Computations to Obtain Wider Tunnels in Protein Structures

by

Somayyeh Zangooei

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

Waterloo, Ontario, Canada, 2011

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

Finding wide tunnels in protein structures is an important problem in Structural Bioinformatics with applications in various areas such as drug design. Several algorithms have been proposed for finding wide tunnels in a fixed protein conformation. However, to the best of our knowledge, none of the existing work have considered widening the tunnel, i.e., finding a wider tunnel in an alternative conformation of the given structure. In this thesis we initiate this line of research by proposing a tunnel-widening algorithm which aims to make the tunnel wider by a slight local change in the structure of the protein.

Given a fixed conformation of a protein with a point located inside it, we first describe an algorithm to identify the widest tunnel from that point to the outside environment of the protein. Then we try to make the tunnel wider by considering various alternative conformations of the protein. We only consider conformations whose energies are not much higher than the energy of the initial conformation. Among these alternative conformations we select the one with the widest tunnel. However, the alternative conformation with the widest tunnel might not be accessible from the initial structure. Thus, in the next step we develop three algorithms for finding a feasible transition pathway from the initial structure to the alternative conformation, i.e., a sequence of intermediate conformations between the initial structure and the alternative conformation such that the energy values of all these intermediate conformations are close to the energy of the initial structure.

We evaluate our tunnel-finding and tunnel-widening algorithms on various proteins. Our experiments show that in most cases we can make the tunnel wider in an alternative conformation. However, there are cases in which we find a wider tunnel in an alternative conformation, but the energy value of the alternative conformation is much higher than the energy of the initial structure. We also implemented our three pathway-finding algorithms and tested them on various instances. Our experiments show that although in most cases we can find a feasible transition pathway, there are cases in which the alternative conformation has energy close to the initial structure, but our algorithms cannot find any feasible pathway from the initial structure to the alternative conformation. Furthermore, there is a trade-off between the running time and accuracy of the three pathway-finding algorithms.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Professor Forbes J. Burkowski, for his continuous support and encouragement during my graduate studies. I greatly appreciate the effort and enthusiasm that he has invested in my research.

I would also like to thank my thesis committee members Professor Brendan J. McConkey and Professor Ming Li for their constructive and insightful comments.

My deepest gratitude goes to my parents, Mohammad-Ali and Maryam, for their unflagging support, love, and encouragement throughout my life. Although I am far away from them, they have always been there for me. Last but not least, my special thanks to my beloved husband, Reza, who has accompanied me with his love, inspiration, and endless support. Without his help, I would never have been able to accomplish this work.

Dedication

To my dear parents

To my beloved husband

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins adopt complex three dimensional structures containing various cavities, pockets, clefts, pores, channels, and tunnels. Understanding and analyzing these structural properties have great theoretical and practical importance as they play a part in protein functionality. In this thesis we are mainly interested in discovering *tunnels*, i.e., routes or paths from the outside environment to a position inside the protein and vice versa. Finding tunnels in protein structures is an important problem in Structural Bioinformatics, with applications in areas such as drug design. The drug (a *ligand* [1]) will bind to a specific part of the protein, called the binding site. In some cases, the binding site is buried deep inside the protein. This is especially applicable to enzymes, in which binding sites are usually conserved in the protein [68]. Thus the ligand needs to find a tunnel from the outside environment to the binding site. This tunnel should be wide enough to guarantee that the ligand does not clash with other atoms. This motivates the problem of finding the widest tunnels in protein structures.

In this thesis we consider three main problems related to finding wide tunnels in proteins: the *tunnel-finding problem*, the *tunnel-widening problem*, and the *pathway-finding problem*. In the tunnel-finding problem we are given a fixed conformation of a protein and the coordinates of a point located inside the protein structure. We refer to this point

[1] A ligand is a substance that attaches to a special region of a biomolecule to serve a biological purpose.

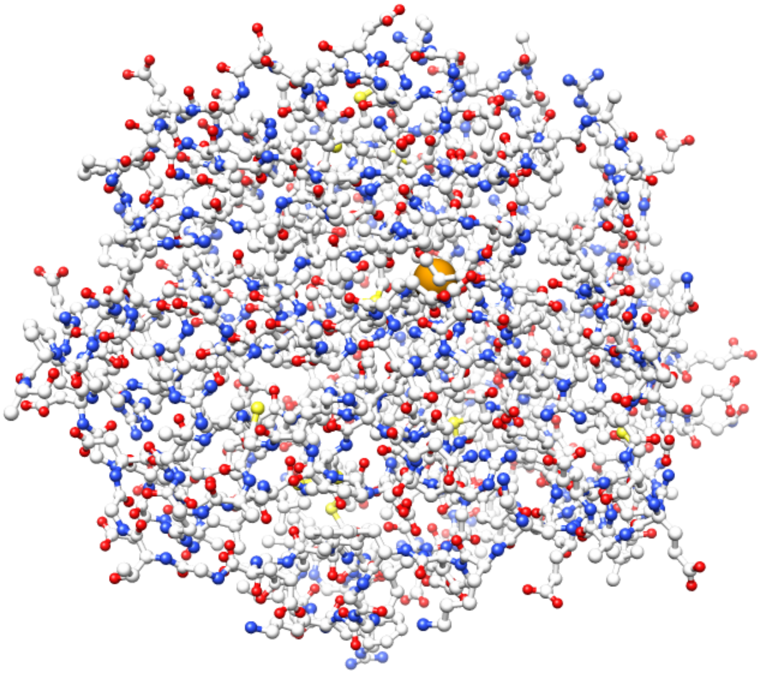


Figure 1.1: Protein 1CV2 with starting point at position (14,15,22). The starting point is shown by an orange sphere.

as the *starting point* of the tunnel. Our objective is to find the widest tunnel from the starting point to the outside environment of the protein. We consider the outside environment to be anywhere outside the convex hull of the protein atoms. Note that in a drug design application we are interested in finding a tunnel from the binding site to the outside environment. For simplicity we assume that the binding site can be modeled by a single point whose coordinates are provided by the user. An instance of this problem is shown in Figure 1.1. We are given the protein with PDB ID 1CV2 and the starting point is at position (14,15,22) in the frame of reference used by the PDB coordinates.

We will describe a *tunnel-finding algorithm* that finds the widest tunnel in a fixed protein conformation from a given starting point. While this is the widest tunnel in the given conformation, it is possible that there exists a wider tunnel from the starting point to the outside environment in a different conformation of the same protein. In the tunnel-widening problem, our objective is to *widen* the tunnel, i.e., to find an alternative conformation of the initial structure with a wider tunnel. This is motivated by some applications in which widening the tunnel is of great importance and interest. For example, consider the drug design application described earlier. It is possible that the tunnel discovered by the tunnel-finding algorithm is not wide enough for the drug, while another conformation of the protein has a sufficiently wide tunnel. Thus knowing that a wider tunnel can exist might lead to improvements in drug design. Note that some alternative conformations have energy values much higher than the energy of the initial conformation and thus the probability of transition from the initial structure to these alternative conformations is very low. Therefore we only consider alternative conformations whose energy values are not much higher than the energy of the initial conformation.

In the pathway-finding problem we attempt to find a *transition pathway* from the initial structure to an alternative conformation, i.e., a sequence of intermediate conformations between the initial and alternative conformations such that each two consecutive conformations have sufficiently similar structures. To be more precise, we are looking for a *feasible* transition pathway, defined as a transition pathway whose conformations do not have energy values much higher than the energy of the initial structure.

1.1 Related Work

Various algorithms and software tools have been proposed to discover and analyze the structural properties of proteins, e.g., POCKET [43], VOIDOO [37], HOLE [66], CAST [44], CAVER [54], MOLE [55], MolAxis [75], CAVER2 [49], and CHUNNEL [15]. Among these algorithms POCKET, VOIDOO, and CAST were developed to find cavities and pockets inside a given protein structure. Some other algorithms such as HOLE and CHUNNEL aim to find *channels*, i.e., holes that go completely through the protein, thus having two entrances or mouths. [2] Therefore, these algorithms are not directly related to our work.

[2] Note that there is a bit of confusion about the definition of channels and tunnels. For example, Coleman and Sharp [15] refer to channels as tunnels.

CAVER, Mole, and CAVER2 attempt to solve the tunnel-finding problem. Thus these algorithms are more relevant to our work and we provide more details about them. CAVER is based on the idea of partitioning the space into a set of three dimensional grid cubes and then using a variant of the Dijkstra's algorithm [20] to find a wide tunnel. The accuracy of the algorithm depends on the resolution of the grid and it does not guarantee the discovery of the widest tunnel. CAVER2 and MOLE improve the CAVER algorithm by constructing a graph based on the Voronoi diagram (or equivalently Delaunay tessellation) [3] derived from the protein atoms and then use a variant of the Dijkstra's algorithm to compute the best tunnel. MOLE tries to find short and wide tunnels by defining the objective function as a combination of the length and width. CAVER2 on the other hand only considers the width and provides an algorithm for computing the widest tunnel in a fixed protein conformation. Our tunnel-finding algorithm (Section 3.1) is based on ideas similar to CAVER2. However, CAVER2 [49] is only described in two dimensions and does not provide the details of the algorithm.

As far as we know, there is no other algorithmic work on the tunnel-widening problem. On the other hand, various techniques are proposed for finding feasible transition pathways between two protein conformations, e.g., *targeted molecular dynamics* [62, 74] and *elastic network interpolation* [34, 35]. However, these techniques are designed for the general case of the problem, while we have a special setting in which the initial and target conformations are structurally close (see Section 3.3 for more details). We proposed several pathway-finding algorithms that are tailored to our special settings and thus solve the pathway-finding problem more efficiently.

1.2 Contributions

The main contribution of this thesis is developing novel techniques and algorithms for the tunnel-widening and pathway-finding problems. For the tunnel-finding problem, although

[3] Refer to Chapter 2 for more details about Voronoi diagram, Delaunay tessellation, and Dijkstra's algorithm.

the idea of our algorithm is based on CAVER2 [49], we have provided a much more comprehensive presentation of the algorithm. Due to lack of details in [49] we had to develop most parts of the algorithm without relying on previous efforts. Regarding the tunnel-widening problem, to the best of our knowledge, this is the first algorithmic work on the problem. We implemented and visualized both tunnel-finding and tunnel-widening algorithms in Chimera/Python and applied them to several proteins of different sizes and verified that their experimental results are promising. We also developed three different algorithms for the pathway-finding problem and compared their performance on various input instances.

1.3 Organization of the Thesis

The thesis is structured as follows. In Chapter 2 we provide some relevant background information on proteins, Voronoi diagrams, Delaunay triangulation, and Dijkstra's algorithm. The descriptions of tunnel-finding, tunnel-widening, and pathway-finding algorithms are provided in Chapter 3. In Chapter 4 we present and analyze the results of applying our algorithms to various input instances. The conclusions of the thesis are provided in Chapter 5.

Chapter 2

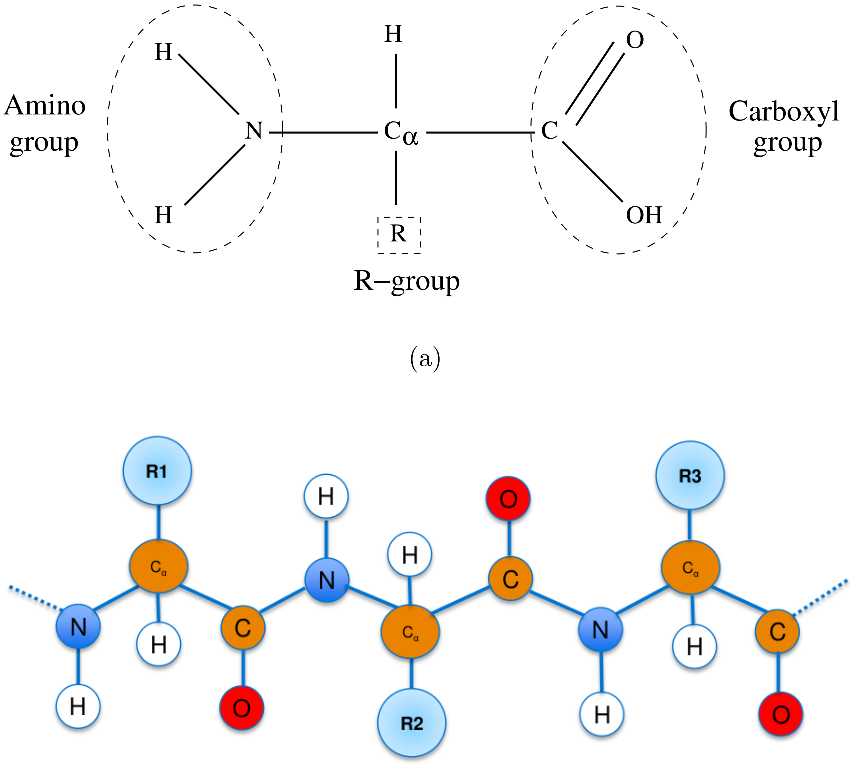
Fundamental Concepts and Definitions

In this chapter we explain some fundamental concepts and definitions needed for this thesis.

2.1 Proteins

Proteins are one of the main components of living organisms and play a vital role in both structural and biological processes [45]. A protein is a chain of amino acids. Each amino acid consists of a side-chain (also called an R-group), an amino group (NH_2), and a carboxyl group ($COOH$) (see Figure 2.1). The side-chain is a group of atoms attached to an α-carbon (C_α). The α-carbon is also connected to the amino group and carboxyl group. Two amino acids can be joined together and form a *peptide bond*. The chain of peptide bonds forms the *protein backbone*. An amino acid that bonds to another amino acid to form a peptide bond is referred to an *amino acid residue*, since it loses a water molecule during the reaction. There are 20 standard amino acids in nature. These amino acids are linked through peptide bonds and form the vast variety of proteins.

Each protein performs a specific function that is related to its three-dimensional structure. This structure can be described by Cartesian coordinates of its atoms. Alternatively,



(a)

(b)

Figure 2.1: (a) General structure of an amino acid. (b) A chain of amino acids.

the three-dimensional structure of a protein can be defined by its *internal coordinates*, i.e., *bond lengths*, *bond angles*, and *dihedral angles* [9]. The bond length refers to the average distance between two bonded atoms and the bond angle describes the angle formed by three successive bonded atoms. A sequence of four consecutive bonded atoms forms a dihedral angle (also referred to as torsion angle). Several efficient algorithms have been proposed to convert the Cartesian coordinates of a protein to its internal coordinates and vice versa [53, 2, 76].

The backbone of a protein can be represented by a sequence of dihedral angles, denoted by ϕ, ψ, and ω. Angle ϕ is determined by a sequence $C - N - C_\alpha - C$ of backbone atoms. In other words, it describes the rotation about the $N - C_\alpha$ bond. Angle ψ involves the sequence $N-C_\alpha-C-N$ of backbone atoms, while ω angle is determined by the consecutive backbone atoms $C_\alpha - C - N - C_\alpha$. The ω dihedral angle is either $0°$ or $180°$. The other two dihedral angles take different values, although not all pairs of ϕ-ψ are possible. The Ramachandran plot [58, 59, 33] shows the possible values of ϕ and ψ dihedral angle pairs for an amino-acid residue in a protein.

2.1.1 Protein Structure

The structure of proteins is complex and can be described in several levels:

- **Primary structure**:

  A protein is comprised of a linear sequence of amino acids covalently joined together by peptide bonds [17]. A typical protein contains between 100-1000 amino acids [28]. The order of the bonded amino acids in the sequence is described by its *primary structure*. Each protein has its own unique sequence which determines its biological function and structure. Figure 2.2 shows the primary structure of a protein.

- **Secondary Structure**:

  The *secondary structure* of the protein considers the local three-dimensional configurations that may appear in the structure. The secondary structure is mainly formed

GSRRASVGSEFMVVDVTIEDSYSTE
SAWVRCDDCFKWRRIPASVVGSIDE
SSRWICMNNSDKRFADCSKSQEMSN
EEINEELGIGQDEADAYDCDAAKRG

Figure 2.2: The primary structure of protein with PDB ID 2L7P.

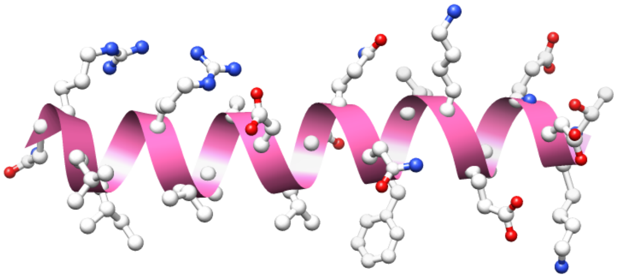
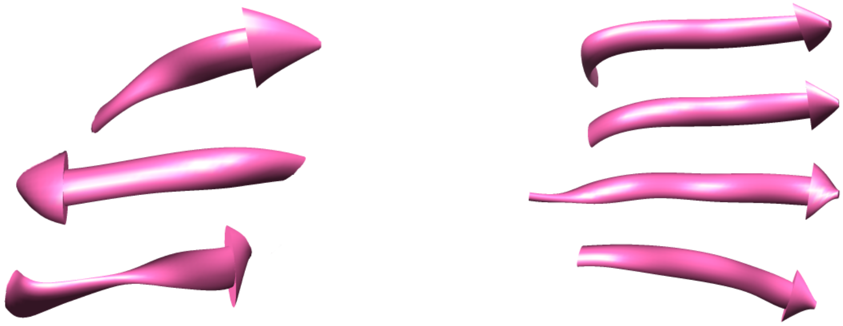


Figure 2.3: Ribbon diagram of an alpha helix with side-chains.

by hydrogen bond interactions between the atoms in the backbone [10]. There are three types of secondary structure: alpha helices, beta sheets, and loops.

- **Alpha Helix:** The *alpha helix* is the most common type of secondary structure in the proteins and consists of many hydrogen bonds between amino acid residues [56]. An alpha helix structure is stabilized by hydrogen bonding interactions between the $N-H$ group of residue n and the $C=O$ group of residue $n+4$. Each alpha helix has 3.6 residues for every complete turn of the helix. The length of each turn is about 5.4 Å [10]. Figure 2.3 shows an alpha helix extracted from the protein with PDB ID 1CV4.
- **Beta Sheet:** Another regular type of secondary structure found in proteins is the *beta sheet*. Similar to alpha helices, the hydrogen bonds are one of the important characteristics of beta sheets. However, in contrast to an alpha helix, the hydrogen bonds are between the amino groups of two chain segments whose amino acids may be quite distant in the primary sequence. Beta sheets consist of several beta strands held by hydrogen bonds. Adjacent beta strands can have



(a) (b)

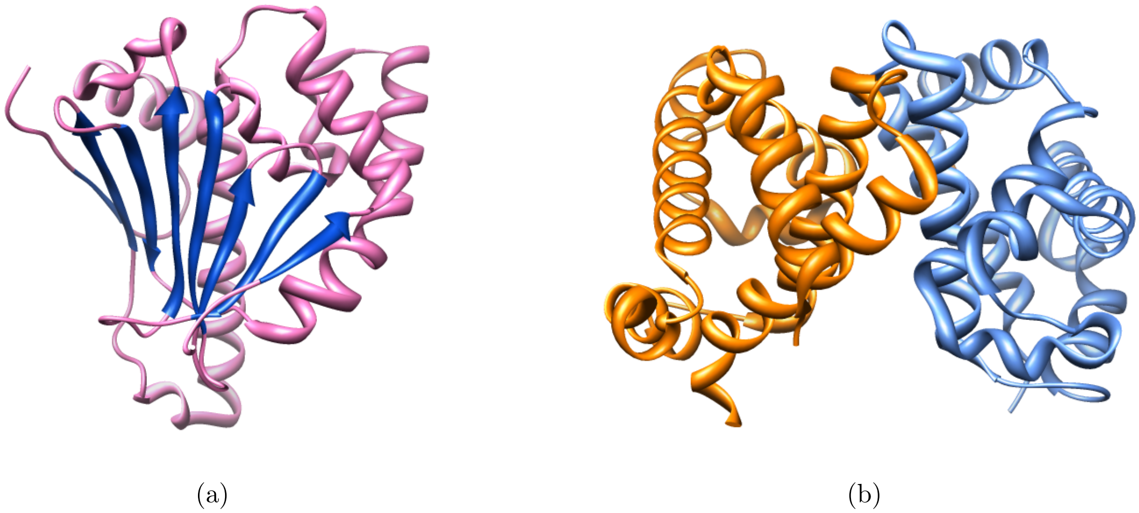
Figure 2.4: (a) Ribbon diagram for antiparallel beta strands (selected form protein 1CV4) (b) Ribbon diagram for parallel beta strands (selected from protein 1CV2).

three possible arrangements and form parallel, antiparallel, or mixed beta sheets. In parallel beta sheets, the beta strands are aligned such that the N-terminal ends (also called amino-terminals) of all strands point to the same directions. However, in antiparallel arrangement the N-terminal end of one strand points to the same direction as the C-terminal end of its adjacent strand. Thus, the arrangement of the N-terminal ends of beta strands alternate. A combination of parallel and antiparallel beta strands forms a mixed beta sheet (see Figure 2.4).

- **Loop:** The *loop* is another category of secondary structure of proteins. A loop consists of a chain of amino acid residues that does not have any hydrogen bond interaction with other regions of protein. Loops have varying lengths and connect the alpha helices and beta sheets [1].

- **Tertiary Structure:**

  The three-dimensional structure of the protein is formed by combining all secondary structures including the alpha helices, beta sheets, and loops. Knowing the tertiary structure of a protein is required for describing the biological function of the protein [9]. The atoms of a protein can be arranged in different configurations in three-



(a)

(b)

Figure 2.5: (a) Tertiary structure of protein 3NMQ (b) Quaternary structure of protein 1YZI.

dimensional space. Each such spatial arrangement is called a protein *conformation*. The tertiary structure of protein with PDB ID 3NMQ is shown in Figure 2.5(a).

- **Quaternary Structure:**

  There are some proteins that are comprised of multiple protein chains or subunits. The spatial arrangement of these subunits is called the *quaternary structure* of the protein [69]. The quaternary structure is stabilized by the same interactions as the ones in the secondary and tertiary structures [7]. Figure 2.5(b) shows the quaternary structure of protein with PDB ID 1YZI.

2.1.2 Rotamers

The atoms of the side-chain of an amino acid residue can adopt different conformations in the space. Each side-chain conformation is called a *rotamer* (see Figure 2.6). A collection of side-chain conformations (rotamers) provides a side-chain conformational space. Predicting

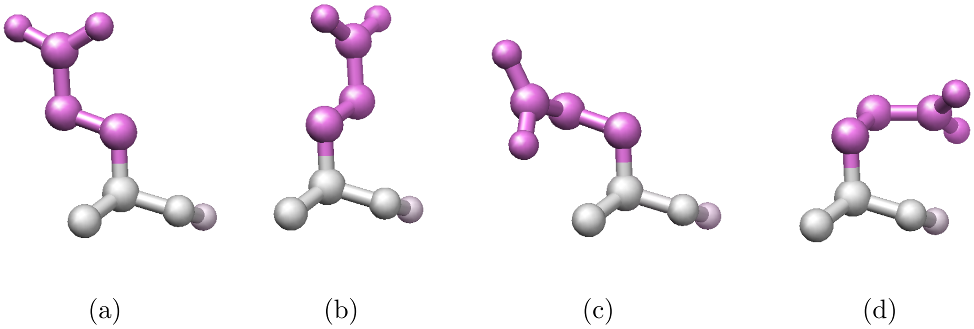


Figure 2.6: Different rotamers for the Glutamic acid (GLU).

the protein side-chain conformations is an important aspect of protein structure prediction. Note that the bond lengths and bond angles are the same in all rotamers of a side-chain, but the side-chain dihedral angles (or chi angles), i.e., the angles defined by each four consecutive side-chain atoms, are different. Each side-chain can have at most five chi angles, denoted by χ_1, χ_2, χ_3, χ_4, and χ_5. The χ_1 angle is the first rotatable side-chain dihedral angle, defined by $N - C_\alpha - C_\beta - C_\gamma$ atoms, χ_2 is defined by $C_\alpha - C_\beta - C_\gamma - C_\delta$, and so on. The information about possible values of chi angles for different rotamers of a side-chain is provided by rotamer libraries, such as the Dunbrack rotamer library [1] [23] and the Richardson rotamer library [46]. [2]

Rotamer libraries describe a discrete conformational space of side-chains. More specifically, they provide information about the dihedral angles of side-chain conformations and observed frequency of each conformation. Furthermore, they usually contain information regarding the variance about dihedral angle means or modes. Rotamer libraries can be backbone-dependent or backbone-independent. Backbone-dependent rotamer libraries provide information about side-chain conformational space as a function of backbone dihedral angles ϕ and ψ [25, 26, 24]. On the other hand, the backbone-independent rotamer libraries do not depend on the backbone dihedral angles [46, 38, 47].

[1] http://dunbrack.fccc.edu/Home.php

[2] http://pibs.duke.edu/databases/rotamer.php

2.1.3 The Protein Data Bank (PDB)

The protein Data Bank (PDB)[3] is a standard repository that provides information about three-dimensional structures of biological molecules such as proteins [5]. Since 1998, The PDB is supported by the Research Collaboratory for Structural Bioinformatics (RCSB).[4] The PDB contains structural information about thousands of biological molecules that have been obtained by X-ray crystallography or NMR spectroscopy. Each structure has been assigned a unique PDB ID which is a four-character alphanumeric identifier. In this thesis, we usually refer to proteins with their PDB IDs. The three-dimensional coordinates of molecule atoms are provided in PDB files and can be downloaded from the PDB website.

2.1.4 Potential Energy and Boltzmann's Distribution

Potential energy of a protein is related to the structural arrangement of its atoms. In other words, the potential energy can be represented as a function of all the relevant atomic coordinates [72]. A protein can adopt different conformations depending on its potential energy. More specifically, according to Boltzmann's distribution [21] the probability that a protein adopts a certain conformation is exponentially related to the negative of its energy. Now consider two conformations C_1 and C_2 of a protein, where the energy of C_i is E_i. The relative population of these two conformations can be computed by the following formula

$$\frac{N_2}{N_1} = e^{\frac{-(E_2-E_1)}{kT}}, \tag{2.1}$$

where N_i is the population of conformation C_i, $k = 0.0019872041$ kcal/mol/K is the Boltzmann's constant, and T is the temperature measured in Kelvin [21]. According to this formula, the relative population of two conformation depends both on the energy difference $\Delta E = E_2 - E_1$ and the temperature T. If the energy of C_2 is much higher than the energy of C_1, N_2/N_1 is very small, and the probability that the protein adopts C_2 is

[3] http://www.pdb.org/pdb/home/home.do

[4] http://home.rcsb.org

very low. On the other hand, increasing the temperature increases the relative population N_2/N_1.

2.2 Voronoi Diagram

The Voronoi diagram is a well-known concept in computational geometry with applications in various areas such as physics, geography, anthropology, astronomy, biology, marketing, etc. [52]. Voronoi diagrams are usually attributed to Dirichlet [22] and Voronoi [71, 70]. Due to applications of these diagrams in different areas, they were independently rediscovered by various researchers (see Chapter 1 of [52] for more information about the history of Voronoi diagrams).

2.2.1 Definitions and Properties

Let $P = \{p_1, p_2, p_3, ..., p_n\}$ be a set of n distinct points (also called sites) in the space S. For simplicity we first describe the Voronoi diagram in two dimensions, i.e., we consider the case $S = \mathbb{R}^2$. We assume that the points are in *general position*, i.e., no three points are collinear and no four points lie on the same circle. The Voronoi diagram of P, denoted by $V(P)$, partitions the space S into n regions. Each region is associated with a site p_i and contains all points that are closer to p_i than to any other site in P. More formally, a point $q \in S$ lies in the region corresponding to the site $p_i \in P$ if and only if the following property holds:

$$\forall p_j \in P\text{:}\;\; d(q, p_i) \leq d(q, p_j),$$

where $d(p, q)$ denotes the Euclidean distance between two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ in $\mathbb{R}^2$, i.e.,

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

For each site p_i, the region associated with p_i is called a *Voronoi cell* and denoted by $V(p_i)$. Therefore, we have

$$V(p_i) = \{x \in S \mid \forall p_j \in P : d(x, p_i) \leq d(x, p_j)\}.$$

Observe that $V(P) = \cup_{i=1}^{n} V(p_i)$.

Figure 2.7 shows the Voronoi diagram of a set of 9 points in $\mathbb{R}^2$. In addition to Voronoi cells, the Voronoi diagram contains Voronoi edges and Voronoi vertices. We say that two Voronoi cells $V(p_i)$ and $V(p_j)$ are adjacent if and only if $V(p_i) \cap V(p_j) \neq \emptyset$. Each two adjacent Voronoi cells share an edge called a *Voronoi edge*. In other words, for each two adjacent Voronoi cells $V(p_i)$ and $V(p_j)$ the Voronoi edge between them is defined as

$$V(p_i) \cap V(p_j) = \{x \in S \mid d(x, p_i) = d(x, p_j)\}.$$

Let q be a point on the Voronoi edge between $V(p_i)$ and $V(p_j)$. Observe that the distance of q to any site in $P \setminus \{p_i, p_j\}$ is more than $d(q, p_i) = d(q, p_j)$. Voronoi edges intersect each other at *Voronoi vertices*. Since the points are in general position, each Voronoi vertex is the intersection of three Voronoi edges and is equidistant from three sites. The largest empty circle centered at a point $s \in S$, denoted by $C(s)$, is defined as the largest circle centered at s that does not contain any site $p_i \in P$ in its interior. Note that $C(s)$ contains at least one site on its boundary.

The Voronoi diagrams have several geometric properties. Here we list a few of them (see [18, 52, 4] for more properties as well as the proofs):

- For every two adjacent Voronoi cells $V(p_i)$ and $V(p_j)$, there exists a point $x \in V(p_i) \cap V(p_j)$ such that the largest empty circle centered at x passes through only p_i and p_j. For example, in Figure 2.8 the largest empty circle centered at x_1 passes through only p_2 and p_9.

- Let x be a Voronoi vertex in the Voronoi diagram. Then the largest empty circle centered at x ($C(x)$) passes through three sites of P. For instance, in Figure 2.8 x_2

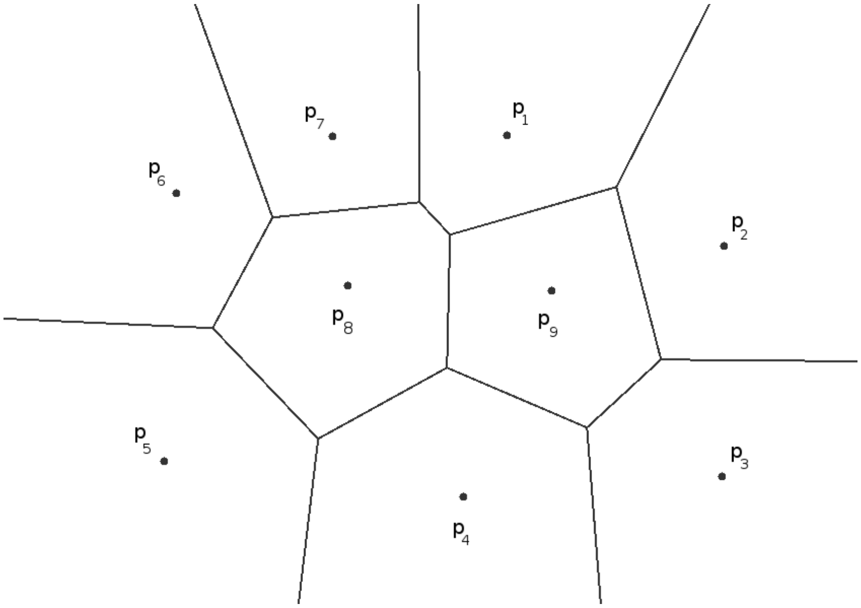


Figure 2.7: The Voronoi diagram for a set of 9 points in the plane.

is a Voronoi vertex and the largest empty circle centered at x_2 passes through p_6, p_7, and p_8.

- Consider the Voronoi diagram of a set P in $\mathbb{R}^2$. Let $n = |P|$, n_e, and n_v be the the number of sites, Voronoi edges, and Voronoi vertices, respectively. Then we have the following bounds on n_e and n_v [18, 52]:

$$n_v - n_e + n = 1 \qquad n \geq 2 \tag{2.2}$$

$$n_e \leq 3n - 6 \qquad n \geq 3 \tag{2.3}$$

$$n_v \leq 2n - 5 \qquad n \geq 3 \tag{2.4}$$

  Thus the number of Voronoi edges and vertices is $O(n)$. The *combinatorial complexity* of Voronoi diagram is defined as the total number of Voronoi cells, vertices, and edges. Thus, the Voronoi diagram in $\mathbb{R}^2$ has linear complexity.

- A site $p_i \in P$ lies on the convex hull of P if and only if the Voronoi cell $V(p_i)$ is

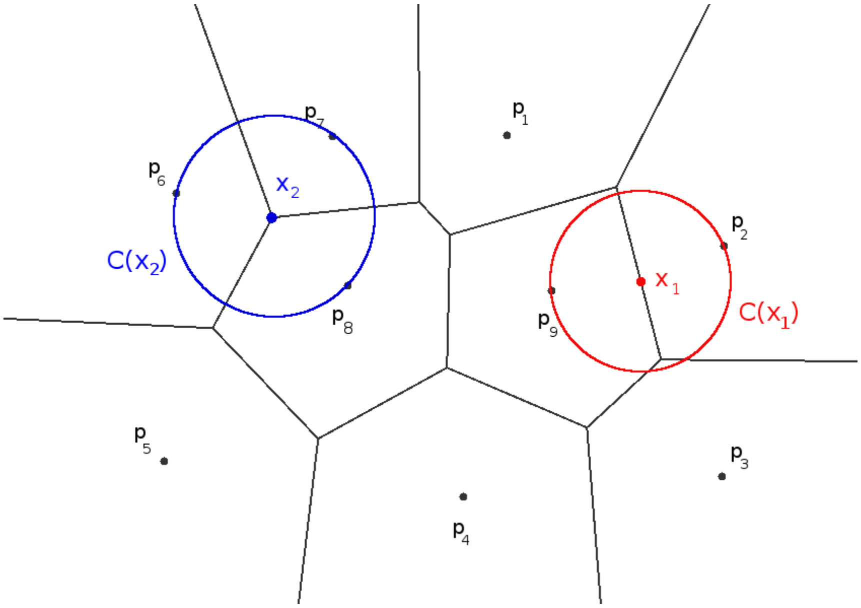


Figure 2.8: Voronoi diagram with largest empty circles for two points.

unbounded.

2.2.2 Higher Dimensions

We can extend the definition of Voronoi diagram to higher dimensions in a straightforward way. In this subsection, we briefly describe the Voronoi diagram in d-dimensional space, i.e., $S = \mathbb{R}^d$. Let $P = \{p_1, p_2, p_3, ..., p_n\}$ be a set of n distinct points (also called sites) in $\mathbb{R}^d$. We assume that sites are in general position, i.e., there exists no k-flat containing $k+2$ points nor a k-sphere containing $k+3$ points, for $1 \leq k \leq d-1$. The Voronoi diagram of P is a partition of $\mathbb{R}^d$ into n regions, called Voronoi cells. Each Voronoi cell is associated with a site p_i and is denoted by $V(p_i)$. More precisely, $V(p_i)$ can be defined as

$$\{x \in \mathbb{R}^d \mid \forall p_j \in P : \quad d(x, p_i) \leq d(x, p_j)\},$$

where $d(p, q)$ denotes the Euclidean distance function between two points p and q in $\mathbb{R}^d$.

In three-dimensional space each Voronoi cell is a convex polyhedron and two adjacent Voronoi cells share a Voronoi facet which is convex polygon [4]. Thus, all points on a Voronoi facet are equidistant from two sites. Similarly, we can define Voronoi edges and Voronoi vertices. All points on a Voronoi edge have the same distance from three sites, while a Voronoi vertex is equidistant from four sites. Thus, the Voronoi diagram in three dimensions consists of faces of order 0 (vertices), 1 (edges), 2 (facets), and 3 (cells). In general, the Voronoi diagram in $\mathbb{R}^d$ contains faces of all dimensions from 0 up to d [3]. The complexity of the Voronoi diagram is defined as the total number of faces of all dimensions. It can be proved that the complexity of the d-dimensional Voronoi diagram is $\Theta(n^{\lceil d/2 \rceil})$ [36, 31]. In particular, the Voronoi diagram in $\mathbb{R}^3$ has quadratic complexity.

2.2.3 Algorithms

Various algorithms are proposed for computing the Voronoi diagram in optimal $O(n \log n)$ time in $\mathbb{R}^2$. Shamos and Hoey [65] designed the first optimal algorithm for the computation of the Voronoi diagram. This algorithm is based on the divide-and-conquer technique. Several other optimal divide-and-conquer algorithms were proposed afterwards [32, 27, 42]. Fortune [29] proposed a plane sweep algorithm to compute the Voronoi diagram in $O(n \log n)$ time.

Next, we briefly describe the algorithms for computing the Voronoi diagram in higher dimensions. These algorithms are based on an elegant connection between Voronoi diagrams and convex polyhedra (see Chapter 11 of the textbook by de Berg et al. [18] for the details). Based on this connection and using efficient algorithms for constructing the convex hull, the Voronoi diagram in $\mathbb{R}^d$ can be computed in $O(n \log n + n^{\lceil d/2 \rceil})$ time [12, 14, 63]. Recall that the complexity of the Voronoi diagram in $\mathbb{R}^d$ is $O(n^{\lceil d/2 \rceil})$. Thus, these algorithms are optimal.

2.3 Delaunay Triangulation

Delaunay triangulation is the dual graph of the Voronoi diagram, originally defined by Voronoi [71] by way of the neighbour relationships in the Voronoi diagram. However, they are attributed to Russian mathematician Boris Nikolaevich Delone who provided a more comprehensive definition of the concept and its properties [19]. Similar to the Voronoi diagram, Delaunay triangulation was rediscovered later in other fields, e.g., Smith [67] and Christ et al. [13]. See [52] for more details about the history of the Delaunay triangulation.

2.3.1 Definitions and Properties

Let $P = \{p_1, p_2, p_3, ..., p_n\}$ be a set of n distinct points in the space S. For simplicity we first describe the Delaunay triangulation in two dimensions, i.e., we consider the case $S = \mathbb{R}^2$. We assume that the points are in general position. Recall from Section 2.2 that the Voronoi diagram of P partitions the space into n regions, one for each point $p_i \in P$. We construct the Delaunay triangulation of P as follows. We connect two points p_i and p_j with a straight line if and only if the Voronoi cells $V(p_i)$ and $V(p_j)$ are adjacent, i.e., $V(p_i) \cap V(p_j) \neq \emptyset$. It can be proved [18] that by doing this we get a *triangulation*, i.e., a subdivision of the plane into triangles. Figure 2.9 shows the Delaunay triangulation for the points of Figure 2.7. Alternatively, the Delaunay triangulation of P can be defined as a triangulation $\mathcal{DT}(P)$ such that the circumcircle of any triangle in $\mathcal{DT}(P)$ contains no point of P. Figure 2.10 shows the example of Figure 2.9 together with the circumcircles.

The Delaunay triangulation has several properties. We describe a few important properties here. The proofs are provided in [18, 52].

- The union of all triangles in $\mathcal{DT}(P)$ is the convex hull of P.
- There is an edge between two points $p_i, p_j \in P$ in $\mathcal{DT}(P)$ if and only if there is a closed circle C that contains p_i and p_j on its boundary and does not contain any other point of P.

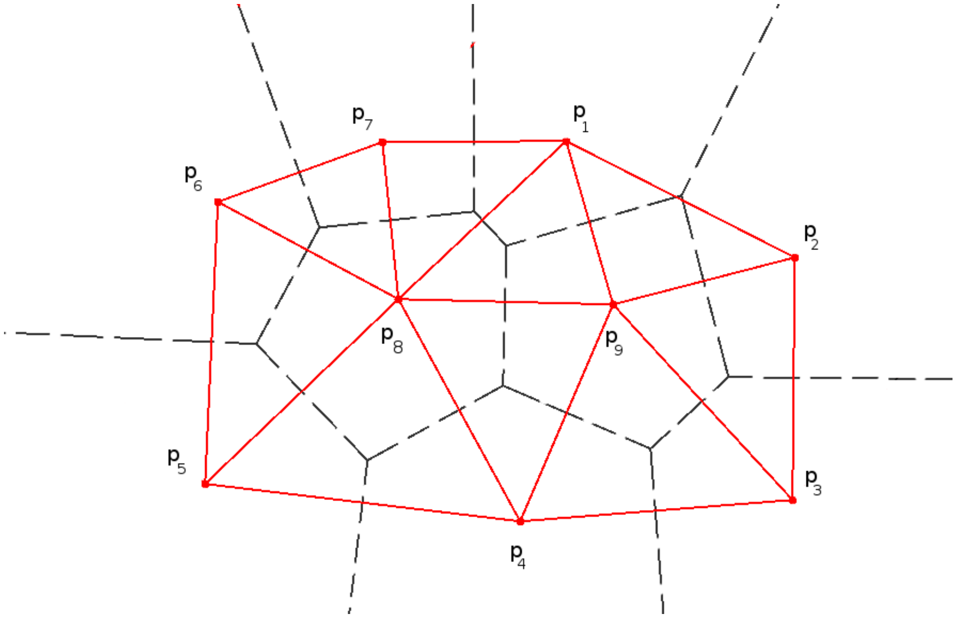


Figure 2.9: The Delaunay triangulation for the points of Figure 2.7.

- The Delaunay triangulation of P maximizes the minimum angle over all triangulations of P.
- Each point in $\mathcal{DT}(P)$ has six surrounding triangles on average.

2.3.2 Higher Dimensions

The Delaunay triangulation concept can be extended to $d > 2$ dimensions. Since triangulation is a two-dimensional geometric notion, the corresponding d-dimensional structure is called a *Delaunay tessellation* for $d \geq 3$. Let $P = \{p_1, p_2, p_3, \ldots, p_n\}$ be a set of n points in general position in $\mathbb{R}^d$. The Delaunay tessellation of P, denoted by $\mathcal{DT}(\mathcal{P})$, is a partition of $\mathbb{R}^d$ into a set of simplices such that circumhypersphere of any simplex in $\mathcal{DT}(\mathcal{P})$ contains no point of P. The Delaunay tessellation of a set of n point in $\mathbb{R}^d$ has at most $O(n^{\lceil d/2 \rceil})$ simplices [64]. In Chapter 3 we compute the Delaunay tessellation of a set P of points in

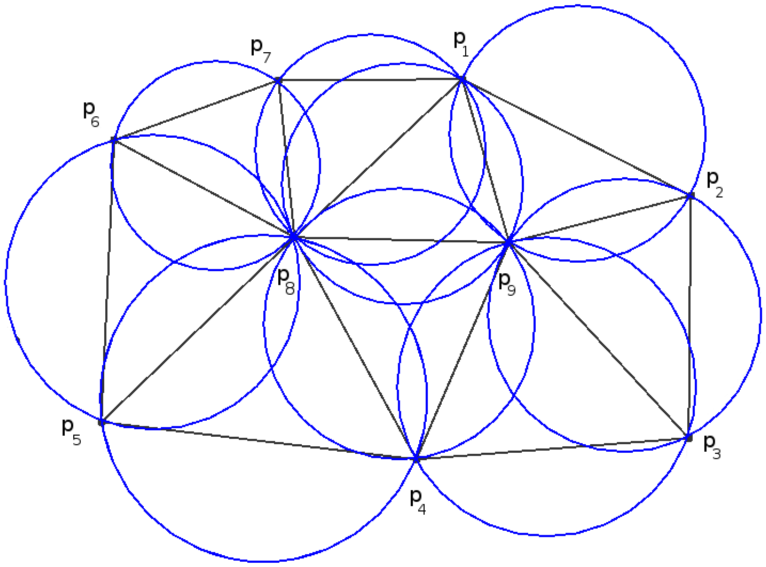


Figure 2.10: The Delaunay triangulation for the points of Figure 2.7 with the circumcircles.

three-dimensional space. In $\mathbb{R}^3$, $\mathcal{DT}(\mathcal{P})$ partitions the space into a set of 3-simplices, i.e., a set of tetrahedra. A tetrahedron t belongs to $\mathcal{DT}(\mathcal{P})$ if and only if the sphere passing through vertices of t does not contain any point of P. The number of tetrahedra in $\mathcal{DT}(\mathcal{P})$ is at most $O(n^2)$.

2.3.3 Algorithms

There is a close relationship between algorithms for the computation of the Delaunay triangulation and algorithms for computing the Voronoi diagram. If we have the Voronoi diagram for a set P of n points in $\mathbb{R}^2$, then we can compute $\mathcal{DT}(\mathcal{P})$ in $O(n)$ time. Recall from Subsection 2.2.3 that the Voronoi diagram of n points in the plane can be computed in $O(n \log n)$ time [65, 29, 32, 27, 42]. Thus we can compute the Delaunay triangulation of n points in $\mathbb{R}^2$ in $O(n \log n)$ time. Similarly, we can use algorithms for computing the Voronoi diagram of a set P in higher dimensions to compute the Delaunay tessellation of P. Recall that the Voronoi diagram of a set of n points in $\mathbb{R}^d$ can be computed in $O(n \log n + n^{\lceil d/2 \rceil})$ time [12, 14, 63]. Thus, we can compute the Delaunay tessellation of a set of n points in $\mathbb{R}^d$ in $O(n \log n + n^{\lceil d/2 \rceil})$ time.

2.4 Dijkstra's Algorithm

Dijkstra's algorithm is a greedy algorithm for finding shortest paths in a weighted graph, proposed by Edsger Dijkstra in 1959 [20]. In this problem we are given a weighted directed graph $G = (V, E)$, with weight function $w : E \to R$, and a source vertex $s \in V$. We want to find the shortest paths from s to all vertices of G. The weight of a path P from s to a vertex $v \in V$ is defined as:

$$w(P) = \sum_{e \in P} w(e).$$

The shortest path from s to v is a path from s to v with minimum weight. In Dijkstra's algorithm [20, 16] we maintain a set S of selected vertices for which we have found the shortest path from the source. We also define a variable $\delta(v)$ for each vertex $v \in V$ as the weight of the shortest path from s to v that only uses the elements in S as intermediate vertices. Initially, we have $S = \emptyset$, $\delta(s) = 0$, and $\delta(v) = \infty$ for each $v \neq s$. At each step we pick a vertex $v \in V \setminus S$ with minimum $\delta(v)$ and insert it into S. We also update the δ values for the neighbours of v. The pseudocode for this algorithm is shown in Figure 2.11.

Next we analyze the running time of Dijkstra's algorithm in terms of $n = |V|$ and $m = |E|$. We can use a priority queue to implement Dijkstra's algorithm. More specifically, the vertices of $V \setminus S$ are stored in a min-priority queue where the priority of each element v is $\delta(v)$. At each step, we use a delete-min operation to pick the vertex u with minimum $\delta(u)$. Observe that each vertex is picked exactly once and therefore we have $O(n)$ delete-min operations. Then we update the δ values of neighbours of u. Each update might lead to a decrease-key operation. Thus, for each vertex u, we can have up to $O(deg(u))$ decrease-key operations. Since each vertex is processed exactly once, the total number of decrease-key operations is $O(\sum_{u \in V} deg(u)) \in O(2m) \in O(m)$. The initialization takes $O(n)$ time. We

```
Let $G = (V, E), w : E \rightarrow R$ be a directed weighted graph
Let $s$ be the source vertex
$S \leftarrow \emptyset$
for $v \in V$
    $\delta(v) \leftarrow \infty$
$\delta(s) \leftarrow 0$
while $S \neq V$
    $u \leftarrow$ a node in $V \setminus S$ with the minimum $\delta(u)$
    for each neighbor $(v \notin S)$ of $u$
        if $\delta(v) > \delta(u) + w(uv)$
            $\delta(v) \leftarrow \delta(u) + w(uv)$
    add $u$ to $S$
```

Figure 2.11: Dijkstra's algorithm.

can implement the priority queue using different data structures. If we use standard heaps, the running time of both delete-min and decrease-key operations is $O(\log n)$ and the total running time will be $O(n + n \log n + m \log n) \in O(m \log n)$. On the other hand, if we use Fibonacci heaps [30] the amortized running time of delete-min and decrease-key operations is $O(\log n)$ and $O(1)$, respectively. Therefore the total running time of the Dijkstra's algorithm using Fibonacci heap will be $O(n + n \log n + m) \in O(m + n \log n)$.

Chapter 3

Methods and Algorithms

In this chapter we describe algorithms for finding and widening tunnels in protein structures. Given a fixed conformation of a protein with a starting point, we first identify and visualize the widest tunnel leading from that point to the outside environment. Then we extend this algorithm and explore the possibility that a slight local change in the structure of the protein can lead to a wider tunnel. More specifically, we consider various alternative conformations of the initial structure whose energies are not much higher than the energy of the initial conformation and select the one with the widest tunnel. In the next step, we attempt to verify that the alternative conformation with the widest tunnel, called the target conformation, is accessible from the initial conformation. In other words, we want to check whether the change in the structure of the protein is feasible. We propose and compare several algorithms for verifying the accessibility of the target conformation from the initial conformation.

Atom	Radius (Å)
Hydrogen (H)	1.20
Carbon (C)	1.70
Nitrogen (N)	1.55
Oxygen (O)	1.52
Sulfur (S)	1.80
Phosphorus (P)	1.80
Potassium (K)	2.75
Iodine (I)	1.98

Table 3.1: Van der Waals radii of protein atoms [6].

3.1 Finding the Widest Tunnel in a Static Protein Structure

Recall the tunnel-finding problem defined in Chapter 1: we are given a fixed conformation of a protein and the coordinates of a starting point. We want to find the widest tunnel from that point to the outside environment of the protein. The starting point is considered to be a single point inside the protein structure whose coordinates are provided by the user and the outside environment is anywhere outside the convex hull of the protein atoms. As stated in Section 1.1, CAVER2 [49] proposed the idea of using a Delaunay tessellation for finding the widest tunnel, but the paper only describes the idea in two dimensions. For completeness, we provide an overview of the algorithm in three dimensions. The protein molecule is represented as a set of spheres, where each sphere corresponds to a single protein atom. The radii of the spheres are set to the van der Waals radii of corresponding atoms. The van der Waals radii of typical atoms constituting biomolecules, taken from Bondi's compilation [6], are shown in Table 3.1. We want to find a route T from the starting point to the outside environment such that the ligand can pass through T without any clash with the protein atoms. Note that the ligand does not necessarily have a spherical shape. Thus the orientation of the ligand during its movement influences the feasibility of a tunnel: a specific orientation might lead to clash while another orientation passes through the tunnel

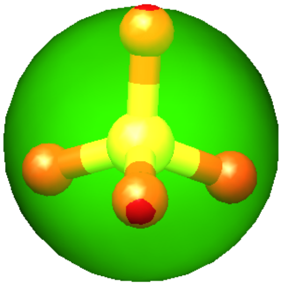


Figure 3.1: A sphere enclosing all ligand atoms.

without any clash. Modeling these changes in the orientation of the ligand or changes in the shape of the ligand while passing through the tunnel is very complicated and beyond the scope of this thesis. Therefore, following CAVER2 we model the ligand by a sphere which encloses all the ligand atoms (see Figure 3.1). If the enclosing sphere of a ligand can pass through a tunnel T without any clash, then we conclude that any orientation of the ligand can safely pass through T. Note that, most of the time, the ligand is a simple ion with a spherical shape. In these cases we do not need the above simplification.

The tunnel can be represented by its centerline, which is a curve connecting the starting point to a point located outside the convex hull of the protein atoms. For each point p on the centerline, the width of the tunnel at p, denoted by $w(p)$, is the smallest distance from p to the van der Waals surfaces of nearby protein atoms. In other words, $w(p)$ is the radius of the largest sphere centered at p that does not clash with protein atoms.

We define the width of tunnel T, denoted by $w^*(T)$, the minimum $w(p)$ for all points p on the centerline of T, i.e., the width of tunnel at its narrowest part. Observe that a sphere of radius at most $w^*(T)$ can safely pass through T. The tunnel-finding algorithm consists of three steps.

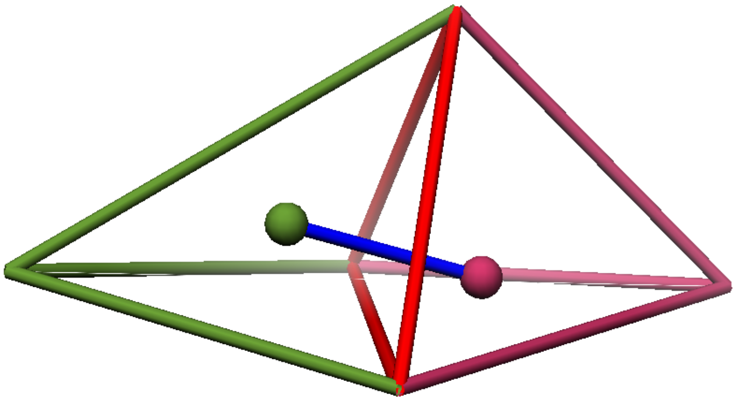


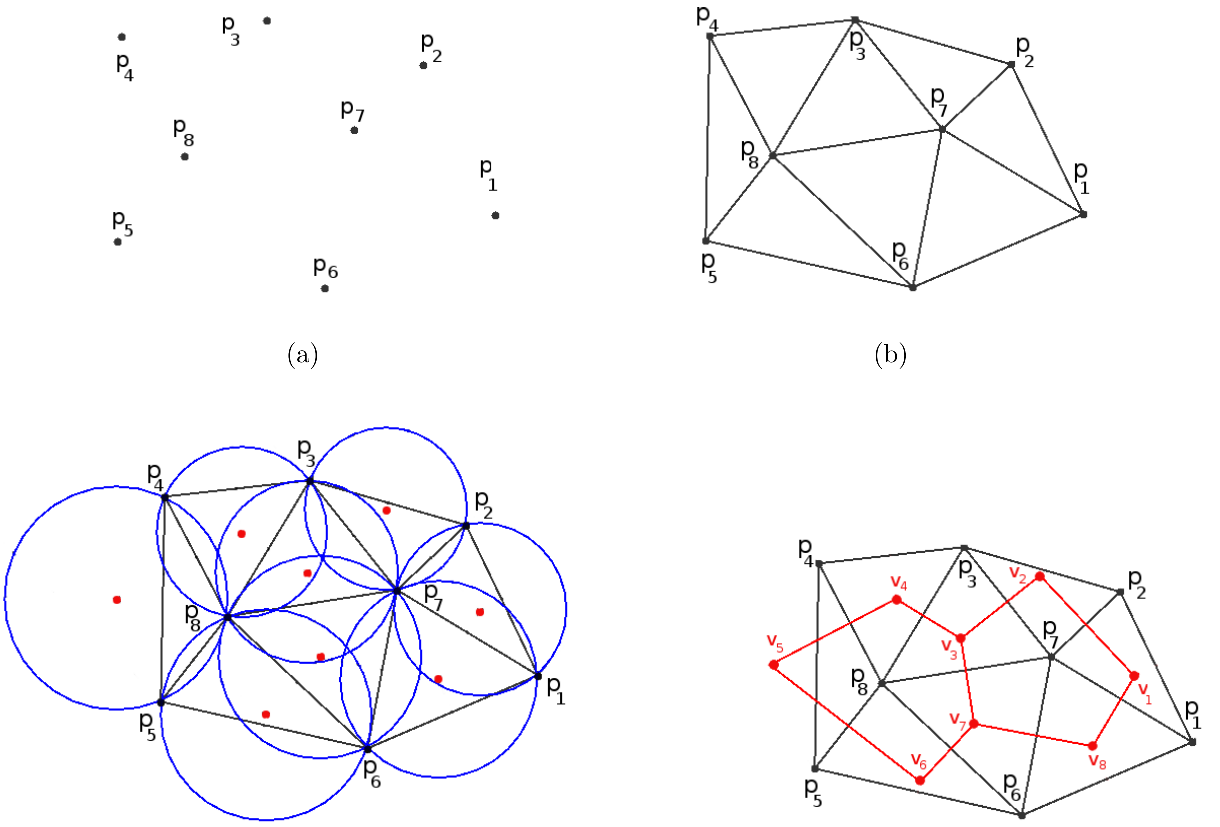
Figure 3.2: Two adjacent tetrahedra in the Delaunay tessellation.

3.1.1 Computing the Delaunay Tessellation

Let P be the set of center points of atoms in the given protein conformation. In the first step, we compute the Delaunay tessellation of P. Recall from Section 2.3 that this tessellation partitions the space into a set of tetrahedra such that any sphere circumscribing a tetrahedron does not contain any point of P in its interior.

3.1.2 Constructing a Graph

In the next step we construct an undirected weighted graph G. The vertices of G correspond to the tetrahedra computed in the Delaunay tessellation. More specifically, for each tetrahedron we consider the center of the sphere that passes through its four vertices. We add an edge between any two vertices of the graph whose corresponding tetrahedra are adjacent, i.e., they have a common face. The weight of the edge is the width of the path between two tetrahedra centers that avoids all other atoms. Consider an edge between the vertices corresponding to two adjacent tetrahedra (see Figure 3.2). These two tetrahedra have a common face, i.e., three common atoms. We compute the radius of the circle that



(a)

(b)

(c)

(d)

Figure 3.3: The first two steps of the tunnel-finding algorithm for a set of 8 points in $\mathbb{R}^2$. (a) A set P of 8 points in $\mathbb{R}^2$. (b) The Delaunay triangulation of P. (c) The vertices of the graph G are centers of the circumcircles of the triangles in the Delaunay triangulation. (d) The graph G for set P.

passes through the centers of these three atoms and then reduce it by the maximum van der Waals radius of the three atoms. Observe that the weight of an edge uv, denoted by $weight(uv)$, shows the width of a route from u to v that does not clash with protein atoms. The narrowest part of this path is on the common face of the two tetrahedra correspond to u and v.

Figure 3.3 shows the first two steps of the tunnel-finding algorithm on a set of 8 points For the sake of illustration we have shown the example in $\mathbb{R}^2$.

3.1.3 Finding the Optimal Path in the Graph

A tunnel T corresponds to a path $\pi(T)$ in G. The width of T equals the minimum weight of edges of $\pi(T)$. Therefore, we define the weight of a path as the minimum weight of its edges and our objective is to find the path with maximum weight. First we find a vertex s of G whose corresponding tetrahedron center has the smallest distance to the starting point. We want to find a path of maximum weight from s to a boundary vertex of G. A vertex is a boundary vertex if it is located outside the convex hull of the protein atoms. Let Π be the set of all paths from s to boundary vertices of G. We want to solve the following optimization problem:

$$\max_{\pi \in \Pi} \min_{uv \in \pi} weight(uv).$$

We use a variant of the Dijkstra algorithm (described in Section 2.4) to find a path of maximum weight. For each vertex u of graph G we maintain a width value, denoted by $width(u)$, holding the width of the current widest path from s to u. In other words we define a mapping of vertices to real numbers:

$$width : V(G) \to \mathbb{R}.$$

Initially, we assign a width of $+\infty$ to s and width of -1 to each other vertex of the graph. We also maintain a set S of selected vertices. Initially, no vertex is selected and we have

```
Let $G = (V, E)$ be an undirected weighted graph
Let $s$ be the source vertex and $B$ be the set of boundary vertices of $G$
    $S \leftarrow \emptyset$
    for $v \in V$
        $prev[v] \leftarrow nil$
        $width(v) \leftarrow -1$
    $width(s) \leftarrow +\infty$
    while $S \cap B = \emptyset$
        $u \leftarrow$ a node in $V \setminus S$ with the maximum width
        for each neighbor $(v \notin S)$ of $u$
            if $width(v) < \min\{width(u), weight(uv)\}$
                $width(v) \leftarrow \min\{width(u), weight(uv)\}$
                $prev[v] \leftarrow u$
        add $u$ to $S$
    $\pi \leftarrow \emptyset$
    $v \leftarrow S \cap B$
    while $prev[v] \neq nil$
        insert $v$ at the beginning of $\pi$
        $v \leftarrow prev[v]$
    return $\pi$
```

Figure 3.4: A greedy algorithm to find the best path in a graph.

$S = \emptyset$. Then at each step we select an unselected vertex u with maximum width and update the widths of its neighbours as follows: For each neighbour v of u we check whether we can find a wider tunnel from s to v through u. Observe that the width of the tunnel from s to v that passes through u is

$$\min\{width(u), weight(uv)\}.$$

If this width is better (larger) than the current width of v we update the width of v and set its predecessor (shown by $prev[v]$) to u. We continue this process until we select a boundary vertex w. Then we use $prev$ fields to recover the optimal path from s to w. The pseudocode for this greedy algorithm is shown in Figure 3.4.

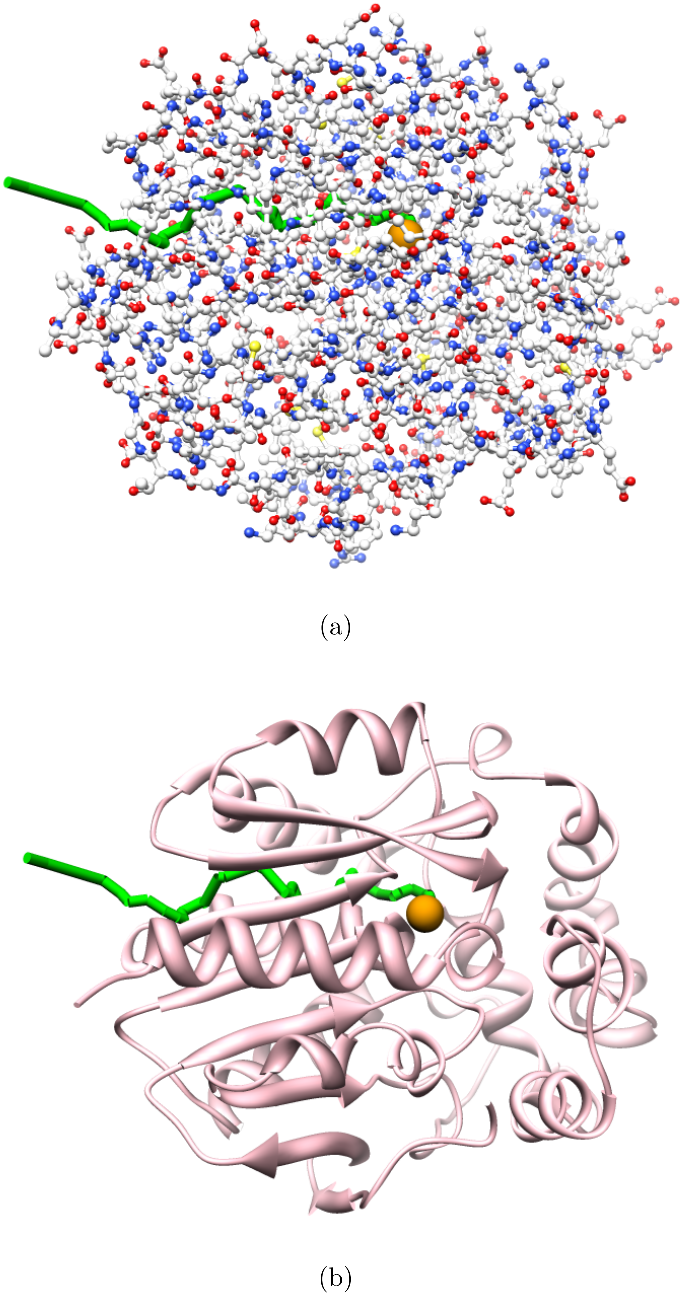
Figure 3.5 shows the widest tunnel found by this algorithm in protein 1CV2 with the starting point having coordinates (14,15,22). The width of the corresponding tunnel is 0.43 Å. More details about this example and other results will be provided in Chapter 4.

3.1.4 Runtime Complexity

In this subsection we analyze the running time of the tunnel-finding algorithm. Let n be the number of atoms in the given protein conformation. In the first step we compute the Delaunay tessellation of n points in $\mathbb{R}^3$. Recall from Section 2.3 that the Delaunay tessellation of a set of n points in $\mathbb{R}^d$ can be computed in $O(n \log n + n^{\lceil d/2 \rceil})$ time and has $O(n^{\lceil d/2 \rceil})$ simplices. Therefore the Delaunay tessellation of atom centers can be computed in $O(n^2)$ time and partitions the space into $O(n^2)$ tetrahedra. In the next step we construct a graph G as described in Subsection 3.1.2. We can compute the center of each tetrahedron in constant time. Therefore the vertices of graph G can be computed in time $O(n^2)$. Observe that each tetrahedron is adjacent to at most four other tetrahedra in the tessellation. Thus, the degree of each vertex in G is at most four and number of edges in this graph is

$$m = \frac{\sum_{v \in V} deg(v)}{2} \in O(4n^2/2) \in O(n^2).$$

Therefore G has $O(n^2)$ vertices and $O(n^2)$ edges. From the representation of the Delaunay tessellation, we can compute the edges of G in time $O(m) \in O(n^2)$. The weight of each edge can be computed in constant time. Thus the first two steps of the tunnel-finding algorithm can be done in $O(n^2)$ time. The last step is the greedy algorithm that finds the optimal path in G. Consider the pseudocode of the greedy algorithm in Figure 3.4. The initialization (line 1-5) can be done in $O(|V|) \in O(n^2)$. At each iteration of the first while loop one vertex is added to S. Therefore we have at most $O(n^2)$ iterations. We can use a max-priority queue to maintain the vertices in $V \setminus S$, where priority of each vertex u is $width(u)$. Thus at each iteration of the first while loop we have one delete-max operation (in line 7) to pick a vertex u. Then at lines 8-11 we update the weights of the neighbours of u. Each such update might lead to an increase-key operation. Therefore we can have up to $O(deg(u))$



(a)

(b)

Figure 3.5: The widest tunnel starting at position (14,15,22) in protein with PDB ID 1CV2. (a) Protein atoms represented using ball and stick option in Chimera. (b) Overall structure of the protein represented using ribbon option in Chimera.

increase-key operations. Since each vertex of G is processed at most once, the total number of delete-max and increase-key operations is $O(n^2)$ and $\sum_{u \in V} O(deg(u)) \in O(2m) \in O(n^2)$. If we use standard heap to implement the priority queue the running time of both increase-key and delete-max operations is the same and equals $O(\log |V|) \in O(\log n^2) \in O(\log n)$. Thus the total running time of the first while loop is $O(n^2 \log n)$. The last while loop is executed $O(n^2)$ times and takes constant time per iteration. Hence, the total running time of the greedy algorithm is $O(n^2 + n^2 \log n + n^2) \in O(n^2 \log n)$ time. Therefore, the tunnel-finding algorithm computes the widest tunnel in a protein with n atoms in $O(n^2 + n^2 + n^2 \log n) \in O(n^2 \log n)$ time.

3.2 Widening the Tunnel Using Alternative Conformations

Using the techniques just described, we can find the widest tunnel in a fixed protein conformation from a given starting point. While this is the widest tunnel in the given conformation, it is possible that there exists a wider tunnel from the starting point to the outside environment in a different conformation of the same protein. In this section we consider *widening* the tunnel, i.e., searching for an alternative conformation of the initial structure with a wider tunnel. More specifically, we develop an algorithm for the tunnel-widening problem, defined and motivated in Chapter 1.

To solve the tunnel-widening problem, we investigate the possibility that a small change in the structure of the protein can lead to a wider tunnel. In other words, we want to reposition some atoms in order to widen the tunnel. Intuitively, the most relevant candidates for relocation are the *bottleneck atoms*, i.e., the atoms that constitute the narrowest part of the tunnel. Therefore, we consider alternative conformations obtained by local changes in the structure of bottleneck region (amino acid residues containing bottleneck atoms) in the initial conformation.

The *tunnel-widening algorithm* first finds the bottleneck atoms of the tunnel (see Figure 3.6). Then it selects the side-chains of their corresponding amino acid residues, called the

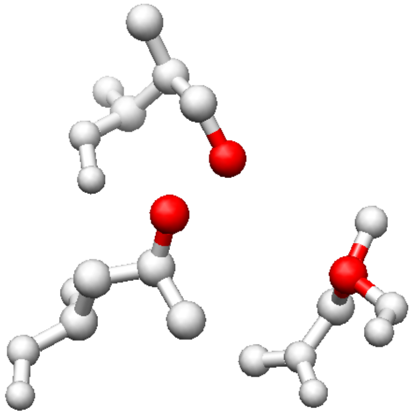


Figure 3.6: The bottleneck atoms (shown in red) and their corresponding residues.

bottleneck side-chains. For each bottleneck side-chain, we obtain an alternative conformation by replacing the side-chain with one of its rotamers as described below. We select the rotamer that has the highest probability of occurrence according to the Dunbrack backbone-dependent rotamer library [23]. Then we make sure that the corresponding rotamer does not clash with other protein atoms. Two atoms are considered to have a clash if their van der Waals spheres overlap by more than a cutoff amount. We used 0.6 Å as the cutoff bound.[1] We select the rotamer with the highest probability that does not have a clash. Figure 3.7 shows how a bottleneck side-chain is replaced by one of its rotamers. Observe that bond lengths and bond angles do not change by this replacement. Therefore, all alternative conformations have bond lengths and bond angles that are the same as the initial conformation. The only difference between these conformations is in the dihedral (chi) angles of the bottleneck side-chains. We then run the tunnel-finding algorithm on each alternative conformation and check whether we can find a wider tunnel. However, it is possible that some alternative conformations have energy values much higher than the energy of the initial conformation. Therefore, the probability of transition from the initial conformation to these alternative conformations is very low. Thus we restrict our attention to *acceptable* alternative conformations, i.e., conformations whose energy values

[1] This is the default value used in Chimera software for clash recognition.

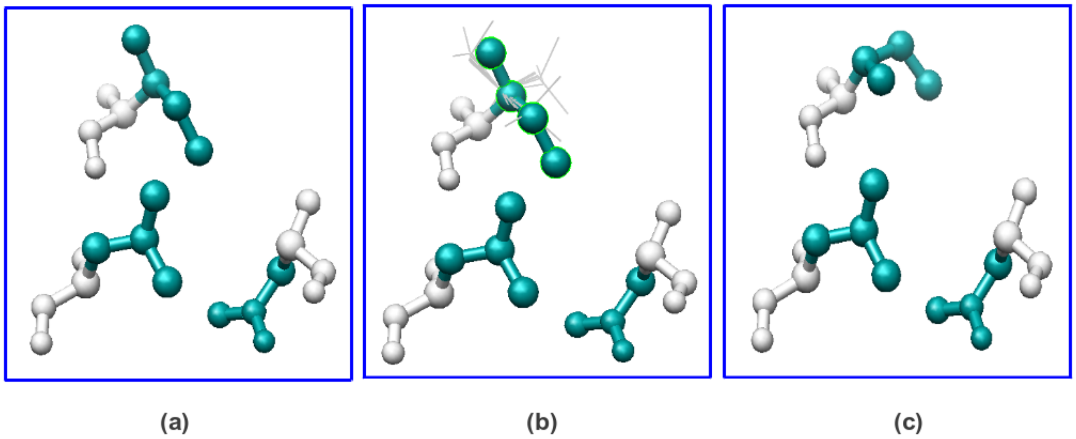


Figure 3.7: Replacing a bottleneck side-chain by one of its rotamers. (a) The bottleneck side-chains are shown in blue. (b) The set of rotamers is shown for the top bottleneck side-chain. (c) The top bottleneck side-chain is replaced by the rotamer that does not have clash with protein atoms and has the highest probability.

are not higher by more than a cutoff parameter when compared with the energy of the initial conformation. The cutoff parameter is set such that an acceptable conformation can be reached from the initial conformation with a reasonable probability. We can select the cutoff parameter based on the Boltzmann's distribution (see Section 2.1.4). Assume that we have two conformations C_1 and C_2, where C_i has energy E_i and population N_i. Table 3.2 shows the relative population $\frac{N_2}{N_1}$ for different values of $\Delta E = E_2 - E_1$ at the temperature $T = 310K$ (body temperature). According to this table, for ΔE =4 kcal/mol the relative population is

$$\frac{N_2}{N_1} = e^{\frac{-4}{0.0019872041 \times 310}} = 0.15\%,$$

which is a reasonable relative population. Note that N_2/N_1 increases in higher temperatures. Therefore we set the cutoff parameter to 4 kcal/mol.

Recall that the widest tunnel found in the protein 1CV2 with the starting point at position (14,15,22) (shown in Figure 3.5) has width 0.43 Å. One of the bottleneck side-chains of this tunnel is the side-chain of residue ASP 108.A. The side-chain dihedral angles

ΔE	N_2/N_1
1	19.72%
2	3.89%
3	0.77%
4	0.15%

Table 3.2: The relative population $\frac{N_2}{N_1}$ for different values of $\Delta E = E_2 - E_1$ (in kcal/mol) at the temperature $T = 310K$.

of this residue in the initial conformation are $\chi_1 = -169.41°$ and $\chi_2 = 74.38°$. By replacing the side-chain of this residue by the rotamer with dihedral angles $\chi_1 = -166.20°$ and $\chi_2 = 11.10°$, we identified a tunnel with width 0.59 Å. The potential energy of the structure changed from -410.820 to -409.563 kcal/mol. Thus the alternative conformation is acceptable and has a wider tunnel. Figure 3.8 shows the widest tunnel in this alternative conformation. More results will be provided in Chapter 4.

3.2.1 Runtime Complexity

In this subsection we analyze the running time of the tunnel-widening algorithm on a protein with n atoms. In the first step we run the tunnel-finding algorithm to compute the widest tunnel T in the given protein conformation. In Subsection 3.1.4 we showed that this can be done in $O(n^2 \log n)$ time. Next we find the bottleneck atoms of T by traversing the edges of $\pi(T)$ (the path in G corresponding to T) and selecting the edge with minimum weight. Since $\pi(T)$ can have at most $O(|V|) \in O(n^2)$ edges, this can be done in $O(n^2)$ time. This gives us three bottleneck side-chains. For each bottleneck side-chain we can compute the best rotamer as described in Section 3.2. Since Dunbrack library contains a constant number of rotamers for each amino acid side-chain, we have constant number of options. For each rotamer we can check whether it has a clash with protein atoms in time $O(n)$. Therefore, finding an alternative conformation of the initial structure takes $O(n)$ time. Then we find the widest tunnel in this alternative conformation in time $O(n^2 \log n)$. Thus the tunnel-widening algorithm takes $O(n + n^2 \log n) \in O(n^2 \log n)$ time for each bottleneck

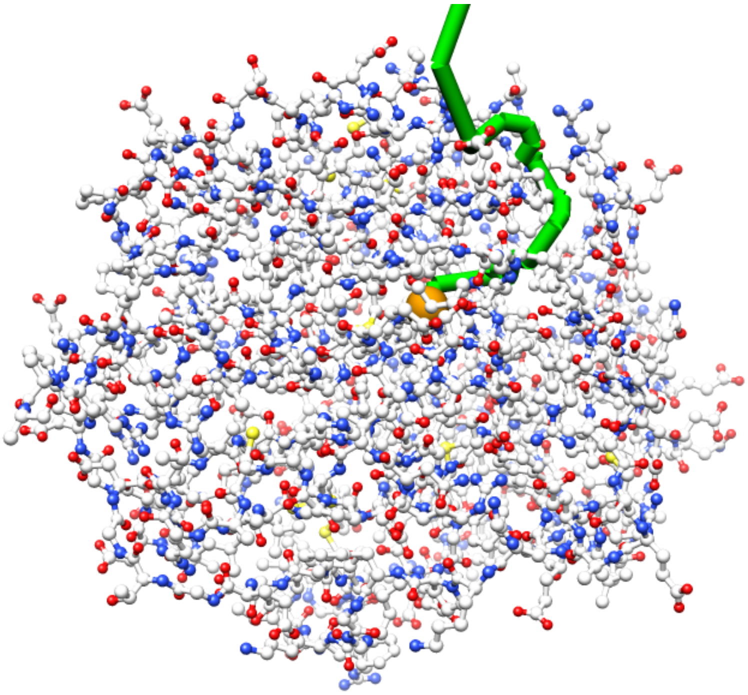


Figure 3.8: The widest tunnel starting at position (14,15,22) in an alternative conformation of protein with PDB ID 1CV2.

side-chain. Since the tunnel T has three bottleneck atoms, the total running time of the tunnel-widening algorithm is $O(n^2 \log n + n^2 + 3 \times n^2 \log n) \in O(n^2 \log n)$.

3.3 Finding Feasible Transition Pathways between Two Protein Conformations

In Section 3.1 we described an algorithm for finding the widest tunnel from a starting point to the outside environment of a fixed protein conformation. Furthermore, the possibility of finding a wider tunnel by a slight local change in the structure of the protein was explored in Section 3.2. In most cases, we can find a wider tunnel in an alternative conformation of the initial structure whose energy is not much higher than the energy of the original conformation. For instance, we found a tunnel of width 0.59 Å in an alternative conformation of protein with PDB ID 1CV2 (see Figure 3.8), while the widest tunnel in the initial conformation had width 0.43 Å (see Figure 3.5). The next step is to ensure that this conformation with the wider tunnel, called the target conformation, is accessible from the initial conformation. In other words, we attempt to find a *transition pathway*, i.e., a sequence $C_0, C_1, \ldots, C_n$ of conformations such that

1. C_0 and C_n are the initial and target conformations, respectively.
2. $C_1, C_2, \ldots, C_{n-1}$ are the intermediate conformations.
3. Each two consecutive conformations, i.e., C_i and C_{i+1} have sufficiently similar structures, possibly based on some user-defined parameters.

Furthermore, we should make sure that the pathway is *feasible*, i.e., the energies of intermediate conformations $C_1, C_2, \ldots, C_n$ are not much higher than the energy of the initial conformation.

We propose several pathway-finding algorithms, i.e., algorithms for finding a feasible transition pathway from the initial to the target conformation. These algorithms are especially designed for our setting, i.e., they use the fact that the only difference between the two conformations is in the dihedral angles of a single side-chain. We refer to this side-chain as the *special side-chain*. For example, the only difference between the two conformations of protein with PDB ID 1CV2 shown in Figures 3.5 and 3.8 is in the side-chain dihedral angles of residue ASP 108.A. Let $\chi_1^0, \chi_2^0, \chi_3^0, \chi_4^0$ be the dihedral (chi) angles of the special side-chain in C_0 and $\chi_1^n, \chi_2^n, \chi_3^n, \chi_4^n$ be the dihedral angles of the special side-chain in C_n.[2] Note that for some amino acid residues we have less than four side-chain dihedral angles. For instance, the special side-chain of ASP 108.A only has two dihedral angles and we have $\chi_1^0 = 169.41°, \chi_2^0 = 74.38°, \chi_1^n = 166.20°$, and $\chi_2^n = 11.10°$. The intermediate conformations discovered by our algorithms have the same structure as C_0 except for the dihedral angles of the special side-chain.

3.3.1 Averaging Algorithm

The first algorithm is deterministic and based on the idea of averaging the dihedral angles of the special side-chain. First we find the intermediate conformation C_i by averaging the chi angles of the special side-chain in the initial and target conformations. More specifically, if $\chi_1^i, \chi_2^i, \chi_3^i, \chi_4^i$ are the chi angles of the special side-chain in C_i, then we have:

$$\chi_j^i = \frac{\chi_j^0 + \chi_j^n}{2}, \quad j = 1, 2, 3, 4$$

Therefore, we obtain an intermediate conformation C_i between C_0 and C_n and we have a partial pathway C_0, C_i, C_n. In the next step we find an intermediate conformation between any two consecutive conformations of the partial pathway, i.e., one intermediate conformation between C_0 and C_i and another one between C_i and C_n. We continue this process until we find as many intermediate conformations as we want (the parameter n that shows the number of intermediate conformations and reflects the trade-off between running time and accuracy). Observe that this approach is equivalent to gradually (and

[2] Recall from Chapter 2 that each side-chain can have at most five chi angles. The only side-chain with five chi angles belongs to ARG. However, χ_5 of ARG is always 180° or 0° and thus most rotamer libraries including Dunbrack rotamer library only consider at most four chi angles.

```
Let $n$ be the number of intermediate conformations (a parameter)
Let $\chi_1^0, \chi_2^0, \chi_3^0, \chi_4^0$ be the chi angles of the special side-chain in the initial conformation
Let $\chi_1^n, \chi_2^n, \chi_3^n, \chi_4^n$ be the chi angles of the special side-chain in the target conformation
    $P = \emptyset$
    for $j \leftarrow 1$ to 4
        increase[$j$] $\leftarrow (\chi_j^n - \chi_j^0)/n$
    for $k \leftarrow 1$ to $n$
        for $j \leftarrow 1$ to 4
            $\chi_j^k \leftarrow \chi_j^0 + k \times$increase[$j$]
        add the intermediate conformation $C_k$ with chi angles $\chi_1^k, \ldots, \chi_4^k$ to $P$
    return $P$
```

Figure 3.9: The averaging algorithm for finding a pathway between two conformations.

linearly) changing the chi angles of the special side-chain from the initial chi angles to the target chi angles. The pseudocode of this algorithm is shown in Figure 3.9. Thus we find a transition pathway $p = C_0, C_1, \ldots, C_n$ from the initial to the target conformation. To verify the feasibility of p, we test that the energy of each intermediate conformation is not much higher than the energy of the initial conformation. Observe that if the number of intermediate conformations is large enough, consecutive conformations will be quite similar in structure and so the chance of having a high energy barrier between them is low.

This algorithm considers just a single pathway between the initial and target conformations and checks if the pathway is feasible. Hence it is possible that the pathway computed by this algorithm is not feasible, while a feasible transition pathway exists. However, surprisingly for most of our test cases, this algorithm works well and can find a feasible pathway from the initial to the target conformation. Table 3.3 shows a feasible pathway found by this algorithm (with parameter n set to 25) between the two conformations of protein with PDB ID 1CV2 shown in Figures 3.5 and 3.8.

The feasible pathway contains 26 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-169.41 , 74.38]	-410.81966760
C_1	[-169.29 , 71.85]	-410.79615608
C_2	[-169.16 , 69.32]	-410.67227634
C_3	[-169.03 , 66.79]	-409.98966834
C_4	[-168.90 , 64.26]	-409.59712804
C_5	[-168.77 , 61.73]	-409.54350222
C_6	[-168.64 , 59.20]	-409.50703560
C_7	[-168.51 , 56.66]	-409.49178566
C_8	[-168.39 , 54.13]	-409.50039557
C_9	[-168.26 , 51.60]	-409.53186945
C_{10}	[-168.13 , 49.07]	-409.58054829
C_{11}	[-168.00 , 46.54]	-409.63909832
C_{12}	[-167.87 , 44.01]	-409.69835481
C_{13}	[-167.74 , 41.48]	-409.74717519
C_{14}	[-167.61 , 38.95]	-409.77577038
C_{15}	[-167.49 , 36.41]	-409.77472919
C_{16}	[-167.36 , 33.88]	-409.73670150
C_{17}	[-167.23 , 31.35]	-409.65621001
C_{18}	[-167.10 , 28.82]	-409.65514503
C_{19}	[-166.97 , 26.29]	-409.74859384
C_{20}	[-166.84 , 23.76]	-409.79650075
C_{21}	[-166.71 , 21.23]	-409.80433784
C_{22}	[-166.59 , 18.69]	-409.77787602
C_{23}	[-166.46 , 16.16]	-409.72413779
C_{24}	[-166.33 , 13.63]	-409.65031759
C_{25}	[-166.20 , 11.10]	-409.56332785

Table 3.3: A feasible transition pathway found by the averaging algorithm between two conformations of protein with PDB ID 1CV2.

3.3.2 Randomized Algorithm

Recall that in the previous algorithm we only considered the pathway obtained by changing the dihedral angles of the special side-chain linearly. In this section we use randomization to find an alternative pathway that might be more desirable. As before we only change the chi angles of the special side-chain. First we find the random intermediate conformation C_i as follows. The chi angles of the special side-chain in C_i $(\chi_1^i, \chi_2^i, \chi_3^i, \chi_4^i)$ are selected randomly between the chi angles of the special side-chain in C_0 and C_n:

$$\chi_j^i = random(\chi_j^0, \chi_j^n), \quad j = 1, 2, 3, 4,$$

where $random(a, b)$ denotes a number between a and b selected uniformly at random. Therefore, we obtain an intermediate conformation C_i between C_0 and C_n and we have a partial pathway C_0, C_i, C_n. In the next step we find a random intermediate conformation between any two consecutive conformations of the partial pathway, i.e., one intermediate conformation between C_0 and C_i and another one between C_i and C_n. We continue this process until some stopping criterion holds. We used the following criterion: we stop if the difference between the chi angles of the special side-chain in every two consecutive conformations is smaller than a predefined threshold, denoted by *diff*. Finally, we check whether the discovered pathway is feasible as before. The pseudocode for this approach is shown in Figure 3.10.

Table 3.4 shows a feasible pathway found by this algorithm (with parameter *diff* set to 8) between the two conformations of protein with PDB ID 1CV2 shown in Figures 3.5 and 3.8.

3.3.3 Greedy Algorithm

In this approach we construct a discrete conformational space and exhaustively search this space to find the best pathway, i.e., a path p from the initial to the target conformation so that the maximum weight among the nodes of p is the smallest possible. The energy of a pathway is defined as the maximum energy of its intermediate conformations and we

```
Let $diff$ be a parameter related to the stopping criterion
Let $\chi_1^0, \chi_2^0, \chi_3^0, \chi_4^0$ be the chi angles of the special side-chain in the initial conformation
Let $\chi_1^n, \chi_2^n, \chi_3^n, \chi_4^n$ be the chi angles of the special side-chain in the target conformation
    $P = \emptyset$
    RandomizedPath($\chi_1^0, \chi_2^0, \chi_3^0, \chi_4^0, \chi_1^n, \chi_2^n, \chi_3^n, \chi_4^n, diff$)
```

```
RandomizedPath($\chi_1, \chi_2, \chi_3, \chi_4, \chi'_1, \chi'_2, \chi'_3, \chi'_4, diff$)
    **if** ($|\chi'_1 - \chi_1| \leq diff$) **and** ($|\chi'_2 - \chi_2| \leq diff$) **and** ($|\chi'_3 - \chi_3| \leq diff$) **and** ($|\chi'_4 - \chi_4| \leq diff$) **then**
        **return**
    **for** $j \leftarrow 1$ to 4
        $\chi''_j$= random($\chi_j, \chi'_j$)
    Add the intermediate conformation $C$ with chi angles $\chi''_1, \ldots, \chi''_4$ to the $P$
    RandomizedPath($\chi_1, \chi_2, \chi_3, \chi_4, \chi''_1, \chi''_2, \chi''_3, \chi''_4, diff$)
    RandomizedPath($\chi''_1, \chi''_2, \chi''_3, \chi''_4, \chi'_1, \chi'_2, \chi'_3, \chi'_4, diff$)
```

Figure 3.10: A randomized algorithm for finding a pathway between two conformations.

The feasible pathway contains 33 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-169.41 , 74.38]	-410.81966760
C_1	[-169.37 , 70.54]	-410.72182894
C_2	[-167.73 , 68.15]	-410.70153355
C_3	[-167.59 , 66.64]	-410.34598284
C_4	[-166.87 , 66.55]	-410.46967395
C_5	[-166.85 , 66.34]	-410.42426539
C_6	[-166.65 , 62.72]	-410.04119105
C_7	[-166.62 , 57.70]	-409.95628561
C_8	[-166.50 , 56.11]	-409.95674797
C_9	[-166.50 , 54.44]	-409.93959565
C_{10}	[-166.49 , 52.78]	-409.92943212
C_{11}	[-166.47 , 47.19]	-409.92775705
C_{12}	[-166.44 , 46.75]	-409.93231049
C_{13}	[-166.44 , 46.25]	-409.93301630
C_{14}	[-166.43 , 46.04]	-409.93423048
C_{15}	[-166.43 , 42.55]	-409.93433533
C_{16}	[-166.43 , 42.08]	-409.93292507
C_{17}	[-166.43 , 41.76]	-409.93175411
C_{18}	[-166.43 , 39.99]	-409.92090589
C_{19}	[-166.43 , 35.99]	-409.85650603
C_{20}	[-166.43 , 33.98]	-409.79718489
C_{21}	[-166.43 , 33.92]	-409.79490550
C_{22}	[-166.43 , 28.41]	-409.67676769
C_{23}	[-166.43 , 28.27]	-409.68192622
C_{24}	[-166.43 , 28.25]	-409.68253844
C_{25}	[-166.43 , 26.81]	-409.72945669
C_{26}	[-166.43 , 26.52]	-409.73745887
C_{27}	[-166.40 , 24.81]	-409.77253733
C_{28}	[-166.38 , 23.47]	-409.78810372
C_{29}	[-166.38 , 21.45]	-409.79372126
C_{30}	[-166.32 , 17.78]	-409.75257572
C_{31}	[-166.24 , 17.68]	-409.74716970
C_{32}	[-166.20 , 11.10]	-409.56332785

Table 3.4: A feasible transition pathway found by the randomized algorithm between two conformations of protein with PDB ID 1CV2.

search for the pathway with the minimum energy. In other words, we want to solve the following optimization problem:

$$\min_{p \in P} \max_{C \in p} E(C),$$

where P is the set of all pathways from the initial to target conformations in the conformational space and $E(C)$ denotes the potential energy of conformation C. In contrast to the previous deterministic algorithm, this approach considers various paths going from the initial to the target conformation. The algorithm consists of three steps.

1. **Constructing a discretized conformational space**

   The first step of the algorithm is to create several intermediate conformations between the initial and target conformations. We can use our special problem setting (all conformations are the same, except for the dihedral angles of a single side-chain) to discretize the conformational space in an efficient way. Let α be a parameter that shows the number of different options (values) that we consider for each chi angle of the special side-chain. In other words we have α possibilities for χ_1 (between χ_1^0 and χ_1^n), α possibilities for χ_2 (between χ_2^0 and χ_2^n) and so on. We divide the interval $[\chi_j^0, \chi_j^n]$ into $\alpha - 1$ equal subintervals. Therefore, the sets of possible values for the j-th chi angle of the special side-chain are as follows:

   $$\{\chi_j^0, \chi_j^0 + \Delta_j, \chi_j^0 + 2\Delta_j, \ldots, \chi_j^0 + (\alpha - 1)\Delta_j\},$$

   where Δ_j is the incremental amount for the j-th chi angle and defined as

   $$\Delta_j = \frac{\chi_j^n - \chi_j^0}{\alpha - 1}, \qquad j = 1, 2, 3, 4.$$

   The conformational space consists of all combinations of these values for the chi angles of the special side-chain, i.e., α^4 intermediate conformations. The parameter α shows the trade-off between the running time and accuracy of our algorithm. A larger value of α leads to more intermediate conformations (a conformational space with better resolution) and therefore a more accurate result. So now we have α^4 intermediate conformations whose only difference is in the dihedral angles of the special side-chain and we should find the best path from the initial to the target conformation through these intermediate conformations.

2. **Constructing a graph**

   In this step we construct a graph G whose nodes correspond to the conformations of the conformational space defined above. G has a source node, denoted by s, corresponding to the initial conformation, and a destination node, denoted by t, corresponding to the target conformation. We connect two conformations C_i and C_k if and only if the difference between the j-th chi angles of the special side-chain (for all $j = 1, 2, 3, 4$) in C_i and C_k is at most Δ_j. For instance, if the chi angles of the special side-chain in C_i and C_k are $\chi_1^i, \chi_2^i, \chi_3^i, \chi_4^i$ and $\chi_1^k, \chi_2^k, \chi_3^k, \chi_4^k$ respectively, then we connect C_i and C_k if and only if

   $$|\chi_j^i - \chi_j^k| \leq \Delta_j \text{ for } 1 \leq j \leq 4.$$

   observe that the j-th chi angle can either decrease by Δ_j, increase by Δ_j, or does not change. Therefore, we have three options for each chi angle. Since the special side-chain has at most four chi angles, each node can have at most $3^4 - 1 = 80$ neighbours (note that we do not count the case in which no chi angle changes). Thus each node has degree at most 80 in G. Furthermore, a weight is assigned to each node that corresponds to the potential energy of its corresponding conformation. The weight of node u is denoted by $weight(u)$. Then we can use a greedy algorithm to find the best path in the graph G, i.e., a path π from the source node to the destination node so that the maximum weight among the nodes of π is the smallest possible.

3. **Finding the best pathway**

   We have a node-weighted graph G and want to find the best path from s to t. Define the weight of a path as the maximum weight of its nodes. Our objective is to find the path with minimum weight. Let Π be the set of all path from s to t in G. We want to solve the following optimization problem:

   $$\min_{\pi \in \Pi} \max_{u \in \pi} weight(u).$$

```
Let $G$ be a node-weighted graph
Let $s$ and $t$ be the source and destination nodes, respectively
$A = \{s\}$
$S = \emptyset$
for $v \in V(G)$
    $prev[v] \leftarrow nil$
while $t \notin S$
    $u \leftarrow$ a node in $A$ with the smallest weight
    for each neighbor ($v \notin A \cup S$) of $u$
        add $v$ to $A$
        $prev[v] \leftarrow u$
    remove $u$ from $A$
    add $u$ to $S$
$\pi = \emptyset$
$v \leftarrow t$
while $prev[v] \neq nil$
    insert $v$ at the beginning of $\pi$
    $v \leftarrow prev[v]$
return $\pi$
```

Figure 3.11: A greedy algorithm for finding the best path in a graph.

We describe a greedy algorithm to efficiently find the best path in G. We maintain a set A of active nodes and a set S of selected nodes. At each iteration, S contains the nodes for which we have found the best path from s, while A maintains the nodes that are not selected yet, but we have found a path from s to them. For each node v we also maintain $prev[v]$ which shows the last node in the best path from the source to v and is initialized to nil. Initially A contains only the source node and S is empty. At each step, we select a node u in A with minimum weight, add u to S, and remove it from A. Furthermore, let v be a neighbour of u which is not in $A \cup S$. We add v to A and set $prev[v]$ to u. We continue this process until we select the destination node. We then use the $prev$ values to find the best path from the source to the destination. The pseudocode for this greedy algorithm is shown in Figure 3.11. We verify the

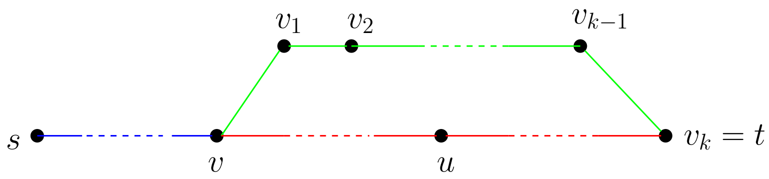


Figure 3.12: Figure for the proof of Theorem 1.

correctness of this algorithm.

Theorem 1. *The greedy algorithm of Figure 3.11 returns a path π of minimum weight from s to t in G.*

Proof. Assume for the sake of contradiction that this is not true and there exists a path π' from s to t in G such that the weight of π' is strictly less than the weight of π. Let u be a node with maximum weight in π. We observe that the weight of u is strictly more than the weights of all nodes in π' (including s and t). We get a contradiction by proving that the greedy algorithm never selects u and thus u cannot be part of π. Let t_u be the iteration in which u is added to S by the greedy algorithm. Suppose that v be the last node in the path from s to u in π that belongs to π' and let $v, v_1, v_2, \ldots, v_k = t$ be the nodes after v in π' (see Figure 3.12). We know that v is added to A before time t_u. Since the weight of v is strictly less than u, v is selected before t_u as well. Therefore v_1, a neighbour of v, is added to S before t_u. In general since the weight of v_i is less than the weight of u, if v_i is active before t_u, then it is selected before t_u, and thus v_{i+1} becomes active before t_u as well. Therefore, all vertices of π' are selected before u. In particular $t = v_k$ is selected before u and the greedy algorithm stops and returns a path before selecting u. This contradiction proves that our original assumption is incorrect and thus π is a path with minimum weight. □

Table 3.5 shows a feasible pathway found by this algorithm (with parameter α set to 20) between the two conformations of protein with PDB ID 1CV2 shown in Figures 3.5 and 3.8.

The feasible pathway contains 38 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-169.41 , 74.38]	-410.81966760
C_1	[-169.25 , 74.38]	-410.85308702
C_2	[-169.09 , 74.38]	-410.88487824
C_3	[-168.93 , 74.38]	-410.91464549
C_4	[-168.77 , 74.38]	-410.94303203
C_5	[-168.61 , 74.38]	-410.96989285
C_6	[-168.45 , 74.38]	-410.99477823
C_7	[-168.29 , 74.38]	-411.01819007
C_8	[-168.13 , 74.38]	-411.04183429
C_9	[-167.97 , 74.38]	-411.06818252
C_{10}	[-167.81 , 74.38]	-411.09332845
C_{11}	[-167.65 , 74.38]	-411.11694227
C_{12}	[-167.49 , 74.38]	-411.13889403
C_{13}	[-167.32 , 74.38]	-411.15954955
C_{14}	[-167.16 , 74.38]	-411.17881593
C_{15}	[-167.00 , 74.38]	-411.19635226
C_{16}	[-166.84 , 74.38]	-411.21266454
C_{17}	[-166.68 , 74.38]	-411.22645843
C_{18}	[-166.52 , 71.22]	-411.11989460
C_{19}	[-166.52 , 68.06]	-410.87583550
C_{20}	[-166.52 , 64.89]	-410.16150717
C_{21}	[-166.52 , 61.73]	-410.04303209
C_{22}	[-166.52 , 58.56]	-409.98754217
C_{23}	[-166.52 , 55.40]	-409.94503613
C_{24}	[-166.52 , 52.24]	-409.92182898
C_{25}	[-166.52 , 49.07]	-409.91656362
C_{26}	[-166.52 , 45.91]	-409.92158947
C_{27}	[-166.52 , 42.74]	-409.92388160
C_{28}	[-166.52 , 39.58]	-409.90862512
C_{29}	[-166.52 , 36.41]	-409.86061404
C_{30}	[-166.52 , 33.25]	-409.76652532
C_{31}	[-166.52 , 30.09]	-409.61449941
C_{32}	[-166.52 , 26.92]	-409.72755227
C_{33}	[-166.52 , 23.76]	-409.78938153
C_{34}	[-166.52 , 20.59]	-409.79517497
C_{35}	[-166.52 , 17.43]	-409.75397726
C_{36}	[-166.36 , 14.26]	-409.67025959
C_{37}	[-166.20 , 11.10]	-409.56332785

Table 3.5: A feasible transition pathway found by the greedy algorithm between two conformations of protein with PDB ID 1CV2.

3.3.4 Runtime Complexity

In this subsection we analyze the running time of three pathway-finding algorithms.

Averaging Algorithm

Consider the averaging algorithm with parameter n. We can compute each intermediate conformation in constant time. Therefore the running time of the averaging algorithm is $O(n)$.

Randomized Algorithm

Let the chi angles of the special side-chain in the initial and target conformation be $(\chi_1^0, \chi_2^0, \chi_3^0, \chi_4^0)$ and $(\chi_1^n, \chi_2^n, \chi_3^n, \chi_4^n)$, respectively, and let d be the *diff* parameter (related to the stopping criterion) in the randomized algorithm. Define δ_j as the difference between the j-th chi angles of the special side-chain in the initial and target conformations, i.e., $\delta_j = |\chi_j^n - \chi_j^0|$.

In order to analyze the expected running time of the randomized algorithm we first consider a relevant algorithm described as follows. Initially we have an interval I of length L, say interval $[0, L)$. At each step we select a point p in the interval uniformly at random, do some constant amount of work, split the interval into two subintervals $I_1 = [0, p)$ and $I_2 = [p, L)$, and then recursively call the algorithm on each subinterval if the length of the subinterval is larger than some parameter d. We refer to this algorithm as the interval-splitting algorithm with parameters (L, d). We analyze the expected running time of the interval-splitting algorithm by considering its recursion tree T. Figure 3.13 shows a simple example of a recursion tree with $L = 10$ and $d = 2$. Observe that the number of subproblems in the i-th level of T is at most 2^i. Since the running time of each subproblem (other than the recursive calls) is constant, the total running time at level i is at most $O(2^i)$. Next we compute the expected number of levels (height of T). We say that we have a *good split* if we have $L/4 \leq p \leq 3L/4$. Otherwise, we say that we have a *bad split*. For example, in Figure 3.13 the splits on I_2 and I_3 are good, while splits on I and I_4 are bad.

Observe that if we have a good split, then the sizes of both subproblems are at most $3L/4$. Therefore the size of each subproblem is reduced by a factor of 3/4 after each good split and after i good splits, the size of subproblem becomes at most $L(3/4)^i$. Recall that we stop when the size of subproblem becomes $\leq d$. Thus we stop after k good splits when

$$L(3/4)^k \leq d \Rightarrow (3/4)^k \leq d/L \Rightarrow (4/3)^k \geq L/d \Rightarrow k \geq \log_{4/3} L/d.$$

Thus we stop after $\lceil \log_{4/3} L/d \rceil$ good splits. Therefore the expected number of levels is at most the expected number of steps in which we have $\lceil \log_{4/3} L/d \rceil$ good splits. Since we select the splitting point uniformly at random, the probability that each split is good is $\frac{3x/4 - x/4}{x} = 1/2$, where x is the length of the interval. Therefore at each step, the probability that the split is good is the same as the probability that split is bad and each equal 1/2. From probability theory that the expected number of steps until we get a good split is $\frac{1}{1/2} = 2$ and the expected number of steps in which we get $\lceil \log_{4/3} L/d \rceil$ good splits is $2\lceil \log_{4/3} L/d \rceil$. Hence the expected number of levels of T is $2\lceil \log_{4/3} L/d \rceil$ and the expected running time of the algorithm (sum over all levels) is

$$\sum_{i=0}^{2\lceil \log_{4/3} L/d \rceil} 2^i \in O(2^{2 \log_{4/3} L/d}) \in O((L/d)^{2 \log_{4/3} 2}) \in O((L/d)^{4.82}).$$

Observe that we can consider the randomized pathway-finding algorithm as four independent executions of the interval-splitting algorithm with parameters (δ_1, d), (δ_2, d), (δ_3, d), and (δ_4, d). Therefore the expected running time of the randomized algorithm is $O((\delta_1/d)^{4.82} + (\delta_2/d)^{4.82} + (\delta_3/d)^{4.82} + (\delta_4/d)^{4.82})$.

Greedy Algorithm

Consider the greedy algorithm with parameter α. The conformational space has $O(\alpha^4)$ conformations. Therefore the graph G has $O(\alpha^4)$ vertices. Recall that the degree of each vertex of G is at most 80. Thus the number of edges in G is $O(80\alpha^4/2) \in O(\alpha^4)$. Constructing each edge or vertex of G and computing the weight of each vertex takes

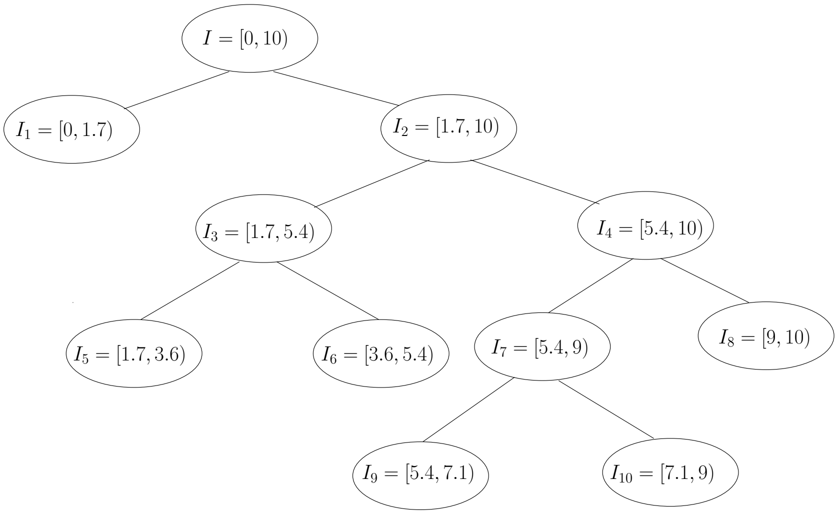


Figure 3.13: A recursion tree for the interval-splitting algorithm with $L = 10$ and $d = 2$.

constant time. Therefore the graph G can be constructed in $O(\alpha^4)$ time. Next we need to apply the greedy algorithm of Figure 3.11 to G. Initialization (lines 1-4) takes $O(|V|) \in O(\alpha^4)$ time. We maintain the vertices in A in a min-priority queue where the priority of each vertex is its weight. At each iteration of the first while loop we use a delete-min operation to select the vertex u in A with the minimum weight. The vertex u is removed from A and added to S. We also add each neighbour of u which is not in $A \cup S$ to A by using an insert operation. So we can have up to $deg(u) \leq 80$ insert operations at each iteration of the first while loop. Observe that u is not added to A again as we do not add vertices in S to A. Thus the first while loop is iterated at most $O(|V|) \in O(\alpha^4)$ times. At each iteration we have a constant number of delete-min and insert operations. If we implement the priority queue with standard heap the running time of delete-min and insert operations is equal to $O(\log |V|) \in O(\log \alpha)$. Therefore the total running time of the first while loop is $O(\alpha^4 \log \alpha)$. The second while loop is executed $O(\alpha^4)$ times and takes constant time per iteration. Thus the total running time of the algorithm of Figure 3.11 is $O(\alpha^4 + \alpha^4 \log \alpha + \alpha^4) \in O(\alpha^4 \log \alpha)$. Overall, the running time of the greedy algorithm is $O(\alpha^4 \log \alpha)$. Observe that there is a trade-off between the accuracy and running time of the algorithm.

Chapter 4

Results and Discussion

In this Chapter we present the results obtained by applying the algorithms described in Chapter 3 to various protein structures. In Section 4.1 we describe the data sources and the visualization software used in our experiments. Furthermore, we briefly explain the software that we used for computing the potential energy of protein structures. Then we provide our experimental results in Section 4.2. More specifically, in Subsections 4.2.1 and 4.2.2 we report the results of applying the tunnel-finding and the tunnel-widening algorithms to several protein structures. Finally, we provide experimental results on the application of pathway-finding algorithms in Subsection 4.2.3.

4.1 Experimental Setup

We first briefly describe the softwares that we used in our experiments, as well as our data sources.

4.1.1 Test Data

We have tested our tunnel finding/widening algorithms on various protein structures taken from the Protein Data Bank (PDB) [5]. As mentioned in Section 2.1.3, the Protein Data

Bank contains three-dimensional structural data of many biological macromolecules. Every structure has a unique identification code, called the PDB ID. The Protein Data Bank provides the structural information of each protein structure in a PDB file. The PDB file is a text file containing the coordinates of the protein atoms.

4.1.2 Visualization Software

After extracting the coordinates of the protein atoms from the PDB file, we can visualize the protein structure using a visualization software. We used the UCSF Chimera software [1] [57] to visualize the protein structures as well as the discovered tunnels. Chimera is an interactive molecular visualization program developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco. [2] Chimera can be downloaded free of charge for academic, non-profit, and personal use. A Python-standard IDLE interactive environment is provided in Chimera which can process Python scripts. Chimera can retrieve files containing atom coordinates from various databases such as PDB, NDP, SCOP, etc. and provides various ways to display a protein structure. Atoms and bonds can be represented by wire-frame, stick, ball and stick, or spheres. The ribbons (flat, edged or rounded) option is available to show the overall structure of the protein. The molecular surface of the protein can be displayed as solid, mesh, or dot. In this thesis, we used the ball and stick option to represent the three-dimensional structure of the proteins.

4.1.3 Computing the Potential Energy

We have used the PyRosetta energy (score) function to compute the potential energy of the protein structures. PyRosetta [3] [11] is a Python-based implementation of the Rosetta molecular modeling package [4] [60] developed for predicting and designing protein struc-

[1] http://www.cgl.ucsf.edu/chimera/

[2] http://www.rbvi.ucsf.edu/

[3] http://www.pyrosetta.org/home

[4] http://www.rosettacommons.org/home

tures, protein folding mechanisms, and protein-protein interactions. The PyRosetta score function is based on the Rosetta energy function [60]. It takes a pose object, i.e., an object which contains all the structural information necessary to define a protein structure, and outputs a score that represents its energy. The Rosetta energy function consists of various components (terms), shown in Table 4.1. Each component c_i is assigned a score weight w_i. The user can assign the desired weights to the energy components to define a custom scoring function. We have applied the default score weights defined in Rosetta (corresponding to the "standard" score function) to compute the energy of the protein structures. The corresponding weights are shown in Table 4.2. The Rosetta energy function, denoted by $\mathcal{F}_E$, is defined as the weighted sum of independent energy components: [5]

$$\mathcal{F}_E = c_1 \times w_1 + c_2 \times w_2 + c_3 \times w_3 + \cdots + c_k \times w_k.$$

4.2 Experimental Results

In this section we describe the results of applying the tunnel-finding, tunnel-widening, and pathway-finding algorithms to various proteins taken from the PDB. Recall that the input to our tunnel-finding and tunnel-widening algorithms consists of a protein conformation together with the coordinates of a starting point inside it. We emphasize that our algorithms do not aim to find the starting points. They assume that the starting points are provided by the user and can be anywhere inside the protein structures. Note that there might not exist a tunnel from some starting points inside the given protein structure to the outside environment. If the tunnel-finding algorithm is provided with such an instance, it reports that a tunnel does not exist. In our experiments we consider various proteins with widely different number of atoms. To illustrate the performance of our algorithms we picked arbitrary points inside these protein structures as the starting points. Since one of the applications of our algorithms is in drug design, we also provided two examples

[5] Note that some components in Table 4.1 are divided into several subcomponents in Table 4.2 and assigned different weights.

Components of Rosetta Energy Function

Name	Description	Functional form	Parameters	Ref.
rama	Ramachandran torsion preferences	$\sum_i -\ln[P(\phi_i, \psi_i \mid aa_i, ss_i)]$	i = residue index ϕ, ψ = backbone torsion angles (36 bins) aa= amino acid type ss= secondary structure type	[8, 61]
LJ	Lennard-Jones interactions	$\sum_i \sum_{j>i} \begin{cases} \left((\frac{r_{ij}}{d_{ij}})^{12} - 2(\frac{r_{ij}}{d_{ij}})^6 \right) e_{ij} & \text{if } \frac{d_{ij}}{r_{ij}} > 0.6 \\ \left(-8759.2(\frac{d_{ij}}{r_{ij}}) + 5672.0 \right) e_{ij} & \text{otherwise} \end{cases}$	i, j =residue indices d = interatomic distance e =geometric mean of atom well depths r= summed van der Waals radii	[40]
hb	Hydrogen bonding	$\sum_i \sum_j \Big[-ln[P(d_{ij} \mid h_j ss_{ij})] - ln[P(\cos\theta_{ij} \mid d_{ij} h_j ss_{ij})]$ $-ln[P(\cos\psi_{ij} \mid d_{ij} h_j ss_{ij})] \Big]$	i =donor residue index j = acceptor residue index d =acceptor-proton interatomic distance h= hybridization (sp^2, sp^3) ss= secondary structure type θ= proton-acceptor-acceptor base bond angle ψ= donor-proton-acceptor bond angle	[39, 73]
solv	Solvation	$\sum_i \left[\Delta G_i^{ref} - \sum_j \left(\frac{2\Delta G_i^{free}}{4\pi^{3/2}\lambda_i r_{ij}^2} e^{-d_{ij}^2} V_j + (\frac{2\Delta G_i^{free}}{4\pi^{3/2}\lambda_j r_{ij}^2} e^{-d_{ij}^2} V_i \right) \right]$	i, j =atom indices d = distance between atoms r =summed van der Waal radii λ= correlation length V=atomic volume $\Delta G^{ref}, \Delta G^{free}$= energy of a fully solvated atom	[40, 41]

Components of Rosetta Energy Function (continued)

Name	Description	Functional form	Parameters	Ref.
pair	Residue pair interactions (electrostatics,disulfides)	$\sum_i \sum_{j>i} -ln\left[\frac{P(aa_i,aa_j\|d_{ij})}{P(aa_i\|d_{ij})P(aa_j\|d_{ij})}\right]$	i,j =residue indices aa =amino acid type d= distance between residues	[40]
dun	Rotamer self-energy	$\sum_i -\ln\left[\frac{P(rot_i\|\phi_i,\psi_i)P(aa_i\|\phi_i,\psi_i)}{P(aa_i)}\right]$	i,j = residue indices ϕ,ψ = backbone torsion angles (36 bins) aa= amino acid type rot=Dunbrack backbone-dependent rotamer	[40, 24]
ref	Unfolded state reference energy	$\sum_{aa} n_{aa}$	aa =amino acid type n = number of residues	[40]

Table 4.1: Components of Rosetta Energy Function [60].

Score	Description	Weight
p_aa_pp	Probability of amino acid at phipsi	0.640
fa_atr	lennard-jones attractive	0.800
fa_rep	lennard-jones repulsive	0.440
fa_intra_rep	lennard-jones repulsive between atoms in the same residue	0.004
hbond_lr_bb	backbone-backbone hbonds distant in primary sequence	1.170
hbond_sr_bb	backbone-backbone hbonds close in primary sequence	1.170
hbond_bb_sc	sidechain-backbone hydrogen bond energy	1.170
hbond_sc	sidechain-sidechain hydrogen bond energy	1.100
fa_sol	lazaridis-jarplus solvation energy	0.650
fa_pair	statistical residue-residue pair potential	0.490
dslf_ss_dst	distance score in current disulfide	1.000
dslf_cs_ang	csangles score in current disulfide	1.000
dslf_ss_dih	dihedral score in current disulfide	1.000
dslf_ca_dih	ca dihedral score in current disulfide	1.000
fa_dun	internal energy of sidechain rotamers as derived from Dunbrack's statistics	0.560
ref	reference energy for each amino acid	1.000

Table 4.2: Default score weights defined in Rosetta.

(proteins 1MJ5 and 1CQW) in which the starting point is located nearby the active site region. More specifically, let P be the set of points in $\mathbb{R}^3$ that correspond to the centers of the atoms of amino acid residues constituting an active site $\mathcal{A}$. We define the centroid of $\mathcal{A}$, denoted by $\mathcal{C}(\mathcal{A})$, as the centroid of points in P and use $\mathcal{C}(\mathcal{A})$ as the starting point.

4.2.1 Finding the Widest Tunnel

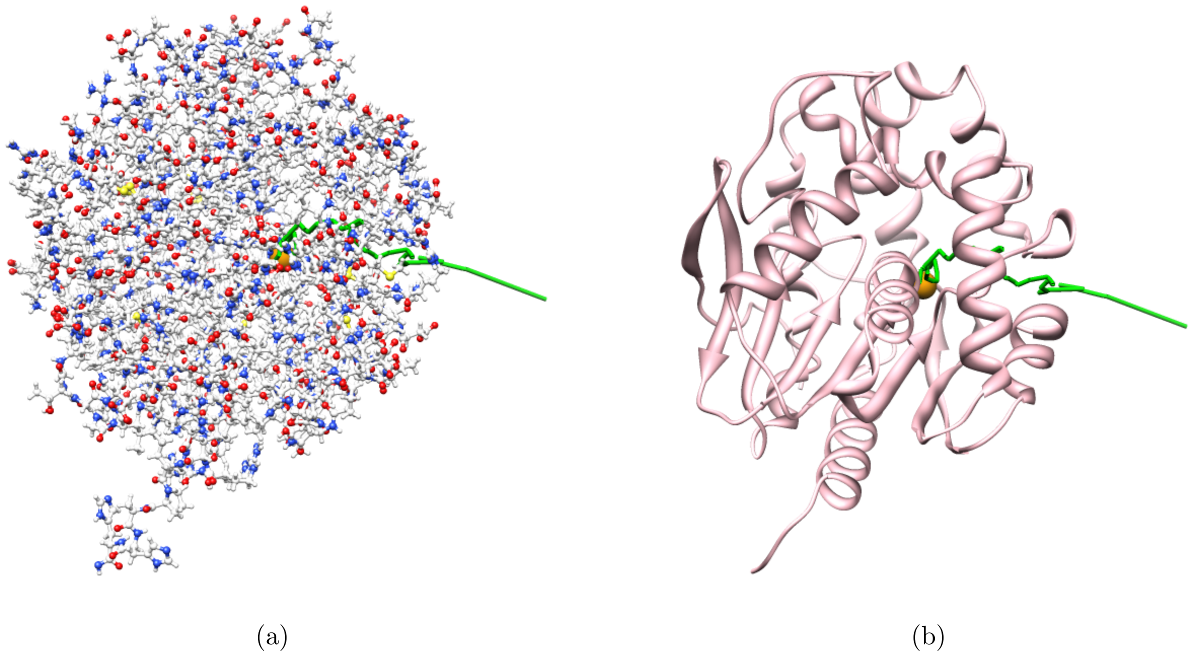
Recall from Section 3.1 that given a fixed conformation of a protein and a position (starting point) inside it, the tunnel-finding algorithm can find the widest tunnel from the starting point to the outside environment of the protein. As stated earlier, the protein conformations are taken from the PDB and visualized in Chimera. We have tested our tunnel-finding algorithm on various protein structures and different starting points. The coordinates of starting points are in the frame of reference used by the PDB coordinates. In all cases, the program discovered and facilitated the visualization of the widest tunnel in the given static conformation in a few seconds. In this subsection we provide the results for several instances.

- **Protein 1MJ5**

  Protein with PDB entry 1MJ5 has one chain containing 302 amino acid residues. This protein has an active site which is located between its two domains and includes the catalytic residues Asp 108, Glu 132, and His 272 [51]. Recall from Chapter 1 that in drug design we are interested in finding wide tunnels from the active site to the outside environment. Therefore, we selected the starting point to be a point with coordinates (16.93,31.44,4.45) which is the centroid of the active site. Then, we applied the tunnel-finding algorithm on this protein with the aforementioned starting point. Figure 4.1 shows the widest tunnel found for this instance. The width of this tunnel is 0.23 Å.

- **Protein 1CQW**

  Protein 1CQW has one chain containing 295 amino acid residues. The active site of this protein involves residues Asp 117.A, TRP 118.A, GLU 141.A, and HIS 283.A.



(a)

(b)

Figure 4.1: The widest tunnel in protein 1MJ5 with the starting point at position (16.93,31.44,4.45). (a) Protein atoms represented using ball and stick option in Chimera. (b) Overall structure of the protein represented using the ribbon option in Chimera.

[50]. To find the widest tunnel from the active site to the outside environment of the protein, we set the starting point to the centroid of the active site, i.e., the point with coordinates (21.92,98.09,39.59). Then, we applied the tunnel-finding algorithm on this protein with the starting point at position (21.92,98.09,39.59). Figure 4.2 shows the widest tunnel discovered by the tunnel-finding algorithm for this protein and starting point. The width of the widest tunnel is 0.54 Å.

- **Protein 1CV2**

  The PDB entry 1CV2 corresponds to the crystal structure of haloalkane dehalogenase LinB enzyme [48]. The length of protein 1CV2 (the number of amino acid residues) is 296. In Section 3.1, we presented the result of applying the tunnel-finding algorithm to this protein with the starting point at position (14,15,22) (see Figure 3.5). Here,

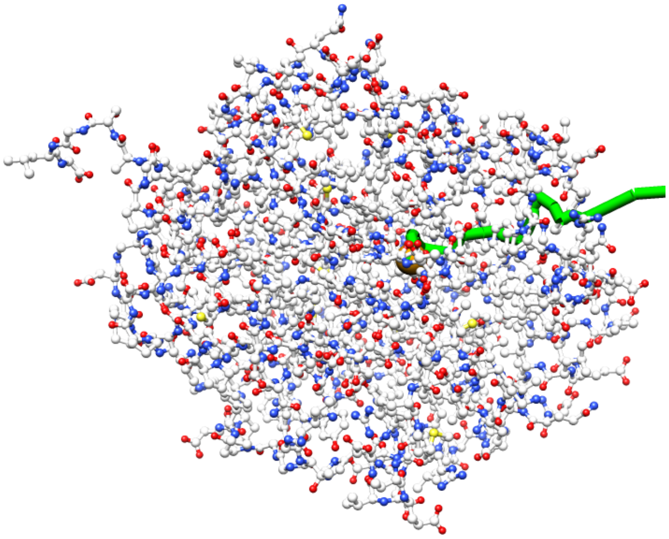


Figure 4.2: The widest tunnel in the protein 1CQW with the starting point at position (21.92,98.09,39.59).

we consider the same protein conformation, but a different starting point. Figure 4.3 shows the widest tunnel discovered in this protein with the starting point at position (24,12,18). The width of the widest tunnel in this conformation is 0.46 Å.

- **Protein 1CV4**

  The protein with PDB ID 1CV4 is a one-chain structure and consists of 164 amino acid residues. Therefore, it is much smaller than the previous proteins. Figure 4.4 shows the widest tunnel found in this protein with the starting point at position (36,7,8). The width of the corresponding tunnel is 0.57 Å.

- **Protein 2YJK**

  Protein 2YJK has 12 chains, where each chain contains 161 amino acid residues. Thus, it is much larger than the previous four proteins. We applied the tunnel-finding algorithm on this protein with the starting point at position (20,5,55). The corresponding widest tunnel is shown in Figure 4.5. The width of the tunnel is 0.86 Å. Despite the large size of the protein, the tunnel-finding algorithm was able to

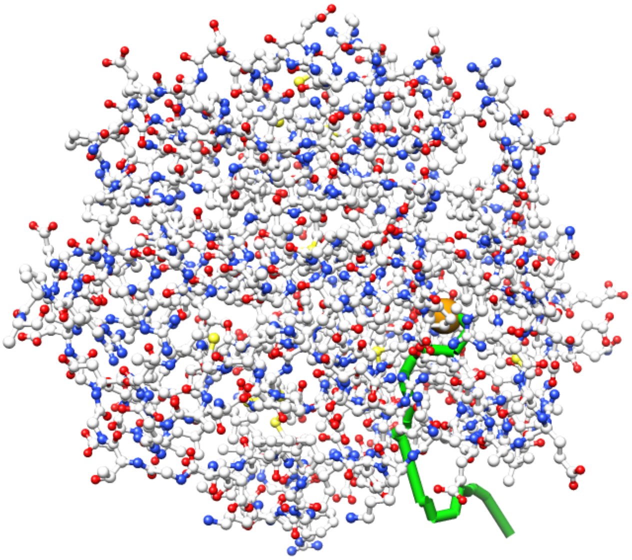


Figure 4.3: The widest tunnel in protein 1CV2 with starting point at position (24,12,18).

find and visualize the widest tunnel in a few seconds. Observe that the tunnel found by the algorithm is long and shorter tunnels might exist. However, recall that the tunnel-finding algorithm finds the widest tunnel, regardless of the length.

- **Protein 1CSW**

  Protein 1CSW has 108 amino acid residues. We applied the tunnel-finding algorithm to this protein with the starting point at position (-2,17,4). The corresponding widest tunnel is shown in Figure 4.6. The width of this tunnel is 0.46 Å.

We also tested the tunnel-finding algorithm on several other protein conformations with different starting points. Table 4.3 provides the results for some of these input instances.

4.2.2 Widening the Tunnel

In Section 3.2 we proposed a tunnel-widening algorithm that aims to find a wider tunnel in an alternative conformation of the initial structure whose energy is not much higher than

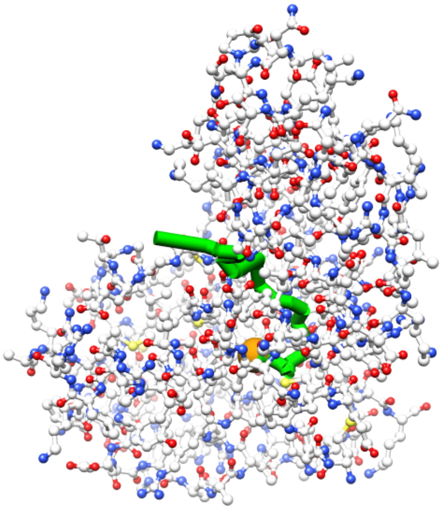


Figure 4.4: The widest tunnel in protein 1CV4 with starting point at position (36,7,8).

the energy of the initial conformation. In that section we reported the result of applying this tunnel-widening algorithm to the protein 1CV2 with the starting point at position (14,15,22). The tunnel-widening algorithm increased the width of the tunnel from 0.43 Å to 0.59 Å. In this subsection we provide more experimental results for the tunnel-widening algorithm. More specifically, we consider the instances used by the tunnel-finding algorithm in Subsection 4.2.1.

- **Protein 1MJ5**

  In Subsection 4.2.1 we reported that the width of the widest tunnel in the initial conformation of protein 1MJ5 with the starting point at position (16.93,31.44,4.45) is 0.23 Å. One of the bottleneck side-chains of this tunnel belongs to the residue HIS 272.A. The sidechain dihedral angles of this residue in the original conformation are $\chi_1 = -174.44°$ and $\chi_2 = 61.70°$. By replacing the sidechain of this residue by the rotamer with dihedral angles $\chi_1 = -177.10°$, $\chi_2 = 72.30°$, we identified a tunnel with width 0.38 Å. The potential energy of the structure changed from -493.260 to -492.636 kcal/mol. Thus we found an alternative conformation with a wider tunnel

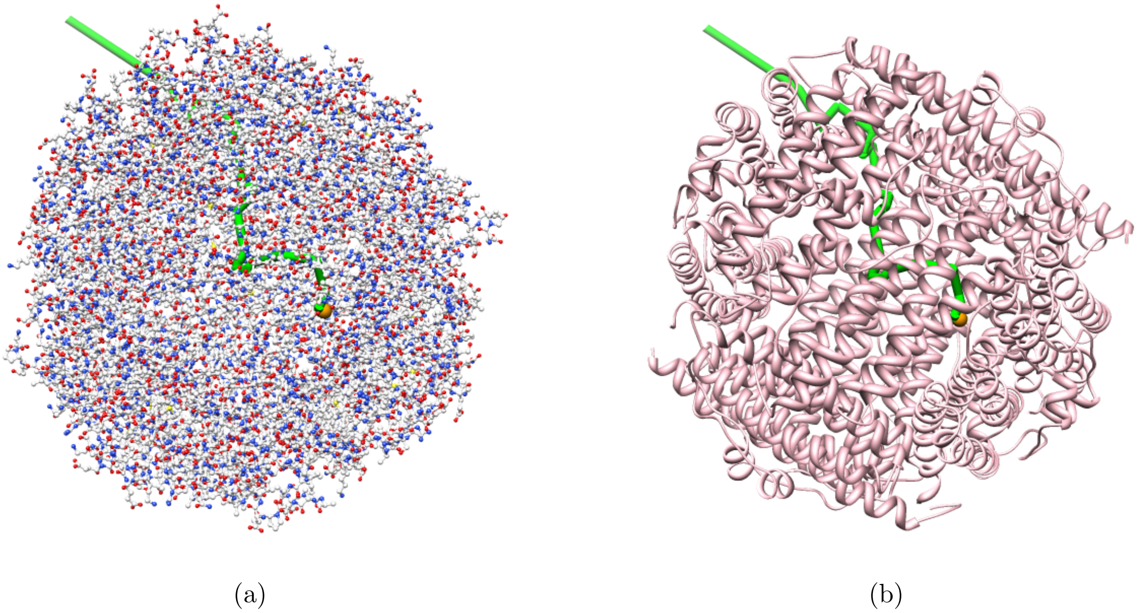


Figure 4.5: The widest tunnel in protein 2YJK with the starting point at position (20,5,55). (a) Protein atoms represented using ball and stick option in Chimera. (b) Overall structure of the protein represented using ribbon option in Chimera.

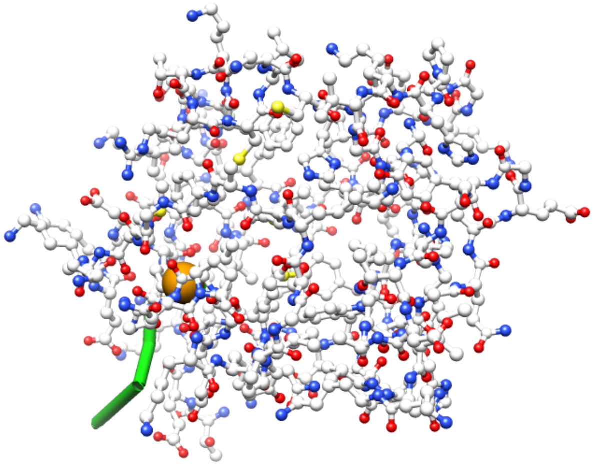


Figure 4.6: The widest tunnel in protein 1CSW with the starting point at position (-2,17,4).

PDB ID	Length (number of residues)	Coordinates of the starting point	Width of the widest tunnel (Å)	Energy (kcal/mol)
1CSW	108	(5,15,9)	0.20	88.119
1CSW	108	(10,21,7)	0.55	88.119
1CV4	164	(36,5,12)	0.78	-27.329
1CV4	164	(35,10,10)	0.77	-27.329
1A30	201	(15,22,2)	0.89	-250.309
1CQW	295	(14,98,43)	0.52	-487.858
1CQW	295	(26,97,36)	0.87	-487.858
1MJ5	302	(8,35,6)	0.13	-493.260
1MJ5	302	(18,32,4)	0.26	-493.260
1MJ5	302	(12,30,4)	0.11	-493.260
2HAD	310	(30,106,27)	0.84	-216.750
1EBV	551	(29,39,190)	0.75	323.946
3N5E	658	(-50,4,30)	0.42	10.396
3S2A	960	(23,-5,27)	0.68	-296.010
1DCE	1796	(58,27,30)	0.77	2502.114

Table 4.3: Width of the widest tunnels in various protein conformations and starting points.

and not much higher potential energy. Figure 4.7 shows the widest tunnel in the corresponding alternative conformation of protein 1MJ5.

- **Protein 1CQW**

  Recall from Subsection 4.2.1 that the width of the widest tunnel in the initial conformation of protein 1CQW with the starting point at position (21.92,98.09,39.59) is 0.54 Å. One of the bottleneck side-chains of this tunnel belongs to the residue HIS 283.A. The sidechain dihedral angles of this residue in the initial conformation are $\chi_1 = -176.69°$ and $\chi_2 = 62.68°$. We discovered a wider tunnel with width 0.63

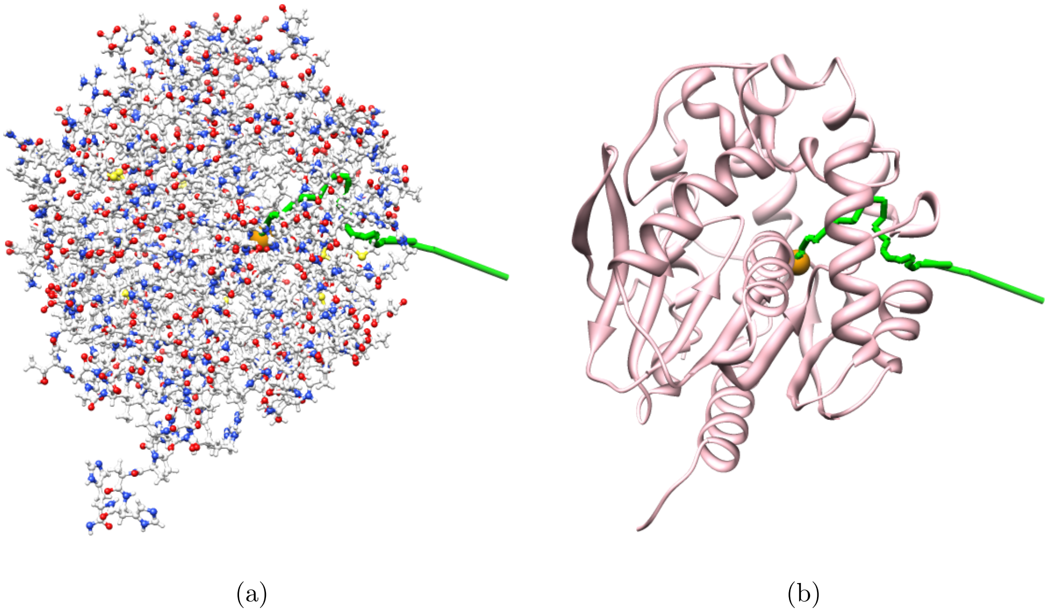


Figure 4.7: The widest tunnel in an alternative conformation of protein 1MJ5 with the starting point at position (16.93,31.44,4.45). (a) Protein atoms represented using ball and stick option in Chimera. (b) Protein atoms represented using ribbon option in Chimera.

Å by replacing the sidechain of this residue by the rotamer with dihedral angles $\chi_1 = -175.80°$ and $\chi_2 = 71.80°$. The potential energy of the structure changed from -487.858 to -488.036 kcal/mol. Therefore, we found an acceptable alternative conformation of protein 1CQW with a wider tunnel. Figure 4.8 shows the widest tunnel in this alternative conformation.

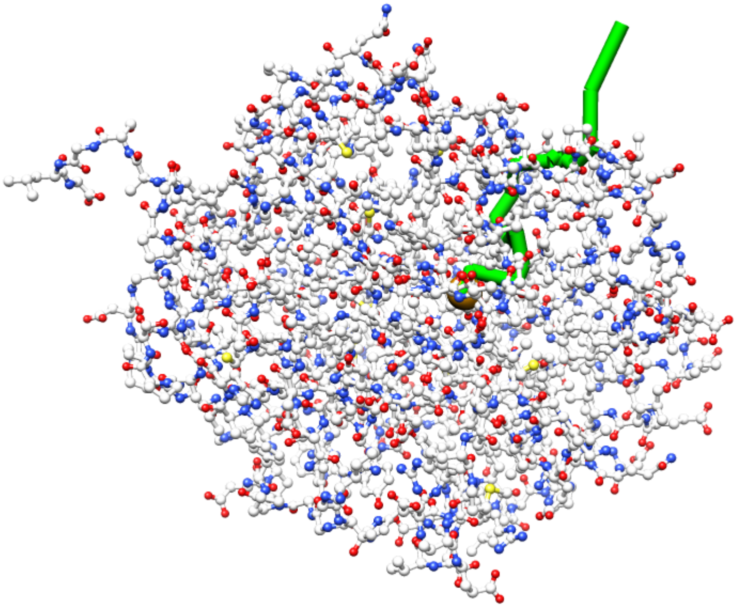


Figure 4.8: The widest tunnel in an alternative conformation of protein 1CQW with the starting point at position (21.92,98.09,39.59)

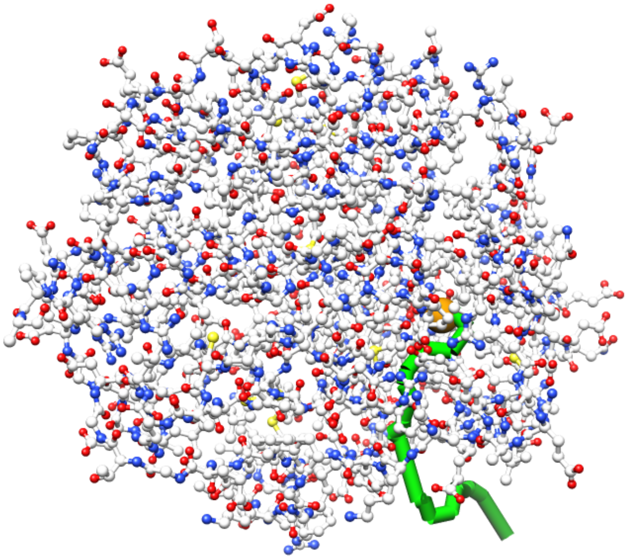


Figure 4.9: The widest tunnel in an alternative conformation of protein 1CV2 with the starting point at position (24,12,18).

- **Protein 1CV2**

  In Subsection 4.2.1 we considered the protein 1CV2 with the starting point at position (24,12,18). The width of the widest tunnel in the initial conformation is 0.46 Å. One of the bottleneck atoms of this tunnel belongs to the residue ASN 38.A. The sidechain dihedral angles of this residue in the original conformation are $\chi_1 = -168.37°$ and $\chi_2 = 42.51°$. By replacing the sidechain of this residue by the rotamer with dihedral angles $\chi_1 = -174.40°$ and $\chi_2 = 68.30°$, we identified a tunnel with width 0.65 Å. The potential energy of the structure changed from -410.819 to -406.236 kcal/mol. Thus we found an alternative conformation with a wider tunnel and not much higher potential energy. Figure 4.9 shows the widest tunnel in an alternative conformation of protein 1CV2.

- **Protein 1CV4**

  In Section 4.2.1 we considered the protein 1CV4 with the starting point at position (36,7,8). The width of the widest tunnel in the initial conformation is 0.57 Å. One of the bottleneck side-chains of this tunnel belongs to the residue ILE 3.A. The sidechain dihedral angles of this residue in the original conformation are $\chi_1 = -173.43°$ and $\chi_2 = 58.54°$. By replacing the sidechain of this residue by the rotamer with dihedral angles $\chi_1 = -170.0°$ and $\chi_2 = 64.10°$, we identified a tunnel with width 0.84 Å. The potential energy of the structure changed from -27.329 to -27.394 kcal/mol. Thus we found an alternative conformation with a wider tunnel and not much higher potential energy. The widest tunnel in this alternative conformation is shown in Figure 4.10.

- **Protein 2YJK**

  Another instance considered in Subsection 4.2.1 is protein 2YJK with the starting point at position (20,5,55). The width of widest tunnel in this instance is 0.86 Å. One of the bottleneck side-chains belongs to the residue TYR 65.G. The side-chain dihedral angles in the initial conformation are $\chi_1 = -59.15°$ and $\chi_2 = -28.56°$. By replacing the side-chain of this residue with the rotamer with dihedral angles $\chi_1 = -69.0°$ and $\chi_2 = -15.0°$ we discovered a wider tunnel with width 0.98 Å. The energy changed from -2044.576 to -1995.577 kcal/mol. Thus, the energy value of this alternative conformation is much higher than the energy of the initial conformation. Therefore, the alternative conformation is not acceptable and the tunnel-widening algorithm fails to find a wider tunnel.

- **Protein 1CSW**

  The last example described in Subsection 4.2.1 was protein 1CSW with the starting point at position (-2,17,4). The width of the widest tunnel in the initial conformation is 0.46 Å. One of the bottleneck side-chains of this tunnel belongs to the residue ARG 91.A. The sidechain dihedral angles of this residue in the original conformation are $\chi_1 = -59.50°$, $\chi_2 = -157.14°$, $\chi_3 = -65.70°$, and $\chi_4 = -78.72°$. By replacing the sidechain of this residue by the rotamer with dihedral angles $\chi_1 = -69.10°$, $\chi_2 = -179.40°$, $\chi_3 = -70.90°$, and $\chi_4 = 169.90°$, we identified a tunnel with width 0.61 Å. The potential energy of the structure changed from 88.119 to 91.691 kcal/mol. Thus we found an alternative conformation with a wider tunnel and not much higher potential energy. Figure 4.11 shows the widest tunnel discovered in the alternative conformation of 1CSW.

We applied the tunnel-widening algorithm to several other protein conformations and various starting points. The results obtained for the instances of Table 4.3 are provided in Table 4.4. As can be seen, the tunnel-widening algorithm can increase the width of the tunnel in all these cases. For instance, the width of tunnel in protein 1DCE with starting point at position (58,27,30) is increased from 0.77 Å to 1.82 Å. Therefore the alternative conformation of 1DCE has a tunnel that is wide enough for Magnesium ion (Mg^{2+}, ionic radius: 0.86 Å), while the widest tunnel in the initial structure is not wide enough for this ligand.

4.2.3 Transition Pathway

Using the tunnel-widening algorithm, we can investigate the possibility of finding a wider tunnel by a slight local change in the structure of the protein. In Subsection 4.2.2, we applied the tunnel-widening algorithm to various instances and in most cases we were able

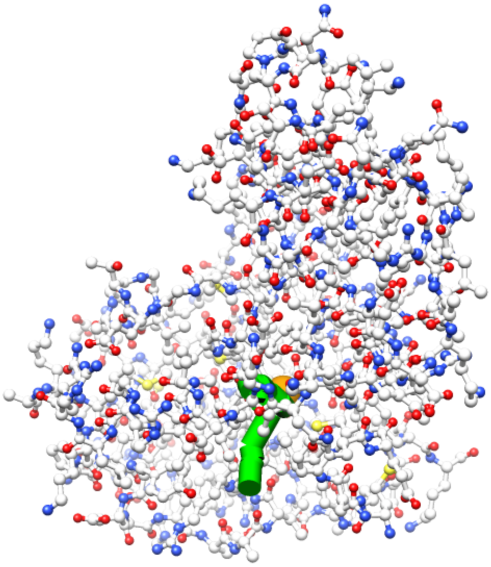


Figure 4.10: The widest tunnel in an alternative conformation of protein 1CV4 with the starting point at position (36,7,8).

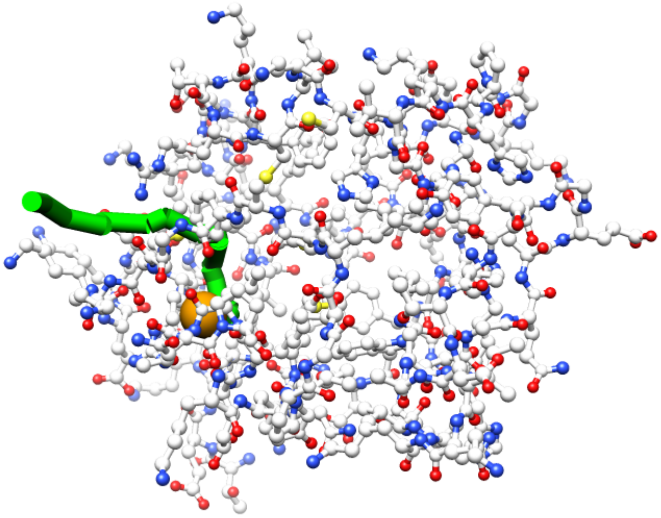


Figure 4.11: The widest tunnel in an alternative conformation of protein 1CSW with the starting point at position (-2,17,4).

PDB ID	Protein length	Starting point	Initial width (Å)	Target width (Å)	Initial energy (kcal/mol)	Final energy (kcal/mol)	Feasible transition pathway
1CSW	108	(5,15,9)	0.20	0.55	88.119	87.995	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1CSW	108	(10,21,7)	0.55	0.72	88.119	87.980	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1CV4	164	(36,5,12)	0.78	0.82	-27.329	-27.394	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1A30	201	(15,22,2)	0.89	1.02	-250.309	-250.562	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1CQW	295	(14,98,43)	0.52	0.77	-487.858	-487.867	YES ($\alpha = 30, \lvert\chi\rvert = 3$)
1CQW	295	(26,97,36)	0.87	1.16	-487.858	-483.982	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1MJ5	302	(8,35,6)	0.13	0.36	-493.260	-491.126	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1MJ5	302	(18,32,4)	0.26	0.50	-493.260	-490.423	YES ($\alpha = 30, \lvert\chi\rvert = 3$)
1MJ5	302	(12,30,4)	0.11	0.29	-493.260	-492.984	YES ($\alpha = 50, \lvert\chi\rvert = 3$)
2HAD	310	(30,106,27)	0.84	0.96	-216.750	-216.566	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1EBV	551	(29,39,190)	0.75	0.84	323.946	323.431	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
3N5E	658	(-50,4,30)	0.42	0.58	10.396	10.001	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1DCE	1796	(58,27,30)	0.77	1.82	2502.114	2501.969	YES ($\alpha = 50, \lvert\chi\rvert = 2$)
1CV4	164	(35,10,10)	0.77	0.82	-27.329	-26.288	NO ($\alpha = 50, \lvert\chi\rvert = 2$)
3S2A	960	(23,-5,27)	0.68	0.78	-296.010	-296.187	NO ($\alpha = 10, \lvert\chi\rvert = 4$)

Table 4.4: Output of the tunnel-widening algorithm on various protein conformations. The fourth and fifth columns show the width of the widest tunnel in the initial and the alternative conformations, respectively. Initial and final energies shown in columns six and seven correspond to the energy of the initial conformation and the energy of the alternative conformation, respectively. The existence or non-existence of a feasible transition pathway between the initial and alternative conformations is reported in the last column.

to find a wider tunnel in an alternative structure of the initial conformation. However, there is no guarantee that this transition from the initial conformation to the alternative conformation is feasible. In Section 3.3 we described methods to ensure that the alternative conformation (also called the target conformation) is accessible from the initial conformation. More specifically, we proposed several algorithms to find a transition pathway between the initial conformation and the target conformation. Here, we consider the protein instances used by the tunnel-widening algorithm in Subsection 4.2.2 and for each instance we check whether a feasible transition pathway between the initial and target conformations can be found.

- **Protein 1MJ5**

  Recall that the tunnel-widening algorithm can increase the width of the widest tunnel from 0.23 Å in the initial structure to 0.38 Å in an alternative conformation. The dihedral angles of the special side-chain are $\chi_1 = 174.44°$ and $\chi_2 = 61.70°$ in the initial conformation and $\chi_1 = 177.10°$ and $\chi_2 = 72.30°$ in the target conformation. The potential energy of the structure changed from -493.260 to -492.636 kcal/mol. We use the pathway-finding algorithms to test whether the target conformation is accessible from the initial conformation. Table 4.5 shows the transition pathway found by the averaging algorithm with parameter $n = 25$ (number of intermediate conformations). According to these results, the energy of all intermediate conformations are close to the energy of the initial conformation and thus the pathway is feasible. The pathways found by the randomized algorithm with parameter *diff*=2 is shown in Table 4.6. It contains 25 conformations and it is feasible as well. Using the greedy algorithm with parameter $\alpha = 20$, we also found a feasible transition pathway containing 33 conformations (see Table 4.7 for the transition pathway). Thus for this instance all algorithms discover feasible pathways from the initial conformation to the target conformation.

The feasible pathway contains 26 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-174.44 , 61.70]	-493.26006920
C_1	[-174.55 , 62.13]	-493.25847803
C_2	[-174.65 , 62.55]	-493.25554065
C_3	[-174.76 , 62.97]	-493.25129358
C_4	[-174.87 , 63.40]	-493.24578433
C_5	[-174.97 , 63.82]	-493.23099982
C_6	[-175.08 , 64.25]	-493.21205177
C_7	[-175.18 , 64.67]	-493.19168453
C_8	[-175.29 , 65.09]	-493.16994061
C_9	[-175.40 , 65.52]	-493.14678598
C_{10}	[-175.50 , 65.94]	-493.12249741
C_{11}	[-175.61 , 66.36]	-493.09687832
C_{12}	[-175.72 , 66.79]	-493.07024355
C_{13}	[-175.82 , 67.21]	-493.04247366
C_{14}	[-175.93 , 67.64]	-493.01344743
C_{15}	[-176.04 , 68.06]	-492.98348870
C_{16}	[-176.14 , 68.48]	-492.95245777
C_{17}	[-176.25 , 68.91]	-492.92033746
C_{18}	[-176.36 , 69.33]	-492.88737557
C_{19}	[-176.46 , 69.76]	-492.85351601
C_{20}	[-176.57 , 70.18]	-492.81891342
C_{21}	[-176.67 , 70.60]	-492.78353365
C_{22}	[-176.78 , 71.03]	-492.74744864
C_{23}	[-176.89 , 71.45]	-492.71074975
C_{24}	[-176.99 , 71.88]	-492.67341794
C_{25}	[-177.10 , 72.30]	-492.63560742

Table 4.5: A feasible transition pathway found by the averaging algorithm with parameter $n = 25$ between two conformations of protein 1MJ5.

The feasible pathway contains 25 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-174.44 , 61.70]	-493.26006920
C_1	[-174.46 , 62.40]	-493.28482552
C_2	[-174.50 , 62.74]	-493.29153582
C_3	[-175.47 , 63.52]	-493.08050736
C_4	[-175.71 , 64.05]	-493.02441105
C_5	[-175.71 , 64.12]	-493.02389749
C_6	[-175.73 , 64.47]	-493.02568257
C_7	[-175.86 , 65.23]	-493.00305915
C_8	[-175.87 , 65.77]	-493.00771843
C_9	[-175.91 , 66.93]	-493.01125516
C_{10}	[-175.95 , 66.99]	-492.99930355
C_{11}	[-175.99 , 67.01]	-492.98939543
C_{12}	[-176.02 , 67.64]	-492.98488331
C_{13}	[-176.08 , 68.46]	-492.97145077
C_{14}	[-176.48 , 68.84]	-492.84423516
C_{15}	[-176.56 , 68.93]	-492.81983565
C_{16}	[-176.76 , 69.06]	-492.75033607
C_{17}	[-176.77 , 70.01]	-492.75253555
C_{18}	[-176.77 , 70.66]	-492.75107853
C_{19}	[-176.78 , 71.02]	-492.74696147
C_{20}	[-176.78 , 71.13]	-492.74662700
C_{21}	[-176.81 , 72.03]	-492.73206946
C_{22}	[-176.96 , 72.28]	-492.68403889
C_{23}	[-177.01 , 72.29]	-492.66613660
C_{24}	[-177.10 , 72.30]	-492.63560742

Table 4.6: A feasible transition pathway found by the randomized algorithm with parameter *diff*=2 between two conformations of protein 1MJ5.

The feasible pathway contains 33 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-174.44 , 61.70]	-493.26006920
C_1	[-174.44 , 62.23]	-493.28208797
C_2	[-174.44 , 62.76]	-493.30335339
C_3	[-174.44 , 63.29]	-493.32384867
C_4	[-174.44 , 63.82]	-493.34342907
C_5	[-174.44 , 64.35]	-493.36218922
C_6	[-174.44 , 64.88]	-493.38006913
C_7	[-174.44 , 65.41]	-493.39633018
C_8	[-174.44 , 65.94]	-493.40647166
C_9	[-174.44 , 66.47]	-493.41286510
C_{10}	[-174.44 , 67.00]	-493.41796759
C_{11}	[-174.44 , 67.53]	-493.42179300
C_{12}	[-174.44 , 68.06]	-493.42430182
C_{13}	[-174.57 , 68.59]	-493.39301807
C_{14}	[-174.71 , 68.59]	-493.35992294
C_{15}	[-174.84 , 68.59]	-493.32621870
C_{16}	[-174.97 , 68.59]	-493.29175406
C_{17}	[-175.11 , 68.59]	-493.25656732
C_{18}	[-175.24 , 68.59]	-493.22057827
C_{19}	[-175.37 , 69.12]	-493.18395945
C_{20}	[-175.50 , 69.12]	-493.14601682
C_{21}	[-175.64 , 69.12]	-493.10744846
C_{22}	[-175.77 , 69.65]	-493.06818096
C_{23}	[-175.90 , 69.65]	-493.02839808
C_{24}	[-176.04 , 69.65]	-492.98797292
C_{25}	[-176.17 , 69.65]	-492.94669591
C_{26}	[-176.30 , 69.65]	-492.90483904
C_{27}	[-176.44 , 70.18]	-492.86213772
C_{28}	[-176.57 , 70.18]	-492.81891342
C_{29}	[-176.70 , 70.71]	-492.77458181
C_{30}	[-176.83 , 71.24]	-492.72915045
C_{31}	[-176.97 , 71.77]	-492.68280261
C_{32}	[-177.10 , 72.30]	-492.63560742

Table 4.7: A feasible transition pathway found by the greedy algorithm with parameter $\alpha = 20$ between two conformations of protein 1MJ5.

- **Protein 1CQW**

  We also applied the tunnel-widening algorithm to protein 1CQW and starting point at position (21.92,98.09,39.59). The width of the widest tunnel increased from 0.54 Å in the initial structure to 0.63 Å in an alternative conformation. The dihedral angles of the special side-chain changed from $\chi_1 = 176.69°$, $\chi_2 = 62.68°$ in the initial structure to $\chi_1 = 175.80°$, $\chi_2 = 71.80°$ in the target conformation. The potential energy of the structure changed from -487.858 to -488.036 kcal/mol. To check whether the target conformation is accessible from the initial conformation, we used the pathway-finding algorithms. Tables 4.8-4.10 show the transition pathways found by the averaging algorithm (with parameter $n = 26$), the randomized algorithm (with parameter *diff*=2), and the greedy algorithm (with parameter $\alpha = 20$), respectively. Observe that all algorithms discover feasible pathways from the initial conformation to the target conformation.

The feasible pathway contains 27 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-176.69 , 62.68]	-487.85805674
C_1	[-176.65 , 63.03]	-487.86890625
C_2	[-176.62 , 63.38]	-487.87969276
C_3	[-176.59 , 63.73]	-487.89037057
C_4	[-176.55 , 64.08]	-487.90050744
C_5	[-176.52 , 64.43]	-487.91047991
C_6	[-176.48 , 64.79]	-487.92025865
C_7	[-176.45 , 65.14]	-487.93001387
C_8	[-176.42 , 65.49]	-487.93935306
C_9	[-176.38 , 65.84]	-487.94815739
C_{10}	[-176.35 , 66.19]	-487.95689415
C_{11}	[-176.31 , 66.54]	-487.96549403
C_{12}	[-176.28 , 66.89]	-487.97335635
C_{13}	[-176.24 , 67.24]	-487.98089835
C_{14}	[-176.21 , 67.59]	-487.98832802
C_{15}	[-176.18 , 67.94]	-487.99496720
C_{16}	[-176.14 , 68.29]	-488.00126880
C_{17}	[-176.11 , 68.64]	-488.00731736
C_{18}	[-176.07 , 68.99]	-488.01274199
C_{19}	[-176.04 , 69.34]	-488.01774347
C_{20}	[-176.01 , 69.70]	-488.02194936
C_{21}	[-175.97 , 70.05]	-488.02586475
C_{22}	[-175.94 , 70.40]	-488.02923849
C_{23}	[-175.90 , 70.75]	-488.03182875
C_{24}	[-175.87 , 71.10]	-488.03384842
C_{25}	[-175.83 , 71.45]	-488.03529487
C_{26}	[-175.80 , 71.80]	-488.03627345

Table 4.8: A feasible transition pathway found by the averaging algorithm with parameter $n = 26$ between two conformations of protein 1CQW.

The feasible pathway contains 25 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-176.69 , 62.68]	-487.85805674
C_1	[-176.64 , 64.11]	-487.89456813
C_2	[-176.60 , 64.64]	-487.90950132
C_3	[-176.60 , 64.70]	-487.91125468
C_4	[-176.59 , 64.71]	-487.91239091
C_5	[-176.58 , 65.22]	-487.92391343
C_6	[-176.58 , 65.60]	-487.93247439
C_7	[-176.53 , 66.71]	-487.95904761
C_8	[-176.42 , 67.05]	-487.97157000
C_9	[-176.42 , 67.10]	-487.97242425
C_{10}	[-176.39 , 67.42]	-487.97984807
C_{11}	[-176.37 , 67.46]	-487.98129883
C_{12}	[-176.37 , 67.49]	-487.98207478
C_{13}	[-176.37 , 68.13]	-487.99420020
C_{14}	[-176.37 , 68.17]	-487.99479106
C_{15}	[-176.35 , 69.21]	-488.01352408
C_{16}	[-176.30 , 69.69]	-488.02226517
C_{17}	[-176.28 , 69.69]	-488.02242986
C_{18}	[-176.27 , 69.69]	-488.02255397
C_{19}	[-176.26 , 69.73]	-488.02322326
C_{20}	[-176.24 , 70.38]	-488.03281364
C_{21}	[-176.24 , 70.93]	-488.03984181
C_{22}	[-176.23 , 70.99]	-488.04063525
C_{23}	[-175.81 , 71.30]	-488.03313578
C_{24}	[-175.80 , 71.80]	-488.03627345

Table 4.9: A feasible transition pathway found by the randomized algorithm with parameter *diff*=2 between two conformations of protein 1CQW.

The feasible pathway contains 27 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-176.69 , 62.68]	-487.85805674
C_1	[-176.64 , 63.14]	-487.87213882
C_2	[-176.60 , 63.59]	-487.88609423
C_3	[-176.56 , 64.05]	-487.89948554
C_4	[-176.51 , 64.50]	-487.91241912
C_5	[-176.47 , 64.96]	-487.92517962
C_6	[-176.42 , 65.42]	-487.93754854
C_7	[-176.38 , 65.87]	-487.94903854
C_8	[-176.33 , 66.33]	-487.96034694
C_9	[-176.29 , 66.78]	-487.97102665
C_{10}	[-176.24 , 67.24]	-487.98089835
C_{11}	[-176.20 , 67.70]	-487.99040421
C_{12}	[-176.16 , 68.15]	-487.99884335
C_{13}	[-176.11 , 68.61]	-488.00677446
C_{14}	[-176.11 , 69.06]	-488.01384485
C_{15}	[-176.11 , 69.52]	-488.02056947
C_{16}	[-176.11 , 69.98]	-488.02707878
C_{17}	[-176.11 , 70.43]	-488.03289247
C_{18}	[-176.11 , 70.89]	-488.03849133
C_{19}	[-176.11 , 71.34]	-488.04329514
C_{20}	[-176.07 , 71.80]	-488.04678208
C_{21}	[-176.02 , 71.80]	-488.04554604
C_{22}	[-175.98 , 71.80]	-488.04415104
C_{23}	[-175.93 , 71.80]	-488.04251486
C_{24}	[-175.89 , 71.80]	-488.04062663
C_{25}	[-175.84 , 71.80]	-488.03853458
C_{26}	[-175.80 , 71.80]	-488.03627345

Table 4.10: A feasible transition pathway found by the greedy algorithm with parameter $\alpha = 20$ between two conformations of protein 1CQW.

- **Protein 1CV2**

  For the protein 1CV2 with the starting point at position (24,12,18), the tunnel-widening algorithm was able to discover a tunnel of width 0.65 Å in an alternative conformation while the widest tunnel in the initial conformation has width 0.46 Å.

The feasible pathway contains 21 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-168.37 , 42.51]	-410.81966760
C_1	[-168.68 , 43.80]	-410.68561901
C_2	[-168.98 , 45.09]	-409.67187670
C_3	[-169.28 , 46.38]	-409.75656046
C_4	[-169.58 , 47.67]	-409.81638012
C_5	[-169.88 , 48.96]	-409.84916252
C_6	[-170.18 , 50.25]	-409.85333819
C_7	[-170.48 , 51.54]	-409.83496357
C_8	[-170.78 , 52.83]	-409.78903649
C_9	[-171.09 , 54.12]	-409.71070030
C_{10}	[-171.39 , 55.41]	-409.59823478
C_{11}	[-171.69 , 56.70]	-409.44965416
C_{12}	[-171.99 , 57.99]	-409.26377632
C_{13}	[-172.29 , 59.27]	-409.03885665
C_{14}	[-172.59 , 60.56]	-408.77312465
C_{15}	[-172.89 , 61.85]	-408.46539127
C_{16}	[-173.19 , 63.14]	-408.11378210
C_{17}	[-173.50 , 64.43]	-407.71682439
C_{18}	[-173.80 , 65.72]	-407.27233613
C_{19}	[-174.10 , 67.01]	-406.77944668
C_{20}	[-174.40 , 68.30]	-406.23611648

Table 4.11: A feasible transition pathway found by the averaging algorithm with parameter $n = 20$ between two conformations of protein 1CV2.

The dihedral angles of the special side-chain has been changed from $\chi_1 = 168.37°$, $\chi_2 = 42.51°$ in the initial conformation to $\chi_1 = 174.4°$, $\chi_2 = 68.3°$ in the alternative conformation. The potential energy of the structure changed from -410.819 to -406.236 kcal/mol. The transition pathways found by the three pathway-finding algorithms are shown in Tables 4.11-4.13

The feasible pathway contains 27 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-168.37 , 42.51]	-410.81966760
C_1	[-168.40 , 44.36]	-410.63609991
C_2	[-168.40 , 45.53]	-409.71748642
C_3	[-168.41 , 45.70]	-409.73042138
C_4	[-168.43 , 45.97]	-409.74957493
C_5	[-168.46 , 46.62]	-409.79273754
C_6	[-168.48 , 47.78]	-409.85650298
C_7	[-168.50 , 47.79]	-409.85630506
C_8	[-168.50 , 47.79]	-409.85630712
C_9	[-168.51 , 51.17]	-409.94330499
C_{10}	[-168.51 , 51.17]	-409.94330435
C_{11}	[-168.51 , 52.47]	-409.93521112
C_{12}	[-168.51 , 53.80]	-409.90214635
C_{13}	[-168.71 , 55.27]	-409.83297432
C_{14}	[-168.81 , 55.87]	-409.79550801
C_{15}	[-168.81 , 56.45]	-409.75816400
C_{16}	[-168.90 , 56.56]	-409.74604855
C_{17}	[-168.91 , 57.79]	-409.64854410
C_{18}	[-169.24 , 58.56]	-409.55271076
C_{19}	[-169.24 , 58.98]	-409.50827238
C_{20}	[-169.26 , 61.70]	-409.15806166
C_{21}	[-169.27 , 62.95]	-408.96079430
C_{22}	[-169.28 , 65.30]	-408.52306287
C_{23}	[-169.49 , 65.59]	-408.43452213
C_{24}	[-169.85 , 65.68]	-408.35897100
C_{25}	[-170.60 , 65.91]	-408.16481674
C_{26}	[-174.40 , 68.30]	-406.23611648

Table 4.12: A feasible transition pathway found by the randomized algorithm with parameter *diff*=5 between two conformations of protein 1CV2.

As can be verified from these results, the energy of all intermediate conformations are close to the energy of the initial conformation and thus the pathways are feasible.

The feasible pathway contains 17 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-168.37 , 42.51]	-410.81966760
C_1	[-168.37 , 43.80]	-410.69641386
C_2	[-168.37 , 45.09]	-409.68311333
C_3	[-168.37 , 46.38]	-409.77841635
C_4	[-168.37 , 47.67]	-409.85301799
C_5	[-168.37 , 48.96]	-409.90646071
C_6	[-168.37 , 50.25]	-409.93813560
C_7	[-168.68 , 51.54]	-409.93768252
C_8	[-168.98 , 51.54]	-409.92546850
C_9	[-169.28 , 51.54]	-409.91078828
C_{10}	[-169.58 , 51.54]	-409.89348759
C_{11}	[-169.88 , 51.54]	-409.87395195
C_{12}	[-170.18 , 51.54]	-409.85571916
C_{13}	[-170.48 , 51.54]	-409.83496357
C_{14}	[-170.78 , 52.83]	-409.78903649
C_{15}	[-171.09 , 54.12]	-409.71070030
C_{16}	[-171.39 , 55.41]	-409.59823478
C_{17}	[-171.69 , 56.70]	-409.44965416
C_{18}	[-171.99 , 57.99]	-409.26377632
C_{19}	[-172.29 , 59.27]	-409.03885665
C_{10}	[-172.59 , 60.56]	-408.77312465
C_{11}	[-172.89 , 61.85]	-408.46539127
C_{12}	[-173.19 , 63.14]	-408.11378210
C_{13}	[-173.50 , 64.43]	-407.71682439
C_{14}	[-173.80 , 65.72]	-407.27233613
C_{15}	[-174.10 , 67.01]	-406.77944668
C_{16}	[-174.40 , 68.30]	-406.23611648

Table 4.13: A feasible transition pathway found by the greedy algorithm with parameter $\alpha = 20$ between two conformations of protein 1CV2.

The feasible pathway contains 21 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-173.43 , 58.54]	-27.32941279
C_1	[-173.26 , 58.82]	-27.34942089
C_2	[-173.09 , 59.10]	-27.36770280
C_3	[-172.92 , 59.38]	-27.38423840
C_4	[-172.75 , 59.65]	-27.39897198
C_5	[-172.58 , 59.93]	-27.41196152
C_6	[-172.40 , 60.21]	-27.42318307
C_7	[-172.23 , 60.49]	-27.43264964
C_8	[-172.06 , 60.77]	-27.44033997
C_9	[-171.89 , 61.04]	-27.44623706
C_{10}	[-171.72 , 61.32]	-27.45039842
C_{11}	[-171.55 , 61.60]	-27.45275351
C_{12}	[-171.37 , 61.88]	-27.45338106
C_{13}	[-171.20 , 62.16]	-27.45221853
C_{14}	[-171.03 , 62.43]	-27.44934651
C_{15}	[-170.86 , 62.71]	-27.44470455
C_{16}	[-170.69 , 62.99]	-27.43829199
C_{17}	[-170.52 , 63.27]	-27.43013062
C_{18}	[-170.34 , 63.54]	-27.42009734
C_{19}	[-170.17 , 63.82]	-27.40823935
C_{20}	[-170.00 , 64.10]	-27.39455827

Table 4.14: A feasible transition pathway found by the averaging algorithm with parameter $n = 20$ between two conformations of protein 1CV4.

- **Protein 1CV4**

  Another instance considered by the tunnel-widening algorithm was protein 1CV4 with the starting point at position (36,7,8). While the widest tunnel in the initial conformation has width 0.57 Å the tunnel-widening algorithm found a tunnel of width 0.84 Åin an alternative conformation of 1CV4.

The feasible pathway contains 19 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-173.43 , 58.54]	-27.32941279
C_1	[-173.33 , 58.88]	-27.34977173
C_2	[-173.33 , 60.22]	-27.40274214
C_3	[-173.33 , 60.37]	-27.40751114
C_4	[-173.33 , 61.03]	-27.42459482
C_5	[-173.33 , 62.24]	-27.44285257
C_6	[-173.33 , 62.35]	-27.44367747
C_7	[-173.32 , 62.97]	-27.44556540
C_8	[-173.32 , 63.49]	-27.44375686
C_9	[-173.31 , 63.51]	-27.44381013
C_{10}	[-173.30 , 63.54]	-27.44370540
C_{11}	[-173.29 , 63.69]	-27.44269976
C_{12}	[-173.27 , 63.82]	-27.44170392
C_{13}	[-173.05 , 63.89]	-27.44465131
C_{14}	[-172.58 , 64.01]	-27.44764543
C_{15}	[-172.25 , 64.02]	-27.44832651
C_{16}	[-172.00 , 64.03]	-27.44719045
C_{17}	[-170.56 , 64.05]	-27.41832578
C_{18}	[-170.00 , 64.10]	-27.39455827

Table 4.15: A feasible transition pathway found by the randomized algorithm with parameter *diff*=2 between two conformations of protein 1CV4.

The dihedral angles of the special side-chain in the initial conformation of 1CV4 are $\chi_1 = 173.4°$ and $\chi_2 = 58.5°$ and the corresponding angles in the alternative conformation are $\chi_1 = 170.0°$ and $\chi_2 = 64.1°$. The potential energy of the structure changed from -27.329 to -27.394 kcal/mol. Tables 4.14-4.16 show the transition pathways found by the pathway-finding algorithms. According to these results, the pathways discovered by all algorithms are feasible.

The feasible pathway contains 25 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-173.43 , 58.54]	-27.32941279
C_1	[-173.26 , 58.82]	-27.34942089
C_2	[-173.09 , 59.10]	-27.36770280
C_3	[-172.92 , 59.38]	-27.38423840
C_4	[-172.75 , 59.65]	-27.39897198
C_5	[-172.58 , 59.93]	-27.41196152
C_6	[-172.40 , 60.21]	-27.42318307
C_7	[-172.23 , 60.49]	-27.43264964
C_8	[-172.06 , 60.77]	-27.44033997
C_9	[-171.89 , 61.04]	-27.44623706
C_{10}	[-171.89 , 61.32]	-27.45061219
C_{11}	[-171.89 , 61.60]	-27.45409249
C_{12}	[-171.89 , 61.88]	-27.45669655
C_{13}	[-171.89 , 62.16]	-27.45837160
C_{14}	[-171.72 , 62.43]	-27.45832214
C_{15}	[-171.55 , 62.43]	-27.45694666
C_{16}	[-171.37 , 62.43]	-27.45496481
C_{17}	[-171.20 , 62.43]	-27.45242213
C_{18}	[-171.03 , 62.43]	-27.44934651
C_{19}	[-170.86 , 62.71]	-27.44470455
C_{20}	[-170.69 , 62.99]	-27.43829199
C_{21}	[-170.52 , 63.27]	-27.43013062
C_{22}	[-170.34 , 63.54]	-27.42009734
C_{23}	[-170.17 , 63.82]	-27.40823935
C_{24}	[-170.00 , 64.10]	-27.39455827

Table 4.16: A feasible transition pathway found by the greedy algorithm with parameter $\alpha = 20$ between two conformations of protein 1CV4.

- **Protein 1CSW**

  In Subsection 4.2.2 we reported that the width of the widest tunnel increases from 0.46 Å in the initial conformation to 0.61 Å in an alternative conformation of 1CSW. The dihedral angles of the special side-chain are change from $\chi_1 = -59.50^\circ$, $\chi_2 =$

$-157.14°$, $\chi_3 = -65.70°$, and $\chi_4 = -78.72°$ in the initial conformation to $\chi_1 = -69.10°$, $\chi_2 = -179.40°$, $\chi_3 = -70.90°$, and $\chi_4 = 169.90°$ in the target conformation. The potential energy of the structure changed from 88.119 to 91.691 kcal/mol. Thus, the tunnel-widening algorithm finds an acceptable alternative conformation with a wider tunnel. Next we use the pathway-finding algorithms to test whether this alternative conformation is accessible from the initial conformation. Table 4.17 shows the transition pathway found by the averaging algorithm with parameter $n = 28$. According to these results, energy of several intermediate conformations are much higher than the energy of the initial conformation. For example the potential energy of C_8 is 523.275 kcal/mol. Thus the discovered pathway is not feasible. The pathway found by the randomized algorithm with parameter *diff*=28 is shown in Table 4.18. Similar to the pathway found by the averaging algorithm, this pathway contains several intermediate conformations with energies much higher than the energy of the initial conformation. The maximum potential energy of intermediate conformations is 999.928 kcal/mol and belongs to C_{25}. Therefore, the randomized algorithm does not discover a feasible pathway. The pathway found by the greedy algorithm with parameter $\alpha = 12$ is shown in Table 4.19. This pathway contains 29 conformations. Several intermediate conformations have potential energies much higher than the energy of the initial conformation. Thus the discovered pathway is not feasible. However, observe that the maximum energy of the conformations in this pathway is 325.435 kcal/mol which is much lower than the maximum energy of the pathway found by the averaging and randomized algorithms. This example shows the ability of the greedy algorithm to find better pathways compared to the other two algorithms. Furthermore, this example shows that in some cases we have an energy barrier between two conformations C and C', even though the potential energies of C and C' are close.

The feasible pathway contains 29 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-59.50 , -157.14 , -65.70 , -78.72]	88.11877308
C_1	[-59.84 , -157.94 , -65.88 , -69.84]	88.76312274
C_2	[-60.19 , -158.73 , -66.07 , -60.96]	91.12595918
C_3	[-60.53 , -159.53 , -66.25 , -52.09]	98.61611918
C_4	[-60.87 , -160.32 , -66.44 , -43.21]	130.92369780
C_5	[-61.21 , -161.12 , -66.62 , -34.33]	229.95535485
C_6	[-61.56 , -161.91 , -66.81 , -25.45]	370.21796456
C_7	[-61.90 , -162.71 , -67.00 , -16.57]	466.92597221
C_8	[-62.24 , -163.50 , -67.18 , -7.69]	523.27534913
C_9	[-62.59 , -164.30 , -67.37 , 1.19]	520.15920069
C_{10}	[-62.93 , -165.09 , -67.55 , 10.07]	435.63204139
C_{11}	[-63.27 , -165.89 , -67.74 , 18.95]	318.80960351
C_{12}	[-63.61 , -166.68 , -67.92 , 27.83]	220.36323140
C_{13}	[-63.96 , -167.48 , -68.11 , 36.71]	156.01762667
C_{14}	[-64.30 , -168.27 , -68.30 , 45.59]	128.44639966
C_{15}	[-64.64 , -169.07 , -68.48 , 54.47]	114.70583466
C_{16}	[-64.99 , -169.86 , -68.67 , 63.35]	106.31996194
C_{17}	[-65.33 , -170.66 , -68.85 , 72.22]	102.70208190
C_{18}	[-65.67 , -171.45 , -69.04 , 81.10]	110.81705131
C_{19}	[-66.01 , -172.25 , -69.22 , 89.98]	170.45693121
C_{20}	[-66.36 , -173.04 , -69.41 , 98.86]	273.74613012
C_{21}	[-66.70 , -173.84 , -69.60 , 107.74]	349.90903557
C_{22}	[-67.04 , -174.63 , -69.78 , 116.62]	363.59276044
C_{23}	[-67.39 , -175.43 , -69.97 , 125.50]	295.21575579
C_{24}	[-67.73 , -176.22 , -70.15 , 134.38]	197.44499990
C_{25}	[-68.07 , -177.02 , -70.34 , 143.26]	114.48624372
C_{26}	[-68.41 , -177.81 , -70.52 , 152.14]	91.23784450
C_{27}	[-68.76 , -178.61 , -70.71 , 161.02]	89.40702470
C_{28}	[-69.10 , -179.40 , -70.90 , 169.90]	91.68776538

Table 4.17: A feasible transition pathway found by the averaging algorithm with parameter $n = 28$ between two conformations of protein 1CSW.

The feasible pathway contains 31 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-59.50 , -157.14 , -65.70 , -78.72]	88.11877308
C_1	[-59.50 , -157.19 , -66.36 , -74.91]	88.19091383
C_2	[-59.50 , -157.20 , -66.37 , -54.46]	94.36287390
C_3	[-59.50 , -157.20 , -66.40 , -48.95]	100.08907109
C_4	[-59.50 , -157.21 , -66.40 , -41.16]	125.94322980
C_5	[-59.51 , -157.21 , -66.40 , -28.29]	284.74896125
C_6	[-59.51 , -157.21 , -66.40 , -28.26]	285.24433610
C_7	[-59.51 , -157.21 , -66.42 , -11.22]	432.75141371
C_8	[-59.51 , -157.22 , -66.43 , -7.23]	416.40757258
C_9	[-59.51 , -157.22 , -66.49 , -4.61]	393.19411797
C_{10}	[-59.51 , -157.22 , -66.58 , 7.96]	252.22855650
C_{11}	[-59.51 , -157.23 , -66.96 , 17.21]	154.30963479
C_{12}	[-59.52 , -157.23 , -66.98 , 22.42]	123.65174902
C_{13}	[-59.54 , -157.27 , -67.00 , 30.35]	108.74681393
C_{14}	[-59.60 , -157.35 , -67.00 , 37.35]	105.05128563
C_{15}	[-59.60 , -157.36 , -68.13 , 40.09]	103.90127575
C_{16}	[-59.61 , -157.36 , -68.53 , 40.12]	103.73521650
C_{17}	[-59.61 , -157.36 , -68.56 , 42.01]	103.32336404
C_{18}	[-59.61 , -157.37 , -68.56 , 62.68]	123.12727228
C_{19}	[-59.61 , -157.37 , -68.58 , 81.20]	360.94924086
C_{20}	[-59.61 , -157.37 , -68.66 , 84.09]	432.25797304
C_{21}	[-59.61 , -157.37 , -68.73 , 84.63]	446.12090322
C_{22}	[-59.61 , -157.37 , -68.83 , 86.67]	500.90401542
C_{23}	[-59.62 , -157.42 , -69.34 , 87.86]	526.14192830
C_{24}	[-59.64 , -158.60 , -69.93 , 97.61]	786.79380396
C_{25}	[-59.75 , -158.88 , -70.65 , 113.28]	999.92742382
C_{26}	[-60.29 , -159.12 , -70.85 , 120.27]	872.11380927
C_{27}	[-60.85 , -165.35 , -70.85 , 146.45]	246.95435113
C_{28}	[-61.11 , -172.05 , -70.86 , 148.03]	128.06620486
C_{29}	[-66.98 , -178.67 , -70.87 , 149.47]	92.98776895
C_{30}	[-69.10 , -179.40 , -70.90 , 169.90]	91.69121722

Table 4.18: A feasible transition pathway found by the randomized algorithm with parameter *diff*=28 between two conformations of protein 1CSW.

The feasible pathway contains 29 conformations as follows:

Conformation	Chi angles of the special side-chain	Energy (kcal/mol)
C_0	[-59.50 , -157.14 , -65.70 , -78.72]	88.11877308
C_1	[-60.30 , -157.14 , -65.70 , -78.72]	87.94953520
C_2	[-61.10 , -157.14 , -65.70 , -78.72]	87.80912306
C_3	[-61.90 , -157.14 , -65.70 , -78.72]	87.72126419
C_4	[-62.70 , -157.14 , -65.70 , -78.72]	87.68187487
C_5	[-63.50 , -157.14 , -65.70 , -78.72]	87.67098305
C_6	[-64.30 , -157.14 , -65.70 , -78.72]	87.69847516
C_7	[-65.10 , -157.14 , -65.70 , -78.72]	87.75758846
C_8	[-65.90 , -157.14 , -65.70 , -58.00]	90.49567308
C_9	[-65.90 , -157.14 , -65.70 , -37.28]	113.65927925
C_{10}	[-65.90 , -157.14 , -65.70 , -16.56]	325.43516078
C_{11}	[-65.90 , -157.14 , -65.70 , 4.15]	325.01742525
C_{12}	[-65.90 , -157.14 , -65.70 , 24.87]	121.54359834
C_{13}	[-66.70 , -158.99 , -66.13 , 45.59]	101.56423665
C_{14}	[-66.70 , -160.85 , -66.13 , 45.59]	101.38880312
C_{15}	[-66.70 , -162.70 , -66.57 , 66.31]	100.63508544
C_{16}	[-66.70 , -164.56 , -67.00 , 66.31]	97.46533112
C_{17}	[-66.70 , -166.41 , -67.43 , 66.31]	96.65485362
C_{18}	[-66.70 , -168.27 , -67.87 , 66.31]	98.50112297
C_{19}	[-66.70 , -170.12 , -68.30 , 66.31]	104.83819265
C_{20}	[-67.50 , -171.98 , -68.73 , 66.31]	122.13109030
C_{21}	[-68.30 , -173.84 , -69.17 , 87.03]	121.70575588
C_{22}	[-68.30 , -173.84 , -69.60 , 87.03]	120.54790522
C_{23}	[-68.30 , -175.69 , -70.03 , 87.03]	131.22686047
C_{24}	[-68.30 , -177.55 , -70.47 , 87.03]	158.12340975
C_{25}	[-69.10 , -179.40 , -70.90 , 107.75]	197.71114282
C_{26}	[-69.10 , -179.40 , -70.90 , 128.46]	183.57063713
C_{27}	[-69.10 , -179.40 , -70.90 , 149.18]	91.74556571
C_{28}	[-69.10 , -179.40 , -70.90 , 169.90]	91.68776538

Table 4.19: A feasible transition pathway found by the greedy algorithm with parameter $\alpha = 12$ between two conformations of protein 1CSW.

Chapter 5

Conclusions

In this thesis we developed efficient algorithms for finding and widening tunnels in protein structures. Given a fixed protein conformation and a starting point inside it, the tunnel-finding algorithm can compute the widest tunnel from the starting point to the outside environment of the protein. Then the tunnel-widening algorithm explores the possibility that a small local change in the structure of the protein conformation might lead to a wider tunnel. More specifically, it considers some alternative conformations obtained by relocating the bottleneck side-chain atoms and picks the conformation with the widest tunnel whose energy is not much higher than the energy of the initial conformation. We also proposed algorithms for finding feasible transition pathways between the initial structure and an alternative conformation to make sure that the alternative conformation is accessible from the initial conformation. More specifically, we introduced three pathway-finding algorithms: averaging, randomized, and greedy algorithms. While averaging and randomized algorithms have better running time, the greedy algorithm gives the most accurate results. Therefore, there is a trade-off between the running time and accuracy of the algorithms.

We implemented these algorithms in Chimera/Python and tested them on various input instances. In all cases the tunnel-finding algorithm computed the widest tunnel if it exists. Note that for some combinations of the protein conformation and the starting point there is no tunnel from the starting point to the outside environment. The tunnel-widening algorithm was able to widen the tunnel in most cases. There were a few cases for which the tunnel-widening algorithm found an alternative conformation C with a wider tunnel but the energy of C was much higher than the energy of the initial conformation. We also used the pathway-finding algorithms to verify that the alternative conformation with wider tunnel and acceptable energy value is actually accessible from the initial conformation, i.e., there is a feasible transition pathway from the initial structure to the alternative conformation. Although in most cases our algorithms were able to find a feasible transition pathway from the initial structure to the alternative conformation, there were cases in which no feasible transition pathway from the initial structure to the alternative conformation was found by our algorithms. Furthermore the three pathway-finding algorithms had comparable performance in most cases, but there were a few input instances for which the greedy algorithm outperformed the averaging and randomized algorithms.

We should point out that we only concentrate on the algorithmic aspects of the tunnel-finding and tunnel-widening problems. In particular, finding a tunnel that is wide enough for a ligand does not guarantee that in real life the ligand actually passes through this tunnel. Various biological factors affect the actual behaviour of ligands. Considering these factors is beyond the scope of this thesis and can be considered as a future work.

One potential extension to our work is to remove the following simplifying assumption that we made in our computations. We modelled the ligand by a sphere enclosing all the ligand atoms. A more accurate model is to consider the actual shape of the ligand. Note that this makes the problem much more complicated as the orientation of the ligand during its movement can influence the feasibility of the tunnel.

References

[1] S. Aluru. *Handbook of computational molecular biology*. Chapman and Hall/CRC, Boca Raton, FL, 2006. 10

[2] C. Alvarado and K. Kazerounian. On the rotational operators in protein structure simulations. *Protein Engineering*, 16(10):717–720, 2003. 8

[3] B. Aronov. A lower bound on voronoi diagram complexity. *Information Processing Letters*, 83:183–185, 2002. 18

[4] F. Aurenhammer and R. Klein. Voronoi diagrams. In J.R. Sack and J.B. Urrutia, editors, *Handbook of Computational Geometry*, pages 201 – 290. Elsevier Science Publishers B.V., 2000. 15, 18

[5] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000. 13, 53

[6] A. Bondi. van der waals volumes and radii. *The Journal of Physical Chemistry*, 68(3):441–451, 1964. 25

[7] P. E. Bourne and H. Weissig. *Biochemistry*. John Wiley and Sons, Ltd., Hoboken, NJ, 2003. 11

[8] P. M. Bowers, C. E. M. Strauss, and D. Baker. De novo protein structure determination using sparse nmr data. *Journal of Biomolecular NMR*, 18:311–318, 2000. 56

[9] F. J. Burkowski. *Structural bioinformatics : an algorithmic approach*. CRC Press, Boca Raton, FL, 2009. 8, 10

[10] E. Buxbaum. *Fundamentals of Protein Structure and Function*. Springer, New York, NY, 2007. 9

[11] S. Chaudhury, S. Lyskov, and J. J. Gray. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics*, 26(5):689–691, 2010. 54

[12] B. Chazelle. An optimal convex hull algorithm and new results on cuttings (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science*, pages 29–38, 1991. 18, 22

[13] N. H. Christ, R. Friedberg, and T. D. Lee. Random lattice field theory: General formulation. *Nuclear Physics B*, 202(1):89 – 125, 1982. 19

[14] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry II. *Discrete & Computational Geometry*, 4:387–421, 1989. 18, 22

[15] K. A. Coleman, R. G.; Sharp. Finding and characterizing tunnels in macromolecules with application to ion channels and pores. *Biophysical journal*, 96(2):632 – 645, 2009. 3

[16] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 3rd edition, 2009. 22

[17] N. J. Darby and T. E. Creighton. *Protein Structure*. Oxford University Press, New York, NY, 1940. 8

[18] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry algorithms and applications*. Springer, Berlin, 3rd edition, 2008. 15, 16, 18, 19

[19] B. Delaunay. Sur la sphère vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934. 19

[20] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 4, 22

[21] K. A. Dill and S. Bromberg. *Molecular driving forces : statistical thermodynamics in biology, chemistry, physics, and nanoscience.* Garland Science, London, 2011. 13

[22] P. G. L. Dirichlet. Über die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *J. Reine Angew. Math.*, 40:209–227, 1850. 14

[23] R. L. Dunbrack. Rotamer libraries in the 21st century. *Current Opinion in Structural Biology*, 12(4):431 – 440, 2002. 12, 34

[24] R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6(8):1661–1681, 1997. 12, 57

[25] R. L. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins. application to side-chain prediction. *Journal of Molecular Biology*, 230(2):543 – 574, 1993. 12

[26] R. L. Dunbrack and M. Karplus. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Structural Biology*, 1(5):334–340, 1994. 12

[27] R. A. Dwyer. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2:137–151, 1987. 18, 21

[28] I. Eidhammer, I. Jonassen, and W. R. Taylor. *Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis.* John Wiley and Sons, Ltd., West Sussex, UK, 3rd edition, 2004. 8

[29] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987. 18, 21

[30] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987. 23

[31] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 2nd edition, 2004. 18

[32] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and computation of voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985. 18, 21

[33] B. K. Ho, A. Thomas, and R. Brasseur. Revisiting the ramachandran plot: Hard-sphere repulsion, electrostatics, and h-bonding in the -helix. *Protein Science*, 12(11):2508–2522, 2003. 8

[34] M. K. Kim, R. L. Jernigan, and G. S. Chirikjian. Efficient generation of feasible pathways for protein conformational transitions. *Biophysical Journal*, 83(3):1620 – 1630, 2002. 4

[35] M. K. Kim, W. Li, B. A. Shapiro, and G. S. Chirikjian. A comparison between elastic network interpolation and md simulation of 16s ribosomal rna. *Journal of Biomolecular Structure and Dynamics*, 21(3):311–468, 2003. 4

[36] V. Klee. On the complexity of d-dimensional voronoi diagrams. *Archiv der Mathematik*, 34:75–80, 1980. 18

[37] G. J. Kleywegt and T. A. Jones. Detection, delineation, measurement and display of cavities in macromolecular structures. *Acta Crystallographica Section D*, 50(2):178–185, Mar 1994. 3

[38] H. KONO and J. DOI. A new method for side-chain conformation prediction using a hopfield network and reproduced rotamers. *Journal of Computational Chemistry*, 17(14):1667–1683, 1996. 12

[39] T. Kortemme, A. V. Morozov, and D. Baker. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes. *Journal of Molecular Biology*, 326(4):1239 – 1259, 2003. 56

[40] B. Kuhlman and D. Baker. Native protein sequences are close to optimal for their structures. *Proceedings of the National Academy of Sciences*, 97(19):10383–10388, 2000. 56, 57

[41] T. Lazaridis and M. Karplus. Heat capacity and compactness of denatured proteins. *Biophysical Chemistry*, 78(1-2):207 – 217, 1999. 56

[42] D. T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. *International Journal Computer and Information Sciences*, 9(3):219, 1980. 18, 21

[43] D. G. Levitt and L. J. Banaszak. POCKET: A computer graphies method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of Molecular Graphics*, 10(4):229 – 234, 1992. 3

[44] J. Liang, C. Woodward, and H. Edelsbrunner. Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science*, 7(9):1884–1897, 1998. 3

[45] A. Light. *Proteins: Structure and Function.* Prentice-Hall, Englewood Cliffs, NJ, 1974. 6

[46] S. C. Lovell, J. M. Word, J. S. Richardson, and D. C. Richardson. The penultimate rotamer library. *Proteins*, 40(3):389–408, 2000. 12

[47] M. De Maeyer, J. Desmet, and I. Lasters. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding and Design*, 2(1):53 – 66, 1997. 12

[48] J. Marek, J. Vvodov, I. K. Smatanov, Y. Nagata, L. A. Svensson, J. Newman, M. Takagi, and J. Damborsk. Crystal structure of the haloalkane dehalogenase from sphingomonas paucimobilis ut26,. *Biochemistry*, 39(46):14082–14086, 2000. 60

[49] P. Medek, P. Beneš, and J. Sochor. Computation of tunnels in protein molecules using Delaunay triangulation. *Journal of WSCG, University of West Bohemia, Pilsen*, 15(1-3):107–114, 2007. 3, 4, 5, 25

[50] J. Newman, T. S. Peat, R. Richard, L. Kan, P. E. Swanson, J. A. Affholter, I. H. Holmes, J. F. Schindler, C. J. Unkefer, and T. C. Terwilliger. Haloalkane dehalogenases: structure of a rhodococcus enzyme. *Biochemistry*, 38(49):16105–16114, 1999. 60

[51] A. J. Oakley, M. Klvaa, M. Otyepka, Y. Nagata, M. C. J. Wilce, and J. Damborsk. Crystal structure of haloalkane dehalogenase linb from sphingomonas paucimobilis ut26 at 0.95 resolution: dynamics of catalytic residues,. *Biochemistry*, 43(4):870–878, 2004. 59

[52] A. Okabe, B. N. Boots, and k. Sugihara. *Spatial tessellations : concepts and applications of Voronoi diagrams.* Wiley and Sons, Chichester, England, 1992. 14, 15, 16, 19

[53] J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai, and C. E. M. Strauss. Practical conversion from torsion space to cartesian space for in silico protein synthesis. *Journal of Computational Chemistry*, 26(10):1063–1068, 2005. 8

[54] M. Petrek, M. Otyepka, P. Banás, P. Kosinová, J. Koca, and J. Damborský. CAVER: a new tool to explore routes from protein clefts, pockets and cavities. *BMC Bioinformatics*, 7:316–324, 2006. 3

[55] M. Petrek, Kosinová P., J. Koca, and M. Otyepka. MOLE: a Voronoi diagram-based explorer of molecular channels, pores, and tunnels. *Structure*, 15(11):1357 – 1363, 2007. 3

[56] G. A. Petsko and D. Ringe. *Protein Structure and Function.* New Science Press Ltd, London, 2004. 9

[57] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF Chimera- A visualization system for exploratory

research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004. 54

[58] G.N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95 – 99, 1963. 8

[59] G.N. Ramachandran and V. Sasisekharan. Conformation of polypeptides and proteins. volume 23 of *Advances in Protein Chemistry*, pages 283 – 437. Academic Press, 1968. 8

[60] C. A. Rohl, C. E.M. S., K. M.S. Misura, and D. Baker. Protein structure prediction using Rosetta. In L. Brand and M. L. Johnson, editors, *Numerical Computer Methods, Part D*, volume 383 of *Methods in Enzymology*, pages 66 – 93. Academic Press, 2004. 54, 55, 57

[61] C. A. Rohl, C. E. M. Strauss, D. Chivian, and D. Baker. Modeling structurally variable regions in homologous proteins with Rosetta. *Proteins: Structure, Function, and Bioinformatics*, 55(3):656–677, 2004. 56

[62] J. Schlitter, M. Engels, and P. Krger. Targeted molecular dynamics: A new approach for searching pathways of conformational transitions. *Journal of Molecular Graphics*, 12(2):84 – 89, 1994. 4

[63] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete & Computational Geometry*, 6:423–434, 1991. 18, 22

[64] R. Seidel. The upper bound theorem for polytopes: An easy proof of its asymptotic version. *Computational Geometry: Theory and Applications*, 5:115–116, 1995. 20

[65] M. I. Shamos and D. Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science*, pages 151–162, 1975. 18, 21

[66] O. S. Smart, J. G. Neduvelil, X. Wang, B. A. Wallace, and M. S. P. Sansom. HOLE: a program for the analysis of the pore dimensions of ion channel structural models. *Journal of Molecular Graphics*, 14(6):354–360, 1996. 3

[67] F. W. Smith. The structure of aggregates and the molecular kinematics of the viscosity of a bernal liquid. *Canadian Journal of Physics*, 7:793–800, 1964. 19

[68] C. S. Tsai. *Biomacromolecules: introduction to structure, function and informatics.* Wiley-Liss, New York, 2007. 1

[69] D. Voet and J. G. Voet. *Biochemistry.* John Wiley and Sons, Ltd., Hoboken, NJ, 3rd edition, 2004. 11

[70] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques — premier Mémoire: Sur quelques propriétés des formes quadtratiques positives parfaites. *J. Reine Angew. Math.*, 133:97–178, 1907. 14

[71] G. Voronoi. Nouvelles applications des parametres continus a la theorie des formes quadratiques. *J. f. d. Reine und Angewandte Mathematik*, 134:198–287, 1908. 14, 19

[72] D. J. Wales. *Energy landscapes.* Cambridge University Press, Cambridge, UK, 2003. 13

[73] W. J. Wedemeyer and D. Baker. Efficient minimization of angle-dependent potentials for polypeptides in internal coordinates. *Proteins: Structure, Function, and Bioinformatics*, 53(2):262–272, 2003. 56

[74] B Wroblowski, J F Diaz, J Schlitter, and Y Engelborghs. Modelling pathways of alpha-chymotrypsin activation and deactivation. *Protein Engineering*, 10(10):1163–1174, 1997. 4

[75] E. Yaffe, D. Fishelovitch, H. J. Wolfson, D. Halperin, and R. Nussinov. MolAxis: Efficient and accurate identification of channels in macromolecules. *Proteins: Structure, Function, and Bioinformatics*, 73(1):72–86, 2008. 3

[76] M. Zhang and L. E. Kavraki. A new method for fast and accurate derivation of molecular conformations. *Journal of Chemical Information and Computer Sciences*, 42(1):64–70, 2002. 8