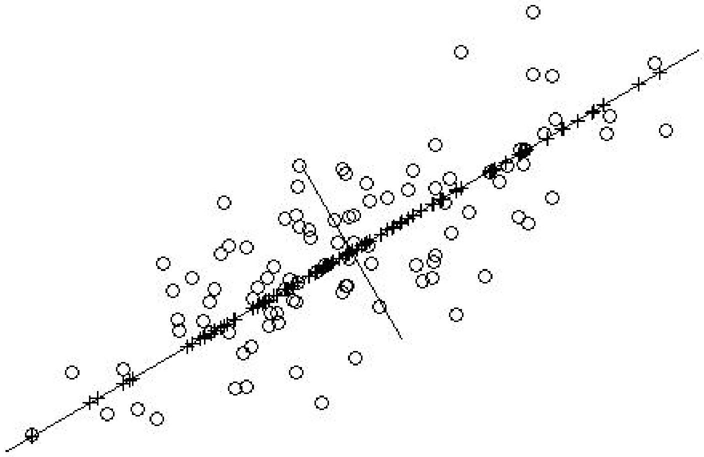


Figure 3.1: Principal component analysis applied to 2-D data
The o symbols are the original 2-D data and the + symbols are the 1-D orthogonal projections of the original 2-D data onto the first principal component.

3.2.2 Singular Value Decomposition

Principal components can be computed in various ways, but the eigenvector decomposition method of Singular Value Decomposition (SVD) is commonly used. The following describes SVD in more detail.

If X is an $n \times m$ matrix and assuming $n \geq m$, then X has a singular value decomposition as follows:

$$X = U \Sigma V^T \tag{3.9}$$

where

- U $n \times n$ matrix. The columns of U are orthogonal to each other, that is $U^T U = I$. The columns of this matrix are known as the eigenvectors of X.
- V $m \times m$ matrix. The columns of V are orthogonal to each other, that is $V^T V = I$.
- Σ $n \times m$ matrix whose off-diagonal entries are all 0's and whose diagonal entries

are the elements σ_1,σ_2,...,σ_m where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$. The σ's are known as the singular values of X.

The $n \times m$ matrix Σ is given by:

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & 0 \\ & & \ddots & \\ & 0 & & \ddots & \\ & & & & \sigma_m \\ & & 0 & & \end{bmatrix} \tag{3.10}$$

Since the matrix X will have many small singular values σ, it may be approximated by a matrix of much lower rank d, where $d \leq m$ as follows:

$$X_d = U_d \Sigma_d V_d^T \tag{3.11}$$

where

- U_d $n \times d$ matrix formed by taking the first d columns of U
- V_d $m \times d$ matrix formed by taking the first d columns of V
- Σ_d $d \times d$ diagonal matrix whose elements are the first d singular values of X

Using only d columns of U and V and only d singular values, the original matrix X may be approximated by X_d with good accuracy. The error in the approximation is given by the Euclidean norm as follows:

$$\begin{aligned} Error &= ||X - X_d|| \\ &= ||U\Sigma V^T - U_d\Sigma_d V_d^T|| \\ &= ||U\Sigma_{d+1\ldots m}V^T|| \\ &= \sum_{j=d+1\ldots m} \sigma_j \end{aligned} \tag{3.12}$$

since

$$\Sigma_{d+1\ldots m} = \begin{bmatrix} 0 & & & & & \\ & \ddots & & & 0 & \\ & & 0 & & & \\ & & & \sigma_{d+1} & & \\ & 0 & & & \ddots & \\ & & & & & \sigma_m \\ & & & 0 & & \end{bmatrix} \tag{3.13}$$

If the dimensionality n of the $n \times m$ matrix X is to the reduced, the eigenvectors, in U are the principal components discussed in the PCA section. In this thesis, $n > m$ in most cases or the number of dimensions n is greater than the number of observations m. In all cases, the value d is always much less than both n and m.

3.2.3 Modeling Protein Flexibility

Principal component analysis has been used by Teodoro et al. to reduce high dimensional protein motion data [5]. In their work, the protein motion is a set of protein conformations described using atomic coordinates. The i-th protein conformation in the data set is represented as a $n = 3N$ vector as follows:

$$X_i = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ x_{3,i} \\ \vdots \\ x_{3N-2,i} \\ x_{3N-1,i} \\ x_{3N,i} \end{bmatrix} \tag{3.14}$$

where, N is the number of atoms in the structure and for $j = 1 \dots N$:

- $x_{3j-2,i}$ elements are the x coordinates
- $x_{3j-1,i}$ elements are the y coordinates
- $x_{3j,i}$ elements are the z coordinates

The data set is a $n \times m$ matrix X, where m is the number of conformations and $n = 3N$ is the number of atomic coordinates.

$$X = [X_1, X_2, \dots, X_m] = \begin{bmatrix} x_{1,1} & \dots & \dots & x_{1,m} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{n,1} & \dots & \dots & x_{n,m} \end{bmatrix} \tag{3.15}$$

Applying PCA to the data set results in a set of principal components that also represent the protein motion. These principal components describe the protein's flexibility in varying degrees and can be combined in different proportions to reconstruct the original conformations in varying degrees of accuracy. As a result, only the first few principal components that retain a certain amount of the original flexibility are used to model the flexibility. The result is an approximate description of the protein flexibility motion, using only a few hundred principal components or degrees of freedom, compared to several thousand atomic coordinates.

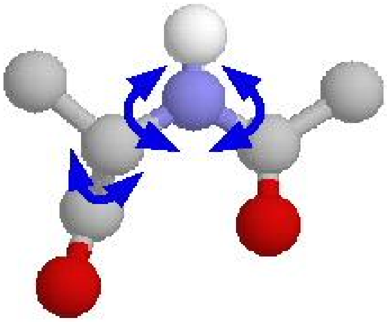


Figure 3.2: Three degrees of freedom associated with a small molecule

Although protein flexibility motion is nonlinear, Teodoro et al. use a linear method like PCA for the following reasons:

- The mappings between high and low dimensional spaces are fast and straightforward to compute
- The validity of the method has been established by comparison with laboratory experimentally derived data [5]
- The increased computational cost, difficulty of implementation, and problematic solution convergence make nonlinear techniques less attractive to use

Despite these points, the application of nonlinear techniques should be investigated further [29]. If the original data is nonlinear, linear methods like PCA will represent the true reduced space in a higher dimensional space than is necessary in order to account for the nonlinearity.

3.2.4 Face Recognition

In face recognition applications that use PCA, there is a training set of faces that is used to find a set of principal components or eigenfaces that describe a lower dimensional feature space called a face space [3]. Each eigenface represents a characteristic feature from the faces in the training set. Therefore, a face can be composed of a combination of these eigenfaces in the right proportions.

Given:

- A set of n-dimensional faces $\Gamma = [\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_m]$
- A new n-dimensional face Γ_0
- A matrix U of d principal components or eigenfaces that can be used to transform a face onto the face space $\Re^n \rightarrow \Re^d$

When a new face enters the system, it is first transformed into the lower dimensional face space using the d eigenfaces. The transformation is computed as:

$$\Phi_p = U^T \Gamma_0 \tag{3.16}$$

where Φ_p is the representation of the new face expressed as a combination of eigenfaces. The matrix U is the column-wise concatenation of eigenfaces that acts as a transformation matrix from the original space to the face space.

An averaging technique is used to determine which class, among a set of predefined face classes, the new face belongs to. This is typically done by classifying the low dimensional representation of the training faces into a set of face classes and calculating the average faces for each class. To determine which class a new face belongs to, the following Euclidean distance is minimized:

$$\epsilon_k = \|\Phi - \Phi_k\| \tag{3.17}$$

where Φ_k is the average face in class k. If ϵ_k is less than some threshold θ_k, the new face is classified as being in class k. Otherwise, the face is classified as being unknown and may be used to represent another face class.

The extreme most transformed training faces bound the face space. This area resembles an ellipsoid [3]. Points that lie within this ellipsoid can be considered as lying within the face space. In some cases, the transformation of new faces will lie outside this space and are considered non faces. Whether or not the new face is actually a face or not can be calculated by calculating the Euclidean distance between the reconstructed face $\Gamma_r = (U^T)^+\Phi$ and the original new face Γ_0.

$$\epsilon_r = \|\Gamma_0 - \Gamma_r\| \tag{3.18}$$

If the distance exceeds a certain threshold θ_r, the face most likely is not a face at all. An appropriate value for θ_r is the radius of the ellipsoid as determined by the distance between the center of the ellipsoid to its farthest point, shown in Figure 3.3.

3.2.5 Flexible Structural Alignment

A dimensionality reduction approach to the flexible structural similarity problem is similar to the approach used in face recognition systems.

Given:

- A set of n-dimensional protein conformations $X = [X_1, X_2, X_3 \dots X_m]$, as described in (3.15)
- An n-dimensional rigid protein structure X_r
- A matrix U_d of d principal components that transforms conformations in the original space to the flexibility space $\Re^n \rightarrow \Re^d$

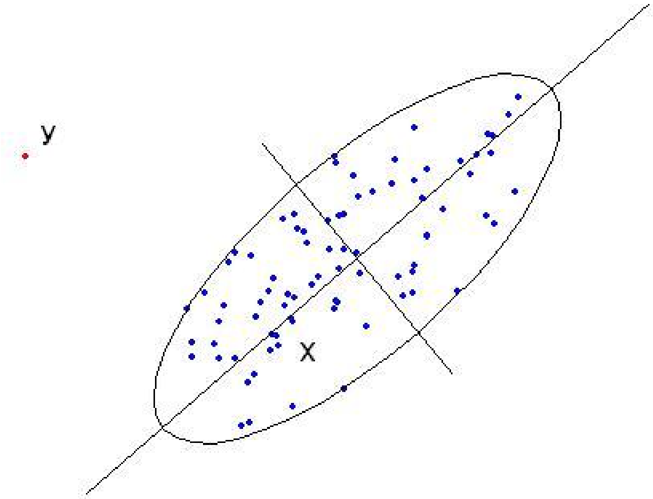


Figure 3.3: Threshold for feasible conformations
Point y is an example of a transformed point lying outside of the ellipsoid describing the space occupied by the training points.

As in the case of face recognition, the application of PCA to the conformations X yields a set of principal components or flexibility features. The flexibility features describe a lower dimensional flexibility space that represents the space where the original conformations X may actually reside. The goal is to find a linear combination of these features that yields a point that lies in the flexibility space and, when it is reconstructed in the high dimensional space, is as similar to the rigid structure X_r as possible. The application of the transformation U_d to the rigid structure X_r yields a d-dimensional point Y that satisfies this criteria. The point Y is simply a projection of X_r onto the flexibility space:

$$Y = {U_d}^T X_r \tag{3.19}$$

When the point Y is transformed back to the high dimensional conformation space, it represents a novel flexed conformation X_f. The transformation back to the original high dimensional conformation space is achieved by:

$$X_f = ({U_d}^T)^+ Y \tag{3.20}$$

Since X_f is the least squares approximation of the original data X_r, it is closer to X_r than any other reconstructed conformation in X.

3.3 The FlexSADRA Algorithm Design

3.3.1 Input

The input for the algorithm is a matrix of m conformations that are represented as n-dimensional vectors. The n-dimensional vectors contain only the atomic coordinates of the nitrogen (N), carbon (C), α-carbon (Cα) and oxygen (O) atoms along the backbone[2]. Side chains are not included.

3.3.2 Scoring

Good flexible structural alignments should be based on two criteria: how close the two structures are aligned and how feasible the flexed structure is. Existing algorithms only measure how close the structures are. However, just because a structure has been flexed to produce a close fit, does not mean that the flexing motion required to achieve the closeness desired is actually possible. Therefore the structural feasibility of the flexed structure should also be determined to see whether or not the alignment is actually feasible. The scoring methods are now discussed in more detail.

[2] Subsequent use of the term backbone in this chapter refer to the N, C, Cα, O atoms of the protein structure

3.3.2.1 Scoring the Fit

Root Mean Squared Deviation (RMSD) is commonly used to represent the distance between two objects. In a structural sense, this value indicates the degree to which two three dimensional structures are similar. The lower the value, the more similar the structures are. The most frequently used calculation for structural similarity is the coordinate based RMSD.

Given:

- n atomic coordinates for structure X
- n atomic coordinates from structure Y

The goal is to find the Euclidean distance in n dimensions.

$$RMSD(X,Y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.21}$$

Typically, an RMSD value of less than 1.0 Å for two structures indicates a very good fit or that two structures are almost identical.

3.3.2.2 Scoring the Structural Feasibility

The structural feasibility of a three dimensional protein structure may be assessed using an energy value that indicates how good the conformation is compared to another. Although this value can be exactly computed using quantum mechanics, the computation would be too costly and take too long for large molecular systems. As an approximation, classical laws of physics may be used to derive energy functions based on the conformation of a protein. Recall that the potential energy value represents how likely it is that a protein will assume a certain conformation. For instance, a protein conformation with a high energy value will not be in a preferred state. Instead the protein molecule will move until it assumes a conformation with a minimal energy state. Low energy values indicate a likely conformation.

The energy function is composed of individual energy terms that correspond to an aspect of the conformation that contributes to the overall energy. These terms are added together to produce a total energy value describing a given protein conformation X.

A simple energy function may consist of terms representing the energies contributed by the bonds between atoms, angles of the bonds, torsions or rotations about the bonds between atoms, and non bonded terms like van der Waals and electrostatics.

$$E_{total} = \sum_{bonds} E_{bonds}(X) + \sum_{angles} E_{angles}(X) + \sum_{torsions} E_{torsions}(X) + \sum_{nonbonded} E_{nonbonded}(X) \tag{3.22}$$

Energy functions are designed differently by different researchers. The variations range from the types of terms used in the function to the specific values of the numerous constants required for the function. The constant values are usually determined through experiments and can vary from one implementation to another. In practice, these values are contained in a forcefield parameter file and must be made available to the algorithm calculating the energy. The implementations and forcefields commonly used include those developed by Amber (Assisted Model Building with Energy Refinement) and CHARMM (Chemistry at HARvard Macromolecular Mechanics) to name only two. The CHARMM forcefield is the one used in this thesis and by NAMD.

The energy calculating application used in this thesis is called MDEnergy (Molecular Dynamics Energy). Like other energy calculating applications, the required input is the entire protein structure, including side chain atoms. Since the algorithm input is only the backbone atoms, the missing side chain atom coordinates are estimated using an application called psfgen.

MDEnergy also allows the user to specify the energy terms and atoms to be included in the total energy calculation. Since the algorithm only uses the backbone atoms, only the conformational energy corresponding to the backbone of the structure is calculated. The conformational energy consists of the bond, angle, and torsion energy terms.

Please refer to the Appendix for more details on the energy calculation, and applications mentioned in this section.

3.3.3 Initial Alignment

An initial alignment must be performed in order to determine equivalence between residues in the two structures to be aligned [7]. There are numerous alignment algorithms that exist to align protein sequences. However, due to structural evolution in proteins, mutations, insertions and deletions might have occurred in the sequences resulting in an alignment that is not meaningful. While the sequences between two protein molecules might be very different, the structures are quite similar. In such a case, the two structures may be aligned using rigid structural alignment methods to obtain an initial sequence alignment. Obtaining an initial alignment is a common step to a starting point in many alignment algorithms [30]. The algorithm only flexes the sections of the protein molecule corresponding to aligned residues between the flexible and rigid proteins.

3.3.4 The FlexSADRA Algorithm

The FlexSADRA algorithm is now presented:
Given:

- X_r $n \times 1$ Cartesian coordinates of the rigid structure backbone atoms
- X $m \times n$ Cartesian coordinates of the m flexible structures backbone atoms
- d dimensionality to reduce to, $d << n$

1. Reduce dimensionality of X to determine a lower dimensional flexibility space.

- Apply PCA to X by using SVD to obtain d principal components that are the columns of the matrix U_d

2. Represent Y in the d dimensional flexibility space.

   - Project X_r using U_d to obtain Y:

$$Y = {U_d}^T X_r \tag{3.23}$$

3. Transform Y back to high dimensional space to obtain a novel flexed conformation X_f that is closer to X_r than any other reconstructed conformation in X.

   - Perform the pseudoinverse of ${U_d}^T$ with Y to obtain X_f:

$$X_f = ({U_d}^T)^+ Y \tag{3.24}$$

4. Assess the quality of X_f

   - Calculate RMSD between X_f and X_r
   - Calculate conformational energy score for X_f

A high level illustrated overview of the first three steps of the FlexSADRA algorithm is shown in Figure 3.4.

3.3.5 Estimating Missing Backbone Coordinates

A limitation of the described algorithm is that the structures being flexed must be of the same length. This is because the projection step of the algorithms can only be performed on objects with the same number of degrees of freedom. Therefore, only the aligned parts of the structures as determined by the initial alignment can be flexed. If the structures have different lengths, the final flexed structure may have gap sections where the atomic

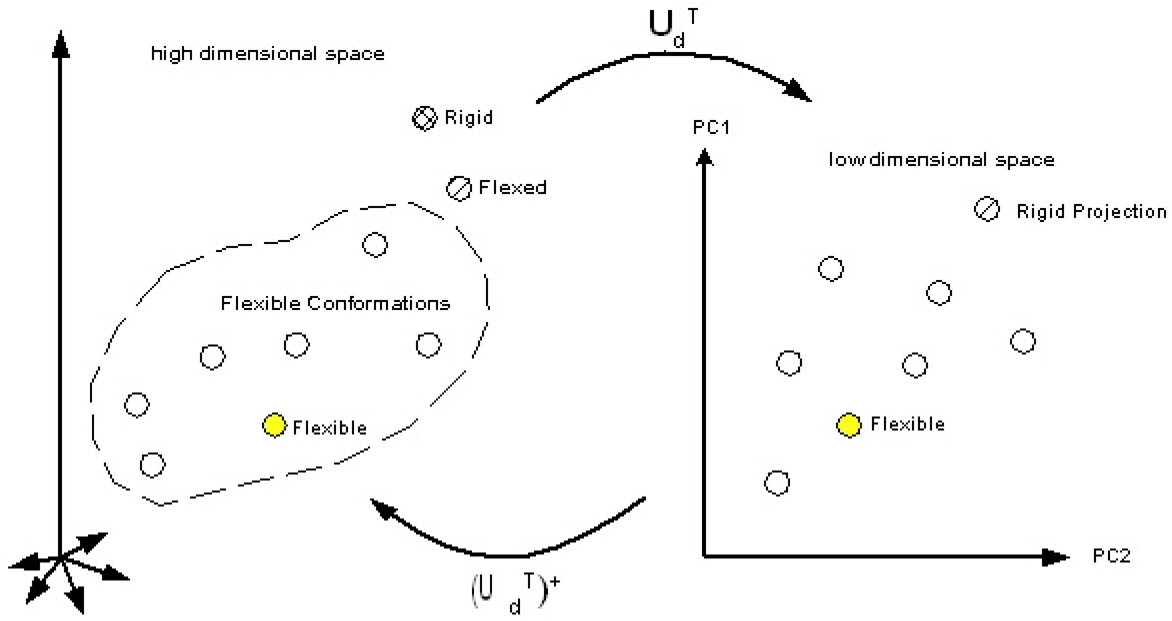


Figure 3.4: Illustrated summary of the FlexSADRA algorithm
The Rigid conformation in high dimensional space is transformed to a low dimensional space represented by two principal components. The transformation is a projection operation via the matrix U_d resulting in the Rigid Projection point. The Rigid Projection is transformed back to high dimensions via the pseudoinverse U_d^+ to yield the Flexed conformation. The Flexed conformation is closer to the Rigid conformation than any other reconstucted point in the Flexible Conformations set (the reconstructed points in the Flexible Conformations set are not shown).

coordinates are missing. This is not desirable since the location of atoms in the gap areas must be known in order for the potential energy to be calculated. In addition, a complete flexed structure is more practical for further analysis than fragments of aligned sections. This problem does not limit the approach to structures of the same length however, since the missing coordinates may be estimated using a variety of methods.

Estimating missing data is a prevalent topic in statistical analysis research. This area of research concerns the analysis of matrix data where some of the values in the matrix are not observed. *Adhoc* methods of dealing with missing data include eliminating entire objects with missing observations or replacing the missing values using mean values. However, both of these methods are not ideal for dealing with the missing atomic coordinate information.

An alternative approach would be to estimate the missing coordinates using a least squares estimate approach [28].

For the case with no missing coordinates, recall the projection X_r of the rigid structure from Step 2 of the algorithm:

$$Y = {U_d}^T X_r \tag{3.25}$$

where

- U_d $n \times d$ principal components
- X_r $n \times 1$ rigid structure in high dimensions
- Y $d \times 1$ projection of rigid structure onto the lower dimensional space

In the case where there are missing coordinates due to gaps in the alignment, introduce the following notation:

- t the length of the alignment
- ${U_d}^*$ $t \times d$ principal components corresponding to the aligned coordinates
- X_r^* $t \times 1$ aligned coordinates of the rigid structure in high dimensions
- Y^* $d \times 1$ projection of the aligned coordinates of the rigid structure

The projection Y^* would be obtained as follows:

$$Y^* = {U_d^*}^T X_r^* \tag{3.26}$$

Using the complete matrix U_d, the least squares estimate for the missing coordinates in X_f is calculated as follows:

$$\hat{X_f} = (U_d^T)^+ Y^* \tag{3.27}$$

This method of estimating missing coordinates is better than the *adhoc* approaches. However, as the number of missing coordinates increases, the accuracy of the estimation decreases [31, 32].

3.4 Algorithm Considerations

3.4.1 Local Linearity

An assumption that is made by using a PCA approach is that the geodesic distances between the points are maintained. In other words, points that are close in the original space are close to each other in the lower dimensional space. However with globally linear techniques like PCA this is not always true, especially when the data is nonlinear.

This can be addressed to some extent by adopting a locally linear perspective to the application of dimensionality reduction described above. Many nonlinear techniques are developed in a way to exploit the notion of local linearity. In these algorithms, a local linear neighbourhood is constructing using the k nearest neighbouring points for any given point. The manifold or subspace is pieced together using the locally linear manifolds described by the local neighbourhoods.

Likewise, instead of projecting the rigid structure onto the space determined by using all the conformations in the flexibility data set, a subspace is constructed using only the k neighbouring conformations from the data set to the rigid conformation. This not only reduces the size of the data but also attempts to focus on the conformations in a local neighbourhood closest to the rigid conformation.

3.4.2 Infeasible Conformations

Following the procedure in face recognition systems, a transformed conformation that is beyond a certain threshold distance away from the center of the flexibility space can be considered highly dissimilar to the flexible protein. These conformations usually have high energy values corresponding to highly unlikely conformations, as depicted in Figure 3.3. In situations like these, a new flexed conformation must be found with a better energy value.

3.4.2.1 Molecular Dynamics Energy Minimization

A method for finding a minimal energy structure is to conduct a short molecular dynamics energy minimization simulation on the flexed structure. The purpose of energy minimization is to refine a protein structure so that the potential energy is minimized. Although potential energy surfaces may consist of a number of local minima, energy minimization only aims to find a local minimum that is closest to an initial structure. In our case, the initial structure is the flexed structure and a local minimum will be another conformation that is similar in structure but with a minimized energy. This is achieved by repairing any distorted geometries by moving atoms to release any internal constraints. For example if there are any bonds in the flexed structure that have been stretched out, this will increase the structure's potential energy. A molecular dynamics minimization will find an appropriate position of atoms to relax the bond to result in a lower energy structure.

3.4.3 Intrinsic Dimensionality

The value of the dimension of the lower dimensional space must be estimated. This problem can be stated as finding the number of degrees of freedom needed to satisfactorily generate the patterns observed in n dimensional space. Using too many or too few components will result in the data being approximated incorrectly. There are different practices to select how many components to use. The most common approaches involve creating a scree plot and doing one of the following:

- Eliminating components whose singular values are lower than a fraction of the mean eigenvalue
- Keeping the number of components required to approximate the original data with minimal error
- Finding the number of components where the curve in the scree plot changes slope [33]

The second method is the method used in this thesis.

3.4.4 Sample Size

The number of conformations m to be included in the data set is another point of consideration. Over the course of the molecular dynamics simulation, conformations are sampled to produce the data set that will be analyzed. If this simulation is not run for a sufficiently long time period, the data will not contain enough variability to consist of a representative sampling of the protein's motion. This affects the quality of analysis that the application of PCA can provide. If the flexibility of the motion is known in advance, the simulation may be run for a period of time and sampling could be performed at intervals to ensure that the particular motion is observed in the data set. In this thesis, the flexibility of the protein is assumed to be unknown. As a result, the duration and sampling frequency used is similar to the procedures used in the work by Teodoro et al. [5]. The main difference is that, the sampling in this thesis was performed for longer periods of time and sampled less frequently. The overall data set generated is smaller. However, it contains more variability between conformations.

3.5 Chapter Summary

This chapter has outlined the problem of flexible protein structural alignment and stated that typical rigid alignment algorithms do not provide accurate results because they do not model flexibility in protein structures. A new algorithm called FlexSADRA has been introduced as a way to address this problem. The algorithm uses a linear dimensionality reduction technique called principal component analysis to decompose high dimensional protein flexibility motion into a significantly smaller set of principal components. These principal components can be linearly combined to represent the original flexibility motion reasonably well. The structural alignment is determined by performing a simple projection operation. This is in contrast to existing flexible structural alignment algorithms, which perform many computations using high dimensional atomic coordinates. The FlexSADRA algorithm also scores the final alignment by calculating the RMSD value and the energy value of the flexed structure. The additional energy score indicates whether or not the flexed structure represents a feasible conformation.

Few researchers have developed algorithms to address the flexible structural alignment problem. None have tried to address it using a dimensionality reduction approach. As a result, the FlexSADRA algorithm is a novel algorithm that presents an entirely different perspective on attacking the flexible structural alignment problem.

Chapter 4

Results and Discussion

This chapter describes a set of tests that were run to demonstrate the performance of the FlexSADRA algorithm introduced in the previous chapter. In each test, two protein molecules are structurally aligned with one being flexible and the other rigid. In most cases, the flexible and rigid proteins used in the tests have been previously aligned in the literature [20]. Previous alignment algorithms focused on flexible proteins that displayed hinge motions. Since proteins also exhibit shear motions, one of the test cases involves a flexible molecule that is known to exhibit shear movements [34, 35]. The rigid proteins in this test case were chosen by searching the Protein Data Bank for structures in the same super family as the flexible molecule. The goal of the tests is to demonstrate that the FlexSADRA aligns two three dimensional protein structures in a reduced dimensionality problem space and yields structurally feasible results. The data and methods used to test the algorithm will be described and the results will be discussed.

4.1 Test Data

4.1.1 Data Representation

The Protein Data Bank (PDB) is a repository for the online processing and distribution of three dimensional biological molecular structural data [36]. It contains several thousand structures, represented in a standardized format. Users can search the repository for information about structures and download the structures in the form of a file. The file contains the atomic coordinates of at least the backbone (N, C, Cα, O) atoms in the molecule [1].

The majority of the structures in the data bank are proteins, protein-nucleic acid complexes, peptides and viruses. The structures are generated through either X-ray crystallography or nuclear magnetic resonance techniques and submitted to the bank by users.

In some cases, the structures in the data bank are missing atomic coordinates. This may be due to a variety of reasons, but the most common reason is because the structure was generated using X-ray crystallography and position data was not obtained for all atoms. In such cases, a tool called psfgen can be used to estimate the missing coordinate values based on parameters in a force field file.

The protein structures on which simulations are carried out are obtained and used as provided from the Protein Data Bank.

4.1.2 Data Generation

NAMD (Not Another Molecular Dynamics) [37] is a parallel molecular dynamics simulation software application created by researchers at the University of Illinois at Urbana-Champaign. This application is used to generate the conformational data used for the tests. In order for the NAMD simulation to run properly, the following are provided:

[1] Subsequent use of the term backbone in this chapter refer to the N, C, Cα, O atoms of the protein structure

- PDB file containing the protein's atomic coordinates
- PSF file containing the protein's structural information
- Forcefield parameter file containing values used to calculate the protein's energy
- Configuration file specifying values for the simulation environment (i.e. pressure, temperature) and values representing how long and how frequently to sample the conformations

The first step of a NAMD simulation is to minimize the energy of the protein structure. This is usually recommended if not required since the protein structure may not be in a minimal energy state. The most common cause for this is because missing atomic coordinates were estimated poorly resulting in steric clashes between atoms. A minimization stage in the simulation will correct any problematic atomic coordinates. After this stage, NAMD will search the protein's potential energy space to produce only conformations that are minimal in energy. In this thesis, the simulations were run with the protein molecule solvated in a TIP3 water model environment [38] at 310 K for 4.5 nanoseconds after an initial 4000 femtoseconds period of energy minimization. The structures were then sampled periodically from the simulation output every 6 picoseconds. In comparison to the simulations run by Teodoro et al.[5], the simulations in this thesis result in a longer simulation with less frequent sampling. The goal was to create a smaller data set, but with more variability between conformations.

4.1.3 Data Pre-processing

Once the data has been generated, the result is a set of vectors that make up a matrix of conformations simulated over time. This matrix is pre-processed before any dimensionality reduction is applied. The following operations are performed on the data set:

4.1.3.1 Backbone Atoms

In the research performed by Amadei et al., it was found that applying dimensionality reduction techniques to only the backbone Cα atoms of the protein structure produced results that were comparable to using all the protein's atoms [39]. In other words, a protein's backbone Cα atoms can represent most of its flexibility. An advantage of this is that a large number of degrees of freedom are eliminated by only considering the backbone Cα atoms.

In this thesis, the applications used to calculate energy and perform structure minimization require the structure to have at least the backbone atoms of the structure available. As a result, the three dimensional atomic coordinates of the backbone atoms, are included in the representation of the structures used in the algorithm. Each structure will be a vector of $3 \times 4 \times N$ coordinates values, where N is the number of residues in the chain[2]. For example, if the molecule has 133 residues, the molecule will be represented by $3 \times 4 \times 133 = 1596$ coordinate values or degrees of freedom. The final data set will consist of an $n \times m$ matrix, where $n = 3 \times 4 \times N$.

4.1.3.2 Rotations and Translations

Since the focus of this work is on protein flexibility, any overall structural rotational or translational movements are removed from the data set following the procedure used by Teodoro et al. and Amadei et al. [5, 39]. The result is that the data will only depict the flexing degrees of freedom of the protein. Finding the mean structure in the data set and then rigidly aligning each structure to the mean structure accomplishes this.

4.1.3.3 Standardization

Standardizing the data is a common pre-processing procedure that is performed to ensure that each observation in the data set is treated equally with respect to the other obser-

[2] Recall that four atoms N, C, Cα, O are at each residue position in the backbone

vations in the data set. Although there are several different ways to accomplish this, the method chosen was to make the conformational data set have a mean of zero and a standard deviation of one. The calculation is:

$$\tilde{X}_i = \frac{X_i - \mu}{\sigma} \tag{4.1}$$

where μ and σ are the mean and standard deviation of the rows of the data set X respectively and X_i is the ith conformation in the data set X.

4.2 Test Candidates

The structures chosen for the tests are well-studied molecules that have been previously aligned or cited in the literature. In particular, the majority of cases have been selected from the Gerstein Molecular Movement Database [35, 34]. The database consists of proteins that are known to undergo predominantly hinge or shear motions. The movements are classified according to domain, fragment and subunit hinge or shear motions. Hinge motions exhibit larger motions compared to shear motions. Therefore the majority of molecules chosen exhibit hinge motions. The following is a list of protein pairs used in the tests.

4.2.1 Predominantly Hinge Motion

4.2.1.1 Human Calmodulin Open / Drosophila Calmodulin Bound

Calcium is used by the human body in small amounts in the form of calcium ions. These ions are used in a variety of biological activities such as: cell signalling, nerve signalling controlling processes such as muscle contraction, fertilization and cell division. This is achieved through the help of calcium pumps and other calcium binding proteins. When calcium is released into a cell, calcium ions can interact with these calcium-sensing proteins and activate a number of biological effects such as causing a muscle to contract, or blocking the entry of additional sperm cells once an egg has been fertilized.

Calmodulin (CALcium MODULated proteIN) acts as an intermediary protein that detects the levels of calcium in a cell and sends signals to various calcium-sensitive enzymes, ion channels and other proteins. The structure of the molecule is composed of two globular domains attached by a flexible linker region. Each end binds to two calcium ions. The flexibility of the protein is due to the linker region, which allows the molecule to wrap around its target protein, with the two globular domains gripping either side of it.

Two Calmodulin molecules from different species were structurally aligned. The flexible molecule was the unbound or open conformation of the Human version of Calmodulin. The rigid Calmodulin molecule from the Drosophila species was in its Calcium-bound state. It is often the case that two otherwise structural similar protein molecules are simply hinge bent with respect to each other. This test was designed to see if the algorithm could determine if the two structures were structurally similar despite the fact that they are from two different species and one is the bound version of the other.

4.2.1.2 Homeodomain Protein / Paired Domain Protein

Aniridia is a human genetic disease that is responsible for mutations in the structure and function of the eye. A person with this disease may have reduced iris size, absence of the fovea and lens deformities. Although the disease was first documented over 150 years ago, it was only recently that the aniridia gene was determined to be the Paired Axial 6 (PAX6) gene. This gene is responsible for the regulation of the development of the eyes and central nervous system. The effect of PAX6 mutations can range from loss of visual acuity to the complete absence of eyes.

The PAX6 gene is not only involved in human eye development, but has been described as a master control gene that indirectly or directly acts upon any other genes involved in eye development in other species. This hypothesis was recently supported by the finding that ectopic expression of PAX6 in Drosophila could induce eye formation on appendages [40]. It was later discovered that genes downstream of PAX6 could also induce ectopic eyes, suggesting that the gene operates not in a hierarchical linear pathway, but as a network with numerous feedback loops.

The homeobox is a conserved sequence motif of about 180 base pairs long and encodes a protein domain called the homeodomain that can bind DNA. The 6PAX[3] protein is a member of the homeodomain family of proteins. These proteins are transcription regulators that bind to specific DNA sequences of other genes to regulate their expression and induce cell development and differentiation. The regulation of transcription of the PAX6 gene induces the development of eyes and nervous system.

The paired box is another conserved sequence motif that was first identified in the paired and gooseberry Drosophila domain genes. This gene encodes a 128-amino-acid domain, the paired domain, which has since been found in other species. These proteins are homologous to the Human 6PAX homeodomain proteins.

A flexible Human 6PAX homeodomain protein was structurally aligned with a rigid Drosophila paired domain protein. Both proteins consist of two ends made up of alpha-helices connected by a flexible linker region. The flexible hinge motion in the linker region is what allows the protein to bind a DNA molecule.

4.2.1.3 Immunoglobulin Fab Elbow Joint / T-Cell Receptor

Antibodies exist in the bloodstream or are attached to cell membranes. They are used by the immune system to identify and attack foreign bodies like bacteria or viruses. Antibodies bind specifically to one antigen, block viral receptions and induce other immune responses. For instance, some antibodies that recognize viruses can prevent them from docking to and infecting a cell just by their sheer size. Others that recognize bacteria can mark them for

[3] 6PAX refers to the name of the PAX protein, while PAX6 refers to the name of the PAX gene

ingestion by a macrophage or neutralize them by binding with them.

Immunoglobulin molecules are glycoproteins in the immunoglobulin super family that function as antibodies. The basic structure of this protein is a monomer that consists of two identical heavy chains and two identical light chains. They combine to form a Y shaped molecule. The heavy chains have one variable and three constant domains. The variable region is connected to the constant regions by a short section called the switch. The light chains have one variable and one constant domain connected by another switch section. The constant domains are the same across all immunoglobulin molecules from the same class, while the variables domains are different.

The heavy chain is further divided into two sections called the Fc and Fab fragments, which are connected by a hinge area. The Fab fragments are the forked ends of the Y part of the molecule, while the Fc fragment is the stem part of the Y part of the molecule. Figure 4.1 is an example of an immunoglobulin molecule and Figure 4.2 is a simplified representation of the molecule showing the location of the light chains, heavy chains and the Fab and Fc fragments.

The immunoglobulin molecule exhibits an elbow-joint like motion in each of its variable-switch-constant domains. The elbow part of the motion is represented by the connecting switch region. The joint motion is a predominantly hinge like motion with small shear movements.

T-cells are like antibodies, however they patrol the body looking for infected cells. T-cell receptors are located on the surface of the T-cell and bind tightly with viral peptides that have infected a cell. Like antibodies, T-cell receptors bind to a specific viral peptide. The immune system creates a variety of T-cells in order to protect against a variety of viruses. The importance of T-cells becomes obvious in the case of the HIV. If left untreated, HIV attacks the immune system depleting it of T-cells, the very system that protects the body from viruses. The infected individual progresses into AIDS when the number of T-cells

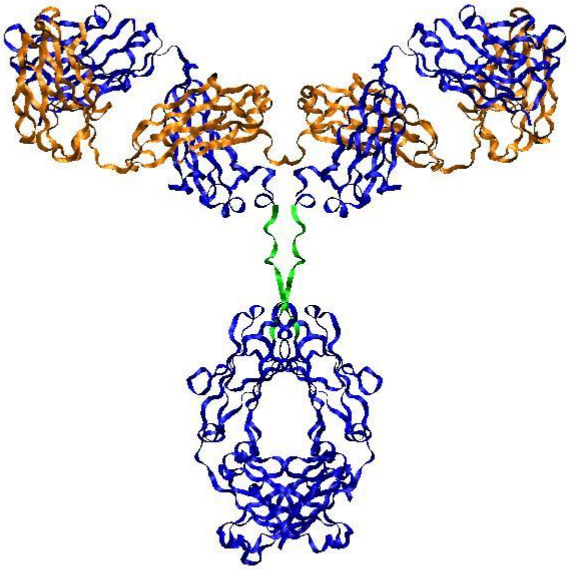
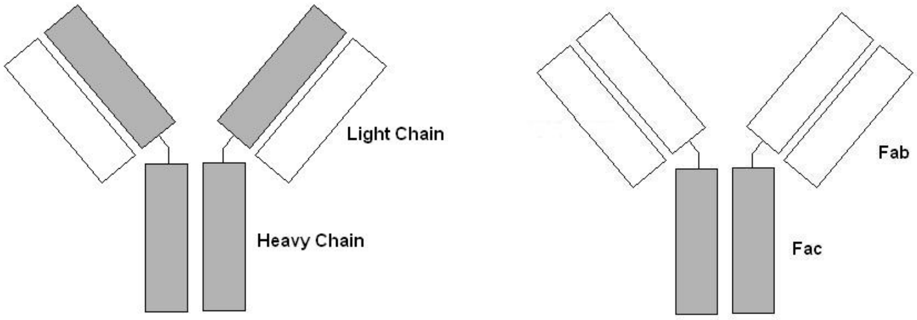


Figure 4.1: Example of a human immunoglobulin molecule

becomes too low. In this state, the individual is vulnerable to infection since the immune system is too weak to protect against foreign invaders.

T-cell receptors are composed of two chains and are similar to one arm of the basic Y shaped antibody. A flexible Fab elbow joint from the light chain of the Mus musculus immunoglobulin molecule was aligned with a chain from the Mus musculus T-cell receptor. The T-cell receptor and the immunoglobulin protein are different molecules, but they are both used by the immune system to perform similar functions. This test was designed to see if a flexible structural alignment could support the fact that these two molecules are functionally very similar despite the great differences in their amino acid sequences.



(a) Light and Heavy Chains

(b) Fab and Fac Fragments

Figure 4.2: Line representation of an example of a human immunoglobulin molecule

4.2.2 Predominantly Shear Motion

4.2.2.1 Apo Calbindin / EF Hand Domain Proteins

Calbindin is a calcium binding protein like Calmodulin. Specifically, it is involved in the uptake of calcium in the fetus, calcification of bones and teeth, and transportation and absorption of calcium in the intestines.

This protein and Calmodulin both belong to the same super family known as the EF hand super family. All the molecules in this super family are homologous since they evolved from the same ancestor. Each molecule also contains two to twelve EF hand domains. The EF hand is approximately thirty amino acids and consists of an alpha-helix (E), loop, and second alpha-helix (F). Calbindin has two EF hand domains that are responsible for Calcium binding and a short linker region. Small shear movements in the helices and loops exhibit the flexibility of this molecule.

In order to compare the structural similarity of molecules within the same super family, a flexible Calbindin molecule from the Bos Taurus species was structurally aligned with a set of rigid molecules from the EF hand super family. The following two molecules were chosen:

- Parvalbumin - This molecule is involved in the termination of muscle contractions by the absorption of calcium.
- α-Actinin - This molecule is involved in sacromere assembly and binds to the actinin molecule.

A summary of the details of the test data and test candidates can be found in Table 4.1.

4.3 Test Procedure

The test procedure is carried out by doing the following steps:

- Obtain the flexible and rigid protein structural information from the Protein Data Bank
- Simulate the flexible protein using the NAMD simulation software
- Pre-process the data
- Apply the FlexSADRA algorithm
- (Optional) Minimize the final flexed structure using the NAMD simulation software

4.4 Comparison with Rigid Structural Alignment

The results of the tests indicate that the FlexSADRA algorithm is able to perform flexible structural alignment in a reduced dimensionality problem space. The criteria used to evaluate the success of the algorithm were the amount of reduction in the number of degrees of freedom, RMSD of the proteins aligned, energy value of the flexible protein, and the consistency of the motion in the flexible protein used in the alignment with the known motion of the protein.

Table 4.1: Summary of test data and test candidates

R/F	Species	PDB Code	Chain	# Residues	DOF	# Conf.
Flexible	Homo Sapiens	1CLL		144	1728	1000
Rigid	Drosophila	2BBM	A	148	1776	1
Flexible	Homo Sapiens	6PAX	A	133	1596	1333
Rigid	Drosophila	1PDN	C	128	1536	1
Flexible	Mus musculus	1MCP	L	220	2640	2319
Rigid	Mus musculus	1TCR	B	237	2844	1
Flexible	Bos Taurus	1CDN		75	900	1000
Rigid	Rattus Rattus	1FI6		88	1056	1
Rigid	Homo Sapiens	1H8B	A	73	876	1

4.4.1 Reduction of Degrees of Freedom

In general, the amount of reduction in the dimensionality of the data was 50 to 70 percent. In many cases, the number of dimensions went from a few thousand to only a few hundred dimensions. The most drastic reduction is in the case of 1CDN and 1H8B where the number of degrees of freedom went from 225 atomic coordinates to only 71 principal components. Table 4.2 lists the reduction in the number of degrees of freedom for each test case.

4.4.2 Structural Fit

Using the smaller dimensional flexibility data set, the flexible structural alignment results in a significant improvement in RMSD, with differences ranging from 5 to 10 Å. The rigid alignments in all the cases result in very high RMSD values with 1CLL having the highest value of 16.28 Å. Without the added flexibility, the structures would seem to be very dissimilar. However, after allowing flexibility in the alignment the RMSD values decrease significantly. In all the cases, the RMSD decreases to less than 2 Å, which indicates that the molecules are actually structurally very similar. The only exception is the 1CLL molecule, whose RMSD value was 6.61 Å. This emphasizes the fact that the FlexSADRA algorithm is a data driven approach and as a result the degree of flexibility is limited to what is represented in the data. The 1CLL molecule undergoes a large conformational change from its open to bound form. Molecular dynamics simulations must be run for an enormously long period of time for large conformational changes to take place. As a result, it is likely that the flexibility data for 1CLL does not contain enough variability to align the two molecules closer than 6.61 Å.

In Figure 4.9 (pg. 69), the flexible connector section bends in order for the 1CLL molecule to match the closed conformation structure of 2BBM. Figure 4.10 (pg. 70) shows the 6PAX molecule's flexible linker section enabling its poorly aligned alpha-helix domain to align better with the corresponding alpha-helix domain in 1PDN. Figure 4.11 (pg. 70) shows that the beta sheet domains in the 1MCP molecule are able to come closer to the 1TCP molecule through a hinge motion about its flexible linker region. In Figures 4.12, 4.13 (pg. 71) the 1CDN molecule exhibits slight movements among the loop and alpha helices to allow a better alignment with 1H8B and 1FI6.

4.4.3 Structural Feasibility

In almost all cases, the energy values for the flexed protein were high with respect to the original energy. This would indicate that the flexed structure does not represent a likely conformation. In order to determine if the flexed structure represented an actual conformation, a short minimization simulation was run on the structure. After minimization, the RMSD was compared with the pre-minimized flexed structure. The difference is on average about 0.7 Å, which indicates that the pre-minimized and minimized structures are almost identical. This can be seen graphically in Figures 4.9, 4.10, 4.11, 4.12 and 4.13 (pgs. 69-71). This indicates that the overall flexed structure is sound since only a few structural changes need to be made to yield a minimized structure. The offending areas may be a result of the fact that there is a certain amount of information lost when using dimensionality reduction techniques. Another factor could be that missing coordinate values in the alignment must be estimated. Poorly estimated coordinate values will increase the energy value.

It can also be case that the flexed structure has a low energy value as in the case of 1CDN and 1H8B. Since the structure already had an acceptable energy value, it was not minimized further.

4.4.4 Consistency with Known Motion

The results also show that the flexibility motion assumed in the alignments is consistent with the motion exhibited in the original conformational data set generated from simulations. The hinge bending motions are seen in alignments for the three hinge bending molecules. For example, the two domains of the 1MCP molecule move together about a hinge in order to come closer to the 1TCR molecule. Shear movements are seen in the alignments for the shear moving molecule. For example, the 1CDN molecule exhibits slight movements among the loop and alpha helices.

The first principal component motion was plotted and compared in order to determine how well it represented the original motion of the protein. All of the plots show that even with only one dimension, the protein motion is represented with reasonable accuracy. The principal component motion calculated for 1CLL and plotted in Figure 4.5 (pg. 66) show that the principal components on the two end domains are larger than the ones in the connecting linker section. This is consistent with the fact that 1CLL undergoes a conformational change involving the two end domains moving a large distance in order to come closer together. The hinge bending motion is also captured by the first principal component of the 1PAX flexibility data in Figure 4.6 (pg. 66) and also for 1MCP in Figure 4.7 (pg. 67). All of the hinge bending molecules consist of domains on either ends of the molecule that are connected by a flexible linker section. The figures show that hinge bending protein motion can be represented using a small number of principal components rather than the conventional high dimensional atomic coordinates. Shear-like movements are also depicted by the first principal component motion for 1CDN. This is shown in Figure 4.8 (pg. 67). These results are also consistent with past research performed in modeling protein flexibility motion using PCA [39, 5].

In general, all the results show a large improvement in structural alignment when flexibility is allowed. The Table 4.2 (pg. 68) summarizes the results presented in this section.

4.4.5 Runtime Complexity

The runtime complexity of a rigid structural alignment that uses an SVD implementation is $O(n^3)$ to perform the SVD operation. This is comparable to the runtime complexity for FlexSADRA, which is discussed in more detail in the next section.

4.5 Comparison with FlexProt

The FlexSADRA algorithm was also compared to another flexible structural alignment algorithm called FlexProt. FlexProt structurally aligns two protein molecules where one is flexible and the other is rigid. The alignment uses the backbone Cα atomic coordinates and decomposes the alignment into a set of rigid fragments. The areas between the rigid fragments are the flexible hinge areas. The same molecules aligned in the rigid structural alignment tests were aligned using FlexProt and the results were compared.

FlexSADRA and FlexProt are very different even though they are both flexible structural alignment algorithms. The biggest difference is that FlexProt performs the structural alignment using the original high dimensional atomic coordinates. The algorithm is not trivial and the alignment consists of a series of steps each involving a number of computations.

FlexProt uses the backbone Cα atom coordinates as input to the algorithm. This is different from the use of the backbone N, C, Cα, O atoms by FlexSADRA. As mentioned previously, the use of four times the number of coordinate values is due to the applications used to calculate the energy values and perform the structure minimization. Despite using a larger number of degrees of freedom to represent the protein structure, the FlexSADRA algorithm is still able to perform the alignment with fewer degrees of freedom than FlexProt.

Overall, the RMSD alignment values are in agreement with the protein pair 1CLL-2BBM as the only exception. As mentioned previously, the FlexSADRA algorithm incorporates as much flexibility into the alignment as dictated by the flexibility data. This is contrasted by FlexProt's more heuristic approach. Although FlexProt is able to determine closer alignments, these alignments are not based on any kind of supporting evidence. If the true flexibility of the 1CLL molecule was actually represented by the simulated flexibility data, the FlexProt alignment would not be accurate because 1CLL would not actually be able to bend in the way FlexProt would require.

The output of FlexProt is a set of disconnected rigid fragments of the flexed protein. It is not clear if this structure represents a structurally feasible conformation. An energy value analysis should be performed on the flexed structures, however it is not possible to do so with most energy calculators since they require connected structures as input. Also, the atomic coordinates for the hinge areas are not provided. In many cases, hinge-like or connecting areas are flexible loops and are still important parts of the molecule. Another problem is that molecules do not move as disconnected rigid fragments. As a result, FlexProt only provides a disjoint partial view of the flexed protein, whereas the FlexSADRA algorithm provides a complete picture of the flexed protein. Flexed areas are not modeled as isolated regions joining rigid fragments, but apply to the molecule as a whole.

Figures 4.3, 4.4 (pg. 63) are examples of the results of the final structural arrangements obtained by FlexSADRA and FlexProt. The results of comparing FlexProt and FlexSADRA are summarized in Table 4.3 (pg. 69).

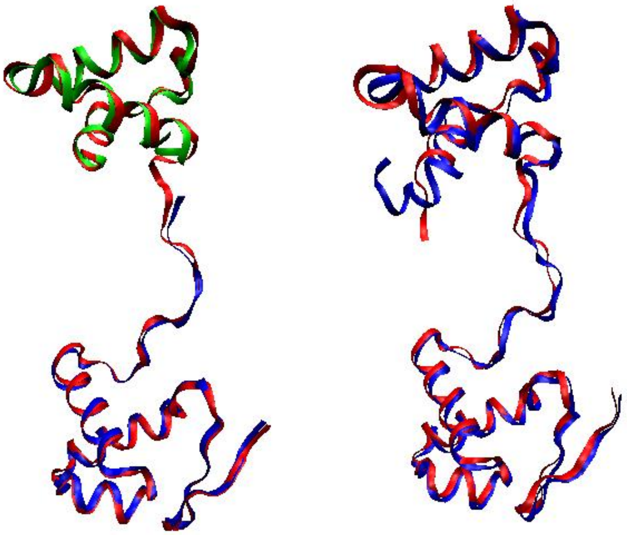


Figure 4.3: FlexProt versus FlexSADRA flexible structural alignment (1PDN and 6PAX) Left - FlexProt flexible structural alignment of 1PDN (red) and 6PAX (blue and green). The chains represent rigid fragments and are not attached. The area between the blue and green fragments represent a flexible hinge area. Right - FlexSADRA flexible structural alignment of 1PDN (red) and 6PAX (blue).

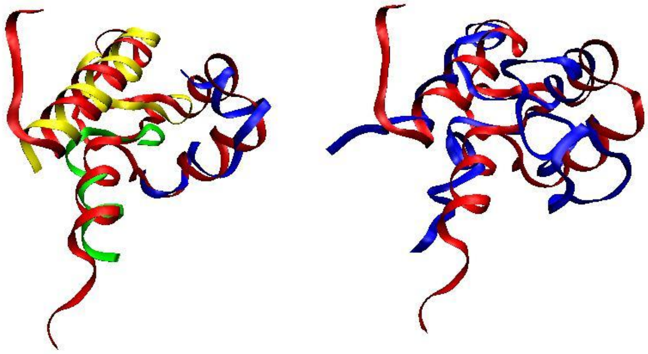


Figure 4.4: FlexProt versus FlexSADRA flexible structural alignment (1H8B and 1CDN) Left - FlexProt flexible structural alignment of 1H8B (red) and 1CDN (yellow, green and blue). The three chains represent rigid fragments and are not attached. The areas between the chains represent flexible hinge areas. Right - FlexSADRA flexible structural alignment of 1H8B (red) and 1CDN (blue). Note that there is a lot of structural information missing in the FlexProt structure.

4.5.1 Runtime Complexity

The complexity of FlexProt is $O(n^4)$ and the alignment is performed by a number of non-trivial steps [20]. In contrast, FlexSADRA performs the alignment in a single step by a simple projection operation. Assuming $n \geq m$, and the rigid and flexible proteins to be aligned are the same length, the complexity of the algorithm is as follows:

- $O(n^3)$ Calculate the principal components (using SVD [41])
- $O(dn)$ Perform the alignment in low dimensions (using projection operation)
- $O(n^3)$ Transform the projection to high dimensions (using matrix inversion [42])

The worse case complexity is therefore: $O(n^3) + O(dn) + O(n^3) = O(n^3)$. The bulk of the computation is in calculating the principal components and performing the matrix inversion. However, principal components can be calculated once, stored and reused for future alignments. Also, there are other methods for computing principal components. For example, the Expectation-Maximization PCA (EMPCA) algorithm developed by Roweis [41] has a runtime complexity of $O(dn^2)$, where d is the number of eigenvectors to calculate.

If the rigid and flexible protein molecules to be aligned are not the same length, then a rigid structural alignment is performed to obtain an initial alignment. In this case, the complexity of the algorithm must include the runtime complexity for this step. There are numerous rigid structural alignment algorithms available and most execute in a matter of seconds depending on the size of the protein molecules to be aligned.

The minimization process of the algorithm is performed using NAMD after the main part of the algorithm is executed. NAMD can be run as a sequential application but usually is run on massively parallel super computers. The minimization of the protein structures used in these tests ran in approximately 30 to 40 seconds.

4.6 Chapter Summary

The FlexSADRA algorithm presented in this thesis was tested on a variety of protein molecules that have been previously studied in the literature. The purpose of the tests was to show that FlexSADRA is able to perform structural alignment in a reduced dimensionality problem space. The results indicate that the algorithm calculates significantly better structural alignments than rigid structural alignment algorithms. In particular, proper alignments for structures are that are simply flexed or bent with respect to each other, or from different species but have similar function, or evolved over time from the same ancestor were obtained. The results also show that the FlexSADRA algorithm produces flexible structural alignments that are comparable to FlexProt. More importantly, the results show that a dimensionality reduction approach to the structural alignment problem is possible.

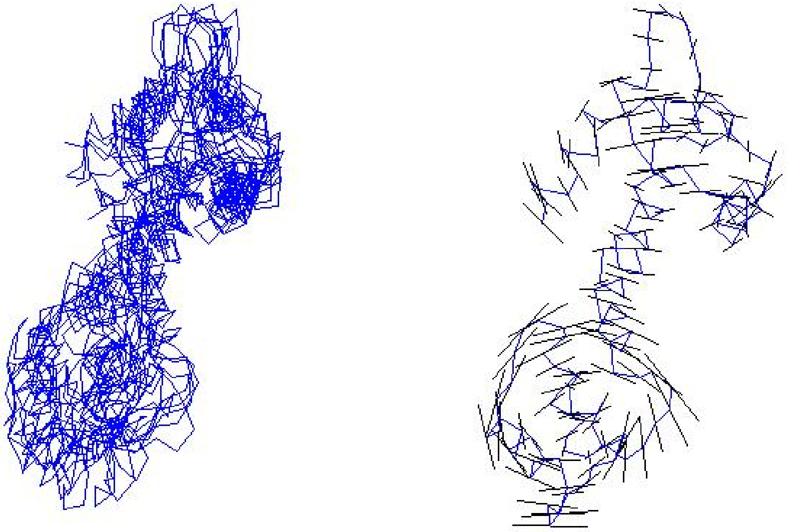


Figure 4.5: 1-D PC representation of high-dimensional 1CLL flexibility motion data Left - A small set of conformations sampled from the data set generated by NAMD. Right - The lines across the atoms positions along the backbone are the 1-D PC representation of the high dimension motion on the left.

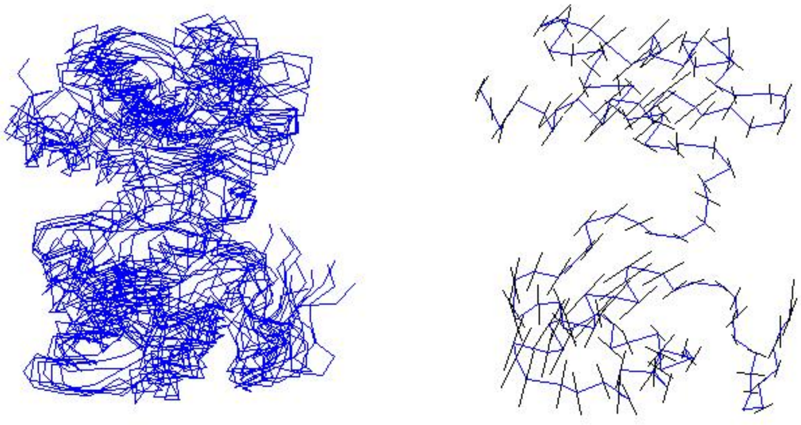


Figure 4.6: 1-D PC representation of high dimensional 6PAX flexibility motion data Left - A small set of conformations sampled from the data set generated by NAMD. Right - The lines across the atoms positions along the backbone are the 1-D PC representation of the high dimension motion on the left.

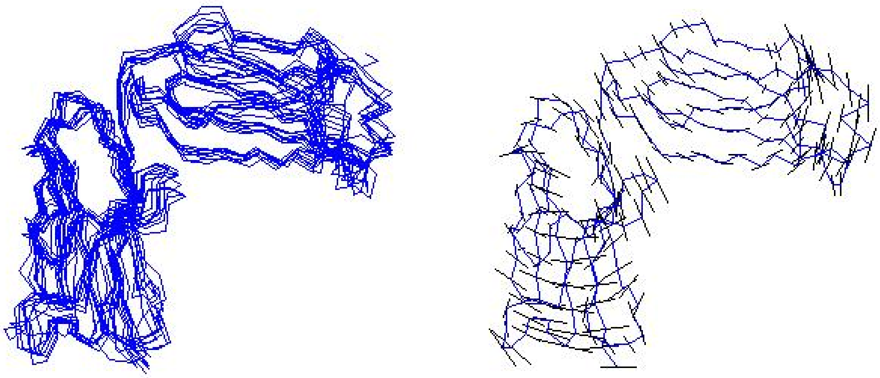


Figure 4.7: 1-D PC representation of high dimensional 1MCP flexibility motion data Left - A small set of conformations sampled from the data set generated by NAMD. Right - The lines across the atoms positions along the backbone are the 1-D PC representation of the high dimension motion on the left.

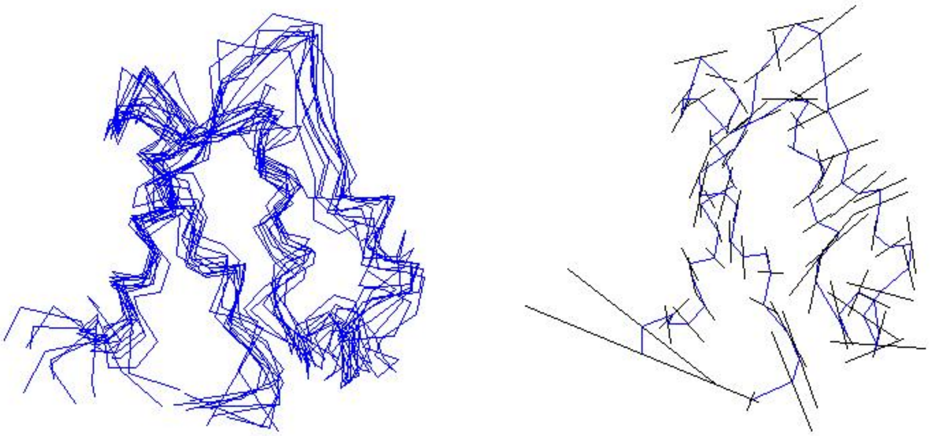


Figure 4.8: 1-D PC representation of high dimensional 1CDN flexibility motion data Left - A small set of conformations sampled from the data set generated by NAMD. Right - The lines across the atoms positions along the backbone are the 1-D PC representation of the high dimension motion on the left.

Table 4.2: Results of Rigid, FlexSADRA and Minimization structural alignments

1CLL-2BBM

Algorithm	**Energy**	**RMSD**	**DOF (% Retained)**	**Alignment Length**
Rigid	36.06	16.28	432	89
FlexSADRA	66.14	7.20	84 (70)	
	54.37	7.03	119 (75)	
	73.63	6.73	167 (80)	
	136.31	6.61	237 (85)	
Min	34.80	6.67		

6PAX-1PDN

Algorithm	**Energy**	**RMSD**	**DOF (% Retained)**	**Alignment Length**
Rigid	43.24	9.70	399	123
FlexSADRA	295.82	2.16	72 (70)	
	342.84	1.94	107 (75)	
	334.11	1.84	157 (80)	
	297.52	1.66	232 (85)	
Min	41.61	1.64		

1MCP-1TCR

Algorithm	**Energy**	**RMSD**	**DOF (% Retained)**	**Alignment Length**
Rigid	46.52	7.64	660	192
FlexSADRA	754.86	3.33	290 (70)	
	633.20	3.16	367 (75)	
	660.01	3.03	466 (80)	
	605.52	2.86	599 (85)	
Min	46.62	3.35		

1CDN-1H8B

Algorithm	**Energy**	**RMSD**	**DOF (% Retained)**	**Alignment Length**
Rigid	42.26	4.11	225	46
FlexSADRA	43.07	2.26	55 (60)	
	39.81	2.06	71 (65)	
	85.90	1.98	92 (70)	
	102.97	1.87	120 (75)	
	138.21	1.78	157 (80)	
	213.29	1.64	208 (85)	
Min				

1CDN-1FI6

Algorithm	**Energy**	**RMSD**	**DOF (% Retained)**	**Alignment Length**
Rigid	42.26	3.72	225	57
FlexSADRA	308.82	2.25	55 (60)	
	187.73	2.14	71 (65)	
	130.30	2.05	92 (70)	
	86.05	1.96	120 (75)	
	150.36	1.78	157 (80)	
	120.93	1.77	208 (85)	
Min	37.42	2.84		

Table 4.3: Results of Rigid, FlexProt and FlexSADRA structural alignments

Test	**# Hinges**	**DOF (% Retained)**		**RMSD**		
Flex-Rigid	FlexProt	FlexProt	FlexSADRA	Rigid	FlexProt	FlexSADRA
1CLL-2BBM	1	432	237 (85)	16.28	2.19	6.61
6PAX-1PDN	1	399	232 (85)	9.70	1.21	1.66
1MCP-1TCR	1	660	599 (85)	7.64	2.55	2.86
1CDN-1FI6	2	225	157 (80)	3.37	2.95	2.06
1CDN-1H8B	2	225	71 (65)	4.11	2.96	1.78

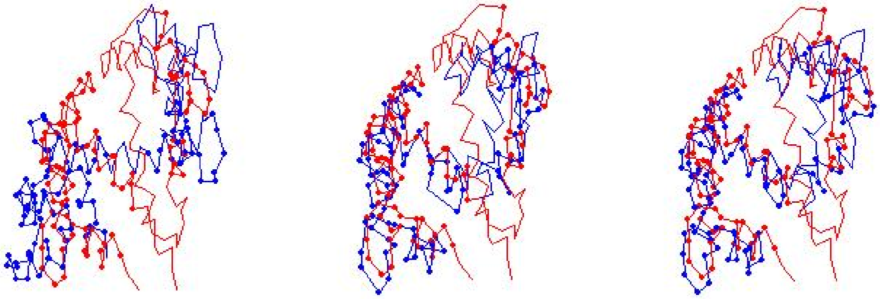


Figure 4.9: FlexSADRA flexible structural alignment of 1CLL and 2BMM
Left - Rigid Alignment. Middle - FlexSADRA Alignment. Right - Minimized Alignment.
2BMM (red) 1CLL (blue).

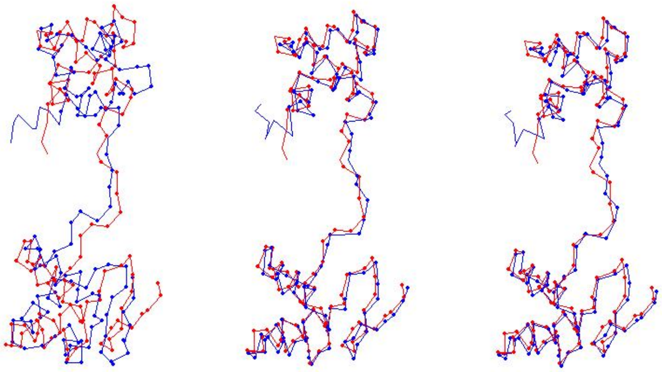


Figure 4.10: FlexSADRA flexible structural alignment of 1PDN and 6PAX
Left - Rigid Alignment. Middle - FlexSADRA Alignment. Right - Minimized Alignment.
1PDN (red) 6PAX (blue).

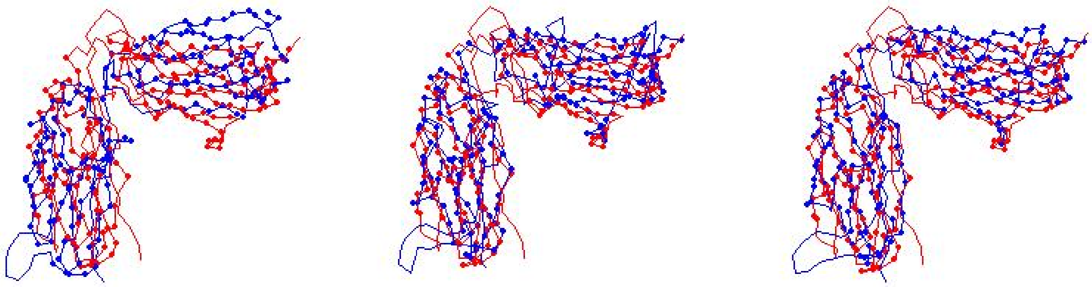


Figure 4.11: FlexSADRA flexible structural alignment of 1MCP and 1TCR
Left - Rigid Alignment. Middle - FlexSADRA Alignment. Right - Minimized Alignment.
1TCR (red) 1MCP (blue).

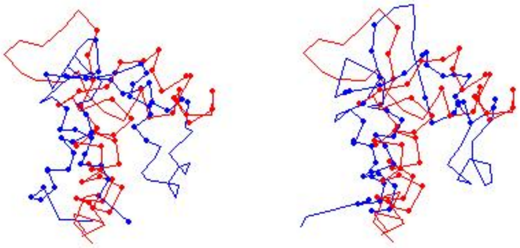


Figure 4.12: FlexSADRA flexible structural alignment of 1CDN and 1H8B
Left - Rigid Alignment. Right - FlexSADRA Alignment. 1H8B (red) 1CDN (blue).

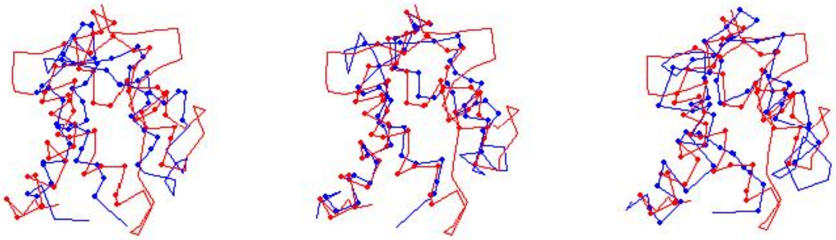


Figure 4.13: FlexSADRA flexible structural alignment of 1CDN and 1FI6
Left - Rigid Alignment. Middle - FlexSADRA Alignment. Right - Minimized Alignment.
1FI6 (red) 1CDN (blue).

Chapter 5

Conclusions

In this thesis an algorithm called FlexSADRA has been designed for the flexible structural alignment of three dimensional protein structures. This algorithm uses a dimensionality reduction approach to model protein flexibility motion and to perform the structural alignment. It has been tested on protein molecules that have previously been structurally aligned or studied in the literature. The results of the tests show that flexible structural alignment can be performed in a reduced dimensionality problem space. The results are better than the results obtained from rigid structural alignments and comparable to the results from another flexible structural alignment algorithm called FlexProt. However in contrast to these algorithms, the FlexSADRA algorithm is simpler and involves significantly fewer degrees of freedom.

5.1 Contributions

The main contribution of this work is its investigation into the viability of addressing a highly complex problem like flexible structural alignment in much lower dimensions. Instead of working with atomic coordinates as done by existing approaches, the FlexSADRA algorithm works with a smaller number of new variables describing the protein motion. In this lower dimensionality problem space, a structural alignment involves performing a simple algebraic projection operation. This is in contrast to all existing structural alignment algorithms, which are intricate, complex and involve many variables. The success of the tests show that FlexSADRA is able to perform flexible structural alignment supporting the contention that well-studied methods like dimensionality reduction techniques can be applied to classic problems such as protein flexible structural alignment.

5.2 Future Research

Extensions of this work are described in the following sections.

5.2.1 Simulated Annealing

The principal components represent flexibility features describing the protein's flexibility motion. When appropriate linear combinations of the components are made, they can be used to construct novel protein conformations. However, these conformations might have high energy values and therefore indicate that the conformation is not structurally feasible. In this case, a molecular dynamics minimization simulation may be run as in the case of this thesis. As an alternative, the space on the low dimensional manifold between the point representing the high energy value and the rest of the low energy data points may be sampled. Since the principal components span the low dimensional space, the sampling involves trying different linear combinations of the principal components within the specified space until one with an acceptable energy value is found. Points closer to the low energy data points will have lower energy values whereas ones farther away will have higher energy values. Sampling in this manner makes use of the principal components, which are readily available and does not rely on the molecular dynamics minimizations, which can be time consuming to configure. The trade off is that a molecular dynamics simulations will produce more accurate results than the simulated annealing approach described.

5.2.2 Nonlinear Dimensionality Reduction

Since protein flexibility motion is nonlinear, nonlinear dimensionality reduction approaches should be applied. A variety of nonlinear dimensionality reduction techniques exist for testing such as Locally Linear Embedding (LLE) [43], Isomap [44], Hessian Eigenmaps [45], Charting [46]. The difficulty with these techniques is that the reverse mapping from low dimensional space back to high dimensional space is not straightforward as it is for a linear technique such as PCA.

5.2.3 Estimating Missing Values

As mentioned earlier, estimating the missing backbone atomic coordinates is done using a simple least squares approach. However, if there are many coordinates missing, this technique does not provide ideal solutions. Instead a more sophisticated approach utilizing an Expectation-Maximization (EM) method may yield more accurate results [31, 32].

5.2.4 Scoring Functions

The conventional way to score a structural alignment is to calculate the RMSD value and structural energy values. These two scores are high dimensional scoring functions because they are calculated using high dimensional structures. In particular, energy calculations are very complex involving many variables and constants. A scoring system similar to the one described in the face recognition section should be investigated. Recall that in the face recognition system, if a new face is projected far from the training points in the face space, the face is considered to be unknown. Similarly, if a structure is projected far away from the data points in the flexibility space, then it may be associated with a high energy conformation and therefore not likely to be structurally feasible. Since this thesis performs the structural alignment in low dimensions, it should be possible to also score the alignments in low dimensional space as well.

Appendix A

Potential Energy

The energy function is composed of individual energy terms that are added together to produce a total energy describing an entire protein conformation X.

$$E_{total} = \sum_{bonds} E_{bonds}(X) + \sum_{angles} E_{angles}(X) + \sum_{torsions} E_{torsions}(X) + \sum_{nonbonded} E_{nonbonded}(X) + \tag{A.1}$$

Although there can be additional terms that can be added to calculate the energy, only the main components are discussed here.

E_{bonds} - This term pertains to the stretching and compressing of the length of bonds. This term can be approximated by a Hooke's Law harmonic function such as:

$$E_{bonds} = K_b(b - b_0)^2 \tag{A.2}$$

where K_b is an experimentally derived constant representing the stiffness of the bond spring, b is the current bond length and b_0 is the bond length at equilibrium. A harmonic function may also be more suitable to represent bond energy, since it takes more energy to compress a bond length than stretch it.

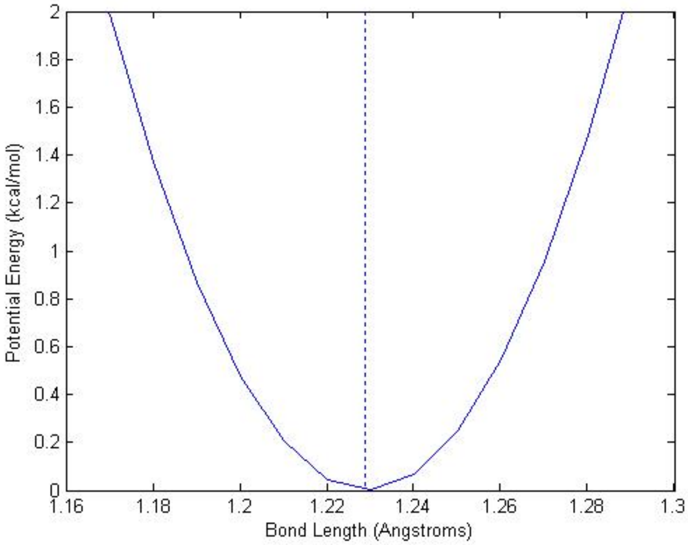


Figure A.1: Example of a bond length energy function.
The energy increase as the bond length moves away from the equilibrium represented by the dotted line.

E_{angles} - This term pertains to the changes in the angles between the bonds. This can be modeled using a Hooke's Law quadratic function:

$$E_{angles} = K_{\theta}(\theta - \theta_0)^2 \tag{A.3}$$

where K_{θ} is an experimentally derived constant, θ is the current bond angle and θ_0 is the bond angle at equilibrium. For a better approximation, this function can also be a Fourier function or a higher order function.

$E_{torsions}$ - This term represents the energy associated with a rotation about a bond. This function can be expressed as a periodic function:

$$E_{torsions} = K_{tor}[1 \pm cos(nw)] \tag{A.4}$$

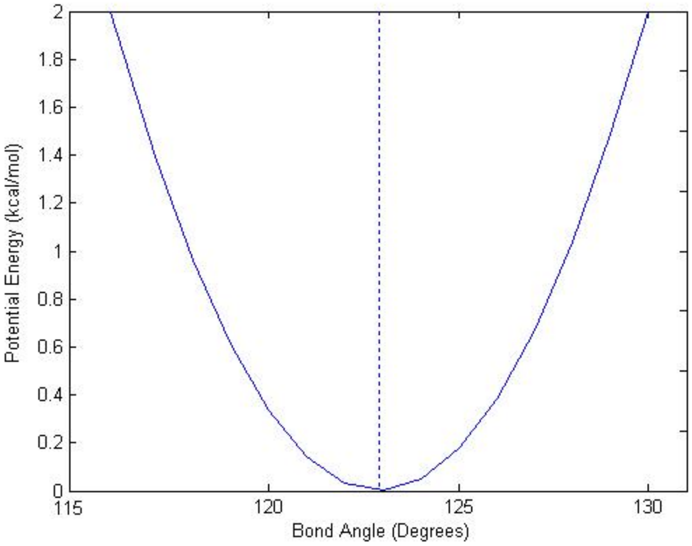


Figure A.2: Example of a bond angle energy function.
The energy increases as the bond angles moves away from the equilibrium angle
represented by the dotted line.

where K_{tor} is a force constant, n is the periodicity and w is the angle.

$E_{nonbonded}$ - These terms include any other energies that pertain to pairs of atoms not related by a bond. These include van der Waals and electrostatic forces.

$$E_{nonbonded} = E_{vdw} + E_{elec} \tag{A.5}$$

van der Waals interactions between non-bonded atoms are strongly repulsive in close proximity, become mildly attractive at intermediate range, and vanish if the atoms are far apart from each other. The van der Waals energy is represented using the Lennard-Jones 12-6 potential:

$$E_{vdw} = K\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{A.6}$$

where K and σ are experimentally derived constants controlling the strength and shape of

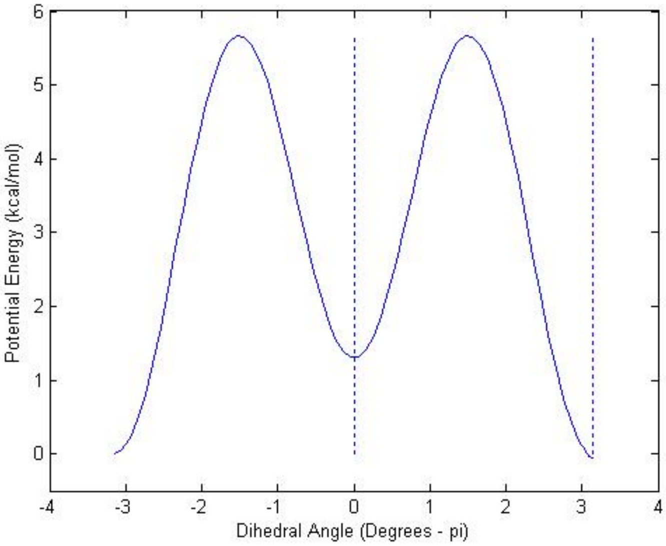


Figure A.3: Example of a torsions energy function.
The energy is the lowest in the cis and trans conformations represented respectively by the middle and right dotted lines. The energy increases as the rotations move away from the cis and trans conformations.

the interaction, and r is the distance between the atom pairs.

Electrostatic forces are the electrical forces that stationary electrically charged atoms will exert on one another. This function is modeled using Coulomb's Law:

$$E_{elec} = K_C \frac{q_i q_j}{r^2} \tag{A.7}$$

where K_C is an electrostatic constant or Coulomb's constant, r is the distance between the atom pairs and q_i, q_j are the electric charges on the atoms.

Further details on the potential energy function are discussed by [47, 48, 49].

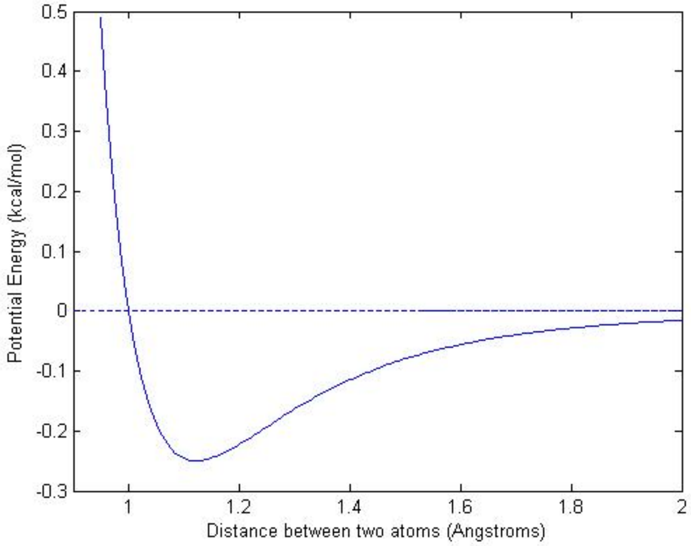


Figure A.4: Example of a Lennard-Jones 12-6 energy function
The energy increases quickly for atoms that are close together, but gradually decreases as the atoms move further apart.

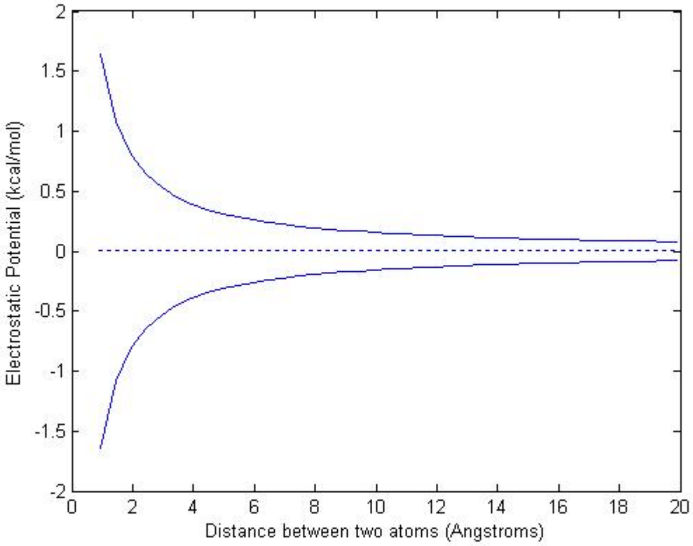


Figure A.5: Example of an electrostatic energy function
The top curve represents the interaction between two atoms of equal charge. The bottom curve represents the interaction between two atoms of opposite charge.

Appendix B

NAMD

NAMD (Not Another Molecular Dynamics) is a parallel, object oriented molecular dynamics simulation software application that was designed for the high performance simulation of biomolecular systems. NAMD uses the Charm++/Converse parallel runtime system to allow scaling on parallel supercomputers and smaller workstations. It is freely available for a wide variety of system platforms.

In addition to the configuration file required to run the simulation, the following files are also required:

- PDB (Protein Data Bank) file: This file contains the atomic coordinates for the protein structure.

- PSF (Protein Structure File): This file contains all of the molecule specific information needed to apply a particular force field to a molecular system.

- Forcefield file: This file contains the constant and other values required to calculate the potential energy of a molecular structure.

- Topology file: This file contains all of the information needed to convert a list of residue names into a complete PSF structure file.

A simple simulation involves a minimization and equilibration run before a long production run is performed. There are several parameters, which must be supplied in a configuration file to NAMD.

The following is an example of the configuration file to run the simulation for 1CDN. Information about the parameters in the file are explained in detail on the NAMD web site [50].

```
#############################################################
## JOB DESCRIPTION                                         ##
#############################################################

# Minimization and Equilibration of
# CDN in a Water Box

#############################################################
## ADJUSTABLE PARAMETERS                                   ##
#############################################################

structure          cdn-solvate.psf
coordinates        cdn-solvate.pdb

set temperature    310
set outputname     cdn-out
set restartname    cdn-restart

firsttimestep      0

#############################################################
## SIMULATION PARAMETERS                                   ##
#############################################################

# Input
paraTypeCharmm     on
parameters          par_all27_prot_lipid.inp
temperature         $temperature

# Force-Field Parameters
exclude             scaled1-4
1-4scaling          1.0
cutoff              12.
switching           on
switchdist          10.
pairlistdist        14
margin     1.0

# Integrator Parameters
timestep            2.0  ;# 2fs/step
rigidBonds          all  ;# needed for 2fs steps
nonbondedFreq       2
fullElectFrequency  2
stepspercycle       20
```

```
# Constant Temperature Control
langevin            on    ;# do langevin dynamics
langevinDamping     5     ;# damping coefficient (gamma) of 5/ps
langevinTemp        $temperature
langevinHydrogen    off    ;# don’t couple langevin bath to hydrogens

# Periodic Boundary Conditions
cellBasisVector1    45 0.0 0.0
cellBasisVector2   0.0  40 0.0
cellBasisVector3   0.0 0.0  42
cellOrigin -0.881323575974 0.396021813154 -5.1848950386
wrapAll on

# PME (for full-system periodic electrostatics)
PME             yes
PMEGridSizeX  50 # grid-pts along cellBasisVector1.
PMEGridSizeY  48 # along cellBasisVector2.
PMEGridSizeZ  48 # along cellBasisVector3

# Constant Pressure Control (variable volume)
useGroupPressure      yes ;# needed for rigidBonds
useFlexibleCell       no
useConstantArea       no

langevinPiston        on
langevinPistonTarget  1.01325 ;#  in bar -> 1 atm
langevinPistonPeriod  200.
langevinPistonDecay   100.
langevinPistonTemp    $temperature

# Output
outputName      $outputname

restartname $restartname
restartsave yes
binaryrestart   yes
restartfreq     100000

dcdfreq         3000
xstFreq         3000

#############################################################
## EXTRA PARAMETERS                                        ##
#############################################################

#############################################################
## EXECUTION SCRIPT                                        ##
#############################################################

# Minimization
minimize 2000;  #2000 time steps
reinitvels      $temperature

run 9000000; # 4.5 ns
```

Appendix C

Utility Applications

C.1 psfgen

psfgen is a molecular structure building tool. Some of its capabilities include estimating missing atomic coordinates, extracting coordinate data from PDB files, deleting selected atoms from a structure and more. The tool is available as a standalone executable for both Unix and Windows platforms.

The main steps required to estimate missing coordinate data include: reading in a topology file, building a protein segment, reading protein coordinates from a PDB file, estimating values of missing coordinates, writing the estimated structure and coordinates to PSF and PDB files. Further details on psfgen are available on the psfgen web page [51].

The following is an example of a psfgen input file for the 1CDN molecule:

```
topology top_all22_prot.inp
alias residue HIS HSD
alias atom ILE CD1 CD
alias atom GLY OXT OT1
segment 1CDN { 1cdn.pdb }
coordpdb 1cdn.pdb 1CDN
```

```
guesscoord
writepdb 1cdn-psfgen.pdb
writepsf 1cdn-psfgen.psf
```

C.2 VMD

VMD (Visual Molecular Dynamics) is a program for displaying, animating, and analyzing large biomolecular systems using 3D graphics and scripting. The application displays structures represented by PDB and PSF files as well as DCD output generated from a NAMD simulations. Display features enable the user to choose different structure representations such as ribbon, cartoon, tube and more. Other analytical or utility capabilities of VMD include the ability to perform rigid structural alignments, add solvent to the structure environment and carry out simple operations on the structure (i.e. select a group of atoms, measure the center of the structure).

The application may also be connected to a live NAMD simulation to allow the user to interaction with the simulation. VMD is freely available and supports computers running on a variety of platforms [52].

C.3 MDEnergy

MDEnergy is a small application to calculate energies of structures represented in DCD or PDB files. Since the program is derived from a simplified version of NAMD, the total energy calculations are identical to the ones calculated by NAMD. The user may specify a subset of atoms on which to perform the energy calculation or specify individual energy terms to use in the energy calculation. More information about MDEnergy is available online [53].

C.4 MatDCD

MatDCD is a program that allows DCD files to be read into Matlab and written to DCD format from Matlab. When reading the DCD file into Matlab, the user may specify which atoms to load. The program is available online [54].

Appendix D

Hardware

The molecular dynamics simulations were run on a high performance multi processor CPU machine provided by SHARCNET [55] at the University of Waterloo. The specifications of the machine are as follows:

- SGI Altix 3700
- 64 Itanium2 processors
- 128 GB memory
- OS: ALE (Advanced Linux Environment - a RedHat variant)

Bibliography

[1] A. Wlodawer and J. Vondrasek, "Inhibitors of hiv-1 protease: A major success of structure-assisted drug design," *Ann. Rev. Biophys. Biomol. Struct.*, vol. 27, pp. 249–284, 1998.

[2] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14863–14868, 1998.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cog. Neu.*, vol. 3, pp. 71–86, 1991.

[4] J. Richards, "Remote sensing digital image analysis," *New York: Springer-Verlag*, 1993.

[5] M. Teodoro, G. P. Jr., and L. Kavraki., "A dimensionality reduction approach to modeling protein flexibility," *In Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pp. 299–308, 2002.

[6] E. Fischer, "Einfluss der configuration auf die wirkung derenzyme," *Ber. Dtsch. Chem. Ges.*, vol. 27, pp. 2985–2993, 1894.

[7] S. Subbiah, *Protein Motions.* Chapman & Hall, 1996.

[8] A. P. et al., "X-ray crystallography at subatomic resolution," 2002.

[9] T. Creighton, *Proteins: Structures and Molecular Properties.* W.H. Freeman and Company, 1992.

[10] B. Alder and T. E. Wainwright, "Phase transition for a hard sphere system," *J. Chem. Phys.*, vol. 27, pp. 1208–1209, 1957.

[11] A. Rahman, "Report a136," *Phys. Rev.*, p. 405, 1964.

[12] F. H. Stillinger and A. Rahman, "Improved simulation of liquid water by molecular dynamics," *J. Chem. Phys.*, vol. 60, pp. 1545–1557, 1974.

[13] J. A. McCammon, B. Gelin, and M. Karplus, "Dynamics of folded proteins," *Nature*, vol. 267, p. 585, 1977.

[14] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.

[15] M. McClure, T. Vasi, and W. Fitch, "Comparative analysis of multiple protein-sequence alignment methods," *Mol. Biol. Evol.*, vol. 11, pp. 571–592, 1994.

[16] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Cryst.*, vol. A32, p. 922, 1976.

[17] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *J. Mol. Biol.*, vol. 233, pp. 123–138, 1993.

[18] A. Falicov and F. Cohen, "A surface of minimum area metric for the structural comparison of proteins," *J. Mol. Biol.*, vol. 258, pp. 871–892, 1996.

[19] I. Shindyalov and P. Bourne, "Protein structure alignment by incremental combinatorial extension (ce) of the optimal path," *Protein Eng*, vol. 11, pp. 739–747, 1998.

[20] M. Shatsky, H. Wolfson, and R. Nussinov, "Flexible protein alignment and hinge detection," *Proteins: Structure, Function, and Genetics*, vol. 48, pp. 242–256, 2002.

[21] Y. Ye and A. Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," *Bioinformatics*, vol. 19, pp. 246–255, 2003.

[22] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[23] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psych.*, vol. 24, pp. 417–441, 1933.

[24] A. Goldstein, L. Harmon, and A. Lesk, "Identification of human faces," *Proc. IEEE*, vol. 9, pp. 748–760, 1971.

[25] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103–108, 1990.

[26] O. Alter, P. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proc. Natl. Acad. Sci. USA*, vol. 97, pp. 8409–8414, 2000.

[27] N. Holter, M. Mitra, A. Maritan, M. Cieplak, J. Banavar, and N. Fedoroff, "Fundamental patterns underlying gene expression profiles: simplicity from complexity," *Proc. Natl. Acad. Sci. USA*, vol. 7, pp. 30–42, January 1996.

[28] I. Jolliffe, *Principal Component Analysis.* New York: Springer, 2nd ed., 2002.

[29] S. Hui and M. Shakeel, "An investigative approach into dimensionality reduction techniques for protein flexibility modeling." ISMB 2004 Poster Presentation http://www.iscb.org/ismb2004/, 2004.

[30] W. Wriggers and K. Schulten, "Protein domain movements: Detection of rigid domains and visualization of effective rotations in comparisons of atomic coordinates," *Proteins: Structure, Function, and Genetics*, vol. 29, pp. 1–14, 1997.

[31] O. Troyanskaya, M. Cantor, G. Sherlock, P. B. P., T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, pp. 520–25, 2001.

[32] R. Little and D. Rubin, *Statistical Analysis with Missing Data.* John Wiley & Sons, Inc., 1987.

[33] R. Cattell, "The scree test for the number of factors," *Multivariate Behavioral Research*, vol. 1, pp. 245–276, 1966.

[34] M. Gerstein and W. Krebs, "A database of macromolecular motions," *Nucleic Acids Res.*, vol. 26, pp. 4280–4290, 1998.

[35] N. Echols, D. Milburn, and M. Gerstein, "Molmovdb: analysis and visualization of conformational change and structural flexibility," *Nucleic Acids Res.*, vol. 31, pp. 478–482, 2003.

[36] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000.

[37] L. Kal, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, J. P. N. Krawetz, A. Shi-

nozaki, K. Varadarajan, and K. Schulten, "Namd2: Greater scalability for parallel molecular dynamics," *J. Comp. Phys.*, vol. 151, pp. 283–312, 1999.

[38] W. L. Jorgensen, J. Chandrasekhar, and J. P. Madura, "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.*, vol. 79, pp. 926–935, 1983.

[39] A. Amadei, A. B. M. Linssen, and H. J. C. Berendsen, "Essential dynamics of proteins," *Prot. Struct. Funct. Genet.*, vol. 17, p. 412425, 1993.

[40] G. Halder, P. Callaerts, and W. Gehring, "Induction of ectopic eyes by targeted expression of the eyeless gene in drosophila," *Science*, vol. 267, pp. 1788–1792, 1995.

[41] S. Roweis, "Em algorithms for pca and sensible pca," *Technical Report - California Institute of Technology, Computation and Neural Systems*, 1997.

[42] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd ed., 1992.

[43] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[44] J. B. Tenenbaum and J. C. L. Vin de Silva, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[45] D. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. USA*, vol. 100, p. 55915596, 2003.

[46] M. Brand, "Charting a manifold," *Neural Information Processing Systems (NIPS)*, vol. NIP 15, 2002.

[47] A. D. M. et al., "All-atom empirical potential for molecular modeling and dynamics studies of proteins," *J. Phys. Chem.*, vol. 102, pp. 3586–3616, 1999.

[48] J. M. Haile, *Molecular Dynamics Simulation: Elementary Methods.* John Wiley & Sons, Inc., 1992.

[49] D. Frenkel and B. Smit, *Understanding Molecular Simulation From Algorithms to Applications.* Academic Press, 2nd ed., 2002.

[50] L. Kal, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, J. P. N. Krawetz, A. Shinozaki, K. Varadarajan, and K. Schulten, "Namd - not another molecular dynamics." http://www.ks.uiuc.edu/Research/namd/, 1999.

[51] J. Gullingsrud and J. Phillips, "psfgen version 1.3.3." http://www.ks.uiuc.edu/Research/vmd/plugins/psfgen/, 2005.

[52] W. Humphrey, A. Dalke, and K. Schulten, "Vmd - visual molecular dynamics." http://www.ks.uiuc.edu/Research/vmd/, 1996.

[53] J. Saam, "Mdenergy - energy evaluation tool." http://www.ks.uiuc.edu/Development/MDTools/mdenergy/, 2004.

[54] J. Gullingsrud, "Matdcd - matlab package dcd reading/writing." http://www.ks.uiuc.edu/Development/MDTools/matdcd/, 2000.

[55] "Sharcnet - shared hierarchical academic research computing network." http://www.sharcnet.ca/, 2005.