Efficient exploration of molecular paths from As-Rigid-As-Possible approaches and motion planning methods

Minh Khoa Nguyen

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Minh Khoa NGUYEN

Thèse dirigée par **Emmanuel MAZER**, DR, CNRS
et codirigée par **Léonard JAILLET**

préparée au sein du **Laboratoire Laboratoire Jean Kuntzmann**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Exploration efficace de chemins moléculaires par approches aussi-rigides-que-possibles et par méthodes de planification de mouvements

Efficient exploration of molecular paths from As-Rigid-As-Possible approaches and motion planning methods

Thèse soutenue publiquement le **15 mars 2018**,
devant le jury composé de :

Monsieur EMMANUEL MAZER
DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES, Directeur de thèse
Monsieur LEONARD JAILLET
INGENIEUR DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Examinateur
Monsieur STEPHANE REDON
DIRECTEUR DE RECHERCHE, INRIA CENTRE DE GRENOBLE RHÔNE-ALPES, Co-directeur de thèse
Monsieur JUAN CORTES
DIRECTEUR DE RECHERCHE, CNRS DELEGATION MIDI-PYRENEES, Rapporteur
Monsieur CHARLES ROBERT
DIRECTEUR DE RECHERCHE, CNRS DELEGATION PARIS, Rapporteur, Président du jury
Monsieur DIRK STRATMANN
MAITRE DE CONFERENCES, UNIVERSITE PIERRE ET MARIE CURIE, Examinateur

Efficient exploration of molecular paths from As-Rigid-As-Possible approaches and motion planning methods.

Minh Khoa Nguyen

Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, Université Grenoble Alpes
Université Grenoble Alpes

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2018

I would like to dedicate this thesis to my loving family, and BoBo. . .

Acknowledgements

First of all, I am very grateful to Inria Grenoble Rhône-Alpes, the Laboratoire Jean Kuntzmann and Université Grenoble Alpes for giving me this opportunity.

I would like to express my indebtedness to Dr. Léonard Jaillet for tremendous help and support, without whom I would not have been able to finish this work.

I would also like to thank Dr. Stéphane Redon for helpful advice and the amazing SAMSON platform where I developed all of the proposed methods in this manuscript.

My thanks to Dr. Emmanuel Mazer for helping us to advance the project and giving useful advice in the regular meetings.

I am also grateful to Mme Imma Preseguer for adminstrative support, M. Jocelyn Gaté for technical support, Sergei Grudinin for his expertise, Guillaume Pagès for the design of the toy model used in this thesis, and Clement Beitone for the implementation of the FIRE and Nudged Elastic Band methods on the SAMSON platform.

I want to thank all the NANO-D members, as well as friends from the CORSE team, for giving help and support, for sharing time and knowledge. I treasured all the time spending with them: Yassine Naimi, Dmitriy Marin, Sivia C. Dias Pinto, Semeho Edorh, Alexandre Hoffmann, Maria Kadukova, Emilie Neveu, François Rousse, Krishna Kant Singh, Žofia Trs'anová, Nadhir Abdellatif, Svetlana Artemova, Marc Aubert, Julie Bourget, Mohamed Yengui, Léonard Jaillet, Stéphane Redon, Imma Preseguer, Sergei Grudinin, Guillaume Pagès, Clement Beitone and Fabian Gruber.

Finally, I would like to express my gratitude to the reviewers and examiners in the jury, Dr. Charles Robert, Dr. Juan Cortés, and Dr. Dirk Stratmann for their questions and examination of my work so that I could improve the quality of this manuscript.

Abstract

Proteins are macromolecules participating in important biophysical processes of living organisms. It has been shown that changes in protein structures can lead to changes in their functions and are found linked to some diseases such as those related to neurodegenerative processes. Hence, an understanding of their structures and interactions with other molecules such as ligands is of major concern for the scientific community and the medical industry for inventing and assessing new drugs.

In this dissertation, we are particularly interested in developing new methods to find for a system made of a single protein or a protein and a ligand, the pathways that allow a transition from one state to another. During a few past decades, a vast amount of computational methods has been proposed to address this problem. However, these methods still have to face two challenges: the high dimensionality of the representation space, associated to the large number of atoms in these systems, and the complexity of the interactions between these atoms.

This dissertation proposes two novel methods to efficiently find relevant pathways for such biomolecular systems. The methods are fast and their solutions can be used, analyzed or improved with more specialized methods. The first proposed method generates interpolation pathways for biomolecular systems using the As-Rigid-As-Possible (ARAP) principle from computer graphics. The method is robust and the generated solutions best preserve the local rigidity of the original system. An energy-based extension of the method is also proposed, which significantly improves the solution paths. However, in the scenarios requiring complex deformations, this approach may still generate unnatural paths. Therefore, we propose a second method called ART-RRT, which combines the ARAP principle for reducing the dimensionality, with the Rapidly-exploring Random Trees from robotics for efficiently exploring possible pathways. This method not only gives a variety of pathways in reasonable time but the pathways are also low-energy and clash-free, with the local rigidity preserved as much as possible. The mono-directional and bi-directional versions of the ART-RRT method were applied for finding ligand-unbinding and protein conformational transition pathways, respectively. The results were found to be in good agreement with experimental data and other state-of-the-art solutions.

Table of contents

List of figures

List of tables

Nomenclature

Abbreviations

2D two-dimensional

3D three-dimensional

A-atom active ARAP atom

ARAP As-Rigid-As-Possible

ARAPi ARAP interpolation

ARAPm ARAP modeling

ART-RRT Our proposed method combining the ARAP principle with a RRT variant, for exploring pathways in high-dimensional systems

DoF Degrees of Freedom

ENM Elastic Network Model

FEL Free Energy Landscape

FIRE Fast Inertial Relaxation Engine

MC Monte Carlo

MD Molecular Dynamics

MEP Minimum-Energy Path

N-atom non-ARAP atom

NEB Nudged Elastic Band

NMA Normal Mode Analysis

P-atom passive ARAP atom

PEL Potential Energy Landscape

RAMD Random Acceleration Molecular Dynamics

RMSD Root Mean Square Distance

RRT Rapidly-exploring Random Tree

Slerp Spherical linear interpolation

SMD Steered Molecular Dynamics

T-RRT Transition-based Rapidly-exploring Random Tree

TMD Targeted Molecular Dynamics

Symbols

δ RRT extension step size

$\mathcal{S}$ maximum number of failures in transition test

γ threshold parameter for determining threshold energy in bi-directional ART-RRT

λ temperature factor for regulating the temperature in transition test

$\mathbf{p_i}$ 3D position of the ith atom

$\mathbf{R_i}$ 3D aligning rotation of the ARAP set $\mathcal{N}_i$

$\mathcal{N}_i$ ARAP set with v_i as the central vertex

$\mathbb{R}^3$ Real coordinate space of three dimensions

C_i ARAP cell containing the initial positions of the vertices in ARAP set $\mathcal{N}_i$

C_i' ARAP cell containing the target positions of the vertices in ARAP set $\mathcal{N}_i$

E Energy

k Boltzmann constant

m number of solving iterations in ARAP modeling

n_F number of time steps for the constrained minimization

T	temperature in transition test
t	interpolation instance
t_F	integration time step for the constrained minimization
v_i	ith vertex in a mesh
C_α	Alpha carbon

Part I

Foundations

Chapter 1

Motivation and Contribution

Structural biology is the study of biomolecular structures and how their structural changes affect their functions [15]. These molecules range from small systems with a few atoms (alkaloids, lipids, steroids, etc.) to large systems with thousands or more atoms (lipids, proteins, nucleic acids, etc.).

In this thesis, we are particularly interested in proteins and their interactions with ligands. Proteins, the building and maintaining materials for living organisms, participate in many life processes such as metabolisms, nutrient transports, hormone regulation, and so on. Ligands are molecules which, upon binding to proteins, may initiate structural changes in the proteins, and hence, their functions. Besides ensuring proper biophysical functions in living organisms, it has been shown that protein structures also play important roles in diseases such as cancer, Parkinson, diabetes, etc. [207]. Therefore, an understanding of their structures and interactions with ligands is of major concern for the scientific community and the medical industry for inventing and assessing new drugs.

During the last century, experimental methods such as protein crystallography, electron microscopy, nuclear magnetic resonance and neutron diffraction have continuously been improved. This allows researchers to assess huge databases of biomolecular structures including protein structures such as the Protein Data Bank [22]. To keep up with the database growth, computational methods have been devised to process and interpret this huge information *in silico*, i.e. with computers. Many biochemical reactions can now be successfully simulated [83, 168, 151]. As a consequence, research costs, especially in drug developments, can be reduced tremendously since computer simulations can help predict the success or failure of new compounds before the laboratory phase. This is a great advantage because the drug development is known to be a costly process in time and budget. This process takes about 8-12 years and the cost can go over 500 million US dollars for a new

drug to go from laboratory to market, yet only one out of 5000-10000 proposed compounds can make it to the customers [154].

Despite the advancement of computer technology and resources, computer simulations involving large molecules such as proteins still pose great challenges for current simulation techniques because of two major issues: the high dimensionality of the spaces representing biomolecular systems due to high number of atoms, and the complex interactions among these atoms governed by physical laws.

The aim of this doctoral thesis is, therefore, to propose new algorithmic methods to find (molecular) pathways for high-dimensional systems, in particular, the protein structural rearrangement and protein-ligand interaction problems. The originality of this research is that it brings together two different disciplines: computer graphics and robotics. From computer graphics, the *As-Rigid-As-Possible* (ARAP) principle is used for dimensionality reduction and from robotics, *Rapidly-exploring Random Tree* (RRT) is applied for searching possible pathways.

Our research led to two principal contributions:

- The ARAP interpolation method for molecular structures, which is a fast and robust method inspired by the ARAP principle for finding interpolation pathways between two given molecular structures.
- The ART-RRT method, which combines the ARAP principle with a RRT variant, for exploring pathways in high-dimensional systems.

The proposed methods presented in this manuscript were implemented in the SAMSON platform [113] developed by the NANO-D team at INRIA Rhône-Alpes where I spent my PhD study.

The manuscript is organized into four parts:

- Part I presents the foundations for this thesis study. This part contains 3 chapters. Chapter 1 gives the motivation, contribution and organization of the manuscript. Chapter 2 introduces the scientific background necessary to understand this work. Chapter 3 presents the algorithmic frameworks of the ARAP and RRT methods on which are based our contributions.
- Part II presents our proposed application of ARAP interpolation for finding molecular interpolation pathways (Chapter 4). This work has been published [174] in the journal

of Computer-Aided Molecular Design. It was also presented as posters in the 19th conference of Groupe Graphisme & Modélisation Moléculaire (GGMM) 2015 and in the 2015 3DSIG conference. This part also presents an energy-based enhancement (Chapter 5) of the method.

- Part III presents the proposed ART-RRT method and its applications. In this part, Chapter 6 describes the method in detail. Chapter 7 applies the mono-directional version of ART-RRT for exploring a simple energy landscape and for finding ligand unbinding pathways. The latter application has been published [175] in the Journal of Computational Chemistry. Chapter 8 presents the application of the bi-directional version of ART-RRT to find pathways for a simple toy model, for protein-ligand interactions, and for protein conformational transitions. The application of bi-directional ART-RRT for protein-ligand interactions was presented in an oral session of the 20th conference of Groupe Graphisme & Modélisation Moléculaire (GGMM) 2017.

- Finally, Part IV concludes the thesis and suggests some perspectives for future work.

Chapter 2

Scientific Background

In this chapter, proteins and ligands, the biomolecules of interest in this thesis, are first introduced. Then, the models to represent biomolecular systems and their interactions are presented. Finally, a survey of state-of-the-art methods for finding molecular pathways is given.

2.1 Protein and Ligand

Proteins are responsible for many cellular functions and are found in all body tissues including muscles, hair, eyes, skin, etc. [102]. Among the important ones, enzymes are catalyzing molecules for accelerating biochemical reactions and hormones are signaling molecules for regulating biophysical processes. Because changes in protein structures can lead to changes in their biophysical functions [8], an understanding of this link is essential for protein engineering and drug research.

Proteins are synthesized from encoded information in genes. From the atomistic point of view, a protein is a macromolecule consisting of a linear chain of amino acids (or residues) called polypeptide. Most amino acids contain 3 groups: an amine ($-NH_2$) group, a carboxyl ($-COOH$) group and a specific *side chain* (Figure 2.1a). These groups are bonded to a common carbon called alpha carbon (C_α). Although there exist more than 500 amino acids in nature, only 20 amino acids are found in proteins [236]. Within a polypeptide, two consecutive amino acids are bonded by a peptide bond ($O=C-NH$). The linear chain ($-N-C_\alpha-C-$) in a polypeptide is called *backbone*. The peptide bond and backbone of a polypeptide are shown in Figure 2.1b.

Since the polypeptide bond is quite rigid, the flexibility in protein backbones manifests itself through rotations about $N-C_\alpha$ and $C_\alpha-C$ bonds in each amino acid. These rotations are quantified by the dihedral angles Φ and Ψ, respectively. For the ith amino acid in a

polypeptide, Φ is the angle between the planes made by $\{C^{i-1}, N^i, C_\alpha^i\}$ and $\{N^i, C_\alpha^i, C^i\}$ while Ψ is the angle between the planes made by $\{N^i, C_\alpha^i, C^i\}$ and $\{C_\alpha^i, C^i, N^{i+1}\}$ (see Figure 2.2b). N^i, C_α^i and C^i are the backbone nitrogen, alpha carbon and carbon atoms of the i^{th} amino acid. C^{i-1} and N^{i+1} are the backbone carbon and nitrogen atoms of the $(i-1)^{th}$ and $(i+1)^{th}$ amino acids, respectively. Ramachandran observed from protein structures that these angles are limited in certain ranges [190]. The Ramachadran plot [191] (Figure 2.2a), which displays Ψ versus Φ angles for all the amino acids in a protein, is used for showing valid regions of these angles or validating proposed models.

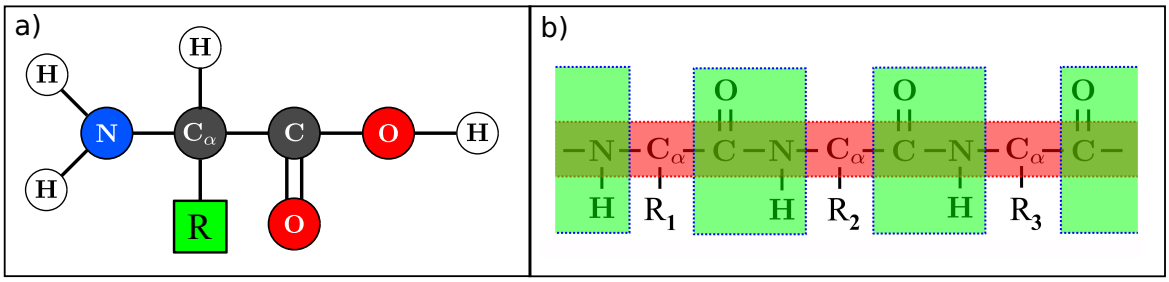


Fig. 2.1 a) An amino acid structure usually contains 3 groups attached to the same C_α atom: an amine group ($-NH_2$), a carboxyl group ($-COOH$), and a side chain **R**. b) A polypeptide chain: the backbone is highlighted by the horizontal red band and the peptide bonds by the green boxes.

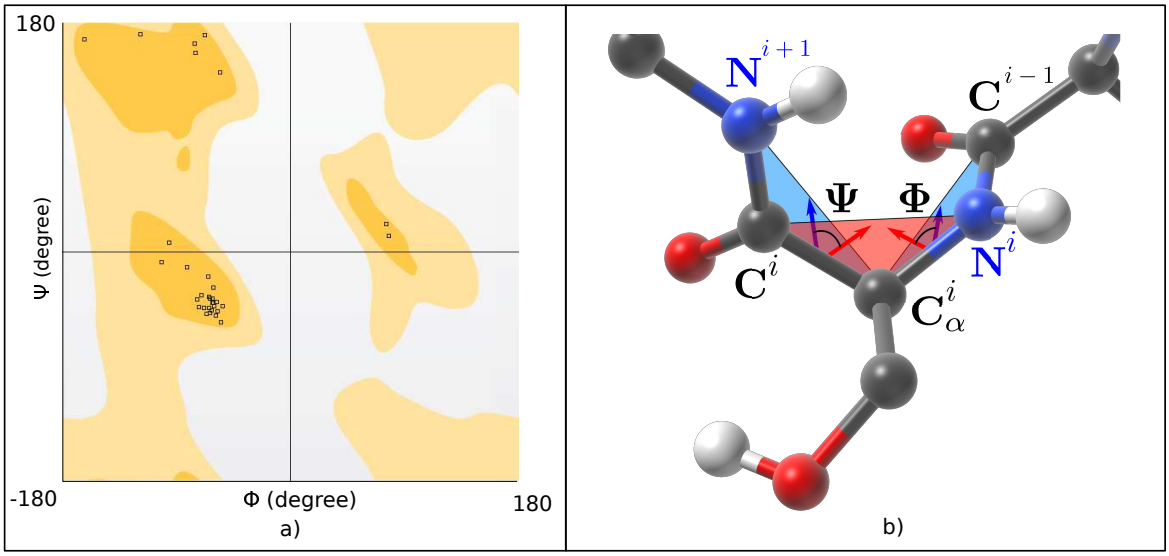


Fig. 2.2 a) Ramachandran plot for a small protein (PDB entry 1YRF). Each dot corresponds to the (Φ,Ψ) pair of an amino acid in the protein. Favorable, less favorable, forbidden regions are shaded in dark yellow, light yellow and grey, respectively. b) Φ angle (about $N-C_\alpha$ bond) and Ψ angle (about $C_\alpha-C$ bond) for an amino acid of 1YRF.

Protein structures are described at four different levels. The primary structure refers to the linear sequence of amino acids of a polypeptide. The secondary structure refers to the alpha-helix where a backbone is folded in spiral shapes, or the beta-sheet where the backbone is arranged in parallel or anti-parallel strands. These shapes are kept stable by the hydrogen bonds among the amino acids in the protein. The tertiary structure refers to the shape where the alpha-helices and beta-sheets are held together by intramolecular forces. The quaternary structure is formed by an assembly of several polypeptide chains interacting with one another. These structures can be conveniently visualized by cartoon (or ribbon) representations as shown in Figure 2.3.

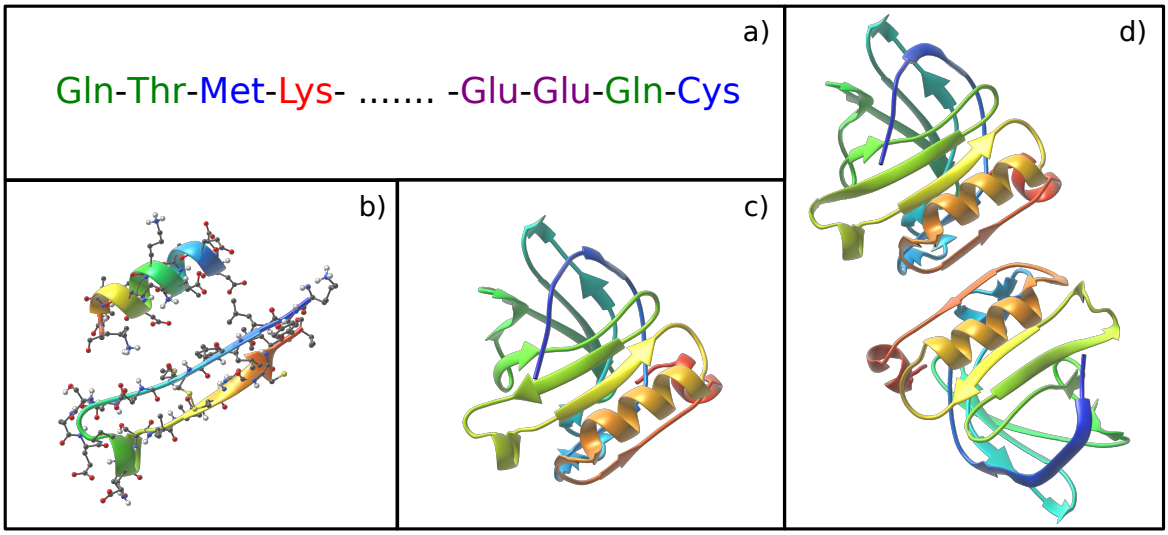


Fig. 2.3 Protein structures (PDB entry 1BEB) visualized in the SAMSON platform: a) A polypeptide containing a chain of amino acids. Each amino acid is shown by its 3-letter code. b) Secondary structure including alpha-helix and beta-sheet shapes. c) Tertiary structure. d) Quaternary structure as an assembly of two polypeptides.

In favorable conditions, polypeptides spontaneously fold in stable shapes (called native structures or native conformations) ready to perform certain biophysical functions [53]. These structures, however, can be altered by the surrounding environment (pH, temperature, etc.) and conformational changes in proteins can lead to changes in their biophysical functions. In particular, misfolded proteins have been found linked to certain neurodegenerative diseases [173]. Hence, the insight into how proteins change from one conformation to another is an important subject of research.

The changes in protein structures can also be caused by interactions with other molecules or ligands. Ligands are substances which form complex with other molecules (also called receptors) to perform particular tasks. They can be small molecules such as ions or even big molecules such as proteins (for e.g. hormones or enzymes). Upon binding, a ligand induces

structural changes in the receptor, and hence in its biophysical function. Several parameters for assessing the biological activity of a ligand are the binding affinity, the residence time, the binding potency and the binding/unbinding rates. Thus, the study of protein-ligand interactions is an important topic because it can provide the prediction of these parameters and insights into these interactions. Figure 2.4 shows an example of a protein-ligand complex.

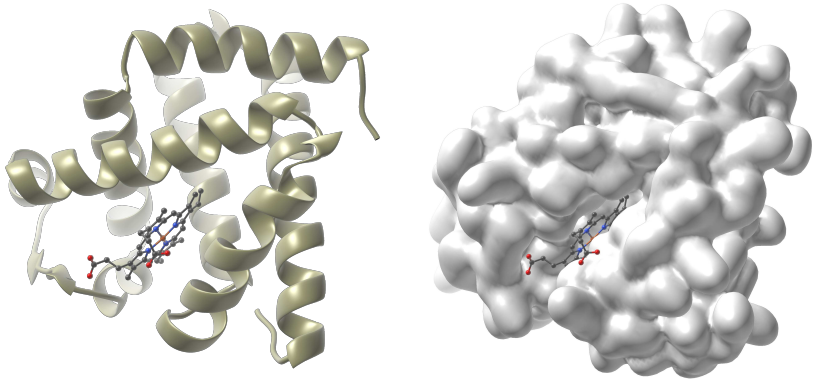


Fig. 2.4 Structure of oxymyoglobin (PDB entry 1MBO) visualized in the SAMSON platform. The ligand (heme group) is represented by sticks and balls while the myoglobin molecule is represented by ribbons (left) and Gaussian spheres (right). Myoglobin stores oxygen before passing it to mitochondria.

2.2 Models of Representation

This section first introduces the models commonly used in computer simulations for describing the kinematics of biomolecules and their interactions. Then, it presents the notion of Potential Energy Landscape, a convenient mathematical framework for studying conformational changes of molecular systems.

2.2.1 Molecular Models and Degrees of Freedom

Molecular models are important tools for understanding, explaining and testing hypothesis about molecular systems. In this thesis study, we employ the popular model which uses point charges to represent atoms (nuclei and electrons) and sticks to represent covalent bonds.

In 3D Cartesian coordinates, each atom position is defined by its x, y, and z coordinates; and hence, a molecule composed of N atoms has $3N$ degrees of freedom (DoF). If the system is centered and aligned on some reference axes, the number of DoF can be reduced to $3N-6$. This number of DoF can be reduced further if the molecules in a system are known or assumed as rigid bodies. In this case, only 6 DoF (rotation and translation) are necessary

to define each molecule in space and an M-molecule system has $6 \times M$ DoF or $6 \times (M-1)$ DoF after centering and alignment.

For proteins, another popular representation is the internal coordinate system which describes a molecule in bond lengths, bond angles and dihedral angles. This representation automatically removes the rotational and translational DoF. Many studies consider only dihedral angles by assuming rigid bond lengths and angles, and hence, significantly reduce the number of DoF. This assumption, as a result, reduces the memory and computational cost in problems involving large systems. The model has been successfully used for many problems such as loop closure (the problem to model loop regions in protein structures) [35, 5, 213, 186], protein folding [12, 220], molecular-pathway search [116, 115, 164], etc. However, a drawback of this simplification is that fixed bond lengths and bond angles can be too restrictive.

Other techniques for model simplification consider only several selected variables such as the center of mass, important bond lengths and/or angles, contacts, etc. In that case, it is assumed that the dynamics of the system spans the subspace made by these collective variables, also called order parameters, while neglecting unimportant DoF. These methods, however, risk losing important information, and hence, require user expertise. Examples of these approaches can be found in [128, 110]. Some other techniques for model simplification are discussed in Section 2.3.3.

2.2.2 Interaction Models

Quantum mechanics uses a complex wave function $\mathbf{\Psi}(\mathbf{x},\mathbf{t})$ to describe a system at a given time and position. The system energy can be obtained by applying the Hamiltonian operator $\hat{\mathbf{H}}$ on the wave function, defined as,

$$\hat{\mathbf{H}} = \hat{\mathbf{T}} + \hat{\mathbf{V}} \tag{2.1}$$

i.e. the sum of the kinetic $\hat{\mathbf{T}}$ and potential $\hat{\mathbf{V}}$ operators.

In molecular systems, the energy is contributed by the interactions among the electrons and nuclei. Thanks to Born-Oppenheimer approximation, the wave function can be separated into the electronic and nuclear parts,

$$\mathbf{\Psi}_{total} = \mathbf{\Psi}_{electronic} \times \mathbf{\Psi}_{nuclear} \tag{2.2}$$

This leads to two fundamental studies: electronic structure and molecular mechanics. The study of electronic structure examines the electron distribution in space with the nuclear

positions fixed while the study of molecular mechanics examines the nuclear positions with the electrons assumed at their optimal distributions. In this manuscript, we are only concerned with the structural changes or the nuclear positions, and hence, the study of molecular mechanics.

Molecular mechanics uses classical mechanics to describe molecular interactions. Each atom (electron and nuclei) is represented by a point charge (particle) in space. The potential energy of a system is contributed by the interactions among its particles. Most models decompose the potential energy into two parts: bonded and non-bonded parts [150], i.e.

$$\begin{aligned} E &= E_{bonded} + E_{non-bonded} \\ &= (E_{bond\,length} + E_{bond\,angle} + E_{torsional\,angle}) + (E_{van\,der\,waals} + E_{electric\,charges}) \end{aligned}$$

The bonded energy is computed from the spring model dependent on bond lengths, bond angles and dihedral angles while the non-bonded energy is computed from the van der Waals and Colombic models dependent on distances among atoms.

Potential forces can then be obtained by computing the gradient of the potential energy,

$$\mathbf{F} = -\nabla E \tag{2.3}$$

Based on this formulation, there exist many force fields to compute potential energy of molecular systems tailored for specific molecular types [160] such as GROMOS [206], AMBER [185], CHARMM [161], OPLS [121], etc. These force fields are simple but require extensive parametrization from experimental data, and hence, are called empirical force fields. They may be used in the following environments: *vacuum*, *implicit solvent* (parameters are added/adjusted to take into account the effect of the solvent), and *explicit solvent* (actual solvent molecules such as water molecules and ions are present in the system). Their parameters and computations are available in software packages such as GROMOS [206], AMBER [38], CHARMM [31], GROMACS [233, 1], etc. These packages assist the researchers in evaluating energy of molecular systems and developing more advanced tools.

2.2.3 Potential Energy Landscape

The notion of Potential Energy Landscape (PEL) can be defined as the mapping from all states of a given molecular system (i.e. its DoF) to their corresponding energy values [237]. PEL is useful for explaining, understanding, and analyzing the important states (or conformations) and reaction pathways among these states.

The important states of a system usually correspond to the minima and first-order saddles on a PEL. On a PEL, conformations at the minima correspond to stable states of molecular systems while those at first-order saddles, where the energy is minimum in all directions except one, are transition states. In protein folding problems, stable states on an energy landscape correspond to the unfolded and folded states and in protein-ligand problems, the binding and unbinding states. Transition states are useful for the calculation of energy barriers and rate constants [49]. Note that the stability of molecular systems also depends on its entropy, which is only taken into account when considering a Free Energy Landscape (FEL). However, for simplicity's sake, the potential energy is assumed to be predominant in this dissertation.

A transition path, or Minimum-Energy Path (MEP), between two stable states is defined as "the path of least resistance" on the PEL connecting two valleys [188]. This path, therefore, passes one or several first-order saddle points lying between two minima on a PEL [188, 98, 214]. Figure 2.5 shows an example of a 2D PEL, with some important states.

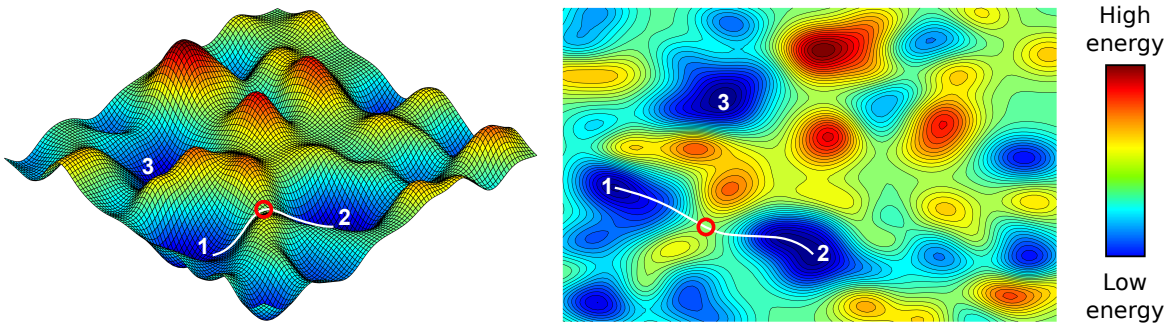


Fig. 2.5 A two-dimensional PEL: surface plot (left) and contour plot (right) of the same PEL. Stable conformations are found at local minima such as those at number 1,2 and 3. The MEP between 1 and 2 is shown as a white line where the red circle indicates the location of a first-order saddle point.

The PEL can already reveal important trends and insights for many systems. Moreover, FELs can be inferred from the investigation of PEL [238]. Hence, the PEL is widely used for many structural-biology problems [237]. In this thesis, to demonstrate the capability of our proposed methods, we employ the PEL for the energy-evaluation purpose because the computation of potential energy is simple and readily available from many software packages such as AMBER, CHARMM, and GROMACS. However, the readers should bear in mind that the proposed methods can be applied to any type of energy landscapes, should their evaluation methods be available.

Let us denote $E(\mathbf{x})$ the potential energy of a state $\mathbf{x}$ in a high-dimensional conformational space. $E(\mathbf{x}+\mathbf{\Delta x})$ can be expressed by using Taylor series to the second order as:

$$E(\mathbf{x}+\mathbf{\Delta x}) \approx E(\mathbf{x}) + \mathbf{g}(\mathbf{x})^T \mathbf{\Delta x} + \frac{1}{2}(\mathbf{\Delta x})^T \mathbf{H}(\mathbf{x})(\mathbf{\Delta x}) \tag{2.4}$$

where $\mathbf{g}(\mathbf{x}) = \nabla E(\mathbf{x})$ is the energy gradient and $\mathbf{H}(\mathbf{x})$ the Hessian matrix.

Local minima and first-order saddle points are stationary points where the gradient $\mathbf{g}(\mathbf{x})$ vanishes, i.e.

$$\mathbf{g}(\mathbf{x}) = \nabla E(\mathbf{x}) = \mathbf{0} \tag{2.5}$$

if $\mathbf{x}$ is a stationary point.

At the minima, the Hessian matrix is positive definite, i.e. all of its eigenvalues are positive. In contrast, at first-order saddle points, all of its eigenvalues are positive except one which is negative. The eigenvector corresponding to the negative eigenvalue gives a hint to trace the MEP. Following this vector in the forward and reverse directions from the saddle point leads to corresponding end states (local minima) of the MEP.

In practice, the PEL of a biomolecular system is typically high-dimensional and much more complex than the one shown in Figure 2.5. The exploration of such a PEL becomes a great challenge because the number of minima can grow exponentially with the size of a system [69, 103, 224]. In that case, the goal is not to find all the minima, but only the ones with lowest-energy, because they are the most stable.

The next section presents the current state-of-the-art methods for exploring molecular pathways, which also includes the methods for finding local minima and first-order saddle points.

2.3 Molecular Pathway Exploration Methods

2.3.1 Minimum-Energy Path search methods

As mentioned above, a MEP connects two stable states and passes through one or more transition states. Therefore, given a PEL, one must first search for the stable states, i.e. local minima. Afterwards, the problem of finding a MEP comes under two categories: single-ended search or double-ended search. Single-ended search methods find pathways with only one known local minimum state, while double-ended search methods find pathways between two known local minima. The sections below present the state-of-the-art methods for local-minimum, single-ended and double-ended searches.

Local-Minimum Search

Since the PELs of biomolecular systems are complex and high-dimensional, most popular methods for searching local minima are iterative methods. Starting from an initial state $\mathbf{x_0}$, these methods formulate a new step $\mathbf{\Delta x}$ to obtain a new state $\mathbf{x_{k+1}}$ from the last state $\mathbf{x_k}$, i.e. $\mathbf{x_{k+1}} = \mathbf{x_k} + \mathbf{\Delta x}$. The methods usually come under either first-order or second-order categories.

First-order methods find the new step $\mathbf{\Delta x} = \mathbf{v_k}\Delta s$ by determining the stepping direction $\mathbf{v_k}$ and step size Δs. Popular first-order methods include the Steepest Descent [244], the damped dynamics 'quick-min' [120], the Conjugate Gradient [80, 184] and more recently, the Fast Inertia Relaxation Engine (FIRE) [24] method. In the Steepest Descent method, the current potential force $\mathbf{F}(\mathbf{x_k})$ is used as the stepping direction $\mathbf{v_k}$ because the potential energy tends to decrease along this direction. The 'quick-min' method also uses the same direction but with an adaptive mechanism to speed up the search progress, whereas Conjugate Gradient methods use a more advanced choice of stepping direction based on the current force, the last force and the last stepping direction. As we will see later in this thesis, we mostly use the FIRE method for optimizing systems toward their closest local minima. FIRE updates the next stepping direction based on the last one and the current force, i.e. $\mathbf{v_k} = (1-\alpha)\mathbf{v_{k-1}} + \alpha\|\mathbf{v_{k-1}}\|\frac{\mathbf{F_k}}{\|\mathbf{F_k}\|}$. Thanks to the adaptive mechanism to adjust α, it has been shown to be robust and efficient compared to most common methods [24].

Second-order methods are based on the Taylor-series approximation of the new energy gradient to the first order, $\mathbf{g_{k+1}} \approx \mathbf{g_k} + \mathbf{H_k}\mathbf{\Delta x}$. If $\mathbf{x_{k+1}}$ is a stationary point, $\mathbf{g_{k+1}} = \mathbf{0}$ and hence,

$$\mathbf{g_k} = -\mathbf{H_k}\mathbf{\Delta x} \tag{2.6}$$

Therefore, $\mathbf{\Delta x}$ can be determined by solving the last equation. These methods, also called Newton-Ralphson methods, must ensure that $\mathbf{H_k}$ remains positive-definite during the search. Most second-order methods are, in fact, quasi-Newton methods because the positive definiteness of the Hessian matrix is maintained by an iterative update procedure such as the Symmetric Rank One [85, 62, 172, 199] or Broyden-Fletcher-Goldfarb-Shanno (BFGS) [33, 79, 88, 210] methods. In addition, to avoid solving Equation 2.6, the BFGS formulas to update the inverse $\mathbf{B_k}$ of the Hessian matrix $\mathbf{H_k}$ are also available by the application of the Sherman-Morrison formula [215].

Second-order methods tend to encounter memory issue due to the storage of the Hessian matrix in high-dimensional problems. This issue can be alleviated by using the limited-memory BFGS (L-BFGS) method [176] where the Hessian matrix is stored in form of a limited number of vectors.

In general, adaptive methods, where the parameters are adjusted during the search, and second-order methods have a better convergence rate than non-adaptive and first-order ones, respectively. However, second-order methods are more computation- and memory- intensive due to the use of Hessian matrices, and hence, less applicable for high-dimensional problems as usually found in structural biology. In this dissertation, we use the FIRE method for local optimization because it is an efficient first-order adaptive method, which is hence, suitable for high-dimensional systems.

Single-ended Search

In single-ended methods, a search is directed uphill from a known local minimum on the PEL toward a saddle point. Finding first-order saddle points is much more difficult than finding local minima because they have a stringent property, i.e. at first-order saddle points, the energy is maximum in only one direction. Hence, the Hessian matrix at the first-order saddle points must have only one negative eigenvalue and the others should be positive.

Methods for locating first-order saddle points include quasi-Newton ones and others. Because the eigenvalues of the Hessian matrix must not be all positive, only certain Hessian update strategies are suitable such as Powell-symmetric-Broyden update [62], the combination of Symmetric Rank One with Powell-symmetric-Broyden or BFGS [25, 75, 26]. The time step and direction must also be carefully controlled in these methods [199].

Other popular methods include the eigenvector following [40, 218], reduced gradient following [189] and dimer [99] methods.

After the saddle point is found, a local minimizer can be used to relax the system from the saddle toward two local minima by taking opposite directions of the eigenvector corresponding to the only negative eigenvalue of the Hessian matrix. The activation-relaxation technique [163, 159, 170] is known for performing the whole procedure.

Double-ended Search

Double-ended methods adjust an initial path connecting two given local minima until this path converges to a MEP. These methods are called chain-of-state methods because the path is composed of a series of states called images of the same structure, evolving from the initial to the final conformations. The quality of the paths produced from these methods are shown to depend on the initial path [214].

A simple way to produce an initial path is by linear interpolation in the Cartesian space between the initial and final states. More sophisticated interpolation methods have also been

developed such as the linear interpolation in internal coordinates or the Linear and Quadratic Synchronous Transit methods [93].

Popular double-ended methods include the Nudged Elastic Band (NEB) [120] and String [240] methods. In the Nudged Elastic Band method (NEB), two types of forces are applied on each image. The potential force is used to drive each image to the lowest energy on the plane perpendicular to the path (a property of MEPs) while the spring force is applied on any pair of consecutive images to prevent the images from clustering together or maintain the distances among them. The String method also uses the potential force in the same manner, however, the distribution of the images along the path is controlled by a user-defined parameterization [240]. Although the MEPs can be obtained with these methods, the transition states can be missed. Therefore, several extensions of the NEB and String methods have been proposed to address this problem [100, 101, 249, 194, 241, 182].

2.3.2 Simulation-based methods

Classical methods to simulate molecular systems are Monte Carlo (MC) [166, 208, 95, 135, 157] and Molecular Dynamics (MD) [9, 4, 58, 198, 47] simulations. The MC simulations, based on statistical mechanics, depends on random sampling of states to simulate a system evolution. In contrast, the MD simulations solve numerically the Newton equation of motion in time domain to obtain a system trajectory. These methods are widely used and continue to be improved and extended because they respect the Boltzmann distribution of the states, which allows the prediction of macroscopic properties such as reaction rate constants.

However, it usually takes a lot of time before the trajectories from these simulations encounter interesting events because the pathways have to pass the transition states which are rare, short-lived and associated with high-energy barriers [60]. Hence, many recent methods have been developed to enhance these simulations through manipulating temperature, potential energy or force.

Among the methods which manipulate the temperature are temperature-accelerated MD [221, 169], parallel tempering [225, 227, 71], simulated annealing [132, 32], etc. By raising the temperature, the temperature-accelerated MD method increases the chance to cross high-energy barriers. The simulated annealing method, which emulates the annealing process in metallurgy, simulates a system at high temperature at first to cross high-energy barriers; then cools it down to arrive at another local minimum. The parallel tempering method simulates different replica of the same system at different temperatures; and then, exchange these replica at a regular time intervals. This method can take advantage of parallel computing by running different-temperature simulations on different threads.

Biased-potential methods such as metadynamics [17], umbrella sampling [231], hyper-dynamics [235, 19, 78], local elevation method [109], basin hopping [239], temperature basin-paving [209] or conformational flooding [89] increase the chance for escaping high-energy barriers and avoid re-visiting the visited local minima.

Common biased-force techniques tend to steer a simulation toward certain directions. Among popular methods, one can find Targeted MD (TMD) [201], Steered MD (SMD) [114], Random Acceleration MD (RAMD) [158], etc. In TMD, a biased force based on the Root Mean Square Distance (RMSD) with respect to the target structure is added. Hence, the target structure must be known beforehand. With SMD, an external force is applied on a group of atoms along a desired direction. During simulation, this force decreases when the controlled atoms move in the prescribed direction, and increases otherwise. Hence, SMD is mainly used for analyzing pathways following the directions known in advance. The RAMD method also applies an external force on a group of atoms. However, this force is activated for a certain number of time steps and then changed in direction if the atom group has not moved significantly. Since RAMD is not biased toward a specific direction, it can be used for exploring all possible pathways, although the success of the method may be sensitive to the parameter choice [37]. During the past few decades, these methods have contributed significantly to the study of ligand-protein interaction pathways [136, 118, 181, 3, 36, 119, 56, 68].

2.3.3 Methods based on simplified models

One of the greatest challenges in structural-biology problems is the high dimensionality of the conformational spaces representing the molecular systems. Dimension reduction is hence very important for simulating these systems because it greatly reduces the number of degrees of freedom, and hence, the computational cost. There are a great variety of approaches for simplifying models in structural-biology problems. However, only two approaches are discussed here because they are popular and more related to our context. The first one includes morphing techniques which are geometry-based and the second one is the Elastic Network Model (ENM). For other approaches, the readers are referred to [128, 110, 232].

Geometric approaches

As shown in Section 2.1, protein structures possess certain flexibility. Therefore, there have been many attempts to deduce protein motions by geometrical means. The first approach to morph a molecular structure between two given conformations is perhaps the linear interpolation in Cartesian coordinate system. However, this method is prone to unrealistic

bond lengths and angles. Another geometric method is the linear interpolation in internal coordinate system as implemented in LSQMAN [133, 134]. This method may also produce distorion in the path conformations due to the accumulation of rigid transformations [27].

Hence, sophisticated methods have been designed to give more realistic pathways. The Linear Synchronous Transit (LST) [93] method generates a path such that each atom-pair distance in an intermediate structure is the interpolated value between those in the initial and target structures. However, the method is more suitable for small-sized systems because it relies on an iterative solver which is computationally expensive. The Morph-Pro method corrects the intermediate conformations generated by linear interpolation such that the distances between any two consecutive C_α atoms lie within a particular range [39]. The Climber method [242] interpolates the adjacent C_α -atom distances while imposing harmonic restraints among them. Gerstein et al., with the Molmovdb server [72] for protein morphing, locates representative hinges and performs restrained interpolation using an adiabatic mapping technique [137]. The generated paths are reasonable in the sense that large structural distortions are avoided. Their visual representations can help to better understand protein motions. With the FRODA method [243], rigidity analysis, geometric constraints and steric constraints are applied to simulate transition paths. A recent geometric targeting method inspired by FRODA but using a different mathematical formalism has also been proposed for finding protein pathways [76].

Elastic network models

The Elastic Network Models (ENM) [14, 94, 203, 230], commonly used with the Normal Mode Analysis (NMA) [54], uses the mass-and-spring model to find the normal modes representing the collective motions of a system. Low-frequency motions, which typically participate in large conformational rearrangements, can then be extracted [46]. The application of ENM for proteins usually represents each amino acid as a point mass centered at C_α atoms and assigns springs among the point masses. Therefore, a system of thousands of atoms can be represented by only hundreds of point masses. The ENM has been successfully used for analyzing protein motions [149, 152], predicting protein motions [187], generating pathways [130], or combined with motion planning methods for exploring protein motions [131] and conformational transition pathways [7].

2.3.4 Motion Planning methods

Motion planning methods typically aim to find a path connecting two given states while satisfying a set of constraints [141, 45, 143]. Research in this field was originally developed for robotics problems, and later on, extended to other domains including structural biology.

While there exists a multitude of motion planning methods, many derived either from the seminal work on the Probabilistic Roadmap (PRM) method [125, 124], or the Rapidly-exploring Random Tree (RRT) method proposed by Lavalle et al. [142, 139, 144]. These approaches rely on stochastic processes for sampling random states in a given space and the accepted states are recorded in a special graph. Therefore, a graph is constructed where the nodes correspond to the accepted states and the edges the possible motions between these states. In the case of PRM, a multiconnected graph is constructed (Figure 2.6) and then used for solving different queries whereas in the RRT methods, one or several trees are built directly from the query nodes (see Figure 2.7).

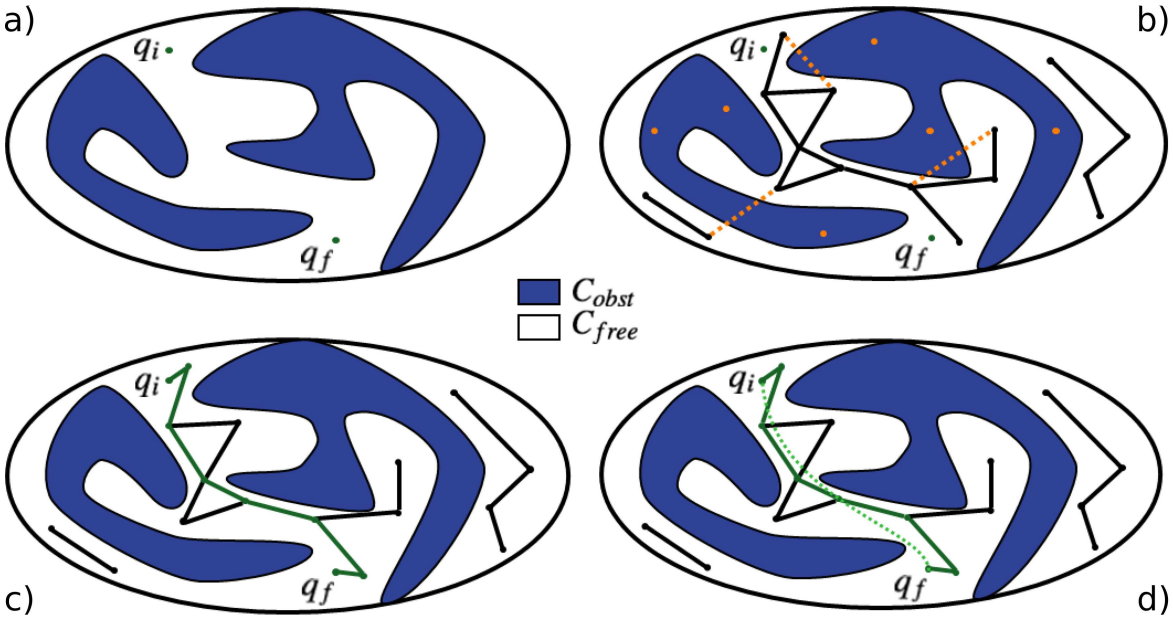


Fig. 2.6 Example of a Probabilistic roadmap (PRM) construction. a) A path is searched in free (white) regions C_{free} to connect the queries q_i and q_f, while avoiding obstacle (blue) regions C_{obst}. b) Random configurations are sampled, building a graph made of nodes and edges corresponding to valid states (black dots) and valid motions (black lines), respectively. Orange dots and orange dotted lines correspond to invalid states and invalid motions, respectively. c) The query conformations are finally connected and a solution path (dark green) is extracted. d) The solution path is optimized (dotted light green).

Motion planning can be applied to structural-biology problems by considering a molecular system as a *robot* and the interactions among atoms as constraints. Then, the solutions

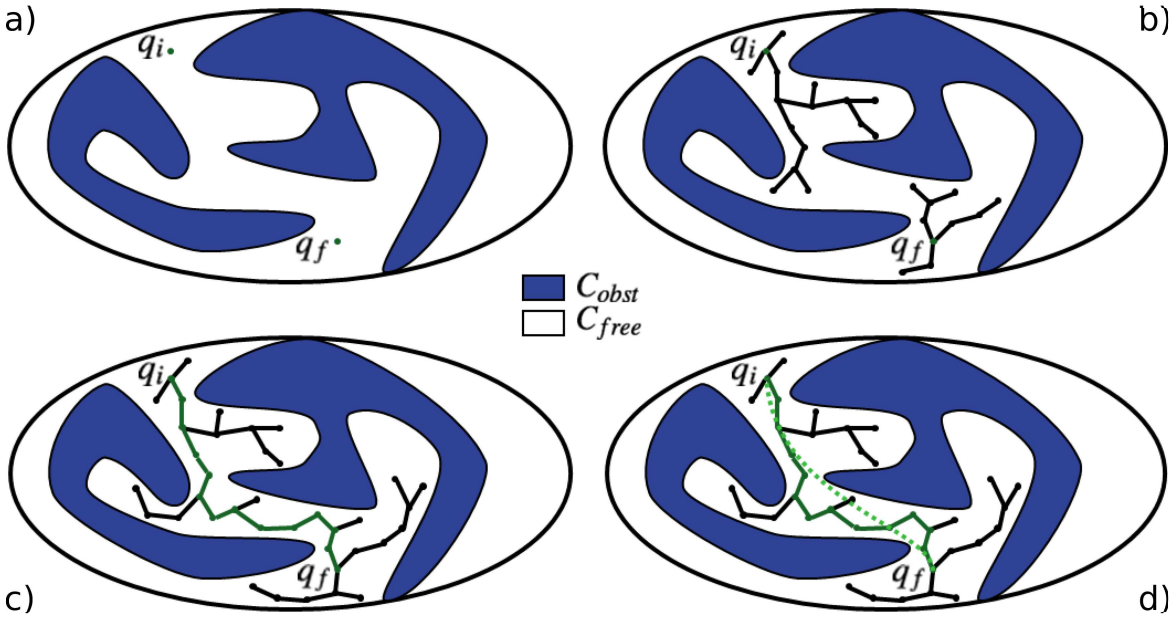


Fig. 2.7 Example of a Rapidly-exploring Random Tree (RRT) construction. a) A path is searched in free (white) regions C_{free} to connect the queries q_i and q_f while avoiding obstacle (blue) regions C_{obst}. b) One or two trees are built from the query nodes, by extending branches toward randomly sample states. c) Once the trees get connected, a solution path (dark green) is extracted. d) The solution path is optimized (dotted light green).

correspond to biologically feasible molecular motions. For the past few years, an increasing number of motion planning strategies have been proposed in this field, with a special focus on proteins and their interactions [6, 86].

Among the PRM-based methods [167], one finds roadmaps built from the samples based on internal coordinates or the rigidity theory [11, 229]. Such approaches have also been extended to stochastic roadmaps (SMR) where the most probable paths are estimated from the transition probability based on local energy variations [13, 43]. Another type of approach builds these roadmaps from the outputs of MC or MD simulations. [219].

Several RRT variants have also been proposed for structural-biology problems. The Transition-based Rapidly-exploring Random Tree (T-RRT) method [116, 115] finds low-energy pathways by combining the RRT exploration with a transition test similar to the one used in Monte Carlo simulations to accept new states. This approach which led to several extensions [67, 65, 64, 186] is also used in one of the proposed methods of this thesis (more details in Part III). Recently, T-RRT has also been combined with the basin-hopping method [239] to improve the exploration of energy landscapes [196]. Inverse kinematics techniques have been incorporated in RRT planners to solve the flexibility problem in protein loops [52, 186]. For ligand unbinding problems, one finds the ML-RRT method which relies on

a mechanistic behavior to find disassembly paths [50, 51]. The method is incorporated in the MoMA-LigPath webserver [63], which has been used in serveral studies for finding ligand unbinding pathways from proteins [193, 117]. The ML-RRT method has also found application in exploring large motions of proteins [16]. In the PathRover method [192], constrains are introduced into RRT for exploring low-energy clash-free motions of proteins. Some methods combine RRT with NMA [54] for exploring large-amplitude motions of proteins [131, 7]. Recently, tree-based methods were proposed [87, 177], where the sampling process was improved by exploring on low-dimensional but highly-probable projected spaces.

Chapter 3

Algorithmic framework

This chapter focuses on the algorithmic structures behind the As-Rigid-As-Possible (ARAP) and the Rapidly-exploring Random Tree (RRT) methods, which are the core of the proposed methods in this dissertation. The final section of the chapter gives an overview of the platform where all the proposed methods in this dissertation were developed.

3.1 The ARAP methodology

3.1.1 ARAP in computer graphics

Nowadays, powerful tools are available in computer graphics for manipulating and animating objects, falling into two main categories: physics-based and geometry-based.

Physics-based methods apply material properties and/or physical laws for morphing[1] objects. For example, surfaces can be seen as thin vicous materials [97]; vibration modes can be used to interpolate shapes [107]; strain fields [247] and elastic deformations [41] can be used for interpolation; and others [29, 245].

Geometry-based methods deform objects from geometric information such as vertices, edges, faces, etc. Two common approaches are the skeleton-based [148, 246] and cage-based [81, 122] ones, which are used in professional 3D computer graphics software such as Autodesk Maya, 3DStudio and Blender. These methods represent an object by a simpler geometry such as an articulated body (skeleton-based) or a simpler mesh (cage-based). Each part in the simplified geometry has a correspondence in the original object, and hence, a designer only needs to manipulate the simplified geometry to obtain deformations in the

[1] Morphing in computer graphics is creating a gradual change from one image to another using computer algorithms.

original object. The drawback of these methods is, hence, the requirement of an efficient algorithm for generating the simplified geometry.

Therefore, other geometry-based methods have emerged which directly manipulate the original geometry of an object such as certain groups of vertices. These methods deform the objects with an objective to preserve certain characteristics of the original structures. The As-Rigid-As-Possible methods on which rely the proposed methods of this dissertation belong to this category. Their objective is to to preserve the local rigidity of the original structures. Other methods include the As-Similar-As-Possible method [129, 251] which preserves the tetrahedral meshes composing an object, or the Example-Driven method [84] which minimizes the deformations of edges, angles and dihedral angles.

Another geometry-based approach is manipulating a set of geometric properties which describe the local rigidity of an object. A deformed shape is obtained by converting the interpolated values of these properties into the object vertex positions. The methods following this principle include the Laplacian surface editing method [223], the linear rotation-invariant coordinate [153], the pyramid coordinate [212] and others [178]. The readers can refer to [30] for a larger survey of shape deformation techniques.

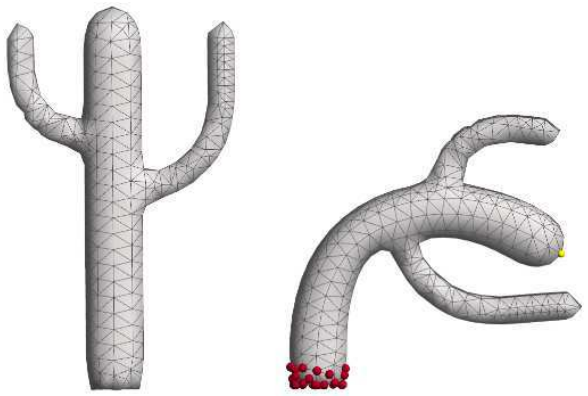


Fig. 3.1 The ARAP modeling method (ARAPm) applied on a cactus mesh (captured from [222]). One control point (in yellow) at the top of the cactus is displaced while other control points (red) at the base are fixed (in red). ARAPm allows to interactively deform the initial mesh on the left, while maintaining the structure as-rigid-as-possible, resulting in the mesh on the right.

The proposed methods in this dissertation rely on the ARAP paradigm that allows to generate new shapes while preserving the local rigidity of some reference shape. The ARAP idea was initially introduced by Alexa et al. for 2D objects [10]. Since then, it has received a lot of attention and apart from its 2D extensions [44, 111, 92, 18], it has been applied on 3D objects with two types of applications. The first application is the interactive manipulation

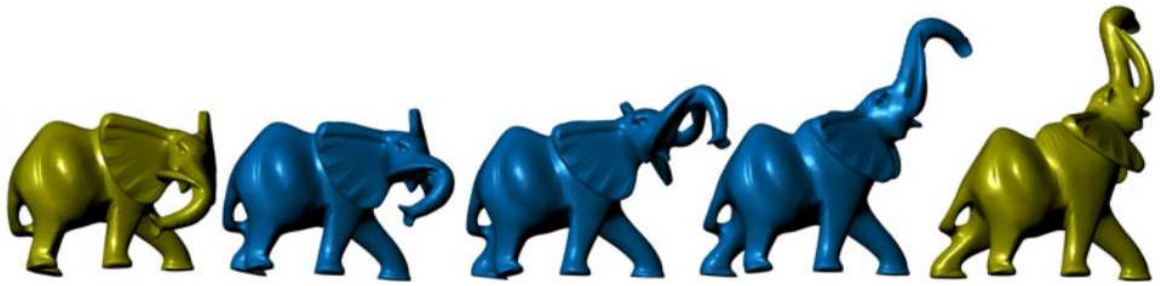


Fig. 3.2 The ARAP interpolation method (ARAPi) applied on an elephant mesh (captured from [147]). The initial and final meshes in yellow are given as input. ARAPi allows to interpolate from the initial mesh to the final one, while maintaining the structure as-rigid-as-possible, resulting in the intermediate poses in blue.

of objects [222, 55, 28, 253], which will be referred as ARAP modeling (**ARAPm**) in this manuscript. In this application, a user only needs to displace a few points on a mesh to obtain an as-rigid-as-possible shape (see Figure 3.1). The other application is the generation of an interpolated path between an initial and a final shape [41, 156, 147], which will be referred as ARAP interpolation (**ARAPi**) in this manuscript. The method distributes continuously the deformations of the shapes along a path, while maintaining the local rigidity as much as possible (see Figure 3.2). Some remarkable improvements of the ARAP methods include the rotation smoothing technique [147], the parallel implementation [253], or the extension of the structural topology [42].

The following subsections present in detail the ARAP principle and its applications for modeling and interpolation, which are used in our proposed methods.

3.1.2 ARAP optimization problem

Various ARAP-based methods address at some stage the following problem: given an initial and a target shape, find a resulting shape which preserves the rigidity of the initial shape as much as possible, while attempting to align with the target shape. The difficulty comes from the two conflicting criteria: the rigidity preservation of the initial shape and the alignment with the target shape. To resolve this problem, the ARAP principle proposes to proceed in three steps (illustrated in Figure 3.3):

1. Decomposition: the structure is decomposed into smaller pieces called ARAP sets.
2. Alignment: the pieces of the initial shape are rotated to align with their counterparts in the target shape.
3. Blending: the rotated pieces are blended to obtain the resulting shape.

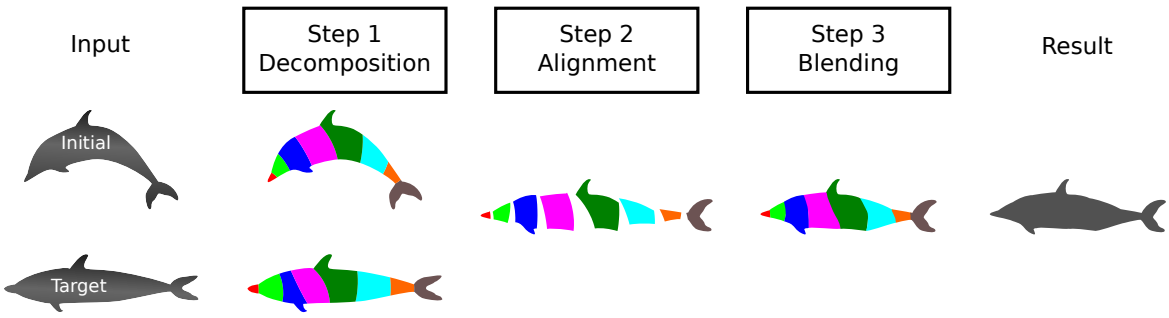


Fig. 3.3 The three steps behind the ARAP principle. The input includes the initial and target shapes of a dolphin. 1) Decomposition step: the input shapes are decomposed into corresponding pieces which share the same color. 2) Alignment step: each piece in the initial shape is rotated to best align with its counterpart in the target shape. 3) Blending step: the rotated pieces are blended to give the output. With such a mechanism, the output tends to preserve the initial shape of each piece, and hence, the local rigidity, while getting close to the target shape.

The details of each step and the algorithm for solving the ARAP optimization problem are given below.

Decomposition step - ARAP sets and cells

Let us assume from now that an object is described by a mesh made of vertices and edges while a shape is described by a set of the vertex positions in a 3D space. Then, for a given mesh made of n vertices $v_0,\dots,v_{n-1}$, we denote $\mathbf{p_0},\dots,\mathbf{p_{n-1}} \in \mathbb{R}^3$ and $\mathbf{p'_0},\dots,\mathbf{p'_{n-1}} \in \mathbb{R}^3$ the positions of these vertices in the initial and target shapes, respectively. From the topology of a given mesh, one can also define n ARAP sets $\mathcal{N}_i$ ($\forall i \in [0,n-1]$), each of which consists of a central vertex v_i and all the vertices whose topological distance to this vertex (i.e. the maximum number of edges that must be traveled to reach this vertex) is lower than a given value k. The maximum distance used to define a set is typically $k = 1$, thus including only the one-ring neighbors [2], i.e. the vertices directly connected to the central vertex v_i . Figure 3.4 illustrates the decomposition of a mesh containing 7 vertices into ARAP sets based on the one-ring neighbor topology.

An ARAP cell consists of the positions of all the vertices of an ARAP set. Hence, from the initial and target shapes, one can define n corresponding initial and target cells $\mathcal{C}_i$ and $\mathcal{C}'_i$, respectively.

[2] this one-ring neighbor topology is also called spoke in [147]. This topology is also the only one used in this thesis study.

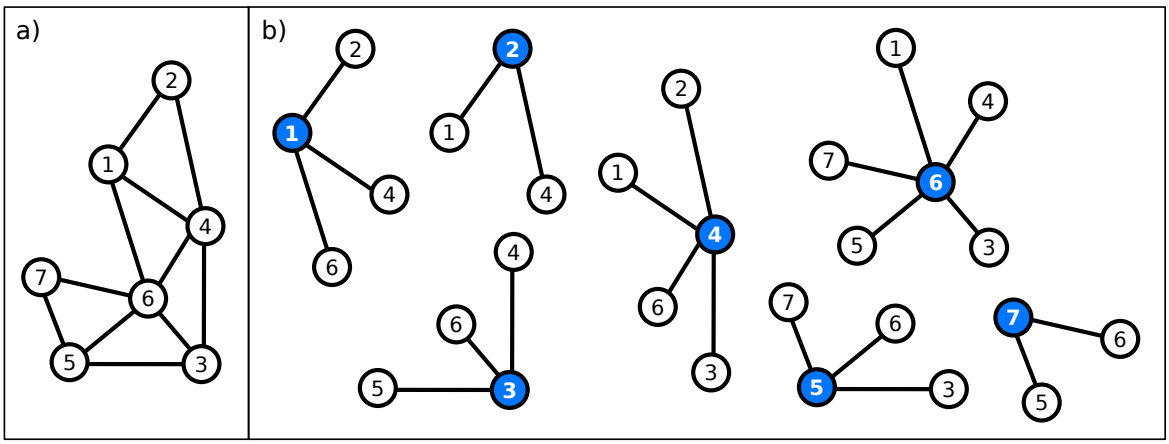


Fig. 3.4 ARAP sets for a simple mesh. Vertices are in circles and edges in solid lines. a) The entire mesh. b) The seven sets built from the one-ring neighbor topology. Blue vertices represent the central vertices.

Alignment step - using rotation

If the cell $\mathcal{C}'_i$ is a rotation of the cell $\mathcal{C}_i$, there exists a rotation $\mathbf{R_i}$ satisfying

$$\mathbf{p'_i} - \mathbf{p'_j} - \mathbf{R_i}(\mathbf{p_i} - \mathbf{p_j}) = \mathbf{0} \qquad ,\forall j \in \mathcal{N}_i \tag{3.1}$$

where $\mathbf{p_i}$ and $\mathbf{p'_i}$ are the central-vertex positions in $\mathcal{C}_i$ and $\mathcal{C}'_i$, whereas $\mathbf{p_j}$ and $\mathbf{p'_j}$ are the neighbor-vertex positions in $\mathcal{C}_i$ and $\mathcal{C}'_i$, respectively.

Therefore, the best rotation to align $\mathcal{C}_i$ with its counterpart $\mathcal{C}'_i$ can be found by minimizing the *cell deformation energy* or *ARAP cell energy* $E(\mathcal{C}_i, \mathcal{C}'_i)$ originally introduced by Sorkine and Alexa [222], and defined as:

$$E(\mathcal{C}_i, \mathcal{C}'_i) = \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\mathbf{p'_i} - \mathbf{p'_j} - \mathbf{R_i}(\mathbf{p_i} - \mathbf{p_j})||^2 \tag{3.2}$$

where ω_{ij} is the weight imposed on the edge connecting v_i and v_j in the set $\mathcal{N}_i$. Hence, it is possible to have different weights to adapt the local rigidity according to the problem. Moreover, one may have $\omega_{ij} \neq \omega_{ji}$ because their corresponding edges come from different sets. In practice, most methods use $\omega_{ij} = 1$ for $\forall i, j$, i.e. uniform edge weight. Figure 3.5 shows an example of a rotation which best aligns two cells after aligning first the central vertices.

The rotation $\mathbf{R_i}$ to minimize Equation 3.2 can be found in the form of a matrix [105] or a quaternion [104]. In our studies, we use a quaternion-based method [155].

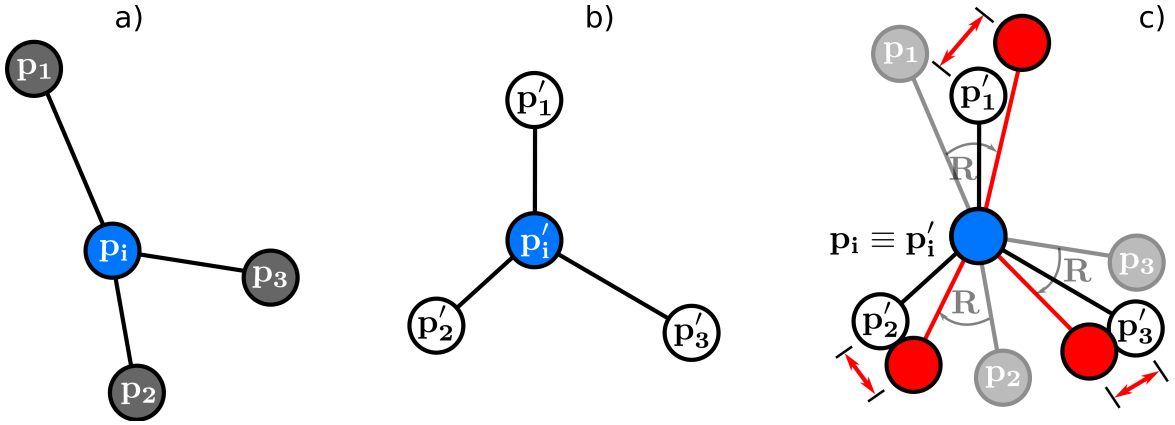


Fig. 3.5 ARAP-cell alignment by rotation: a) The initial cell with $\mathbf{p_i}$ as the central vertex position. b) The target cell with the corresponding $\mathbf{p'_i}$. c) The initial cell is rotated by $\mathbf{R}$ to align with the target cell. This rotation minimizes the sum of the squared distances represented by the red arrows.

Blending step - ARAP energy

After the aligning rotations are found, each initial cell is rotated to align with its corresponding target cell. All the rotated cells are then blended by minimizing the *ARAP energy* defined as:

$$E_{ARAP} = \sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_\mathbf{i} - \widehat{\mathbf{p}}_\mathbf{j} - \mathbf{R_i}(\mathbf{p_i} - \mathbf{p_j})||^2 \tag{3.3}$$

where ω_i is the weight of the ARAP set $\mathcal{N}_i$ and $\omega_i = 1, \forall i$ in the case of uniform cell weight. In the equation, $\mathbf{R_i}(\mathbf{p_i} - \mathbf{p_j})$ are the rotated edges, and hence, the blending step finds a set of vertex positions $\widehat{\mathbf{p}}_\mathbf{i}$ and $\widehat{\mathbf{p}}_\mathbf{j}$ where each cell in the result is as close as possible to the corresponding rotated cell. Finally, note that the ARAP formulation in Equation 3.3 is translation-invariant. Hence, in practice, at least one vertex position needs to be constrained to set the global mesh position.

Algorithm

The solution of the ARAP optimization problem is shown in Algorithm 1. The input contains the vertex positions in the initial and target shapes, the mesh topology (vertices and edges), the set $\mathcal{T}$ containing n_c constrained vertices with their constrained positions.

First, the cells of the initial and target shapes are constructed (line 2-3) and the aligning rotations are computed (line 4). Then, the constrained-vertex positions are set for the solution (line 5-6). Finally, the positions of the rest of the vertices are computed by minimizing Equation 3.3 (line 7).

Algorithm 1: ARAP optimization

Input : Initial and final vertex positions $\mathbf{p_i}$ and $\mathbf{p'_i}$, respectively, $\forall i$.
Vertex connectivity (edges).
Set $\mathcal{T}$ containing n_c constrained vertices and their positions $\overline{\mathbf{p}}_\mathbf{i}$ for $i \in \mathcal{T}$.

Output : ARAP vertex positions $\widehat{\mathbf{p}}_\mathbf{i}$.

1 **for** $i = 0, \ldots, n-1$ **do**
2 $\quad \mathcal{C}_i \leftarrow$ `ComputeCell`$(\mathbf{p_i})$;
3 $\quad \mathcal{C}'_i \leftarrow$ `ComputeCell`$(\mathbf{p'_i})$;
4 $\quad \mathbf{R_i} \leftarrow$ `ComputeRotation`$(\mathcal{C}_i, \mathcal{C}'_i)$;
5 **for** $i \in \mathcal{T}$ **do**
6 $\quad \widehat{\mathbf{p}}_\mathbf{i} \leftarrow \overline{\mathbf{p}}_\mathbf{i}$;
7 $\widehat{\mathbf{p}}_\mathbf{0}, \ldots, \widehat{\mathbf{p}}_\mathbf{n-1} \leftarrow$ `MinimizeEnergy`(E_{ARAP});

The minimization of the ARAP energy leads to solving the linear algebra problem $\mathbf{L}\widehat{\mathbf{p}} = \mathbf{b}$ [222]. If there are n_c constrained vertices, $\mathbf{L}$ is a $(n-n_c) \times (n-n_c)$ matrix. The $(n-n_c) \times 3$ matrix $\widehat{\mathbf{p}}$ is a concatenation of the unknown vertex positions and $\mathbf{b}$ is a matrix of the same size. Appendix A.1 details the construction of these matrices and how they are used to solve this optimization problem. It is important to note that the matrix $\mathbf{L}$ is symmetric and positive-definite. If the one-ring neighbor topology is used, $\mathbf{L}$ becomes a sparse matrix, and hence, the solution can be efficiently computed because the algorithm complexity is almost linear, i.e. approximately $\mathcal{O}(n-n_c)$.

3.1.3 Modeling application (ARAPm)

ARAP modeling is a useful method for 3D graphical design. A designer can obtain flexible shapes by constraining the motion of certain vertices on a mesh. The efficiency of the method allows real-time mesh editing. This mechanism has been proposed to manipulate molecular systems in the SAMSON software platform [113] as illustrated in Figure 3.6. It shows how a large deformation can be produced by displacing a very limited set of atoms, while preserving the local rigidity.

The ARAP modeling method is described in Algorithm 2. The input consists of the initial shape (initial vertex positions $\mathbf{p}_i$), the vertex connectivity (mesh edges), the set $\mathcal{T}$ containing the constrained vertices, the constrained-vertex positions $\bar{\mathbf{p}}_i$ and the number of ARAP optimization steps m. This algorithm only differs from the one for the ARAP optimization problem in that it has no target shape and is solved by iteration. Therefore, after the initial cells are constructed (line 2), the target shape is constructed by considering the vertex positions in the initial shape and the constrained vertex positions (line 3-6). Then, m iterations are performed (line 7) to adjust the target shape. At each iteration, the target cells are updated by the target-vertex positions (line 9), which allows to compute the aligning

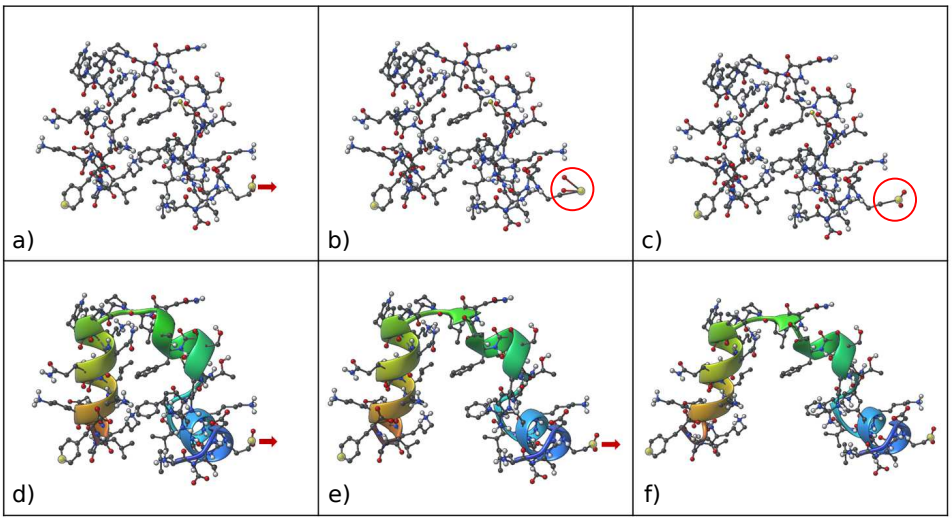


Fig. 3.6 ARAP modeling applied on a molecular structure (PDB id: 1YRF) as simulated in the SAMSON software platform. a) The two atoms in yellow at the bottom left and right are constrained atoms. b) The right constrained atom is displaced and the left one is fixed. The picture shows the resulting shape in the case where the ARAP modeling is not applied. c) Thanks to ARAP modeling, the displacement leads to the entire structure modified while the structural rigidity is preserved as much as possible and the position of the left constrained atom stays fixed. d), e) and f) Large deformations of the system by successive motions of the right constrained atom. Note how the secondary structure is largely preserved.

rotations (line 10). These rotations are in turn used to estimate the new target-vertex positions (line 11). The final target-vertex positions are the output of the algorithm. The more iterations performed, the closer the result is to the initial shape, yet the constrained-vertex positions are controlled. Figure 3.7 shows the effect of the number of iterations for a simple structure with two constrained vertices.

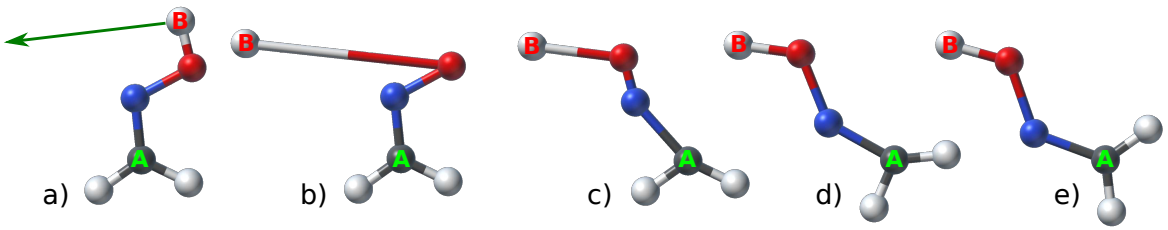


Fig. 3.7 Effect of the number of iterations in ARAPm on a simple structure: a) Initial shape with constrained vertices (A and B). The arrow shows the displacement vector for atom B. b) Atom B is displaced to a new position while atom A is kept fixed. This shape serves as the initial target shape in Algorithm 2. c) d) and e) show the results after the 1st, 5th, and 20th iteration, respectively. As one can see, the more iterations are performed, the closer the result is to the initial shape, yet the constrained-vertex positions are controlled.

Algorithm 2: ARAP modeling

Input : Initial vertex positions $\mathbf{p_i}$.
Vertex connectivity (edges).
Set $\mathcal{T}$ containing n_c constrained vertices and their positions $\overline{\mathbf{p}}_\mathbf{i}$ for $i \in \mathcal{T}$.
Number of solving iterations m.

Output : Final target-vertex positions $\widehat{\mathbf{p}}_\mathbf{i}$.

1 **for** $i = 0, \ldots, n-1$ **do**
2 $\mathcal{C}_i \leftarrow \texttt{ComputeCell}(\mathbf{p_i})$;
3 **if** $i \notin \mathcal{T}$ **then**
4 $\widehat{\mathbf{p}}_\mathbf{i} \leftarrow \mathbf{p_i}$;
5 **else**
6 $\widehat{\mathbf{p}}_\mathbf{i} \leftarrow \overline{\mathbf{p}}_\mathbf{i}$;
7 **for** $iter = 0, \ldots, m-1$ **do**
8 **for** $i = 0, \ldots, n-1$ **do**
9 $\mathcal{C}'_i \leftarrow \texttt{ComputeCell}(\widehat{\mathbf{p}}_\mathbf{i})$;
10 $\mathbf{R_i} \leftarrow \texttt{ComputeRotation}(\mathcal{C}_i, \mathcal{C}'_i)$;
11 $\widehat{\mathbf{p}}_\mathbf{0}, \ldots, \widehat{\mathbf{p}}_\mathbf{n-1} \leftarrow \texttt{MinimizeEnergy}(E_{ARAP})$;
12 **return** ARAP vertex positions $\hat{\mathbf{p}}_\mathbf{i}$;

3.1.4 Interpolation application (ARAPi)

Given an initial and a target shape of a structure, an interpolation method generates a series of intermediate shapes making a smooth transition between the initial and target shapes. Our ARAP interpolation method is an adaptation of the method proposed in [10]. The central idea is that to get an intermediate shape, the initial cells are rotated by a fraction of the aligning rotations and the rotated cells are then blended together. Hence, *ARAPi* requires a method for interpolating rotations that will be detailed later on.

Our implementation for ARAPi is shown in Algorithm 3. The input is similar to the one in the algorithm for the ARAP optimization problem presented in Section 3.1.2 except for an additional parameter $\mathcal{L}$ which is the requested number of intermediate shapes for the path (the initial and target shapes are included).

The algorithm first constructs the initial and target cells, and computes the cell aligning rotations (line 1-4). For each intermediate shape (line 5), an interpolation instance $t \in [0,1]$ is computed (line 6), such that $t = 0$ and $t = 1$ correspond to the initial and target shapes, respectively. Then, the rotation of each cell $\mathbf{R_i}(t)$ is interpolated between the identity transform $\mathbf{I}$ and its aligning rotation $\mathbf{R_i}$ based on t (line 7-8). Next, the constrained-vertex positions are linearly interpolated between their positions in the initial and target shapes (line 9-10). Finally, line 11 finds all the vertex positions by minimizing the ARAP energy for the intermediate shape at t, $E_{ARAP}(t)$, defined as,

$$E_{ARAP}(t) = \sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_\mathbf{i}(t) - \widehat{\mathbf{p}}_\mathbf{j}(t) - \mathbf{R}_\mathbf{i}(t)(\mathbf{p}_\mathbf{i} - \mathbf{p}_\mathbf{j})||^2. \tag{3.4}$$

The minimization of this equation can be shown to lead to solving a linear system $\mathbf{L}\widehat{\mathbf{p}}(t) = \mathbf{b}(t)$ (see Appendix A.2) where the square matrix $\mathbf{L}$ is independent on t. The rotation interpolation method is detailed below.

Algorithm 3: ARAP interpolation

Input : Initial and target vertex positions $\mathbf{p_i}$ and $\mathbf{p'_i}$, respectively, $\forall i$.
Vertex conectivity (edges).
Set $\mathcal{T}$ containing n_c constrained vertices.
Number of shapes on $\mathcal{L}$ on the interpolation path.
Output : A smooth path consisting of $\mathcal{L}$ shapes.

for $i = 0, \ldots, n-1$ **do**
  $\mathcal{C}_i \leftarrow$ `ComputeCell`$(\mathbf{p_i})$;
  $\mathcal{C}'_i \leftarrow$ `ComputeCell`$(\mathbf{p'_i})$;
  $\mathbf{R_i} \leftarrow$ `ComputeRotation`$(\mathcal{C}_i, \mathcal{C}'_i)$;
for $l = 0, \ldots, \mathcal{L}-1$ **do**
  $t = \frac{l}{\mathcal{L}-1}$;
  for $i = 0, \ldots, n-1$ **do**
    $\mathbf{R_i}(t) \leftarrow$ `InterpolateRotation`$(\mathbf{I}, \mathbf{R_i}, t)$;
  for $i \in \mathcal{T}$ **do**
    $\widehat{\mathbf{p}}_\mathbf{i}(t) \leftarrow (1-t)\mathbf{p_i} + t\mathbf{p'_i}$;
  $\widehat{\mathbf{p}}_\mathbf{0}(t), \ldots, \widehat{\mathbf{p}}_{\mathbf{n-1}}(t) \leftarrow$ `MinimizeEnergy`$(E_{ARAP}(t))$;

Rotation Interpolation using Slerp

We use the Spherical linear interpolation (Slerp) method to interpolate rotations [217]. This method produces a rotation at a constant angular velocity along the shortest great arc (*i.e.* a geodesic) on a unit quaternion sphere. The formula to compute a Slerp between two normalized quaternions $\mathbf{p}$ and $\mathbf{q}$ with parameter $t \in [0,1]$ is:

$$\text{Slerp}(\mathbf{p}, \mathbf{q}, t) = \mathbf{p}(\mathbf{p}^* \mathbf{q})^t \tag{3.5}$$

where $\mathbf{p}^*$ is the quaternion conjugate of $\mathbf{p}$. This expression can be rewritten without the t exponentiation as:

$$\text{Slerp}(\mathbf{p}, \mathbf{q}, t) = \frac{\sin(1-t)\theta}{\sin\theta}\mathbf{p} + \frac{\sin t\theta}{\sin\theta}\mathbf{q} \tag{3.6}$$

where $\cos\theta = \mathbf{p} \cdot \mathbf{q}$ is the inner product of two quaternions (i.e. if $\mathbf{p} = [s, (x,y,z)]$ and $\mathbf{q} = [s', (x', y', z')]$, then $\mathbf{p} \cdot \mathbf{q} = ss' + xx' + yy' + zz'$).

3.2 Rapidly-exploring random trees (RRT)

RRT [144] is a motion planning method originally developed for robotics applications and recently applied to structural biology (see Section 2.3.4). Essentially, RRT constructs a tree in a given n-dimensional space where each node represents a state and each edge connecting a pair of nodes indicates a possible transition between these states. RRT has gained popularity thanks to the Voronoi-bias property which enables the tree to grow preferably toward the unexplored regions of the space, thus avoiding the regions already visited.

While there are many possible implementations of RRT, this section presents only two types of growth for the *Connect* version of RRT[3]: the mono-directional variant where one tree is grown and the bi-directional variant where two trees are grown. The former is suited for problems where only one state is known such as ligand-unbinding-pathway search, while the latter is better suited for problems where two states are known such as transition-pathway search.

3.2.1 Mono-directional RRT

The mono-directional variant of RRT is presented in Algorithm 4. The tree $\mathcal{T}$ is initialized with a single node built from the initial state q_{start} (line 1). Then, as long as a given stopping condition is not satisfied, the target states q_t are randomly sampled and new branches are created toward them (line 2 to 4). Typically, this tree-growth process is stopped when the number of tree nodes exceeds a maximum value or when the tree reaches a desired region of the space.

Algorithm 4: Mono-directional RRT.

Input : An initial state q_{start}.
A `StoppingCondition` criterion for stopping the tree growth.
Output : A tree $\mathcal{T}$.

```
1 T ← InitTree(q_start);
2 while StoppingCondition = false do
3     q_t ← RandomState();
4     q_ext ← ExtendBranch(T, q_t);
5 return T;
```

Algorithm 5 shows the `ExtendBranch`$(\mathcal{T}, q_t)$ function for extending a tree $\mathcal{T}$ toward a target state q_t. First, the nearest node q_n to q_t in the tree is obtained at line 1. To improve the efficiency, advanced algorithms can be used to perform this step, such as the Approximate

[3]This version introduced in [139] attempts to extend a full tree branch for each random sample, which differs from the original *Extend* version proposed in [142].

Nearest Neighbor (ANN) search [112]. A new state q_{new} is then generated between q_n and q_t (line 2). The function $\texttt{Extend}(q_n, q_t)$ usually performs linear interpolation, i.e.

$$q_{new} = \delta \frac{q_t - q_n}{\|q_t - q_n\|} + q_n \tag{3.7}$$

where δ is the edge length or the extension step size of RRT. If q_{new} is found valid (line 3), it is added as a new node in the tree and a new edge is created between q_{new} and the nearest node (line 4 and 5). In robotics, a state is typically considered as valid if it does not lead to collisions either between the robot and the environment or between the robot and itself. However, other or additional conditions may be imposed. This extension scheme is repeated (line 6-7) as long as the new states are found valid and the target state q_t is not reached (not indicated in the algorithm for clarity). The function finally returns the last extended node.

Algorithm 5: $\texttt{ExtendBranch}(\mathcal{T}, q_t)$ function.

Input : current tree $\mathcal{T}$, a target state q_t.
Output : A new branch added to $\mathcal{T}$ with leaf node q_{new}.

1 $q_n \leftarrow \texttt{NearestState}(\mathcal{T}, q_t)$;
2 $q_{new} \leftarrow \texttt{Extend}(q_n, q_t)$;
3 **while** $\mathit{TestState}(\mathcal{T}, q_{new}) = \mathit{true}$ **do**
4 $\quad \texttt{AddNode}(\mathcal{T}, q_{new})$;
5 $\quad \texttt{AddEdge}(q_n, q_{new})$;
6 $\quad q_n \leftarrow q_{new}$;
7 $\quad q_{new} \leftarrow \texttt{Extend}(q_n, q_t)$;
8 **return** q_{new};

To summarize, the growth of an RRT tree iterates the three elementary steps illustrated in Figure 3.8: 1) sampling a random state which becomes a target state for tree extension, 2) searching the nearest node with respect to this target state, 3) growing the tree from the nearest node toward the target state.

3.2.2 Bi-directional RRT

Algorithm 6 shows how bi-directional RRT can find a path connecting the states q_{start} and q_{goal} by growing two trees from these states. Line 1 and 2 initialize the trees with their corresponding initial states. The trees are then grown until a stopping condition is fulfilled or until the trees get connected (line 3-9). For each iteration, a target state q_t is randomly sampled (line 4) and the `GrowTrees` function is applied, which attempts to extend $\mathcal{T}_{start}$ toward q_t and then connect $\mathcal{T}_{goal}$ with $\mathcal{T}_{start}$ (line 5). If the trees get connected, the process ends (line 7). Otherwise, another call to `GrowTrees` is performed, this time extending $\mathcal{T}_{goal}$

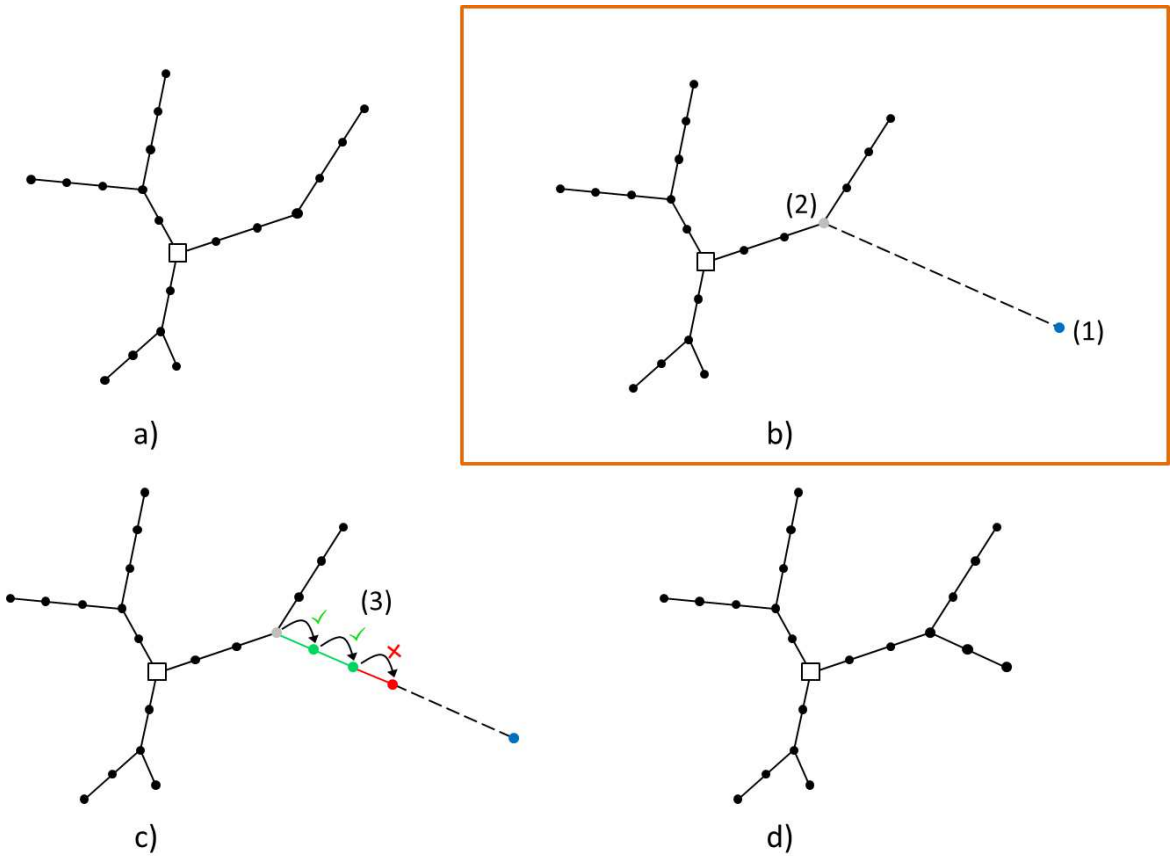


Fig. 3.8 An extension stage in RRT: a) The tree at a given stage of the search. The white square represents the root node. b) A state (1) is randomly sampled in a state space represented by the orange box (this state can be unrealistic, its role is only for providing a direction) and the nearest node from this conformation (2) is obtained in the tree. Note that the space bounds are typically set to be much larger compared with the volume covered by the tree. c) Several extension steps are attempted (3) from the nearest node toward the sampled state based on some acceptance criterion. The extension stops as soon as a new state is rejected. d) The resulting tree after the extension stage. This scheme is known to preferably explore new regions of the space.

toward q_t and connecting $\mathcal{T}_{start}$ with $\mathcal{T}_{goal}$ (line 9). Once the process is stopped and if the trees are connected, a path connecting q_{start} to q_{goal} is extracted (line 11).

The `GrowTrees` function used in Algorithm 6 is detailed in Algorithm 7. This function proceeds in two stages: first, an extension stage grows a branch for one tree toward a target state (line 1); then, a connection stage grows a branch for the second tree toward the last extended node in the previous extension stage (line 2). Typically, the `ConnectBranch` function is exactly the same as the `ExtendBranch` function, but other strategies can be used as will be shown later on in the bi-directional version of our proposed method ART-RRT.

Algorithm 6: Bi-directional RRT.

Input : The initial states q_{start} and q_{goal}.
Output : A path $\mathcal{P}$ connecting the initial states if any.

```
1  T_start ← InitTree(q_start);
2  T_goal ← InitTree(q_goal);
3  while StoppingCondition = false do
4      q_t = RandomState();
5      connected ← GrowTrees(T_start, T_goal, q_t);
6      if connected then
7          break;
8      else
9          connected ← GrowTrees(T_goal, T_start, q_t);
10 if connected then
11     P ← ExtractPath(q_start, q_goal);
```

Finally, the function returns whether this process leads to the connection of both trees (line 3). Figure 3.9 illustrates the extension and connection stages that appear in the `GrowTrees` function.

Algorithm 7: `GrowTrees`$(\mathcal{T}_A, \mathcal{T}_B, q_t)$.

Input : The trees $\mathcal{T}_A$ and $\mathcal{T}_B$, a target state q_t.
Output : A boolean, set to true if the two trees are connected.

```
1  q_ext^A ← ExtendBranch(T_A, q_t);
2  q_ext^B ← ConnectBranch(T_B, q_ext^A);
3  return q_ext^A = q_ext^B;
```

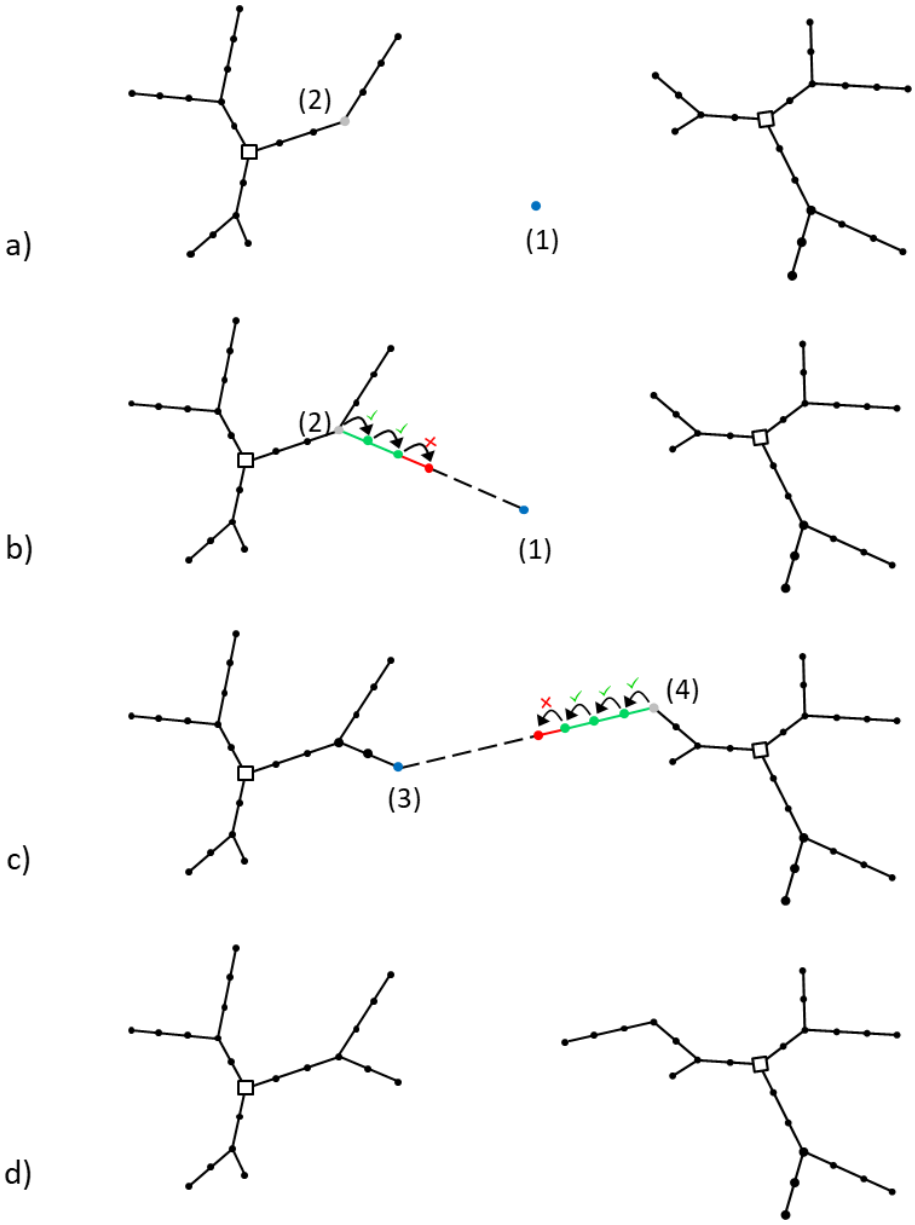


Fig. 3.9 Extension and connection stages used in the `GrowTrees` function of bi-directional RRT: a) The trees at a given stage of the search with the sampled state (1). The white squares represents the root nodes. b) Extension stage: `ExtendBranch` is applied on the left tree toward the target state. State (2) corresponds to the nearest node in the left tree to the target state. At the end of this step, two new nodes are accepted. c) Connection stage: `ConnectBranch` is applied on the right tree toward the last extended node (3) in the left tree. At the end of this step, three new nodes are accepted in the right tree. d) The trees after applying the `GrowTrees` function.

3.3 Software platform

All the proposed methods in this dissertation were developed in the Software for Adaptive Modeling and Simulation of Nanosystems (SAMSON) [113] developed by the team where the author spent his PhD study. SAMSON provides a multi-purpose platform for the research and analysis in various fields including material science and structural biology. Many modules are available in SAMSON for molecule editing, visualization, simulation, etc. Moreover, the users can also develop their own modules with the available Software Development Kit, and then share them with the scientific community or the public. In this dissertation, the author also employed some modules developed by his colleagues and shared through the SAMSON website. Figure 3.10 shows a nanotube and a protein structure visualized in SAMSON and Figure 3.11 shows a portion of the graphical user interface of SAMSON.

The proposed methods were implemented in serial C++ codes as SAMSON modules, deployed on the SAMSON platform 0.6.0, on Windows 10 64-bit operating system, Intel Core i7-3940XM CPU 3.20 GHz and 16GB RAM.

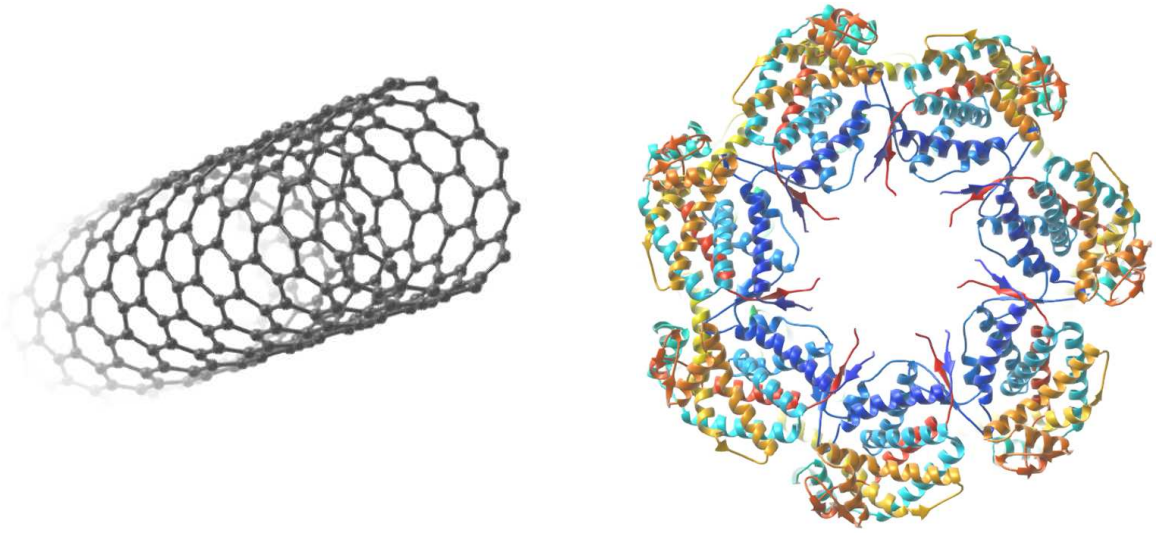


Fig. 3.10 a) A nano-tube created in SAMSON b) A secondary-structure visualization of GroEL (pdb entry 1SS8) in SAMSON.

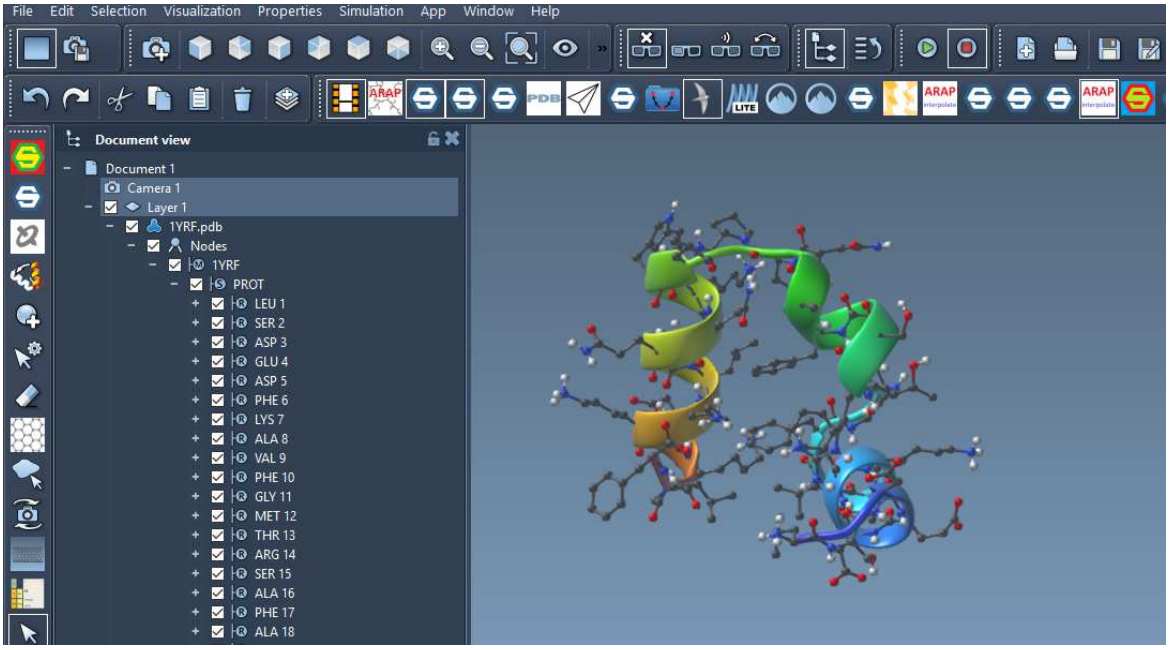


Fig. 3.11 A portion of the SAMSON Graphical User Interface.

Part II

ARAP interpolation pathways for molecular systems

Chapter 4

Basic method

4.1 Method

This section presents our methodology, which is based on ARAP interpolation (ARAPi), for generating molecular paths. This work was also published in the journal of Computer-aided Molecular Design in 2017 [174]. The approach follows the global framework presented in Figure 4.1. The input is a pair of structure (intial and target) described by a set of atoms embedded in the 3D cartesian space and connected by covalent bonds. The output is a morphing path comprising intermediate conformations. As we will see, the ARAP interpolation method can efficiently compute a consistent path which tend to preserve the local rigidity of the initial structure.

4.1.1 Preprocessing

The ARAPi method requires a one-to-one atom mapping (or matching) between the intial and target structures. To obtain such a mapping, a sequence alignment of the residues in both structures is first performed using the MUSCLE method [73]. Atoms of the aligned residues are then mapped by their PDB names[1]. More robust mapping methods could be introduced, but this is outside the scope of our present work. Only matched atoms will be treated by the ARAP method and hence labeled as **ARAP atoms**, whereas the other atoms are labeled as **non-ARAP atoms**.

The *ARAP topology* is then built as follows: each ARAP atom is considered as a vertex and each bond between two ARAP atoms in the initial structure as an edge[2]. However,

[1] More specifically, two atoms are mapped if they share the same name or have equivalent names such as 1HG2 and HG21.

[2] Optionally, to build edges, one could consider only the bonds which are present in both the initial and the target structures.

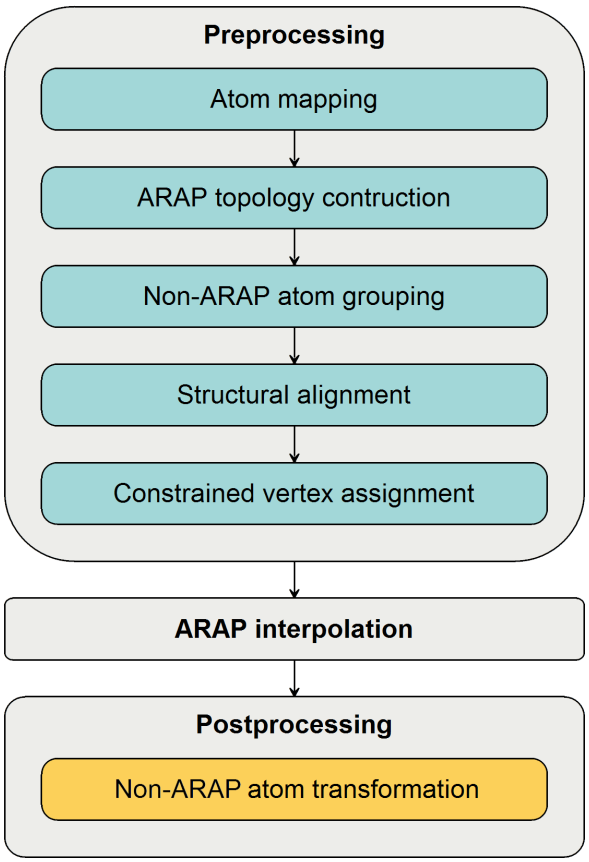


Fig. 4.1 Global framework to generate molecular paths from an initial to a target conformation using the ARAP interpolation method.

such a simple construction may produce disconnected components[3] due to missing residues in the initial structure or a broken connectivity when discarding non-ARAP atoms. Since the ARAPi method requires that all vertices belong to one connected component, two additional procedures are done to recover the connectivity. First, if the initial structure is itself disconnected due to missing residues, we reconnect the structure by creating extra edges. Hence, an edge is added between the alpha carbons just before and just after each missing group of residues. Second, to create connections among ARAP atoms, we use a recursive procedure called *connect* (also illustrated in Figure 4.2), which operates as follows. For each ARAP atom a, $connect(a, m_i)$ is first called for each neighbor m_i of a. The procedure creates an edge between a and m_i, if m_i is also an ARAP atom. If m_i is not an ARAP atom, then $connect(a, n_j)$ is called on each neighbor n_j of m_i. To ensure the termination of the recursive procedure, we check that each atom is visited at most once. The next step forms groups of connected non-ARAP atoms. Each group is associated to a reference ARAP atom that

[3] A connected component is a set of vertices and edges such that any vertex can be reached from another vertex by traversing through a series of edges in the same component.

shares a bond with one of the atoms in the group. As we will see later, the non-ARAP atom positions are deduced from the ARAP atom positions in the intermediate conformations.

ARAP interpolation may only provide deformations. Hence, we need additional constraints to define the global rotation and translation of the system. To determine these constraints, we perform a 3D structural alignment of the initial and the target structures. This cancels the need for any global rotation, while improving the robustness of the solution obtained, by removing large local rotations during the ARAP interpolation. We use the fast QCP variant developed by Liu et al. [155] of the quaternion superposition methods [104, 126] to align both structures. The last step of the preprocessing phase settles the translational part by defining an arbitrary ARAP atom as constrained. This atom position will be linearly interpolated during the ARAP interpolation phase.

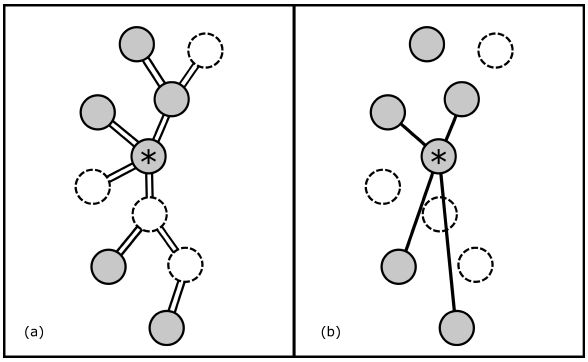


Fig. 4.2 Example of the ARAP connectivity construction by the *connect* procedure for the ARAP atom marked with a star. ARAP atoms are grey and non-ARAP atoms are white. (a) The initial molecular topology. (b) ARAP edges created from the marked ARAP atom and represented by bold lines. When the *connect* procedure stops for the marked atom, the top-left ARAP atom is not connected, since its connection to the marked atom passes through another ARAP atom.

In summary, the preprocessing phase provides the following inputs to the ARAP algorithm: the (aligned) vertex positions of the initial and target states, the edges that link these vertices together, and the index of the constrained vertex.

4.1.2 ARAP interpolation

Given the ARAP topology, i.e. vertices, edges, and constrained vertices obtained in the preprocessing phase, the algorithm for ARAP interpolation (ARAPi) presented in Section 3.1.4 is applied to generate intermediate conformations. However, since a direct application of ARAPi would not allow to reach the target state perfectly, we propose to modify the ARAP energy defined in Equation 3.4 to address this issue.

Modified ARAP energy

With ARAPi, the computed shape at $t = 1$ does not coincide with the target shape. To overcome this limitation, we introduce for each edge e_{ij} an extra rotation $\mathbf{R_{ij}}$ and a stretching factor s_{ij}. Hence, after finding the aligning rotation $\mathbf{R_i}$, we find, for each set $\mathcal{N}_i$ and each $j \in \mathcal{N}_i$, the rotation $\mathbf{R_{ij}}$ and stretching factor s_{ij} such that:

$$s_{ij}\mathbf{R_{ij}}\mathbf{R_i}(\mathbf{p_i} - \mathbf{p_j}) = \mathbf{p'_i} - \mathbf{p'_j} \tag{4.1}$$

The ARAP energy for an intermediate shape must be adapted accordingly:

$$E_{ARAP}(t) = \sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_\mathbf{i}(t) - \widehat{\mathbf{p}}_\mathbf{j}(t) - s_{ij}(t)\mathbf{R_{ij}}(t)\mathbf{R_i}(t)(\mathbf{p_i} - \mathbf{p_j})||^2 \tag{4.2}$$

where $s_{ij}(t)$ is linearly interpolated between 1 and s_{ij}, i.e. $s_{ij}(t) = (1-t) + t \cdot s_{ij}$ and $\mathbf{R_{ij}}(t)$, like $\mathbf{R_i}(t)$, are computed using the Slerp method for rotation interpolation presented in Section 3.1.4.

Similar to the minimization of E_{ARAP}, the minimization of $E_{ARAP}(t)$ shown in the last equation leads to solving the linear algebra system $\mathbf{L}\widehat{\mathbf{p}}(t) = \mathbf{b}(t)$. These matrices have the same size with those in the problem of minimizing E_{ARAP}. Because the symmetric and positive-definite matrix $\mathbf{L}$ is independent of t, it can be factorized using the Cholesky decomposition only once and used to solve for all the intermediate shapes. The complexity of this implementation of ARAP interpolation is, therefore almost $\mathcal{O}(\mathcal{L}(n - n_c))$ where n is the total number of vertices, n_c the number of constrained vertices and $\mathcal{L}$ the interpolation-path size. The readers are referred to Appendix A.3 for the construction of these matrices.

4.1.3 Postprocessing

Since the ARAPi method only computes the ARAP-atom positions, this postprocessing phase computes the non-ARAP-atom positions based on the ARAP-atom positions.

During the preprocessing phase, each non-ARAP group was associated to an ARAP atom. Hence, in the intermediate conformation at t, the position of an atom n from a non-ARAP atom group associated to the ARAP atom a can be computed as following,

$$\mathbf{p}_n(t) = \mathbf{p}_a(t) + \mathbf{R}_a(t)\,(\mathbf{p}_n - \mathbf{p}_a) \tag{4.3}$$

where $\mathbf{p}_n(t)$ and $\mathbf{p}_a(t)$ are the positions of the non-ARAP and ARAP atoms in the conformation at t, respectively. $\mathbf{p}_n$ and $\mathbf{p}_a$ are the positions of the non-ARAP and ARAP atoms in the

initial structure, respectively. $\mathbf{R}_a(t)$ is the interpolated rotation of the cell $\mathcal{C}_a$ whose central atom is atom a.

4.2 Experiments and results for ARAP interpolation

We have implemented the proposed method in C++ as a module of the SAMSON software platform [113]. The Eigen library is used for solving the linear system of equations [90].

All the benchmarks presented here come from those proposed by the authors of the so-called geometric targeting method in [76]. Since the ARAP method requires connected components, only 13 benchmarks of single-chain molecules were considered. Among them, Heparin Cofactor II was removed because its target structure has too many missing residues (more than 20 consecutive residues). Therefore, only 12 benchmarks are presented here. For each of them, the ARAP interpolation method is applied to generate a path composed of 20 conformations. The benchmark names, as well as the computational times, are shown in Table 4.1. The ARAP time is shown per conformation because it grows linearly with the number of conformations of the path.

As one can see, our methodology has globally a very low computational cost. The longest total time to obtain a path of 20 conformations is 1.08 s for DNA Polymerase which has 7313 ARAP atoms. The interpolation time per conformation per ARAP atom (not shown in the table) is about 0.006 ms for all the benchmarks. This value is 13 to 82 times smaller for the same quantity shown in [76].

When visually inspecting the paths obtained by the ARAP interpolation method, the results appear reasonable except for the Collagenase and Spindle Assembly Checkpoint Protein, where steric clashes occur. We provide below a more detailed analysis of the intermediate structures obtained from the ARAP paths.

Table 4.1 Benchmark details and running time. A chain id is specified after each pdb code.

#	Protein Name	Initial/Target Structure	No. of non-ARAP and ARAP atoms	CPU Time (ms)				Figure
				Pre-processing	ARAP time per conformation	Post-processing	Total	
1	5'-Nucleotidase	1HP1(A)/1HPU(C)	2/4027	130	22	0	570	4.8a
2	Adenylate Kinase	4AKE(A)/1AKE(A)	0/1656	47	9	0	227	4.5a
3	Alcohol Dehydrogenase	8ADH(A)/6ADH(A)	4/2784	87	16	0	407	4.6a
4	Calmodulin	1CFD(A)/1CFC(A)	1072/1166	49	6	33	202	4.5b
5	Collagenase	1NQD(A)/1NQJ(B)	0/901	26	5	0	126	4.7a
6	Dengue 2 Virus Envelope Glycoprotein	1OAN(A)/1OK8(A)	163/2962	95	17	6	441	4.8d
7	Dihydrofolate Reductase	1RX2(A)/1RX6(A)	1/1268	36	7	0	176	4.6b
8	Diphtheria Toxin	1DDT(A)/1MDT(A)	0/4021	130	23	0	590	4.8b
9	DNA Polymerase	1IH7(A)/1IG9(A)	26/7313	261	41	1	1082	4.8c
10	Pyrophosphokinase	1HKA(A)/1Q0N(A)	0/1267	34	7	0	174	4.6c
11	Pyruvate Phosphate Dikinase	1KBL(A)/2R82(A)	7/6745	234	38	0	994	4.8e
12	Spindle Assembly Checkpoint Protein	1DUJ(A)/1KLQ(A)	1479/1509	64	8	46	270	4.7b

4.2.1 Preservation of bond lengths, bond angles and dihedral angles

To show the structure preservation property of the ARAP interpolation method, we look at how bond lengths, bond angles and dihedral angles in each intermediate conformation deviate from those in the initial structure.

Consider the first benchmark, 5'-Nucleotidase, which has a large rotation (about 90°) in one part of the structure (Figure 4.8a). For this benchmark, Figure 4.3a represents the absolute value of the change in bond length during interpolation with respect to the initial structure with our approach. The horizontal axis shows the conformation sequence index along the path, where 1 corresponds to the first conformation and 20 corresponds to the last one. The vertical axis is the change in bond length in angstrom (Å). The blue line plots the average change in bond length in each conformation. For each conformation, a blue bar shows the standard deviation from the mean value. The dotted and solid red lines plot the maximum and minimum values, respectively, of the change in bond length for each conformation. Therefore, the area between the maximum and minimum curves represents the range of the (absolute) change in bond length. The *deviation area*, *i.e.* the area between the blue bars, is where most values are observed.

The maximum curve in Figure 4.3a starts at 0 Å for the first conformation because here, it coincides exactly with the initial structure. The maximum value increases until 0.616 Å (the 11^{th} conformation), then decreases and finally reaches 0.281 Å at the target structure. Our method does indeed reach the target structure. However, this final value is not 0 Å because the bond lengths of the target structure deviate from those of the initial structure to some degree. The deviation area is bounded in the range [-0.012, 0.025] Å, indicating that a large number of bond lengths generated by the ARAP interpolation do not deviate more than 0.025 Å from those in the initial structure.

The same type of plot for the linear interpolation method[4] is shown in Figure 4.3d. The maximum values are significantly larger (the peak is around 1.118 Å at the 10^{th} conformation) compared with those of the ARAP interpolation method. The deviation area has also higher mean and standard deviation values, which suggests that the ARAP interpolation method preserves bond lengths much better than the linear interpolation method.

Similarly, Figure 4.3b and 4.3e show that ARAP interpolation preserves bond angles better than linear interpolation. On the other hand, the results on the dihedral angle do not differ greatly in both methods, as shown in Figure 4.3c and 4.3f. This is because the chosen one-ring topology of a cell contains no dihedral angles, and hence dihedral angles are not constrained by the proposed method. Therefore, the method naturally favors

[4]The path by the linear interpolation method is also generated after the initial and target conformations are 3D structurally aligned.

Table 4.2 Comparison of maximum mean values of changes in bond lengths, bond angles, dihedral angles and consecutive-C_α distances.

#	Protein Name	Bond change ($\times 10^{-3}$Å)		Angle change (°)		Dihedral angle change (°)		Consecutive-C_α distance change ($\times 10^{-3}$Å)	
		ARAP	Linear	ARAP	Linear	ARAP	Linear	ARAP	Linear
1	5'-Nucleotidase	9	124	1.4	4.7	6.4	6.4	19	274
2	Adenylate Kinase	16	92	2.7	5.0	14.4	14.4	46	97
3	Alcohol Dehydrogenase	23	81	3.6	5.8	19.8	19.8	53	53
4	Calmodulin	4	71	1.3	2.9	7.2	7.2	7	82
5	Collagenase	18	118	3.5	6.0	14.5	14.5	57	188
6	Dengue 2 Virus Envelope Glycoprotein	11	141	2.9	6.9	18.0	18.0	27	208
7	Dihydrofolate Reductase	23	64	2.4	4.7	9.7	9.7	57	72
8	Diphtheria Toxin	19	216	3.0	10.5	11.1	11.1	40	483
9	DNA Polymerase	6	57	1.6	3.7	10.8	10.8	14	30
10	Pyrophosphokinase	12	80	2.5	5.0	10.0	10.0	22	81
11	Pyruvate Phosphate Dikinase	15	93	1.4	3.6	7.8	7.8	32	227
12	Spindle Assembly Checkpoint Protein	23	200	4.8	10.5	22.8	22.8	46	352

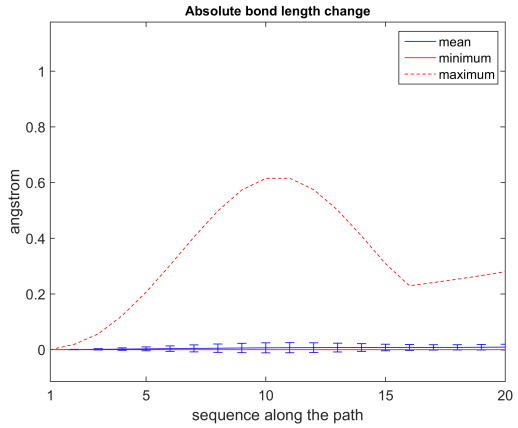
global deformations through changes in dihedral angles, which is typically expected in conformational changes of proteins. Further discussion on the dihedral angle can be found in Section 4.3.

We have detailed above the results for 5'-Nucleotidase, but the same behavior is also observed for the other benchmarks (see Appendix B). Table 4.2 shows the maximum mean value across the path regarding the change in bond lengths, bond angles and dihedral angles for each benchmark. For bond lengths and bond angles, the values in the ARAP columns are always lower than those in the Linear columns, *i.e.* the ARAP method preserves these quantities better. However, the results for dihedral angles do not follow this pattern. In fact, maximal changes in dihedral angles occur at the final conformation, which is why the values are the same for both methods in the table.

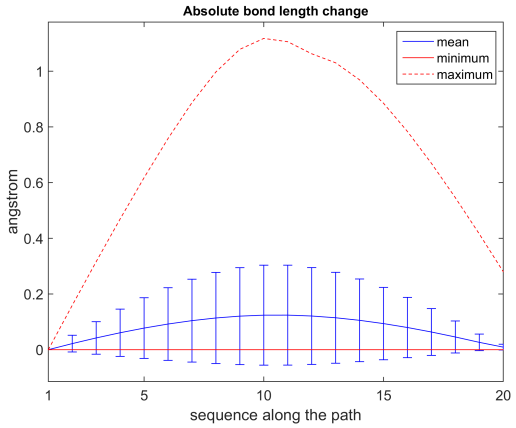
4.2.2 Preservation of distances between consecutive alpha carbons

Most coarse-grain methods use distances among C_α atoms for morphing. In [130], these distances are interpolated between the initial and target structures, whereas in [39] those between consecutive C_α atoms in intermediate structures are constrained in the range [3.7, 3.9] Å. Figure 4.4a shows how consecutive-C_α distances evolve along the path for 5'-Nucleotidase using the proposed method. Although the value range area is within [3.56, 3.82] Å, the deviation area indicates that most values lie in the range [3.77, 3.86] Å. This shows that the ARAP interpolation method tends to preserve consecutive-C_α distances as well. The linear interpolation method does not preserve well this quantity as shown in Figure 4.4b.

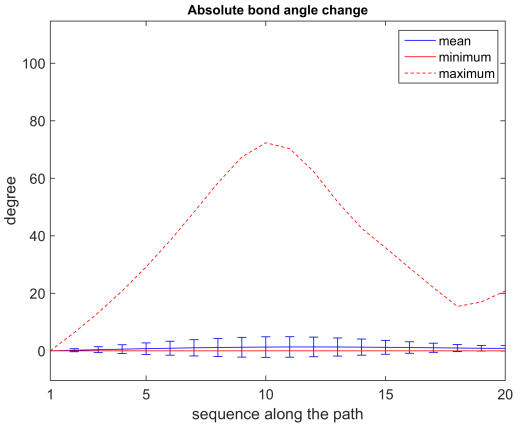
(a) ARAP interpolation



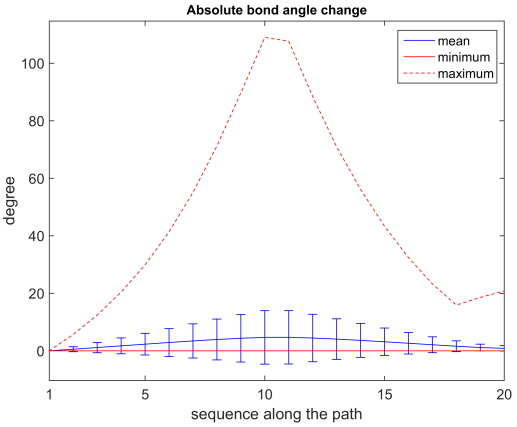
(d) Linear interpolation



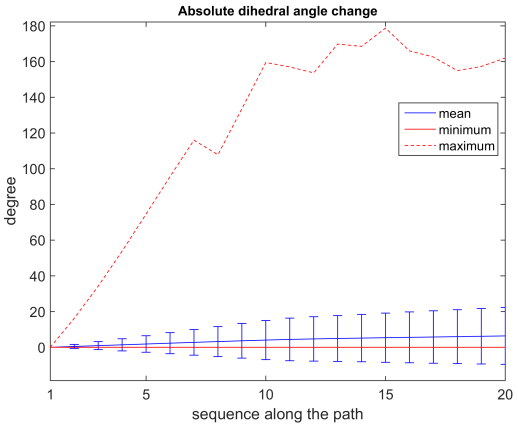
(b) ARAP interpolation



(e) Linear interpolation



(c) ARAP interpolation



(f) Linear interpolation

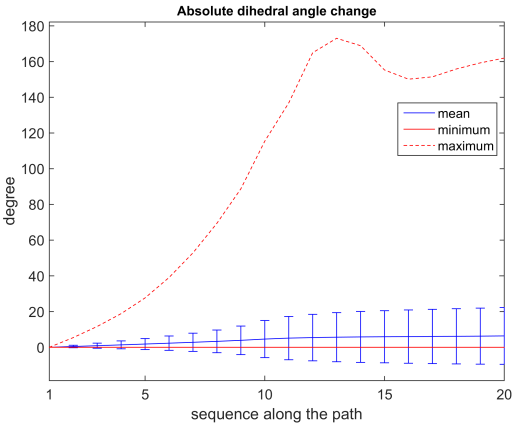


Fig. 4.3 Statistics of absolute changes in bond length, bond angle and dihedral angle for 5'-Nucleotidase. Results from ARAP interpolation for (a) bond length, (b) bond angle, (c) dihedral angle. Results from linear interpolation for (d) bond length, (e) bond angle, (f) dihedral angle.

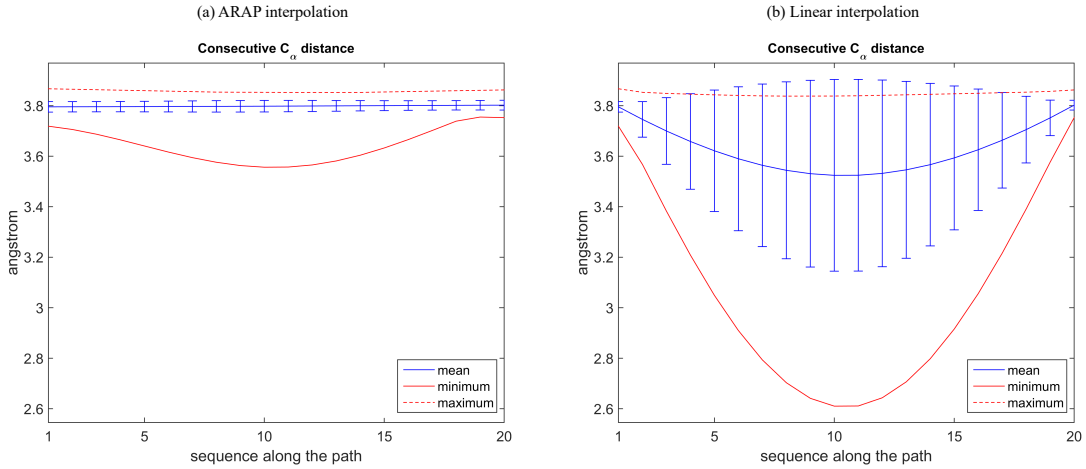


Fig. 4.4 Statistics of consecutive-C_α distances for 5'-Nucleotidase from (a) ARAP interpolation, (b) Linear interpolation.

As Table 4.2 indicates, this behavior is also observed for the other benchmarks (see also Appendix B for the plots of the rest of the benchmarks). Similar to bond lengths and bond angles, the changes in consecutive-C_α distances are always lower in the ARAP method than those in the linear interpolation method.

4.2.3 Structural motions along the path

In general, the ARAP interpolation method generates reasonable global motions for all the benchmarks.

The closing motions of Adenylate Kinase and Calmodulin are captured by the ARAP method in Figure 4.5.

Figure 4.6a and 4.6b show the shear motions of one part of Alcohol Dehydrogenase and Dihydrofolate Reductase, respectively. This type of motion is also observed in Pyrophosphokinase (Figure 4.6c). A slight change in the loops (green and light blue) on the left is caused by a shear motion of the green helix and light-blue loop.

In Dengue 2 Virus Envelope Glycoprotein (Figure 4.8d), a closing motion of a hinged domain (red, orange and yellow) and a rotation of about 180° of the small light blue loop are captured.

Our proposed method agrees with [76] for the motion of Diphtheria Toxin, *i.e.* a rotation of about 180° of a large domain (see Figure 4.8b). Note that this motion was not found by the Yale Morph Server [137].

In Figure 4.8a, our method shows a rotation of about 90° of one part of 5'-Nucleotidase. Figure 4.8c (DNA Polymerase) shows the closing motion of a group of green helices and slight rotations of other domains (red and yellow).

For Pyruvate Phosphate Dikinase (Figure 4.8e), one observes an opening motion carried out by two domains. One domain rotates by 90° (yellow and orange) while another rotates less than 90° (in green and purple).

Some limitations of the proposed method are shown in the case of Collagenase and Spindle Assembly Checkpoint Protein. Steric clashes are observed during the formation of the dark blue helix loops for both cases (Figure 4.7). This is due to the chosen Slerp method for interpolating rotations, which restricts the rotation angle to the range $[0, \pi]$ while the helix formation requires a larger rotation. In the case of Spindle Assembly Checkpoint Protein, one also observes clashes between the red strand and other strands of the beta sheet. This is because the ARAP method does not guarantee the absence of steric clashes between distant (non-bonded) parts of the structure.

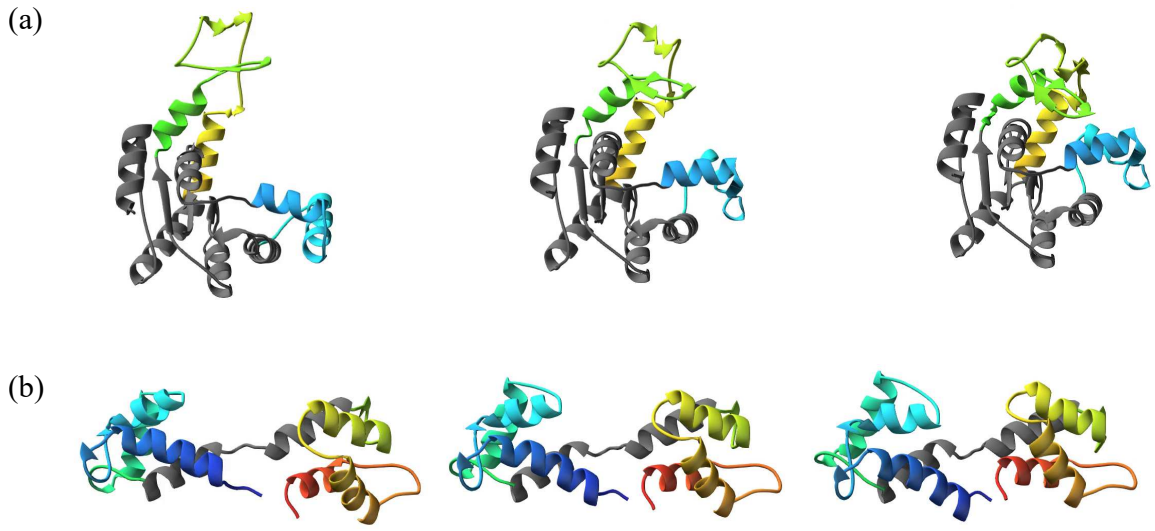


Fig. 4.5 Open-to-close motions in the colored parts of (a) Adenylate Kinase. (b) Calmodulin.

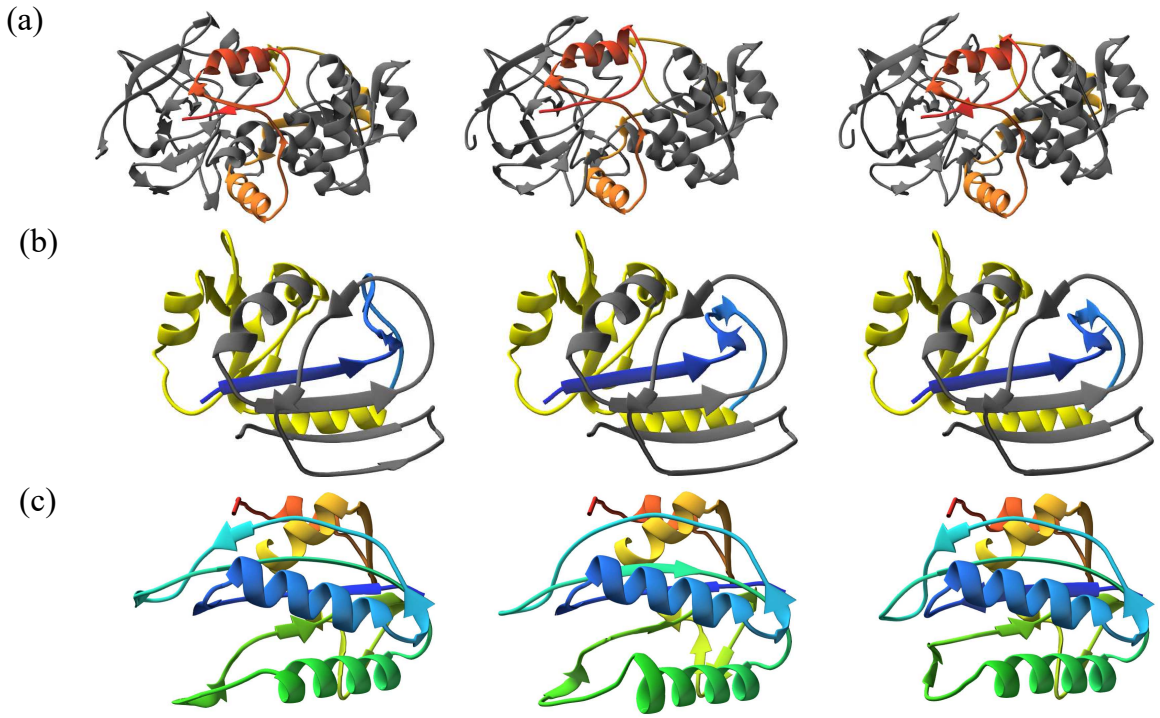


Fig. 4.6 Shear motions. (a) Alcohol Dehydrogenase: shear motion of the colored part compared to the static grey part. (b) Dihydrofolate Reductase: shear motion of the blue strand of the beta sheet. (c) Pyrophosphokinase: shear motion of the green helix and the light blue loops.

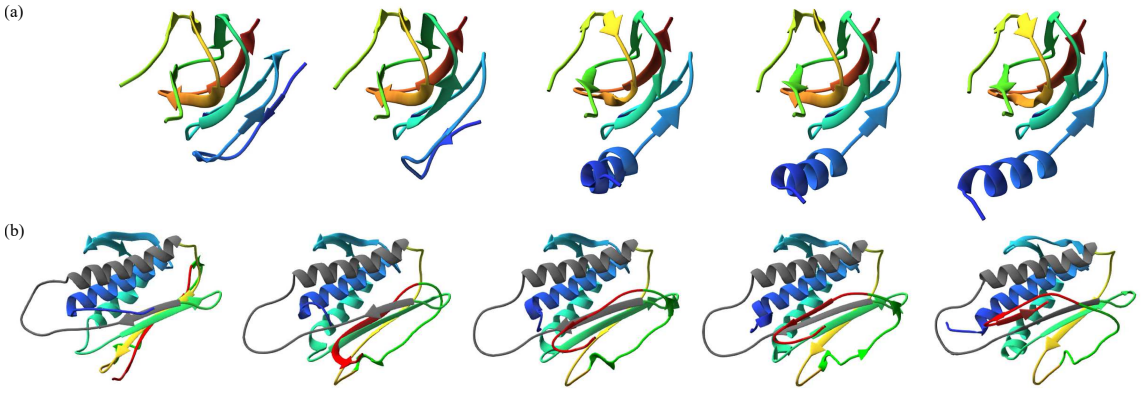


Fig. 4.7 (a) Collagenase: Steric clashes occur in the formation of the dark blue helix loop. (b) Spindle Assembly Checkpoint Protein: steric clashes occur in the formation of the dark blue helix and the motion of the red loop.

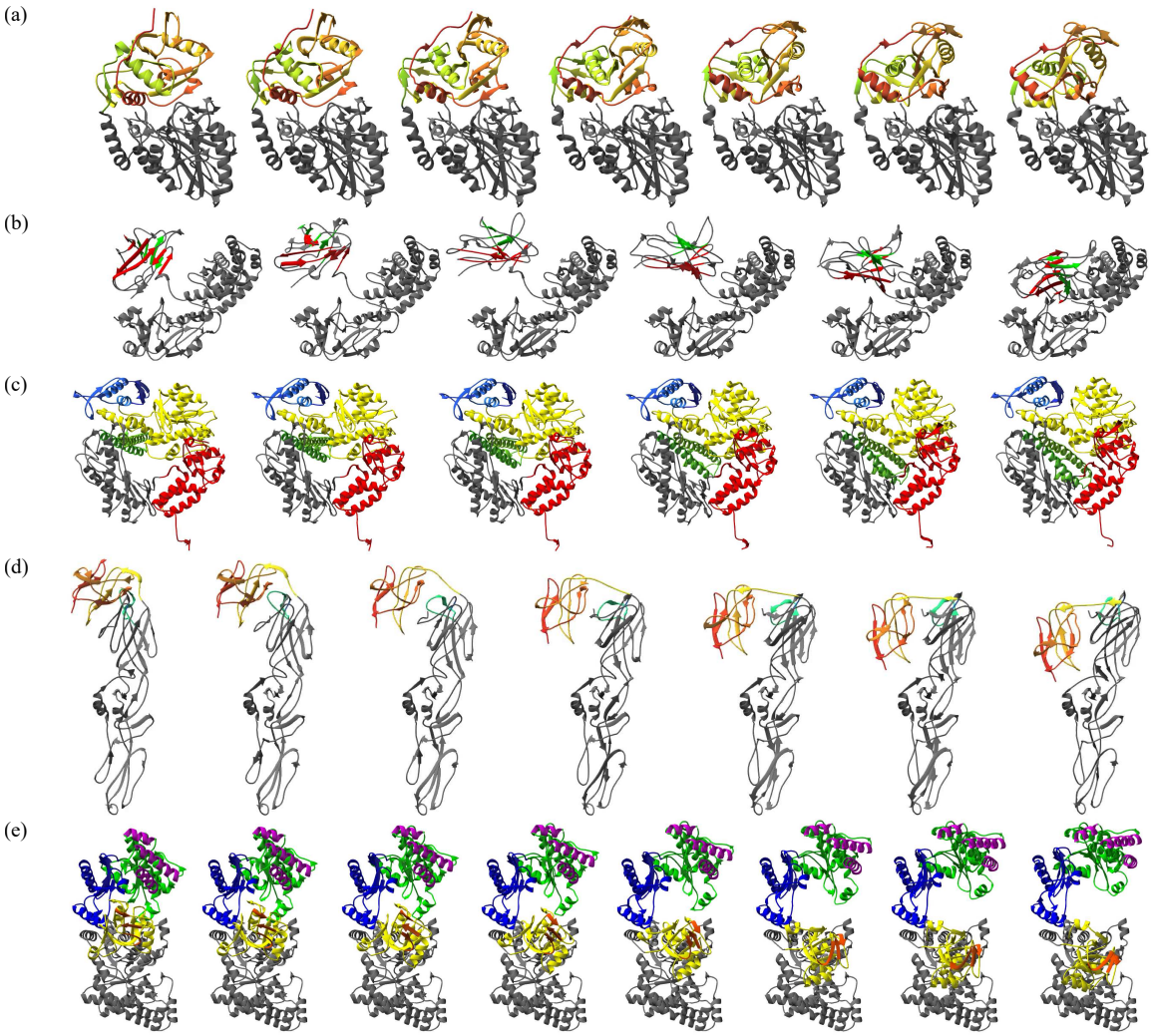


Fig. 4.8 Motions of (a) 5'-Nucleotidase: Rotation of about 90^o of the colored part can be detected by paying attention to the red helix. (b) Diphtheria Toxin: Rotation of about 180^o of the colored part can be detected by observing the beta sheets, *i.e.* the red sheet goes behind the green one in the final picture whereas it is ahead in the the first picture. (c) DNA Polymerase: Closing motion of the green helices. (d) Dengue 2 Virus Envelope Glycoprotein: The closing motion of the red and yellow domains by rotation. The small light blue loop also rotates by 180^o. (e) Pyruvate Phosphate Dikinase: The opening motion accomplished by the green-purple domain and the yellow-orange domain. A rotation of 90^o of the yellow domain can be seen clearly by the orientation of the orange arrows. The rotation of the green-purple domain is smaller.

4.3 Discussion and Conclusion

4.3.1 Discussion

Arap energy and dihedral angles

As already stated in Section 2.2, one classical way to represent molecular systems is through the use of dihedral angles (see e.g. [250]). Hence, one can represent the backbone of a protein in terms of ϕ, ψ, ω angles and the mobility of the side chains in terms of dihedral angles χ. This internal coordinate representation is commonly used, since it is known that bond lengths and bond angles vary only slightly at room temperature whereas major conformational rearrangements are often due to dihedral-angle variations [200].

There is an interesting connection between the internal-coordinate representation and the ARAP-energy definition. If the one-ring topology is used for ARAP cell construction, any move in internal coordinates is just a combination of rigid rotations of cells, which results in zero ARAP energy. On the other hand, conformations produced by ARAP interpolation may not result from changes in internal coordinates because our solution only minimizes the ARAP energy, which may be not zero along the path. Therefore, the conformations produced by ARAP interpolation may be close to those produced by changes in internal coordinates, but they also include changes in bond lengths and bond angles. This helps ARAP interpolation to overcome two obstacles faced by internal coordinates moves: the impossibility to reach the target conformation due to fixed bond lengths and bond angles, as well as the loop closure problem [213].

Mesh quality

In our methodology, ARAP edges are built from covalent bonds in molecular systems (through the recursive *connect* procedure). Hence, the ARAP topology constructed during the preprocessing phase does not form a mesh composed of triangles or tetrahedrons, as is typically the case in computer graphics. Despite the simple mesh that we used, the results have shown that the method works efficiently and produces paths with adequate geometric properties.

Large rotations

One limitation of the proposed methodology is that the amplitudes of local rotations are smaller than π due to the Slerp method. In our case, the initial structural alignment performed during the preprocessing phase alleviates this problem to some degree. However, this may

not be sufficient to represent helix formation, as shown for Collagenase and the Spindle Assembly Checkpoint Protein. In 2D computer graphics, this issue can be resolved by correcting rotation angles, so that the absolute angle differences between adjacent cells are lower than π [18]. Unfortunately, this cannot be directly applied in 3D because the rotation axes of adjacent cells do not align. Some recent approaches have been proposed to address this problem [123], but they are not very adapted to the meshes representing molecular systems. Hence, original methods still need to be developed to address this limitation.

Symmetry

The method presented so far is not symmetric, *i.e.* the result of an interpolation from the target to the initial structure would differ from that of the reverse direction. However, this property may be interesting for some applications. An easy way to tackle this limitation is to perform ARAP interpolation from both directions and take the average of the results. However, this solution would double the computational cost while in many cases, the result from one direction can be already close to the average produced by a bi-directional solution. Hence, such a symmetric version of ARAP has not been implemented yet.

4.3.2 Conclusion

This chapter presented a new morphing method for generating interpolation paths between two given molecular structures. This method relies on the ARAP technique used in computer graphics to edit complex meshes while preserving local characteristics of the initial structure. Despite being a purely geometrical approach, the proposed method generates interpolation paths at a very low computational cost, while preserving well bond lengths and bond angles.

In the next chapter, we present the energy-based enhancement for the ARAP interpolated paths.

Chapter 5

Energy-based enhancement for ARAP interpolation paths

Chapter 4 has shown that ARAP interpolation paths are geometrically reasonable because they tend to preserve the bond lengths and bond angles of the initial structure. However, to render these paths biologically plausible, they need to be optimized. In structural biology, path-optimization methods such as the Nudged Elastic Band (NEB) [120, 101] or String [182] methods are able to locally optimize a given path toward a Minimum-Energy Path (MEP). Therefore, we would like to propose a framework in this chapter to generate an optimized path close to a MEP from a given pair of initial and target structures.

As stated in Section 2.3.1, MEPs are interesting to researchers because essential properties can be extracted from them, such as reaction coordinates or transition states. Path optimization methods usually produce different MEPs from different initial paths. Among these optimized paths, the ones with the lowest energy barriers are most interesting because they require least energy for a reaction/transition to happen. Therefore, in this chapter, we would also like to compare the effect of three path generation methods: ARAP interpolation (ARAPi), Linear interpolation (Linear) and Linear Synchronous Transit (LST) for generating initial paths in the proposed framework.

In the following sections, the framework for generating optimized paths is first presented with an overview of the possible methods for generating initial paths; then, the experiments and results.

5.1 Framework for optimized-path generation

The framework for obtaining an optimized path from an arbitrary initial and a target protein structure is shown in Figure 5.1. It is composed of two main stages: *input processing* and *path processing*. The first stage includes *structure reparation* where missing atoms or residues are reconstructed and *structure preparation* where the mutation issue is resolved. After the first stage, the initial and target structures have the same set of atoms and are locally minimized. The second stage includes *path generation*, *path reparation* and *path optimization*. It takes in the initial and target structures resulting from the first stage and gives the optimized path as the output. The details on these stages are presented below.

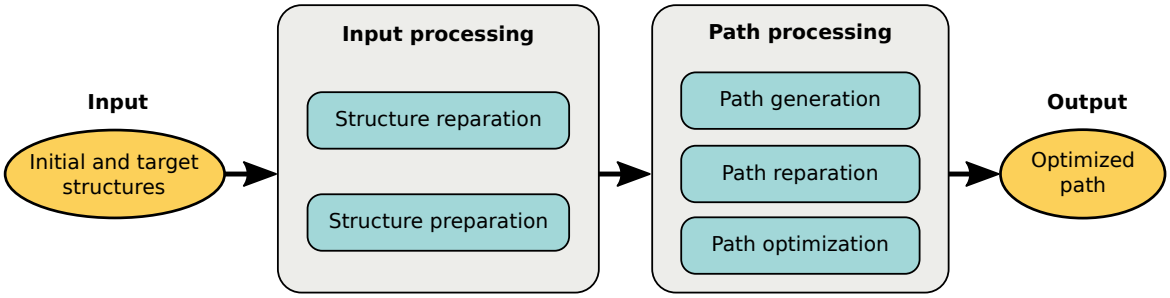


Fig. 5.1 The proposed framework for generating energy-optimized paths from an initial and a target structure.

5.1.1 Input processing

Structure reparation

Because our approach requires energy evaluations, the complete protein structures must be available, which is not usually the case due to missing atoms or residues in the obtained structures. Therefore, the objective of this step is to repair the structures by adding missing residues and atoms. Many tools are available for this task and the ones that we use will be mentioned in the section on the results.

Structure preparation

To apply a path-optimization method such as the NEB method, the initial and target structures must have exactly the same set of atoms. However, this is rarely the case because biological structures taken from online databases are often mutated structures. To deal with this problem, we generate new initial and target structures by applying our ARAPi method presented in

Section 4.1 with no intermediate states, i.e. generating a path containing only two states. The first and last states in this path are the new initial and target structures, respectively.

The use of ARAPi for generating a new initial structure and a new target structure has the following advantages. First, the new target structure has the same set of atoms as the new initial structure (they also have the same set of atoms as the old initial structure). Secondly, in the new target structure, the positions of non-mutated atoms are unchanged while the positions of the mutated atoms are rigidly displaced compared with those in the old target structure. Thirdly, the new initial structure and the new target structure are geometrically aligned. Figure 5.2 shows the contrast between the new target structure and the old target structure as an example.

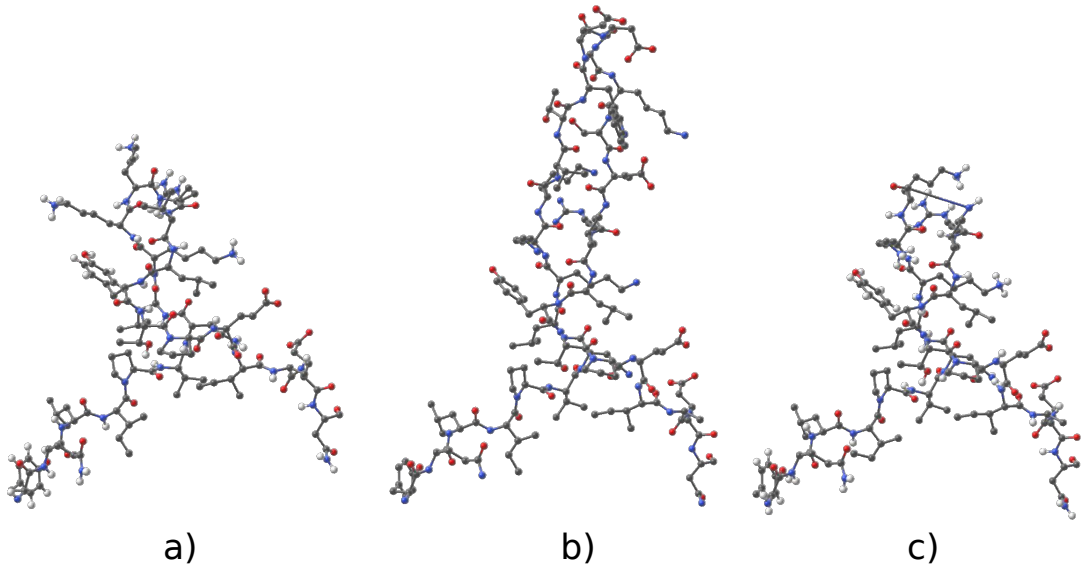


Fig. 5.2 A structure-preparation result using the ARAP interpolation method. The picture shows only one corresponding part from the initial and target protein structures. a) The new initial structure generated with the ARAP interpolation method. This structure has the same set of atoms as the old initial structure (not shown). It is also 3D geometrically aligned with the old target structure. b) The old target structure. c) The new target structure generated with the ARAP interpolation method. This structure has exactly the same set of atoms as the new initial structure. Moreover, the matched atoms in the new target structure with the old target structures have the same positions while the unmatched atoms in the new target structure (for e.g. hydrogen atoms in white balls) with the old target structures are rigidly displaced. The figure (c) also shows that the new target structure has a long bond on the top that connects two distant residues due to extra residues in the old target structure. This bond-stretch problem is dealt afterwards by local minimization.

Finally, since most path-optimization methods find paths between two local minima, the new initial and new target structures are relaxed to their local minima by a local minimization

process. This minimization process can also resolve the bond-stretch problem due to mutation such as the one in Figure 5.2c.

5.1.2 Path processing

Path generation

After the input processing stage, the new initial and target structures are at their local minima and have exactly the same set of atoms. Let us call them the initial and target conformations from now on. Many path-generation methods are available, however, we consider only three methods here: ARAP interpolation (ARAPi), linear interpolation (Linear), and Linear Synchronous Transit interpolation (LST).

The ARAPi method generates intermediate structures by conserving the local rigidity as-much-as possible, following the principle presented earlier in Section 4.1.2 of Chapter 4.

The Linear method generates intermediate structures by linearly interpolating atom positions between the initial and target conformations. Let us define t as the interpolation instance such that $t \in [0,1]$, and $t = 0$ and $t = 1$ correspond to the initial and target conformations, respectively. Then, the position of the ith atom in the tth intermediate conformation is defined as,

$$\mathbf{p_i}(t) = (1-t)\mathbf{p_i}(0) + t\mathbf{p_i}(1) \tag{5.1}$$

The LST method, originally proposed by Halgren et al. [93], has been employed alone or in combination with more sophisticate methods [180, 20, 21] for generating good initial guess of reaction paths when studying chemical reactions. The method can also be applied to more complex systems such as proteins, however, some special care should be taken to avoid the default quadratic complexity of the method. The idea behind LST is to generate intermediate structures, by linearly interpolating the distances between any atom pair in the initial and target structures. Because this condition cannot be satisfied for all the atom pairs, the atom positions $\mathbf{p_i}(t)$ in the tth intermediate conformation are solved by minimizing the following energy formula:

$$E(t) = \sum_{i>j} \frac{\left(r_{ij}(t) - \bar{r}_{ij}(t)\right)^2}{\left(\bar{r}_{ij}\right)^4} + \beta \sum_i \|\mathbf{p_i}(t) - \bar{\mathbf{p}}_\mathbf{i}(t)\| \tag{5.2}$$

where $\bar{\mathbf{p}}_\mathbf{i}(t)$ is the interpolated atom position calculated as from Equation 5.1. $r_{ij}(t) = \|\mathbf{p_i}(t) - \mathbf{p_j}(t)\|$ is the distance between the ith and jth atoms in the intermediate conformation and $\bar{r}_{ij}(t)$ is the prescribed distance between the ith and jth atoms and calculated by linear interpolation, i.e. $\bar{r}_{ij} = (1-t)\|\mathbf{p_i}(0) - \mathbf{p_j}(0)\| + t\|\mathbf{p_i}(1) - \mathbf{p_j}(1)\|$.

The role of the second term in Equation 5.2 is to suppress the translational and rotational variations. Hence β is typically taken small enough so that this second term is much smaller than the first one.

In our implementation of the LST method, to locate each intermediate structure, we start from a structure whose atom positions are linearly interpolated between the initial and final structure, i.e. $\overline{\mathbf{p}}_\mathbf{i}(t)$. Then, the FIRE optimizer is applied to minimize this structure for a number of iteration n_{LST}. Finally, since considering all the pairs of atoms would incur a quadratic cost, we considered only the atom pairs which are covalently bonded.

Path reparation

This step aims to remove local steric clashes and ring clashes (covalent bonds crossing aromatic-ring surfaces) to improve the input for the path-optimization step. Moreover, path optimization may not remove the ring clashes, from our experience.

A steric clash is detected whenever the distance between any pair of non-bonded atoms is smaller than a predefined threshold d_{steric}. To remove this clash type, we optimize a system of springs between these atoms to push them apart. Besides, extra springs are also established for the bonds involving these clashing atoms to preserve these bond lengths. An example of this model is shown in Figure 5.3.

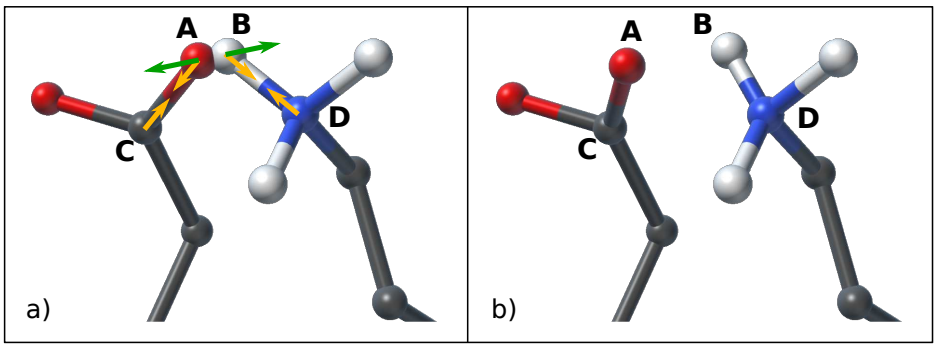


Fig. 5.3 An example of a steric-clash removal: a) Two clashing atoms are: a hydrogen (white ball B) and an oxygen (red ball A). The spring forces for the steric-clash removal are applied on clashing atoms (green arrows) and atoms bonded to them (orange arrows). b) The result after applying steric-clash removal.

The spring force $\mathbf{F_C}$ applied on the atom C which is bonded to the clashing atom A is defined as

$$\mathbf{F_C} = k_{bonded}(\|\mathbf{p_A} - \mathbf{p_C}\| - d^0_{CA})^2 \frac{\mathbf{p_A} - \mathbf{p_C}}{\|\mathbf{p_A} - \mathbf{p_C}\|}$$

where k_{bonded} is the spring constant for bonded atoms and d^0_{CA} is the initial bond length between the atoms A and C. $\mathbf{p_A}$ and $\mathbf{p_C}$ are the current positions of the atoms A and C.

The force applied on the clashing atom A is $\mathbf{F_A} = -\mathbf{F_C} + \mathbf{F_{AB}}$, i.e. the sum of the opposite force to $\mathbf{F_C}$ and the spring force from the interaction with its clashing atom B, $\mathbf{F_{AB}}$, defined as,

$$\mathbf{F_{AB}} = k_{steric}(\|\mathbf{p_B} - \mathbf{p_A}\| - d_{steric})^2 \frac{\mathbf{p_B} - \mathbf{p_A}}{\|\mathbf{p_B} - \mathbf{p_A}\|} \tag{5.3}$$

where k_{steric} is the force constant for steric clashing atoms and $\mathbf{p_B}$ is the current position of atom B.

The forces applied on atom B and D can be derived similarly as for atoms A and C, respectively.

For ring clashes, we first detect all the aromatic rings in the structures. This is simple because the standard amino acids in proteins which have aromatic rings are known such as histidine, proline, phenylalanine, tyrosine, and tryptophan. Then, a ring clash is detected if any bond cuts through a ring surface. For each ring, we first locate its center of mass made by all of the ring atoms. The ring surface is decomposed into triangular surfaces, each of which is defined by the center of mass and a ring bond. Then, we geometrically check whether a bond nearby cuts any of the triangular surfaces. To remove a ring clash, external forces are applied to push each atoms of the clashing bond outside the ring. The pushing direction is the vector $\mathbf{r_{ring}}$ pointing from the ring center of mass to the cut-point position (the intersection of the clashing bond and any triangular surface of the ring). Springs are also used for bonded atoms in the rings to maintain the ring shape. An illustration of the ring-clash detection and removal is shown in Figure 5.4.

Suppose A and B are the atoms of the clashing bond of a ring clash. The spring force applied on atom A is defined as,

$$\mathbf{F_A} = k_{ring}(\|\mathbf{p_A} - \mathbf{r_c}\| - d_{ring})^2 \frac{\mathbf{r_{ring}}}{\|\mathbf{r_{ring}}\|} \tag{5.4}$$

where k_{ring} is the force constant for clashing-ring atoms. $\mathbf{p_A}$ is the current position of atom A, $\mathbf{r_c}$ is the current ring center, and d_{ring} is the minimum distance along $\mathbf{r_{ring}}$ for resolving ring clashes. The force $\mathbf{F_B}$ applied on atom B can be expressed in a similar manner.

The forces of a ring atom is the sum of the spring forces from the interaction with its two neighbor (bonded) atoms. For example the force $\mathbf{F_{R_2}}$ applying on the ring atom R_2 is

$$\mathbf{F_{R_2}} = \mathbf{F_{R_2R_1}} + \mathbf{F_{R_2R_3}} \tag{5.5}$$

where $\mathbf{F_{R_2R_1}}$ and $\mathbf{F_{R_2R_3}}$ are the spring forces among bonded atom, and hence, defined as,

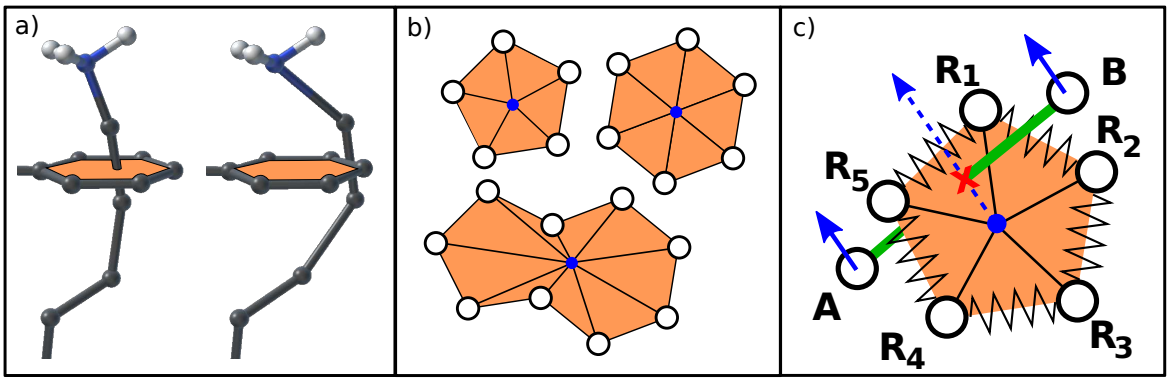


Fig. 5.4 Example of a ring-clash removal: a) A ring clash occurs when a bond passes through an aromatic-ring surface (left) and the result after applying ring-clash removal. b) Three types of rings in 20 standard amino acids and their triangular-surface decomposition. White balls represent atoms and small blue circles represents the center of mass of a ring or ring compound. c) The cutting point (red cross) between the clashing bond and one of the triangular surfaces is determined. Then, the force direction (blue dotted arrow) for clash removal is determined. Finally, the clash removing forces (blue solid arrow) are applied on the two atoms of the clashing bonds. The springs among bonded atoms of the ring, which are presented by zigzag lines, are there to maintain the ring shape during the escape of the clashing bond.

$$\mathbf{F_{R_2R_1}} = k_{bonded}(\|\mathbf{p_{R_1}} - \mathbf{p_{R_2}}\| - d^0_{R_1R_2})^2 \frac{\mathbf{p_{R_1}} - \mathbf{p_{R_2}}}{\|\mathbf{p_{R_1}} - \mathbf{p_{R_2}}\|}$$
$$\mathbf{F_{R_2R_3}} = k_{bonded}(\|\mathbf{p_{R_3}} - \mathbf{p_{R_2}}\| - d^0_{R_3R_2})^2 \frac{\mathbf{p_{R_3}} - \mathbf{p_{R_2}}}{\|\mathbf{p_{R_3}} - \mathbf{p_{R_2}}\|}$$

where $d^0_{R_1R_2}$ and $d^0_{R_3R_2}$ are the initial lengths of the bonds between R_1 and R_2, and between R_3 and R_2, respectively. We use the same force constant k_{bonded} for bonded atoms in both steric-clash and ring-clash removals.

Path optimization

Any path optimization method can be used for this step. In our case, we used the NEB method which adjusts an initial path iteratively to obtain the MEP. We briefly present below the NEB process. More details can be found in [120].

Let $\mathbf{R_i}$ $(i \in [0, \mathcal{L}-1])$ denote the ith conformation of a path of size $\mathcal{L}$. Then the NEB force applied on an intermediate conformation[1] $\mathbf{R_i}$ $(i \notin \{0, \mathcal{L}-1\})$ is defined as,

$$\mathbf{F_i^{NEB}} = \mathbf{F_i^{\perp}} + \mathbf{F_i^{\parallel}} \tag{5.6}$$

where $\mathbf{F_i^{\perp}}$ is the component of the potential force perpendicular to the tangent $\hat{\tau}_\mathbf{i}$ of the path, i.e.

$$\mathbf{F_i^{\perp}} = (-\nabla E(\mathbf{R_i})) - \langle -\nabla E(\mathbf{R_i}), \hat{\tau}_\mathbf{i} \rangle \, \hat{\tau}_\mathbf{i} \tag{5.7}$$

and $\mathbf{F_i^{\parallel}}$ is the component of the spring force $\mathbf{F_i^s}$ parallel to the tangent, i.e.

$$\mathbf{F_i^{\parallel}} = \langle \mathbf{F_i^s}, \hat{\tau}_\mathbf{i} \rangle \, \hat{\tau}_\mathbf{i} \tag{5.8}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the dot product of two vectors. The tangent can be simply defined as,

$$\tau_\mathbf{i} = \frac{\mathbf{R_{i+1}} - \mathbf{R_i}}{|\mathbf{R_{i+1}} - \mathbf{R_i}|} + \frac{\mathbf{R_i} - \mathbf{R_{i-1}}}{|\mathbf{R_i} - \mathbf{R_{i-1}}|} \tag{5.9}$$

and $\hat{\tau}_\mathbf{i} = \frac{\tau_\mathbf{i}}{|\tau_\mathbf{i}|}$, which is the unit vector bisecting the angle made by $\mathbf{R_{i+1}} - \mathbf{R_i}$ and $\mathbf{R_i} - \mathbf{R_{i-1}}$. Finally, the spring force is defined as,

$$\mathbf{F_i^s} = k_{i+1}(\mathbf{R_{i+1}} - \mathbf{R_i}) - k_i(\mathbf{R_i} - \mathbf{R_{i-1}}). \tag{5.10}$$

$\mathbf{F_i^{\perp}}$ drives the conformation $\mathbf{R_i}$ to the position where $\mathbf{F_i^{\perp}}$ vanishes (a property of MEPs) while $\mathbf{F_i^{\parallel}}$ prevents the conformations from clustering at certain local minima.

Since Equation 5.9 may badly approximate the tangent, we actually used an additional spring force perpendicular to the path to compensate such approximation as proposed in [120]. Hence, the NEB force is modified as,

$$\mathbf{F_i^{NEB}} = \mathbf{F_i^{\perp}} + \mathbf{F_i^{\parallel}} + f(\phi_i)(\mathbf{F_i^s} - < \mathbf{F_i^s}, \hat{\tau}_\mathbf{i} > \hat{\tau}_\mathbf{i}) \tag{5.11}$$

where $f(\phi_i)$ controls the quantity of the additional spring force according to $\cos\phi_i = \frac{<\mathbf{R_{i+1}}-\mathbf{R_i}, \mathbf{R_i}-\mathbf{R_{i-1}}>}{|\mathbf{R_{i+1}}-\mathbf{R_i}||\mathbf{R_i}-\mathbf{R_{i-1}}|}$ and

$$f(\phi_i) = \frac{1}{2}(1 + \cos(\pi(\cos\phi_i))) \tag{5.12}$$

[1] These conformations are also called *images* in the reference papers.

With the NEB force clearly defined, the conformations on the path are optimized until certain condition is met. In our case, we use the FIRE method for moving the conformations and stop the process after a maximum number of updates n_{NEB} is reached.

5.2 Experimental Validation

5.2.1 Setup

We have applied this framework on the 12 proteins that were already used to validate the ARAP interpolation method (see Chapter 4). The purpose is to compare the performance of three path generation methods (ARAPi, Linear, LST) when combined with the NEB method.

A summary of the experiments is shown in Table 5.1. In each case, an initial and a target structure are obtained from the Protein Data Bank [22]. For input reparation, we use MODELLER [197] integrated in Chimera software [106] and SwissPDB software [91] for reconstructing missing residues and heavy atoms, respectively. Afterwards, we use the GROMOS43a1 force field parameters and the command *pdb2gmx* in Gromacs [146] to add hydrogen atoms. The topology output are then used for the energy valuation by Gromacs integrated in the SAMSON platform [113]. For energy minimization, we use the FIRE method because it has been shown to outperform several common methods and does not consume as much memory as the BFGS or L-BFGS methods [24]. Table 5.2 summarizes the tools that are used in the proposed framework.

For removing clashes, we apply the method presented earlier (also implemented as a module in the SAMSON platform), with the chosen parameters shown in Table 5.3. The clash removal is an adaptive process applied for the maximum of 200 steps with an initial step size of 0.005 fs. For each step, it constrains the atom displacement to not exceed 0.1

Table 5.1 Experiments and results for the energy-based enhancement of paths.

Experiment ID	Name	Initial/Target (pdb and chain code)	no. atoms	Distance d_m (Å)	Path size $\mathcal{L}$	NEB time (s)
1	5'-Nucleotidase	1HP1(A)/1HPU(C)	5123	32.19	97	534.5
2	Adenylate Kinase	4AKE(A)/1AKE(A)	2085	21.47	64	99.9
3	Alcohol Dehydrogenase	8ADH(A)/6ADH(A)	3516	12.31	37	119.1
4	Calmodulin	1CFD(A)/1CFC(A)	1459	13.41	40	39.2
5	Collagenase	1NQD(A)/1NQJ(B)	1257	38.70	116	112.1
6	Dengue 2 Virus Envelope Glycoprotein	1OAN(A)/1OK8(A)	3866	32.66	98	336.2
7	Dihydrofolate Reductase	1RX2(A)/1RX6(A)	1602	12.10	36	42.2
8	Diphtheria Toxin	1DDT(A)/1MDT(A)	5223	49.89	150	874.4
9	DNA Polymerase	1IH7(A)/1IG9(A)	9525	30.61	92	1153.2
10	Pyrophosphokinase	1HKA(A)/1Q0N(A)	1597	24.79	74	83.7
11	Pyruvate Phosphate Dikinase	1KBL(A)/2R82(A)	8541	47.85	144	1619.7
12	Spindle Assembly Checkpoint Protein	1DUJ(A)/1KLQ(A)	1934	37.54	113	174.3

Input processing	
Input reparation	Fixing missing residues: MODELLER integrated in Chimera Fixing missing heavy atoms: SwissPDB software Adding hydrogen atoms: Gromacs software (*pdb2gmx*) with GROMOS43a1 parameters
Input preparation	Handling mutation: ARAPi with no intermediate conformations Minimizing structures: Gromacs integrated in SAMSON with the FIRE method
Path processing	
Path generation	ARAPi, Linear or LST
Path reparation	Steric and ring clash removal module in SAMSON
Path optimization	NEB with the FIRE method

Table 5.2 Summary of the tools used in the proposed framework for generating energy-based optimized paths.

Table 5.3 Parameters for the clash removal.

Parameters	Value
k_{steric}	5 N.nm^{-2}
d_{steric}	0.11 nm
k_{ring}	10 N.nm^{-2}
d_{ring}	0.3 nm
k_{bonded}	0.05 N.nm^{-2}

Å. In addition, if the new-state energy is greater than the last-state energy, the new state is rejected and the step size is reduced by half. Contrarily, if the new-state energy is smaller than the last-state energy, the new state is accepted and the step size is increased by 1.2. The process is stopped as soon as all the steric and ring clashes are resolved. This adaptive process ensures that the minimization always goes downhill.

The number of conformations of a path is defined based on d_m, the maximum displacement of all the atoms between the initial and target structures, i.e.

$$d_m = \max_i(|\mathbf{p_i}(0) - \mathbf{p_i}(1)|) \tag{5.13}$$

We chose to have 3 conformations/Å, and hence $\mathcal{L}$, the total number of conformations along a path (including the initial and target states) is equal to:

$$\mathcal{L} = [\frac{d_m}{d_0}] \tag{5.14}$$

where $[\cdot]$ is the rounding operator which rounds a value to their nearest integer value and $d_0 = \frac{1}{3}$Å. The values of d_m and $\mathcal{L}$ for the experiments are also shown in Table 5.1.

In the LST method, we use $\beta = 10^{-6}$ as in [93] and $n_{LST} = 1000$. The NEB method is implemented as a multi-core module on the SAMSON platform by one of our colleagues.

We use the uniform spring constants , i.e. $k_{i+1} = k_i = 1000$ eV.Å^{-2} for $\forall i$, and the number of iterations $n_{NEB} = 1000$. We found that these parameters give stable and converging results.

5.2.2 Results

Processing Time

The total processing time of each experiment is shown in Figure 5.5. This value includes the time from path generation, path reparation, and path optimization, using either the ARAPi, Linear, or LST method. The figure shows that, in all the experiments, the paths generated from the ARAPi method consume least time while those generated by the LST method consume most time, although the differences are only on the order of 10%.

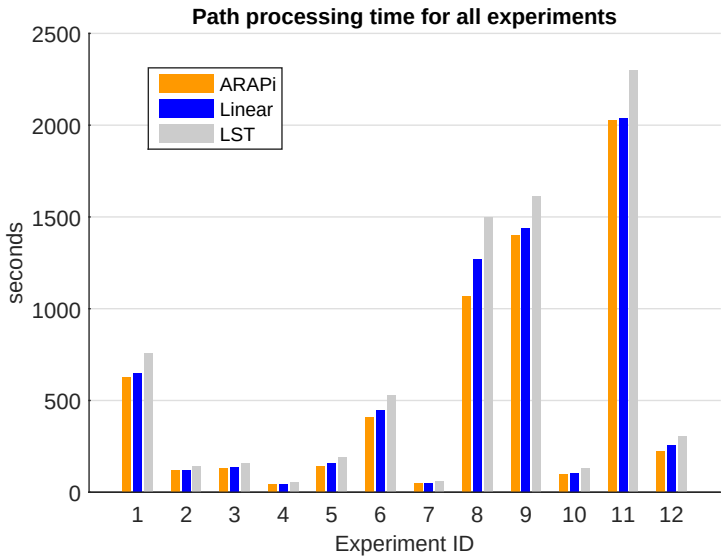


Fig. 5.5 Path-processing time of each experiment for the paths generated with the ARAPi, Linear, and the LST method.

Now, let us consider the path-generation time per conformation per atom for each experiment. As seen from Figure 5.6, this time is almost constant for all the path-generation methods, because the algorithmic complexity is almost linear for the chosen methods[2]. The Linear method gives results in the shortest time, followed very closely by ARAPi. The LST method is 7 to 8 times slower than the ARAPi and Linear methods. When averaging over all the experiments, the time per conformation per atom for the ARAPi, the Linear, and LST methods are 0.0323 ms, 0.0295 ms and 0.2396 ms respectively. The reason why the LST

[2]Our implementation of LST has approximately linear complexity because we consider only covalently bonded atoms as the atom pairs.

method do not perform as well as the other methods is the requirement of an iterative solver. Here, we chose the number of iterations quite high, $n_{LST} = 1000$, to get conformations close to the converged conformations with LST. However, we will see later that the paths obtained from the LST method still do not lead to good-quality optimized paths after the NEB method is applied.

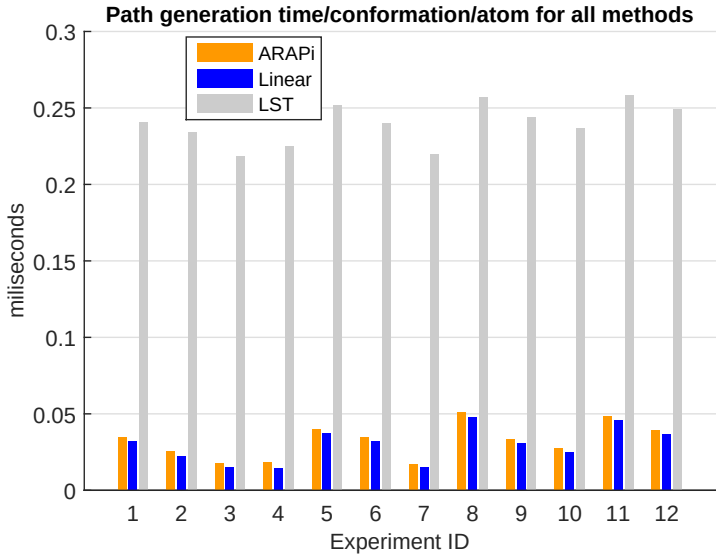


Fig. 5.6 Path-generation time per conformation per atom for all the experiments for the paths generated with the ARAPi, Linear and LST method.

The last column in Table 5.1 shows the path-optimization time for each experiment. Each value in this column is the average time value from all the path-generation methods because the optimization time is similar regardless of the path-generation methods[3].

Number of clashes

Figure 5.7 shows for each experiment the total number of steric and ring clashes detected for the paths from different generation methods by summing up the clashes in each intermediate conformation. Since the number of steric clashes can strongly vary (between 105 to 64035) these plots show the (base 10) logarithm values of the total number of clashes augmented by 1 to avoid the undefined logarithm value.

As one can see, the ARAPi method leads to fewer steric clashes and more ring clashes than the other methods. It is obvious that ARAPi is less prone to steric clashes because it better preserves the local rigidity of the original structure (bond lengths and bond angles). However, it is more prone to ring clashes because the method does not have any restraint

[3] Also, because the number of NEB iteration is fixed and the path size is the same for each experiment.

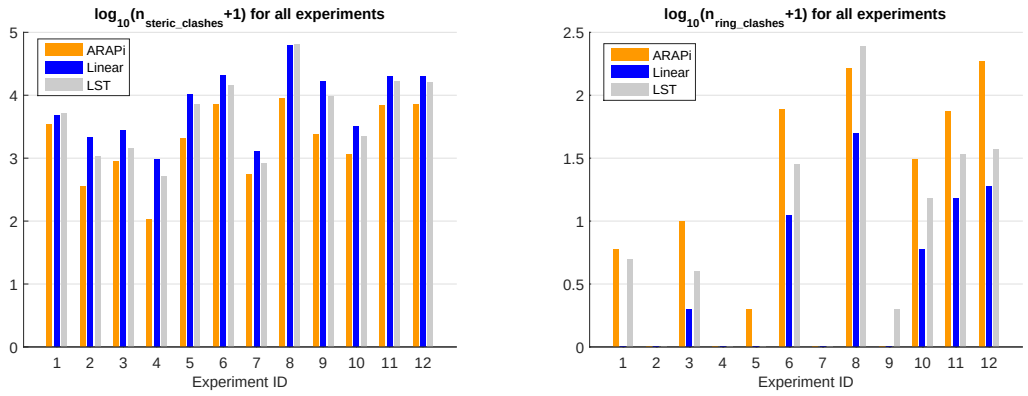


Fig. 5.7 Number of steric clashes (left) and ring clashes (right) for all the experiments.

regarding this problem. In fact, as will be shown later, because the Linear and LST methods tend to have some part of the structure shrunk along the path, the ring surface areas are reduced, which lower the possibility of ring clashes.

Thanks to our clash removal method, all the steric and ring clashes can be removed totally. Figure 5.8 shows the time for removing both types of clashes in all the experiments. It shows that the clash removal took the least time for the ARAPi paths in all the experiments. This is because the ARAPi paths have the least number of steric clashes among the presented path-generation methods.

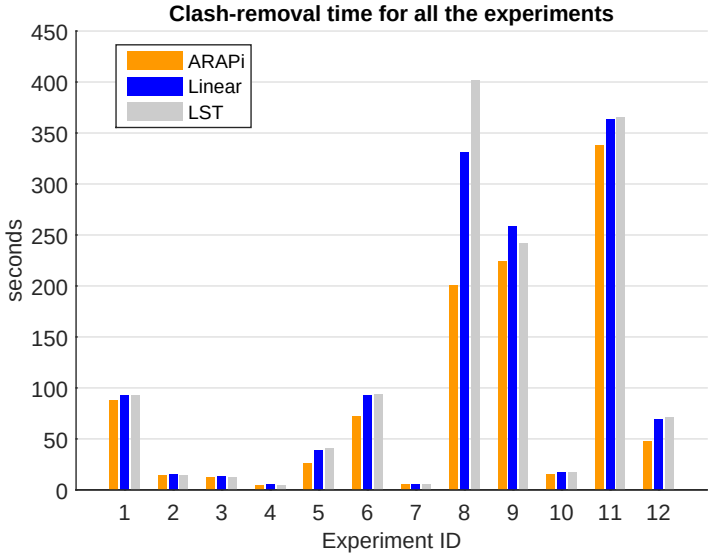


Fig. 5.8 Clash-removal time for all the experiments.

Reduction of energy barrier

To assess a path quality, we compute the potential energy barrier $\bar{E}$ of each path. This quantity is computed as the difference between the maximum energy of the path conformations and the energy of the initial conformation of a path, i.e.

$$\bar{E} = \max_{i \in [0, \mathcal{L}-1]} E(\mathbf{R_i}) - E(\mathbf{R_0}) \tag{5.15}$$

In our framework, the energy barrier is reduced twice: after the clash removal and after the path optimization. Hence, for each experiment with each path-generation method, we computed two reduction factors, $f_{gen/clash}$ and $f_{clash/neb}$, defined as,

$$f_{gen/clash} = log_{10}(\frac{\bar{E}_{gen}}{\bar{E}_{clash}}) \tag{5.16}$$

$$f_{clash/neb} = log_{10}(\frac{\bar{E}_{clash}}{\bar{E}_{neb}}) \tag{5.17}$$

For each experiment and each path generation method, $\bar{E}_{gen}$ is the path energy barrier after the path generation, $\bar{E}_{clash}$ is the path energy barrier after the application of the clash remover, and $\bar{E}_{neb}$ is the path energy barrier after the application of the NEB method.

Hence, $f_{gen/clash}$ is the reduction factor in potential energy barrier of a path after clash removal compared with the one before clash removal and $f_{clash/neb}$ is the reduction factor in potential energy barrier of a path after applying the NEB method and the one before applying the NEB method.

Figure 5.9 shows the values of $f_{gen/clash}$ for all the experiments with all the path-generation methods. It shows an important energy reduction ($10^{2.3}$-$10^{14.3}$ times) thanks to the clash remover.

Figure 5.10 shows the values of $f_{clash/neb}$ for all the experiments with all the path-generation methods. It shows further energy reduction ($10^{2.7}$-$10^{4.8}$ times) thanks to the NEB method.

The final potential energy barriers of the optimized paths for all the experiments are shown in Figure 5.11. As seen from the figure, the optimized paths from by the ARAPi method have the lowest energy barrier for all the experiment. The LST paths have lower energy barriers than the Linear paths in only 2 experiments (Experiment 5 and 8). It shows that although the LST method was applied for 1000 iterations, which were very time-consuming, its paths still do not yield lower energy barriers after the NEB method is applied than the paths from the two other methods.

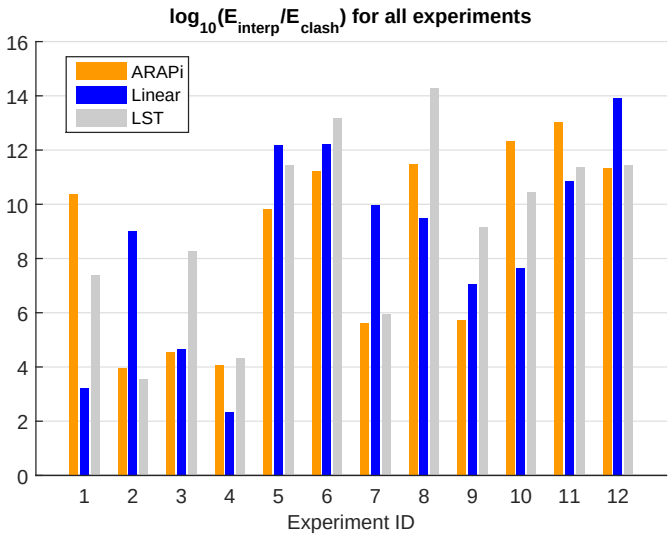


Fig. 5.9 Energy reduction thanks to clash removal for each experiment.

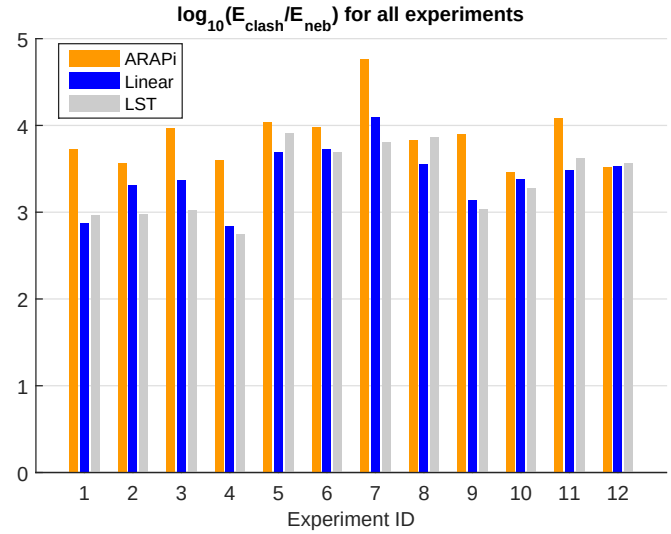


Fig. 5.10 Energy reduction thanks to the NEB method for each experiment.

These results show the efficiency of the NEB method for path optimization, the important role of the clash removal method, and the role of the path-generation method on the final energy barriers of the optimized paths. It is also important to emphasize that the NEB method alone was not able to resolve the ring clashes in the author's experience.

Visual inspection of the optimized paths

In general, the motions obtained with the ARAPi method are subject to less structural degeneration than those obtained with the Linear or LST methods. Here, we consider only

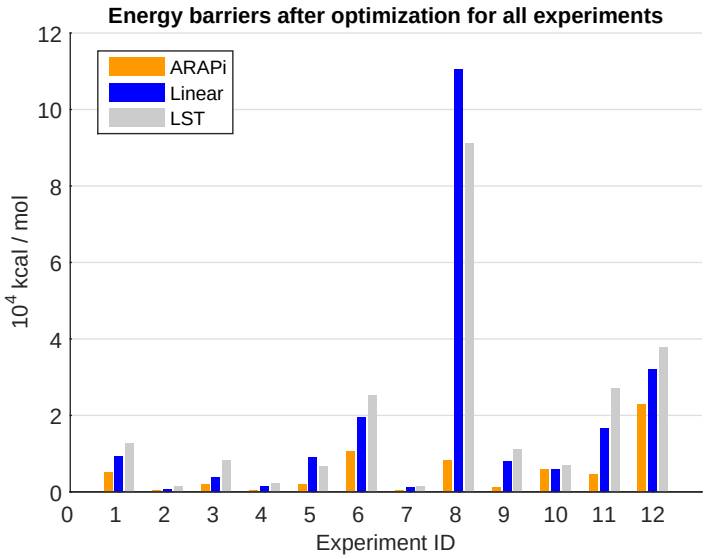


Fig. 5.11 Final potential energy barriers of the optimized paths for each experiment.

as example the case of the Diphtheria Toxin shown in Figure 5.12. This figure also shows the motions of Diphtheria Toxin after clash removals and optimization. Actually, the visual changes are noticeable mostly due to the path-optimization with the NEB method, since the clash remover only resolves local steric and ring clashes, and hence, these clashes did not affect the secondary structure in our case.

As seen from the figure, significant changes are observed in the protein structures of the paths after the path optimization, especially for those generated from the Linear and LST methods. In contrast, the ARAPi path and its optimized ones do not differ greatly because the ARAPi path is already very close to its optimized one. This behavior has also been observed for the rest of the experiments (not shown). A closer look at Figure 5.12 reveals that the red-and-yellow domain is shrunk at some moment along the optimized Linear and optimized LST paths while the optimized ARAPi path does not show this phenomenon. We found that the conformations the most shrunk were the ones with the highest energy along the paths (not shown). This explains why the optimized Linear and optimized LST paths have higher potential-energy barriers than the optimized ARAPi path.

Although the optimized ARAPi paths tend to have lower energy barriers, self-intersections along the paths were detected for the three following systems: the 5'-Nucleotidase (Figure 5.13), the Dengue 2 Virus Envelope Glycoprotein (Figure 5.14), and the Spindle Assembly Checkpoint protein (Figure 5.15). This problem has two causes. First, the ARAPi method only preserves the local rigidity, and hence, does not guarantee the absence of self-intersections between two distant parts of the same structure. Second, the Slerp method used

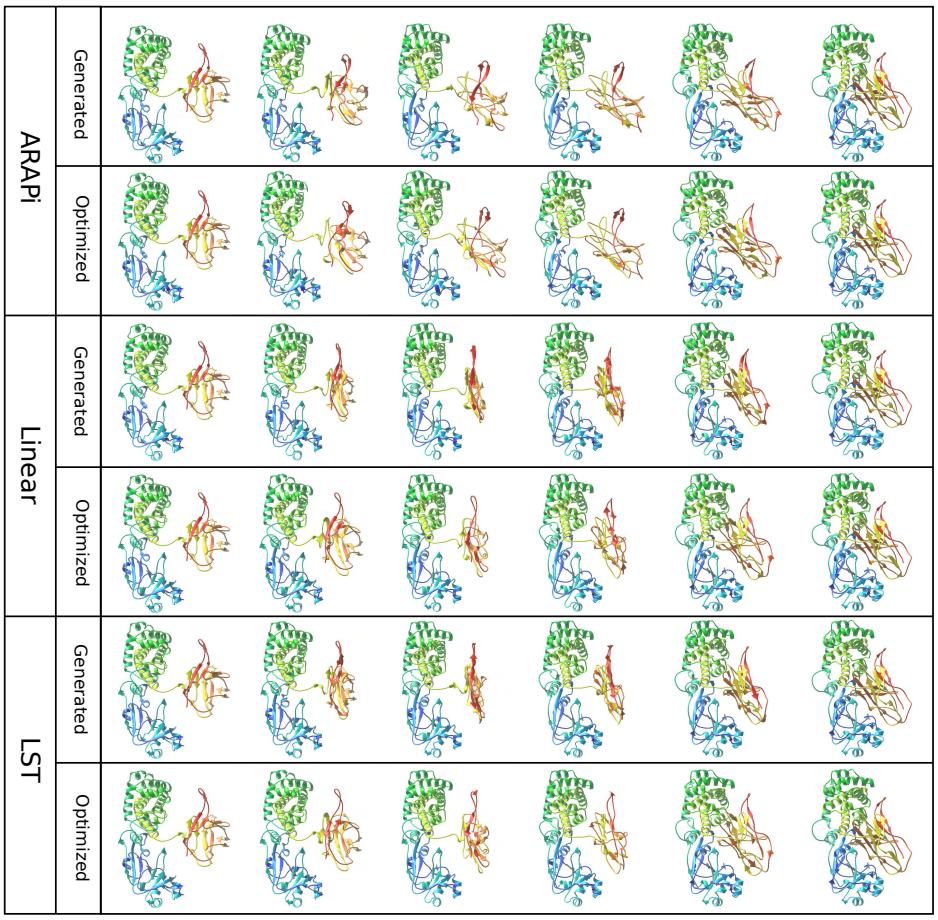


Fig. 5.12 The paths after path generation and optimization from three path-generation methods (ARAPi, Linear and LST) for Diphtheria Toxin. The paths after clash removal are not shown because clash removal does not strongly alter secondary structures. The ARAPi paths does not show strong differences between the generated one and optimized one. This result shows that the ARAPi method can generates paths close to the optimized ones. Visible structure degeneration is found in the Linear and LST paths after the path generation. Thanks to the path optimization method, this problem is reduced as seen in the optimized paths generated from the Linear and LST methods. However, the optimization cannot entirely remove the shrinkage problem, which is the source of high potential-energy barrier in the optimized Linear and LST paths.

for rotation interpolation is limited to the maximum rotation angle of 180 degrees (see the limitation of the method discussed in Section 4.3). In fact, the self-intersection problem is challenging to any deterministic path-generation method. The Linear and LST paths also encounter this problem for Dengue 2 Virus Envelope Glycoprotein, and Spindle Assembly Checkpoint protein as shown in Figure 5.14 and 5.15, respectively. Although, the shrinkage

in the Linear and LST paths of 5'-Nucleotidase helps to avoid this problem (see Figure 5.13), it raises the potential energy barrier as shown above for Diphtheria Toxin.

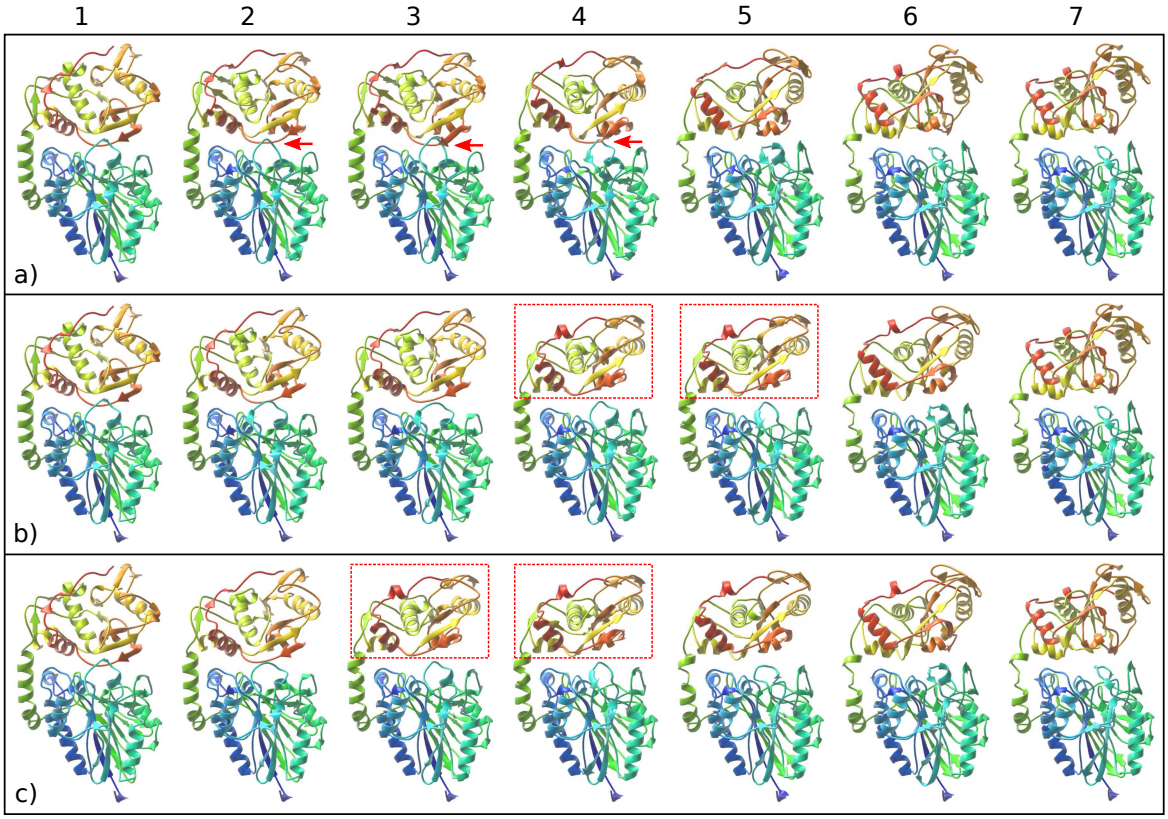


Fig. 5.13 Optimized paths for 5'-Nucleotidase from the a) ARAPi method b) Linear method c) LST method. Self-intersections (pointed by red arrows) are found for the optimized-ARAPi path (sequence a). The optimized Linear and LST paths do not have this problem but they have the shrinkage problem (for e.g. the protein parts in the red dotted rectangular boxes in sequences b and c are smaller in size than the same parts in the rest of the snapshots).

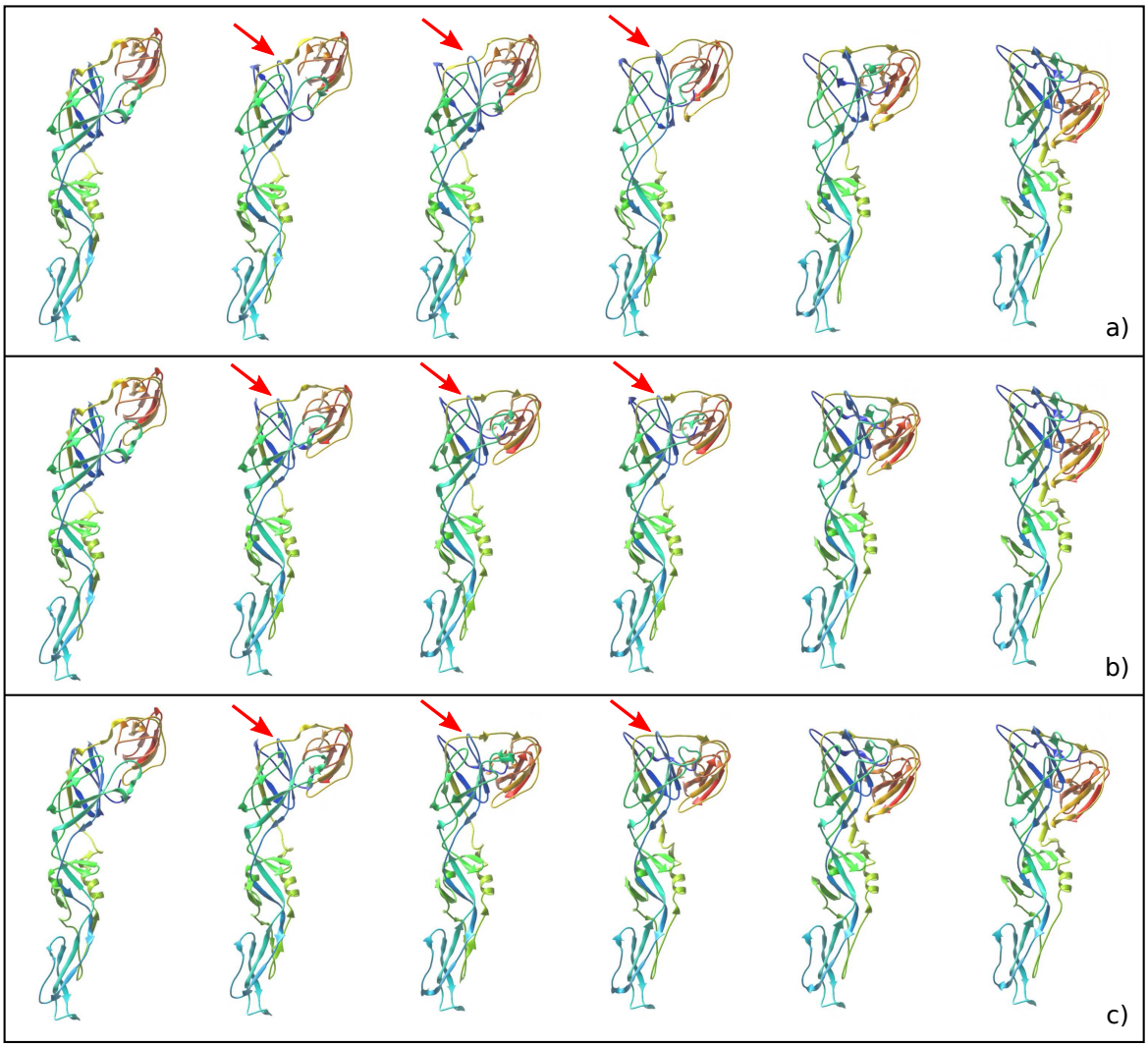


Fig. 5.14 Optimized paths for the Dengue 2 Virus Envelope Glycoprotein from a) the ARAPi method b) Linear method c) LST method. The self-intersection problem is found in the optimized paths from all the path-generation methods, as pointed by the red arrows. For each path (each sequence), the yellow loop is in front of the blue loop in the second snapshot; it then enters the blue loop in the third snapshot; and finally escapes behind the blue loop in the fourth snapshot.

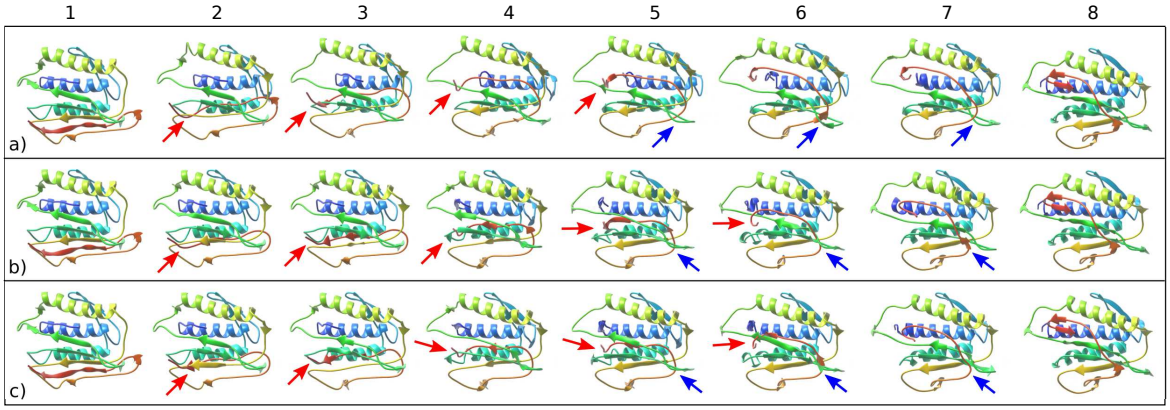


Fig. 5.15 Optimized paths for Spindle Assembly Checkpoint protein from the a) ARAPi method b) Linear method c) LST method. The self-intersection problem is found in the optimized paths from all the path-generation methods, as pointed by the red and blue arrows. In sequence (a), the red-end loop crosses the yellow loop (transition from a2 to a3), the cyan loop (a3 to a4), and the green loop (a4 to a5). The blue arrows in sequence (a) point to another location of self-intersection: the orange loop enters (a5 to a6) and then escapes (a6 to a7) the green loop. Similar self-intersection locations and behaviors are found for sequence b and c.

5.3 Conclusion and Discussion

We have proposed a framework for generating an optimized path from an initial structure and a target structure. It was applied on 12 experiments for generating optimized conformational transition pathways of proteins. With this framework, we assessed the quality of the solutions obtained after optimization with the NEB method from initial paths generated by three interpolation methods (ARAPi, Linear, and LST).

The results showed that the energy barriers of the optimized paths generated from the ARAPi method tend to be smaller. The initial paths generated with ARAPi are least prone to steric clash although this tendency is the opposite for ring clashes. However, our proposed clash removal method could remove effectively these clashes. The time for removing all clashes (steric and ring clashes) in the ARAPi paths are lower than in the Linear and LST paths. The clash removal also reduces the energy barriers of the paths significantly. The NEB method, as a path optimizer, reduces the path energy barriers further. The final optimized paths arising from the ARAPi method have the lowest energy barriers compared with those from the Linear and LST methods. However, the NEB method changes the path only locally in the case of ARAPi paths, i.e. the initial ARAPi paths are very close to their optimized ones. In contrast, the Linear and LST paths are very different from their optimized ones because they are more likely subjected to structure degeneration. Despite a huge reduction in energy barriers, the optimized paths arising from the Linear or LST method have higher energy barriers than those arising from ARAPi. This means that the initial paths have a great impact on the optimized solutions and the ARAPi appeared to be the best candidate for generating initial paths in this framework compared with the Linear and LST methods.

Obviously, the implementation of the proposed framework also has several limitations that are worth to be mentioned. First, the energy barriers can be missed because the paths obtained are discrete. There have been several proposals for finding the "true" saddle from an approximated one such as the Dimer method [99] or the climbing-image NEB [101] that could be applied in the future. Second, the ARAPi, Linear and LST methods only give one path among numerously possible paths. As we have seen, even though the ARAPi paths give lower energy barriers, self-intersections can still be present along the path, which render them physically invalid. This is a difficult problem that requires a more global exploration of the energy landscape to find a solution. The following part of the manuscript will present ART-RRT, a sampling-based method which combines the ARAP techniques for structure morphing and dimension reduction, with the RRT method from robotics for an effective exploration of these energy landscapes.

Part III

ART-RRT exploration of pathways for molecular systems

Chapter 6

ART-RRT method

This chapter presents the ART-RRT method, which combines ARAP with T-RRT [115], a variant of RRT, to search low-energy paths in high-dimensional energy landscapes. In ART-RRT, ARAP serves as a dimension-reduction and morphing method for generating large feasible motions of molecular systems through the manipulation of few control atoms, while T-RRT is used for efficiently exploring the conformational space. First, we will explain the principle of the method for the mono-directional variant of ART-RRT. Then, we present the specific challenges related to the bi-directional variant of ART-RRT and our proposal to address them.

6.1 Mono-directional ART-RRT

ART-RRT follows the same global scheme as RRT to explore the conformational space. Hence, mono-directional ART-RRT is based on Algorithm 4 (mono-directional RRT) and the related algorithm 5 (`ExtendBranch`) presented in Section 3.2. The specificity of ART-RRT will appear in the implementations of the function `RandomState` of mono-directional RRT and the functions `NearestState`, `Extend` and `TestState` inside `ExtendBranch`. The implementation details of these functions in ART-RRT are described below.

6.1.1 `RandomState`

In ART-RRT, to integrate the ARAP methods into the RRT framework, all the atoms in a model are classified as one of the three following types: active ARAP atoms (**A-atoms**), passive ARAP atoms (**P-atoms**) and non-ARAP atoms (**N-atoms**). As an example, in the application for ligand-unbinding problem from protein which will be shown in the next chapter, we assign A-atoms and P-atoms to ligand atoms, whereas N-atoms to protein atoms.

In `RandomState`, only A-atom positions are randomly sampled. Therefore, we define the generated target state at this stage as q_t^A which contains only the sampled positions of A-atoms. The positions for the other atom types will be determined by other procedures, which will be detailed below.

6.1.2 `NearestState`

Since only A-atoms are considered at the sampling stage, the metric used to compute the nearest state relies only on these atoms. More specifically, we compute the RMSD between the target state q_t^A and the nodes of the tree $\mathcal{T}$ considering only A-atoms.

6.1.3 `Extend`

The implementation of $\texttt{Extend}(q_n, q_t)$ in ART-RRT, which generates a new state q_{new} from the nearest node q_n toward the target state q_t^A, is shown in Algorithm 8.

The first three lines in the algorithm determines the positions for all the atom types in the new state. First, the A-atom positions in the new state are generated by the standard extension mechanism in RRT, i.e. by linear interpolation between q_n^A (q_n^A contains only the positions of A-atoms in the nearest node q_n) and q_t^A (line 1). Then, the P-atom positions in the new state are computed by ARAPm, using the A-atom positions in the new state q_{new}^A as constraints (line 2). Here, the A-atom positions q_n^A and P-atom positions q_n^P in the nearest node are used to define the initial shape for the ARAPm algorithm. The N-atom positions in the new state are assigned from their positions in the nearest node (line 3). At this stage, the new state q_{new}^0 can be constructed by gathering the atom positions from all the atom types (line 4). Finally at line 5, q_{new}^0 is relaxed using a constrained minimization (see below) to obtain the final new state q_{new}. Therefore, q_{new} tends to have the local rigidity preserved and low energy thanks to the ARAPm method and the constrained minimization, respectively.

Algorithm 8: $\texttt{Extend}(q_n, q_t^A)$ in ART-RRT.

Input : nearest node q_n and target state q_t^A.
Output : new state q_{new}.

1 $q_{new}^A \leftarrow \texttt{StandardExtend}(q_n^A, q_t^A)$;
2 $q_{new}^P \leftarrow \texttt{ARAPm}(q_{new}^A, q_n^A, q_n^P)$;
3 $q_{new}^N \leftarrow q_n^N$;
4 $q_{new}^0 \leftarrow \{q_{new}^A, q_{new}^P, q_{new}^N\}$;
5 $q_{new} \leftarrow \texttt{ConstrainedOptimize}(q_{new}^0)$;
6 **return** q_{new};

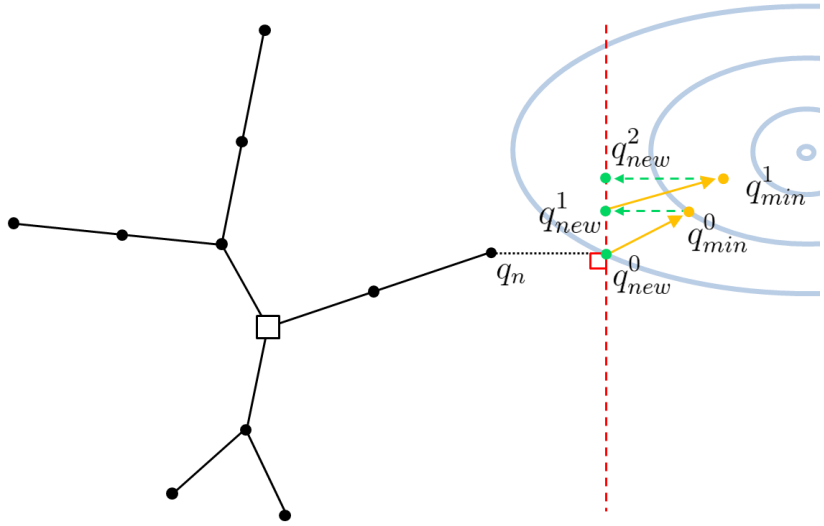


Fig. 6.1 Constrained minimization in ART-RRT. The concentric arcs represent iso-energy curves of a well on an energy landscape. The circle near the right border of the figure signifies the bottom of the well, which is also a local minimum on the energy landscape. q^0_{new} is the state generated by ARAPm. q^0_{min} is the first minimized state using FIRE method. q^1_{new} is the projected state of q^0_{min} onto the plane perpendicular to the direction made by q_n and q^0_{new}. This plane is represented by the dash vertical red line. q^1_{min} and q^2_{new} are the minimized and projected states, respectively for the second step.

For the constrained minimization, we use the FIRE method [24] to locally minimize q^0_{new}. Moreover, to ensure the tree extension and avoid backtracking to some previous states, the result of each minimization step is projected on the hyperplane orthogonal to the current expansion direction. This process is illustrated in Figure 6.1. The first minimization step is applied on q^0_{new} to give q^0_{min}. This state is then projected on the hyperplane perpendicular to the direction made by q_n and q^0_{new}, giving the projected state q^1_{new}. By repeating the minimization and projection step, one obtain q^1_{min} and q^2_{new}, respectively. In ART-RRT, the constrained minimization are performed n_{CO} times for each new node to give a final new state q_{new} which is then subjected to the `TestState` test described below.

6.1.4 `TestState`

The `TestState` function is used for checking the validity of a new state before adding it to the tree. Here, we use a transition test based on the relative local energies as in T-RRT, a RRT variant.

Originally, RRT was used in applications where new states were considered valid if they were not colliding with the environment. The transition test in T-RRT allows to extend RRT to any general cost space, i.e. any space where a cost can be associated to each state. T-RRT

is suitable for biological applications [115, 66] because molecular energy can be assigned as the natural cost for a given state. The transition test in T-RRT is based on the Metropolis criterion [165] and combined with an adaptive mechanism, which allows the tree to explore more in low-energy regions and to quickly escape high-energy barriers. In ART-RRT, the same transition test is used, and hence, a new state is accepted with a transition probability p defined as:

$$p = \begin{cases} \exp(-\frac{\Delta E}{kT}) & \text{if } \Delta E > 0, \\ 1 & \text{otherwise,} \end{cases} \tag{6.1}$$

where $\Delta E = E_{new} - E_n$ is the energy difference between the state energy E_{new} of the new state q_{new} and the energy E_n of the node q_n. k is the Boltzmann constant. T is a temperature factor that does not necessarily carry any physical meaning and is considered as a parameter of the algorithm. The use of the temperature parameter is well known in a variety of molecular simulations such as Monte Carlo simulation [23] where the temperature is kept constant or Simulated Annealing [132] where it is changed during the simulation. In ART-RRT, as in T-RRT, the temperature parameter is adaptively changed for controlling the difficulty of the transition test.

At the beginning, T is set to a low value to only permit the tree expansion on very easy positive slopes (in addition to flat and negative ones). Then, during the exploration, if the number of consecutive state rejections reaches a maximum value $\mathcal{S}$ (for Severity), the temperature increases by a factor λ to ease the following transition test. In contrast, for each state acceptance, the temperature decreases by the same factor λ ($\lambda > 1$), thus making the following tests more severe. Hence, T is automatically regulated along the exploration, balancing the search between unexplored regions and low-energy regions.

6.2 Bi-directional ART-RRT

Similar to bi-directional RRT, bi-directional ART-RRT also uses the `GrowTrees` function for growing two trees. This function includes `ExtendBranch` for extending a branch from one tree, and `ConnectBranch` for connecting two trees. In bi-directional ART-RRT, the `ExtendBranch` and `ConnectBranch` functions also have the same routine shown in Algorithm 5. In particular, the version of the `Extend` function called inside `ExtendBranch` is the one described in Algorithm 8 for mono-directional ART-RRT. However, some other modifications are necessary to realize bi-directional ART-RRT, mainly due to the following reasons.

- First, the `ExtendBranch` and `ConnectBranch` functions cannot work in the same manner in bi-directional ART-RRT, as in bi-directional RRT. This is because the use of the `ExtendBranch` function from mono-directional ART-RRT for connecting the trees would interpolate only the A-atom positions, and hence, cannot lead to the connection of two trees. Figure 6.2 shows an example where the A-atom positions of two nodes (each one in each tree) would match whereas the rest of the atom positions would not because the use of ARAPm does not guarantee the matching of P-atom positions. One possible solution is to use the linear interpolation method in place of ARAPm when connecting two trees. However, we propose here another approach which uses ARAPi to connect two trees, which will be detailed below. The use of ARAPi gives more realistic connection paths because the conformations of the path have their local rigidity preserved as much as possible as shown in Chapter 4.
- Second, it would not be convenient to rely on the transition test described in Section 6.1.4 to check the validity of the new states when connecting the trees. For example, let us consider a connecting branch where the node at each end of this branch comes from each tree. Suppose that when going from one end to the other of the branch, the energy of the nodes is in increasing order. Apparently, the transition test based on the Metropolis criterion would likely reject most of the nodes in this branch. However, the same branch, when going from the other end in the reverse direction, will have their node energies in decreasing order; and hence, the same transition test would immediately accept all of the branch nodes. Hence, for bi-directional ART-RRT, we will propose another condition for accepting the states during the branch connection.

In summary, to realize bi-directional ART-RRT, we use the `GrowTrees` function described in Algorithm 7 for growing the trees. The routines of `ExtendBranch` and `ConnectBranch` are the same and described in Algorithm 5. The implementations of `NearestState`, `Extend`, and `TestState` used in `ExtendBranch` are the ones from mono-directional ART-RRT and described in Section 6.1. In contrast, the implementations of `NearestState`, `Extend`, and `TestState` used in `ConnectBranch` are different and described below.

6.2.1 `NearestState` in `ConnectBranch`

When attempting to connect the trees in bi-directional ART-RRT, we search for the nearest node based on the RMSD considering both the A-atoms and P-atoms. P-atoms are also considered because if only the A-atoms are considered as in mono-directional ART-RRT (Section 6.1.2), one could wrongly interpret the trees as connected when the A-atoms from the nodes in both trees are matched whereas the P-atoms are not (c.f. Figure 6.2).

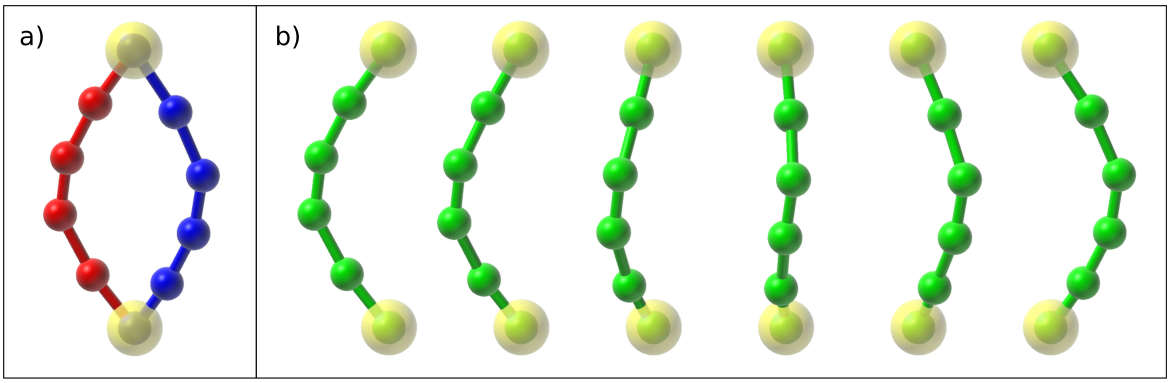


Fig. 6.2 a) The picture shows two states (each one from each tree) for a system of 6 atoms, where one state includes the red and the highlighted-yellow atoms and the other includes the blue and highlighted-yellow ones. The highlighted-yellow atoms are A-atoms whereas the rest are P-atoms. Hence, the picture shows a situation where both states have A-atom positions matched while the P-atom positions not matched. This situation could happen in bi-directional ART-RRT and if only ARAPm was used, the tree-growing process would stop prematurely because the RMSD distance, which considers only A-atoms, between these states would be zero. b) Considering also P-atoms for computing state distances when connecting the trees prevents this problem from happening. Moreover, the use of ARAPi method creates a smooth transition path between the states as shown in the picture.

6.2.2 `Extend` in `ConnectBranch`

During branch connections, we want to have an extension that also allows P-atoms to reach the target node. This is precisely what ARAPi allows to achieve. Hence, we use the `Extend` function as proposed in Algorithm 8 with some modifications. First, it takes in input the nearest state q_n, the A-atom positions q_t^A of the target state, but also the P-atom positions q_t^P of the target state. Note that unlike the branch-extension process, during the branch-connection process, the atom positions of all the atom types are known for the nearest node and the target node from the other tree; and hence, q_t^P is available. Second, instead of using ARAPm, ARAPi is used, i.e. line 2 in Algorithm 8 is replaced by the following line for `Extend` in `ConnectBranch`,

$$q_{new}^P \leftarrow \texttt{ARAPi}(q_t^A, q_t^P, q_n^A, q_n^P, t). \tag{6.2}$$

where q_t^A and q_t^P contain the A-atom positions and P-atom positions in the target state, respectively. Similarly, q_n^A and q_n^P contain the A-atom positions and P-atom positions in the nearest node, respectively. Finally, t is the interpolation instance computed as:

$$t = \begin{cases} \frac{\delta}{d} & \text{if } \delta < d, \\ 1 & \text{otherwise,} \end{cases} \tag{6.3}$$

where δ is the RRT extension step and d is the RMSD distance between the q_n and the q_t considering only the A-atom positions and P-atom positions. The positions of N-atoms are not considered because these atoms are not controlled by the ARAPi method.

6.2.3 `TestState` in `ConnectBranch`

As discussed above, it is not convenient to use the transition test based on the Metropolis criterion for the branch-connection process. Therefore, we propose an acceptance condition based on threshold energy for this process in bi-directional ART-RRT. A threshold energy E_{th} is computed based on the the energy of the nodes to connect, and defined as:

$$E_{threshold} = \gamma(E_{max} - E_{min}) + E_{min} \tag{6.4}$$

where E_{min} and E_{max} are the minimum and maximum energies of the two nodes to connect, respectively. $\gamma \geq 1$ is a threshold parameter for tuning the difficulty of the test. Hence, when $\gamma = 1$, the threshold is equal to the maximum potential energy of the two nodes, E_{max}. When $\gamma > 1$, the threshold energy is greater than E_{max}, which makes it easier to accept new states.

6.2.4 Global behavior

Finally, the global behavior of bi-directional ART-RRT is illustrated in Figure 6.3. It shows that the two trees are successively extended, through branch extensions driven by ARAPm and branch connections driven by ARAPi. This process allows to find effectively low-energy solutions thanks to the exploratory strength from T-RRT and the dimensionality reduction from ARAP mechanisms. We will now present the applications of the mono-directional and bi-directional variants of ART-RRT in the following chapters.

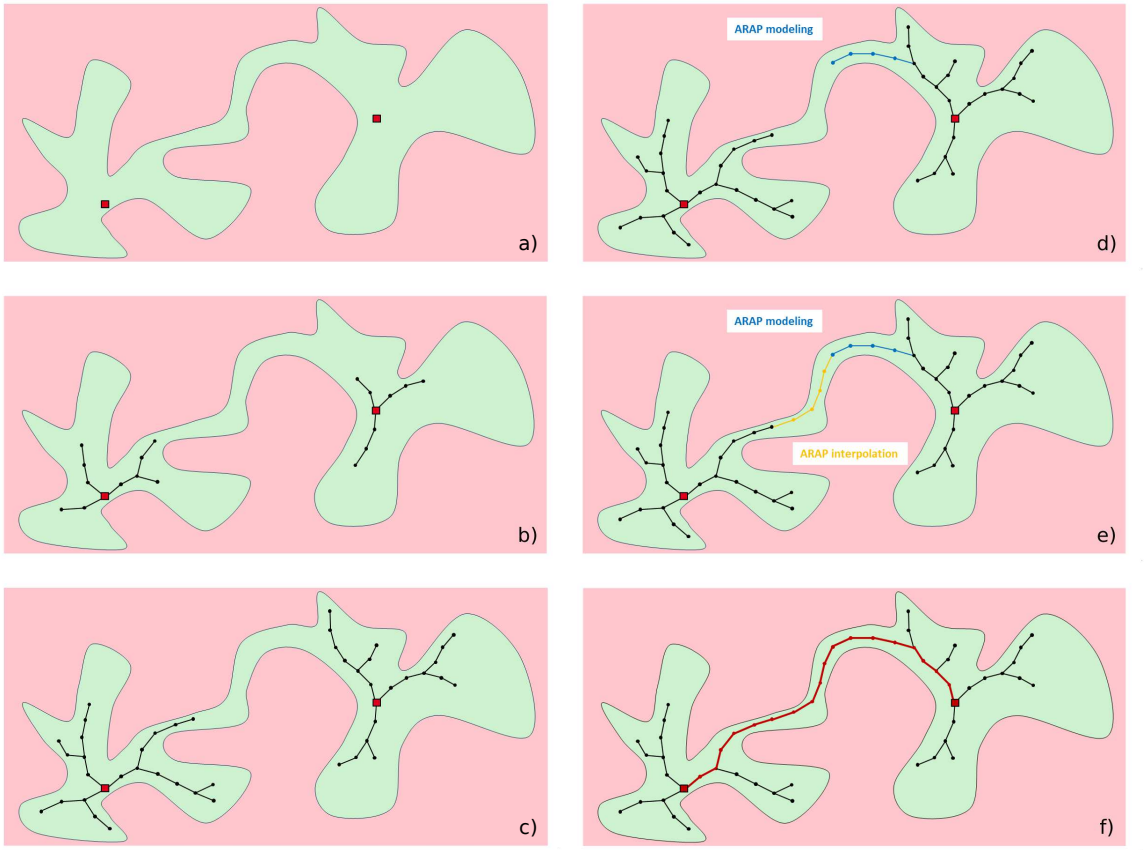


Fig. 6.3 Global behavior of bi-directional ART-RRT. Pink and green areas represent high-energy and low-energy regions, respectively. The red squares represent the start and goal states. a) The given start and goal states in low-energy regions. b) and c) two trees are built following low-energy regions. d) The branch extension driven by ARAPm grows new nodes in low-energy regions. e) The branch connection driven by ARAPi. f) A low-energy path is found thanks to the exploratory strength from RRT and the dimensionality reduction from the ARAP mechanisms.

Chapter 7

Applications of mono-directional ART-RRT

This chapter presents the applications of mono-directional ART-RRT[1] for exploring the dihedral-angle space of dialanine, and then for finding ligand-unbinding pathways from receptors.

7.1 Exploring the dihedral-angle space of dialanine

This section investigates the capabilities of mono-directional ART-RRT for exploring the Φ and Ψ dihedral angles of dialanine. Dialanine is a simple structure composed of two alanine peptides, with a well known energy landscape. Hence, it has been widely used to evaluate new methods (see e.g. [195, 115, 138, 67, 202]).

7.1.1 Benchmark

To prepare the system, hydrogen atoms were generated using the command *pdb2gmx* in GROMACS [1] with the force field parameters GROMOS43a1 [234, 57]. The topology output was then used to evaluate the energy from GROMACS integrated in the SAMSON platform [113].

Figure 7.1a shows the system at its lowest energy state where $\Phi = -118.1°$ and $\Psi = 142.3°$. Three carbon atoms are selected as A-atoms, where the position of one of them (A-atom 1 in the figure) is kept fixed while the other two (A-atoms 2 and 3) are mobile. Two different sampling volumes are designed for the mobile A-atoms. Both of them are cubes

[1] For convenience, mono-directional ART-RRT is shortened as ART-RRT in this chapter unless stated otherwise.

centered on the position of the A-atom 1. The edge size of the sampling volume for the A-atom 2 is 9 Å, and for the A-atom 3 is 14 Å. Figure 7.1b and c show the sampling volumes for the A-atoms 2 and 3, respectively.

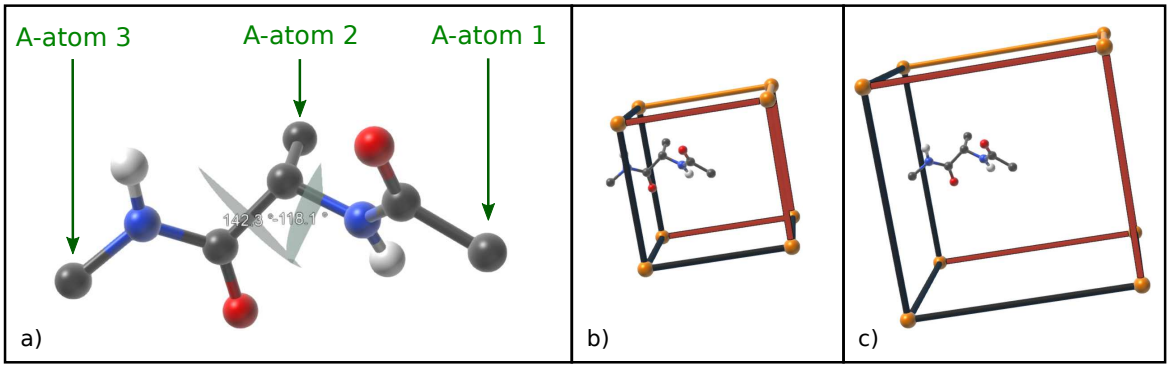


Fig. 7.1 Dialanine benchmark. a) The conformation of dialanine corresponding to the lowest minimum-energy state, where $\Phi = -118.1°$ and $\Psi = 142.3°$. In the experiments, three carbon atoms are used as A-atoms: A-atom 1 is fixed, A-atoms 2 and 3 are mobile. b) The sampling volume for A-atom 2. c) The sampling volume for A-atom 3.

Here, we would like to see how mono-directional ART-RRT explores the dihedral-angle space (Φ and Ψ) of dialanine. Therefore, the method is applied to obtain a tree made of 2000 nodes, with the parameters shown in Table 7.1.

Table 7.1 Parameters used in mono-directional ART-RRT for exploring the dihedral-angle space of dialanine.

Parameter description	Notation	Value
Extension step in RRT tree	δ	2 Å
Initial transition test temperature	T	0.001 K
Temperature factor in transition test	λ	2
FIRE integration time step	t_F	1 fs
FIRE number of steps	n_F	10
Max number of failures in transition test	$\mathcal{S}$	10
Number of ARAP iterations	m	100

7.1.2 Result

Four experiments are done with different variants of mono-directional ART-RRT to examine the use of the constrained minimization and transition test. The experiment details are shown in Table 7.2, with some corresponding statistics. In addition, the maximum and average

energies of the tree nodes are presented in Figure 7.2, and the projection of the tree nodes onto the Φ and Ψ space are shown in Figure 7.3, where dark/blue and bright/red colors represent low-energy and high-energy regions, respectively.

Table 7.2 Summary of the results for the different experiments to explore the dihedral-angle space of dialanine with mono-directional ART-RRT.

Experiment	Figure	ART-RRT variant		No. of nearest neighbor searches	No. of transition tests	Time (s)
		Constrained minimization	Transition test			
1	7.3a	No	No	797	0	8.0
2	7.3b	Yes	No	727	0	7.3
3	7.3c	No	Yes	13372	14987	134.1
4	7.3d	Yes	Yes	6497	8164	65.0

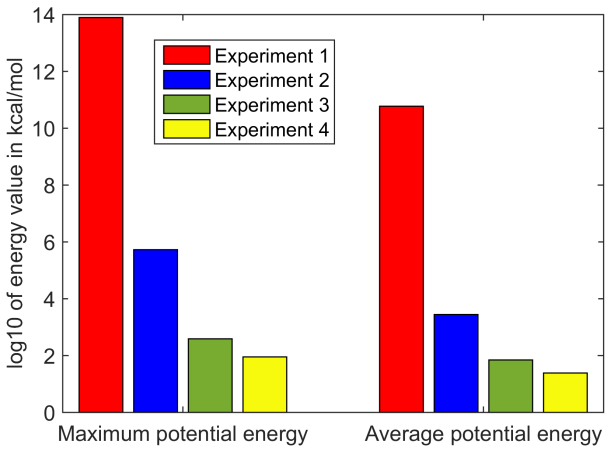


Fig. 7.2 Maximum and average energy of the tree nodes for the experiments with dialanine. The combination of constrained minimization and transition test (case 4) leads to the states with lowest energies.

First, let us consider Experiment 1 where ART-RRT is used while removing both the constrained minimization and the transition test. As shown in Figure 7.3a, the method is able to cover relatively homogeneously the dihedral-angle space (Φ and Ψ) in low computational time (8.0 s). However, the energy of the tree states can be high since energies are not directly taken into account (see Figure 7.2).

In Experiment 2, transition test remains unused while constrained minimization is applied. Figure 7.2 shows that the constrained minimization does reduce the energy of the tree nodes. However, on the dihedral-angle map of Figure 7.3b, we see that the high-energy regions corresponding to unfavorable dihedral angles are not avoided. What happens here is that the minimization mostly reduces the energy terms related to bond lengths and angle bends which are those penalizing the most the total energy, and affects less the dihedral terms.

Now, when looking at Table 7.2, one can see that the computational time is slightly lower for Experiment 2 than for Experiment 1 even though Experiment 2 employs the constrained minimization. Actually, this is because the constrained minimization tends to give minimized states deviated from the linear interpolation path toward the randomly sampled state, leading to longer branches to reach the sampled state. Consequently, fewer branches are built, and hence, fewer number of nearest neighbor searches are performed, as confirmed by lower number of nearest neighbor searches in Experiment 2 than in Experiment 1 (see Table 7.2).

In Experiment 3, the transition test is used whereas the constrained minimization is not. Figure 7.2 shows that this variant leads to lower energy than Experiment 2. This time, the states appear more in low-energy regions of the (Φ, Ψ) map as shown in Figure 7.3c. However, the computational time for this experiment (134 s) is much greater than for the two previous experiments (8.0s and 7.3s), due to a greater number of transition tests.

Finally, it is the combination of both the constrained minimization and the transition test (Experiment 4), that leads to the lowest maximum and average energy values as shown in Figure 7.2. Moreover, it is interesting to notice that the computational time in Experiment 4 is more than twice as low as in Experiment 3. This is because the constrained minimization not only tends to reduce the number of nearest neighbor searches, but it also gives low-energy states which are more easily accepted by the transition test, and hence, reduces the number of minimization iterations to perform. As shown in Table 7.2, the number of nearest neighbor searches and transition tests are approximately twice lower in Experiment 4 than in Experiment 3. Figure 7.3d also shows that the explored states are located in much lower-energy regions than those from the other variants.

This section has shown that mono-directional ART-RRT can efficiently explore the dihedral-angle space of dialanine. The combination of the constrained minimization and the transition test helps to direct the exploration toward low-energy regions and reduces the computational time.

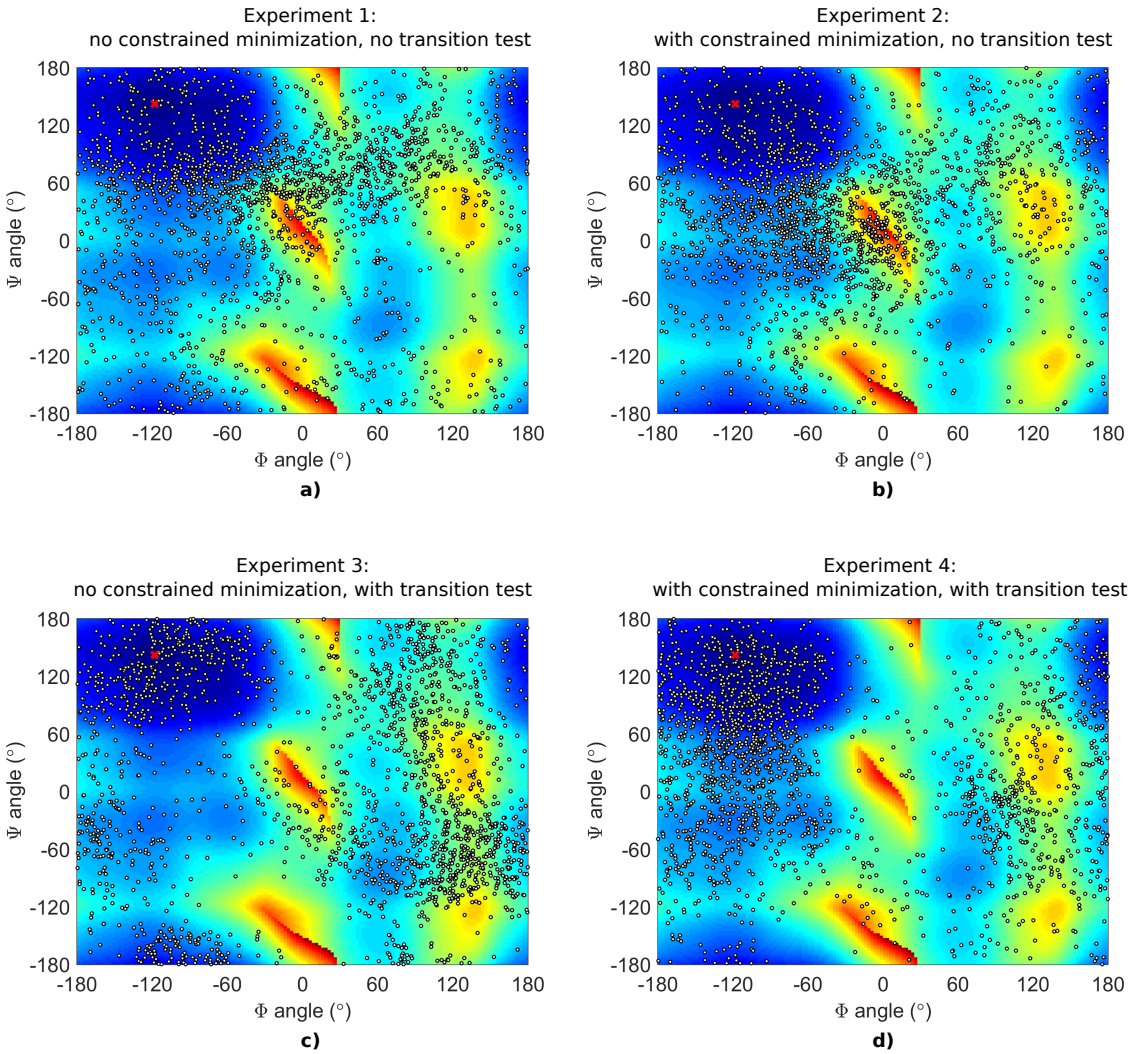


Fig. 7.3 Exploring the dihedral-angle space of dialanine with mono-directional ART-RRT. The background plots show the energy landscape projected on the dihedral angles of dialanine. Dark/blue regions and bright/red areas represent low-energy and high-energy regions. Tiny dots represent the tree states whereas the red cross corresponds to the initial state (root node of the tree). The plots show the exploration of ART-RRT for different variants. a) Experiment 1: neither the constrained minimization nor the transition test is used. b) Experiment 2: the transition test is not used whereas the constrained minimization is used. c) Experiment 3: the transition test is used whereas the constrained minimization is not used. d) Experiment 4: both constrained minimization and the transition test are used. As one can see, it is when combining both the constrained minimization and the transition test that the low-energy regions of the space are the best explored.

7.2 Finding ligand unbinding pathways from receptors

This section shows the application of mono-directional ART-RRT for efficiently generating ligand unbinding pathways. The work has also been published in the Journal of Computational Chemistry [175]. As one will see below, ART-RRT does not require a reaction coordinate to guide the search and can be used for finding pathways with known or unknown directions beforehand. The method is evaluated on several benchmarks and the obtained solutions are compared with the results from other state-of-the-art approaches such as the Steered Molecular Dymamics (SMD) [114], Manhattan-like Rapidly-exploring Random Tree (ML-RRT) [50, 51] and RAMD [158] methods. We show that the method is time-efficient and produces pathways in good agreement with other state-of-the-art solutions. These paths can serve as first approximations that can be used, analyzed or improved with more specialized methods.

7.2.1 Benchmarks

The ART-RRT method was implemented in C++ as a module of the SAMSON platform [113]. Energy and forces were evaluated with the GROMOS96 43a1 force field in vacuum [234, 57] using GROMACS [1] integrated in SAMSON.

For structure preparation, missing residues were modeled by MODELLER [197] integrated inside Chimera [183] and missing atoms by the swissPDB software [91]. Molecular topologies were generated by the *pdb2gmx* command in GROMACS for proteins and by the PRODRG server [205] for ligands. Before running ART-RRT, the systems were relaxed to their local minima using the FIRE method [24]. The parameters used for ART-RRT are shown in Table 7.3. This choice was based on trials and errors on several benchmarks and could probably be further improved, or tuned for specific scenarios. However, the optimization of these parameters is left for future investigations.

We applied the method to three cases of ligand unbinding pathways shown in Table 7.4. These benchmarks were chosen because their pathways had already been investigated by other well-known methods.

Since ART-RRT is based on a stochastic process, we ran the method for the first two benchmarks 20 times each. For the third benchmark, we ran ART-RRT 50 times because we observed a greater variety of pathways. For each benchmark, only two carbon atoms at the extremity of the ligand were assigned as A-atoms (this information will be detailed in the Result section), and their displacements were controlled by the RRT scheme. The other ligand atoms labeled as P-atoms passively followed thanks to the ARAP modeling (ARAPm) method. The protein atoms labeled as N-atoms were passively moved thanks to

Table 7.3 Parameters used in mono-directional ART-RRT for finding ligand unbinding pathways.

Parameter description	Notation	Value
Number of ARAP iterations	m	20
Extension step in RRT tree	δ	1 Å
Initial transition test temperature	T	0.001 K
Temperature factor in transition test	λ	2
Max number of failures in transition test	$\mathcal{S}$	1
FIRE integration time step	t_F	$1 \times 10^{-15} s$
FIRE number of steps	n_F	10

Table 7.4 Ligand-unbinding benchmarks for evaluating the mono-directional ART-RRT.

Id	Description	PDB	Chain Id	# ligand atoms	# protein atoms	Reference
I	Imatinib in protein kinase c-Kit	1T46	A	54	3178	[248]
II	Thiodigalactoside in lactose permease	1PV7	A	31	4292	[51, 118]
III	Retinoic acid in human receptor	2LBD	A	22	2350	[136]

the constrained minimization, except for one arbitrarily chosen atom whose position was always fixed, in order to avoid a global translation of the protein with the ligand as the ligand escaped from the binding site. Each exploration was stopped as soon as the center of mass of the ligand was 40 Å away from its original position. Then, a solution path was extracted from the tree by linking the root node to the last accepted node. The sampling volume, *i.e.* the region where the search is performed for A-atoms, was either a cubic volume or a rectangular volume restricted to the region of interest, and was centered on the ligand when the ligand was at the bound state.

In benchmark I, we studied the influence of different sampling volumes on ART-RRT results. In benchmark II, we analyzed the effect of postprocessing ART-RRT paths with a path-optimization method.

7.2.2 Results

Unbinding of imatinib from the c-Kit protein kinase

This experiment involves imatinib, a type II kinase initially bound to the inactive form of the c-Kit protein kinase. A better understanding of the interaction mechanisms between kinase and its inhibitors is of major importance since they are involved in essential physiological processes [48]. This experiment is motivated by the study performed in [248],

which uses the SMD method to examine two candidate channels called ATP channel and allosteric-pocket (AP) channel.

Our goal is to show that ART-RRT is able to find these pathways at low computational cost, without the need for a reaction coordinate or an explicit bias. Since the setting we used largely differs from the SMD study (for example, we do not consider boundary conditions, explicit solvent nor constraints on alpha carbon atoms), we only expect to find the same types of pathways with potentially slight variations.

The initial complex and the A-atoms on imatinib are shown in Figure 7.4. The figure also illustrates two sampling-domain setups that we used for this benchmark: a cubic volume implying no direction-bias, and a rectangular volume favoring solutions through the two candidate channels.

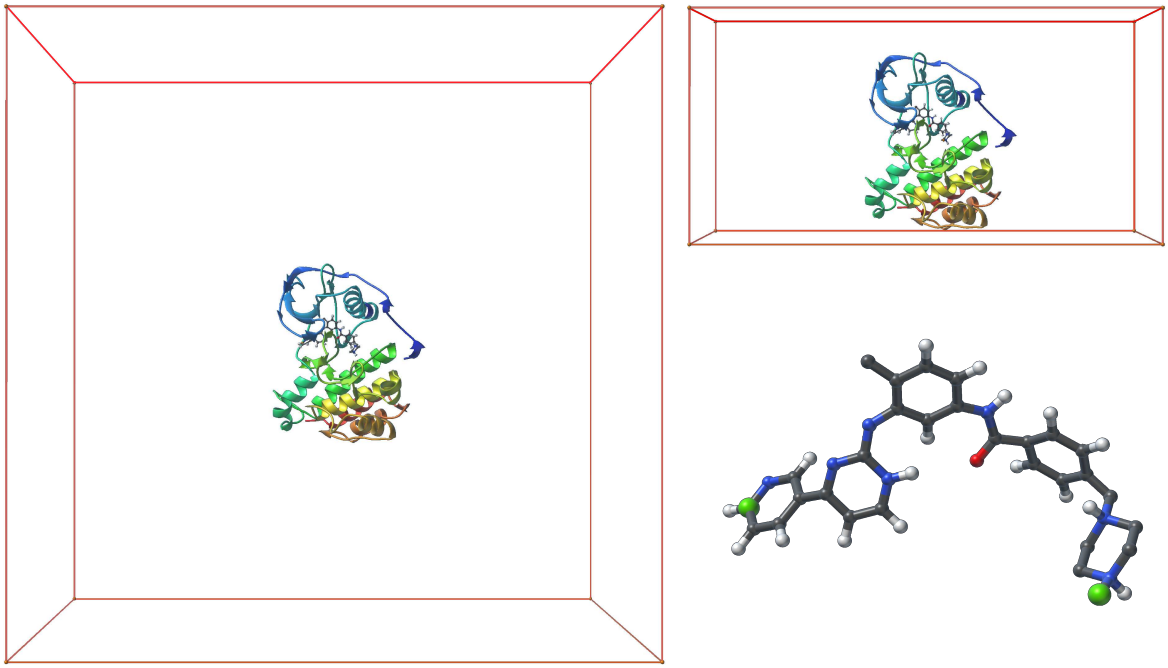


Fig. 7.4 Left (the cubic setup): the protein-ligand complex for Benchmark I at its initial state inside a cubic volume (centered on the ligand) defining the sampling domain for the A-atoms. Top-right (the rectangular setup): the complex inside the rectangular sampling domain centered on the ligand. Bottom-right: A closer view on the ligand (imatinib) where two carbon atoms (in green) are set as A-atoms.

Figure 7.5 shows all the paths found by ART-RRT for the cubic and rectangular setups. In general, most paths belong to the candidate channels in both setups. However, other pathways were also found for the cubic setup, in addition to those along the candidate channels.

Table 7.5 presents a summary of the results by ART-RRT for the ATP and AP pathways, as well as other pathways that do not follow these two roads. As seen from the table, the

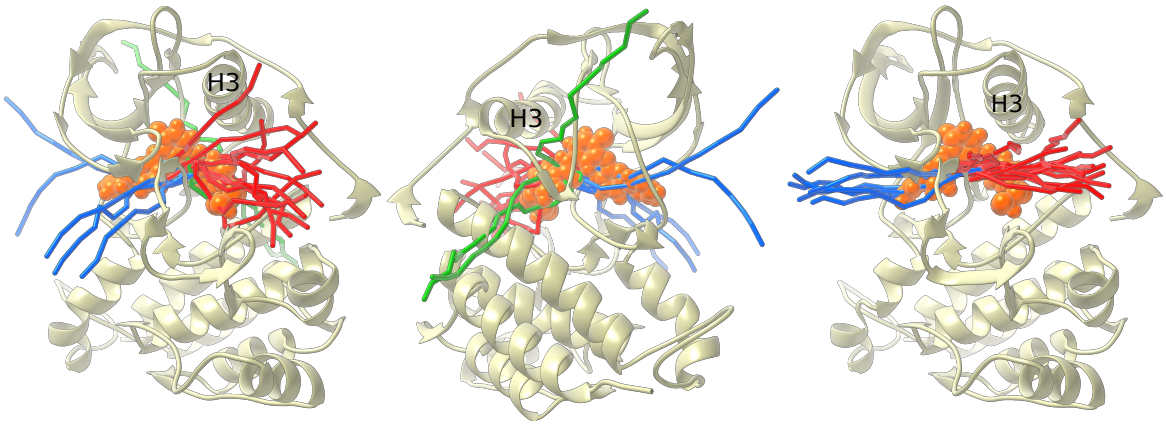


Fig. 7.5 The ART-RRT paths for both setups in Benchmark 1. Views from both sides of the protein are shown for the cubic setup. The protein is represented by the ribbons and the ligand in orange balls. The ATP and AP paths are represented by the blue and red sticks, respectively. The other paths are represented by the green sticks. Each stick traces the center of mass of the ligand.

average computational time to find a path is quite short compared with classical MD (416.4 ± 47.6 seconds for the cubic setup, and 254.8 ± 57.7 for the rectangular setup). The rectangular setup took less time because the search did not wander in a space as large as in the cubic setup.

Table 7.5 Summary of results for imatinib unbinding from the c-Kit protein kinase.

	Cubic setup			Rectangular setup		
	# paths	Energy barrier ($kJ.mol^{-1}$)	comput. time (s)	# paths	Energy barrier ($kJ.mol^{-1}$)	comput. time (s)
ATP channel	5	1600.3 ± 214.0	403.9 ± 27.4	6	1617.2 ± 334.4	255.2 ± 43.0
AP channel	12	1587.4 ± 351.7	425.4 ± 57.0	14	1427.9 ± 294.0	254.6 ± 62.9
Other	3	2995.4 ± 1273.3	401.4 ± 4.8	0	-	-
Total	20	1717.0 ± 761.8	416.4 ± 47.6	20	1484.7 ± 318.7	254.8 ± 57.7

The table also shows that the paths in the "Other" category have significantly higher energy barriers compared to the rest, which explains why fewer paths are found for this category. In addition, the AP paths have generally smaller energy barriers than the ATP paths for both setups.

Figure 7.6 (for the rectangular setup) and Figure 7.7 (for the cubic setup) help us understand why the AP pathway tends to have smaller energy barrier than the ATP pathway. Each plot in these figures shows the mean and standard deviation of the maximum displacement of the alpha carbons from the initial bound state, for either the ATP or AP paths. According

to previous studies [108, 254, 248], there are four regions which move significantly during the unbinding process of imatinib from the c-Kit protein (see Figure 7.8): the JMR (GLY-1 to ASP-15), the β-sheet (LEU-25 to LEU-61), the helix αC (HIS-66 to GLY-84) and the A-loop (CYS-187 to LEU-209). Note that our residue numbering is different from the one of the original pdb file, due to the reconstruction of the missing residues. As shown in both figures, these regions are also found to be the most mobile in the ART-RRT results. In the rectangular setup, the three most mobile regions are the β-sheet, helix αC, and A-loop for the ATP paths (Figure 7.6a), while all of the four mentioned regions are found to be most mobile in the AP paths (Figure 7.6b). The same behavior is observed for the cubic setup (Figure 7.7) except that more mobile regions show up and the RMSD values are also smaller in general compared with the results in the rectangular setup. This is because the cubic setup allows for more exploring directions, and hence, the ligand can push less on the residues of the important regions while pushing more on other residues on the channel lining. In both figures, larger displacements of the alpha carbon atoms in the ATP paths than in the AP paths are observed. For example, the maximum mean value in Figure 7.6 for the ATP paths is greater than 3 Å, and for the AP paths is less than 2 Å. This implies that several residues must be displaced to a greater extent to facilitate the ligand passage along the ATP channel than the AP channel.

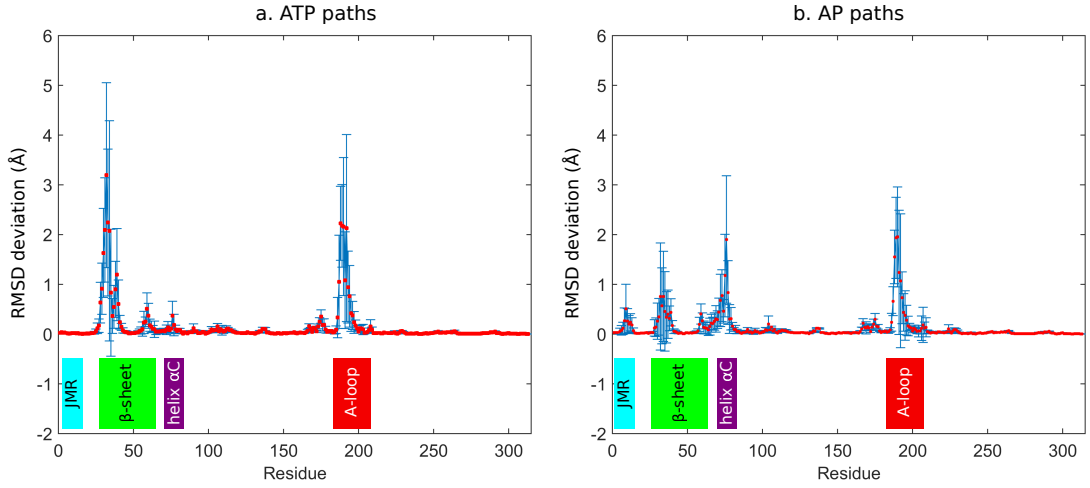


Fig. 7.6 Benchmark 1 with the rectangular setup: The mean (red dots) and standard deviation (vertical blue bars) values of the maximum RMSD deviations from the initial bound state of alpha carbons for a) the ATP paths and b) the AP paths. The residues of 4 important regions (JMR, β-sheet, helix αC and A-loop) are spanned by the colored boxes. The AP paths involve motions in all these 4 regions while the ATP paths involve motions in all of them except the JMR.

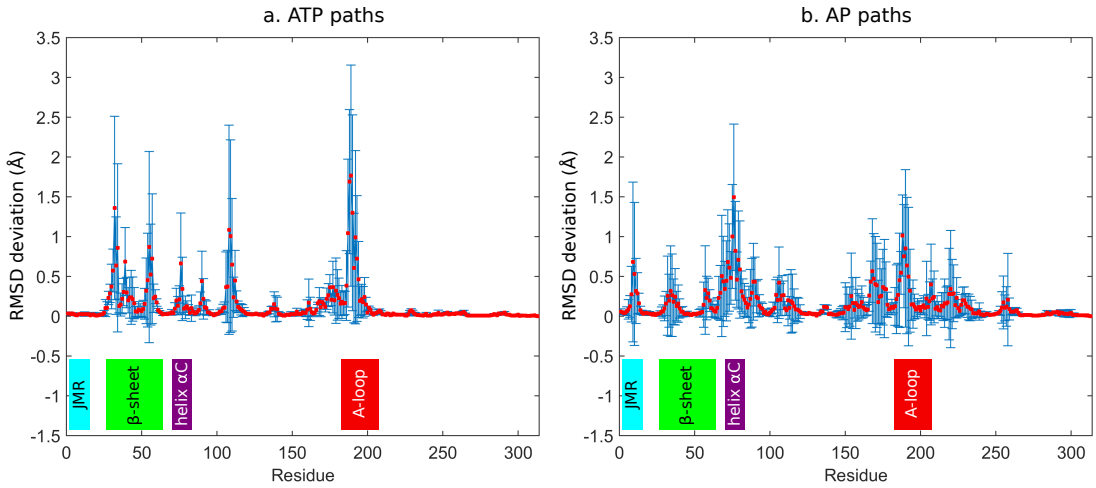


Fig. 7.7 Benchmark 1 with the cubic setup: The mean (red dots) and standard deviation (vertical blue bars) values of the maximum RMSD deviations from the initial bound state of alpha carbons for a) the ATP paths and b) the AP paths. The residues of 4 important regions (JMR, β-sheet, helix αC and A-loop) are spanned by the colored boxes. Similar to the rectangular setup, the AP paths involve motions in all of the 4 regions while the ATP paths involve motions in all of them except the JMR. Besides these important regions, the cubic setup leads to significant motions in other regions.

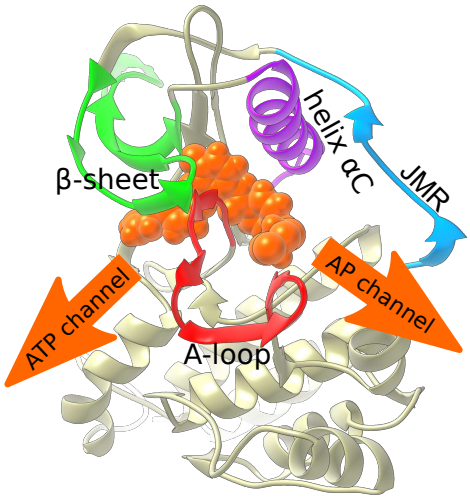


Fig. 7.8 Location of the JMR (blue), β-sheet (green), helix αC (purple) and A-loop (red). The ligand is represented by orange balls.

Let us now examine in detail the candidate pathways. For this purpose, two representative paths are picked from the rectangular setup, i.e. paths which give the similar patterns to those of the mean values in Figure 7.6. One of them represents the ATP path and the other represents the AP path. Figure 7.9 shows the average displacement of the alpha carbons of the most mobile residues (only the alpha carbons which have the maximum displacement

more than 0.5 Å are considered) and the van der Waals (vdW) energy along the paths. The curves in the left plot rise from 0 Å to a maximum value, indicating the opening of the channels, then fall to stable levels, indicating the closing of the channels after the ligand escape. Our results agree with the SMD study that the passage along the ATP channel leads to more displacement for the residues involved. Our interpretation is that the passage along the ATP channel, therefore, requires more energy (see Table 7.5) to push the residues obstructing the channel. The only difference in the left plot of Figure 7.9 with that in the SMD study is that our measurement is lower in value, probably due to the lack of water molecules in our experiment. Note that the water molecules can keep the channels open wider by filling the ligand place after its escape. The right plot in Figure 7.9 shows the vdW energy of these representative paths. The vdW energy barriers of our curves are about 0.03×10^4 kcal/mol, which is close to that found by the SMD study (0.02×10^4 kcal/mol). Interestingly, similar to the SMD study, the vdW energy barrier of the ATP path also occurs before that of the AP path.

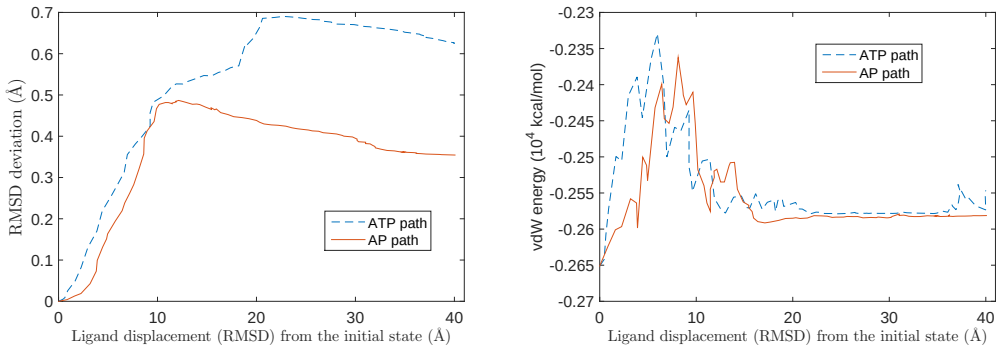


Fig. 7.9 Results for the representative ATP path and AP path of the Rectangular setup. The left plot represents the average RMSD deviation of the alpha carbons composing the ATP and AP channels from the initial bound states along the dissociation pathways. One can see higher RMSD values for the ATP path, implying more motion of the residues to give way for the ligand along this channel. The right plot shows the van der Waals energy along the dissociation pathways. Similarly to the result from [248], it appears that the energy barrier of the ATP path occurs before that of the AP path.

The results from our experiment support the hypothesis that the AP channel is preferred for the ligand unbinding. The same conclusion is found in a recent study of the same protein family using the umbrella sampling simulation [226]. However, whether the ATP or AP channels are preferred for the ligand unbinding is still debatable since a different conclusion is reached by the SMD study [248]. The differences between our conclusion and that from the SMD study may come from the experimental setup. In particular, our experiment is done in vacuum, while the SMD study is done with explicit solvent. Secondly, the chosen direction

of the pulling force in the SMD method may not be ideal for the escape of the ligand while our method is sampling-based and favors the passage along low-energy regions. In any case, the main purpose of this experiment is to show the capability of our method to efficiently find the main pathways found by other methods from the literature.

Unbinding of Thiodigalactosid from Lactose permease

We used ART-RRT to simulate the unbinding of Thiodigalactosid from the Lactose permease, a twelve-alpha-helical membrane transport protein [2]. The goal of this experiment is to compare the ART-RRT method with the ML-RRT method proposed in [51]. The ML-RRT method, which also relies on the RRT exploration scheme, represents both the ligand and the protein in internal coordinate system, *i.e.* a set of dihedral angles. For this benchmark, ML-RRT allows full flexibility of the ligand, whereas only certain parts of the protein are flexible. ML-RRT also divides the whole system in passive and active parts, where the active parts are controlled by the RRT scheme and the passive parts are displaced as soon as steric collisions are detected with the active parts.

The sampling volume and A-atoms on the ligand for ART-RRT are shown in Figure 7.10. This volume is limited to the upper part of the protein since we want to study the ligand unbinding toward the periplasmic side of the protein only.

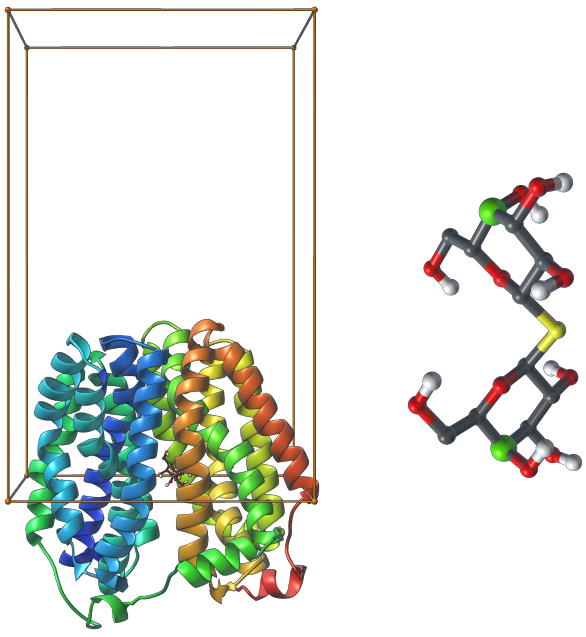


Fig. 7.10 Left: The ligand-protein complex for Benchmark II (in ribbon) with the sampling volume for the A-atoms. This box is biased toward the periplasmic side of the protein. Right: a closer view on the ligand with the two A-atoms in green.

For this benchmark, ART-RRT takes 136.5 $\pm$ 21.1 seconds to find each path versus 1 hour for the ML-RRT method. In average, this corresponds to a speed-up of more than 26 times compared with the ML-RRT method. The computational time is also much smaller than what is required for classical MD simulations.

For this study, we also investigate the effect of a path-optimization method to locally improve the paths obtained with the ART-RRT method. Therefore, we apply the Nudged Elastic Band (NEB) method [120, 101] to optimize the ART-RRT paths. The NEB method is implemented as a parallel module in the SAMSON software platform [113]. To ensure the stability of the NEB results, we only keep a limited number of points along the ART-RRT paths [100]. Hence, each ART-RRT path is cut down to about 28-38 conformations per path before the NEB method is applied. The total computational time to obtain a path with ART-RRT that are later optimized with the NEB method is 181.0 $\pm$ 30.2 seconds. This post-treatment operation is, hence, computationally cheap (about 44.5 seconds more is spent for each path) and leads to much lower energy barriers (2 to 16 times), as shown in Figure 7.11. However, this optimization only adjusts a given path *locally*, while its nature remains unchanged (see Figure 7.12).

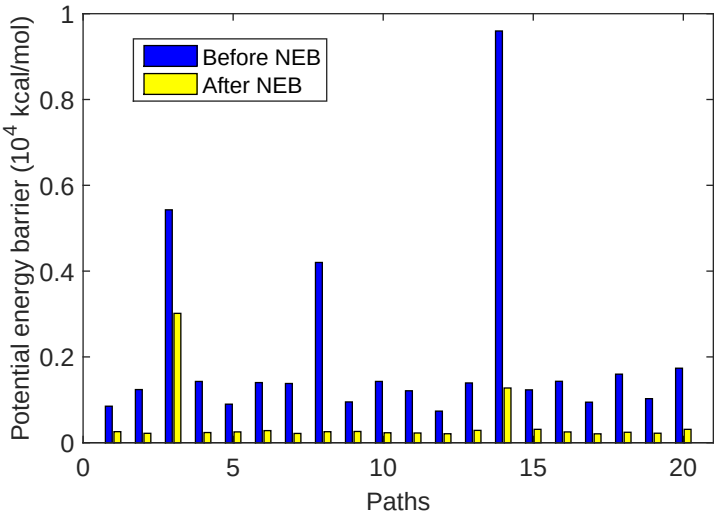


Fig. 7.11 Potential energy barriers for the paths directly obtained with ART-RRT and those postprocessed with the NEB method. This optimization step greatly reduces the energy barriers (from 2 to 16 times).

To compare with the ML-RRT method, we recorded the contacts that the ligand makes with the protein along its unbinding pathway. As defined in Ref.[51], a contact is recorded when the distance between a protein atom and a ligand atom is lower than the sum of their van der Waals radii plus 1 Å.

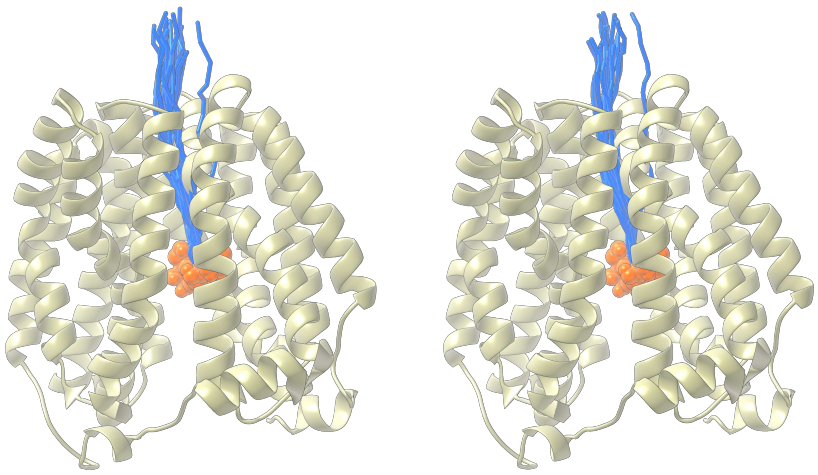


Fig. 7.12 All the paths found for Benchmark II (in blue sticks). Left: paths obtained by ART-RRT. Right: optimized paths obtained by ART-RRT + NEB. The protein and ligand are represented by ribbons and orange balls, respectively. The NEB optimization only adjust the paths locally, while the path natures remain unchanged.

The top part of Figure 7.13 shows the probability of contact between the ligand and the protein during unbinding, for the residues reported in Ref.[51], along three segments of the pathway: 0-10 Å, 10-20 Å, and after 20 Å. Precisely, each box indicates the percentage of paths in which a contact is present between the ligand and a particular residue for a given path segment. We observe that all the contacts reported by the ML-RRT method are also found with the ART-RRT method. Moreover, the contact patterns are similar: for example, residues GLU-269 to ASP-237, PHE-27 to ASN-245 and THR-45 to HIS-35 (from left to right in the figure) appear at the beginning, the middle and the end of the unbinding path, respectively. Interestingly, this list of residues are also reported to have hydrogen bonding and hydrophobic interactions with the ligand in another study using the SMD method [118].

The bottom part of Figure 7.13 shows more contacts found by ART-RRT which appear at least 30 % of the paths. In [118], the residues GLU-269, HIS-322, ARG-144, ARG-302, GLU-325 and GLU-126 are deemed essential for the lactose transport. ART-RRT detected the interaction of the ligand with 5 out of these 6 residues (GLU-269, HIS-322, ARG-114, ARG-302, GLU-325) whereas only GLU-269 and HIS-322 are reported in the ML-RRT results. GLU-325 is not shown in the figure because its presence is less than 30 % of the paths while GLU-126 was not considered in our experimental setup because it does not belong to the passage toward the periplasmic side of the protein.

We also analyzed the effect of the NEB method on the contact pattern (see Figure 7.14). The comparison between this figure and Figure 7.13 shows that the contact pattern is essentially preserved although the NEB method reduces the energy barriers of the paths

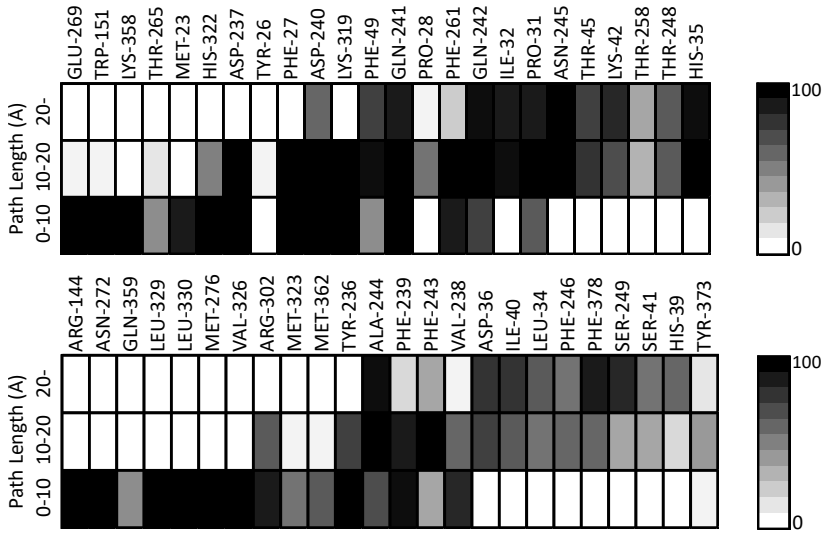


Fig. 7.13 Contacts made by the protein residues and the ligand along 3 segments of the unbinding paths (0-10 Å, 10-20 Å, and after 20 Å) found by ART-RRT. The grey scale shows the percentage of times a specific contact is present for that particular path segment over all pathways. On the top is the contact result with the residues also found by ML-RRT. On the bottom is the contact result with other residues found by ART-RRT. Only residues which have contact at least 30 % of the paths are shown.

significantly (see Figure 7.11). Five contacts (GLN-359, MET-323, MET-362, PHE-246 and TYR-373) become less present (less than 30 % of the path), and hence, do not show up in this figure. Only one contact (with TYR-26) is no longer present in the paths after the optimization.

Hence, the protein-ligand contact analysis shows that ART-RRT can give results comparable to those obtained with either the ML-RRT or the SMD method [118], but in a much shorter computational time. Moreover, the contact analysis and Figure 7.12 show that although the NEB method remarkably reduces the energy barriers of the ART-RRT paths, it does not significantly change the path natures.

Figure 7.15 shows the maximum displacements of the alpha carbons along the unbinding pathway for the paths before and after NEB optimization. As one can see, the most mobile residues are PHE-20 to PHE-55, ILE-230 to LEU-271, and MET-365 to LEU-385. The displacements of the residues from PHE-20 to PHE-55 and ILE-230 to LEU-271 are not surprising because these residues lie on the channel lining. The displacements of the residues from MET-365 to LEU-385 are induced by the motion of the residues ILE-230 to LEU-271. The figure also shows the effect of the NEB method that may reduce the RMSD values of some of the most mobile residues while slightly increasing the mobility for few other residues.

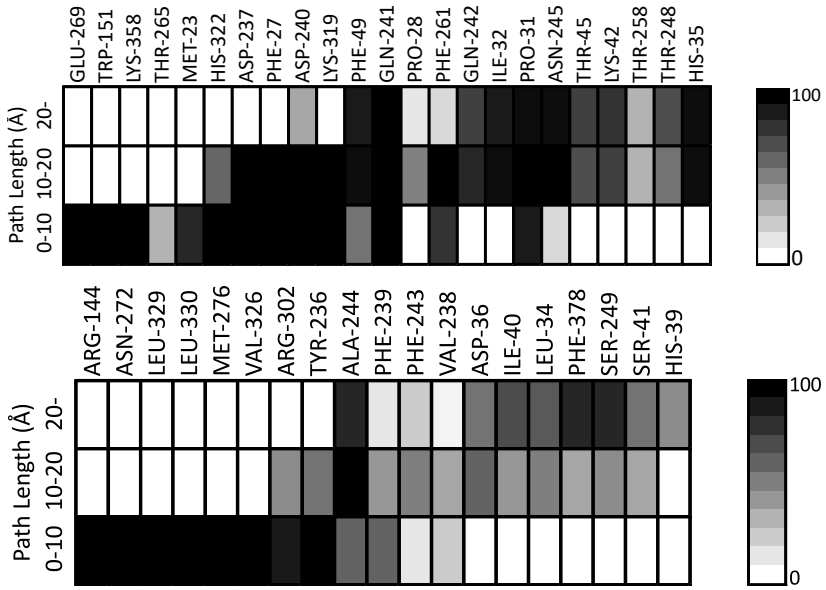


Fig. 7.14 Contacts between the protein residues and the ligand along 3 segments of the optimized paths (0-10 Å, 10-20 Å and after 20 Å) after NEB. The grey scale shows the percentage of times a specific contact is present for that particular path segment over all pathways. On the top is the contact result with the residues also found by ML-RRT (TYR-26 is no longer in this list). On the bottom is the contact result with other residues found by ART-RRT. Only residues which have contact at least 30 % of the paths are shown. GLN-359, MET-323, MET-362, PHE-246 and TYR-373 are no longer in this list.

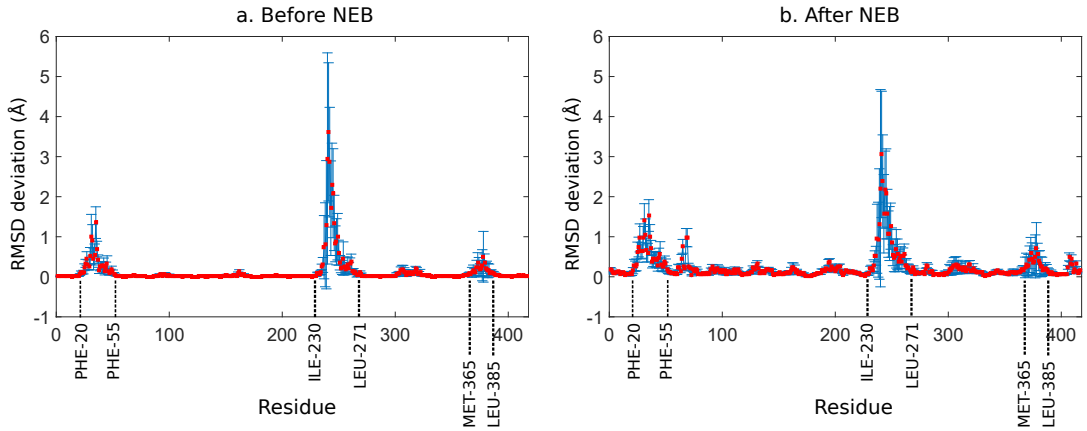


Fig. 7.15 The mean (red dots) and standard deviation (blue vertical bars) values for the maximum displacement of the alpha carbons from the initial binding state before and after optimization. The residues PHE-20 to PHE-55, ILE-230 to LEU-271, and MET-365 to LEU-385 are moved the most by the ligand during the unbinding. The paths after the optimization lead to lower RMSD values for the most mobile region, but more regions are subjected to some small displacements.

To observe how the narrowest constriction of the channel (made by the gap between residues ILE-40 and ASN-245) reacts to the ligand-unbinding event, we measured the maximum alpha-carbon distances between these two residues along the unbinding paths. We found that its maximum opening is 12.1 ± 0.4 Å and 12.3 ± 0.4 Å for the paths before and after optimization, respectively. This result is slightly smaller than 15 Å that is the distance found by the ML-RRT method and reported by another experimental study [252] as necessary for lactose transport. The difference may be due to two reasons. First, our method lets the protein react according to the potential forces, and hence, large motions of the protein are not addressed. Second, water molecules which can widen the channel by taking the ligand place during the unbinding process are not modeled in our study. Despite this quantitative discrepancy, the paths found by our method show similar characteristics with those produced by the ML-RRT method and the SMD method [118].

Unbinding of retinoic acid hormone from its receptor

Nuclear hormone receptors are involved in many cellular processes such as reproduction, transcription, etc. and hence, subjected to many researches [136, 181]. Here, we study the unbinding pathways of retinoic acid from its receptor as in [136], where the SMD method was used. The bound state of the protein-ligand complex is modeled from PDB entry 2LBD. Figure 7.16 shows the two A-atoms on the ligand and their cubic sampling volume. As noted above, since a large variety of pathways were found for this benchmark, ART-RRT was run 50 times in order to produce averages.

In Reference [136], three pathways (I, II, III) were chosen for the SMD simulations based on the bound structure of 2LBD. Pathway I occurs through the space between helices H11, H12 and the loop made by them. Pathway II is through the space beneath helices H11 and H12. In our result, we also include the space between helix H3 and the loop H11-H12 in pathway II. Pathway III is a tunnel that can be seen by looking at the molecular surface between helix H3 and the loop H1-H3. The ART-RRT method was capable of finding all these pathways (see the left picture of Figure 7.17).

Table 7.6 shows the number of paths found for each pathway as well as the average energy barrier and computational time. The ART-RRT method spent about 399 ± 59.1 seconds to find each path.

In addition to the mentioned pathways, ART-RRT also found pathways IV, V, VI and Other (pathways which do not belong to the other six categories). Pathway IV is through the space between H11 and the N-terminal of H7. Pathway V is between H6, H7 and the β-sheet between H5 and H6. Pathway VI is through the space between the C-terminal of H1 and the β-sheet between H5 and H6. All the ART-RRT paths along these pathways can be seen in

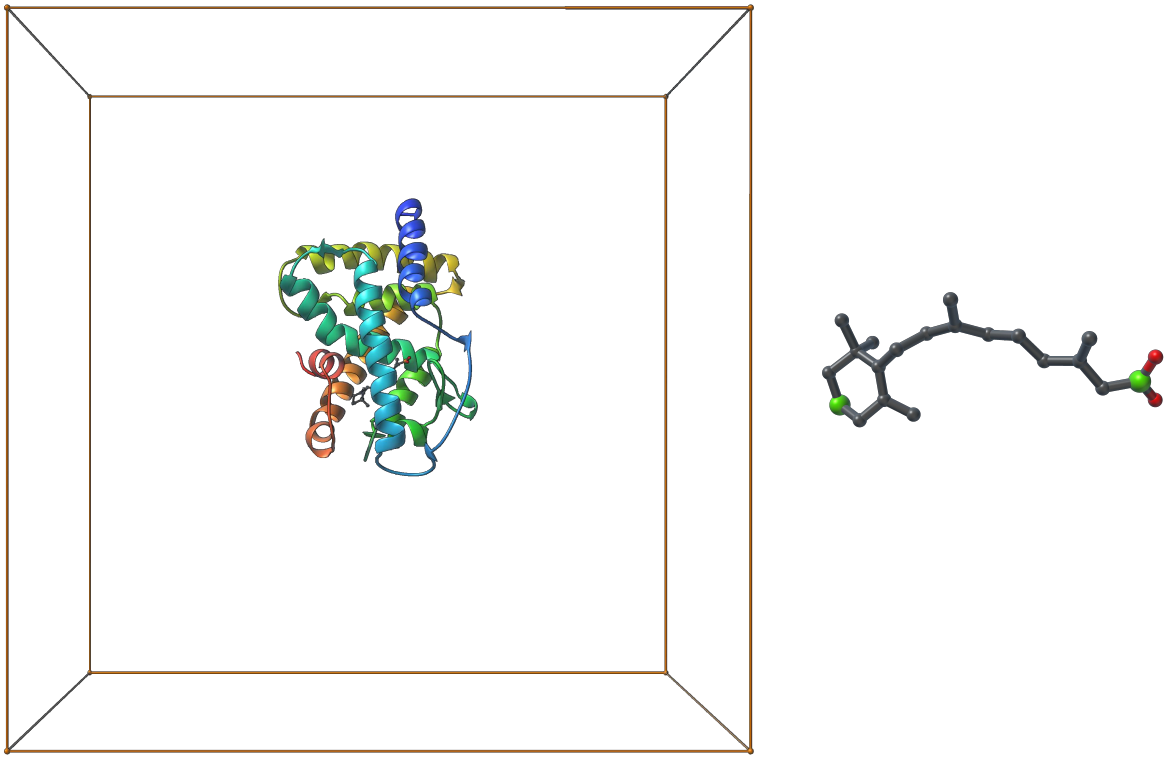


Fig. 7.16 Left: the system (in ribbons) inside a cubic sampling volume for the A-atoms in the ligand of Benchmark III. Right: a closer view on the ligand with the A-atoms in green.

Table 7.6 Summary of the results for retinoic acid hormone unbinding from its receptor.

	# paths	Energy barrier ($kJ.mol^{-1}$)	comput. time (s)
Path I	10	943.8 ± 313.3	411.8 ± 57.7
Path II	6	933.8 ± 338.0	364.6 ± 19.6
Path III	13	907.2 ± 207.0	390.1 ± 61.1
Path IV	14	895.1 ± 208.4	383.3 ± 59
Path V	4	1405.6 ± 182.9	471.4 ± 50.9
Path VI	1	1201.7	474.9
Other	2	1466.0 ± 216.5	422.5 ± 8.2
Total	50	982.4 ± 300.0	399.0 ± 59.1

the right picture of Figure 7.17. Interestingly, pathways IV, V and VI are also reported by another study that employs the RAMD method to find ligand unbinding pathways for another nuclear hormone receptor [181]. This shows the efficacy of the ART-RRT method in finding a large diversity of candidate pathways in just a few minutes for each path.

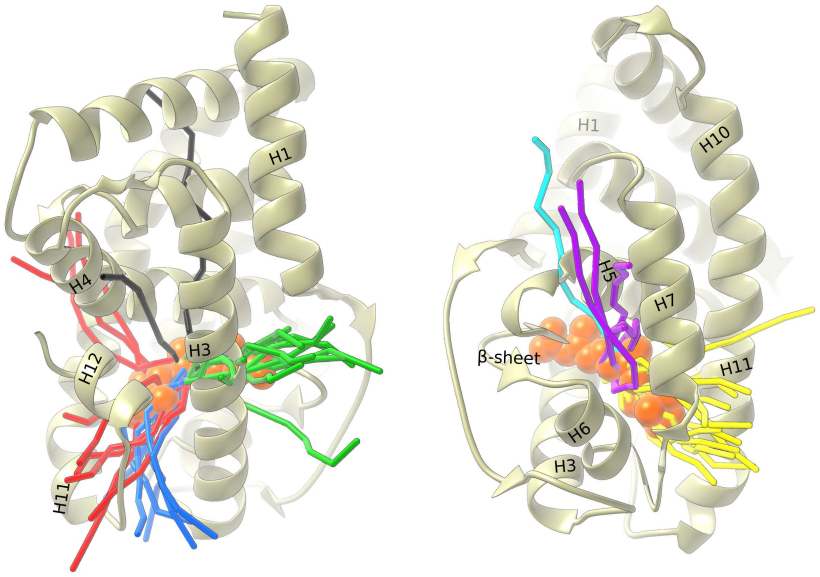


Fig. 7.17 Paths (in colored sticks) obtained by ART-RRT for Benchmark III. The protein is represented by ribbons and the ligand by orange balls. Two different views are shown for clarity. The left picture shows pathways I in red, II in blue, III in green and Other in black. The right picture shows pathways IV in yellow, V in purple and VI in cyan.

7.2.3 Conclusion & Discussion

This section presented the application of mono-directional ART-RRT for searching ligand-unbinding pathways. The experiments have shown that the ART-RRT method is fast and able to find a diversity of low-energy pathways. The method can also be easily tuned to focus the search on specific regions of the space. Overall, the results are in good agreement compared to those found by state-of-the-art approaches.

The paths found by ART-RRT could be further refined to compute minimum-energy paths, or could be used in other advanced methods such as transition path sampling to generate a path ensemble and estimate free-energy differences [59] or reaction rate constants [61].

Despite the preliminary success for the presented benchmarks, the current method still has several drawbacks such as the inability to address large protein motions and handle explicit solvent. Several possible improvements for the method are worth considering. Firstly, the investigation on how the method parameters, as well as the location and the number of A-atoms affect the results, could give a deeper understanding of the method. At the moment, only two A-atoms are selected and located at the extremities of the ligand. Secondly, the placement of A-atoms on the proteins would be an interesting strategy to sample large protein motions during the unbinding process. Thirdly, the adaptation of the current method for solvated systems would be beneficial for many users. Fourthly, the method could be extended

for more complex problems such as conformational changes of a protein and protein-protein interactions.

Finally, we have seen in the previous chapter that the method can be extended to use two exploration trees for finding the pathways between two given states. Some applications of bi-directional ART-RRT will be presented in the next chapter.

Chapter 8

Applications of bi-directional ART-RRT

This chapter presents the applications of bi-directional ART-RRT [1] to find pathways between two given states. First, a simple toy model is presented. It serves as proof of concept and also shows the potential of the method to address the loop-exploration problem in proteins [228]. Then, ART-RRT is applied for finding protein conformational transition pathways between two given protein structures. Finally, we present its application for finding pathways of a protein-ligand system given a bound and an unbound state.

8.1 Toy model

8.1.1 Model Description

This section investigates the capability of the ART-RRT method for finding feasible loop motions as those appearing in proteins to get from one state to another. Therefore, we use a simple model as shown in Figure 8.1 which contains a string and a hollow S-shape (the logo of the SAMSON software platform [113]). The flexible string, which represents a loop, contains grey, red and green atoms with any two consecutive atoms connected by a covalent bond. The blue atoms, which form the S-shape, represent the obstacles in the space. The goal is to search the paths for the string to change from the state shown in Figure 8.1a to the one shown in Figure 8.1b without colliding with the obstacles. All the atoms in the string are mobile except the terminal ones.

For this system, the cost/energy associated to a given state comes from two sources. First, spring forces are applied on the bonded atoms of the string for penalizing bond stretch or bond shrinkage. Second, van der Waals forces are applied for penalizing collisions of the string atoms with the obstacle atoms.

[1] For convenience, *bi-directional ART-RRT* is shortened as *ART-RRT* in this chapter unless stated otherwise.

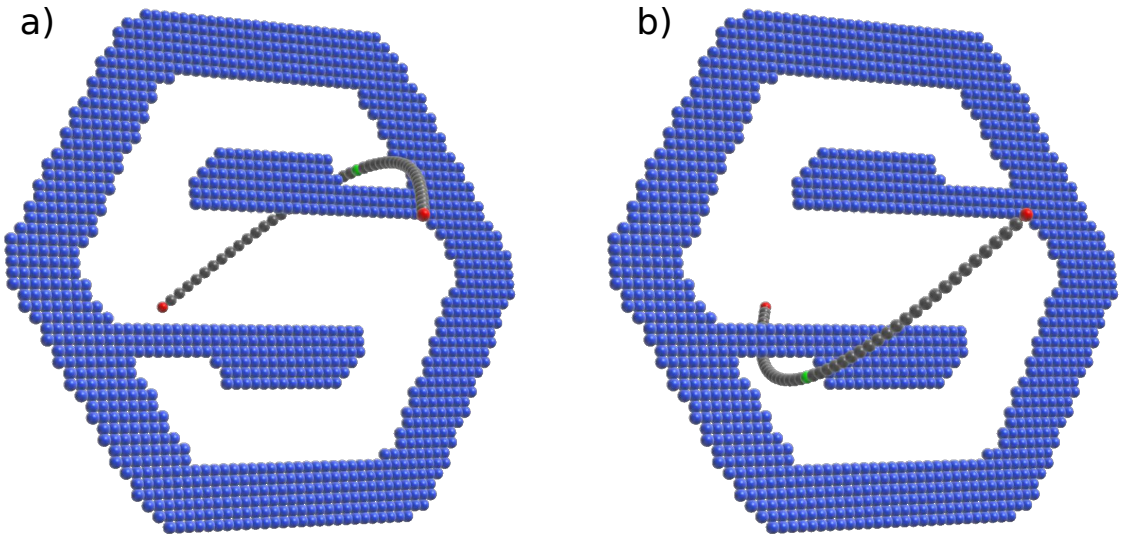


Fig. 8.1 The toy model consists of a string and an S-shape. The string is composed of grey, red and green atoms with any two consecutive atoms connected by a covalent bond. The blue atoms forming the S-shape play the role of obstacles. The goal is to search the paths for the string to change from the state in (a) to the state in (b) without colliding with the obstacles. The red and green atoms are A-atoms while the grey atoms are P-atoms. The string terminals (red-atom positions) are fixed while the green atom is mobile. The blue atoms are N-atoms and their positions are also fixed.

8.1.2 Results

For this model, we also investigate how the parameter setting of bi-directional ART-RRT affects the results. A summary of the parameters under investigation is shown in Table 8.1. In the string, the terminal atoms and the middle atom are A-atoms whereas the rest are P-atoms (see Figure 8.1). Because the terminal-atom positions are fixed, the only mobile A-atom is the middle one. Two sampling volumes are examined for this atom: a square in the plane containing the obstacle atoms and a cube. Both of them are centered at the center of mass of all the obstacle atoms and have edge lengths of 30 Å (see Figure 8.2). Hence, the square and cubic volumes reduce the search-space dimensions to only 2 and 3, respectively. The second investigated parameter is the maximum number of failures δ used in the transition test. Finally, we investigate the utility of the constrained minimization in several ways. The first one is the absence of the constrained minimization. The second one uses the constrained minimization with the green-atom position fixed, and the third one with the green-atom position free (i.e. the green-atom position is also modified by the constrained minimization). The other parameters are the same as those previously shown in Table 7.3. In addition, the energy threshold coefficient for $TestState$ during branch connection is $\gamma = 1$.

Table 8.1 Bi-directional ART-RRT parameters for the toy-model study with different setting options.

Parameters	Possible settings
Sampling volume	Square Cube
Max number failures in transition test ($\mathcal{S}$)	1 10
Constrained minimization	Inactive Active with fixed A-atoms Active with free A-atoms

ART-RRT is applied 20 times to search for 20 paths of the string. The obstacles are N-atoms and their positions are always fixed.

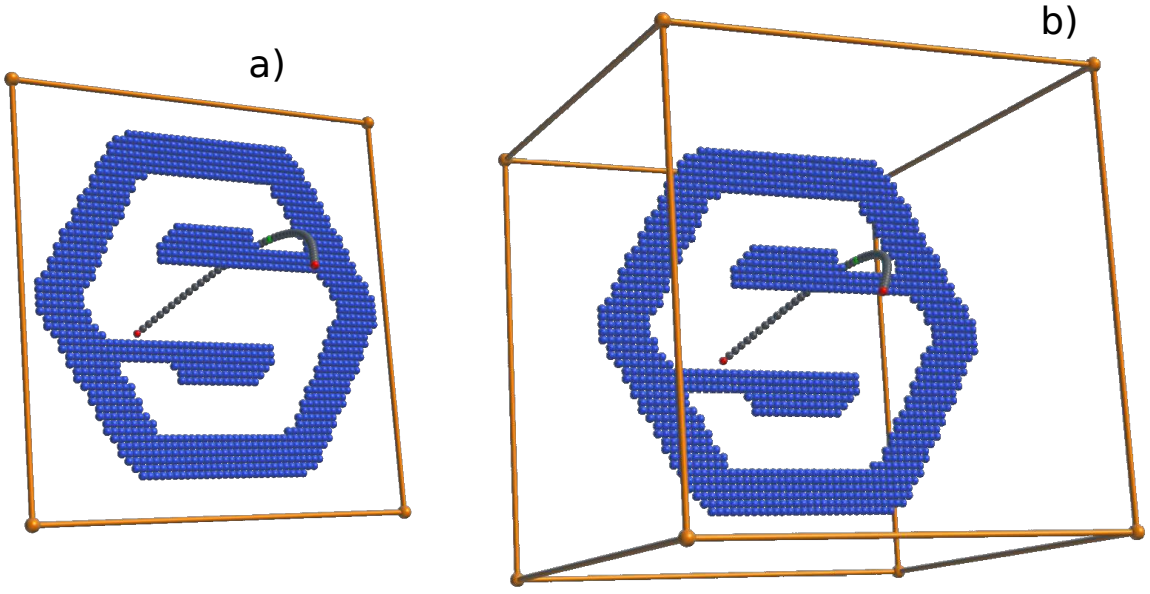


Fig. 8.2 Two sampling volumes for the mobile A-atom (in green): a) A square with edge lengths of 30 Å. b) A cube with edge lengths of 30 Å.

One valid path from the square setup and one valid path from the cubic setup found by ART-RRT are shown in Figure 8.3. The path from the cubic setup shows more flexes in the string than the one from the square setup because the mobile A-atom has one extra DoF in the cubic setup. The ART-RRT method may also give invalid paths, however. Here, a path is considered as invalid when the string collides with some obstacles or passes over them. Figure 8.4 illustrates these two cases.

The percentage of valid paths found by the ART-RRT method for the square and cubic setups is shown in Figure 8.5. Let us first analyze the result for the square setup shown in

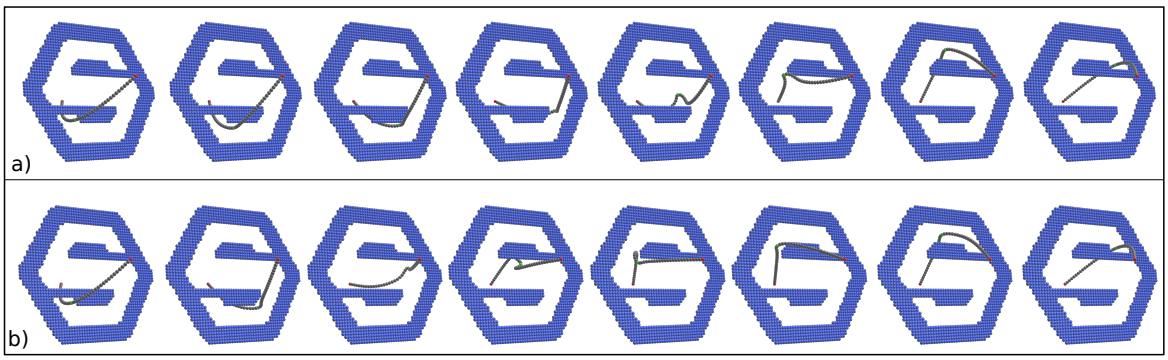


Fig. 8.3 Valid paths from a) the square setup and b) the cubic setup.

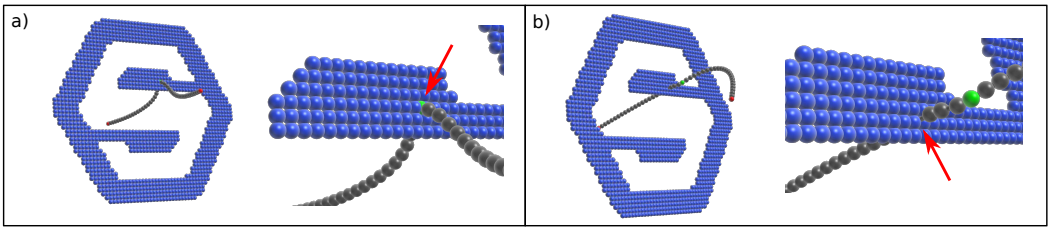


Fig. 8.4 Two types of invalid ART-RRT paths due to: a) atom collision between the string and the obstacles, b) bond collision between a string bond and the obstacles.

Figure 8.5a. The figure shows fewer valid paths when $\mathcal{S} = 1$ than when $\mathcal{S} = 10$. This is understandable because the method has more chance to reject bad states when $\mathcal{S} = 10$ before adjusting the temperature parameter. When observing the result found with constrained minimization and $\mathcal{S} = 1$, the invalid paths are found due to atom collisions when the A-atoms are fixed, and due to bond collisions when the A-atoms are free. This is due to the force-field design where atom collisions are much more penalized than bond stretches. Therefore, when the A-atoms are free, the states with atom collisions are transformed to those with bond stretches which have lower energy.

The result for the cubic setup shown in Figure 8.5b shows that greater $\mathcal{S}$ gives more valid paths. Moreover, contrarily to the square setup, we also observe here that more valid paths are found when the A-atoms are free than when they are fixed. This is because fixing A-atoms restricts the constrained minimization to a lower-dimensional plane, and hence, lower-energy states can be missed.

At first glance, these results seem to show that the constrained minimization has a negative impact on the results. However, its utility is proven when comparing the potential energy barriers of the paths found with different use of the constrained minimization. As shown in Figure 8.6, the energy barriers are smaller when the constrained minimization is used than when it is not used. The energy barriers are also smaller when the A-atoms are free than

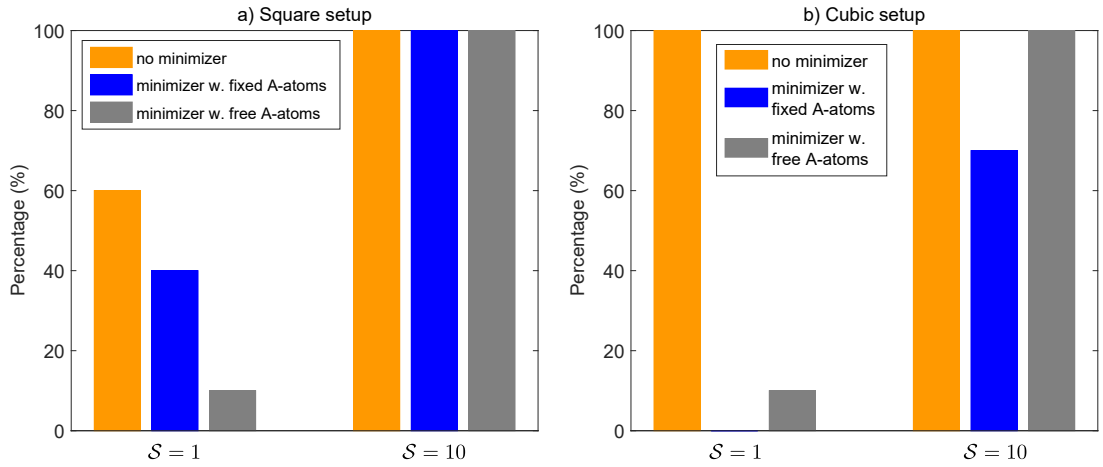


Fig. 8.5 Percentage of valid paths found by the ART-RRT method for a) the square setup, b) the cubic setup. The heights of the bars show the mean energy values and the red vertical ticks on the top of the bars show the standard deviation of energy values.

when they are fixed. In particular, one can see for the cubic setup that although 100% valid paths can be found with $\mathcal{S} = 1$ and no constrained minimization (Figure 8.5b, left orange bar), the path energy barrier is much higher (Figure 8.6b, left orange bar) than the other cases. Therefore, for this benchmark, to obtain valid paths with low-energy, the best choice of parameters appears to be $\mathcal{S} = 10$ and the constrained minimization with free A-atoms.

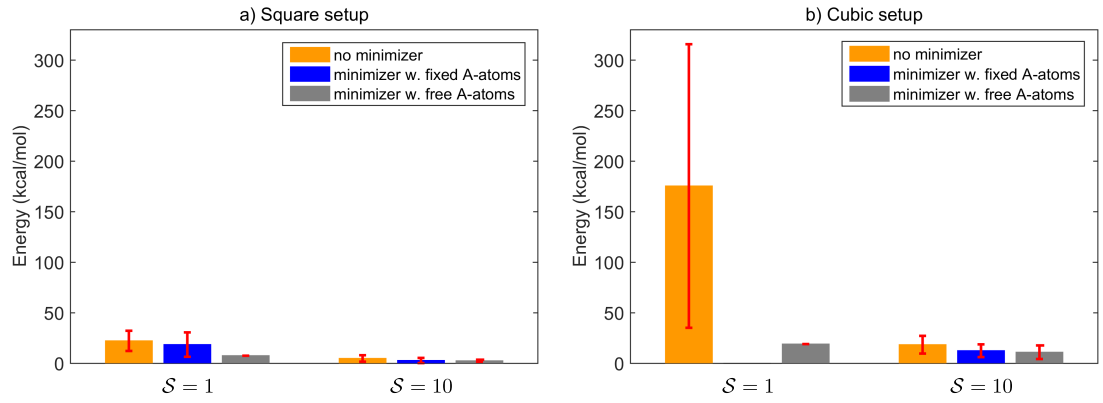


Fig. 8.6 Potential energy barriers of the valid paths found by the ART-RRT method for a) the square setup, b) the cubic setup.

The time to find an ART-RRT path for the square and cubic setups ranges from 8-200 seconds as shown in Figure 8.7. The plots show that the paths can be obtained faster without the minimization (yellow bars vs. blue and grey bars). In addition, the time is also shorter when $\mathcal{S} = 1$ than when $\mathcal{S} = 10$. This is because greater $\mathcal{S}$ allows more transition tests,

which raises the computational cost. However, greater $\mathcal{S}$ when used with the constrained minimization gives paths with lower energy barriers as discussed above. Hence, a balance between the path quality and the computational cost still needs to be decided.

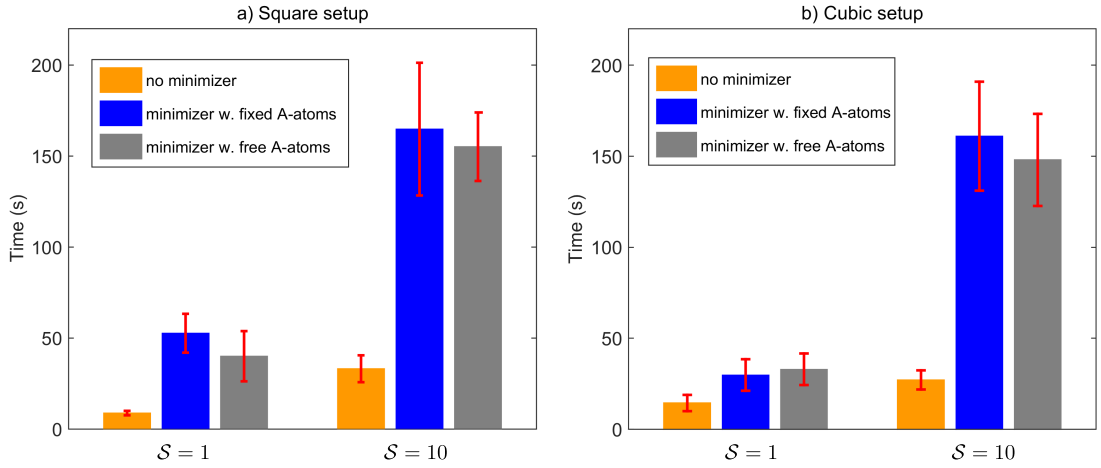


Fig. 8.7 The time for finding an ART-RRT path for the toy model with a) the square setup, b) the cubic setup.

8.1.3 Conclusion

This section showed the application of bi-directional ART-RRT on a toy model to find paths for a flexible loop with fixed ends. This experiment is a proof of concept to show that ART-RRT could be used for exploring loop motions in proteins. Additional tests on real biological systems would further explore the potential of the method for this problem. However, due to time constraints, they will be left for future investigations.

The experiments also showed the effects of several parameters used in the method. First, the sampling volume can be configured for restraining A-atoms. Secondly, increasing the maximum number of failures in transition test and using the constrained minimization improve the quality of the found paths. However, they also increase the computational time. Thirdly, the constrained minimization gives better results when the A-atoms are free than when they are fixed. Therefore, in the following sections where we examine the applications of the bi-directional ART-RRT method to find pathways for protein-ligand interactions and protein conformational transitions, the constrained minimization is always applied with free A-atoms.

8.2 Finding protein transition pathways

This section shows the capability and efficiency of the bi-directional ART-RRT method for searching protein conformational transition pathways between two given structures (a start and a goal structures). The following subsections present the processing method, the benchmarks and the results.

8.2.1 Processing method

The overall process to generate the transition pathways with ART-RRT between a pair of protein structures is adapted from the framework described in Figure 5.1 of Chapter 5.

It performs the input processing stage based on the same tools presented in Table 5.2. However, in path processing, only the bi-directional ART-RRT method is applied for path generation. The path reparation step is not needed because ART-RRT is able to give paths free from steric and ring clashes. The path optimization step is removed as well, because ART-RRT can already generate low-energy paths. Still, this option can be considered for future studies.

8.2.2 Benchmarks

The details of the benchmarks are shown in Table 8.2. We considered 7 benchmarks, each of which contains an initial and a target structure whose PDB and chain IDs are shown in the third column. The fourth column shows the total number of atoms and residues of the initial structure after the input-processing stage of the framework presented in Figure 5.1.

Table 8.2 Benchmark details for finding protein conformational transition pathways with bi-directional ART-RRT.

Benchmark ID	Protein Name	Initial/Target	No. of atoms/residues
1	Adenylate Kinase (AdK)	4AKE(A)/1AKE(A)	2085/214
2	Cyanovirin-N (CVN)	2EZM(A)/1L5E(A)	992/101
3	Maltose-binding protein	1OMP(A)/3MBP(A)	3663/370
4	5'-Nucleotidase	1HP1(A)/1HPU(C)	5123/516
5	Dengue 2 Virus Envelope Glycoprotein	1OAN(A)/1OK8(A)	3866/394
6	Spindle Assembly Checkpoint Protein	1DUJ(A)/1KLQ(A)	1934/187

We examine benchmarks 1-3 to compare the ART-RRT results with those from other state-of-the-art methods. Benchmarks 4-6 correspond to the cases where the paths generated by the ARAP interpolation and its energy-based enhancement have self-intersection (see Chapter 5). Each run of the ART-RRT method is terminated as soon as a path is found or the elapsed time reaches 10 000 seconds. For each one of benchmarks 1 to 3, ART-RRT is run

10 times, for obtaining 10 paths. For benchmarks 4 to 6, we only present one solution path to show that the ART-RRT method can find paths which are free from self-intersection.

For these benchmarks, we introduce an alignment strategy into the ART-RRT method, i.e. as soon as a new state is accepted, it is superimposed onto the start state (start conformation) to remove global rigid transforms (translation and rotation), and then, the superimposed state is added as a node into the tree. The fast quaternion-based method for structure superposition [155] is used for this task. We found that this strategy reduces the searching time, as will be shown in the Results and Discussion section.

For all the benchmarks, C_α atoms from several residues are chosen as A-atoms. The A-atoms presented here have been chosen through a trial and error process. The optimization of this selection is left for future work. After the A-atoms are selected, the same sampling volume is defined for them. This volume is a cube of 200 Å, whose center is at the center of mass considering all the A-atom positions in the start and goal conformations.

The bi-directional ART-RRT parameters are presented in Table 8.3 unless stated otherwise in the Results and Discussion section.

Table 8.3 Bi-directional ART-RRT parameters for finding protein conformational transition pathways.

Parameter description	Notation	Value
Number of ARAP iterations	m	20
Extension step in RRT tree (RRT step size)	δ	1 Å
Initial transition test temperature	T	0.001 K
Temperature factor in transition test	λ	2
Max number of failures in transition test	$\mathcal{S}$	1
Energy threshold coefficient for branch connection	γ	1
FIRE integration time step	t_F	$1 \times 10^{-15} s$
FIRE number of steps	n_F	10

8.2.3 Results and Discussion

For the path analysis, the following values are computed:

- The total positional displacement d_i of the ith residue in each path is calculated as,

$$d_i = \sum_{j=1}^{\mathcal{L}-1} \|\mathbf{p_i}(j) - \mathbf{p_i}(j-1)\| \tag{8.1}$$

where $\mathbf{p_i}(j)$ and $\mathbf{p_i}(j-1)$ are the positions of the ith C_α atom in two consecutive conformations of a path and $\mathcal{L}$ is the path size ($j=0$ and $j=\mathcal{L}-1$ correspond to the start and goal conformations, respectively).

- The total angular displacement a_i of the ith residue in each path is calculated as,

$$a_i = \sum_{j=1}^{\mathcal{L}-1} \sqrt{(\Phi_i(j)-\Phi_i(j-1))^2 + (\Psi_i(j)-\Psi_i(j-1))^2} \tag{8.2}$$

where $\Phi_i(j)$ and $\Phi_i(j-1)$ are the Φ angles of the ith residue in two consecutive conformations of the path. Likewise, $\Psi_i(j)$ and $\Psi_i(j-1)$ are the Ψ angles of the ith residue in two consecutive conformations of the path. a_i measures the total change in the dihedral angles (Φ and Ψ) of the ith residue in a path.

- The maximum RMSD distance between two consecutive conformations in each path, $RMSD_{max}$ is calculated as,

$$RMSD_{max} = \max_{j\in[1,\mathcal{L}-1]} RMSD(\mathcal{S}_{j-1}, \mathcal{S}_j) \tag{8.3}$$

where $RMSD(\mathcal{S}_{j-1}, \mathcal{S}_j)$ is the all-atom RMSD distance between two consecutive conformations. In standard RRT, a fixed extension step size gives states equally distributed along the final solution path. However, distances between conformations could vary in ART-RRT for various reasons. First, because the extension step is only applied to A-atoms. Second, the constrained minimization and alignment processes may change all the atom positions, and hence, change these distances. However, we will see that these variations are, in practice, very limited and with distances even smaller than the given RRT extension step size in the input.

Adenylate Kinase (ADK)

Adenylate kinase (ADK) is the catalyzing enzyme for the transformation among the adenine nucleotides (ATP, AMP and ADP) which stores and provides energy to cells. ADK contains 3 domains: LID (residues 122-159), NMPbind (residues 30-59), and CORE (the rest of the residues). It is suggested that the LID and NMPbind domains are mobile with respect to the CORE domain [171, 7].

The search for the transition path between the "open" conformation of ADK (4AKE chain A) and its "closed" conformation (1AKE chain A) has been performed by several robotics-inspired methods [96, 7]. The PDST-based method [96] relies on the Path-Directed

Subdivision Tree (PDST) planner with a scoring scheme for building a tree. The method rotates only selected dihedral angles for sampling new probable conformations. The NMA-RRT-based method proposed in [7], in contrast, relies on the RRT planner with normal mode analysis for sampling new conformations.

For this benchmark, we applied the bi-directional ART-RRT method (with the alignment strategy) 10 times to obtain 10 paths. The C_α atoms from the residues GLY-12 and ARG-123 are chosen as A-atoms. The average time to find each path with the ART-RRT method is only 25.93 ± 6.41 seconds. For this benchmark, the time for finding a path is reported at 3 hours 58 minutes by the PDST-based method and 0.4 hours by the NMA-RRT-based method. The potential energy and the path size (number of path conformations) is 1668.24 ± 293.81 kcal/mol and 40 ± 3, respectively. The $RMSD_{max}$ of the paths is 0.72 ± 0.23 Å, which is less than the RRT extension step size (1 Å).

We compared the ART-RRT path conformations with some experimentally solved structures of the same protein family. Table 8.4 shows the nearest distance of our path conformations with these structures. The distances reported are the all-atom RMSD computed with the *cealign* command in PyMOL [204]. This metric is the same as the one used in [77] with which we compare our results. The percentage along the path shows where the experimental structure is found in our paths. Hence, 0%, would correspond to the start conformation in the path while 100% would correspond to the goal conformations.

As shown in the table, 1AK2(A) appears first in our path; then 2RH5(A), 2RH5(B), 2RH5(C), 1DVR(A), 2AK3(A) and 1E4Y(A). The elastic network interpolation method [77] also reported 1AK2(A) to be in front of 1DVR(A), and (2RH5(A), 2RH5(B), 2RH5(C), 2AK3(A), 1E4Y(A)) to be in the same order. The order of 2RH5(A), 2RH5(B), 2RH5(C) and 1E4Y and their path percentages in our result are comparable with those found by the PDST-based method [96]. The order of 2RH5(A), 2RH5(B) and 1E4Y and their path percentages are also in accordance with the results found by the NMA-RRT-based method [7]. This shows that our methods can give the same predictions as the state-of-the-art methods, and moreover, in competitive time.

Table 8.4 The nearest distances of the ART-RRT path conformations from experimental structures. The all-atom RMSD is calculated using the *cealign* command in PyMOL [204].

PDB & chain ID	1AK2(A)	2RH5(A)	2RH5(B)	2RH5(C)	1DVR(A)	2AK3(A)	1E4Y(A)
lowest RMSD (Å)	3.04	2.04	2.02	2.40 ± 0.03	2.51 ± 0.06	1.91 ± 0.05	1.45
% along the path	2.49 ± 0.17	7.48 ± 0.50	14.95 ± 0.99	27.89 ± 2.08	60.71 ± 5.20	78.25 ± 3.41	92.76 ± 0.96

The total positional displacements of the residues are shown on the left plot of Figure 8.8. Interestingly, large-amplitude motions are found in the regions covered by the residues 30-60 and 120-155 (see also Figure 8.9), which correspond to the NMPbind and LID domains,

respectively. This observation is in agreement with those from the NMA-RRT-based method and another experimental study [171]. The residues in these regions also do not experience large deviation (shown by the shaded green color in the figure), i.e. all the paths found have the same motion for these residues. It is also interesting to notice that the residue GLY-12 which contains one A-atom does not experience large displacement.

The total angular displacements of the residues are shown on the right plot of Figure 8.8. The deviation in this plot is very small, which implies that this pattern is the same in all of the ART-RRT paths. The four highest spikes, which signify largest changes in dihedral angles, are found at residues GLY-12, GLY-46, ALA-99 and GLN-160. It is interesting that the residue ARG-123, which contains one A-atom, does not experiment large angular displacement.

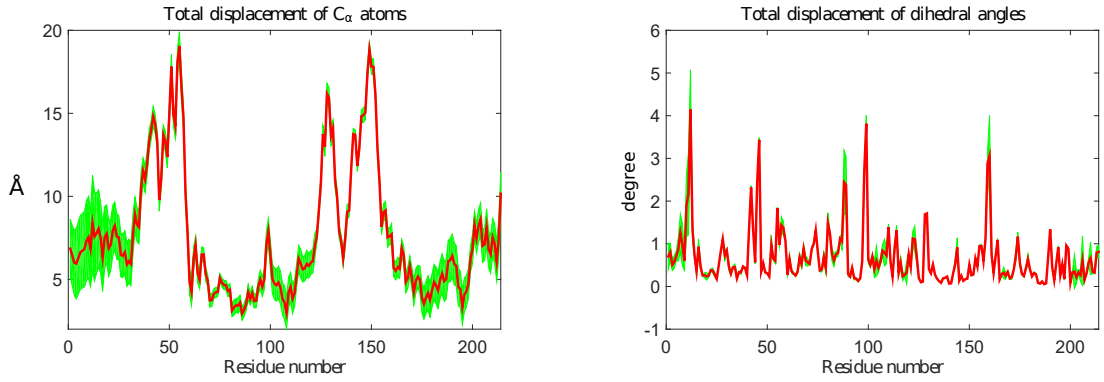


Fig. 8.8 The left plot shows the total positional displacements of the C_α atoms of ADK in the ART-RRT paths. Large motions are observed for the residues 30-60 (NMPbind domain) and 120-155 (LID domain). The right plot shows the total angular displacements of the ADK residues in the ART-RRT paths. In both plots, the red curve traces the mean value and shaded area represent the standard deviation.

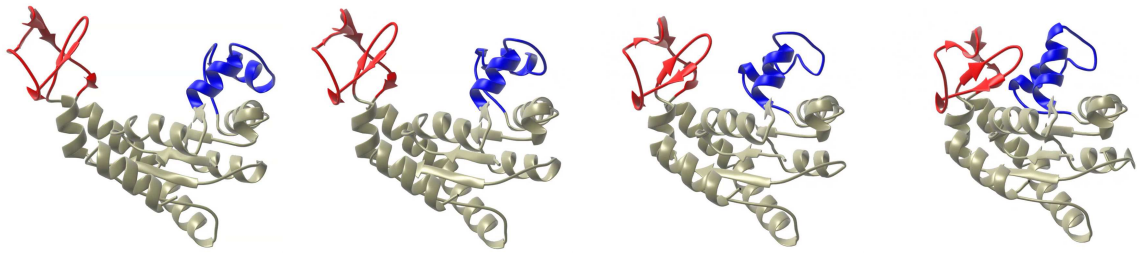


Fig. 8.9 Motion of an ART-RRT path found for ADK. Large motions are found in the LID domain (red) and NMPbind domain (blue). This observation is in agreement with those from the NMA-RRT-based method and another experimental study [171].

The fact that the largest positional and angular displacements are not necessarily found at the residues containing A-atoms implies that A-atoms only play a stimulation role. In the ART-RRT method, large structural motions are determined by the ARAP method while the constrained minimization reduces the energy of the system and removes potential steric clashes. The role of the constrained minimization is also very important because a clash even between one single pair of atoms can raise the potential energy of a conformation extremely high due to van der Waals interaction. If these states were not minimized, the method would easily accept high-energy states, and consequently, the found paths would be prone to clashes and unnatural motions.

Although the A-atoms only play the stimulation role, the number and choice of these atoms are very important because they can strongly affect the searching time and the path characteristics as shown for the next benchmark.

Cyanovirin-N (CVN)

The Cyanovirin-N protein is an inhibitor of the activities of several viruses. This includes, in particular, the human immunodeficiency virus (HIV). Therefore, CVN has been proposed for preventing the sexual transmission of HIV virus [179]. Here, we examine the transition paths from the monomer conformation (2EZM chain A) to the swapped-domain conformation (1L5E chain A). The same study has been done by other methods such as PathRover [192], SIMS [87] and a method based on the Path Directed Subdivision Tree (PDST) planner [96].

For this benchmark, we investigate the effect of the alignment strategy, as well as the number and choice of A-atoms. The experiment details are shown in Table 8.5. The average time to find a path with the ART-RRT method for all the experiments ranges from 118.31 to 1666.14 seconds, i.e. less than 30 minutes. The SIMS method reported 26 minutes to 1.3 hours and the PDST-based method reported 2.5 hours for this benchmark.

We turn on the alignment strategy for Experiment 2b and turn it off for Experiment 2a to study the effect of the alignment strategy on the ART-RRT results. These experiments also have the same set of A-atoms for a faire comparison. Table 8.5 shows that the ART-RRT paths found with the alignment strategy (Experiment 2b) have smaller path size and consume less time than those found without the alignment strategy (Experiment 2a). This is because the alignment strategy implicitly removes the translational and rotational degrees of freedom, and hence, the solution paths can be found more quickly and with smaller sizes. However, the potential energy barriers in the paths found with the alignment strategy are 21% greater than those found without the alignment strategy (2438 ± 511.38 vs. 2014.76 ± 428.50). In the rest of the experiments, we keep the alignment strategy, considering that the gain in time is generally worth the loss in energy barrier reduction. Note that, in any cases, these paths

can be postprocessed using the energy-based enhancement proposed in Chapter 5 or other sophisticated method.

Experiments 2b to 2e are used for investigating the effects of different choices of A-atoms. The same number of A-atoms (three A-atoms) is used in both experiments. As seen from the table, the solution paths from experiment 2b have smaller energy barrier, smaller size and lower computational time. Clearly, different choices of A-atoms give different performance and results. This tendency also appears when comparing the results of experiment 2c and 2d where two A-atoms are used.

The impact of the number of A-atoms on the results are analyzed from experiments 2b, 2c and 2d. As one can see, the A-atoms in Experiment 2c and 2d are subsets of those in Experiment 2b. The paths obtained from experiments 2c and 2d, where two A-atoms are used, have lower energy barriers, greater path sizes and cost much more time (3.8 - 14 times), than those obtained from Experiment 2b where 3 A-atoms are used. Experiments 2c and 2d also costs more time than Experiment 2e where 3 A-atoms are used. Hence, one can see that, at least for CVN, varying the number of A-atoms allow users to balance the trade off between computational time and quality of the final path.

Table 8.5 Selected A-atoms and statistical results in the experiments for CVN.

Experiment ID	Residues containing A-atoms ($C\alpha$)	Alignment strategy	$RMSD_{max}$ (Å)	Path size	Potential energy barrier (kcal/mol)	Time (s)
2a	LEU-1, ASP-44, GLU-101	no	1.03 ± 0.06	283 ± 68	2014.76 ± 428.50	241.20 ± 57.85
2b	LEU-1, ASP-44, GLU-101	yes	0.71 ± 0.10	201 ± 83	2438 ± 511.38	118.31 ± 27.78
2c	LEU-1, ASP-44	yes	0.88 ± 0.20	435 ± 190	1682.26 ± 166.94	1666.14 ± 1104.63
2d	ASP-44, GLU-101	yes	0.70 ± 0.17	286 ± 32	2090.86 ± 425.23	427.79 ± 225.83
2e	THR-25, ALA-64, GLY-96	yes	0.95 ± 0.32 (7.54 ± 13.51)	269 ± 45	3166.31 ± 619.11	224.10 ± 110.56

In all the experiments for CVN, the $RMSD_{max}$ is close to 1 Å which conforms with the RRT extension step size, except for Experiment 2e where this value is much greater (7.54 ± 13.51). After further analysis, we figured out that for this experiment, 2 paths out of 10 have an artifact which arises from the structure-superposition algorithm used for the alignment strategy. For certain cases, two successive conformations in a path which have small RMSD (all-atom RMSD), become strongly distant after aligned on the same reference conformation (i.e. large all-atom RMSD). This phenomenon happens, however, only rarely in our experiments (detected only for two paths and only in Experiment 2e). Therefore, the rest of the values shown for Experiment 2e in Table 8.5 are presented excluding these two paths with the artifacts.

Figure 8.10 shows the total positional and angular displacements of the residues in the ART-RRT paths for each experiment of CVN. The plots have larger deviation area (in shaded green color) than those for ADK, which means the ART-RRT methods find a greater variety of paths for CVN. This also shows that our method is capable to find different types of

pathways thanks to the underlying stochastic process. The plots in Figure 8.10 also show large positional and angular displacements at the residues near those containing the A-atoms. However, it is not necessary that the residues containing A-atoms always experience the largest positional and angular motions. An example is the total positional displacement of ASP-44 (the left plot) in Experiment 2c which is smaller than that of GLY-15 which has no A-atom, or the total positional displacement of THR-25 (the left plot) in Experiment 2e which is smaller than that of GLY-15 which has no A-atom. It shows once again that A-atoms may only play a role of stimulation.

Figure 8.11 shows several different pathways obtained from Experiments 2a and 2b. Each path is shown by a sequence of snapshots captured from the same point of view. Let us focus on the motion of the red end in the protein with respect to that of the blue end in the first three snapshots of these paths. In Experiment 2a, sequence (a) shows a path where the red end directly separates from the blue end, while sequence (b) shows the red end that passes first above the blue end. For Experiment 2b, the sequences (c) and (d) show similar motions of the red end with respect to the blue end as those in sequences (a) and (b) of Experiment 2a, respectively. In addition, sequence (e) of Experiment 2b shows the red end passing above the blue end. These result illustrates the diversity of path that we may obtain with ART-RRT.

Maltose binding protein (MBP)

Maltose binding protein is involved in active transport or chemotaxis in bacteria. The crystal structures of MBP have been discovered in ligand-free open form (1OMP chain A) and in ligand-bound closed form (pdb id 3MBP). The transition of MBP from its open form to its closed form has been simulated with accelerated MD [34] and with the SIMS method [87]. Here, we are interested in the comparison of the ART-RRT paths with those obtained from these methods. The C_α atoms from the residues GLU-172 and SER-270 are chosen as A-atoms. 10 paths were obtained from 10 runs of the ART-RRT method (with the alignment strategy). The motion of an ART-RRT path found for this case is shown in Figure 8.12.

In average, each path was obtained after only 43.28 $\pm$ 3.11 seconds which is much less than the time reported by the SIMS method for this benchmark which is 5-15 hours. With ART-RRT, the path size is 23 $\pm$ 3 conformations and the potential energy barrier is 6009.97 $\pm$ 1349.95 kcal/mol. Unfortunately, we did not find comparisons with the results from the literature for these values. Finally, the maximum distance between two consecutive path conformations is 0.83 $\pm$ 0.19 Å, which is very close to 1 Å, the prescribed RRT step size. This means the consecutive-state distance in ART-RRT paths also conforms with the RRT extension step size.

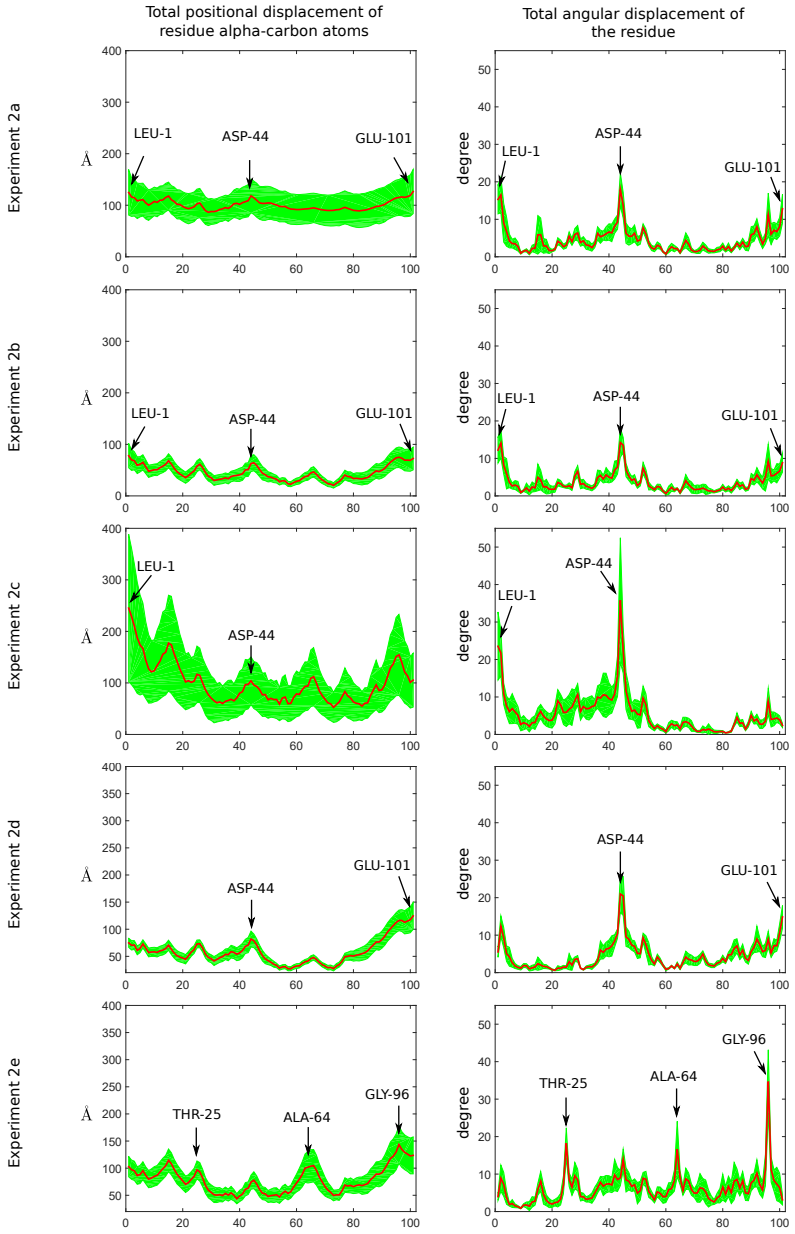


Fig. 8.10 Total positional and angular displacements of the CVN residues for all the experiments. Each experiment has two plots: the left and right ones are for the total positional and angular displacements, respectively. The residues containing A-atoms for each experiments are annotated with arrows. In general, peak values are found near the residues containing A-atoms, i.e. these residues have large motions (positional and angular). However, it is not necessary that these residues experience the largest motions, for e.g. in the left plot for Experiment 2c, ASP-44 has smaller total positional displacement than some other residues which do not contain any A-atoms.

Table 8.6 A-atoms and results for the benchmarks with self-intersections.

Experiment ID	Name	Residue containing A-atoms (C_α)	RRT step size (Å)	$RMSD_{max}$ (Å)	Path size	Potential energy barrier (kcal/mol)	Time (s)
4	5'-Nucleotidase	ASN-180, GLY-356	1	0.23	61	8141.38	137.94
5a	Dengue 2 Virus Envelope Glycoprotein	ASN-37, LEU-292, GLN-316, PHE-373	1	0.989749	89	25166.20	342.30
5b			0.5	0.46	385	8362.23	1838.74
6	Spindle Assembly Checkpoint Protein	GLY-11, ALA-19, ASP-109, GLY-173, ARG-186, SER-197	1	0.80	190	19268.61	353.75

We compared our path conformations with the NMR structures of the PDB accession code 2H25, which is supposed to be the "semi-closed" state of MBP [87]. The command *align* in PyMOL [204] is used for measuring the all-atom RMSD. We found that the closest conformations in our paths are 1.82-2.28 Å from the NMR structures which is in good agreement with the results found in [87].

Benchmarks with self-intersection

The benchmarks 4 to 6 in this section come from the benchmarks in Chapter 5 where the paths found from the ARAP interpolation method and its energy-based enhancement approach have self-intersection. For simplicity, the energy-based enhancement of the ARAP interpolation method will be called ARAPi-enhanced method in this section. For these benchmarks, we run ART-RRT (with the alignment strategy) only once for each setup, just to show that the method is able to find paths free from this problem. The A-atom choice and the results are summarized in Table 8.6.

All the experiments were done with the RRT step size of 1 Å except for Benchmark 5 where an extra experiment was performed with the RRT step size 0.5 Å (Experiment 5b), because with 1 Å, the self-intersection was not completely resolved (more details are provided below). As seen from Table 8.6, the $RMSD_{max}$ values are always smaller than the RRT extension step size, i.e. the distances between consecutive states in the ART-RRT paths conform with the RRT extension step size when considering all the atoms. The table also shows different path sizes, potential energy barriers and running time for these experiments.

Figure 8.13 shows for 5'-Nucleotidase the paths obtained either with the ARAPi-enhanced method or with the ART-RRT method. As one can see, self-intersection is totally absent in the ART-RRT path.

Figure 8.14 shows the motions of Benchmark 5 generated by the ARAPi-enhanced method and the ART-RRT method. The figure shows a self-intersection in the ARAPi-enhanced path. The self-intersection problem in the ART-RRT path at the RRT extension step size of 1 Å is less visible (see Figure 8.15 for more details) while the self-intersection problem in the ART-RRT path at the RRT step size of 0.5 Å is totally absent. This is because

self-intersections are more likely to occur for large step size and 1 Å is slightly too coarse in this case. Note that the running time and path size of the path at 0.5 Å are greater (5.4 times greater in time and 4.3 times greater in path size) than those of the the path at 1 Å . However, the potential energy barrier is 3 times smaller due to the absence of self-intersection.

Figure 8.16 shows the ART-RRT motion for Benchmark 6 in contrast with the motion found by the ARAPi-enhanced method. As one can see, when using ART-RRT, large-amplitude motions of the blue-end loop and the red-end loop avoid collisions with the green beta sheet and hence self-intersections do not occur.

We also applied the NEB method to optimize the ART-RRT paths which are free from self-intersection, i.e. paths obtained from experiments 4, 5b and 6. As shown in Figure 8.17, a significant decrease in the potential energy barriers of the optimized paths (blue bars) is observed compared with the non-optimized paths (orange bars). The figure also shows lower energy barriers in the optimized ART-RRT paths (blue bars) than in the optimized ARAP-interpolation paths, i.e. ARAPi-enhanced paths (grey bars).

8.2.4 Conclusion

This section has shown the application of the bi-directional ART-RRT method for generating protein conformational transition pathways. Two types of benchmarks were presented: those studied by other state-of-the-art methods and those where the energy improvement of the ARAP interpolation method could not deal with the problem of self-intersection. As shown by the results, ART-RRT can find for these problems low-energy, clash-free, and self-intersection-free paths in efficient time. The distances between consecutive conformations in the ART-RRT path conforms with the RRT extension step size also for all the atoms, although this restraint is only applied for the A-atoms. The experiments also show that although the A-atoms only play a role of stimulation for structural motions, the number and choice of these atoms can affect the performance of the method and the characteristics of the solution paths. In summary, the success of ART-RRT when applied to the benchmarks subjected to self-intersections show that the method is a good alternative to ARAP interpolation when such a problem is likely to happen.

One interesting direction of work would be to design a method for automatically selecting or suggesting A-atoms to the users. Another improvement could be a fix for the artifact (though observed very rarely) caused by the alignment strategy, which comes from the chosen superposition method.

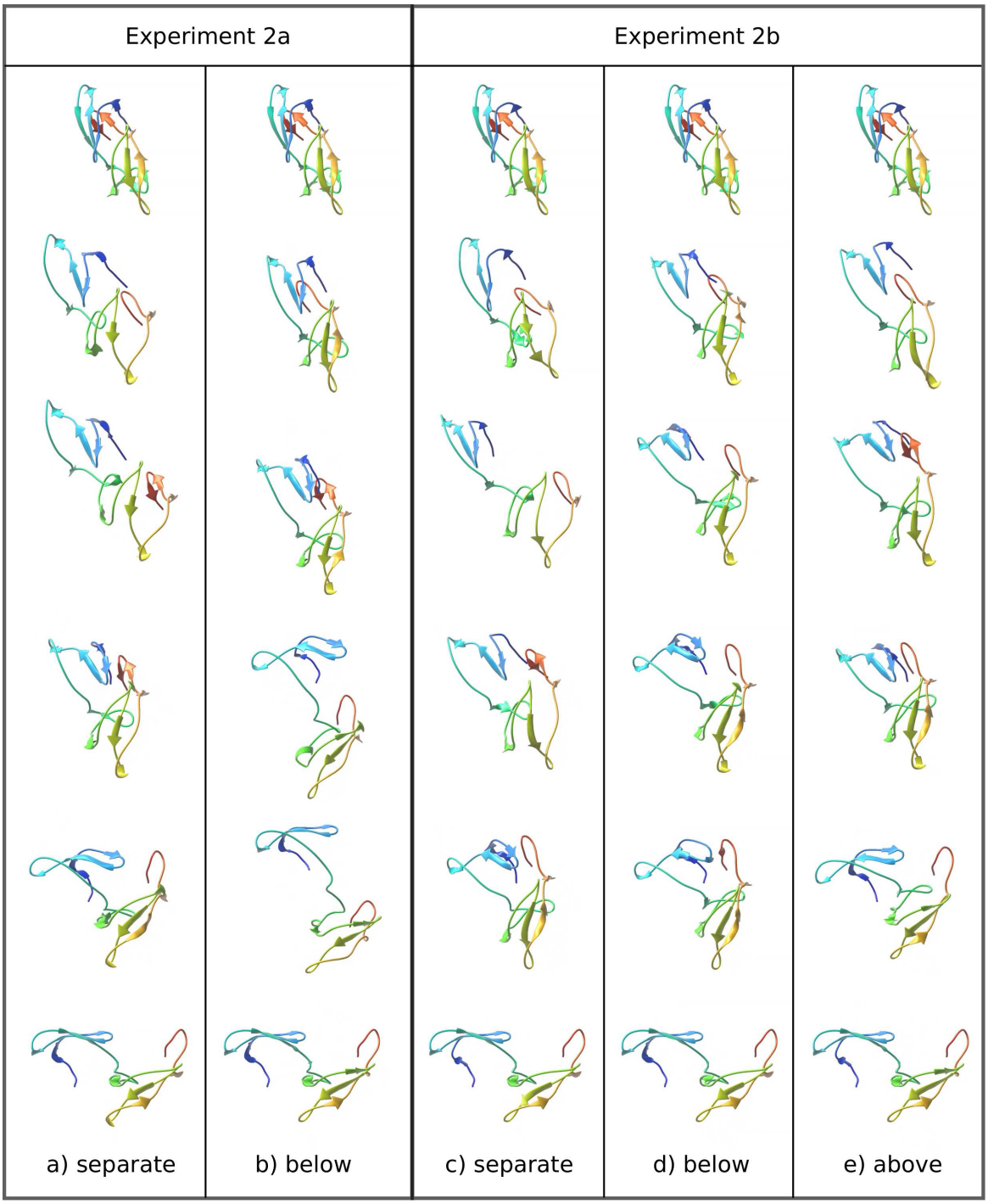


Fig. 8.11 Motions of several ART-RRT paths found for CVN from Experiment 2a and 2b. The ART-RRT methods found a variety of paths. For example, in the first 3 snapshots in the paths of Experiment 2a, the red end is separated from the blue end (sequence a) and passes below the blue end (sequence b). In the first 3 snapshots of Experiment 2b, the red end is separated from (sequence c), passes below (sequence d) and passes above (sequence e) the blue end.

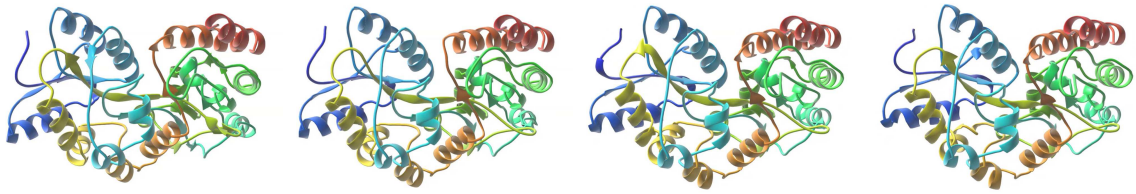


Fig. 8.12 Motion of an ART-RRT path found for MBP.

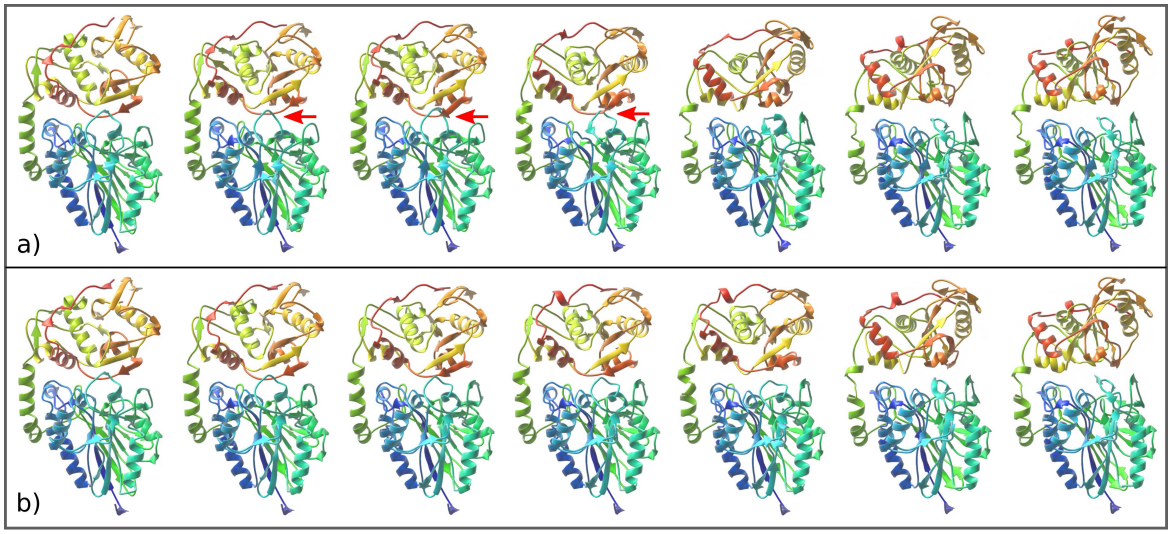


Fig. 8.13 Snapshots of the paths obtained for 5'-Nucleotidase with a) the ARAPi-enhanced method and b) ART-RRT. In the ARAPi-enhanced path (sequence a), the self-intersection occurs between the light blue loop and the light red loop as highlighted with the red arrows. In the ART-RRT path (sequence b), these loops avoid the problem by slipping on each other.

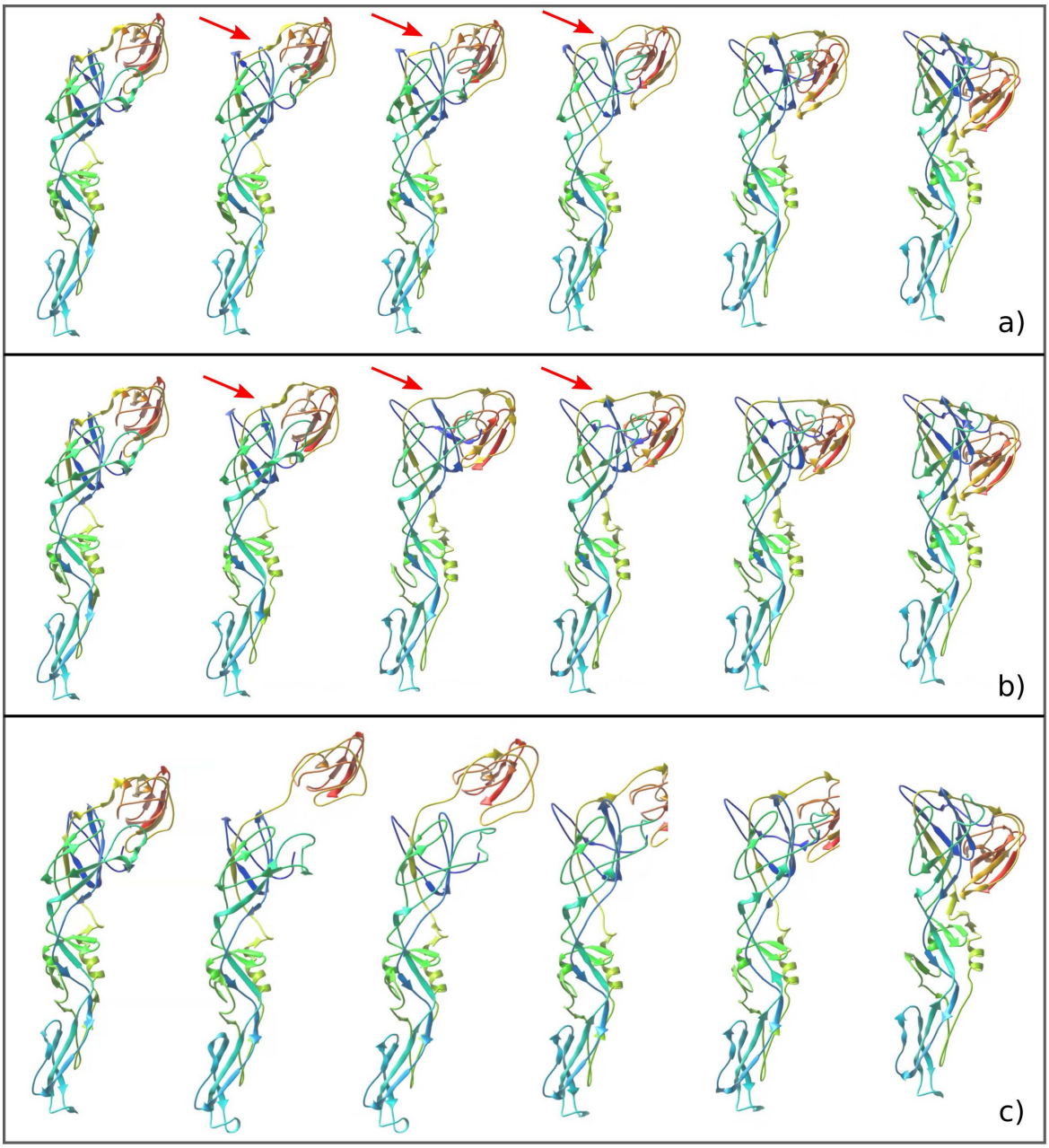


Fig. 8.14 Snapshots from the paths obtained by the ARAPi-enhanced method (sequence a), the ART-RRT method with RRT step size of 1 Å (sequence b), and the ART-RRT method with RRT step size of 0.5 Å (sequence c). In the ARAPi-enhanced path (sequence a), the self-intersection occurs at the yellow loop (pointed by the red arrows) which crosses the blue loop twice. This problem is less severe for the ART-RRT path at the RRT extension step size of 1 Å (sequence b) and is completely absent for the ART-RRT path at the RRT extension step size of 0.5 Å (sequence c).

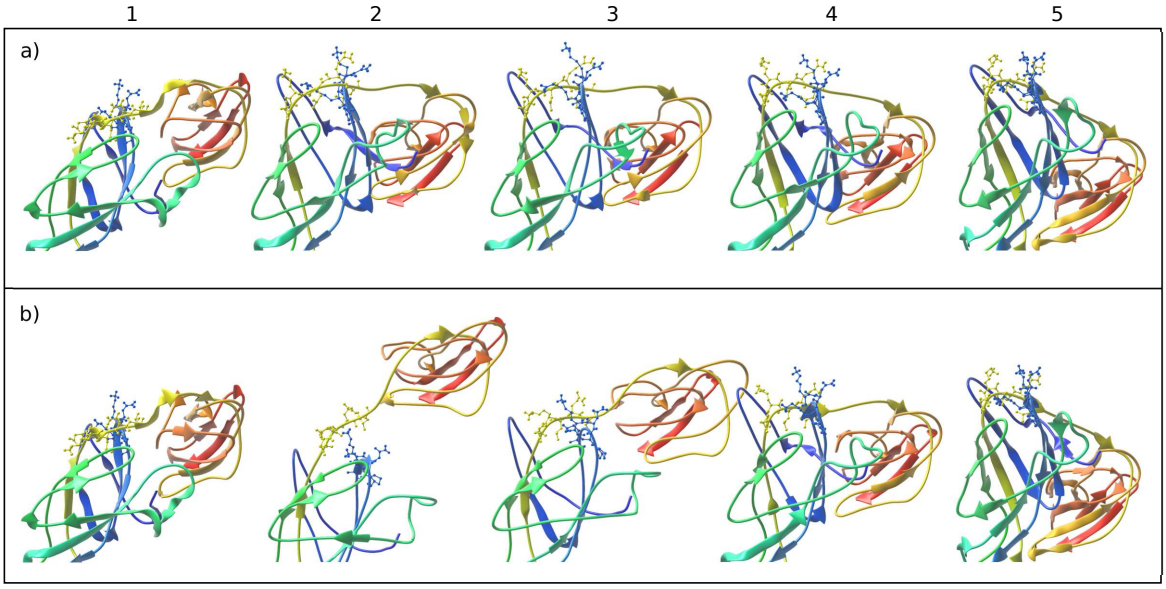


Fig. 8.15 Closer view on the ART-RRT paths for Benchmark 5 shows that self-intersection is still present for the ART-RRT path obtained at the RRT extension step size of 1 Å (sequence a): the yellow-loop atoms are in front the blue-loop atoms in the snapshot a2, then cross behind the blue-loop atoms in the snapshot a3. The problem is not present for the ART-RRT path obtained at the RRT extension step size of 0.5 Å (sequence b).

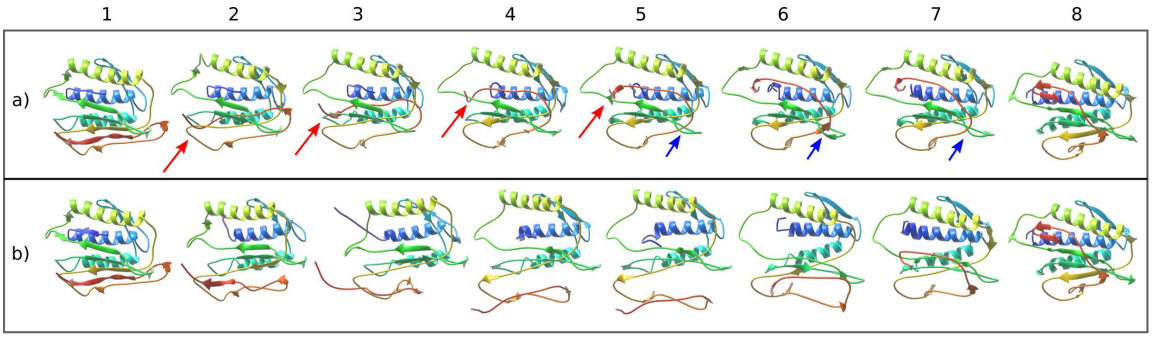


Fig. 8.16 Snapshots from the paths obtained by the ARAPi-enhanced method (sequence a) and by the ART-RRT method (sequence b) for Spindle Assembly Checkpoint Protein (Benchmark 6). In the ARAP-interpolation path (sequence a), self-intersections of the red-end loop are highlighted by red arrows: the red end loop crosses the yellow loop (transition from a2 to a3) the green loops (transitons from a3 to a4 and from a4 to a5). The self-intersections of the orange loop are highlighted by blue arrows: the loop crosses the green beta sheet two times (transitions from a5 to a6 and from a6 to a7). In the ART-RRT path, self-intersections are totally absent.

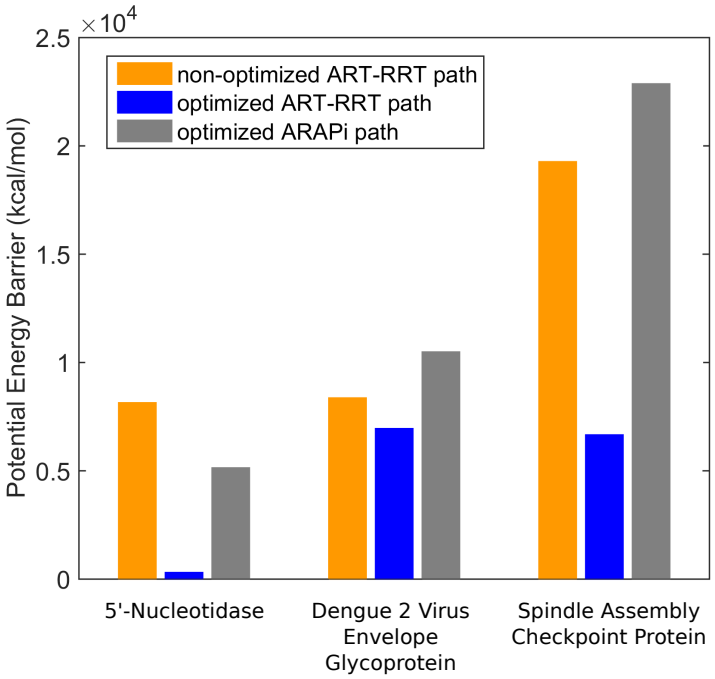


Fig. 8.17 Potential energy barriers of the non-optimized ART-RRT paths (orange bars), optimized ART-RRT paths (blue bars) and the optimized ARAP-interpolation paths (i.e. ARAPi-enhanced paths, grey bars). The energy barriers are lower for the optimized ART-RRT paths than for the non-optimized ART-RRT paths. The energy barriers are also lower for the optimized ART-RRT paths than for the optimized ARAP-interpolation paths because the ART-RRT paths are free from self-intersection.

8.3 Finding protein-ligand interaction pathways

The purpose of this section is twofold. First, it shows that bi-directional ART-RRT can also be adapted for finding protein-ligand pathways given a bound and an unbound state. Secondly, it shows that by placing A-atoms on both the protein and the ligand, exploring large motions for both molecules is possible.

8.3.1 Benchmarks

The system that we consider is a sugar binding protein with maltotriose. The bound state of the protein and ligand is taken from the PDB id 2GHA, chain A. Figure 8.18a shows the bound state of the protein-ligand complex where the ligand is held tightly inside the protein. The unbound state of the protein is modeled after the PDB id 2GHB, chain A. The input processing follows the same framework proposed in Chapter 5 and illustrated in Figure 5.1, with the same tools presented in Table 5.2. Molecular topologies were generated by the *pdb2gmx* command in GROMACS for the protein and with the PRODRG server [205] for the ligand. To design an unbound state, the ligand structure is displaced (by translation) outside of the binding pocket to a desired position in space while the protein structure is modeled after the PDB id 2GHB, chain A. Finally, the bound and unbound states are minimized using the FIRE minimimizer with the same parameters as those shown in Table 7.3. Here, we design two unbound states for the system. Figure 8.19a shows the unbound target state A where the ligand is displaced outside and near the binding pocket. Figure 8.19b shows the unbound target state B where the ligand is displaced outside and at the back of the binding pocket. Figure 8.19c shows the initial bound state.

8.3.2 Results

We run the bi-directional ART-RRT method for finding a path from the initial state (bound state) to one of the target states (unbound states). The protein has two A-atoms that are the C_α atoms of the residues ILE-16 and GLN-212. The ligand molecule also has two A-atoms that are the oxygen atoms in green colors in Figure 8.18b.

Four experiments are considered (see Table 8.7), with target states A or B and a maximum number of failures in transition test δ equal to 1 or 2. The other parameters remain fixed and have the same values as those shown in Table 7.3.

In all the experiments, the same sampling volume is used for all the A-atoms of the protein and ligand. This volume is a cube centered at the center of mass considering the

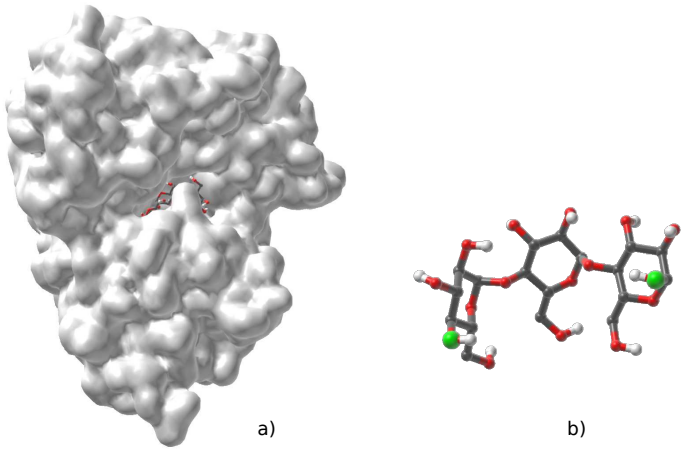


Fig. 8.18 a) The bound state modelled from PDB id 2GHA chain A. The protein is represented by Gaussian surface while the ligand is represented by sticks and balls. One can see that the ligand is held tightly inside the protein. b) The closer view of the ligand with A-atoms in green colors.

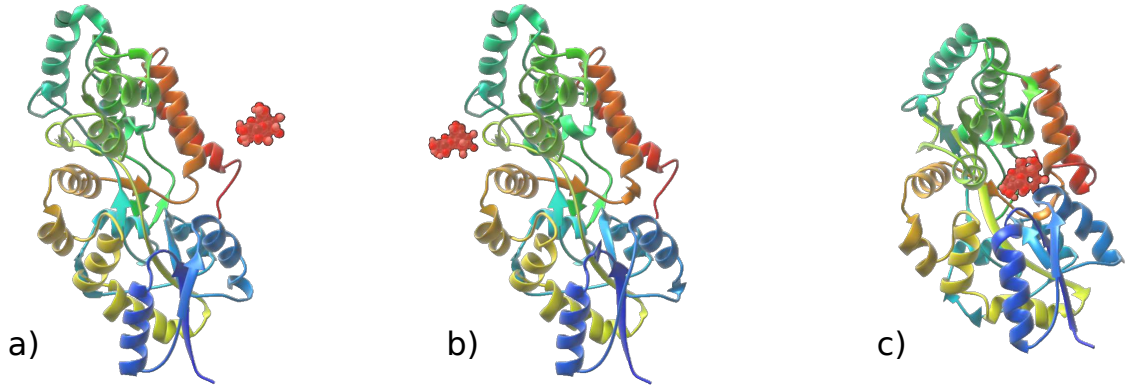


Fig. 8.19 The target states and initial state used in the experiment. The protein is respresented by ribbons and the ligand is represented by van der Waals balls: a) target state A, where the ligand is displaced outside, however, still stays close to the binding pocket, b) target state B, where the ligand is displaced behind the protein. c) initial state (protein-ligand bound state)

positions of all the A-atoms (of the protein and ligand) in the initial and target states of each experiment.

Because our goal here is only to assess the ability of the method in finding a reasonable path, we do not present statistics based on averages, and hence, the bi-directional ART-RRT is run only once per experiment.

Figure 8.20 shows the solutions found for all the experiments in Table 8.7. The paths found for Experiments A1 and A2 are relatively similar because the ligand position in target

Table 8.7 Experiments for finding protein-ligand pathways with bi-directional ART-RRT.

Experiment	Target state id	Maximum number of failures for transition test $\mathcal{S}$
A1	A	1
A2	A	2
B1	B	1
B2	B	2

state A is close to its position in the bound state. However, the paths found for Experiment B1 and B2 are totally different. The ligand in Experiment B1 passes through the protein body while the ligand in Experiment B2 makes a roundabout on the protein surface to reach the target state. After careful examination, all the paths were found to be clash-free and without any self-intersections.

Some experiment results are shown in Table 8.8. The table shows that experiments A1 and B1 where $\mathcal{S} = 1$ take less time than experiments A2 and B2 where $\mathcal{S} = 2$, respectively, because the former tend to have fewer transition tests due to smaller $\mathcal{S}$. However, greater $\mathcal{S}$ does not necessarily means lower potential energy barriers of the paths found. Hence, the energy barrier is higher in Experiment B1 where $\mathcal{S} = 2$, than in Experiment A1 where $\mathcal{S} = 1$ and the opposite result is found for experiments B2 and A2. A further analysis of the influence of these parameters would require more runs to obtain averages. However, due to time constraints, this task is left for future work.

Table 8.8 Experiment results of the protein-ligand paths found with bi-directional ART-RRT.

Experiment id	Path size	Energy barrier (kcal/mol)	Time (s)
A1	46	6353.9	211
A2	71	14808.7	847
B1	114	12367.6	834
B2	214	8226.7	2807

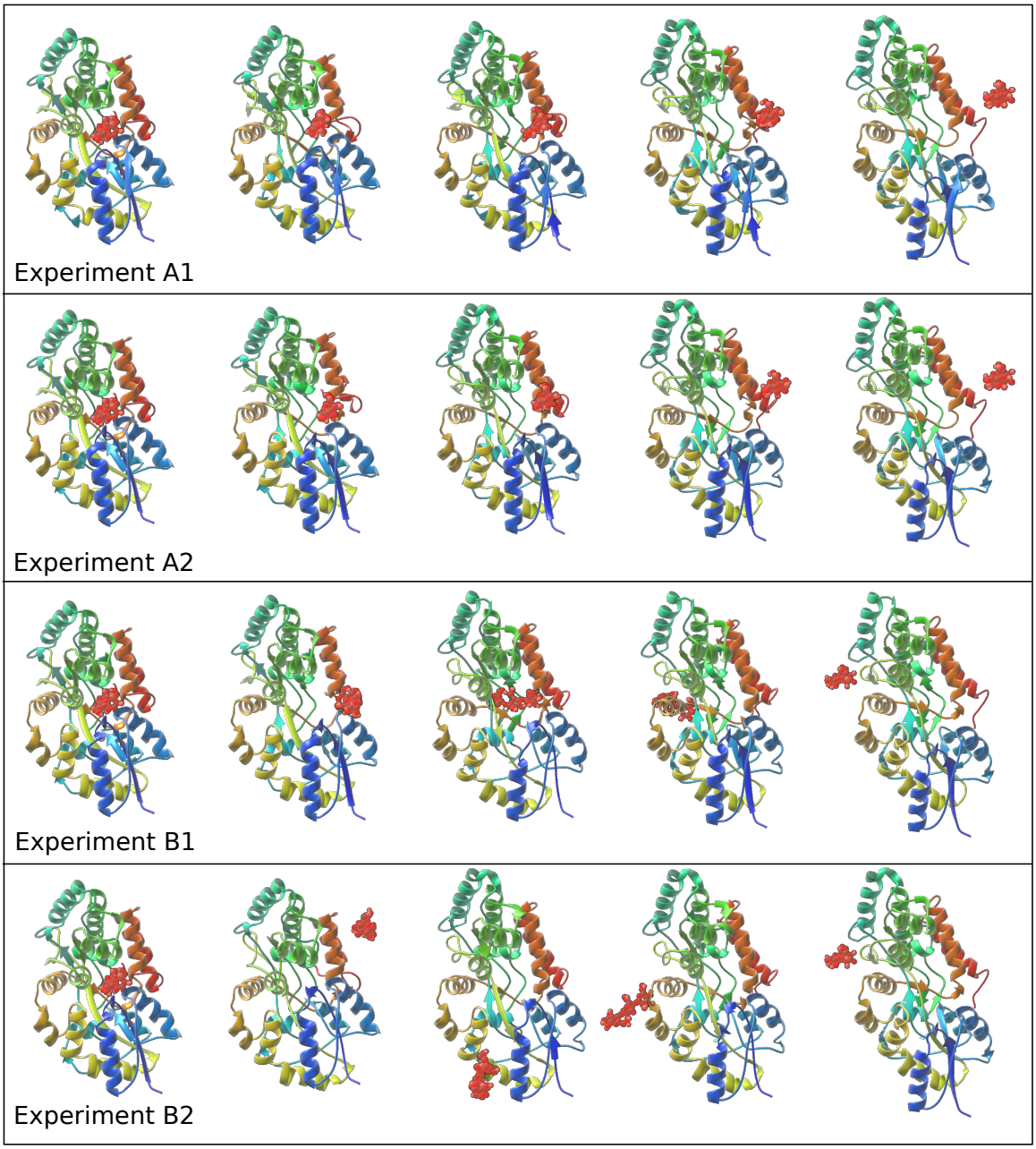


Fig. 8.20 Protein-ligand paths found by bi-directional ART-RRT. The paths found for Experiment A1 and A2 are very similar because the ligand positions in the initial and target states are close to each other. The paths found for Experiment B1 and B2 are totally different. In Experiment B1, the ligand slightly moves out of the binding pocket, unfold itself, then crosses the protein body. In contrast, the ligand in Experiment B2 make a roundabout on the protein surface.

Part IV

Conclusion and Perspective

Conclusion

In this dissertation, two novel methods were proposed for finding molecular pathways, mostly applied either on conformational changes of proteins or on the unbinding of ligands from proteins.

The first proposed method, based on the As-Rigid-As-Possible (ARAP) principle from computer graphics, generates interpolation paths at low computational cost, while preserving the local rigidity of the original structure. The obtained paths were shown to have reasonable motions, i.e. with small variations in bond lengths and bond angles and in good agreement with the results from other studies.

An enhancement of this method was also proposed to obtain low-energy molecular pathways. In this context, a general framework was presented to generate Minimum-Energy Paths (MEP) from two given biomolecular structures. This framework is robust enough to work with raw inputs having missing hydrogen atoms, non-optimized structures and potential mutations. We applied this framework for generating MEP paths from three different path-generation methods including the proposed ARAP interpolation. The results showed that the potential energy barriers were significantly lower for the optimized paths generated with the ARAP interpolation method than for those generated with the two other methods.

Despite an improvement in the path quality, the energy-based enhancement has only localized effects. Hence, for complex cases, even the ARAP interpolation paths may encounter problems such as self-intersections that cannot be repaired. This led us to the development of the second proposed method, ART-RRT, which combines the ARAP principle for reducing the dimensionality and handling the structure flexibility, with the Rapidly-exploring Random Tree (RRT) from robotics for exploring possible pathways. The ARAP modeling (ARAPm) method is integrated in mono-directional ART-RRT for extending branches of an exploration tree through the control of only few selected atoms. In addition, in bi-directional ART-RRT, the ARAP interpolation (ARAPi) method is used for connecting two exploration trees.

The exploration capabilities of mono-directional ART-RRT were first demonstrated on the simple dialanine benchmark. This benchmark also highlighted the interest of combining the transition test and constrained minimization to efficiently guide the exploration toward low-energy regions. The method was then applied for finding ligand-unbinding pathways for several benchmarks. We have shown that it was able to find at low computational cost a variety of pathways that were in good agreement with results from other methods.

Bi-directional ART-RRT was applied first to a toy model, showing the potential of the method to explore the conformational space of protein loops. The method was then used to find protein conformational transition pathways for well-known benchmarks and for those where self-intersection was detected in the optimized ARAP interpolation paths. The

results showed that the paths had similar characteristics with other state-of-the-art solutions. Moreover, they were also low-energy, clash-free, and, for a small enough step size, without self-intersection. Finally, we illustrated with an example the capabilities of bi-directional ART-RRT for finding pathways of protein-ligand systems.

All the conducted experiments have shown that both versions of ART-RRT are time-efficient and in most of the cases robust in the sense that the paths were found with low-energy and in good agreement with either experimental data or other state-of-the-art solutions.

The methods presented in this thesis will be made accessible to the scientific community through three modules in the SAMSON software platform: one for ARAP interpolation only, one for mono-directional ART-RRT to find ligand-unbinding pathways, and one for bi-directional ART-RRT to find protein conformational transition pathways.

Future work

Other potential applications

The ARAP interpolation and ART-RRT methods offer two choices for generating biomolecular pathways. The first one is fast and gives satisfying motions in simple cases. However, it may not handle complex scenarios where large deformations are required and self intersections easily occur. The ART-RRT method is computationally more expensive, yet it gives a variety of paths to choose from. Moreover, these paths are of higher quality, i.e. clash-free, and without self-intersection. We have seen how these paths can be post-processed with path optimization methods such as the Nudged Elastic Band. It would be interesting to test these paths with some other methods such as the String method.

In the future, we would also like to use the paths generated by the proposed methods as first guesses for more complex approaches such as the Weighted Histogram Analysis Method (WHAM) [140] or the Transition Path Sampling (TPS) [59, 61] for estimating free-energy differences and predicting reaction rate constants.

Section 8.1 has shown the potential application of bi-directional ART-RRT for sampling protein loops. This problem is very well-known because loops are active parts of protein interacting with other molecules. Hence, many methods, especially robotics-inspired ones [35, 162, 145, 228], have been applied for studying them. We would like to further examine the performance of the ART-RRT method when compared with these existing approaches.

Section 8.3 has shown that bi-directional ART-RRT can be used for finding protein-ligand pathways even when the target state is not easily accessible from the initial state. In the

future, it would be worth conducting a large-scale analysis of the method on this type of problem.

In Section 8.3, a scenario was proposed where A-atoms were placed on both the protein and ligand. This opens up the possibility of simulating protein-protein interactions since A-atoms could be placed on two or several proteins at the same time for exploring their large-amplitude motions.

Future investigations and improvements

Besides the applications investigated in this thesis study and the variety of other potential applications, there is still room for future analysis and improvements of the proposed methods.

For the ARAP interpolation method, several enhancements can be considered. The edge and cell weights could be varied according to some extra information about the local flexibility. For example, edge weights can be placed more on double bonds than single bonds because double bonds are more rigid. In ARAP meshes, extra edges could be added to represent hydrogen bonds or other long-distance interactions to enforce the rigidity of these parts. The cell definition can also be extended to include more elements than the one-ring neighbor of the current setting to impose the rigidity on larger-size cells. Finally, more sophisticated techniques as the one proposed in [123] could be used to overcome the restriction of the rotation angle in the current rotation interpolation method, hence, allow larger-amplitude motions.

Although all the experiments with the ART-RRT method were done in vaccuum, the results were found in good agreement with experimental data and other state-of-the-art solutions. However, the extension of the ART-RRT methods for solvated systems would be interesting because these systems are more realistic. The atoms of the solvent molecules such as waters and ions could be assigned as N-atoms so that their motions could be adapted based on the motions of the protein or ligand by the constrained minimization.

Although the effects of several parameters in the ART-RRT methods were already analyzed in Section 7.1 and Section 8.1, more investigations are needed to get a deeper understanding of their impacts. Ideally, a method which optimally tunes these parameters and is adapted for specific systems would be desirable. Similarly, an automatic and optimal selection of the A-atoms would be beneficial for the user. Such a method would have to balance the number of A-atoms and the computational cost because more A-atoms would give more structural flexibility but would also imply a higher-dimensional search space. The placement of A-atoms at hinge locations is an interesting possibility. The proposed methods in [211, 74, 127, 216, 70, 82] could be applied for hinge detection. The analysis of the

structural differences between the initial and target conformations can also give hints about the locations for A-atoms.

References

[1] Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25.

[2] Abramson, J., Smirnova, I., Kasho, V., Verner, G., Kaback, H. R., and Iwata, S. (2003). Structure and mechanism of the lactose permease of Escherichia coli. *Science*, 301(5633):610–615.

[3] Aci-Sèche, S., Genest, M., and Garnier, N. (2011). Ligand entry pathways in the ligand binding domain of PPARγ receptor. *FEBS Letters*, 585(16):2599–2603.

[4] Adcock, S. A. and McCammon, J. A. (2006). Molecular dynamics: survey of methods for simulating the activity of proteins. *Chemical reviews*, 106(5):1589–1615.

[5] Adhikari, A. N., Peng, J., Wilde, M., Xu, J., Freed, K. F., and Sosnick, T. R. (2012). Modeling large regions in proteins: Applications to loops, termini, and folding. *Protein Science*, 21(1):107–121.

[6] Al-Bluwi, I., Siméon, T., and Cortés, J. (2012). Motion planning algorithms for molecular simulations: A survey. *Computer Science Review*, 6(4):125–143.

[7] Al-Bluwi, I., Vaisset, M., Siméon, T., and Cortés, J. (2013). Modeling protein conformational transitions by a combination of coarse-grained normal mode analysis and robotics-inspired methods. *BMC structural biology*, 13(1):S2.

[8] Alberts, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2013). *Essential cell biology*. Garland Science.

[9] Alder, B. J. and Wainwright, T. (1959). Studies in molecular dynamics. I. General method. *The Journal of Chemical Physics*, 31(2):459–466.

[10] Alexa, M., Cohen-Or, D., and Levin, D. (2000). As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164. ACM Press/Addison-Wesley Publishing Co.

[11] Amato, N. M., Dill, K. A., and Song, G. (2003). Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *Journal of Computational Biology*, 10(3-4):239–255.

[12] Amato, N. M. and Song, G. (2002). Using motion planning to study protein folding pathways. *Journal of Computational Biology*, 9(2):149–168.

[13] Apaydin, M. S., Brutlag, D. L., Guestrin, C., Hsu, D., Latombe, J.-C., and Varma, C. (2003). Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. *Journal of Computational Biology*, 10(3-4):257–281.

[14] Bahar, I., Atilgan, A. R., and Erman, B. (1997). Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding and Design*, 2(3):173–181.

[15] Banaszak, L. J. (2000). *Foundations of structural biology*. Academic Press.

[16] Barbe, S., Cortés, J., Siméon, T., Monsan, P., Remaud-Siméon, M., and André, I. (2011). A mixed molecular modeling-robotics approach to investigate lipase large molecular motions. *Proteins: Structure, Function, and Bioinformatics*, 79(8):2517–2529.

[17] Barducci, A., Bonomi, M., and Parrinello, M. (2011). Metadynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):826–843.

[18] Baxter, W., Barla, P., and Anjyo, K.-i. (2008). Rigid shape interpolation using normal equations. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 59–64. ACM.

[19] Becker, K. E. and Fichthorn, K. A. (2006). Accelerated molecular dynamics simulation of the thermal desorption of n-alkanes from the basal plane of graphite. *The Journal of chemical physics*, 125(18):184706.

[20] Behn, A., Zimmerman, P. M., Bell, A. T., and Head-Gordon, M. (2011a). Efficient exploration of reaction paths via a freezing string method. *The Journal of chemical physics*, 135(22):224108.

[21] Behn, A., Zimmerman, P. M., Bell, A. T., and Head-Gordon, M. (2011b). Incorporating linear synchronous transit interpolation into the growing string method: Algorithm and applications. *Journal of chemical theory and computation*, 7(12):4019–4025.

[22] Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J. D., and Zardecki, C. (2002). The Protein Data Bank. *Acta Crystallographica Section D*, 58:899–907.

[23] Binder, K. (1987). Monte Carlo methods. *Quantum Monte Carlo Methods*, page 241.

[24] Bitzek, E., Koskinen, P., Gähler, F., Moseler, M., and Gumbsch, P. (2006). Structural relaxation made simple. *Physical review letters*, 97(17):170201.

[25] Bofill, J. M. (1994). Updated Hessian matrix and the restricted step method for locating transition structures. *Journal of Computational Chemistry*, 15(1):1–11.

[26] Bofill, J. M. (2003). Remarks on the updated Hessian matrix methods. *International journal of quantum chemistry*, 94(6):324–332.

[27] Booth, A. G. (2001). Visualizing protein conformational changes on a personal computer–alpha carbon pseudo bonding as a constraint for interpolation in internal coordinate space. *Journal of Molecular Graphics and Modelling*, 19(6):481–486.

[28] Borosán, P., Howard, R., Zhang, S., and Nealen, A. (2010). Hybrid mesh editing. In *Eurographics (short papers)*, pages 41–44.

[29] Botsch, M., Pauly, M., Gross, M. H., and Kobbelt, L. (2006). PriMo: Coupled Prisms for Intuitive Surface Modeling. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 11–20, Aire-la-Ville, Switzerland. Eurographics Association.

[30] Botsch, M. and Sorkine, O. (2008). On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230.

[31] Brooks, B. R., Brooks, C. L., MacKerell, A. D., Nilsson, L., Petrella, R. J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S., et al. (2009). CHARMM: the biomolecular simulation program. *Journal of computational chemistry*, 30(10):1545–1614.

[32] Brooks, S. P. and Morgan, B. J. (1995). Optimization using simulated annealing. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 44(2):241–257.

[33] Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90.

[34] Bucher, D., Grant, B. J., Markwick, P. R., and McCammon, J. A. (2011). Accessing a Hidden Conformation of the Maltose Binding Protein Using Accelerated Molecular Dynamics. *PLOS Computational Biology*, 7(4):1–10.

[35] Canutescu, A. A. and Dunbrack, R. L. (2003). Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein science*, 12(5):963–972.

[36] Capelli, A. M. and Costantino, G. (2014). Unbinding pathways of VEGFR2 inhibitors revealed by steered molecular dynamics. *Journal of chemical information and modeling*, 54(11):3124–3136.

[37] Carlsson, P., Burendahl, S., and Nilsson, L. (2006). Unbinding of retinoic acid from the retinoic acid receptor by random expulsion molecular dynamics. *Biophysical journal*, 91(9):3151–3161.

[38] Case, D. A., Cheatham, T. E., Darden, T., Gohlke, H., Luo, R., Merz, K. M., Onufriev, A., Simmerling, C., Wang, B., and Woods, R. J. (2005). The Amber biomolecular simulation programs. *Journal of computational chemistry*, 26(16):1668–1688.

[39] Castellana, N. E., Lushnikov, A., Rotkiewicz, P., Sefcovic, N., Pevzner, P. A., Godzik, A., and Vyatkina, K. (2013). MORPH-PRO: a novel algorithm and web server for protein morphing. *Algorithms for Molecular Biology*, 8(1):1–9.

[40] Cerjan, C. J. and Miller, W. H. (1981). On finding transition states. *The Journal of Chemical Physics*, 75(6):2800–2806.

[41] Chao, I., Pinkall, U., Sanan, P., and Schröder, P. (2010). A simple geometric model for elastic deformations. *ACM transactions on graphics (TOG)*, 29(4):38.

[42] Chen, S.-Y., Gao, L., Lai, Y.-K., and Xia, S. (2017). Rigidity controllable as-rigid-as-possible shape deformation. *Graphical Models*, 91(Supplement C):13 – 21.

[43] Chiang, T.-H., Apaydin, M. S., Brutlag, D. L., Hsu, D., and Latombe, J.-C. (2006). *Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation*, pages 410–424. Springer Berlin Heidelberg, Berlin, Heidelberg.

[44] Choi, J. and Szymczak, A. (2003). On Coherent Rotation Angles for As-Rigid-As-Possible Shape Interpolation. In *Proceedings of the 15th Canadian Conference on Computational Geometry (CCCG'03)*, pages 111–114.

[45] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementation*. A Bradford Book.

[46] Chou, K.-C. (1988). Low-frequency collective motion in biomacromolecules and its biological functions. *Biophysical chemistry*, 30(1):3–48.

[47] Ciccotti, G., Ferrario, M., and Schuette, C. (2014). *Molecular Dynamics Simulation*. MDPI AG.

[48] Cohen, P. (2002). Protein kinases — the major drug targets of the twenty-first century? *Nature reviews Drug discovery*, 1(4):309–315.

[49] Connors, K. A. (1990). *Chemical kinetics: the study of reaction rates in solution*. John Wiley & Sons.

[50] Cortés, J., Jaillet, L., and Siméon, T. (2008). Disassembly path planning for complex articulated objects. *IEEE Transactions on Robotics*, 24(2):475–481.

[51] Cortés, J., Le, D. T., Iehl, R., and Siméon, T. (2010). Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method. *Physical Chemistry Chemical Physics*, 12(29):8268–8276.

[52] Cortés, J., Siméon, T., Remaud-Siméon, M., and Tran, V. (2004). Geometric algorithms for the conformational analysis of long protein loops. *Journal of computational chemistry*, 25(7):956–967.

[53] Creighton, T. E. (1979). Experimental studies of protein folding and unfolding. *Progress in Biophysics and Molecular Biology*, 33(Supplement C):231 – 297.

[54] Cui, Q. and Bahar, I. (2005). *Normal mode analysis: theory and applications to biological and chemical systems*. CRC press.

[55] Cuno, A., Esperança, C., Oliveira, A., and Cavalcanti, P. R. (2007). 3D as-rigid-as-possible deformations using MLS. In *Proceedings of the 27th computer graphics international conference*, pages 115–122.

[56] Cuzzolin, A., Sturlese, M., Deganutti, G., Salmaso, V., Sabbadin, D., Ciancetta, A., and Moro, S. (2016). Deciphering the complexity of ligand–protein recognition pathways using supervised molecular dynamics (SuMD) simulations. *Journal of chemical information and modeling*, 56(4):687–705.

[57] Daura, X., Mark, A. E., and Van Gunsteren, W. F. (1998). Parametrization of aliphatic CHn united atoms of GROMOS96 force field. *Journal of computational chemistry*, 19(5):535–547.

[58] Day, R. and Daggett, V. (2003). All-atom simulations of protein folding and unfolding. *Advances in protein chemistry*, 66:373–403.

[59] Dellago, C. (2007). Transition Path Sampling and the Calculation of Free Energies. In Chipot, C. and Pohorille, A., editors, *Free Energy Calculations: Theory and Applications in Chemistry and Biology*, pages 249–276. Springer-Verlag Berlin Heidelberg, 1 edition.

[60] Dellago, C. and Bolhuis, P. G. (2009). *Transition Path Sampling and Other Advanced Simulation Techniques for Rare Events*, pages 167–233. Springer Berlin Heidelberg, Berlin, Heidelberg.

[61] Dellago, C., Bolhuis, P. G., Csajka, F. S., and Chandler, D. (1998). Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics*, 108(5):1964–1977.

[62] Dennis Jr, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM.

[63] Devaurs, D., Bouard, L., Vaisset, M., Zanon, C., Al-Bluwi, I., Iehl, R., Siméon, T., and Cortés, J. (2013a). MoMA-LigPath: a web server to simulate protein–ligand unbinding. *Nucleic acids research*, 41(W1):W297–W302.

[64] Devaurs, D., Molloy, K., Vaisset, M., Shehu, A., Siméon, T., and Cortés, J. (2015). Characterizing energy landscapes of peptides using a combination of stochastic algorithms. *IEEE transactions on nanobioscience*, 14(5):545–552.

[65] Devaurs, D., Shehu, A., Siméon, T., and Cortés, J. (2014a). Sampling-based methods for a full characterization of energy landscapes of small peptides. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 37–44.

[66] Devaurs, D., Siméon, T., and Cortés, J. (2014b). A multi-tree extension of the Transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 2991–2996.

[67] Devaurs, D., Vaisset, M., Siméon, T., and Cortés, J. (2013b). A multi-tree approach to compute transition paths on energy landscapes. In *Workshop on Artificial Intelligence and Robotics Methods in Computational Biology, AAAI '13*, pages pp. 8–13, Bellevue, United States.

[68] Dickson, A. and Lotz, S. D. (2017). Multiple Ligand Unbinding Pathways and Ligand-Induced Destabilization Revealed by WExplore. *Biophysical Journal*, 112(4):620–629.

[69] Doye, J. P. K., Miller, M. A., and Wales, D. J. (1999). Evolution of the potential energy surface with size for lennard-jones clusters. *The Journal of Chemical Physics*, 111(18):8417–8428.

[70] Dziubiński, M., Daniluk, P., and Lesyng, B. (2015). ResiCon: a method for the identification of dynamic domains, hinges and interfacial regions in proteins. *Bioinformatics*, 32(1):25–34.

[71] Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

[72] Echols, N., Milburn, D., and Gerstein, M. (2003). MolMovDB: analysis and visualization of conformational change and structural flexibility. *Nucleic Acids Research*, 31(1):478–482.

[73] Edgar, R. C. (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, 5(1):1.

[74] Emekli, U., Schneidman-Duhovny, D., Wolfson, H. J., Nussinov, R., and Haliloglu, T. (2008). HingeProt: automated prediction of hinges in protein structures. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1219–1227.

[75] Farkas, Ö. and Schlegel, H. B. (1999). Methods for optimizing large molecules. II. Quadratic search. *The Journal of Chemical Physics*, 111(24):10806–10814.

[76] Farrell, D. W., Speranskiy, K., and Thorpe, M. (2010). Generating stereochemically acceptable protein pathways. *Proteins: Structure, Function, and Bioinformatics*, 78(14):2908–2921.

[77] Feng, Y., Yang, L., Kloczkowski, A., and Jernigan, R. L. (2009). The energy profiles of atomic conformational transition intermediates of adenylate kinase. *Proteins: Structure, Function, and Bioinformatics*, 77(3):551–558.

[78] Fichthorn, K. A. and Mubin, S. (2015). Hyperdynamics made simple: Accelerated molecular dynamics with the Bond-Boost method. *Computational Materials Science*, 100:104–110.

[79] Fletcher, R. (1970). A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322.

[80] Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154.

[81] Floater, M. S., Kós, G., and Reimers, M. (2005). Mean value coordinates in 3D. *Computer Aided Geometric Design*, 22(7):623–631.

[82] Fotoohifiroozabadi, S., Mohamad, M. S., and Deris, S. (2017). NAHAL-Flex: A Numerical and Alphabetical Hinge Detection Algorithm for Flexible Protein Structure Alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP(99):1–1.

[83] Freddolino, P. L., Arkhipov, A. S., Larson, S. B., McPherson, A., and Schulten, K. (2006). Molecular dynamics simulations of the complete satellite tobacco mosaic virus. *Structure*, 14(3):437–449.

[84] Fröhlich, S. and Botsch, M. (2011). Example-Driven Deformations Based on Discrete Shells. *Computer Graphics Forum*, 30(8):2246–2257.

[85] Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical optimization*. Academic Press.

[86] Gipson, B., Hsu, D., Kavraki, L. E., and Latombe, J.-C. (2012). Computational models of protein kinematics and dynamics: Beyond simulation. *Annual review of analytical chemistry*, 5:273–291.

[87] Gipson, B., Moll, M., and Kavraki, L. E. (2013). SIMS: A Hybrid Method for Rapid Conformational Analysis. *PLOS One*, 8(7):1–12.

[88] Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26.

[89] Grubmüller, H. (1995). Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Physical Review E*, 52(3):2893.

[90] Guennebaud, G., Jacob, B., et al. (2010). Eigen v3. http://eigen.tuxfamily.org.

[91] Guex, N. and Peitsch, M. C. (1997). SWISS-MODEL and the Swiss-Pdb Viewer: An environment for comparative protein modeling. *ELECTROPHORESIS*, 18(15):2714–2723.

[92] Guo, H., Fu, X., Chen, F., Yang, H., Wang, Y., and Li, H. (2008). As-rigid-as-possible shape deformation and interpolation. *Journal of Visual Communication and Image Representation*, 19(4):245–255.

[93] Halgren, T. A. and Lipscomb, W. N. (1977). The synchronous-transit method for determining reaction pathways and locating molecular transition states. *Chemical Physics Letters*, 49(2):225–232.

[94] Haliloglu, T., Bahar, I., and Erman, B. (1997). Gaussian dynamics of folded proteins. *Physical review letters*, 79(16):3090.

[95] Hansmann, U. H. and Okamoto, Y. (1999). New Monte Carlo algorithms for protein folding. *Current opinion in structural biology*, 9(2):177–183.

[96] Haspel, N., Moll, M., Baker, M. L., Chiu, W., and Kavraki, L. E. (2010). Tracing conformational changes in proteins. *BMC structural biology*, 10(1):S1.

[97] Heeren, B., Rumpf, M., Wardetzky, M., and Wirth, B. (2012). Time-Discrete Geodesics in the Space of Shells. *Computer Graphics Forum*, 31(5):1755–1764.

[98] Henkelman, G., Jóhannesson, G., and Jónsson, H. (2002). *Methods for Finding Saddle Points and Minimum Energy Paths*, pages 269–302. Springer Netherlands, Dordrecht.

[99] Henkelman, G. and Jónsson, H. (1999). A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *The Journal of chemical physics*, 111(15):7010–7022.

[100] Henkelman, G. and Jónsson, H. (2000). Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *The Journal of chemical physics*, 113(22):9978–9985.

[101] Henkelman, G., Uberuaga, B. P., and Jónsson, H. (2000). A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of chemical physics*, 113(22):9901–9904.

[102] Hermann, J. R. (2006). *Protein and the Body*. Division of Agricultural Sciences and Natural Resources, Oklahoma State University.

[103] Hoare, M. (1979). Structure and dynamics of simple microclusters. *Advances in chemical physics*, 40(1979):49–135.

[104] Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

[105] Horn, B. K., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135.

[106] Huang, C. C., Couch, G. S., Pettersen, E. F., and Ferrin, T. E. (1996). Chimera: an extensible molecular modeling application constructed using standard components. In *Pacific Symposium on Biocomputing '96: Hawaii, USA*, volume 1, page 724. World Scientific.

[107] Huang, J., Tong, Y., Zhou, K., Bao, H., and Desbrun, M. (2011). Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):983–992.

[108] Hubbard, S. R. (2004). Juxtamembrane autoinhibition in receptor tyrosine kinases. *Nature Reviews Molecular Cell Biology*, 5(6):464–471.

[109] Huber, T., Torda, A. E., and van Gunsteren, W. F. (1994). Local elevation: a method for improving the searching properties of molecular dynamics simulation. *Journal of computer-aided molecular design*, 8(6):695–708.

[110] Hummer, G. and Kevrekidis, I. G. (2003). Coarse molecular dynamics of a peptide fragment: Free energy, kinetics, and long-time dynamics computations. *The Journal of chemical physics*, 118(23):10762–10773.

[111] Igarashi, T., Moscovich, T., and Hughes, J. F. (2005). As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)*, 24(3):1134–1141.

[112] Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM.

[113] INRIA. (2017). SAMSON: Software for Adaptive Modeling and Simulation Of Nanosystems. Version 0.6.0.

[114] Izrailev, S., Stepaniants, S., Isralewitz, B., Kosztin, D., Lu, H., Molnar, F., Wriggers, W., and Schulten, K. (1999). Steered molecular dynamics. In *Computational molecular dynamics: challenges, methods, ideas*, pages 39–65. Springer.

[115] Jaillet, L., Corcho, F. J., Pérez, J.-J., and Cortés, J. (2011). Randomized tree construction algorithm to explore energy landscapes. *Journal of computational chemistry*, 32(16):3464–3474.

[116] Jaillet, L., Cortés, J., and Siméon, T. (2010). Sampling-based path planning on configuration-space costmaps. *IEEE Transactions on Robotics*, 26(4):635–646.

[117] Jamal, M. S., Parveen, S., Beg, M. A., Suhail, M., Chaudhary, A. G. A., Damanhouri, G. A., Abuzenadah, A. M., and Rehan, M. (2014). Anticancer Compound Plumbagin and Its Molecular Targets: A Structural Insight into the Inhibitory Mechanisms Using Computational Approaches. *PLOS ONE*, 9(2):1–12.

[118] Jensen, M. Ø., Yin, Y., Tajkhorshid, E., and Schulten, K. (2007). Sugar transport across lactose permease probed by steered molecular dynamics. *Biophysical journal*, 93(1):92–102.

[119] Jin, H., Zhu, J., Dong, Y., and Han, W. (2016). Exploring the different ligand escape pathways in acylaminoacyl peptidase by random acceleration and steered molecular dynamics simulations. *RSC Advances*, 6(13):10987–10996.

[120] Jónsson, H., Mills, G., and Jacobsen, K. W. (1998). *Nudged elastic band method for finding minimum energy paths of transitions*, chapter 16, pages 385–404. World Scientific.

[121] Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *Journal of the American Chemical Society*, 118(45):11225–11236.

[122] Ju, T., Schaefer, S., and Warren, J. (2005). Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics*, 24(3):561–566.

[123] Kaji, S. (2016). *Tetrisation of Triangular Meshes and Its Application in Shape Blending*, pages 7–19. Springer Singapore, Singapore.

[124] Kavraki, L. E., Kolountzakis, M. N., and Latombe, J.-C. (1998). Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1):166–171.

[125] Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580.

[126] Kearsley, S. K. (1989). On the orthogonal transformation used for structural comparisons. *Acta Crystallographica Section A: Foundations of Crystallography*, 45(2):208–210.

[127] Keating, K. S., Flores, S. C., Gerstein, M. B., and Kuhn, L. A. (2009). StoneHinge: hinge prediction by network analysis of individual protein structures. *Protein Science*, 18(2):359–371.

[128] Kevrekidis, I. G., Gear, C. W., Hyman, J. M., Kevrekidid, P. G., Runborg, O., Theodoropoulos, C., et al. (2003). Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762.

[129] Kilian, M., Mitra, N. J., and Pottmann, H. (2007). Geometric modeling in shape space. *ACM Transactions on Graphics (TOG)*, 26(3):64.

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

[131] Kirillova, S., Cortés, J., Stefaniu, A., and Siméon, T. (2008). An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins. *Proteins: Structure, Function, and Bioinformatics*, 70(1):131–143.

[132] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

[133] Kleywegt, G. J. and Jones, T. A. (1995). Where freedom is given, liberties are taken. *Structure*, 3(6):535–540.

[134] Kleywegt, G. J. and Jones, T. A. (1996). Phi/psi-chology: Ramachandran revisited. *Structure*, 4(12):1395–1400.

[135] Kolinski, A. and Skolnick, J. (1994). Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme. *Proteins: Structure, Function, and Bioinformatics*, 18(4):338–352.

[136] Kosztin, D., Izrailev, S., and Schulten, K. (1999). Unbinding of retinoic acid from its receptor studied by steered molecular dynamics. *Biophysical journal*, 76(1):188–197.

[137] Krebs, W. G. and Gerstein, M. (2000). The morph server: a standardized system for analyzing and visualizing macromolecular motions in a database framework. *Nucleic Acids Research*, 28(8):1665–1675.

[138] Krieger, E. and Vriend, G. (2015). New ways to boost molecular dynamics simulations. *Journal of computational chemistry*, 36(13):996–1007.

[139] Kuffner, J. J. and LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 995–1001.

[140] Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The Weighted Histogram Analysis Method for Free-Energy Calculations on Biomolecules. I. The Method. *Journal of Computational Chemistry*, 13(8):1011–1021.

[141] Latombe, J.-C. (1991). *Robot Motion Planning*. Springer US.

[142] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. TR 98–11, Computer Science Department., Iowa State University.

[143] LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

[144] Lavalle, S. M., Kuffner, J. J., and Jr. (2000). Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308.

[145] Lee, J., Lee, D., Park, H., Coutsias, E. A., and Seok, C. (2010). Protein loop modeling by using fragment assembly and analytical loop closure. *Proteins: Structure, Function, and Bioinformatics*, 78(16):3428–3436.

[146] Lemkul, J. A. (2017). GROMACS tutorials. *GROMACS Tutorials*.

[147] Levi, Z. and Gotsman, C. (2015). Smooth rotation enhanced as-rigid-as-possible mesh animation. *IEEE transactions on visualization and computer graphics*, 21(2):264–277.

[148] Lewis, J. P., Cordner, M., and Fong, N. (2000). Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co.

[149] Li, H., Chang, Y.-Y., Yang, L.-W., and Bahar, I. (2015). *i*GNM 2.0: the Gaussian network model database for biomolecular structural dynamics. *Nucleic acids research*, 44(D1):D415–D422.

[150] Lifson, S. (1983). *Potential Energy Functions for Structural Molecular Biology*, pages 1–44. Springer US, Boston, MA.

[151] Lin, M.-H., Hsu, H.-J., Bartenschlager, R., and Fischer, W. B. (2014). Membrane undulation induced by NS4A of Dengue virus: a molecular dynamics simulation study. *Journal of Biomolecular Structure and Dynamics*, 32(10):1552–1562.

[152] Lindahl, E., Azuara, C., Koehl, P., and Delarue, M. (2006). NOMAD-Ref: visualization, deformation and refinement of macromolecular structures based on all-atom normal mode analysis. *Nucleic Acids Research*, 34(suppl_2):W52–W56.

[153] Lipman, Y., Sorkine, O., Levin, D., and Cohen-Or, D. (2005). Linear Rotation-invariant Coordinates for Meshes. *ACM Transactions on Graphics*, 24(3):479–487.

[154] Lipsky, M. S. and Sharp, L. K. (2001). From idea to market: the drug approval process. *The Journal of the American Board of Family Practice*, 14(5):362–367.

[155] Liu, P., Agrafiotis, D. K., and Theobald, D. L. (2010). Fast determination of the optimal rotational matrix for macromolecular superpositions. *Journal of computational chemistry*, 31(7):1561–1563.

[156] Liu, Y.-S., Yan, H.-B., and Martin, R. R. (2011). As-rigid-as-possible surface morphing. *Journal of Computer Science and Technology*, 26(3):548–557.

[157] Lomakin, A., Asherie, N., and Benedek, G. B. (1996). Monte Carlo study of phase separation in aqueous protein solutions. *The Journal of chemical physics*, 104(4):1646–1656.

[158] Lüdemann, S. K., Lounnas, V., and Wade, R. C. (2000). How do substrates enter and products exit the buried active site of cytochrome P450cam? 1. Random expulsion molecular dynamics investigation of ligand access channels and mechanisms. *Journal of molecular biology*, 303(5):797–811.

[159] Machado-Charry, E., Béland, L. K., Caliste, D., Genovese, L., Deutsch, T., Mousseau, N., and Pochet, P. (2011). Optimized energy landscape exploration using the ab initio based activation-relaxation technique. *The Journal of chemical physics*, 135(3):034102.

[160] Mackerell, A. D. (2004). Empirical force fields for biological macromolecules: overview and issues. *Journal of computational chemistry*, 25(13):1584–1604.

[161] MacKerell Jr, A. D., Bashford, D., Bellott, M., Dunbrack Jr, R. L., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., et al. (1998). All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The journal of physical chemistry B*, 102(18):3586–3616.

[162] Mandell, D. J., Coutsias, E. A., and Kortemme, T. (2009). Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nature methods*, 6(8):551–552.

[163] Marinica, M.-C., Willaime, F., and Mousseau, N. (2011). Energy landscape of small clusters of self-interstitial dumbbells in iron. *Physical Review B*, 83(9):094119.

[164] Maximova, T., Plaku, E., and Shehu, A. (2015). Computing transition paths in multiple-basin proteins with a probabilistic roadmap algorithm guided by structure data. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 35–42.

[165] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

[166] Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341.

[167] Moll, M., Schwarz, D., and Kavraki, L. E. (2008). *Roadmap methods for protein folding*, pages 219–239. Humana Press, Totowa, NJ.

[168] Monhemi, H., Housaindokht, M. R., Moosavi-Movahedi, A. A., and Bozorgmehr, M. R. (2014). How a protein can remain stable in a solvent with high content of urea: insights from molecular dynamics simulation of Candida antarctica lipase B in urea: choline chloride deep eutectic solvent. *Physical Chemistry Chemical Physics*, 16(28):14882–14893.

[169] Montalenti, F., Sørensen, M., and Voter, A. (2001). Closing the gap between experiment and theory: Crystal growth by temperature accelerated dynamics. *Physical review letters*, 87(12):126101.

[170] Mousseau, N., Béland, L. K., Brommer, P., Joly, J.-F., El-Mellouhi, F., Machado-Charry, E., Marinica, M.-C., and Pochet, P. (2012). The activation-relaxation technique: ART nouveau and kinetic ART. *Journal of Atomic, Molecular, and Optical Physics*, 2012.

[171] Müller, C., Schlauderer, G., Reinstein, J., and Schulz, G. E. (1996). Adenylate kinase motions during catalysis: an energetic counterweight balancing substrate binding. *Structure*, 4(2):147–156.

[172] Murtagh, B. A. and Sargent, R. W. (1970). Computational experience with quadratically convergent minimisation methods. *The Computer Journal*, 13(2):185–194.

[173] Nandi, P. (1996). Protein conformation and disease. *Veterinary research*, 27(4-5):373–382.

[174] Nguyen, M. K., Jaillet, L., and Redon, S. (2017). As-Rigid-As-Possible molecular interpolation paths. *Journal of Computer-Aided Molecular Design*, 31(4):403–417.

[175] Nguyen, M. K., Jaillet, L., and Redon, S. (2018). ART-RRT: As-Rigid-As-Possible exploration of ligand unbinding pathways. *Journal of Computational Chemistry*, pages n/a–n/a.

[176] Nocedal, J. (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782.

[177] Novinskaya, A., Devaurs, D., Moll, M., and Kavraki, L. E. (2017). Defining Low-Dimensional Projections to Guide Protein Conformational Sampling. *Journal of Computational Biology*, 24(1):79–89.

[178] Paries, N., Degener, P., and Klein, R. (2007). Simple and efficient mesh editing with consistent local frames. In *15th Pacific Conference on Computer Graphics and Applications, 2007. PG'07*, pages 461–464.

[179] Pauwels, R. and De Clercq, E. (1996). Development of vaginal microbicides for the prevention of heterosexual transmission of HIV. *JAIDS Journal of Acquired Immune Deficiency Syndromes*, 11(3):211–221.

[180] Peng, C. and Bernhard Schlegel, H. (1993). Combining Synchronous Transit and Quasi-Newton Methods to Find Transition States. *Israel Journal of Chemistry*, 33(4):449–454.

[181] Peräkylä, M. (2009). Ligand unbinding pathways from the vitamin D receptor studied by molecular dynamics simulations. *European Biophysics Journal*, 38(2):185–198.

[182] Peters, B., Heyden, A., Bell, A. T., and Chakraborty, A. (2004). A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. *The Journal of chemical physics*, 120(17):7877–7886.

[183] Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., and Ferrin, T. E. (2004). UCSF Chimera — a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612.

[184] Polak, E. (1971). *Computational methods in optimization: a unified approach*. Academic press.

[185] Ponder, J. W. and Case, D. A. (2003). Force Fields for Protein Simulations. In *Protein Simulations*, volume 66 of *Advances in Protein Chemistry*, pages 27 – 85. Academic Press.

[186] Porta, J. M. and Jaillet, L. (2013). Exploring the energy landscapes of flexible molecular loops using higher-dimensional continuation. *Journal of computational chemistry*, 34(3):234–244.

[187] Putz, I. and Brock, O. (2017). Elastic network model of learned maintained contacts to predict protein motion. *PLOS ONE*, 12(8):1–46.

[188] Quapp, W. and Heidrich, D. (1984). Analysis of the concept of minimum energy path on the potential energy surface of chemically reacting systems. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, 66(3):245–260.

[189] Quapp, W., Hirsch, M., Imig, O., and Heidrich, D. (1998). Searching for saddle points of potential energy surfaces by following a reduced gradient. *Journal of computational chemistry*, 19(9):1087–1100.

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

[191] Ramakrishnan, C. (2001). Ramachandran and his map. *Resonance*, 6(10):48–56.

[192] Raveh, B., Enosh, A., Schueler-Furman, O., and Halperin, D. (2009). Rapid Sampling of Molecular Motions with Prior Information Constraints. *PLOS Computational Biology*, 5(2):1–17.

[193] Rehan, M., Beg, M. A., Parveen, S., Damanhouri, G. A., and Zaher, G. F. (2014). Computational Insights into the Inhibitory Mechanism of Human AKT1 by an Orally Active Inhibitor, MK-2206. *PLOS ONE*, 9(10):1–12.

[194] Ren, W. and Vanden-Eijnden, E. (2013). A climbing string method for saddle point search. *The Journal of chemical physics*, 138(13):134105.

[195] Ren, W., Vanden-Eijnden, E., Maragakis, P., and E, W. (2005). Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide. *The Journal of chemical physics*, 123(13):134109.

[196] Roth, C.-A., Dreyfus, T., Robert, C. H., and Cazals, F. (2016). Hybridizing rapidly exploring random trees and basin hopping yields an improved exploration of energy landscapes. *Journal of computational chemistry*, 37(8):739–752.

[197] Šali, A. and Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*, 234(3):779–815.

[198] Scheraga, H. A., Khalili, M., and Liwo, A. (2007). Protein-folding dynamics: overview of molecular simulation techniques. *Annual Review of Physical Chemistry*, 58(1):57–83. PMID: 17034338.

[199] Schlegel, H. B. (2011). Geometry optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):790–809.

[200] Schlick, T. (2010). *Molecular Modeling and Simulation: An Interdisciplinary Guide*, volume 21. Springer-Verlag New York.

[201] Schlitter, J., Engels, M., Krüger, P., Jacoby, E., and Wollmer, A. (1993). Targeted molecular dynamics simulation of conformational change-application to the T $\leftrightarrow$ R transition in insulin. *Molecular Simulation*, 10(2-6):291–308.

[202] Schneider, E., Dai, L., Topper, R. Q., Drechsel-Grau, C., and Tuckerman, M. E. (2017). Stochastic neural network approach for learning high-dimensional free energy surfaces. *Physical Review Letters*, 119(15):150601.

[203] Schröder, G. F., Brunger, A. T., and Levitt, M. (2007). Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. *Structure*, 15(12):1630–1641.

[204] Schrödinger, LLC (2015). The PyMOL Molecular Graphics System, Version 1.8.

[205] Schüttelkopf, A. W. and van Aalten, D. M. F. (2004). *PRODRG*: a tool for high-throughput crystallography of protein–ligand complexes. *Acta Crystallographica Section D*, 60(8):1355–1363.

[206] Scott, W. R., Hünenberger, P. H., Tironi, I. G., Mark, A. E., Billeter, S. R., Fennen, J., Torda, A. E., Huber, T., Krüger, P., and van Gunsteren, W. F. (1999). The GROMOS biomolecular simulation program package. *The Journal of Physical Chemistry A*, 103(19):3596–3607.

[207] Selkoe, D. J. (2003). Folding proteins in fatal ways. *Nature*, 426(6968):900.

[208] Shakhnovich, E., Farztdinov, G., Gutin, A., and Karplus, M. (1991). Protein folding bottlenecks: A lattice Monte Carlo simulation. *Physical review letters*, 67(12):1665.

[209] Shanker, S. and Bandyopadhyay, P. (2011). Monte Carlo temperature basin paving with effective fragment potential: An efficient and fast method for finding low-energy structures of water clusters (H2O) 20 and (H2O) 25. *The Journal of Physical Chemistry A*, 115(42):11866–11875.

[210] Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*, 24(111):647–656.

[211] Shatsky, M., Nussinov, R., and Wolfson, H. J. (2002). Flexible protein alignment and hinge detection. *Proteins: Structure, Function, and Bioinformatics*, 48(2):242–256.

[212] Sheffer, A. and Kraevoy, V. (2004). Pyramid coordinates for morphing and deformation. In *Proceedings of 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, pages 68–75.

[213] Shehu, A. and Kavraki, L. E. (2012). Modeling structures and motions of loops in protein molecules. *Entropy*, 14(2):252–290.

[214] Sheppard, D., Terrell, R., and Henkelman, G. (2008). Optimization methods for finding minimum energy paths. *The Journal of chemical physics*, 128(13):134106.

[215] Sherman, J. and Morrison, W. J. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127.

[216] Shibuya, T. (2010). Fast hinge detection algorithms for flexible protein structures. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(2):333–341.

[217] Shoemake, K. (1985). Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM.

[218] Simons, J., Joergensen, P., Taylor, H., and Ozment, J. (1983). Walking on potential energy surfaces. *The Journal of Physical Chemistry*, 87(15):2745–2753.

[219] Singhal, N., Snow, C. D., and Pande, V. S. (2004). Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *The Journal of chemical physics*, 121(1):415–425.

[220] Song, G. and Amato, N. M. (2004). A motion-planning approach to folding: From paper craft to protein folding. *IEEE Transactions on Robotics and Automation*, 20(1):60–71.

[221] So/rensen, M. R. and Voter, A. F. (2000). Temperature-accelerated dynamics for simulation of infrequent events. *The Journal of Chemical Physics*, 112(21):9599–9606.

[222] Sorkine, O. and Alexa, M. (2007). As-rigid-as-possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 109–116, Aire-la-Ville, Switzerland. Eurographics Association.

[223] Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., and Seidel, H.-P. (2004). Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM.

[224] Stillinger, F. H. (1999). Exponential multiplicity of inherent structures. *Phys. Rev. E*, 59:48–51.

[225] Sugita, Y. and Okamoto, Y. (1999). Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, 314(1):141–151.

[226] Sun, H., Tian, S., Zhou, S., Li, Y., Li, D., Xu, L., Shen, M., Pan, P., and Hou, T. (2015). Revealing the favorable dissociation pathway of type II kinase inhibitors via enhanced sampling simulations and two-end-state calculations. *Scientific reports*, 5.

[227] Swendsen, R. H. and Wang, J.-S. (1986). Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607.

[228] Tang, K., Zhang, J., and Liang, J. (2014). Fast protein loop sampling and structure prediction using distance-guided sequential chain-growth Monte Carlo method. *PLOS Computational Biology*, 10(4):1–16.

[229] Thomas, S., Tang, X., Tapia, L., and Amato, N. M. (2007). Simulating protein motions with rigidity analysis. *Journal of Computational Biology*, 14(6):839–855.

[230] Thorpe, M. (2007). Comment on elastic network models and proteins. *Physical biology*, 4(1):60.

[231] Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199.

[232] Tozzini, V. (2005). Coarse-grained models for proteins. *Current opinion in structural biology*, 15(2):144–150.

[233] Van Der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. (2005). GROMACS: fast, flexible, and free. *Journal of computational chemistry*, 26(16):1701–1718.

[234] van Gunsteren, W. (1996). *Biomolecular Simulation: The GROMOS96 Manual and User Guide*. Biomos ; Zürich.

[235] Voter, A. F. (1997). Hyperdynamics: Accelerated molecular dynamics of infrequent events. *Physical Review Letters*, 78(20):3908.

[236] Wagner, I. and Musso, H. (1983). New naturally occurring amino acids. *Angewandte Chemie International Edition*, 22(11):816–828.

[237] Wales, D. (2003). *Energy landscapes: Applications to clusters, biomolecules and glasses*. Cambridge University Press.

[238] Wales, D. J. and Bogdan, T. V. (2006). Potential Energy and Free Energy Landscapes. *The Journal of Physical Chemistry B*, 110(42):20765–20776. PMID: 17048885.

[239] Wales, D. J. and Scheraga, H. A. (1999). Global optimization of clusters, crystals, and biomolecules. *Science*, 285(5432):1368–1372.

[240] Weinan, E., Ren, W., and Vanden-Eijnden, E. (2002). String method for the study of rare events. *Physical Review B*, 66(5):052301.

[241] Weinan, E., Ren, W., and Vanden-Eijnden, E. (2005). Finite Temperature String Method for the Study of Rare Events. *The Journal of Physical Chemistry B*, 109(14):6688–6693. PMID: 16851751.

[242] Weiss, D. R. and Levitt, M. (2009). Can morphing methods predict intermediate structures? *Journal of molecular biology*, 385(2):665–674.

[243] Wells, S., Menor, S., Hespenheide, B., and Thorpe, M. (2005). Constrained geometric simulation of diffusive motion in proteins. *Physical Biology*, 2(4):S127.

[244] Wright, S. J. and Nocedal, J. (1999). Numerical optimization. *Springer Science*, 35(67-68):7.

[245] Xu, D., Zhang, H., Wang, Q., and Bao, H. (2006). Poisson shape interpolation. *Graphical Models*, 68(3):268–281.

[246] Yan, H.-B., Hu, S., Martin, R. R., and Yang, Y.-L. (2008). Shape deformation using a skeleton to drive simplex transformations. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):693–706.

[247] Yan, H.-B., Hu, S.-M., and Martin, R. R. (2007). 3D morphing using strain field interpolation. *Journal of Computer Science and Technology*, 22(1):147–155.

[248] Yang, L.-J., Zou, J., Xie, H.-Z., Li, L.-L., Wei, Y.-Q., and Yang, S.-Y. (2009). Steered molecular dynamics simulations reveal the likelier dissociation pathway of imatinib from its targeting kinases c-Kit and Abl. *PLOS ONE*, 4(12):1–8.

[249] Zarkevich, N. A. and Johnson, D. D. (2015). Nudged-elastic band method with two climbing images: Finding transition states in complex energy landscapes. *The Journal of Chemical Physics*, 142(2):024106.

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

[251] Zhang, Z., Li, G., Lu, H., Ouyang, Y., Yin, M., and Xian, C. (2015). Fast As-isometric-as-possible Shape Interpolation. *Computers & Graphics*, 46(C):244–256.

[252] Zhou, Y., Guan, L., Freites, J. A., and Kaback, H. R. (2008). Opening and closing of the periplasmic gate in lactose permease. *Proceedings of the National Academy of Sciences*, 105(10):3774–3778.

[253] Zollhöfer, M., Sert, E., Greiner, G., and Süßmuth, J. (2012). GPU based ARAP Deformation using Volumetric Lattices. In *Eurographics 2012, Cagliari, Sardinia, Italy*, pages 85–88. The Eurographics Association.

[254] Zou, J., Wang, Y.-D., Ma, F.-X., Xiang, M.-L., Shi, B., Wei, Y.-Q., and Yang, S.-Y. (2008). Detailed conformational dynamics of juxtamembrane region and activation loop in c-Kit kinase activation process. *Proteins: Structure, Function, and Bioinformatics*, 72(1):323–332.

Appendix A

ARAP energy minimization

The following mathematical development assumes that the one-ring neighbor topology is used for constructing ARAP sets.

A.1 Minimizing ARAP energy

The problem of minimizing Equation 3.3 can be reduced to solving the linear system $\mathbf{L}\widehat{\mathbf{p}} = \mathbf{b}$. If the set V_c contains n_c indices of the constrained vertices, $\mathbf{L}$ is a $(n-n_c) \times (n-n_c)$ matrix. The $(n-n_c) \times 3$ matrix $\widehat{p}$ is a concatenation of unknown vertex positions and b is a matrix of the same size. The derivation of these matrices can be found in [222]. However, it is also briefly presented here.

First, let us set to zero the derivative of Equation 3.3 with respect to an unknown vertex positions $\widehat{\mathbf{p}}_\mathbf{k}$, i.e. $(k \notin V_c)$:

$$\frac{\partial E}{\partial \widehat{\mathbf{p}}_\mathbf{k}} = \frac{\partial}{\partial \widehat{\mathbf{p}}_\mathbf{k}} \Big(\omega_k \sum_{j \in \mathcal{N}_k} \omega_{kj} ||\widehat{\mathbf{p}}_\mathbf{k} - \widehat{\mathbf{p}}_\mathbf{j} - \mathbf{R}_\mathbf{k}(\mathbf{p}_\mathbf{k} - \mathbf{p}_\mathbf{j})||^2 \quad + \sum_{j \in \mathcal{N}_k} \omega_j \omega_{jk} ||\widehat{\mathbf{p}}_\mathbf{j} - \widehat{\mathbf{p}}_\mathbf{k} - \mathbf{R}_\mathbf{j}(\mathbf{p}_\mathbf{j} - \mathbf{p}_\mathbf{k})||^2 \Big) = 0$$

Hence, we have:

$$2\omega_k \sum_{j \in \mathcal{N}_k} \omega_{kj} \left(\widehat{\mathbf{p}}_\mathbf{k} - \widehat{\mathbf{p}}_\mathbf{j} - \mathbf{R}_\mathbf{k}(\mathbf{p}_\mathbf{k} - \mathbf{p}_\mathbf{j}) \right) - 2 \sum_{j \in \mathcal{N}_k} \omega_j \omega_{jk} \left(\widehat{\mathbf{p}}_\mathbf{j} - \widehat{\mathbf{p}}_\mathbf{k} - \mathbf{R}_\mathbf{j}(\mathbf{p}_\mathbf{j} - \mathbf{p}_\mathbf{k}) \right) = 0$$

And:

$$\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}-\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}=\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}\mathbf{R}_\mathbf{k}+\omega_j\omega_{jk}\mathbf{R}_\mathbf{j}\right)(\mathbf{p}_\mathbf{k}-\mathbf{p}_\mathbf{j}) \tag{A.1}$$

This last equation is used to construct matrices $\mathbf{L}$ and $\mathbf{b}$ when the set $\mathcal{N}_k$ does not contain any constrained vertices ($\mathcal{N}_k \cap V_c = \emptyset$). Hence, the diagonal and off-diagonal coefficients of $\mathbf{L}$ are those in front of $\widehat{\mathbf{p}}_\mathbf{k}$ and $\widehat{\mathbf{p}}_\mathbf{j}$ in the equation, respectively. The transpose of the right hand side constitutes a row in $\mathbf{b}$. When $\mathcal{N}_k$ contains constrained vertices ($\mathcal{N}_k \cap V_c \neq \emptyset$), the following equation is used instead:

$$\begin{aligned}&\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}-\sum_{j\in\mathcal{N}_k\setminus V_c}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}\\ =&\sum_{c\in\mathcal{N}_k\cap V_c}(\omega_k\omega_{kc}+\omega_c\omega_{ck})\overline{\mathbf{p}}_\mathbf{c}+\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}\mathbf{R}_\mathbf{k}+\omega_j\omega_{jk}\mathbf{R}_\mathbf{j}\right)(\mathbf{p}_\mathbf{k}-\mathbf{p}_\mathbf{j})\end{aligned} \tag{A.2}$$

It can be shown that in case of one-ring neighbor topology, $\mathbf{L}$ is a sparse, symmetric and positive definite matrix, for which we can thus compute a Cholesky decomposition. Because its coefficients are only independent on the weights and the connectivity of the vertices, this decomposition can be performed only once.

It should be noted that $\mathbf{L}\widehat{\mathbf{p}} = \mathbf{b}$ has a unique solution for $\widehat{\mathbf{p}}$ only when $\mathbf{L}$ has full rank. This requires that the structure be fully connected[1]. and at least one vertex is constrained.

A.2 Minimizing ARAP energy for interpolation

The derivation of the linear system $\mathbf{L}\widehat{\mathbf{p}}(t) = \mathbf{b}(t)$ for the minimization of Equation 3.4 can be proceeded similarly. In fact, one just has to add the interpolation instance (t) to the appropriate terms of Equations A.1 and A.2 to arrive at Equations A.3 and A.4:

$$\begin{aligned}&\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}(t)-\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}(t)\\ =&\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}\mathbf{R}_\mathbf{k}(t)+\omega_j\omega_{jk}\mathbf{R}_\mathbf{j}(t)\right)(\mathbf{p}_\mathbf{k}-\mathbf{p}_\mathbf{j})\end{aligned} \tag{A.3}$$

[1] A structure is fully connected when any vertex can be reached from another vertex in the structure by traversing through a set of vertices and edges which are also in the same structure.

$$\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}(t)-\sum_{j\in\mathcal{N}_k\setminus V_c}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}(t)$$
$$=\sum_{c\in\mathcal{N}_k\cap V_c}(\omega_k\omega_{kc}+\omega_c\omega_{ck})\overline{\mathbf{p}}_\mathbf{c}+\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}\mathbf{R}_\mathbf{k}(t)+\omega_j\omega_{jk}\mathbf{R}_\mathbf{j}(t)\right)(\mathbf{p_k}-\mathbf{p_j}) \tag{A.4}$$

The formation of matrix $\mathbf{L}$ is done similarly as in Section A.1. In fact, $\mathbf{L}$ is exactly the same as that in Section A.1 because it is independent on t. Hence, one can compute a Cholesky decomposition for $\mathbf{L}$ only once and use it to solve for all the intermediate shapes of the interpolation path.

A.3 Minimizing ARAP energy for interpolation with reaching goal condition

Similar to Section A.2, the derivation of the linear system $\mathbf{L}\widehat{\mathbf{p}}(t)=\mathbf{b}(t)$ for the minimization of Equation 4.2 will lead to Equations A.5 and A.6:

$$\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}(t)-\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}(t)$$
$$=\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}s_{kj}(t)\mathbf{R}_\mathbf{kj}(t)\mathbf{R}_\mathbf{k}(t)+\omega_j\omega_{jk}s_{jk}(t)\mathbf{R}_\mathbf{jk}(t)\mathbf{R}_\mathbf{j}(t)\right)(\mathbf{p_k}-\mathbf{p_j}) \tag{A.5}$$

$$\left(\sum_{j\in\mathcal{N}_k}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\right)\widehat{\mathbf{p}}_\mathbf{k}(t)-\sum_{j\in\mathcal{N}_k\setminus V_c}(\omega_k\omega_{kj}+\omega_j\omega_{jk})\widehat{\mathbf{p}}_\mathbf{j}(t)$$
$$=\sum_{c\in\mathcal{N}_k\cap V_c}(\omega_k\omega_{kc}+\omega_c\omega_{ck})\overline{\mathbf{p}}_\mathbf{c}+\sum_{j\in\mathcal{N}_k}\left(\omega_k\omega_{kj}s_{kj}(t)\mathbf{R}_\mathbf{kj}(t)\mathbf{R}_\mathbf{k}(t)+\omega_j\omega_{jk}s_{jk}(t)\mathbf{R}_\mathbf{jk}(t)\mathbf{R}_\mathbf{j}(t)\right)(\mathbf{p_k}-\mathbf{p_j}) \tag{A.6}$$

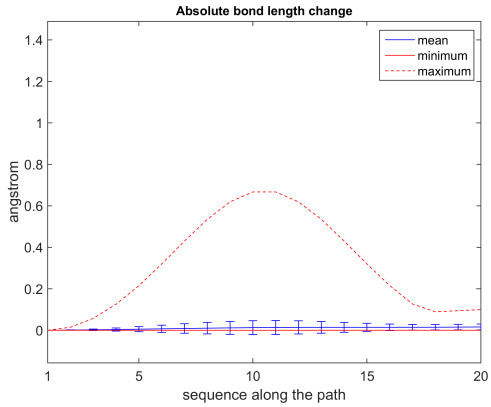
Appendix B

Supplementary material of the ARAP interpolation method for molecular systems

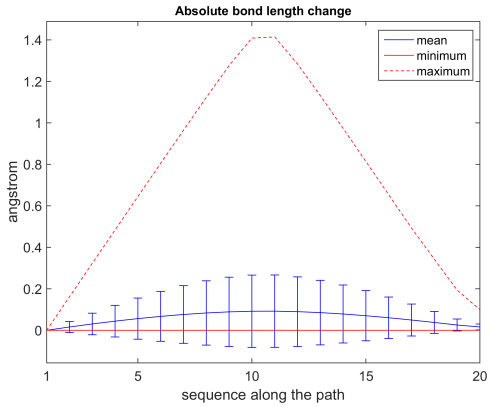
Section 4.2 showed the comparison between the ARAP interpolation method and the linear interpolation method regarding the preservation of bond lengths, bond angles, dihedral angles, and consecutive-C_α distances only for 5'-Nucleotidase. This appendix shows the results of the same comparison for the rest of the benchmarks mentioned in the Section 4.2.

B.1 Bond lengths, bond angles and dihedral angles

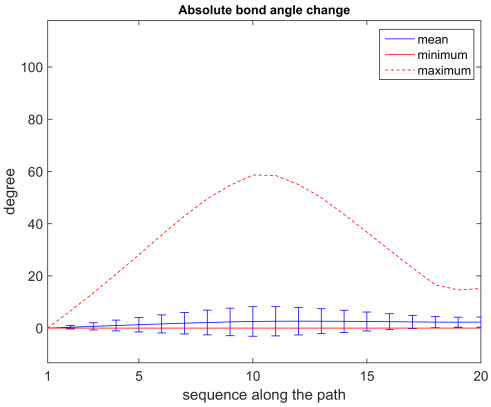
Section 4.2.1 showed how the ARAP interpolation method preserves the bond lengths, bond angles, and dihedral angles compared with the linear interpolation method for the case of 5'-Nucleotidase. This section shows the same comparison between the ARAP interpolation method and linear interpolation method for the rest of the benchmarks mentioned in Section 4.2.



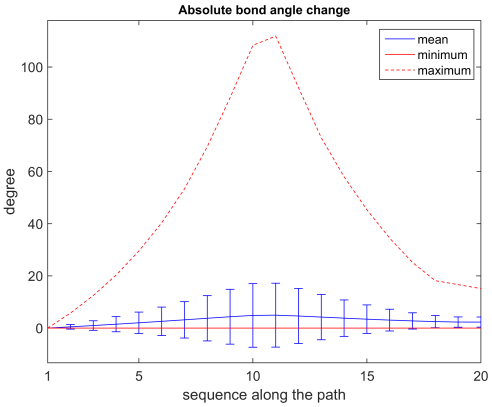
(a) ARAP interpolation



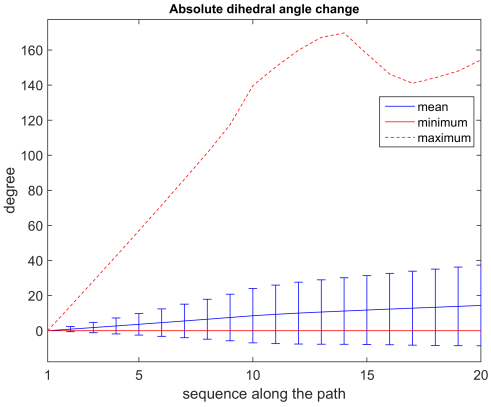
(b) Linear interpolation



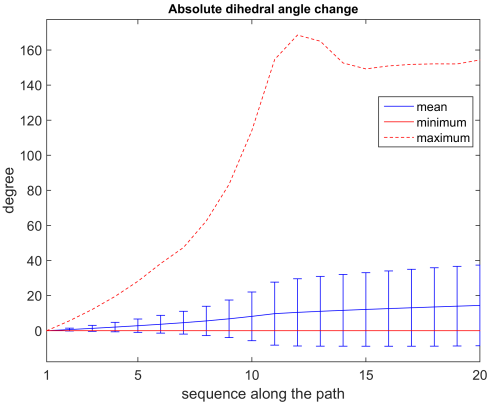
(c) ARAP interpolation



(d) Linear interpolation

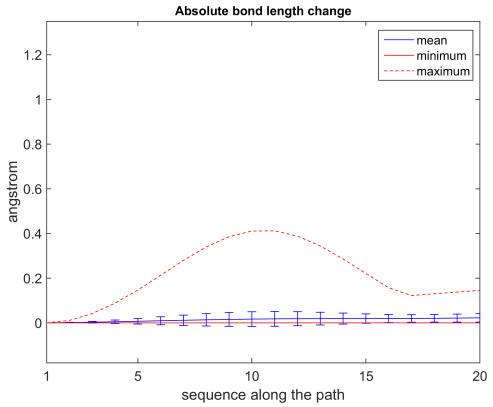


(e) ARAP interpolation

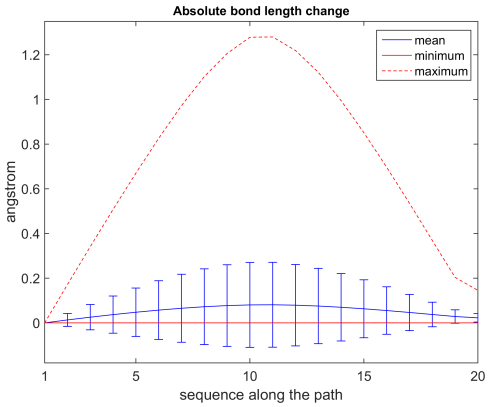


(f) Linear interpolation

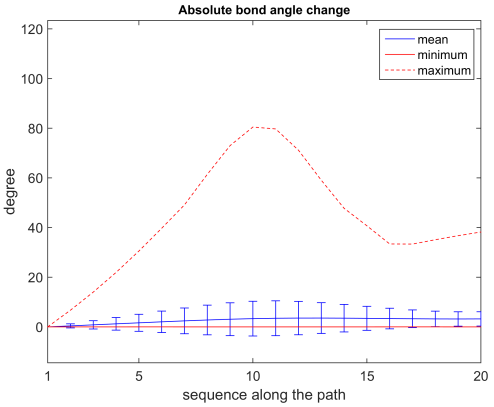
Fig. B.1 Statistics of absolute changes in bond length, bond angle and dihedral angle for Adenylate Kinase.



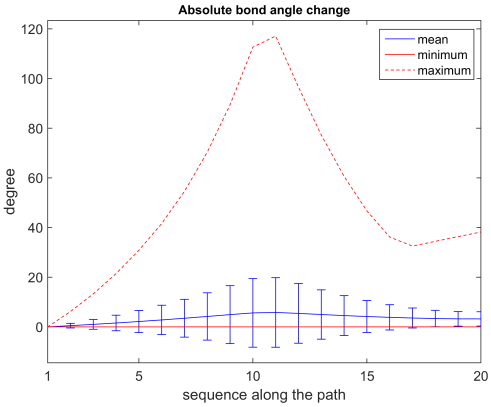
(a) ARAP interpolation



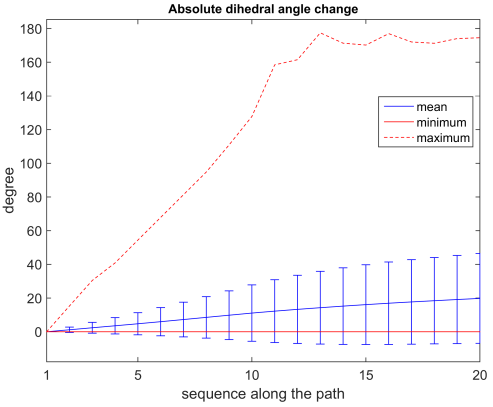
(b) Linear interpolation



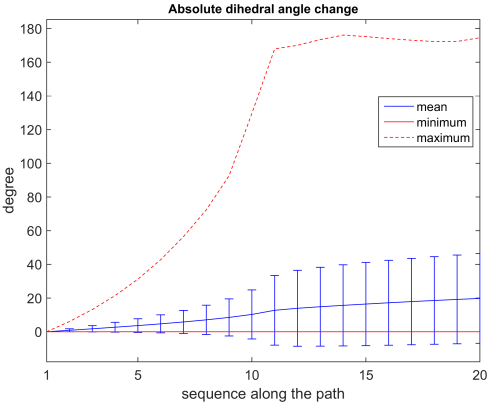
(c) ARAP interpolation



(d) Linear interpolation

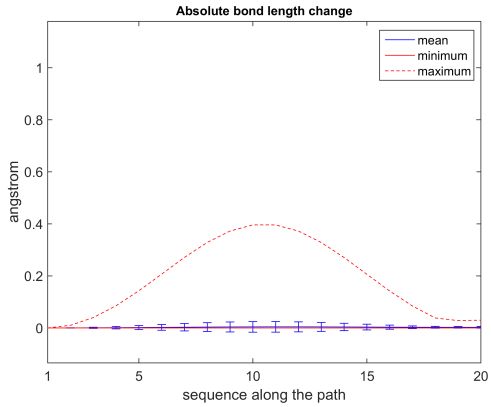


(e) ARAP interpolation

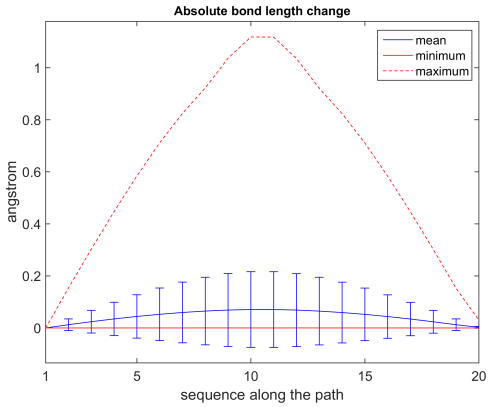


(f) Linear interpolation

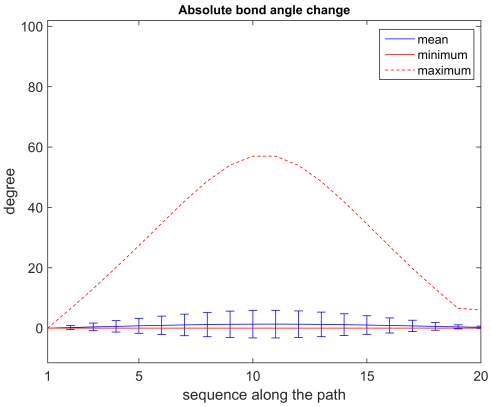
Fig. B.2 Statistics of absolute changes in bond length, bond angle and dihedral angle for Alcohol Dehydrogenase.



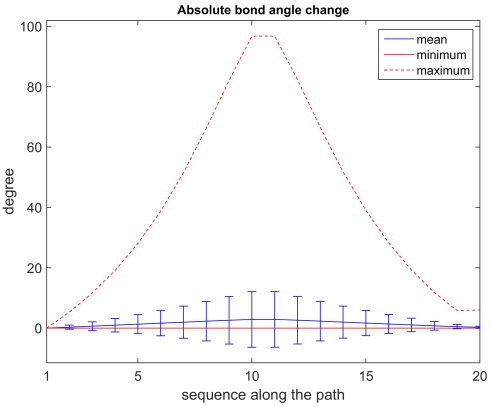
(a) ARAP interpolation



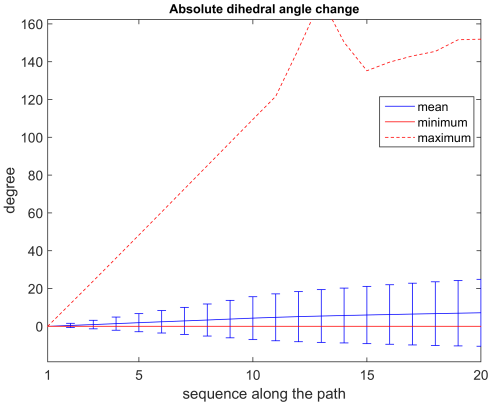
(b) Linear interpolation



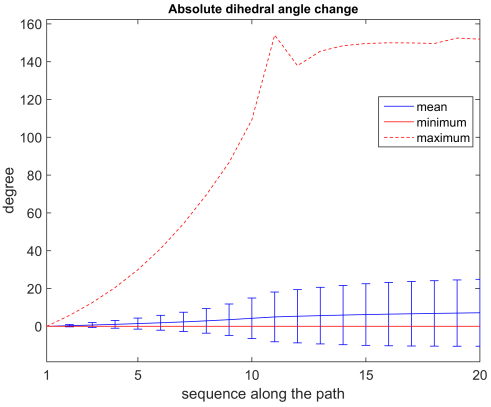
(c) ARAP interpolation



(d) Linear interpolation

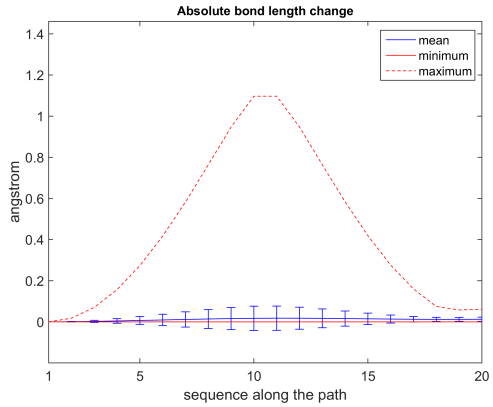


(e) ARAP interpolation

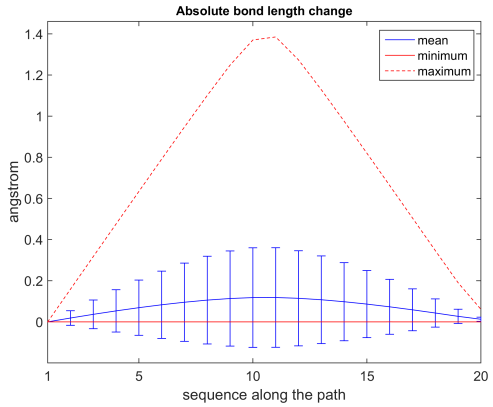


(f) Linear interpolation

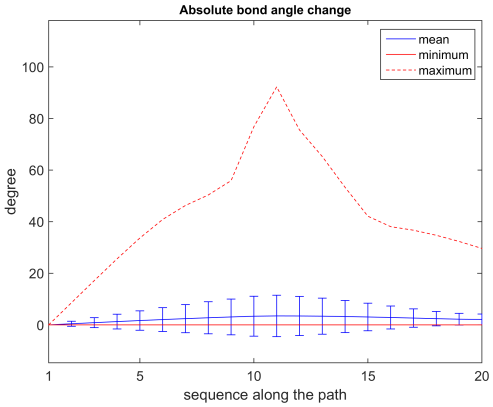
Fig. B.3 Statistics of absolute changes in bond length, bond angle and dihedral angle for Calmodulin.



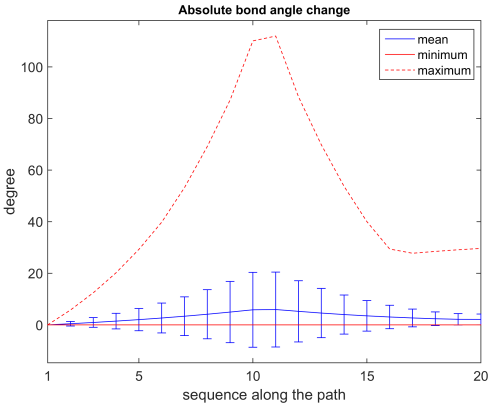
(a) ARAP interpolation



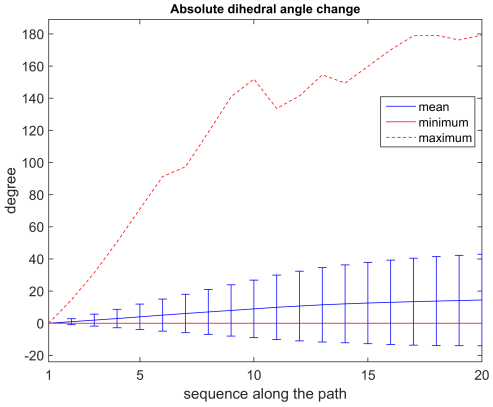
(b) Linear interpolation



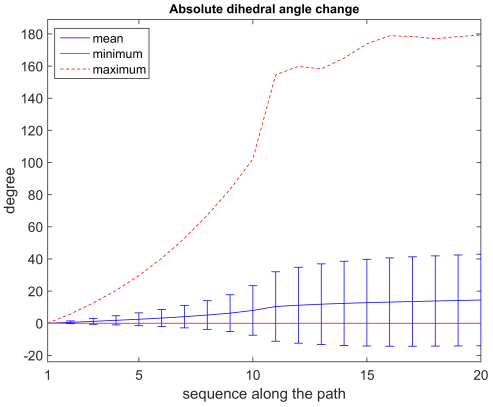
(c) ARAP interpolation



(d) Linear interpolation

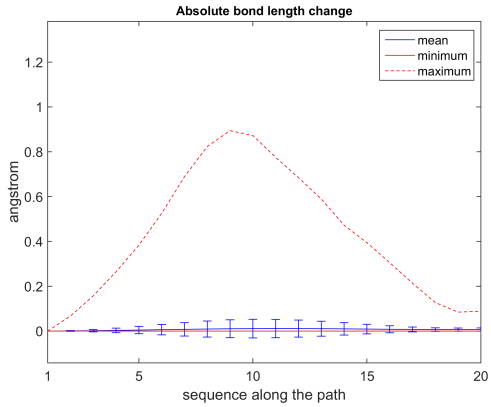


(e) ARAP interpolation

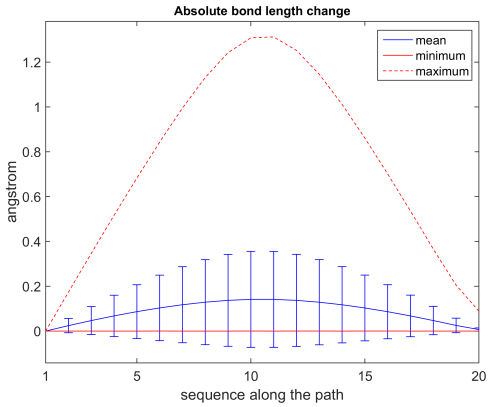


(f) Linear interpolation

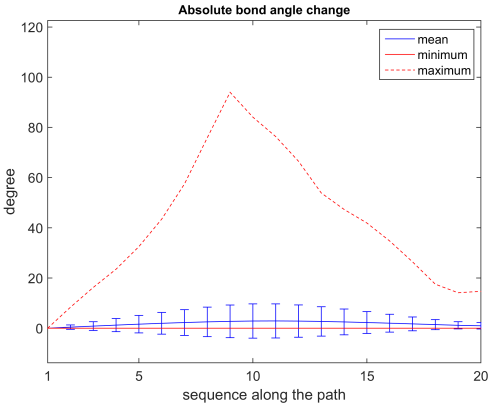
Fig. B.4 Statistics of absolute changes in bond length, bond angle and dihedral angle for Collagenase.



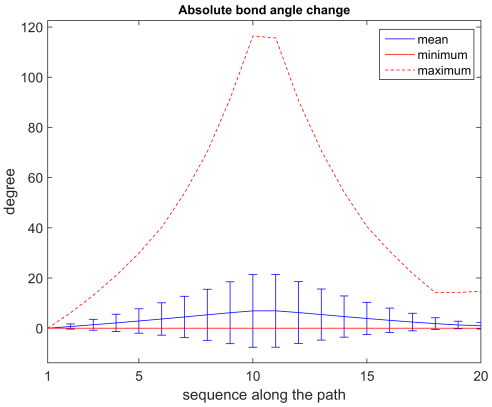
(a) ARAP interpolation



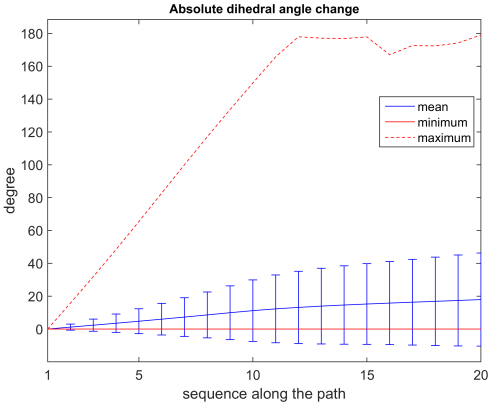
(b) Linear interpolation



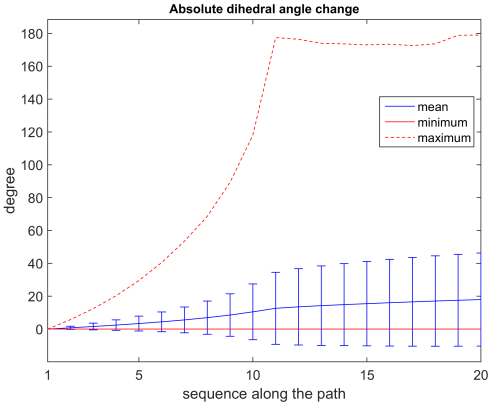
(c) ARAP interpolation



(d) Linear interpolation

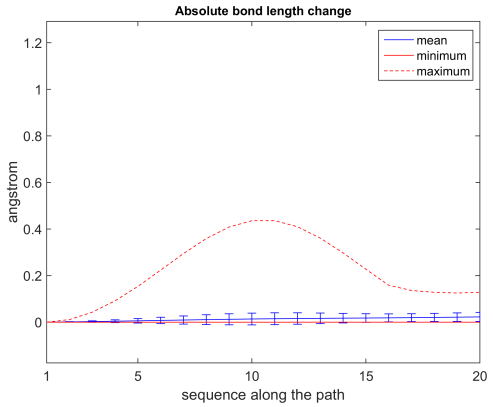


(e) ARAP interpolation

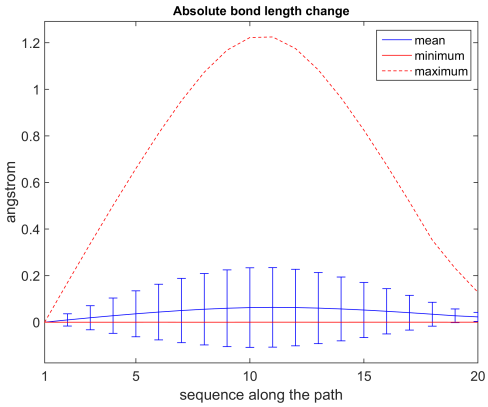


(f) Linear interpolation

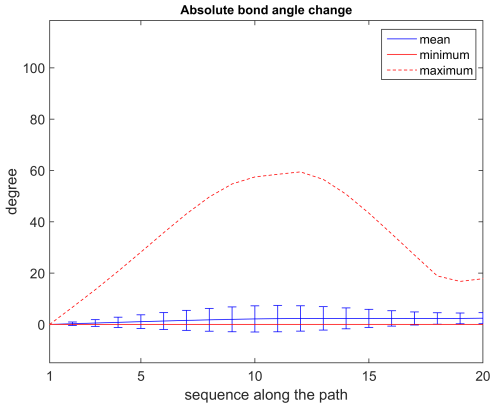
Fig. B.5 Statistics of absolute changes in bond length, bond angle and dihedral angle for Dengue 2 Virus Envelope Glycoprotein.



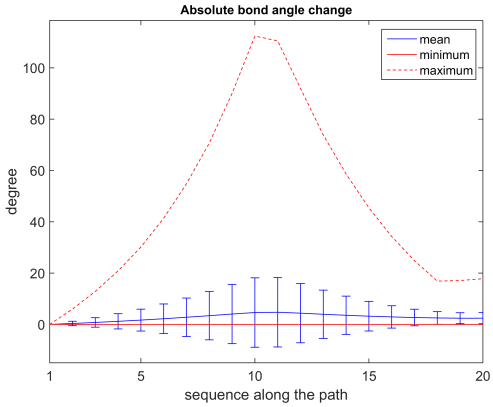
(a) ARAP interpolation



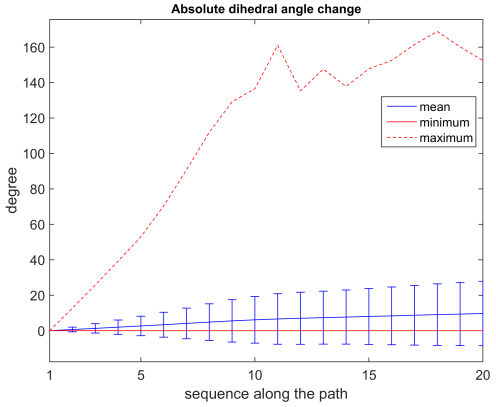
(b) Linear interpolation



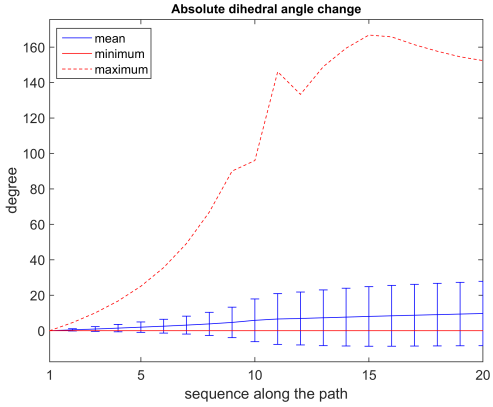
(c) ARAP interpolation



(d) Linear interpolation

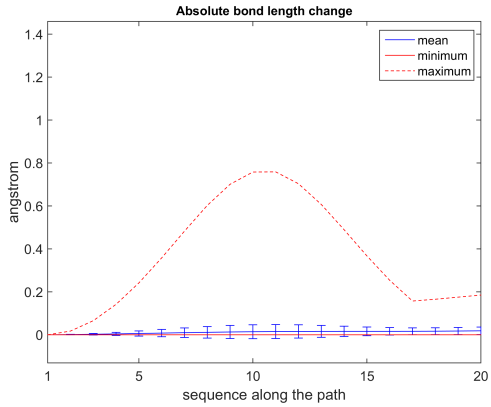


(e) ARAP interpolation

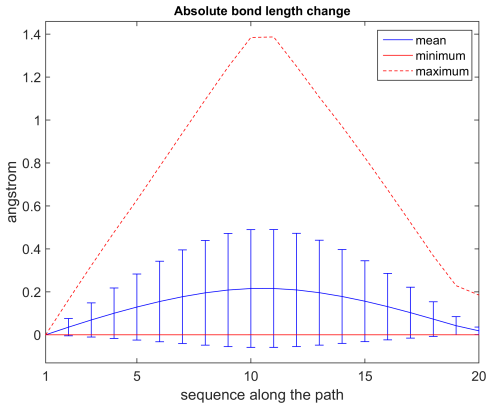


(f) Linear interpolation

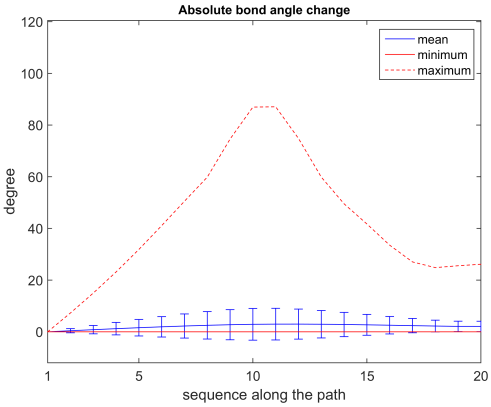
Fig. B.6 Statistics of absolute changes in bond length, bond angle and dihedral angle for Dihydrofolate Reductase.



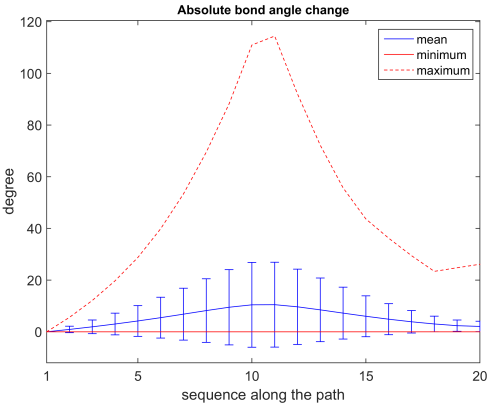
(a) ARAP interpolation



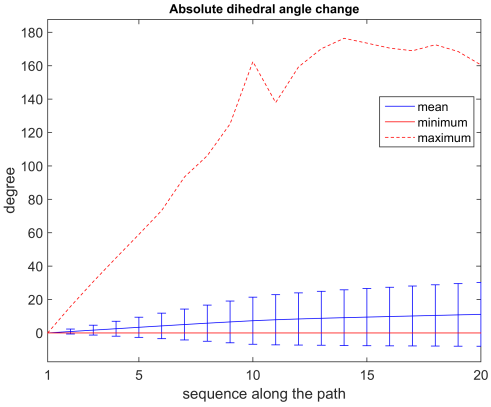
(b) Linear interpolation



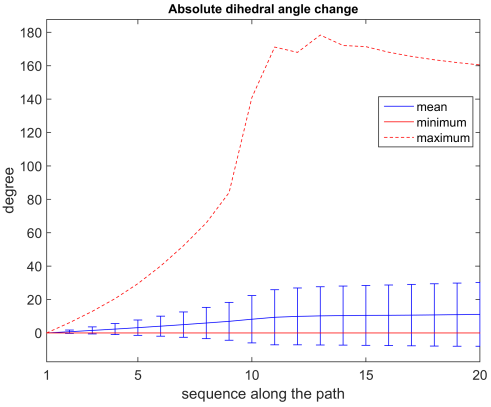
(c) ARAP interpolation



(d) Linear interpolation

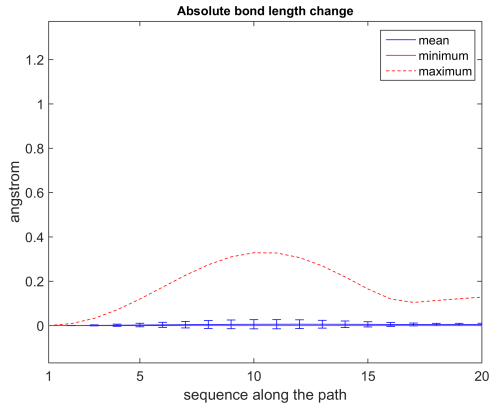


(e) ARAP interpolation

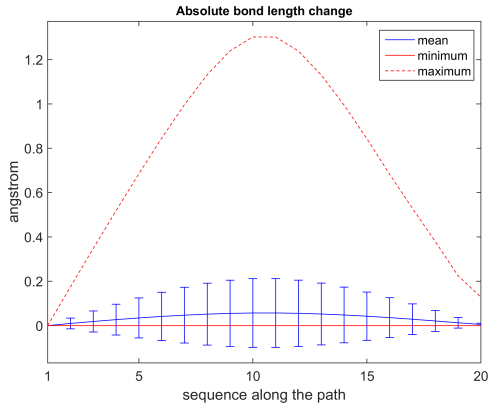


(f) Linear interpolation

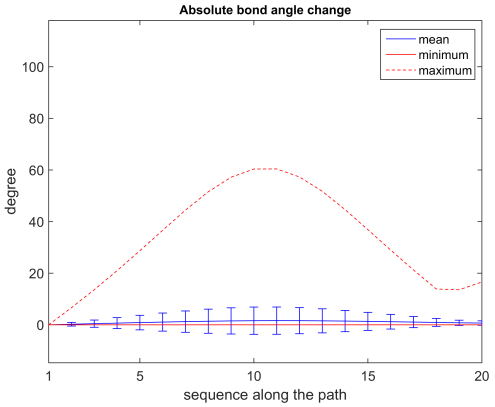
Fig. B.7 Statistics of absolute changes in bond length, bond angle and dihedral angle for Diphtheria Toxin.



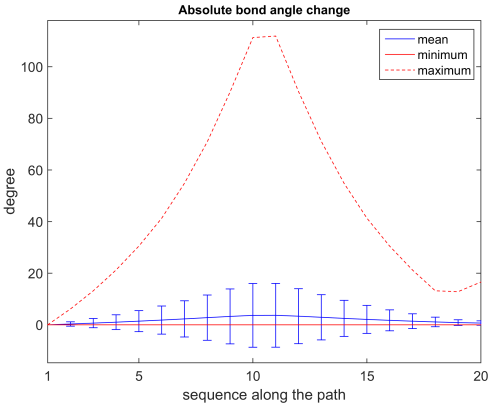
(a) ARAP interpolation



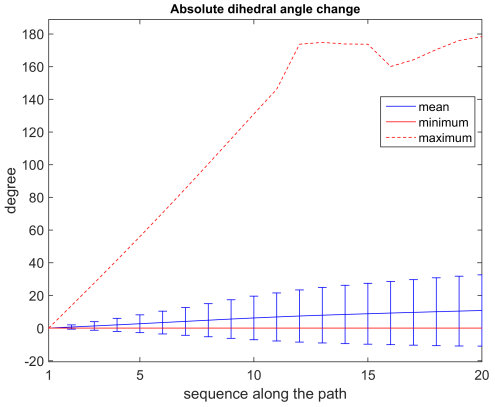
(b) Linear interpolation



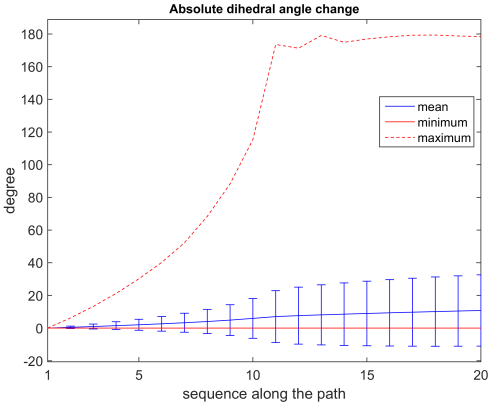
(c) ARAP interpolation



(d) Linear interpolation

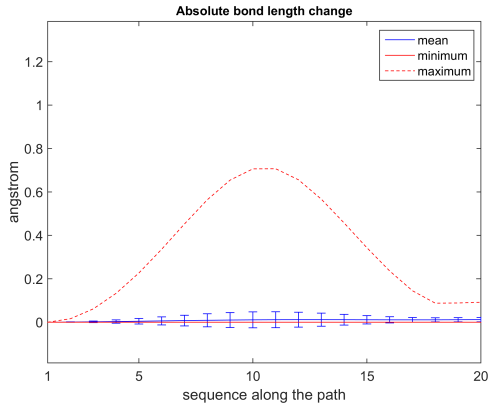


(e) ARAP interpolation

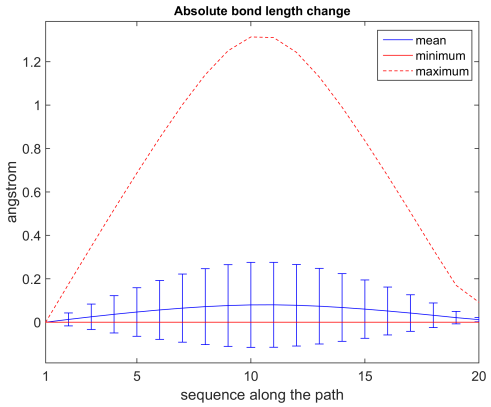


(f) Linear interpolation

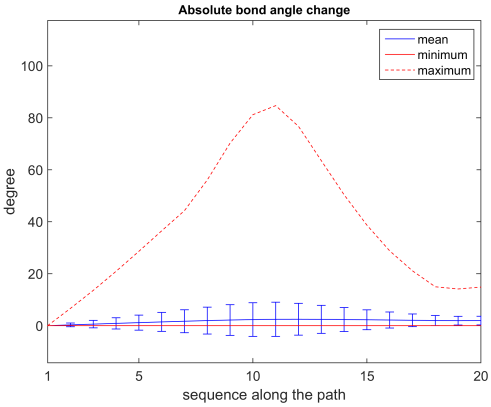
Fig. B.8 Statistics of absolute changes in bond length, bond angle and dihedral angle for DNA Polymerase.



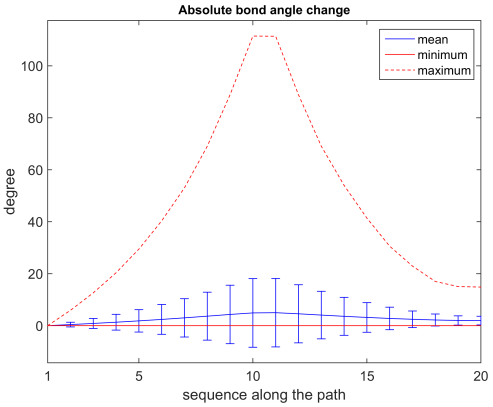
(a) ARAP interpolation



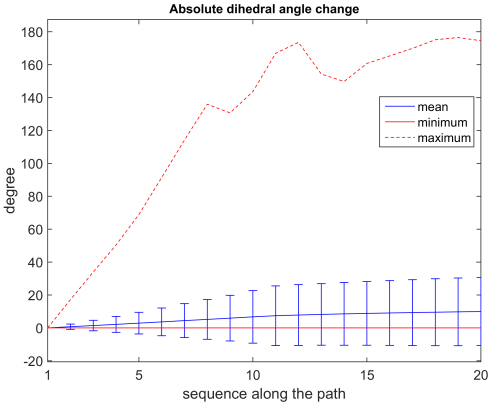
(b) Linear interpolation



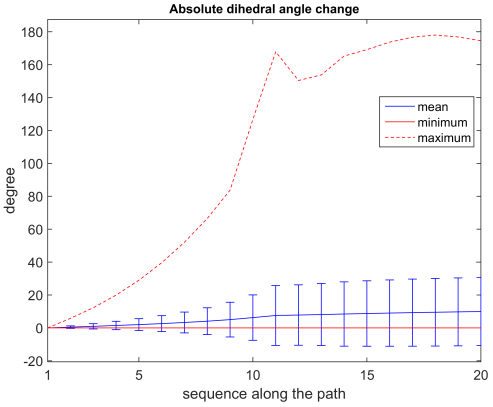
(c) ARAP interpolation



(d) Linear interpolation

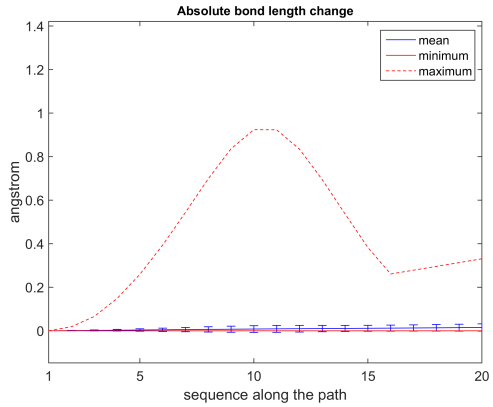


(e) ARAP interpolation

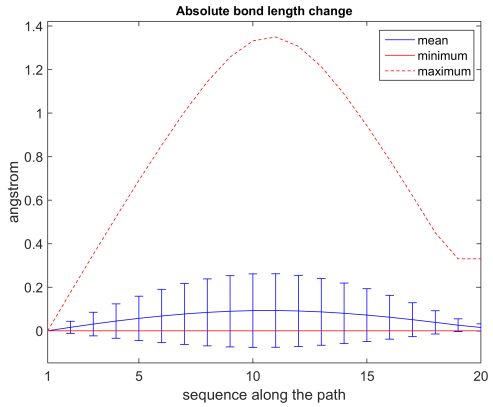


(f) Linear interpolation

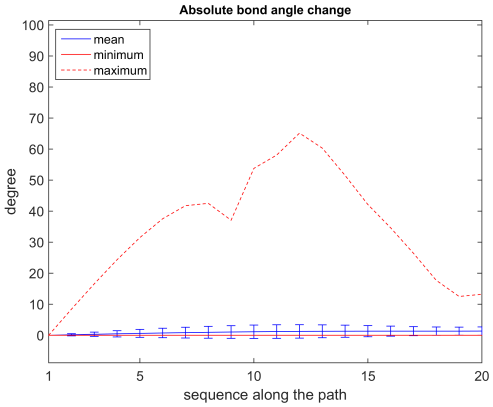
Fig. B.9 Statistics of absolute changes in bond length, bond angle and dihedral angle for Pyrophosphokinase.



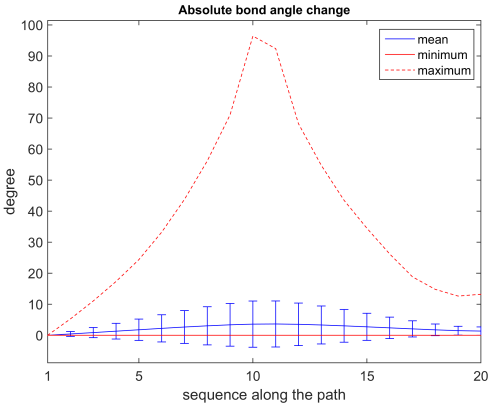
(a) ARAP interpolation



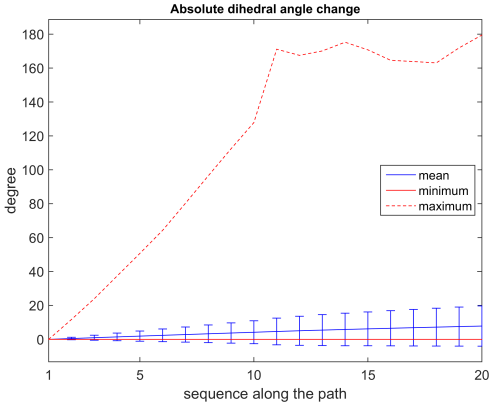
(b) Linear interpolation



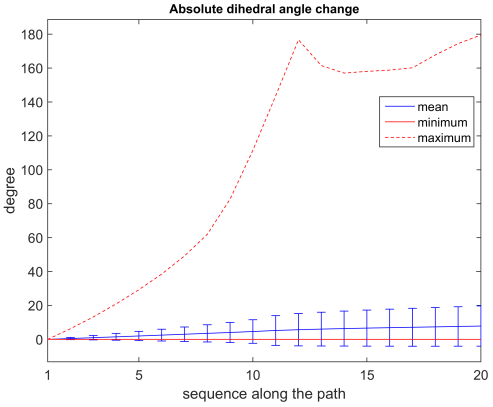
(c) ARAP interpolation



(d) Linear interpolation

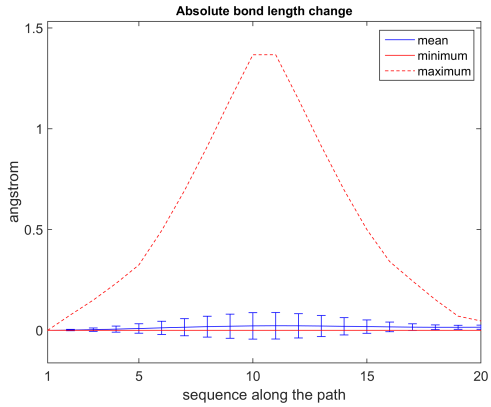


(e) ARAP interpolation

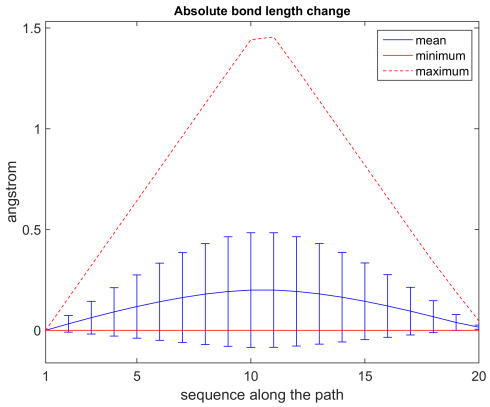


(f) Linear interpolation

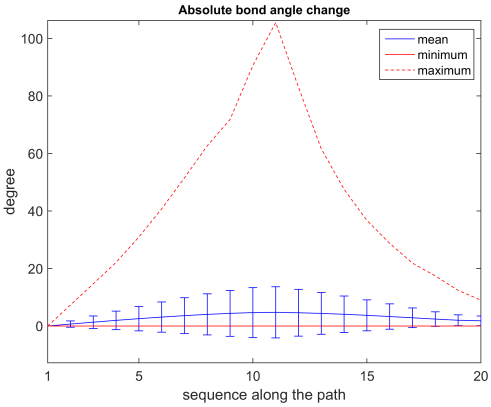
Fig. B.10 Statistics of absolute changes in bond length, bond angle and dihedral angle for Pyruvate Phosphate Dikinase.



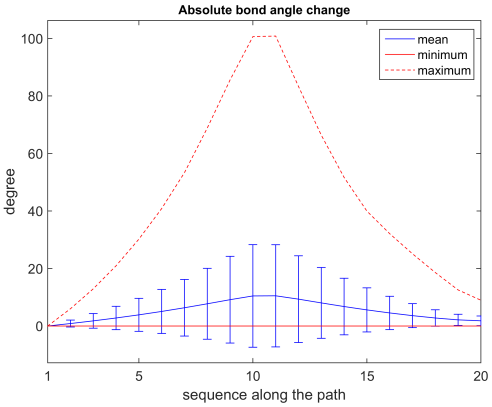
(a) ARAP interpolation



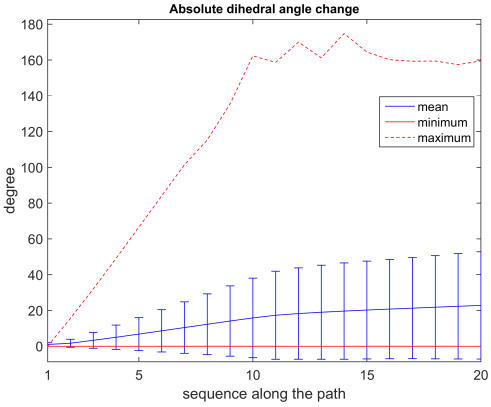
(b) Linear interpolation



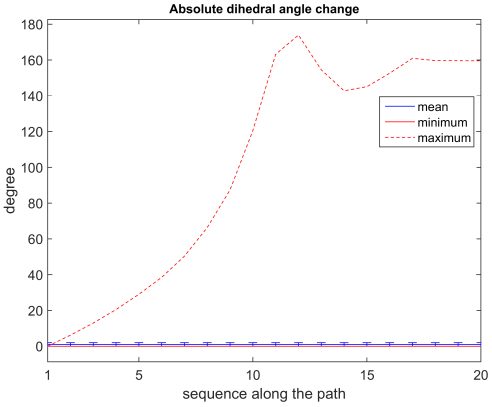
(c) ARAP interpolation



(d) Linear interpolation



(e) ARAP interpolation

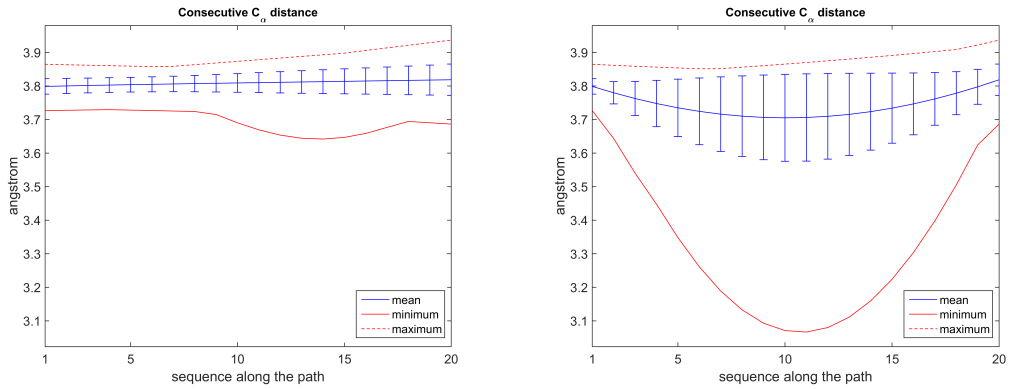


(f) Linear interpolation

Fig. B.11 Statistics of absolute changes in bond length, bond angle and dihedral angle for Spindle Assembly Checkpoint Protein.

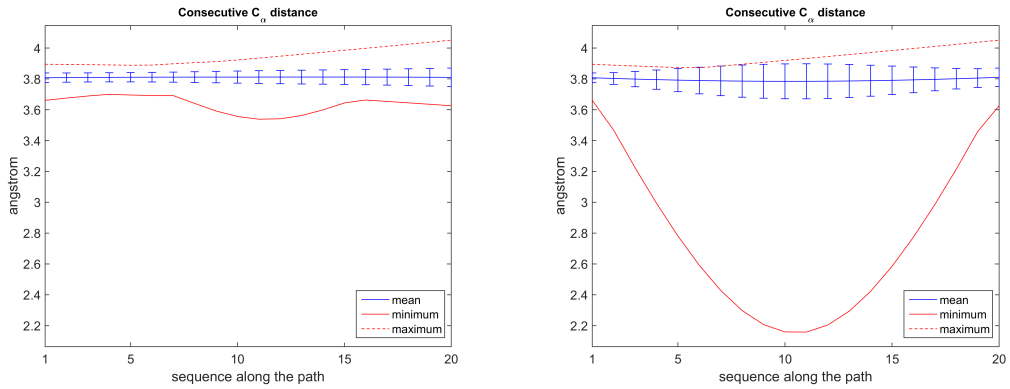
B.2 C_α distance

Section 4.2.2 showed how the ARAP interpolation method preserves consecutive-C_α distances compared with the linear interpolation method for the case of 5'-Nucleotidase. This section shows the same comparison between the ARAP interpolation method and linear interpolation method for the rest of the benchmarks mentioned in Section 4.2.



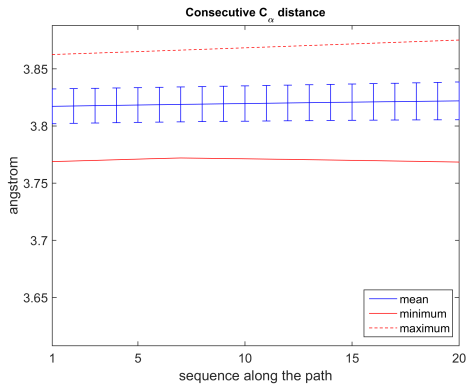
(a) ARAP interpolation (b) Linear interpolation

Fig. B.12 Statistics of consecutive-C_α distances for Adenylate Kinase.

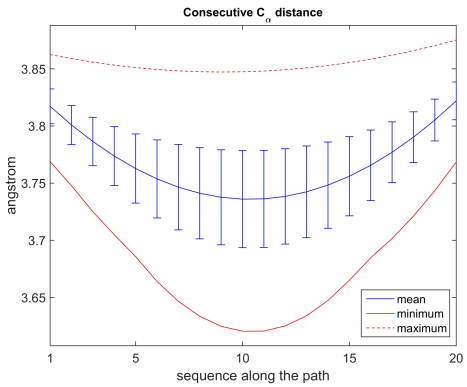


(a) ARAP interpolation (b) Linear interpolation

Fig. B.13 Statistics of consecutive-C_α distances for Alcohol Dehydrogenase.

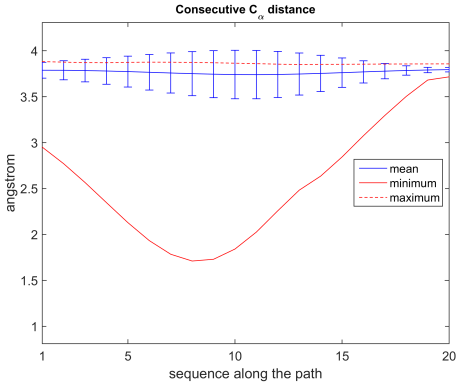


(a) ARAP interpolation

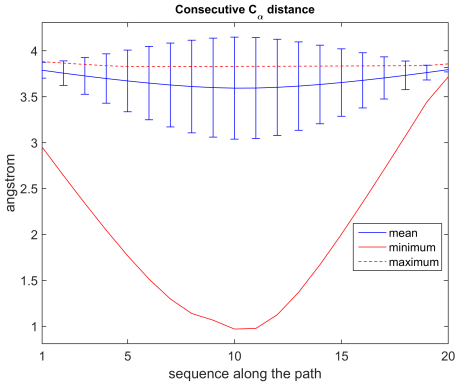


(b) Linear interpolation

Fig. B.14 Statistics of consecutive-C_α distances for Calmodulin.

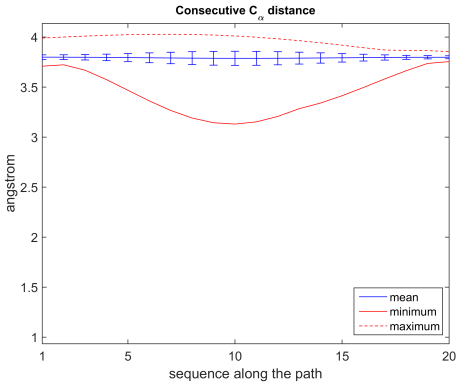


(a) ARAP interpolation

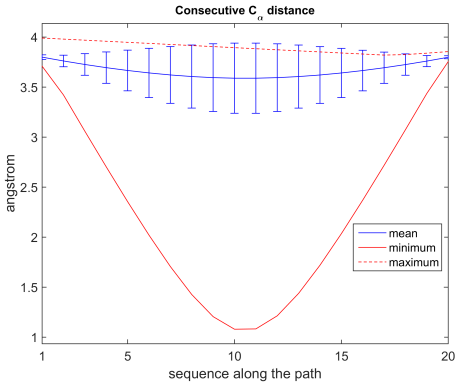


(b) Linear interpolation

Fig. B.15 Statistics of consecutive-C_α distances for Collagenase.

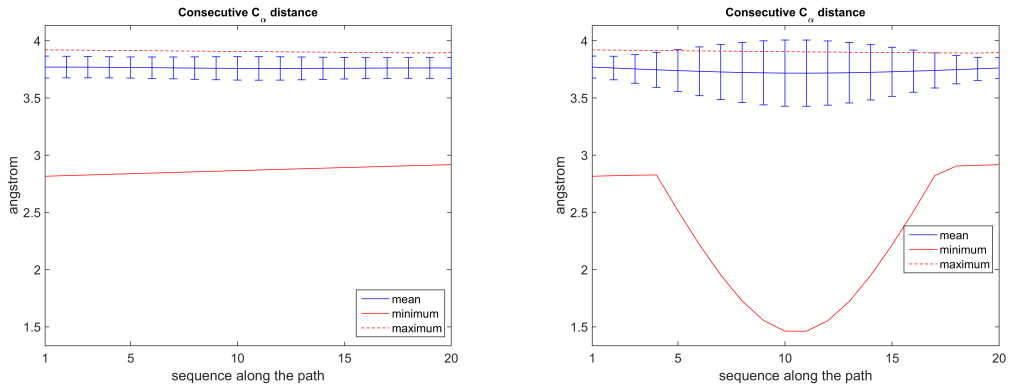


(a) ARAP interpolation



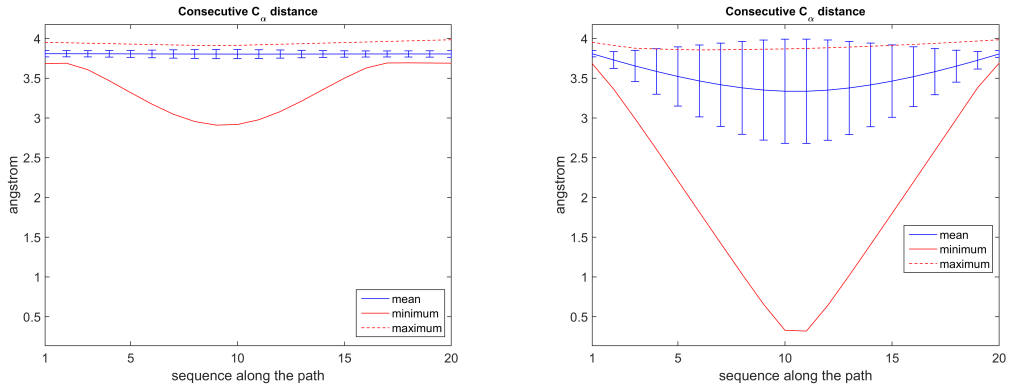
(b) Linear interpolation

Fig. B.16 Statistics of consecutive-C_α distances for Dengue 2 Virus Envelope Glycoprotein.



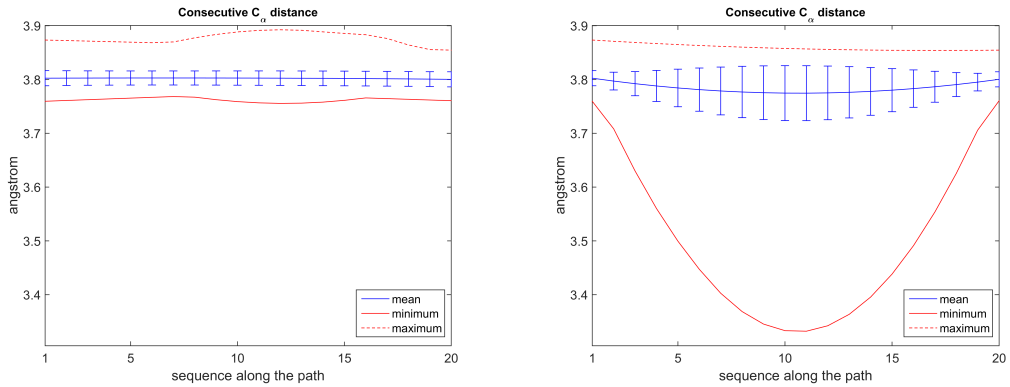
(a) ARAP interpolation (b) Linear interpolation

Fig. B.17 Statistics of consecutive-C_α distances for Dihydrofolate Reductase.



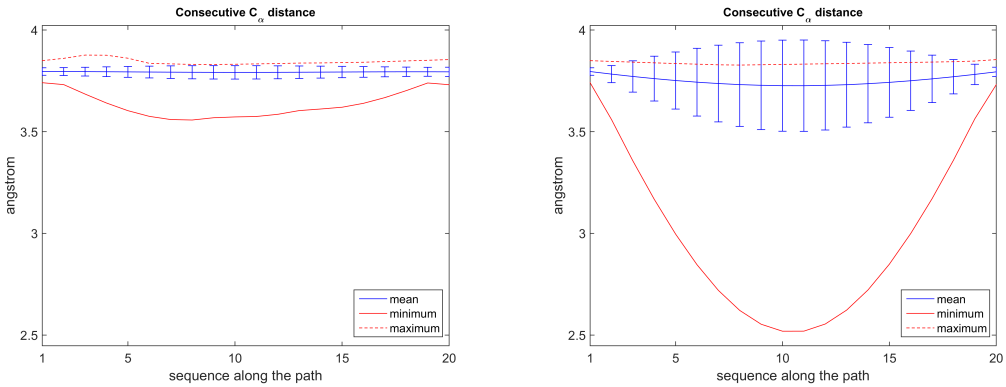
(a) ARAP interpolation (b) Linear interpolation

Fig. B.18 Statistics of consecutive-C_α distances for Diphtheria Toxin.



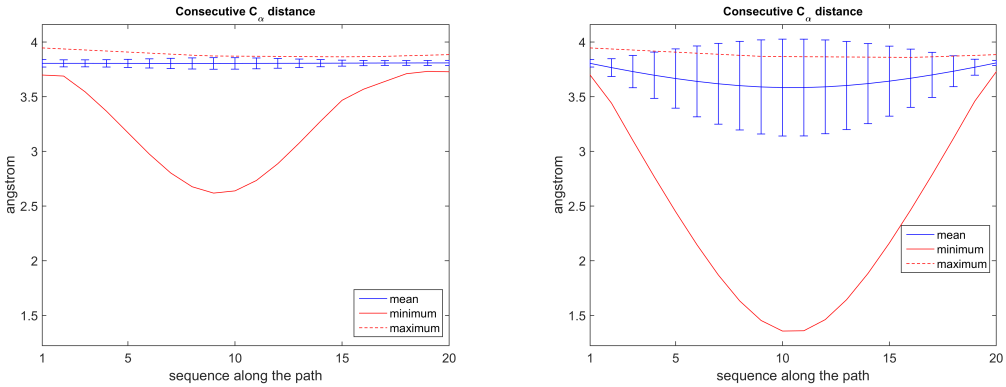
(a) ARAP interpolation (b) Linear interpolation

Fig. B.19 Statistics of consecutive-C_α distances for DNA Polymerase.



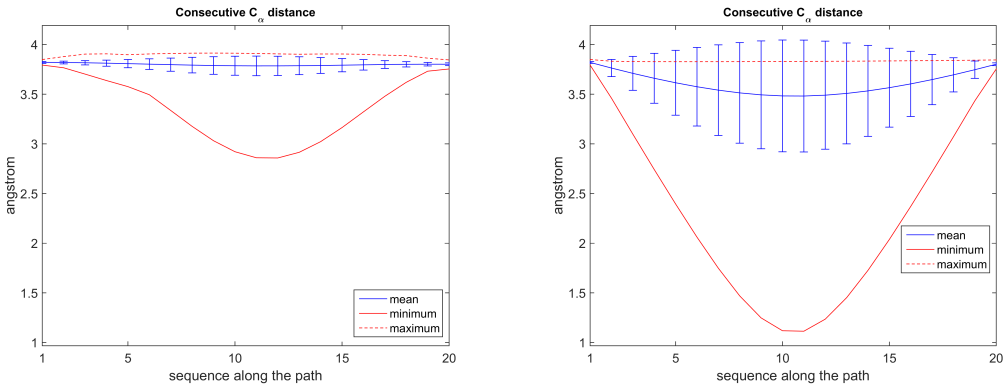
(a) ARAP interpolation (b) Linear interpolation

Fig. B.20 Statistics of consecutive-C_α distances for Pyrophosphokinase.



(a) ARAP interpolation (b) Linear interpolation

Fig. B.21 Statistics of consecutive-C_α distances for Pyruvate Phosphate Dikinase.



(a) ARAP interpolation (b) Linear interpolation

Fig. B.22 Statistics of consecutive-C_α distances for Spindle Assembly Checkpoint Protein.