ABSTRACT

Title of dissertation: PROTEIN-PROTEIN DOCKING USING LONG RANGE NUCLEAR MAGNETIC RESONANCE CONSTRAINTS

Konstantin Berlin, Doctor of Philosophy, 2010

Dissertation directed by: Dianne P. O'Leary
Department of Computer Science
David Fushman
Department of Chemistry and Biochemistry

One of the main methods for experimentally determining protein structure is nuclear magnetic resonance (NMR) spectroscopy. The advantage of using NMR compared to other methods is that the molecule may be studied in its natural state and environment. However, NMR is limited in its facility to analyze multi-domain molecules because of the scarcity of inter-atomic NMR constraints between the domains. In those cases it might be possible to dock the domains based on long range NMR constraints that are related to the molecule's overall structure.

We present two computational methods for rigid docking based on long range NMR constraints. The first docking method is based on the overall alignment tensor of the complex. The docking algorithm is based on the minimization of the difference between the predicted and experimental alignment tensor. In order to efficiently dock the complex we introduce a new, computationally efficient method called PATI for predicting the molecular alignment tensor based on the three-dimensional structure of the molecule. The increase in speed compared to the currently best-known

method (PALES) is achieved by re-expressing the problem as one of numerical integration, rather than a simple uniform sampling (as in the PALES method), and by using a convex hull rather than a detailed representation of the surface of a molecule. Using PATI, we derive a method called PATIDOCK for efficiently docking a two-domain complex based solely on the novel idea of using the difference between the experimental alignment tensor and the predicted alignment tensor computed by PATI. We show that the alignment tensor fundamentally contains enough information to accurately dock a two-domain complex, and that we can very quickly dock the two domains by pre-computing the right set of data.

A second new docking method is based on a similar concept but using the rotational diffusion tensor. We derive a minimization algorithm for this docking method by separating the problem into two simpler minimization problems and approximating our energy function by a quadratic equation.

These methods provide two new efficient procedures for protein docking computations.

PROTEIN-PROTEIN DOCKING USING LONG RANGE NUCLEAR MAGNETIC RESONANCE CONSTRAINTS

by

Konstantin Berlin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Dianne P. O'Leary, Chair/Co-Advisor
Professor David Fushman, Co-Advisor
Associate Professor Alan Sussman
Associate Professor Chau-Wen Tseng
Professor Sergei Sukharev

Dedication

To my late father Alexander, who pushed me onto my academic path, but did not live to see me graduate.

Acknowledgments

I would like to express my immense gratitude to my advisors: David Fushman, for helping develop and guide my thesis; and Dianne O'Leary, for her tireless effort of making me understand what I was doing and forcing me to make it better. Without their guidance and support this thesis would not be possible.

I would like to thank my thesis committee, Alan Sussman, Chau-Wen Tseng, and Sergei Sukharev, for taking the time to read my manuscript, view my presentations, and provide invaluable feedback on my work. In addition, I would like to thank my co-workers, the members of the Fushman Lab, for years of support, feedback, and creating a great work environment.

I would like to acknowledge all the people who helped with proofreading my drafts, including Vera Zolotaryova, Sergey Koren, and Elizabeth Bailey.

Finally, I would like to thank my family and friends for their support. Specifically, my mother for supporting me (financially and otherwise) through my countless years in graduate school and my wife Leah for the emotional supporting, patience, constant motivation, writing advice, and multiple proofreadings of my thesis.

My work was partially made possible by the NIH grant GM065334.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The fundamental mechanism of any system is determined by the way in which components of that system come together and interact. The fundamental component that allows inanimate molecules to function as a living system is DNA (deoxyribonucleic acid). DNA contains the instructions for the production of proteins, which in turn are the machinery that run the cells in a living organism.

Proteins are large molecules made up of amino acids, and they are responsible for all functionality of a living organism. For example, structural proteins provide support in bones and connective tissues. Protein enzymes act as catalysts for chemical reactions in the body. Hemoglobin proteins transport oxygen, and Rhodopsin proteins absorb photons in order to facilitate vision. Proteins are also used for signaling and communication, among other functions [40].

The synthesis of proteins is a multi-step process: Genes encoded in the DNA are transcribed into mRNA, which are then used by ribosomes to assemble a sequence of amino acids. The assembled sequence of amino acids then folds into a globular form that we call a protein. A specific protein's functionality is partially determined by the structure that it folds into, which is dependent on its physical environment as well as its amino acid sequence.

There are two primary ways of studying the functionality of proteins. We can look at the sequences of DNA and how they correlate with physically observable behavior, or we can observe protein behavior directly. Until recently, it was not possible to data mine DNA sequences for protein functionality due to lack of computer power and DNA sequencing data. With the tremendous growth of computing power over the past few decades and improvement in DNA sequencing techniques, it has finally become feasible to start analyzing DNA for clues about how living organisms function.

In DNA analysis, DNA is examined in hopes of finding relationships between sequences and the overall behavior of the system. DNA analysis has generated large amounts of data over the years, but has run into difficulty predicting which proteins are present in a living organism, and their functionalities [40]. The difficulty with predicting protein functionality from a DNA sequence is that DNA contains only the instruction for the amino acid sequence; it does not contain information on what structure that amino acid sequence will fold into, or information on how that protein will interact with other proteins – which determines the protein's functionality.

Only in the past few years has computer power developed to the point of allowing for direct analysis of protein behavior, rather than the indirect analysis of DNA. The key to understanding the functionality of a protein is to find how it interacts with other parts of an organism. One of the most common interactions with other proteins is through binding. Understanding how proteins bind to one other is especially critical to successful drug design. The orientation and positioning of the bound proteins relative to one another is referred to as "domain positioning", and

is a fundamental topic in structural biology.

A molecule that results from multiple proteins binding to one another is referred to as a "complex" or a "multi-domain protein", where "domain" refers to the individual protein.[1] Ideally, one would determine the complete structure of a multi-domain protein to ascertain how the domains bind to one another.

There are two primary methods for experimentally determining protein structure. Both of these methods are limited in their ability to determine structure for a large multi-domain protein.

The first method is X-ray crystallography. In X-ray crystallography, a crystal of the protein is grown and then exposed to a beam of X-ray radiation which produces a diffraction pattern. Using specialized software in conjunction with other constraint information, it becomes possible to analyze the diffraction pattern and determine the three-dimensional structure of the protein [40]. One of the fundamental drawbacks of X-ray crystallography is that the crystal structure may not represent the actual conformation of the molecule due to packing forces exerted during crystallization, and due to the fact that motion is almost completely restricted in crystals. Additionally, getting proteins to crystallize is a notoriously frustrating process that may not end with success.

The second method for experimentally determining protein structure is nuclear magnetic resonance (NMR) spectroscopy. This method involves placing the molecule

[1]Multi-domain protein could also refer to one single amino acid sequence whose different parts fold separately and then bind to each other. This is the reason why we refer to the protein as "multi-domain" rather than "multi-protein".

in a static magnetic field, exposing it to a second oscillating magnetic field, and then collecting and analyzing the resulting data. The device that houses and controls the magnets is called an NMR Spectrometer. In order to produce meaningful data the magnets must to be controlled by a very specific series of instructions, referred to as a pulse sequence. One of the properties that an NMR experiment measures is the Nuclear Overhauser Effect (NOE). NOE gives constraints on the distances between two atoms in a molecule. As in X-ray crystallography, the resulting data is used as constraints in a software package that computes the three-dimensional structure. The advantage of using NMR over X-ray crystallography is that the molecule may be studied in its natural state and environment. However, NMR is limited in its facility to analyze multi-domain proteins because of the scarcity of NOEs between inter-domain atoms. Even if NOEs are observed, the high rate of motion between the domains may make the data uninterpretable.

Even though there is scarce information between the multiple domains of a complex, usually there is significant NMR data between atoms that are inside the individual domains, which makes it possible to determine the structure of the individual domain but not the position of the domains relative to each other. To determine how the domains are positioned relative to each other, we can use global molecular properties. For example it has been shown that global NMR properties like the molecular alignment tensor [61, 4] and the diffusion tensor [13], and non-NMR data from Small-angle X-ray scattering [66, 17] are dependent on the shape of the molecule, which is directly dependent on the relative positions of the domains.

The proper positioning of domains in a multi-domain protein is referred to

as *protein-protein docking*. In *rigid protein-protein docking* it is assumed that the structure of the individual domains is known while in *flexible protein-protein docking* movement of atoms inside the domain and interacting regions is allowed. To solve the rigid docking problem, an energy function is created that rates the feasibility of a particular domain positioning. The more feasible the docking, the lower the energy function value. Global minimization is performed on this energy function to find the domain positioning that provides the lowest energy value.

A number of methods exist that perform two-domain docking *ab initio* (see e.g., [65, 54, 18]). The energy functions that these programs use are based on heuristics derived from evaluation of the surface complementarities, electrostatic interactions, van der Waals repulsion, etc.. The resulting energy functions are extremely complex, computationally expensive, and not convex, requiring the use of algorithms such as simulated annealing or genetic algorithms to search the entire space. The results are unreliable because of the stochastic nature of the algorithms, the difficulty in accurately ranking the multitude of possible solutions that those programs return, and the fact that the results are not backed by any observed experimental data.

To overcome the potential problems with *ab initio* docking, other docking methods have been developed that rely on experimental information instead of heuristics [26, 74, 75]. Those methods prove effective in determining structure when enough inter-domain constraints are available. Unfortunately, it might be difficult or impossible to measure inter-domain constraints for a large number of multi-domain proteins, due to weak interactions or movement between the domains.

In order to dock multi-domain proteins in the absence of inter-domain constraints we look at the molecule's global NMR properties. Since change in different parts of the molecule can affect its overall global properties, we can think of a molecule's global properties as "long range NMR constraints". One such global property is the (molecular) alignment tensor. The alignment tensor can be observed in an NMR experiment by introducing barriers into a solution, therefore biasing certain orientations of the molecule relative to others. The alignment tensor is a long range NMR constraint since bias in orientations depends on the molecule's overall shape. Another global property is the rotational diffusion tensor. The rotational diffusion tensor is a reflection of how fast a molecule is re-orientating around its axes in a solvent. Since the rate of re-orientation is related to the molecule's overall shape, the rotational diffusion tensor is also a long range NMR constraint.

In this thesis we develop and analyze two separate but similar methods for docking two-domain proteins based on long range NMR constraints. In the rest of this chapter we outline our contribution and introduce the computational concepts used in our methods. In Chapter 2, we describe and analyze PATI, a new computationally efficient method we developed for predicting the alignment tensor based on molecular shape. In Chapter 3, we describe PATIDOCK, a new computationally efficient method of docking based solely on the difference between the experimental alignment tensor derived from NMR data and the predicted alignment tensor computed by PATI. In Chapter 4, we present an improvement on ROTDIF, a method for computing the experimental rotational diffusion tensor, and present an improved docking method, ELMDOCK, that uses the difference between the

experimental rotational diffusion tensor computed by ROTDIF and the predicted rotational diffusion tensor to dock two-domain proteins. Finally, in Chapter 5 we sum up our work, and discuss possible future directions.

1.2 Contribution

The thesis contains three major contributions in the field of protein structure determination.

The first main contribution is presented in Chapter 2 and includes the following:

- We develop a new, computationally efficient method called PATI for computing the molecular alignment tensor based on the molecular shape.
  - We introduce and derive the formulas and methods for using numerical integration to reduce the dimensionality of the computation, improve the speed, and better control the accuracy of the result.
  - We introduce and develop the concept of using a convex hull instead of molecular shape to improve the speed of the method by significantly reducing the number of sample points that are used to represent molecular shape.
- We compare the accuracy of our method to that of the best known methods for computing the molecular alignment.
- We extensively analyze the types of errors in those methods and show that

the accuracy of PATI is equivalent to or better than all other methods.

The second major contribution is presented in Chapter 3 and includes the following:

- We introduce the novel idea of docking a two-domain complex based on the overall alignment tensor.
- We show that it is fundamentally possible to accurately dock a wide variety of proteins in an experimental setting, assuming perfect prediction of the alignment tensor.
- We develop a computationally efficient method called PATIDOCK for docking two-domain molecules based on the experimental alignment tensor and our developed method PATI for predicting the alignment tensor.
  - We introduce a way of combining and precomputing information to efficiently recompute the energy function under translational motion of the second domain.
  - We analytically derive the Jacobian of the energy function in order to efficiently minimize the energy function.
- We show that PATIDOCK is able to handle experimental errors.
- We analyze the accuracy of PATIDOCK on real experimental data, and combine PATIDOCK with additional experimental constraints to improve results.

The third major contribution is presented in Chapter 4 and includes the following:

- We computationally improve upon ROTDIF, a method for computing the experimental diffusion tensor.
- We develop a computationally efficient method, ELMDOCK, for docking two-domain molecules based on the experimental diffusion tensor.
  - We break the docking problem down into two components that are individually much faster to solve than the full problem.
  - We derive a method for efficiently computing the steps and the initial guess in the minimization of our energy function.
  - We derive a method for efficiently approximating our energy function, further speeding up the minimization.
- We analyze how robust ELMDOCK is to common experimental errors.
- We analyze the accuracy of ELMDOCK on real experimental data.

1.3 Numerical Background

In this section we review key numerical concepts that are used in our methods. Section 1.3.1 describes matrix properties, and Section 1.3.2 describes methods for deriving a simplified representation of a molecule. We heavily reference both of these sections when deriving our PATI, PATIDOCK, and ELMDOCK methods. Additional standard numerical methods are presented in the Appendix, which surveys numerical algorithms for minimization (Appendix A) and integration (Appendix B).

1.3.1 Properties of Symmetric Matrices

In this section we define and present properties of symmetric matrices that are used throughout the thesis. This section is fundamental to understanding the properties of both the diffusion and the alignment tensors, as both are expressed as symmetric matrices.

Definition 1.1 (Cardinality)**.** *If S is a set of n elements then $|S| = n$ is the cardinality of the set.*

Definition 1.2 (Frobenius-norm)**.** *If $\mathbf{A}$ is an $m \times n$ matrix, then the Frobenius-norm of $\mathbf{A}$ is*

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}. \tag{1.1}$$

We use the Frobenius-norm as the main method of quantifying the size of a matrix or the difference between two matrices.

Definition 1.3 (Symmetric Matrix)**.** $\mathbf{A}$ *is a symmetric matrix if and only if* $\mathbf{A} = \mathbf{A}^T$.

Both the alignment tensor and the diffusion tensor are 3×3 symmetric matrices, and therefore by definition have at most six degrees of freedom.

In order to efficiently use and analyze the alignment and diffusion tensors it is necessary to separate the tensor (alignment or diffusion) into orientational and magnitudinal components.

Definition 1.4 (Orthogonal Matrix)**.** $\mathbf{V}$ *is an orthogonal matrix if*

$$\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}. \tag{1.2}$$

If $\mathbf{V}$ *is a special orthogonal matrix then also* $\det \mathbf{V} = 1$.

Definition 1.5 (Sorted Eigendecomposition)**.** *Let* $\mathbf{A}$ *be a* 3×3 *symmetric matrix. Then the sorted eigendecomposition of A is*

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix}^T, \tag{1.3}$$

where $\lambda_1 \leq \lambda_2 \leq \lambda_3$ *are the principal values (eigenvalues),* $\mathbf{V}$ *is a special orthogonal matrix, and* V_1*,* V_2*,* V_3 *(each of dimension* 3×1*) are the associated directions of the principal components (eigenvectors).*

Using eigendecomposition we are able to separate the orientation (eigenvectors) from the magnitude (eigenvalues) in the alignment and diffusion tensors. Observe that we still have six degrees of freedom, with three parameters describing orientation (as we will explain in Definition 1.9), and three parameters describing magnitude of the tensors.

Eigendecompositions are not unique and can cause ambiguity when comparing orientation of multiple symmetric matrices.

Definition 1.6 (Four-Fold Ambiguity)**.** *Let* $\mathbf{A}$ *be a* 3×3 *symmetric matrix with a eigendecomposition* $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ *where no two eigenvalues are equal. Since the eigendecomposition of* $\mathbf{A}$ *is insensitive to the eigenvectors being pointed in the opposite direction, there are eight different sorted eigendecompositions of* $\mathbf{A}$*. If we only look at the sorted eigendecompositions where* $\mathbf{V}$ *is a special orthogonal matrix then*

there are four equivalent sorted eigendecompistions of $\mathbf{A}$*:*

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix}^T, \\
&= \begin{bmatrix} -\mathbf{V}_1 & \mathbf{V}_2 & -\mathbf{V}_3 \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} -\mathbf{V}_1 & \mathbf{V}_2 & -\mathbf{V}_3 \end{bmatrix}^T, \\
&= \begin{bmatrix} \mathbf{V}_1 & -\mathbf{V}_2 & -\mathbf{V}_3 \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} \mathbf{V}_1 & -\mathbf{V}_2 & -\mathbf{V}_3 \end{bmatrix}^T, \\
&= \begin{bmatrix} -\mathbf{V}_1 & -\mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} -\mathbf{V}_1 & -\mathbf{V}_2 & \mathbf{V}_3 \end{bmatrix}^T.
\end{aligned} \tag{1.4}$$

We refer to the fact that there are four equivalent eigendecompositions as four-fold ambiguity.

The alignment tensor, in addition to being a symmetric matrix, is also a traceless matrix.

Definition 1.7 (Trace)**.** *The trace of a matrix is the sum of its eigenvalues. If the trace is zero, the matrix is said to be traceless.*

The alignment tensor has five degrees of freedom, three in the orientation and two in the magnitude.

The diffusion tensor, unlike the alignment tensor, has a trace but its eigenvalues have to be positive, and therefore is a positive definite symmetric matrix.

Definition 1.8 (Positive Definite Matrix)**.** $\mathbf{A}$ *is a positive definite symmetric* 3×3 *matrix if and only if the eigenvalues* λ_1*,* λ_2*, and* λ_3 *are positive.*

In some cases it is useful to represent the orientation of a symmetric matrix by a set of angles instead of an orthonormal matrix.

Definition 1.9 (Euler Representation)**.** *If* $\mathbf{V}$ *is an orthogonal* 3×3 *matrix, then* $\mathbf{V}$ *can alternatively be represented by the three Euler angles* α,β, *and* γ *that define the Euler rotation* $\mathbf{R}$*, such that*

$$\begin{aligned}\mathbf{V} &= \mathbf{R}(\alpha, \beta, \gamma) \\ &\equiv \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\cos\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta \\ -\sin\beta\cos\gamma & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}.\end{aligned} \tag{1.5}$$

Note that there are multiple ways to define an Euler rotation matrix. We use this definition in ELMDOCK (see Section 4.2), but present an alternative definition in PATI (see Section 2.2.2).

Alternatively, any rotation can also be represented by an axis-angle representation. An axis-angle representation of a rotation parameterizes the rotation by two values: A unit vector indicating the orientation of the axis, $\mathbf{u}$, and an angle, θ, describing the magnitude of the rotation about that axis. The direction of rotation around the axis $\mathbf{u}$ is determined by the right-hand rule. One advantage of the axis-angle representation over Euler angles representation is that one can easily quantify the magnitude of the rotation by the size of the rotation angle θ.

Definition 1.10 (Angle of Axis-Angle Representation)**.** *Given a rotation matrix* $\mathbf{R}$*, the angle of the axis-angle representation* θ *is computed as*

$$\theta = \arccos(\frac{1}{2}(R_{11} + R_{22} + R_{33} - 1)). \tag{1.6}$$

1.3.2 Simplified Representations of a Molecule

A central theme in this thesis is the construction of a simplified representation of a molecule. Here we present four different methods, where the first three methods are based on finding an ellipsoidal representation of a molecule, and the last one is based on the convex hull of a molecule.

1.3.2.1 Minimum Volume Ellipsoid

The first method for finding an equivalent ellipsoid around a molecule is to find the minimum volume ellipsoid containing all of its atoms.

Definition 1.11 (Ellipsoid)**.** *An ellipsoid $\mathcal{E}$ in $\mathbb{R}^3$ is defined as*

$$\mathcal{E}(\mathbf{A}, \mathbf{c}) = \left\{ \mathbf{x} \mid (\mathbf{x} - \mathbf{c})^T \mathbf{A} (\mathbf{x} - \mathbf{c}) = 1 \right\},$$

where $\mathbf{A}$ is an 3×3 symmetric positive definite matrix that defines the shape of the ellipsoid, and $\mathbf{c} \in \mathbb{R}^3$ is its center.

The ellipsoid's principal semi-axes can be derived by a sorted eigendecomposition of $\mathbf{A}$, such that

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_1 \mathbf{V}_2 \mathbf{V}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \mathbf{V}_2 \mathbf{V}_3 \end{bmatrix}^T, \tag{1.7}$$

where lengths of the principal semi-axes are

$$\ell_1 = 1/\sqrt{\lambda_1}, \ell_2 = 1/\sqrt{\lambda_2}, \ell_3 = \sqrt{\lambda_3}, \tag{1.8}$$

and $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_3$ are their associated directions. If $\ell_2 = \ell_3$ then the ellipsoid is referred to as a *prolate ellipsoid*.

Definition 1.12 (MVE)**.** *The minimum volume ellipsoid (MVE) around a set of points $P \subset \mathbb{R}^3$, is an ellipsoid with the smallest volume that contains all points in P.*

The MVE has applications in wide variety of fields, including computational geometry, clustering, and statistics. Methods for deriving a MVE have been extensively studied and multiple methods for its computation have been proposed (see Introduction in [68]). The most intuitive method is to express the problem as a minimization problem [77]. The volume of an ellipsoid $\mathcal{E}$ is equal to

$$Vol(\mathcal{E}) = \frac{\pi^{3/2}}{\Gamma(3/2+1)} \frac{1}{\sqrt{|\det \mathbf{A}|}}, \tag{1.9}$$

where $\Gamma(\cdot)$ is the Gamma function. Therefore, solving the minimization problem:

$$\begin{aligned} &\arg\min_{\mathbf{A},\mathbf{c}} \log\left(\det \mathbf{A}^{-\frac{1}{2}}\right) \\ &\text{s.t. } (\mathbf{x}-\mathbf{c})^T\mathbf{A}(\mathbf{x}-\mathbf{c}) \leq 1 \ \forall \mathbf{x} \in P, \end{aligned} \tag{1.10}$$

yields a solution for $\mathbf{A}$ and $\mathbf{c}$.

Another approach to computing the MVE is a randomized algorithm that incrementally grows the ellipsoid by adding points [83].

1.3.2.2 Gyration Ellipsoid

The second method for constructing an equivalent ellipsoid is based on the gyration tensor of the molecule. This method has been suggested in Fernandes et al. [30].

Definition 1.13 (GE). *Given a set of N points $P \subset \mathbb{R}^3$ where p^m is the m-th point in the set, the center of the gyration ellipsoid is $c = [c_1, c_2, c_3]$, where*

$$c_i = \frac{1}{N} \sum_{m=1}^{N} p_i^m, \tag{1.11}$$

the gyration tensor is

$$G_{ij} = \frac{1}{N} \sum_{m=1}^{N} (p_i^m - c_i)(p_j^m - c_j), \;\; i, j = 1, 2, 3, \tag{1.12}$$

and the matrix that defines the shape of the ellipsoid is

$$\mathbf{A} = \mathbf{V} \begin{bmatrix} 1/(5\lambda_1) & 0 & 0 \\ 0 & 1/(5\lambda_2) & 0 \\ 0 & 0 & 1/(5\lambda_3) \end{bmatrix} \mathbf{V}^T, \tag{1.13}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of $\mathbf{G}$ and $\mathbf{V}$ is the matrix of the associated eigenvectors. The gyration ellipsoid (GE) of P is $\mathcal{E}(\mathbf{A}, \mathbf{c})$.

1.3.2.3 Principal Component Analysis Ellipsoid

A third method for finding an equivalent ellipsoid for an arbitrary molecule is based on the principal component analysis of the surface points of the molecule. First we observe that the correct representation of the surface of an object depends on what the object is interacting with. For example, a fly net is a solid surface to a fly, but to an air molecule, it is extremely porous. The protein's surface representation therefore depends on the type of solvent it is in. The larger the molecules of the solvent, the smoother the protein's surface. This concept was formulated by Richards in [55].

Definition 1.14 (Richards' smooth molecular surface)**.** *The Richards' smooth molecular surface of a molecule is the surface which an exterior probe-sphere touches as it is rolled over the spherical atoms of that molecule.*

Figure 1.1 shows the Cyanovirin-N molecule and its Richards' smooth molecular surface.

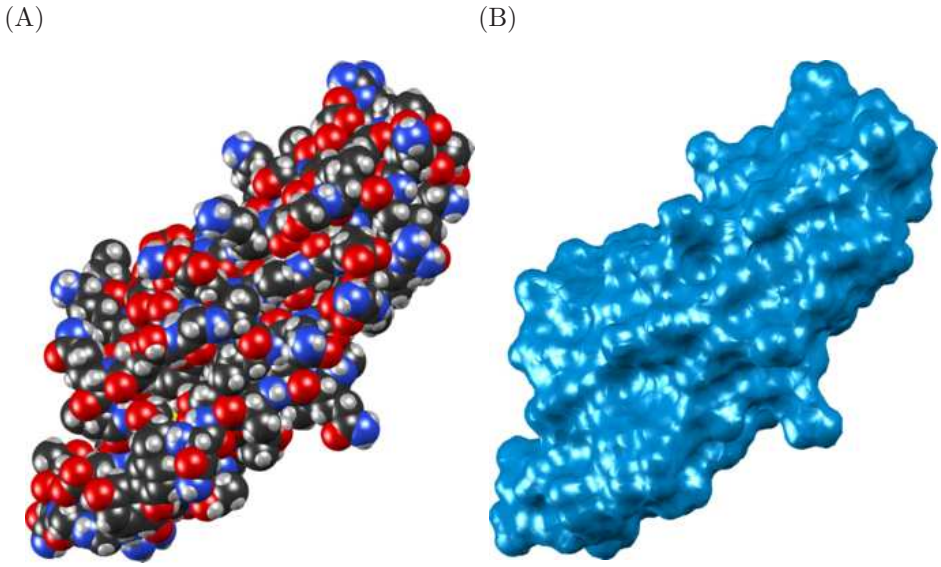

Figure 1.1: Visual representations of Cyanovirin-N. (A) Van der Waals surface of Cyanovirin-N. (B) Richards' smooth molecular surface of Cyanovirin-N in water.

To calculate the Richards' smooth molecular surface, we use the program designed by Varshney et al. [80, 79]. We refer to this program as SURF. SURF's usage and parameters are further discussed in Ryabov et al. [58].

To find a principal component analysis ellipsoid (PCAE), $\mathcal{E}^*$, of a molecule, we look for an ellipsoid that has the same covariance matrix as the triangulation points from the Richards' smooth molecular surface of the molecule. This method was first presented in Ryabov et al. [58].

A covariance matrix of the surface can be thought of as a simple description of the surface's shape. If the covariances are small, then the points are close to the center of the molecule, and the molecule is small. The larger the elements of the covariance matrix, the larger the molecule.

Define S to be a finite set of sample points from the Richards' smooth molecular surface[2] of the molecule M, where $|S| = n$ is the number of points in S, and $\mathbf{s}^k \in S$ is the k^{th} point.

The mean of S is

$$\mu_i = \frac{\sum_{v=1}^{n} s_i^v}{n}, \text{ for } i = 1, 2, 3, \tag{1.14}$$

the covariance matrix is

$$C_{i,j} = \frac{\sum_{v=1}^{n} s_i^v s_j^v}{n} - \mu_i \mu_j, \text{ for } i, j = 1, 2, 3, \tag{1.15}$$

and the sorted eigendecomposition of the covariance matrix is

$$\mathbf{C} = \mathbf{V} \begin{bmatrix} \hat{\lambda}_1 & 0 & 0 \\ 0 & \hat{\lambda}_2 & 0 \\ 0 & 0 & \hat{\lambda}_3 \end{bmatrix} \mathbf{V}^T. \tag{1.16}$$

By Theorem (C.1), the PCAE ellipsoid (an ellipsoid that has the same covari-

[2]We take the vertices from the surface mesh computed by Varshney et al.. [80, 79] as our surface points. The methods of sampling the surface is not limited to this particular method, and one could use more advanced techniques to get a set of surface points that could lead to more accurate computation of the diffusion tensor.

ance matrix) of S is

$$\mathcal{E}^* = \mathcal{E}\left(\mathbf{V}\begin{bmatrix} 1/(3\hat{\lambda}_1) & 0 & 0 \\ 0 & 1/(3\hat{\lambda}_2) & 0 \\ 0 & 0 & 1/(3\hat{\lambda}_3) \end{bmatrix}\mathbf{V}^T, \mu\right). \tag{1.17}$$

To compensate for the fact that water can attach to a molecule and form a hydration layer, Ryabov et al. [58] introduced the "Hydration Layer Thickness" (HLT) parameter. The parameter increases the radii of the atoms to simulate the attachment of water. The increase or decrease in the radii directly manipulates the equivalent ellipsoid. Figure 1.2 presents the Richard's molecular surface of ubiquitin/UBA complex [85] with no Hydration Layer Thickness (Figure 1.2A) and a Hydration Layer Thickness of 2.8Å (Figure 1.2B).

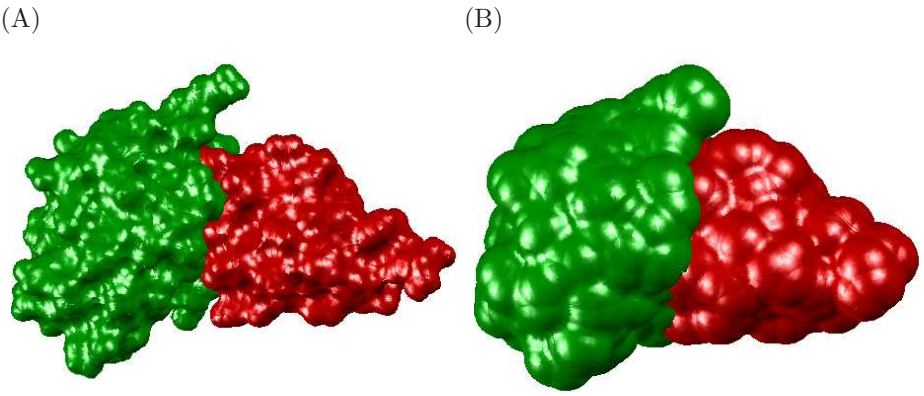

Figure 1.2: Richards' molecular surface of the ubiquitin/UBA complex [85] (PDB code 2JY6) in water. Domain A is drawn in green and domain B is drawn in red. (A) The surface with HLT=0Å. (B) The surface with HLT=2.8Å.

1.3.2.4 Convex Hull

The last method for simplifying the representation of the molecule is to compute the convex hull of the centers of the molecule's atoms.

Definition 1.15 (Convex Hull)**.** *The convex hull of a set of points $S \in \mathbb{R}^3$ is the boundary of the minimal convex set containing S.*

Intuitively, a convex hull in three dimension can be visualized as the surface of a plastic bag that has been tightly wrapped around a set of points in space. See Berg [22] for details on how to compute the convex hull. We compute the convex hull of molecules in PATI (Chapter 2) and PATIDOCK (Chapter 3).

Figure 1.3 shows the four simplified representations of a molecule for the specific case of the Cyanovirin-N molecule.

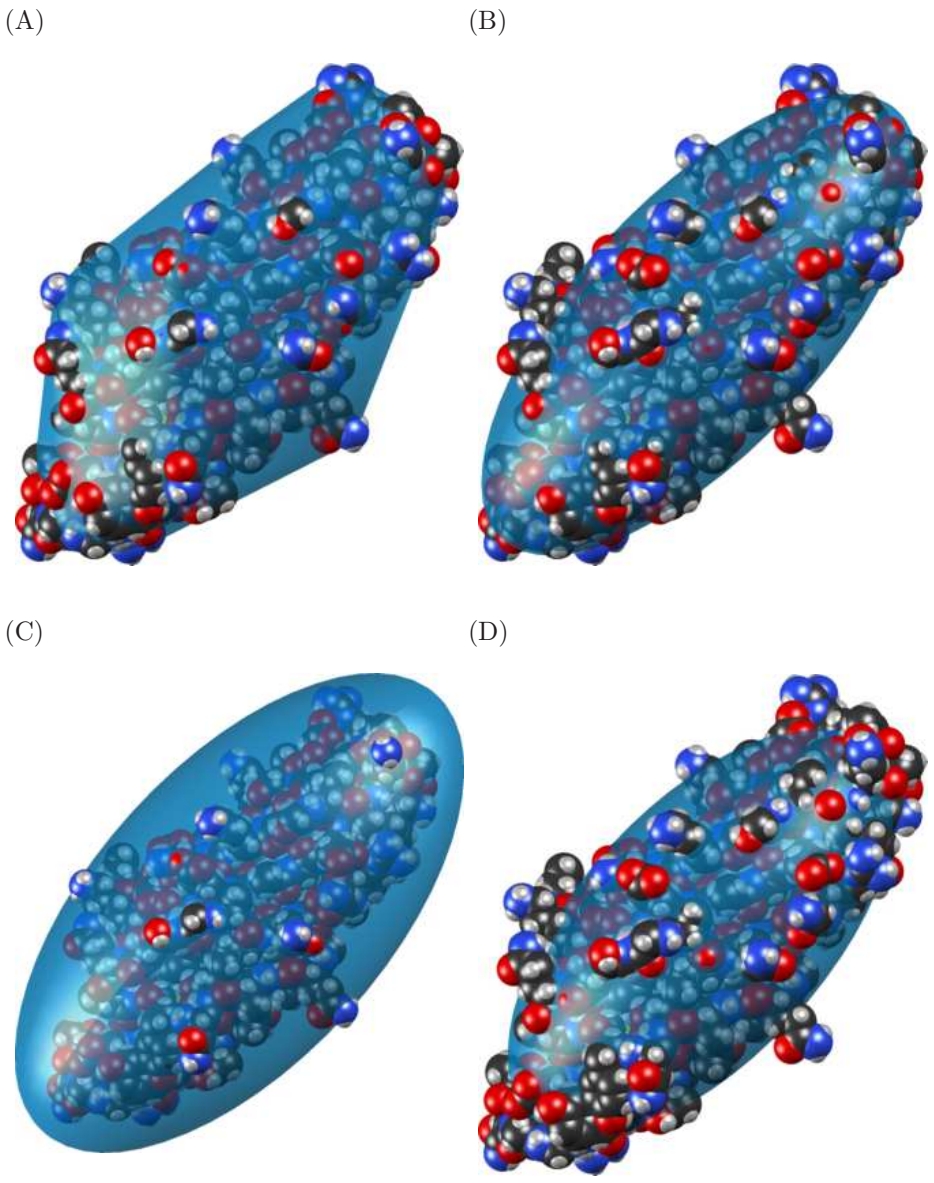

Figure 1.3: Convex hull and equivalent ellipsoids for the Cyanovirin-N molecule drawn on top of its van der Waals surface. (A) The convex hull around the molecule. (B) The GE representation. (C) The MVE representation. (D) The PCAE representation with HLT=0Å.

Chapter 2

Prediction of Alignment Tensor using Integration (PATI)

The work presented in this chapter is taken from Berlin et al. [6]. In this chapter we describe a new, computationally efficient method for computing the molecular alignment tensor based on the molecular shape. The increase in speed is achieved by re-expressing the problem as one of numerical integration, rather than a simple uniform sampling (as in the PALES method), and by using a convex hull rather than a detailed representation of the surface of a molecule. This method is applicable to bicelles, PEG/hexanol, and other alignment media that can be modeled by steric restrictions introduced by a planar barrier. This method is used to further explore and compare various representations of protein shape by an equivalent ellipsoid. We also examine the accuracy of the alignment tensor and residual dipolar couplings (RDC) prediction using various *ab initio* methods. We separately quantify the inaccuracy in RDC prediction caused by the inaccuracy in the orientation and in the magnitude of the alignment tensor, concluding that orientation accuracy is much more important in accurate prediction of RDCs.

2.1 Introduction

Knowledge of protein structure plays a critical role in our understanding of the molecular mechanisms underlying biological processes. One of the main methods for

obtaining structural information at atomic-level resolution is the use of nuclear magnetic resonance (NMR) spectroscopy for determining structural constraints. The NMR-derived constraints, such as NOEs, hydrogen bonds, and torsion angles, are intrinsically local or short-range and could be insufficient for accurate structure determination of biological macromolecules and their complexes due to the scarcity of long-distance structural information. Residual dipolar couplings (RDCs), resulting from partial alignment of solute molecules relative to the magnetic field, provide valuable structural information in terms of global, long-range orientational constraints [4]. A commonly used method for aligning molecules in solution takes advantage of the anisotropy of molecular shape by imposing steric restrictions on the allowed orientations of the molecule (e.g. by means of bicelles [67], stretched gels [70, 60], or PEG/hexanol-based media [56]). Such steric alignment can often be modeled as caused by planar obstacles, and we will refer to this simplified model of molecular alignment as the *barrier model*. The alignment of a rigid molecule can be described by the so-called molecular alignment tensor. Accurate prediction of the molecular alignment tensor, and with it of the RDCs, is important for NMR-based structure determination and validation as well as applications to dynamic and disordered systems (see e.g., [9, 82, 11, 2, 8, 49]). The sensitivity of the alignment tensor to molecular shape has the potential for improving structure characterization, especially in multi-domain systems and macromolecular complexes (e.g., [57]), by fully integrating RDC prediction into structure refinement protocols to directly drive structure optimization. Future progress in this direction critically depends on the efficiency and accuracy of the alignment tensor prediction.

Several methods for computing the molecular alignment tensor *ab initio*, i.e. based solely on the three-dimensional shape of the molecule, have recently been proposed. In the method by Zweckstetter and Bax [87, 86], implemented in a program called PALES, the alignment tensor is computed by uniformly sampling all orientations of a molecule (see e.g., [29]) at various distances away from a planar barrier, and averaging over only those orientations in which the molecule's surface does not collide with the barrier. The computational efficiency of this method is limited due to the fact that it must compute collisions between an arbitrary shape and a plane for every sample in the four-dimensional problem space.

Simpler methods based on the barrier model, but representing the shape of the molecule by an equivalent ellipsoid, have also been proposed. In Fernandes et al. [30], the alignment tensor was computed by approximating the molecule as an axially-symmetric prolate ellipsoid and analytically solving the barrier model for the alignment tensor. In Almond and Axelsen [1] and Azurmendi and Bush [3] the barrier method is also used, but the formulae are derived empirically.

Here we describe a new, computationally efficient method for computing the molecular alignment tensor based on the barrier model. The increase in speed is achieved by re-expressing the problem as one of numerical integration, rather than one of simple uniform sampling.[1] This formulation allowed us to simplify the problem by reducing its dimensionality from four to two. In addition to the reduction in computational complexity, numerical integration has the advantage of (i) allowing control over the size of numerical error, and (ii) allowing a more efficient

[1] See Appendix B for background on numerical integration.

sampling of the problem space [76]. Computational geometry techniques are used to increase the computational speed further. We will refer to our method as PATI (Prediction of Alignment Tensor using Integration). PATI can also be used with an equivalent ellipsoid of the molecule instead of the full surface. We will refer to this simplification as PATI-E. This simplified method is used to explore and compare various representations of protein shape by a (fully anisotropic) equivalent ellipsoid: based on the gyration tensor [30], the actual molecular surface [58], or the minimum-volume ellipsoid.

Finally, we examine the accuracy of the proposed methods (PATI and PATI-E) and the existing *ab initio* methods for RDC prediction. This analysis separately quantifies the effect of inaccuracy in the predicted RDCs caused by the inaccuracy in the orientation or in the magnitude of the alignment tensor. The results obtained for several proteins show that (i) the predicted RDCs and their agreement with experimental data are very sensitive to errors in orientation of the alignment tensor, and (ii) all *ab initio* prediction methods tested here give a rather crude estimate of the RDCs.

2.2 Theory

For a rigid molecule, the molecular alignment tensor $\mathbf{A}$ with respect to the magnetic field $\mathbf{B}$ is described by a 3×3 symmetric traceless matrix [87], sometimes referred to as the Saupe matrix [61], with the following elements ($i, j = 1, 2, 3$):

$$A_{ij} = \frac{1}{2} < F'_{ij} >, \quad F'_{ij} = 3 \cos\theta_i \cos\theta_j - \delta_{ij}, \tag{2.1}$$

where θ_i is the angle between molecular axis i and the magnetic field $\mathbf{B}$, $< ... >$ is the average over all possible orientations of the molecule in solution, and δ is the Kronecker delta.

The RDC value D_{PQ} for a specific bond PQ is related to the alignment tensor and the bond's orientation relative to the molecule's coordinate frame by the following equation:

$$\begin{aligned} D_{PQ} &= C_{PQ} \sum_{i,j} A_{ij} \cos\phi_i \cos\phi_j, \\ C_{PQ} &= -S_{LS} \frac{\mu_0 \gamma_P \gamma_Q \hbar}{4\pi^2 r_{PQ}^3}, \end{aligned} \tag{2.2}$$

where ϕ_i is the angle between the PQ bond and the molecular axis i, S_{LS} is the Lipari-Szabo generalized order parameter, μ_0 is the permeability of free space, γ_P and γ_Q are the gyromagnetic ratios of the corresponding nuclei, $\hbar$ is the reduced Planck's constant, and r_{PQ} is the length of the bond. PQ can represent bonds such as NH, $C_\alpha H_\alpha$, $C_\alpha C'$, and $C'N$. See Cavanagh et al. [16] for the values of constants in equation (2.2).

2.2.1 The Model for the Alignment Tensor

Given the three-dimensional structure of an arbitrary molecule, we will focus on the computation of its alignment tensor $\mathbf{A}$, defined in equation (2.1). We model the planar barrier causing steric alignment[2] of the molecule as a set of two infinite planes with the surface normals in the z direction, positioned at a distance $2h$ from each other. The molecule is centered around some point $\mathbf{m}$ (e.g., its center of mass),

[2]Alignment that is caused by the spatial constraints introduced by the barriers.

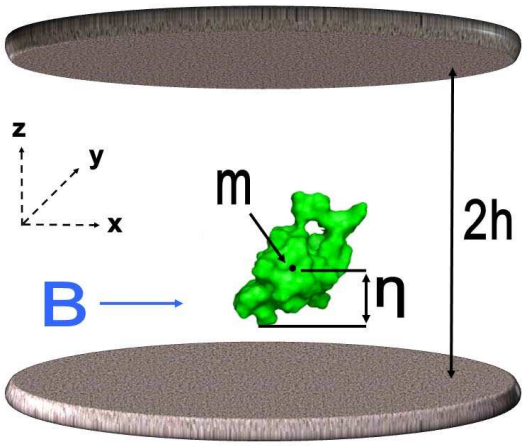

Figure 2.1: Planar barrier model for molecular alignment.

which lies somewhere inside the convex hull of the molecule's surface. The direction of the magnetic field is given by a unit vector $\mathbf{B}$, where

$$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \; b_1^2 + b_2^2 + b_3^2 = 1. \tag{2.3}$$

Figure 2.1 shows a schematic representation of the planar barrier model. Note that due to the symmetry of the system, the possible orientations of the molecule positioned between 0 and h along the z-axis are mirror images (over the $x - y$ plane) of the possible orientations when the molecule is between h and $2h$. Thus we can simplify the model by considering only the bottom plane and positioning the molecule's center at a height from 0 to h above this plane.

The orientation of the molecule's coordinate frame relative to the Cartesian coordinate system in Figure 2.1 can be defined by three Euler angles α, β, and γ, which determine the rotation matrix $\mathbf{R}(\alpha, \beta, \gamma)$. (See Section 1.3.1.)

For a specific molecule, we define S to be a finite set of sample points from its molecular surface (e.g. van der Waals surface or Richards molecular surface), and the center $\mathbf{m}$ of the molecule to be some point inside the convex hull of this molecular surface. Referring to Figure 2.1, to characterize the vertical extent of the molecule under the rotation $\mathbf{R}(\alpha, \beta, \gamma)$ around its center $\mathbf{m}$, we define $\eta(\alpha, \beta, \gamma)$, to be the difference between the z-coordinate of the center of the molecule and the minimum z-coordinate value of all the rotated points in S:

$$\eta(\alpha, \beta, \gamma) = -\min_{\mathbf{s_k} \in S}\{(\mathbf{R}(\alpha, \beta, \gamma)(\mathbf{s_k} - \mathbf{m})\,) \cdot \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\}, \tag{2.4}$$

where $\mathbf{s_k}$ gives the coordinates of the k-th point in S. Note that $\eta(\alpha, \beta, \gamma)$ sets the lower limit on the height of the center of the molecule at a given orientation.

We rewrite F, from equation (2.1), in terms of a general rotation matrix $\mathbf{R}$ and the magnetic field (2.3),

$$F'_{ij}(\alpha, \beta, \gamma) = 3(R_{1i}b_1 + R_{2i}b_2 + R_{3i}b_3)(R_{1j}b_1 + R_{2j}b_2 + R_{3j}b_3) - \delta_{ij}, \tag{2.5}$$

and average the F' values at height a with the mirror cases of $2h - a$ into one equation

$$\begin{aligned} \bar{F}_{ij}(\alpha, \beta, \gamma) &= \frac{3}{2}(R_{1i}b_1 + R_{2i}b_2 + R_{3i}b_3)(R_{1j}b_1 + R_{2j}b_2 + R_{3j}b_3) \\ &\quad + \frac{3}{2}(R_{1i}b_1 + R_{2i}b_2 - R_{3i}b_3)(R_{1j}b_1 + R_{2j}b_2 - R_{3j}b_3) - \delta_{ij}, \\ &= 3(R_{1i}b_1 + R_{2i}b_2)(R_{1j}b_1 + R_{2j}b_2) + 3R_{3i}R_{3j}b_3^2 - \delta_{ij}. \end{aligned} \tag{2.6}$$

Due to the symmetry of the system, $\bar{\mathbf{F}}$ can be used instead of $\mathbf{F'}$ to simplify our model to just one plane and a height from 0 to h.

For any rotation $\mathbf{R}(\alpha, \beta, \gamma)$, the center of the molecule cannot be located at a height between 0 and $\eta(\alpha, \beta, \gamma)$. Therefore, the range of interest is from $\eta(\alpha, \beta, \gamma)$

to h. The alignment tensor $\mathbf{A}$ is then computed by summing $\bar{\mathbf{F}}$ weighted by the probability of the current height and orientation, for all allowed orientations and heights from $\eta(\alpha, \beta, \gamma)$ to h.

We assume equal *a priori* probabilities of all orientations at all heights, and write the analytical expression for A_{ij} from equation (2.1). To obtain a uniform distribution of the Euler angles we multiply our integrand by the Jacobian $J = \sin\beta/(8\pi^2)$ [50] to obtain

$$A_{ij} = \frac{1}{2N} \int_{\gamma_0}^{\gamma_1} \int_{\beta_0}^{\beta_1} \int_{\alpha_0}^{\alpha_1} \int_{\eta(\alpha,\beta,\gamma)}^{h} \bar{F}_{ij} J \, dz \, d\alpha \, d\beta \, d\gamma, \tag{2.7}$$

where N is the normalization factor

$$N = \int_{\gamma_0}^{\gamma_1} \int_{\beta_0}^{\beta_1} \int_{\alpha_0}^{\alpha_1} \int_{\eta(\alpha,\beta,\gamma)}^{h} J \, dz \, d\alpha \, d\beta \, d\gamma, \tag{2.8}$$

and $[\alpha_0, \alpha_1]$, $[\beta_0, \beta_1]$, $[\gamma_0, \gamma_1]$ are the ranges in which α,β,γ are defined.

We observe that the approach taken in PALES is equivalent to solving equation 2.7 using uniform sampling. Uniform sampling is an inefficient method for solving equation (2.7), as compared to adaptive integration, since it requires a greater number of function evaluations and does not provide a way to control the error of the computation. See Appendix B for information on adaptive integration.

From a physical point of view, we should be able to eliminate two of the four integrals because the height of the molecule is insensitive to the rotation around the z-axis and the RDCs are constant in relation to molecule's translation. In the next section we reduce the four-dimensional problem in PALES to a two-dimensional problem.

2.2.2 Computation of the Alignment Tensor: General Case

In this section we show, using the Euler z-y-z rotation, that the expression for the alignment tensor $\mathbf{A}$ of an arbitrary molecule can be simplified from the quadruple integral to a double integral.

Define the Euler z-y-z rotation matrix as $\mathbf{R}(\alpha, \beta\,\gamma) = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_z(\alpha)$, where

$$\begin{aligned}
\mathbf{R}_z(\alpha) &= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{R}_y(\beta) &= \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \\
\mathbf{R}_z(\gamma) &= \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{2.9}$$

Multiplying the three matrices yields the full expression

$$\mathbf{R}(\alpha,\beta,\gamma) = \begin{bmatrix} \cos\gamma\cos\beta\cos\alpha - \sin\gamma\sin\alpha & -\cos\gamma\cos\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta \\ \sin\gamma\cos\beta\cos\alpha + \cos\gamma\sin\alpha & -\sin\gamma\cos\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta \\ -\sin\beta\cos\alpha & \sin\beta\sin\alpha & \cos\beta \end{bmatrix}. \tag{2.10}$$

We now write the equations for $A_{11}, A_{22}, A_{33}, A_{21}, A_{31}, A_{32}$, and N, recalling

that the Jacobian is $J = \sin\beta/(8\pi^2)$:

$$\begin{aligned}
A_{11} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{11}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
A_{22} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{22}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
A_{33} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{33}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
A_{21} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{21}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
A_{31} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{31}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
A_{32} &= \frac{1}{16N\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma)}^{h}\bar{F}_{32}\sin\beta\,dz\,d\alpha\,d\beta\,d\gamma,\\
N &= \frac{1}{8\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_{\eta(\alpha,\beta,\gamma,\gamma)}^{h}\sin(\beta)\,dz\,d\alpha\,d\beta\,d\gamma.
\end{aligned} \tag{2.11}$$

We observe that γ does not contribute to the vertical size of the molecule, and redefine $\eta(\alpha,\beta,\gamma)$ as $\eta(\alpha,\beta)$. Integrating by γ and z first gives us

$$\begin{aligned}
A_{11} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi}(3\cos^2\alpha\cos^2\beta - 3\cos\alpha + 1)(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
A_{22} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi} -(3\cos^2\alpha\cos^2\beta - 3\cos^2\alpha - 3\cos^2\beta + 2)\\
&\qquad\qquad \times (h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
A_{33} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi} -(3\cos^2\beta - 1)(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
A_{21} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi} 3\cos\alpha\sin\alpha\sin^2\beta(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
A_{31} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi} 3\cos\alpha\sin\beta\cos\beta(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
A_{32} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_0^{\pi} -3\sin\alpha\sin\beta\cos\alpha(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,
\end{aligned} \tag{2.12}$$

where

$$\begin{aligned}
N &= \frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi}(h-\eta(\alpha,\beta))\sin\beta\,d\beta\,d\alpha,\\
S_c &= 1 - 3b_3^2.
\end{aligned} \tag{2.13}$$

We perform a change of variable, $u = \cos\beta$, obtaining

$$\begin{aligned}
A_{11} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1}(3u^2\cos^2\alpha - 3\cos^2\alpha + 1)(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
A_{22} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1}(3u^2\cos^2\alpha + 3\cos^2\alpha - 2)(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
A_{33} &= \frac{S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1}(1 - 3u^2)(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
A_{21} &= \frac{3S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1}\sin\alpha\cos\alpha(1 - u^2)(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
A_{31} &= \frac{3S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1}u\cos\alpha\sqrt{1 - u^2}(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
A_{32} &= \frac{3S_c}{16N\pi}\int_0^{2\pi}\int_{-1}^{1} -u\sin\alpha\sqrt{1 - u^2}(h - \eta(\alpha, \arccos u))\,du\,d\alpha,\\
N &= \frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1} h - \eta(\alpha, \arccos u)\,du\,d\alpha,\\
S_c &= 1 - 3b_3^2.
\end{aligned} \tag{2.14}$$

Integrating the terms that do not involve η finally gives us:

$$\begin{aligned}
A_{ij} &= \frac{1}{N}\int_0^{2\pi}\int_{-1}^{1}F_{ij}(\alpha, u)\eta(\alpha, \arccos u)\,du\,d\alpha, \quad i, j = 1, 2, 3,\\
N &= h - \frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1}\eta(\alpha, \arccos u)\,du\,d\alpha,
\end{aligned} \tag{2.15}$$

where

$$\begin{aligned}
F_{11}(\alpha, u) &= \frac{S_c}{16\pi}[3(1 - u^2)\cos^2\alpha - 1],\\
F_{22}(\alpha, u) &= \frac{S_c}{16\pi}[3(1 - u^2)\sin^2\alpha - 1],\\
F_{33}(\alpha, u) &= \frac{S_c}{16\pi}(3u^2 - 1),\\
F_{21}(\alpha, u) &= -3\frac{S_c}{16\pi}(1 - u^2)\sin\alpha\cos\alpha,\\
F_{31}(\alpha, u) &= -3\frac{S_c}{16\pi}u\sqrt{1 - u^2}\cos\alpha,\\
F_{32}(\alpha, u) &= 3\frac{S_c}{16\pi}u\sqrt{1 - u^2}\sin\alpha,\\
S_c &= 1 - 3b_3^2.
\end{aligned} \tag{2.16}$$

Because $\mathbf{A}$ is a traceless symmetric tensor [87], only A_{11} and A_{22}, A_{21}, A_{31}, and A_{32} need to be computed, while $A_{33} = -(A_{11} + A_{22})$, $A_{12} = A_{21}$, $A_{13} = A_{31}$, and $A_{23} = A_{32}$. One can multiply the alignment tensor by -0.8 to account for the incomplete bicelle alignment, and to match the sign returned by PALES. The height h can be determined by the formula $d/(2V_f)$, where d is the barrier thickness (≈ 40Å for DMPC/DHPC bicelles) and V_f ($\ll 1$) is the sample volume fraction occupied by the barriers (see [87, 86]).

Thus, all one needs to know in order to compute the alignment tensor is $\eta(\alpha, \beta)$, defined in equation (2.4). Being an intrinsic geometric property of the molecule, $\eta(\alpha, \beta)$ can be computed separately, regardless of the barrier.

2.2.2.1 Computing η

In the PALES approach [87, 86], $\mathbf{A}$ is estimated based on forming a mesh of the molecular surface and then rotating all the mesh triangles of this surface to check if any part of the mesh is below the barrier. Observe that the complexity of each rotation is proportional to the number of triangles in the mesh. It is possible to reduce the computation using mesh simplification (see [43, 44]); however even this is overly complex. An infinite planar barrier is not sensitive to cavities on the surface of the molecule; therefore, a convex hull of the molecule is a sufficient representation of the molecule's surface. Additional mesh simplifications could be performed on the convex hull to further reduce the number of points.

To compute η for an arbitrary molecule under a rotation $\mathbf{R}$, we simply com-

pute the convex hull of the atom positions of the molecule and consider the vertices of the convex hull as the set S in equation (2.4). We add the van der Waals radius of the atom associated with the minimum z-value to equation (2.4) to form η for the rotation $\mathbf{R}$. Figure 1.3A shows the convex hull around the Cyanovirin-N molecule. The number of points used to represent the molecule drops dramatically, from 40708 in the molecular surface representation (see [80, 79]), to just 57 in the convex hull representation. For any rotation $\mathbf{R}$, the relative error in η between the two representations is less than 5% and the absolute error is less than 0.5Å. Also the alignment tensors and the RDCs predicted by PATI (our method) and PALES are almost identical, as shown below.

2.2.3 Special Case of an Ellipsoid

A potential simplification for computing the alignment tensor is to represent a molecule by an equivalent ellipsoid. In this section we derive the analytical expression for η for an arbitrary ellipsoid. Due to the symmetry of the ellipsoid we consider only one octant in our analysis, expressing all points p on the ellipsoid in that octant as

$$p(x, y) = (x, y, z(x, y)), \tag{2.17}$$

where

$$z(x, y) = c\sqrt{\left(1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}\right)}, \tag{2.18}$$

for $x \in [0, a]$ and $y \in [0, b\sqrt{(1 - x^2/a^2)}]$.

A rotation of the ellipsoid by $\mathbf{R}(\alpha, \beta, \gamma)$ transforms the coordinates of these

points into

$$\begin{aligned} x' &= R_{11}(\alpha,\beta,\gamma)x + R_{12}(\alpha,\beta,\gamma)y + R_{13}(\alpha,\beta,\gamma)z(x,y), \\ y' &= R_{21}(\alpha,\beta,\gamma)x + R_{22}(\alpha,\beta,\gamma)y + R_{23}(\alpha,\beta,\gamma)z(x,y), \\ z' &= R_{31}(\alpha,\beta,\gamma)x + R_{32}(\alpha,\beta,\gamma)y + R_{33}(\alpha,\beta,\gamma)z(x,y). \end{aligned} \tag{2.19}$$

We observe that

$$\eta(\alpha,\beta,\gamma) = z'(x_*, y_*), \tag{2.20}$$

where $x_*(\alpha,\beta,\gamma)$ and $y_*(\alpha,\beta,\gamma)$ minimize z'.

To find the minimum/maximum value of our rotated ellipsoid, we solve $\nabla z'(x,y) = 0$:

$$\frac{\partial z'(x,y)}{\partial x} = R_{31} - R_{33}\frac{cx}{a^2\sqrt{\left(1-\frac{x^2}{a^2}-\frac{y^2}{b^2}\right)}} = 0, \tag{2.21}$$

$$\frac{\partial z'(x,y)}{\partial y} = R_{32} - R_{33}\frac{cy}{b^2\sqrt{\left(1-\frac{x^2}{a^2}-\frac{y^2}{b^2}\right)}} = 0. \tag{2.22}$$

Let

$$x_* = \frac{R_{31}a^2}{\sqrt{R_{31}^2a^2 + R_{32}^2b^2 + R_{33}^2c^2}}, \tag{2.23}$$

$$y_* = \frac{R_{32}b^2}{\sqrt{R_{31}^2a^2 + R_{32}^2b^2 + R_{33}^2c^2}}. \tag{2.24}$$

It is easy to verify that x_* and y_* solve equations (2.21) and (2.22):

$$\begin{aligned}\frac{\partial z'(x_*,y_*)}{\partial x} &= R_{31} - R_{33}\frac{cx_*}{a^2\sqrt{\left(1-\frac{x_*^2}{a^2}-\frac{y_*^2}{b^2}\right)}} \\ &= R_{31} - R_{33}\frac{\frac{cR_{31}a^2}{\sqrt{R_{31}^2a^2+R_{32}^2b^2+R_{33}^2c^2}}}{\frac{a^2R_{33}c}{\sqrt{R_{31}^2a^2+R_{32}^2b^2+R_{33}^2c^2}}} = 0, \qquad (2.25)\\ \frac{\partial z'(x_*,y_*)}{\partial y} &= R_{32} - R_{33}\frac{cy_*}{b^2\sqrt{\left(1-\frac{x_*^2}{a^2}-\frac{y_*^2}{b^2}\right)}} \\ &= R_{32} - R_{33}\frac{\frac{cR_{32}b^2}{\sqrt{R_{31}^2a^2+R_{32}^2b^2+R_{33}^2c^2}}}{\frac{b^2R_{33}c}{\sqrt{R_{31}^2a^2+R_{32}^2b^2+R_{33}^2c^2}}} = 0. \qquad (2.26)\end{aligned}$$

Therefore, x_*, y_* minimizes z', and the optimal value of z is

$$z_* = \frac{R_{33}c^2}{\sqrt{R_{31}^2a^2 + R_{32}^2b^2 + R_{33}^2c^2}}, \qquad (2.27)$$

since $R_{31}^2 + R_{32}^2 + R_{33}^2 = 1$. Therefore, from equations (2.20) and (2.19), our solution is

$$\begin{aligned}\eta(\alpha,\beta,\gamma) &= R_{31}(\alpha,\beta,\gamma)x_* + R_{32}(\alpha,\beta,\gamma)y_* + R_{33}(\alpha,\beta,\gamma)z_* \\ &= \sqrt{R_{31}^2a^2 + R_{32}^2b^2 + R_{33}^2c^2},\end{aligned} \qquad (2.28)$$

where γ is arbitrary for the Euler rotation defined in equation (2.9).

2.3 Results

In this section we present a comprehensive comparison of several methods for computing RDCs *ab initio*. All the RDC data analyzed here are for the backbone NH bonds located in structurally well-defined regions of proteins, i.e. the α-helices and β-sheets. The RDC data were retrieved from the BMRB repository using the

PDB code of the molecule. Only the RDC values measured using the neutral bicelle alignment medium (or, in the case of the B3 domain of protein G, the PEG/hexanol-based medium) are used. The 9 proteins and their codes in the Protein Data Bank are listed in Table 2.1.

We assess the quality of our results by computing the *quality factor* between the vector of experimental RDCs, D_{exp}, and our predicted RDCs for those same bonds, D_{pred}, as [19]:

$$Q = \frac{\|D_{exp} - D_{pred}\|_F}{\|D_{exp}\|_F}. \tag{2.29}$$

Note that the predicted magnitude of the RDC values depends on the experimental conditions (which determine the barrier height h) and selection of values for equation constants, e.g. C_{PQ}. These factors affect all RDCs approximately uniformly, and hence can be represented by a scaling factor. Therefore, in order to make our analysis less sensitive to possible errors in experimental conditions and imperfect selection of values for constants, we also introduce the scaled quality factor to quantify the agreement between the experimental and predicted data with an unknown scaling factor. We define the *scaled quality factor* as

$$Q_s = \min_{\rho} \frac{\|D_{exp} - \rho D_{pred}\|_F}{\|D_{exp}\|_F}, \tag{2.30}$$

where the scalar ρ can be computed by linear least squares. (Note that both PATI and PALES can predict the magnitude of RDC values with reasonable accuracy. See Table D.1 for the values of ρ.)

First, we present the Q values for the experimental alignment tensor. We define the *experimental* alignment tensor, $\tilde{\mathbf{A}}$, as the alignment tensor that optimally fits

the data, i.e. gives the lowest Q value between the experimental and back-calculated RDC data. This quality factor allows us to examine whether the experimental data are well approximated by the theoretical equation for RDCs. We derive $\tilde{\mathbf{A}}$ by solving a linear least-squares problem of the form

$$\begin{bmatrix} (v_1^1)^2-(v_3^1)^2 & (v_2^1)^2-(v_3^1)^2 & 2v_1^1v_2^i & 2v_1^1v_3^1 & 2v_2^1v_3^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (v_1^i)^2-(v_3^i)^2 & (v_2^i)^2-(v_3^i)^2 & 2v_1^iv_2^i & 2v_1^iv_3^i & 2v_2^iv_3^i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (v_1^n)^2-(v_3^n)^2 & (v_2^n)^2-(v_3^n)^2 & 2v_1^nv_2^n & 2v_1^nv_3^n & 2v_2^nv_3^n \end{bmatrix} \begin{bmatrix} \tilde{A}_{11} \\ \tilde{A}_{22} \\ \tilde{A}_{12} \\ \tilde{A}_{13} \\ \tilde{A}_{23} \end{bmatrix} \approx \begin{bmatrix} D_{exp}^1/C_{NH} \\ \vdots \\ D_{exp}^i/C_{NH} \\ \vdots \\ D_{exp}^n/C_{NH} \end{bmatrix}, \tag{2.31}$$

where $\mathbf{v}^i = [v_1^i, v_2^i, v_3^i]$ is the normalized vector representing the orientation of the ith bond relative to the molecular coordinate frame, n is the number of bonds, and C_{NH} is the value of C_{PQ} in equation (2.2) for a NH bond. The linear least-squares problem can be solved by standard methods; see, e.g., [45].

Note that $\tilde{\mathbf{A}}$ can be decomposed into the experimental rotation (eigenvectors) $\tilde{\mathbf{V}}$ and experimental magnitudes (eigenvalues) $\tilde{A}_1, \tilde{A}_2, \tilde{A}_3$, where

$$\tilde{\mathbf{A}} = \tilde{\mathbf{V}}\tilde{\Lambda}\tilde{\mathbf{V}}^T = \begin{bmatrix} \tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2 & \tilde{\mathbf{V}}_3 \end{bmatrix} \begin{bmatrix} \tilde{A}_1 & 0 & 0 \\ 0 & \tilde{A}_2 & 0 \\ 0 & 0 & \tilde{A}_3 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2 & \tilde{\mathbf{V}}_3 \end{bmatrix}^T. \tag{2.32}$$

The Q values for the experimental alignment tensor derived using equation (2.31) are presented in Column 3 ("LS") in Table 2.1. The corresponding Q values for the best-fit alignment tensor derived from PALES are presented in Column 4, labeled PALES-LS. Naturally, $Q_s = Q$ for both methods. It is worth emphasizing

here that this quality factor measures the actual quality of the experimental data (i.e. how well they fit the theoretical equation for RDCs) and therefore provides the baseline Q value for subsequent evaluation of the prediction methods. Note also that the values in the parentheses, the relative error in the alignment tensor, confirm that equation (2.31) gives the same experimental alignment tensor $\tilde{\mathbf{A}}$ as the alignment tensor derived using PALES's best-fit algorithm.

Table 2.1 shows that the experimental RDC data are of high quality and consistent with the theoretical formulation of the RDC (equations (2.1–2.2)). This is not surprising given that these RDCs were used as constraints in the calculation/refinement of the corresponding protein structures. The quality of the agreement is illustrated in Figure 2.2A for Cyanovirin-N. (See also Figures D.1A–D.8A in the Appendix).)

The results of our *ab initio* calculations are presented in Table 2.2, for PATI, PALES, and for the ellipsoidal approximation methods using the MVE model. The MVE data are used in this table, as this model provides on average a slightly more accurate estimation of the alignment tensor compared to the other two equivalent ellipsoid models considered in this study.[3] Surprisingly, the scaled quality factor Q_s was rather high for all prediction methods, indicating a generally marginal agreement with experimental data, as illustrated in Figure 2.2D and Appendix Figures D.1D–D.8D. PATI and PALES calculations gave on average a slightly better agreement with the data compared to the other methods. It should be emphasized here that PATI gives almost identical results to PALES, as evident from Figure 2.3. In order

[3]The results for GE and PCAE are presented in Table D.2 and Table D.3

Table 2.1: Quality Factors $Q = Q_s$ for the Experimental Data

Protein	PDB[a]	LS[b,c]	PALES-LS[b,c,d]
Cellular factor BAF[14]	2ezx	0.03	0.03 (0.00)
B1 domain of protein G[41]	3gb1	0.05	0.05 (0.00)
B3 domain of protein G[71]	2oed	0.04	0.04 (0.00)
Rat apo-S100B[28]	1b4c	0.11	0.11 (0.00)
Cyanovirin-N[10]	2ezm	0.04	0.04 (0.00)
Gα interacting protein[21]	1cmz	0.08	0.08 (0.00)
Ubiquitin[19]	1d3z	0.04	0.04 (0.00)
Hen lysozyme[62]	1e8l	0.06	0.06 (0.00)
Oxidized putidaredoxin[39]	1yjj	0.08	0.08 (0.00)
Mean		0.06	0.06 (0.00)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for all calculations.

[b] Values represent the quality factor Q between the predicted and experimental data.

[c] Values represent the scaled quality factor Q_s between the predicted and experimental data.

[d] Values in parentheses represent the relative error between $\tilde{\mathbf{A}}$ and the experimental alignment tensor derived using PALES-LS.

to understand the reasons for the observed inaccuracy in our predictions, we now break down the contributions to the errors into those due to the eigenvalue of the

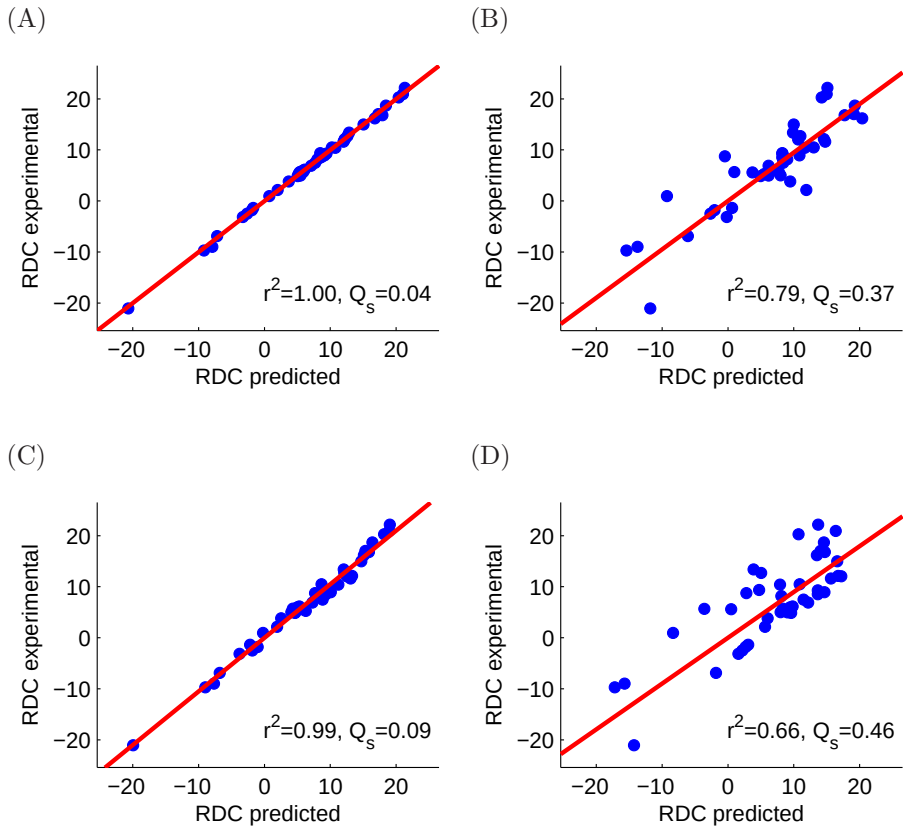

Figure 2.2: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in Cyanovirin-N, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated. Similar graphs for the rest of the molecules studied here can be found in Appendix D.

predicted alignment tensor and those due to inaccuracy in its orientation.

In Table 2.3 we compare the Q_s values for "synthetic" alignment tensors that

Table 2.2: Quality factors Q_s from RDC Prediction for *ab initio* Methods

PDB[a]	PATI[b,c]	PALES[b,c,e]	PATI-E[b,c,d]	Almond[b,c,d]	PROLFIT[b,c,d]
2ezx	0.26 (0.94)	0.27 (0.94)	0.19 (0.96)	0.20 (0.96)	0.12 (0.99)
3gb1	0.14 (0.99)	0.11 (0.99)	0.27 (0.96)	0.29 (0.95)	0.20 (0.97)
2oed	0.24 (0.98)	0.19 (0.98)	0.18 (0.98)	0.17 (0.98)	0.29 (0.97)
1b4c	0.22 (0.93)	0.22 (0.93)	0.43 (0.74)	0.42 (0.75)	0.55 (0.58)
2ezm	0.46 (0.66)	0.47 (0.66)	0.53 (0.56)	0.54 (0.54)	0.49 (0.61)
1cmz	0.32 (0.90)	0.30 (0.92)	0.38 (0.86)	0.39 (0.85)	0.37 (0.88)
1d3z	0.20 (0.93)	0.23 (0.91)	0.37 (0.81)	0.41 (0.77)	0.20 (0.91)
1e8l	0.31 (0.92)	0.31 (0.91)	0.42 (0.88)	0.43 (0.87)	0.26 (0.95)
1yjj	0.52 (0.75)	0.60 (0.67)	0.56 (0.76)	0.56 (0.75)	0.86 (0.34)
Mean	0.30 (0.89)	0.30 (0.88)	0.37 (0.84)	0.38 (0.83)	0.37 (0.80)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the scaled quality factor Q_s between the predicted and experimental data.

[c] Values in the parentheses represent the squared Pearson's correlation coefficient, r^2 (also known as coefficient of determination).

[d] MVE ellipsoidal representation was used.

[e] All PALES prediction calculations were run with options "-bic -H -dGrid 0.5 -rA 3.1".

have the same orientation as the *ab initio* calculated tensors but the correct (experimental) eigenvalues. We constructed these tensors by combining the rotation matrix $\mathbf{V}$ determined from our five models, GE, PCAE, MVE, PATI, and PALES,

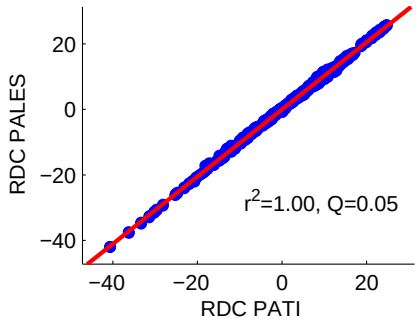

Figure 2.3: The agreement between RDC values predicted using PATI and those from PALES prediction. Shown are the $^1H^{15}N$ RDCs for all backbone amides for all molecules studied here. The (unscaled) quality factor Q between the two sets of RDC values is 0.05, the RMSD is 0.6 Hz, and the maximum deviation is 1.7 Hz.

with the experimental eigenvalues $\tilde{A}_1, \tilde{A}_2, \tilde{A}_3$ of the alignment tensor derived from $\tilde{\mathbf{A}}$. Such a comparison is expected to rank the methods based on the accuracy of prediction of the tensor's orientation. Since the orientation of $\mathbf{V}$ for the equivalent-ellipsoid-based methods is derived directly from the orientation of the ellipsoid, this table also provides a direct comparison of the ellipsoid models. Note that there are six different combinations for $\mathbf{V}$, since it is unknown *a priori* which $\tilde{A}_i$ is associated with which $\mathbf{V}_j$. The smallest of the six Q_s values is shown. Naturally, $Q_s = Q$ in this case.

As evident from Table 2.3, correcting the eigenvalues of the alignment tensor while keeping its predicted orientation did not improve the agreement with experimental data. (See also Figure 2.2B.) There are large variations among the various models in the accuracy of the predicted orientation of the alignment tensor. Of the three ellipsoid models tested here, MVE gave on average a somewhat better orientation (as documented in the Supplementary Material), while PATI and PALES

Table 2.3: Quality of Prediction for the Orientation of Alignment Tensor

PDB[a,d]	PATI[b,c,d]	PALES[b,c,d]	GE[b,c,d]	MVE[b,c,d]	PCAE[b,c,d],
2ezx	0.27 (10°)	0.28 (11°)	0.13 (5°)	0.11 (4°)	0.15 (5°)
3gb1	0.15 (17°)	0.12 (10°)	0.21 (12°)	0.21 (28°)	0.10 (14°)
2oed	0.23 (15°)	0.19 (13°)	0.33 (20°)	0.19 (14°)	0.25 (14°)
1b4c	0.25 (11°)	0.25 (11°)	0.68 (43°)	0.32 (15°)	0.62 (31°)
2ezm	0.37 (37°)	0.37 (39°)	0.45 (26°)	0.55 (33°)	0.41 (33°)
1cmz	0.31 (25°)	0.29 (23°)	0.33 (36°)	0.33 (24°)	0.36 (37°)
1d3z	0.19 (16°)	0.21 (16°)	0.42 (23°)	0.20 (29°)	0.29 (23°)
1e8l	0.38 (42°)	0.37 (41°)	0.33 (27°)	0.18 (16°)	0.33 (26°)
1yjj	0.53 (25°)	0.61 (30°)	0.87 (48°)	0.59 (26°)	0.85 (48°)
Mean	0.30 (22°)	0.30 (22°)	0.42 (27°)	0.30 (21°)	0.37 (26°)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the quality factor Q between the predicted and experimental data.

[c] Values represent the scaled quality factor Q_s between the predicted and experimental data.

[d] Values in the parentheses represent the angle difference between the orientation of the experimental and predicted tensors. (The angle was derived using the axis-angle representation of rotation. See Definition 1.10 for details.)

Table 2.4: Quality of Prediction for the Magnitude of Alignment Tensor

PDB[a]	PATI[b,c]	PALES[b,c]	PATI-E[b,c,d]	Almond[b,c,d]	PROLFIT[b,c,d]
2ezx	0.04 (1.00)	0.04 (1.00)	0.04 (1.00)	0.03 (1.00)	0.12 (0.99)
3gb1	0.06 (1.00)	0.05 (1.00)	0.09 (0.99)	0.11 (0.99)	0.20 (0.96)
2oed	0.04 (1.00)	0.04 (1.00)	0.04 (1.00)	0.04 (1.00)	0.16 (0.99)
1b4c	0.12 (0.98)	0.12 (0.98)	0.19 (0.96)	0.19 (0.96)	0.33 (0.88)
2ezm	0.09 (0.99)	0.08 (0.99)	0.08 (0.99)	0.05 (1.00)	0.25 (0.90)
1cmz	0.08 (0.99)	0.08 (0.99)	0.12 (0.98)	0.14 (0.98)	0.16 (0.98)
1d3z	0.06 (0.99)	0.08 (0.99)	0.19 (0.93)	0.23 (0.90)	0.17 (0.93)
1e8l	0.11 (0.99)	0.10 (0.99)	0.06 (1.00)	0.06 (1.00)	0.20 (0.97)
1yjj	0.29 (0.92)	0.31 (0.90)	0.31 (0.90)	0.29 (0.91)	0.20 (0.96)
Mean	0.10 (0.98)	0.10 (0.98)	0.12 (0.97)	0.13 (0.97)	0.20 (0.95)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the scaled quality factor Q_s between the predicted and experimental data. The smallest of the six possible values is shown.

[c] Values in the parentheses represent the squared Pearson's correlation coefficient, r^2.

[d] MVE ellipsoidal representation was used.

yielded generally similar results.

We then constructed "synthetic" alignment tensors that have the correct orientation (i.e. the $\tilde{\mathbf{V}}$ matrices derived from the experimental tensors $\tilde{\mathbf{A}}$) but the same eigenvalues (A_1, A_2, A_3) as the *ab initio* calculated tensors. Table 2.4 dis-

plays the Q_s values for five prediction methods, PATI, PALES, PATI-E, Almond, and PROLFIT. From this table, it is clear that using the correct orientation of the tensor dramatically improved the agreement with experimental data (cf. Table 2.2). This improvement is illustrated in Figure 2.2C for Cyanovirin-N and in the Supplementary Material for the other molecules.

Note that Column 3 ("PATI-E") and Column 5 ("PROLFIT") in Table 2.4 show that an additional degree of freedom provided by a fully anisotropic ellipsoid *versus* an axially-symmetric prolate ellipsoid approximation gives an improvement in the Q_s.

Thus, the analysis presented above demonstrates that accurate prediction of the orientation of the alignment tensor is critical for the agreement with experimental data. Accurate prediction of the eigenvalues of the tensor is important, too. However, when experimental RDCs are available, one can make an educated guess, based on the observed histogram/distribution of the data, about the magnitude of the tensor components (e.g., as described in [5]) and scale the predicted alignment tensor appropriately, whereas there is no obvious way to predict the orientation of the tensor.

2.4 Conclusions

We have reformulated the planar barrier model as a numerical integration problem and implemented it in a program called PATI. Our method has accuracy similar to PALES but is computationally more efficient and allows for finer control

over numerical error. In addition, the convex hull provides a simpler representation of the surface, thus further increasing the computational efficiency of the proposed method. This could allow PATI-based RDC prediction to be incorporated into the existing structure determination/refinement protocols. Because the molecular alignment tensor (and hence the RDC) is sensitive to the overall size and shape of the molecule, this would provide additional structural constraints that could potentially improve the accuracy of structure determination by NMR.

We compared several methods (old and new) for the computation of an equivalent ellipsoid of a molecule. We examined the accuracy of these equivalent ellipsoid models in predicting the alignment tensor and showed that the minimal volume ellipsoid gives on average a slightly better prediction of the alignment tensor orientation.

Finally, we compared all these methods against an extensive set of experimental RDC data. The analysis of the discrepancy between the experimental and predicted values emphasized the importance of the accurate prediction of the orientation of the alignment tensor. Possible sources of inaccuracy in *ab initio* alignment tensor prediction are the dynamic nature (structural flexibility) of protein molecules, not accounted for in the current prediction models, as well as the fact that the simple steric barrier model might not fully allow the correct alignment of all the molecules.

The increased efficiency in computation of the alignment tensor relative to PALES is not significant for a single computation, but will be very important when the computation is repeated a large number of times, as in the next chapter, where we use PATI to develop a molecular docking method based on the alignment tensor.

Chapter 3

Docking Based on the Alignment Tensor (PATIDOCK)

The work presented in this chapter is taken from Berlin et al. [7]. In this chapter we present and evaluate a rigid-body molecular docking method, called PATIDOCK, that relies solely on the three-dimensional structure of the individual components and the experimentally derived residual dipolar couplings (RDC) for the complex. We show that, given an accurate *ab initio* predictor of the alignment tensor from a protein structure, it is possible to accurately assemble a protein-protein complex by utilizing the RDC's sensitivity to molecular shape to guide the docking. The proposed docking method is robust against experimental errors in the RDCs and computationally efficient. We analyze the accuracy and efficiency of this method using experimental or synthetic RDC data for several proteins, as well as synthetic data for a large variety of protein-protein complexes. We also test our method on two protein systems for which the structure of the complex and steric-alignment data are available (Lys48-linked diubiquitin and a complex of ubiquitin and a ubiquitin-associated domain) and analyze the effect of flexible unstructured tails on the outcome of docking. The results demonstrate that it is fundamentally possible to assemble a protein-protein complex based solely on experimental RDC data and the prediction of the alignment tensor from three-dimensional structures. Additionally we show a method for combining RDCs with other experimental data,

such as ambiguous constraints from interface mapping, to further improve structure characterization of the protein complexes.

3.1 Introduction

Detailed understanding of molecular mechanisms underlying biological function requires knowledge of the three-dimensional structure of biomacromolecules and their complexes. Nuclear magnetic resonance (NMR) spectroscopy is one of the main methods for obtaining information on molecular structure and interactions at atomic-level resolution [16]. A major challenge in using NMR for accurate structure determination of multidomain systems and macromolecular complexes is the limited amount of long-distance structural information. Intermolecular Nuclear Overhauser Effect (NOE) contacts are often scarce, difficult to detect, and could be affected by intermolecular motions. Chemical shift perturbation (CSP) mapping is another powerful method for general identification of the interface. However, its informational content is highly ambiguous because CSPs do not identify pair-wise contacts and should be used with caution, since a perturbation of the local electronic environment of a nucleus does not necessarily indicate direct involvement of the corresponding atom in the interactions. Moreover, both NOEs and CSPs are limited to the contact area and could be insufficient for accurate spatial arrangement of the interacting partners. Residual dipolar couplings (RDCs), resulting from partial molecular alignment in a magnetic field [69, 67], could supplement the scarce interdomain data, because they contain valuable structural information in terms of

global, long-range orientational constraints (reviewed in [4]). In addition, RDCs also inevitably reflect (hence are sensitive to) the physical properties of the solute molecule responsible for its alignment. Thus, a commonly used method for aligning proteins in solution takes advantage of the anisotropy of molecular shape by imposing steric restrictions on the allowed orientations of the molecule. Such steric alignment can often be modeled as caused by planar obstacles (see e.g., [67, 87]); we will refer to this simplified model of molecular alignment as the *barrier model* (See Section 2.2.1).

The alignment of a rigid molecule can be characterized by the so-called alignment tensor. Several methods have been developed in [87, 30, 1, 3], and in Chapter 2, to use the barrier model for predicting the alignment tensor (and with it the RDCs) either directly from the 3D shape of the molecule or indirectly, using an ellipsoid representation. The RDCs' sensitivity to molecular shape has the potential for improving structure characterization, especially in multi-domain systems and macromolecular complexes, by fully integrating RDC prediction into structure refinement protocols to directly drive structure optimization. In fact, RDCs have been used to orient domains and bonds relative to each other either directly, using rigid-body rotation [31, 64, 27, 78, 36], or by incorporating RDCs as orientational restraints into protein docking [73] (see e.g., the reviews [12, 38]). However, none of these methods has used the information on the shape of the molecule (including not only the intervector/interdomain orientation but also the actual positioning of the individual domains) embedded in the measured RDCs.

Another physical property sensitive to molecular shape is the overall rotational

diffusion tensor, characterizing the rates and anisotropy of the overall tumbling of a molecule in solution. Interestingly, although they reflect distinct physical phenomena (rotation versus orientation) the diffusion and the alignment tensors are oriented similarly, provided the alignment is caused by neutral planar obstacles [20]. As demonstrated recently by Ryabov and Fushman [57], the sensitivity of the overall rotational diffusion tensor to molecular shape can be utilized to guide molecular docking. One would expect that the alignment tensor could be used similarly. Given that accurate RDC measurements for a wide variety of bond vectors are readily available, the use of the alignment tensor to guide molecular assembly could be of significant value for a broad range of macromolecular systems. However, to our knowledge, the ability to dock molecules using the alignment tensor has not been demonstrated, and RDCs have never been used to completely drive molecular docking, i.e. not only orient but also properly position molecules/domains relative to each other in a complex.

In this chapter we demonstrate that it is possible to determine the structure of a complex by utilizing the sensitivity of RDCs to molecular shape, provided that the structures of the individual components of the complex are available. We describe a method for rigid-body molecular docking based solely on the orientation- and shape-related information embedded in the experimental RDCs/alignment tensor of the complex. This method, called PATIDOCK, uses PATI, the method described in Chapter 2, for *ab initio* prediction of the alignment tensor from the three-dimensional shape of a molecule. We demonstrate that PATIDOCK can deterministically and efficiently perform rigid-body docking based on the alignment

tensor. In addition, we analyze the robustness of PATIDOCK under certain types of experimental errors, examine its performance in applications to real experimental data, and discuss challenges and various ways of refining the results by including other available experimental restraints and integrating our method into more sophisticated docking approaches.

3.2 Methods

Here we present a method, called PATIDOCK, for rigid-body assembly of a molecule made up of two distinct sets of atoms (hereafter called domains) whose structures are known, by using experimental RDC values exclusively. The method is based on first rotating/aligning the two domains using the corresponding subsets of the RDC values (see e.g., [31, 27, 36]) and then translating/positioning them relative to each other in order to minimize the difference between the predicted $\mathbf{A}$ and the experimental $\tilde{\mathbf{A}}$ alignment tensors. $\mathbf{A}$ is computed for the complex using the barrier-model-based algorithm PATI, while $\tilde{\mathbf{A}}$ is derived directly from the RDC values, measured for the whole molecule, using a linear least squares approach (see e.g., Chapter 2, [45]) and the (already aligned) 3D structures of the individual domains. As discussed in Chapter 2, PATI predicts RDCs with the same accuracy as the program PALES [87], while its computational efficiency is achieved by using numerical integration and a convex hull representation of the molecular surface. Note that while some parts of the docking algorithm are specific to the use of PATI, the general algorithm and key concepts can be applied to any current or future

method for alignment tensor prediction.

3.2.1 Formulation

We formulate the docking algorithm as a minimization problem. The algorithm is based on minimizing the difference between the *predicted alignment tensor* $\mathbf{A}$, computed based on the structure/shape of the molecule, and the *experimental alignment tensor* $\tilde{\mathbf{A}}$, derived directly from the experimental RDC values.

Let the set S of atoms of a molecule be subdivided into two distinct sets (domains), S_1 and S_2, such that $S_1 \cap S_2 = \emptyset$, $S_1 \cup S_2 = S$, no RDC-active bond is shared between the two sets, and each set contains enough bond vectors/RDCs associated with it to provide a proper sampling of the orientational space required for accurate determination of the alignment tensors [34]. We define $\mathbf{A}(\mathbf{R}_c, \mathbf{x})$ as the predicted alignment tensor of S, where the coordinates of atoms in S_1 remain static and the coordinates of atoms in S_2 are rotated by some rotation matrix $\mathbf{R}_c$ and then translated by $\mathbf{x} = [x_1, x_2, x_3]$. Our goal is to first properly orient the two sets by finding the *optimal rotation matrix*, $\mathbf{R}^*$, and then to find the *optimal translation vector* $\mathbf{x}^*$ that minimizes the difference between $\mathbf{A}(\mathbf{R}^*, \mathbf{x})$ and $\tilde{\mathbf{A}}$. The separation of orientation from translation is possible because inter-domain orientation can be obtained directly from the experimental RDCs and bond vectors for each set [31, 27, 36], regardless of their relative position.

To solve for $\mathbf{R}^*$ we simply align S_1 and S_2 relative to each other using experimental RDC data, as described in [31, 27, 36]. We first compute the experimental

alignment tensors, $\mathbf{A}_1$ and $\mathbf{A}_2$, of S_1 and S_2, respectively. The alignment tensors have eigendecompositions $\mathbf{A}_1 = \mathbf{R}_1\mathbf{D}_1\mathbf{R}_1^T$ and $\mathbf{A}_2 = \mathbf{R}_2\mathbf{D}_2\mathbf{R}_2^T$, where $\mathbf{R}_1$, $\mathbf{R}_2$ are rotation matrices (orthogonal matrices with determinant of 1) and $\mathbf{D}_1$, $\mathbf{D}_2$ are the diagonal matrices of principal components of the corresponding alignment tensors. Therefore, $\mathbf{R}^*$ can be derived by solving the equation $\mathbf{R}^*\mathbf{R}_2 = \mathbf{R}_1$:

$$\mathbf{R}^* = \mathbf{R}_1\mathbf{R}_2^T. \tag{3.1}$$

Note that due to orientational degeneracy of the alignment tensor there is a four-fold ambiguity in the relative alignment of domains, hence four possible solutions for $\mathbf{R}^*$[36]. One can find these possible solutions by computing an eigendecomposition of $\mathbf{A}_2$, determining the four assignments of signs to the columns of $\mathbf{R}_2$ that make $\det(\mathbf{R}_2) = 1$, and using equation (3.1) for each one.

Knowing the optimal rotation matrix $\mathbf{R}^*$, we find the optimal translation vector $\mathbf{x}^*$ by solving a nonlinear least squares problem. Since $\mathbf{R}^*$ is derived directly from the experimental RDC data independent of $\mathbf{x}^*$, in the rest of the chapter (except for the last sections) we assume that the two subsets are already properly aligned and simplify the notation from $\mathbf{A}(\mathbf{R}_c, \mathbf{x})$ to $\mathbf{A}(\mathbf{x})$. Our nonlinear least squares problem is then formulated as:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \chi^2(\mathbf{x}), \tag{3.2}$$

where the target function is defined as

$$\chi^2(\mathbf{x}) = \sum_{i,j=1}^{3} \left(A_{ij}(\mathbf{x}) - \tilde{A}_{ij} \right)^2. \tag{3.3}$$

and the computation of $\mathbf{A}(\mathbf{x})$ is described in the next section.

3.2.2 Efficient Computation of the Alignment Tensor

In this section we reformulate PATI, from the formulae presented in equation (2.15), to one that can be efficiently recomputed multiple times on S under different translations of S_2.

Since the molecule consists of two domains with an unknown translation $\mathbf{x}^*$ between them, η will depend on translation $\mathbf{x}$, α, and u. (This implies that $\mathbf{A}$ and N also depend on $\mathbf{x}$.) Therefore, we modify our Chapter 2 notation from $\eta(\alpha, \beta)$ to $\eta(\mathbf{x}, \alpha, u)$, where $\mathbf{x}$ is the vector of translation of the coordinates of all atoms of S_2.

Without loss of generality, let the center of S_1 be at $\mathbf{0}$, and the center of S_2 be at $\hat{\mathbf{m}}$, both of which are inside their associate convex hulls. We compute η for S_1 and S_2 separately, and call them $\eta_1(\alpha, u)$ and $\eta_2(\alpha, u)$. Note that $\eta_1(\alpha, u)$ and $\eta_2(\alpha, u)$ do not depend on $\mathbf{x}$. The combined $\eta(\mathbf{x}, \alpha, u)$ of the two sets (domains) is the largest of the two η, where η_2 is adjusted to reflect that S_2 is centered at $\hat{\mathbf{m}} + \mathbf{x}$, and is computed as

$$\eta(\mathbf{x}, \alpha, u) = \begin{cases} \eta_1(\alpha, u) & \text{if } \eta_1(\alpha, u) \geq \eta_2(\alpha, u) - \Upsilon(\mathbf{x}), \\ \eta_2(\alpha, u) - \Upsilon(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{3.4}$$

where

$$\Upsilon(\mathbf{x}) = \sum_{i=1}^{3} R_{3i}(\alpha, \arccos u, 0)(\hat{m}_i + x_i). \tag{3.5}$$

Precomputing $F(\alpha, u)$ (equation (2.16)), $\eta_1(\alpha, u)$, $\eta_2(\alpha, u)$, and $\mathbf{R}(\alpha, \arccos u, 0)$ for a fine enough set of $[\alpha, u]$ allows us to quickly compute $\mathbf{A}(\mathbf{x})$ for multiple values of $\mathbf{x}$.

3.2.3 Algorithm

In this section we describe how to solve the minimization problem posed in equation (3.2). We use a nonlinear least squares solver, specifically the Levenberg-Marquardt algorithm [46], due to the limited number of local minima, local convexity, and smoothness of our target function. The Levenberg-Marquardt method allows us to find the solution with many fewer function evaluations than direct search algorithms like simulated annealing because we can efficiently compute a good descent direction for our problem.

An efficient nonlinear least squares solver requires a Jacobian to be computed, or approximated using finite differences. Fortunately in this case, the Jacobian elements can be computed analytically:

$$\begin{aligned}\frac{\partial A_{ij}(\mathbf{x})}{\partial x_k} = \frac{1}{N(\mathbf{x})}\int_0^{2\pi}\int_{-1}^{1} F_{ij}(\alpha,u)\frac{\partial \eta(\mathbf{x},\alpha,u)}{\partial x_k}\,du\,d\alpha \\ + \frac{A_{ij}(\mathbf{x})}{4\pi N(\mathbf{x})}\int_0^{2\pi}\int_{-1}^{1}\frac{\partial \eta(\mathbf{x},\alpha,u)}{\partial x_k}\,du\,d\alpha,\end{aligned} \tag{3.6}$$

where

$$\frac{\partial \eta(\mathbf{x},\alpha,u)}{\partial x_k} = \begin{cases} 0 & \text{if } \eta_1(\alpha,u) \geq \eta_2(\alpha,u) - \Upsilon(\mathbf{x}), \\ -R_{3k}(\alpha, \arccos u, 0) & \text{otherwise,}\end{cases} \tag{3.7}$$

and $i,j,k = 1,2,3$.

Due to translational symmetry of the problem, there can be two significant local minimizers of our target function: the actual minimizer, and the incorrect minimizer where domain S_2 is located on the opposite side of domain S_1 (see e.g., Figure 3.4 in the Results section). In addition, if the convex hull of S_2 is fully inside S_1 then our target function has derivatives of 0, and the minimization algorithm

might become trapped on a plateau. Therefore, picking the right set of initial guesses is important.

To assure that the convex hull of S_2 is not inside S_1 we place any initial starting point x_0^i at a distance $d = \max_{\alpha,u} \eta_1(\alpha, u)$ from the center of S_1. We pick a set of six initial positions, $[d, 0, 0]$, $[-d, 0, 0]$, $[0, d, 0]$, $[0, -d, 0]$, $[0, 0, d]$, and $[0, 0, -d]$, to make sure that during the minimization we approach S_1 from different directions and therefore are likely to find all the minimizers. We refer to this method for finding the optimal translation between two domains as PATIDOCK-t. Additionally, we refer to the method that first aligns the two domains using equation (3.1) and then finds the optimal translation using PATIDOCK-t as PATIDOCK.

3.2.4 Additional Constraints

As demonstrated in Chapter 2, there is inaccuracy in barrier model-based prediction of the alignment tensor of a molecule. This inaccuracy would contribute to errors in the docking solution if we just minimized the target function $\chi^2(\mathbf{x})$ (equation (3.3)). In order to mimic a real situation, when additional experimental data are available, we examine whether the RDC-based docking could be improved by introducing additional restraints to enforce intermolecular distance constraints and avoid steric clashes.

Obviously, introduction of specific intermolecular distance constraints (e.g. from NOEs) would significantly improve docking by positioning the corresponding atoms (hence the domains carrying them) at the proper distance from each other.

However, intermolecular NOEs are often unavailable or averaged out by molecular motions such as domain dynamics, on/off rates, etc. Therefore, we analyze the effect of adding "milder", ambiguous restraints, often used for molecular docking based on interface mapping [25, 24] using chemical shift perturbations (CSPs). CSPs quantify NMR signal shifts in the presence of a binding partner, and their observation represents the basic and perhaps the simplest way to monitor intermolecular interactions by NMR. The CSPs provide a general qualitative map of atoms/residues involved in the interface, without any specific information about pair-wise contacts. Thus, we construct a "CSP-like" energy function based on ambiguous information of intermolecular contacts. To prove the concept of including additional constraints into RDC-guided docking, we forgo the complicated modeling and data refinement of the actual CSPs. Instead we simply label an atom as being "CSP-active" if the CSP for it is significantly high. For the molecules for which we do not have CSP data, for simple testing purposes we generate a synthetic CSP-active list by selecting all the atoms in one domain that are within a certain distance, d_Ω, of any atom in the other domain, and would therefore potentially experience a CSP in an experimental setting. We define the subsets of atoms from S_1 and S_2 that are CSP-active as I_1 and I_2 respectively.

Let $D_{ij}(\mathbf{x})$ be the distance between two atoms, $s_i \in S_1$ and $s_j \in S_2$, when the atoms in S_2 are translated by $\mathbf{x}$. To generate the energy function for the CSP-like constraints we weigh an atom in the CSP-active set as 0 if it is currently interacting with atoms in the other domain; otherwise we assign some penalizing value as the atom's weight. To handle outliers we stop the growth of the penalty at a cutoff

distance d_Ω^{cut}. Specifically, the CSP-active weights for the two domains are

$$\Omega_1^i(\mathbf{x}) = \begin{cases} 0 & \text{if } \min\limits_j D_{ij}(\mathbf{x}) \leq d_\Omega \text{ or } s_i \notin I_1, \\ \min\limits_j D_{ij}(\mathbf{x}) - d_\Omega & \text{if } d_\Omega < \min\limits_j D_{ij}(\mathbf{x}) \leq d_\Omega^{cut} \text{ and } s_i \in I_1, \\ d_\Omega^{cut} - d_\Omega & \text{otherwise,} \end{cases} \tag{3.8}$$

and

$$\Omega_2^j(\mathbf{x}) = \begin{cases} 0 & \text{if } \min\limits_i D_{ij}(\mathbf{x}) \leq d_\Omega \text{ or } s_j \notin I_2, \\ \min\limits_i D_{ij}(\mathbf{x}) - d_\Omega & \text{if } d_\Omega < \min\limits_i D_{ij}(\mathbf{x}) \leq d_\Omega^{cut} \text{ and } s_j \in I_2, \\ d_\Omega^{cut} - d_\Omega & \text{otherwise.} \end{cases} \tag{3.9}$$

We sum the average weights to form the target function for the CSP-like interactions:

$$\chi_\Omega^2(\mathbf{x}) = \sum_i \frac{[\Omega_1^i(\mathbf{x})]^2}{|I_1|} + \sum_j \frac{[\Omega_2^j(\mathbf{x})]^2}{|I_2|}, \tag{3.10}$$

where $|\cdot|$ is the cardinality of the set.

To prevent physically impossible overlap (steric clash) of the domains we assign a penalizing value to atoms that are closer than a given distance d_Ψ to atoms in the opposing domain. The weights

$$\Psi_1^i(\mathbf{x}) = \begin{cases} d_\Psi - \min\limits_j D_{ij}(\mathbf{x}) & \text{if } \min\limits_j D_{ij}(\mathbf{x}) < d_\Psi, \\ 0 & \text{otherwise,} \end{cases} \tag{3.11}$$

$$\Psi_2^j(\mathbf{x}) = \begin{cases} d_\Psi - \min\limits_i D_{ij}(\mathbf{x}) & \text{if } \min\limits_i D_{ij}(\mathbf{x}) < d_\Psi, \\ 0 & \text{otherwise,} \end{cases} \tag{3.12}$$

form the target function for the domain-overlapping constraints:

$$\chi_\Psi^2(\mathbf{x}) = \sum_i [\Psi_1^i(\mathbf{x})]^2 + \sum_j [\Psi_2^j(\mathbf{x})]^2. \tag{3.13}$$

We now combine the alignment tensor, CSP-like, and domain-overlapping constraints into one energy function

$$\chi_F^2(\mathbf{x}) = \kappa\chi^2(\mathbf{x}) + \chi_\Omega^2(\mathbf{x}) + 100\chi_\Psi^2(\mathbf{x}). \tag{3.14}$$

In our experiments, $d_\Omega = 4$Å, $d_\Psi = 0.9$Å, $d_\Omega^{cut} = 10$Å. The weight of 100 for χ_Ψ^2 was chosen as just a very large value that would penalize even minimal overlap significantly more than any violation of a CSP-like interaction. We set the value of κ in Section 3.3.7.

We reformulate equation (3.2) to use χ_F^2 instead of χ^2, and solve this problem to improve the minimizer from PATIDOCK. We refer to this method as *PATIDOCK+*. The new target function cannot be solved using local minimization. Therefore, we use a branch and bound method [42] to deterministically solve equation (3.14) for the global minimizer.

3.3 Results and Discussion

In order to examine the feasibility of molecular docking guided by RDCs, we applied PATIDOCK-t, PATIDOCK, and PATIDOCK+ to several protein systems. Potential sources of inaccuracy in our docking approach are errors in the experimental data (RDCs) and the inaccuracy in the barrier model prediction of molecular alignment. To separate and quantify these errors we tested our method on two distinct datasets as well as two protein-protein systems. The first dataset, which we refer to as COMPLEX, is a set of 84 protein-protein complexes described in Mintseris et al. [48]. This dataset provides a wide variety of interprotein contacts

and molecular shapes, but it contains no experimental RDC data. We used this dataset to generate synthetic RDC data and examine the validity of our docking method and its sensitivity to common measurement errors due to experimental imprecision. This allowed us to test our method under "ideal experimental conditions", i.e. when the simple barrier model (see Section 2.2.1) is an adequate physical model for molecular alignment, and the only errors in the data originate from (random) experimental noise in the measurements.

The second dataset, which we refer to as SINGLE, consists of 7 monomeric proteins for which experimental RDC data (in bicelles- or PEG/hexanol-based media) are available in the BMRB database [72]. We utilized this dataset to test PATI predictions in Chapter 2. These experimental RDC data are used here to gauge the accuracy of our docking method under real experimental conditions and the inaccuracies inherent to the barrier model's prediction of the alignment tensor. Similar to the COMPLEX dataset we also generated synthetic RDC data for this set of proteins, as a control. Since these are single-domain proteins, to use this dataset for testing docking, we artificially created a molecular "complex" using a plane to arbitrarily bisect each protein molecule into two distinct sets of atoms. See Figure 3.1A and Figure 3.1B for an illustration of how Cyanovirin-N is cut into two domains by a plane.

Finally we applied our method to two protein-protein systems for which we have experimental RDC and CSP data: ubiquitin/UBA complex [85] (PDB code 2JY6) and lysine-48-linked di-Ubiqutin [73] (PDB code 2BGF). These complexes allow us to present a "real world" practical application for PATIDOCK. We show

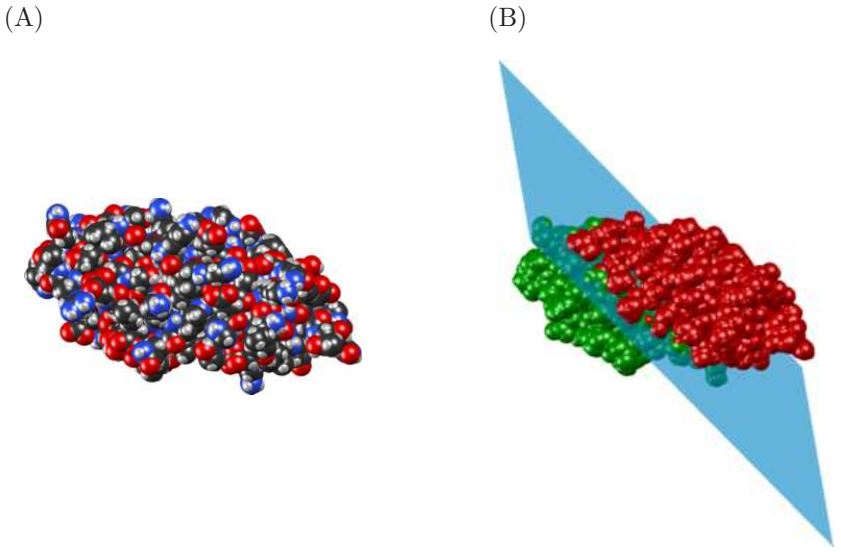

Figure 3.1: Illustration of the bisection of Cyanovirin-N (PDB code 2EZM). (A) Van der Waals surface of Cyanovirin-N. (B) Illustration of how the protein is split into two domains with approximately equal number of atoms by a plane. The first domain is colored green, the second domain is red.

that it is possible to quickly get a good solution for a complex using only the alignment tensor. In addition we show that combining our method with a more complicated energy function that accounts for additional factors such as van der Waals interactions and CSPs can yield an accurate solution in practice.

We implemented PATIDOCK in MATLAB 7.8.0 and performed all calculations and timing on a single 1.7 GHz Pentium M processor with 1.5 GB RAM, running Windows XP. The set of $[\alpha, u]$ values for which we precompute $\mathbf{F}$, η_1, η_2, and $\mathbf{R}$ was determined by the adaptive numerical integration of equation (2.15) with an absolute error of 0.05 (using MATLAB's *quad* function, see e.g., [76]). The latter value was determined empirically based on the highest tolerance value which still gave docking solutions accurate to within 0.2Å for synthetic RDCs for all complexes

in the COMPLEX (excluding one outlier) and SINGLE datasets. Note that the more accurate numerical integration is, the more $[\alpha, u]$ values are needed to compute the integral, hence the slower is the overall docking process. Precomputing $\mathbf{F}$, η_1, η_2, and $\mathbf{R}$ functions for the specific $[\alpha, u]$ set allows us to quickly recompute the integrals for different translations of the second domain without having to reevaluate these computationally expensive functions.

Due to the four-fold ambiguity of the relative orientation of domain S_2 with respect to S_1 and the existence of multiple local minimizers (with regard to translation) for each orientation, we expect to have at least eight potential solutions. The solutions can be ranked by the RMSD between the experimental structure of S_2 and the predicted one, where the atom positions in S_2 are adjusted by $\mathbf{R}^*$ and $\mathbf{x}^*$ (recall that S_1 is fixed in space). Since $\mathbf{R}^*$ can be directly computed from the experimental RDC data independent of our model, we first focus our analysis on the minimizers that come from the correct orientation of the two domains. We then present the results for the complete docking method that also includes automatic alignment of the two domains, in addition to their positioning relative to each other.

3.3.1 Docking Using Ideal Synthetic Data

In order to demonstrate the feasibility of structural assembly of molecular complexes based solely on RDC data, we first applied PATIDOCK-t to synthetic data generated for proteins from the COMPLEX and SINGLE datasets.

To test our ability to find the correct minimizer under ideal conditions, for

each complex we generated a *synthetic alignment tensor* $\tilde{\mathbf{A}}_{syn}$ using PATI prediction. From this and the three-dimensional structure of the complex we calculated RDCs for all amide NH bonds, which we call *synthetic RDCs*, assuming that there is no noise in experimental measurements. The synthetic RDCs along with the three-dimensional structures of the two domains comprise the input to our minimization algorithm. We will rate our results based on the "Best Displacement", the smallest Euclidean norm between all the computed translations and the known correct translation. The results for PATIDOCK-t, using $\tilde{\mathbf{A}}_{syn}$ as the experimental alignment tensor, are presented in Table 3.1 (columns "0 Hz", "Time(s)", and "#Sol.") for the SINGLE dataset. The results for the COMPLEX dataset under ideal conditions (labeled "0 Hz" in Figure 3.2) are very similar (also see Supporting Information). These results clearly demonstrate that it is possible, under ideal conditions, to accurately and efficiently assemble molecular complexes based solely on RDC data.

3.3.2 Robustness of RDC-Guided Docking to Experimental Noise

In reality, RDC values always contain measurement errors, which are usually below 1 Hz. To assess the effect of such errors on the RDC-guided docking we added to the synthetic RDCs normally distributed noise with standard deviation of 1 Hz or 3 Hz. This allowed us to test whether it is possible to accurately dock a complex based solely on the alignment tensor in the presence of considerable (1 Hz) or extreme (3 Hz) noise in the data. Figure 3.2 shows errors in the docking

solutions for the COMPLEX dataset in the presence or absence of random noise in the generated RDC values. Very similar results were obtained using synthetic RDC data (with noise) generated for the SINGLE dataset; see Table 3.1, columns "1 Hz" and "3 Hz".

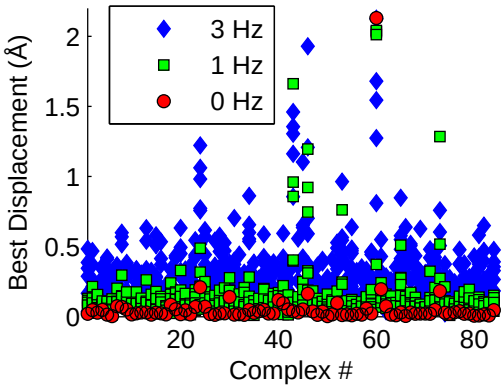

Figure 3.2: PATIDOCK-t docking results for the 84 complexes in the COMPLEX dataset using synthetic RDC values with no noise (0 Hz, red circles) or in the presence of a Gaussian noise with the standard deviation of 1 Hz (green squares) or 3 Hz (blue diamonds). In the case of noisy data, docking of each complex was performed six times, with individual RDC errors randomly selected from a normal distribution. All six results for each complex with RDC errors are plotted. For the purposes of visualization a few outliers for complexes 41, 53, and 74 that have a very small number of NH bonds are not displayed. Note that the deviation from the dataset average for some complexes is due to a small size of one of the domains relative to the other, which reduces the sensitivity of the molecular shape and the alignment tensor to interdomain translations.

From these results (Figure 3.2 and Table 3.1) we conclude that PATIDOCK-t is able to find correct docking solutions for a wide variety of proteins even under heavy (3 Hz) experimental noise. These results validate the concept of molecular docking based exclusively on the alignment tensor.

PATIDOCK-t is also extremely fast, as it takes only seconds to dock two do-

Table 3.1: The results of RDC-guided docking using PATIDOCK-t for the SINGLE dataset based on synthetic RDC data with added experimental noise.

Protein	0 Hz[a]	1 Hz[a,b]	3 Hz[a,b]	Time[c]	#[d]
B1 domain of protein G[41]	0.07 [0.07]	0.28 [0.25]	0.79 [0.78]	1.21	2
B3 domain of protein G[71]	0.09 [0.05]	0.30 [0.19]	1.25 [0.72]	1.36	2
Cyanovirin-N [10]	0.02 [0.02]	0.27 [0.16]	0.75 [0.43]	2.57	3
Gα interacting protein[21]	0.03 [0.02]	0.24 [0.13]	0.91 [0.46]	2.47	2
Ubiquitin[19]	0.02 [0.02]	0.23 [0.18]	0.67 [0.57]	1.86	2
Hen lysozyme[62]	0.05 [0.04]	0.16 [0.13]	0.53 [0.43]	1.94	2
Oxidized putidaredoxin[39]	0.06 [0.05]	0.22 [0.18]	0.62 [0.51]	1.97	2
Mean	0.05 [0.04]	0.24 [0.17]	0.79 [0.56]	1.91	2.14

[a] Best Displacement (in Å), computed as the smallest Euclidean norm between all the computed translations (solutions) and the known correct translation. The values in brackets represent the RMSD (in Hz) between the synthetic RDCs and the predicted RDCs at the solution. The column labels represent the size of the standard deviation of the normally distributed noise added to synthetic RDCs. "0 Hz" corresponds to no noise added to synthetic RDCs.

[b] The values represent an average of twelve independent runs.

[c] The average elapsed time (in seconds) required for docking the total of twenty five runs for "0 Hz", "1 Hz", and "3 Hz".

[d] The number of possible solutions, all of which have a very similar predicted alignment tensor.

mains on a slow laptop. This speed makes it feasible to perform RDC-based docking at each iteration step of a more complicated flexible docking algorithm, for example by analyzing docking of multiple conformers (models) at each minimization itera-

tion. Another potential consequence of the speed is that it opens up the possibility of extending the docking algorithm to three or more molecules. Since we are able to accurately dock molecules given perfect prediction of the alignment tensor, the accuracy of the results in practice will depend on how well we can predict the alignment tensor in an experimental setting.

3.3.3 Docking using Experimental RDC Data

Having established the ability to accurately assemble molecular complexes using synthetic data, we next test our method on the alignment tensors derived from actual experimental data, in order to understand how errors in prediction of the alignment tensor affect the overall accuracy of docking. We use for this purpose the 7 proteins of the SINGLE dataset. The alignment tensor prediction and the limitations of the barrier model for these proteins were addressed in detail in Chapter 2. Since the errors in the experimental RDC data for these proteins are about or smaller than 1 Hz, based on our results with synthetic data (Table 3.1) we expected to get a good solution provided that the barrier model is a good predictor of the alignment tensor. The results for PATIDOCK-t are shown in Table 3.2.

Surprisingly, these solutions are worse than one would expect based just on the errors in the experimental data. Given that with synthetic RDC data these proteins were docked properly (see Table 3.1) this suggests that the alignment tensor predicted using a simple barrier model differs from the actual tensor, and this discrepancy could translate into an error (about 4.3Å) in the docking solution. In

Table 3.2: The results of RDC-guided docking using PATIDOCK-t for the SINGLE dataset based on experimental RDC data.

Protein	PDB[a]	Disp.[b]	Time[c]	$RMSD_{RDC}$[d]	#[e]
B1 domain of protein G	3gb1	2.01	1.43	1.18	2
B3 domain of protein G	2oed	4.17	1.57	1.33	2
Cyanovirin-N	2ezm	5.01	1.98	3.89	2
Gα interacting protein	1cmz	6.21	1.84	1.32	2
Ubiquitin	1d3z	3.90	1.62	1.34	2
Hen lysozyme	1e8l	3.44	3.51	7.23	2
Oxidized putidaredoxin	1yjj	5.18	2.47	4.45	2
Mean		4.27	2.06	2.96	2.00

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for all calculations.

[b] Best Displacement (in Å), computed as the smallest Euclidean norm between all the computed translations (solutions) and the known correct translation.

[c] The elapsed time (in seconds) required for docking.

[d] The RMSD (in Hz) between the experimental and the predicted RDC values at the best predicted minimizer.

[e] The number of possible solutions, all of which have a very similar predicted alignment tensor.

fact, as shown in Section 2.3, the inaccuracy in alignment tensor prediction can be separated into an error in the magnitude (scaling) of the tensor and an error in its orientation. On the positive side, however, the results in Table 3.2 show that by

using only RDC data we are able to place the second domain on average within a radius of 4.3Å of its proper position.

3.3.4 Docking Using Experimental RDC Data: Combining Alignment and Translation

The docking efforts presented above focused on domain translation, while keeping interdomain orientation the same as in the original structure. We now combine our method for determining the correct translation with the method for aligning the two domains based on the orientations of the alignment tensor of the complex "reported" by each individual domain [31, 27, 36]. This is the complete method, PATIDOCK, that takes two domains with arbitrary positions and orientations, and the associated experimental RDC values, and assembles their complex automatically with no human intervention at any step.

We first align the two domains by extracting (from the experimental RDC data for the complex) the alignment tensors "seen" by each domain and using equation (3.1) to properly orient the second domain relative to the first one. We then use PATIDOCK to compute the proper translation between the now aligned domains. Due to the four-fold ambiguity in alignment we expect the number of solutions and the computation time to increase by a factor of four. The results for PATIDOCK with all potential solutions are shown in Table 3.3. Note that no domain alignment was performed in the docking shown in Table 3.2, so the values in "Disp." column are also "$RMSD_2$" values as defined in Table 3.3.

Table 3.3: The results of RDC-guided docking using PATIDOCK for the SINGLE dataset based on experimental RDC data.

Protein	RMSD[a]	$RMSD_2$[b]	Time[c]	$RMSD_{RDC}$[d]	#[e]
B1 domain of protein G	1.02	2.23	6.42	1.45	8
B3 domain of protein G	1.80	4.49	4.83	1.09	8
Cyanovirin-N	2.35	5.76	5.10	4.45	8
Gα interacting protein	2.59	6.50	6.03	1.69	8
Ubiquitin	1.83	3.93	9.09	1.68	9
Hen lysozyme	1.65	3.35	8.29	7.27	10
Oxidized putidaredoxin	2.62	5.61	8.19	4.58	9
Mean	1.98	4.55	6.85	3.17	8.57

[a] The RMSD (in Å) between the original complex structure and the predicted complex. The structures are optimally rotated and centered using the center of mass [47].

[b] The RMSD (in Å) between the coordinates of atoms of the second domain for the original and the predicted complex.

[c] The elapsed time (in seconds) required for docking of all four orientations.

[d] The RMSD (in Hz) between the experimental and the predicted RDC values at the best predicted minimizer.

[e] The number of possible solutions, all of which have a very similar predicted alignment tensor.

The increase in $RMSD_2$ values from the fixed-orientation assembly in Table 3.2 (values are in the "Disp." column) to the align-and-translate assembly in Table 3.3 is small, showing that alignment of domains by using experimental RDC values

is an extremely accurate technique and is not a significant contributor of error to structure assembly. As expected, there is a four-fold increase in the number of possible solutions and the running time, but the combined algorithm still completes in less than 10 seconds.

3.3.5 Application to a Real System: Ubiquitin/UBA Complex

We now test our method on a protein complex for which experimental RDC and CSP data are available: the complex of human ubiquitin (Ub) with the UBA domain of ubiquilin-1[85] (PDB code 2JY6). Using the experimental CSP data we defined as CSP-active residues L8, T9, G10, K48, E51, R54, Q62, H68, L71, and L73 in Ub, and M557, G558, L560, I570, A571, N577, E581, R582, L584 in UBA. See Figure 3.3 for the mapping of the CSP-active residues onto the Ub/UBA complex. In this section we will only use the RDC data, while the CSP data will be included in a later section.

A potential complication for the rigid-body docking approach arises in the case of the Ub/UBA complex from the fact that both proteins have extended unstructured and highly flexible tails. In fact, residues 73-76 in Ub and 536-544 in the UBA construct used in the experimental study experience large-amplitude motions [85] on a ps-ns time scale, which is many orders of magnitude faster than the time scale (˜100 ms) of a NMR experiment. These motions are also present in the Ub/UBA complex, reflecting the fact that the tails do not participate in the binding [85]. Naturally, such tails present a significant challenge for shape-sensitive compu-

tations like those in the current study, because no single structure can represent the ensemble/motion-averaged molecular shape relevant for a particular experiment. This raises important questions that have not been addressed in the literature so far: could flexible tails simply be neglected (clipped off) in such calculations or should they be represented by a structural ensemble, and how large does the latter need to be? In order to address these questions, we performed docking for both the structural ensembles and the clipped (tailless) structures. Because the RDC data were measured in the PEG/hexanol medium [78], the actual inter-barrier distance was unknown and had to be estimated. We set $h = 400$Å, a value that gives the correct scaling between the predicted and experimentally determined alignment tensor at the known solution.

To sample various orientations of the tails (not present in the original PDB structure of the complex), we extracted 10 representative orientations of Ub's C-terminus from the NMR ensemble of Ub monomer (PDB code 1D3Z [19]) and 10 possible orientations of the N-terminus of the UBA domain from its NMR ensemble in the monomeric state (PDB code 2JY5[85]). These conformations of the tails were superimposed onto the corresponding domains in the complex structure (2JY6), thus creating an ensemble of 100 possible models for the Ub/UBA complex (shown in Figure 3.3). We refer to this Ub/UBA complex as *Structure 2jy6-I*. From the 100 models of Structure 2jy6-I we were able to estimate the variance in the docking solutions that the two tails introduce. The results are presented in Table 3.4.

Because averaging by fast reorientations of the tails is expected to diminish the tails' effect on the alignment tensor, we clipped off the two tails from the struc-

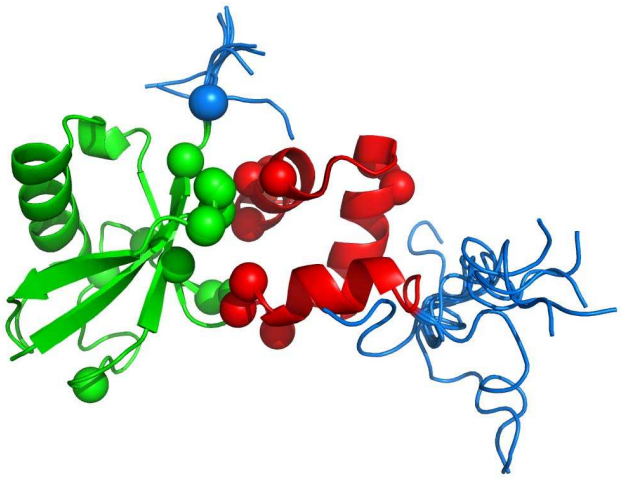

Figure 3.3: A cartoon representation of the ensemble of 100 possible models for the Ub/UBA complex (Structure 2jy6-I). Ub is colored green, UBA is in red, the flexible tails are colored blue, and the CSP-active residues are represented by spheres around their C_α atoms.

tures of the corresponding proteins and then docked the two tailless molecules using PATIDOCK-t and PATIDOCK. We refer to the tailless Ub/UBA complex as *Structure 2jy6-II*; the results are presented in Table 3.4. Figure 3.4 shows the isosurface plot of the energy function χ^2 for the tailless Ub/UBA complex and the visualization of the two solutions from PATIDOCK-t. The isosurface plot clearly demonstrates that there are two distinct minima in the energy function, both of which were found by our program. As can be seen from Figure 3.4C and Figure 3.4D, the reason for the two minima is that both solutions have very similar convex hulls due to the geometric symmetry inherent in the problem.

As evident from Table 3.4, the conformation(s) of the tail can have a profound effect on the results of docking. The solution varies on average by 2Å over all the possible combinations of tail orientations, whereas removing the tails improves the results significantly. This suggests that a potential solution for dealing with

Table 3.4: The results of docking the Ubiquitin/UBA complex using PATIDOCK-t and PATIDOCK.

Struct.[a]	Method[b]	RMSD[c]	$RMSD_2$[d]	Time	$RMSD_{RDC}$[e]	#[f]
2jy6-I	PATIDOCK-t	3.37[g] (1.05)[h]	8.72[g] (1.97)[h]	2.02[g] (0.44)[h]	4.33[g] (1.16)[h]	2.01[g]
2jy6-I	PATIDOCK	3.43[g] (1.07)[h]	8.71[g] (2.03)[h]	6.00[g] (1.19)[h]	4.42[g] (1.17)[h]	8.31[g]
2jy6-II	PATIDOCK-t	1.32	4.23	1.87	4.13	2
2jy6-II	PATIDOCK	1.25	3.72	4.95	4.23	8

[a] 2jy6-I is the ensemble of 100 structures representing various conformations of Ub and UBA tails (see text), whereas in 2jy6-II the tails were clipped off.

[b] The method that was used to dock the complex.

[c] The RMSD (in Å) between the original complex structure and the predicted complex. The structures are optimally rotated and centered using the center of mass [47].

[d] The RMSD (in Å) between the coordinates of atoms of the second domain for the original and predicted complex.

[e] The RMSD (in Hz) between the experimental and the predicted RDC values at the best predicted minimizer.

[f] The number of possible solutions, all of which have a very similar overall alignment tensor.

[g] Values are the means of the individual values for the best solution of each of the 100 models.

[h] Values in the parentheses are the standard deviations of the individual values for the best solution of each of the 100 models.

flexible tails in RDC-guided docking is to clip them off rather than using a specific conformation or trying to deduce the "averaged" conformation of the tail. Without the tails, using PATIDOCK, we get an $RMSD_2$ of about 3.7Å, which is somewhat smaller than but close to the expected value of 4.5Å (see above).

3.3.6 Application to a Real Dual-Domain System: Lys48-linked di-Ubiquitin

Finally, we tested our method on a dual-domain system for which both experimental RDC and CSP data are available: the Lys48-linked di-Ubiquitin [78, 36, 73] (PDB code 2BGF). Using the experimental CSP data we define hydrophobic-patch residues L8, I44, and L70 on both of the domains to be CSP-active. See Figure 3.5 for the mapping of the CSP-active residues onto the di-Ubiquitin (Ub_2) structure. The CSP data will be used in Section 3.3.7. Because the RDC data were measured in the PEG/hexanol medium [78], the actual inter-barrier distance was unknown and had to be estimated. We set $h = 550$Å, a value that gives the correct scaling between the predicted and experimentally determined alignment tensor at the known solution.

As in the case of the Ub/UBA complex, a potential complication for the rigid-body docking approach arises from the unstructured and highly flexible C-terminal tails comprising residues 73-76 of each domain [36], though the tail in Ubiquitin is much shorter than that of UBA. We therefore performed a similar analysis to that in the previous section. However, instead of superimposing the tails onto the Ub_2 complex, we simply took the ensemble of the 10 models from the Ub_2 structure 2BGF (shown in Figure 3.3). We refer to this ensemble as *Structure 2bgf-I*. Similarly, we created *Structure 2bgf-II* by taking the first model in 2BGF and clipping off residues 73-76 of both domains. The results for the ensemble and the clipped (tailless) structures are presented in Table 3.5.

Table 3.5: The results of docking Lys48-linked di-Ubiquitin using PATIDOCK-t and PATIDOCK.

Struct.[a]	Method[b]	RMSD[c]	$RMSD_2$[d]	Time	$RMSD_{RDC}$[e]	#[f]
2bgf-I	PATIDOCK-t	1.35[g] (0.35)[h]	3.96[g] (1.10)[h]	2.35[g] (0.44)[h]	3.56[g] (0.37)[h]	2.00[g]
2bgf-I	PATIDOCK	1.49[g] (0.30)[h]	4.33[g] (0.67)[h]	7.05[g] (0.68)[h]	3.48[g] (0.34)[h]	8.10[g]
2bgf-II	PATIDOCK-t	1.07	3.53	2.80	3.45	2
2bgf-II	PATIDOCK	1.19	3.69	6.56	3.45	8

[a] 2bgf-I is the ensemble of 10 structures representing various conformations of the C-terminal tails of both Ubiquitin domains (see text), whereas in 2bgf-II the tails were clipped off.

[b] The method that was used to dock the complex.

[c] The RMSD (in Å) between the original complex structure and the predicted complex. The structures are optimally rotated and centered using the center of mass [47].

[d] The RMSD (in Å) between the coordinates of atoms of the second domain for the original and the predicted complex.

[e] The RMSD (in Hz) between the experimental and the predicted RDC values at the best predicted minimizer.

[f] The number of possible solutions, all of which have a very similar overall alignment tensor.

[g] Values are the means of the individual values for the best solution of each of the 10 models.

[h] Values in the parentheses are the standard deviations of the individual values for the best solution of each of the 10 models.

As above, the conformation of the tail has noticeable effect on the results of docking, although significantly less than in the Ub/UBA complex. The solution varies on average by 1Å among all the possible tails' conformations, and removing the tails improves the results slightly. These results further support the conclusion

that the potential solution for dealing with flexible tails in RDC-guided docking is to clip off the tails. Without the tails, using PATIDOCK, we get an $RMSD_2$ of 3.7Å, which is somewhat smaller than but close to the expected value of 4.5Å (see above).

3.3.7 Docking Using Experimental RDC Data Combined with Ambiguous Interface-Related Restraints

The results in previous sections using real experimental data give a good hint at the errors that one can expect when using the barrier model as the alignment tensor predictor. Thus, we expect that in practice the error in domain positioning using PATIDOCK would be about 4.3Å. The fact that the results are a relatively short distance from the actual solution demonstrates that the alignment-tensor-based χ^2 is a useful constraint.

We now seek to improve upon the previous results by combining CSP-like constraints along with the alignment tensor constraints by minimizing χ^2_F (see equation (3.14)). The combination of constraints should lead to a better and more reliable overall solution.

To properly set κ we analyzed at the known solution the values of the three target functions that make up χ^2_F. We seek a value of κ that will weigh the errors from the CSP-like constraints and the alignment tensor constraint equally at the known solution. The errors are presented in Table 3.6.

We took the ratio χ^2_Ω/χ^2 for 2bgf-II (1.23×10^5) as the value of κ for the target

Table 3.6: The values of the energy functions at the known solution.

Structure[a]	χ^2_Ω	χ^2_Ψ	χ^2	χ^2_Ω/χ^2
3gb1	0	0	9.01×10^{-8}	N/A
2oed	0	0	1.72×10^{-7}	N/A
2ezm	0	0	4.78×10^{-7}	N/A
1cmz	0	0	1.50×10^{-7}	N/A
1d3z	0	0	2.08×10^{-7}	N/A
1e8l	0	0	7.35×10^{-7}	N/A
1yjj	0	0	6.40×10^{-7}	N/A
2jy6-II	1.78×10^{-1}	0	4.46×10^{-7}	4.00×10^{5}
2bgf-II	4.46×10^{-2}	0	3.64×10^{-7}	1.23×10^{5}

[a] See Results section in the main text for structure references.

function χ^2_F. We believe that the value for 2bgf-II is the best estimate we have for κ because of the large number of outliers in the list of CSP-active residues for 2jy6-II.

The results of applying PATIDOCK+ to the SINGLE dataset, Ub/UBA, and Ub_2 are presented in Table 3.7. Note that we are now able to select the correct structure out of all possible solutions by picking the one with the lowest χ^2_F value. The cartoon representations of the solutions for the two protein-protein systems are presented in Figure 3.6.

As evident from Table 3.7, the addition of ambiguous, CSP-like restraints significantly improved the solution for all proteins, compared to the results in Table

Table 3.7: The results for PATIDOCK+ using a combination of CSP-like and alignment tensor constraints.

Protein	Structure[a]	RMSD[b]	$RMSD_2$[c]	$RMSD_{RDC}$[d]	#Sol.[e]
B1 domain of protein G	3gb1	0.92	1.93	1.44	1
B3 domain of protein G	2oed	1.23	3.29	1.55	1
Cyanovirin-N	2ezm	1.52	3.94	3.92	1
Gα interacting protein	1cmz	1.20	3.53	2.67	1
Ubiquitin	1d3z	1.01	2.45	2.33	1
Hen lysozyme	1e8l	0.91	1.96	7.41	1
Oxidized putidaredoxin	1yjj	1.36	3.18	4.35	1
Ubiquitin/UBA	2jy6-II	0.56	1.37	5.00	1
di-Ubiquitin	2bgf-II	0.77	1.72	4.30	1
Mean		1.05	2.60	3.66	1.00

[a] See previous tables and Results section for structure references.

[b] The RMSD (in Å) between the original complex structure and the predicted complex. The structures are optimally rotated and centered using the center of mass [47].

[c] The RMSD (in Å) between the coordinates of atoms of the second domain for the original and predicted complex.

[d] The RMSD (in Hz) between the experimental and the predicted RDC values at the best predicted minimizer.

[e] The number of possible solutions, all of which have a very similar χ^2_F.

3.3, Table 3.4, and Table 3.5. The docked solutions for the two "real" complexes (Ub/UBA and Ub_2) based entirely on experimental RDC and CSP data have $RMSD_2$ below 2Å. This indicates that combining RDCs with other experimental intermolecular constraints in a real situation could be a powerful method for quickly yielding good docking solutions. The additional benefit of using CSP-like restraints is that we now are able to correctly identify the best solution from the eight or more possible symmetry-related solutions based just on the χ^2_F values.

3.4 Conclusion

In this chapter we demonstrated that it is fundamentally possible to assemble a protein-protein complex based solely on experimental RDC data and the prediction of the alignment tensor from three-dimensional structures, provided that the structures of the individual components are available. The PATIDOCK method described here is robust with respect to large experimental errors in RDC data. Accuracy can be increased, at the expense of time, by changing the tolerance to the numerical integration routine in PATI. However, the improvement in accuracy is limited by the inherent inability of the barrier model to fully model the physical conditions. When applied to real experimental data, it gives on average a 4Å error in the relative positioning of the molecules. We determined that the resulting structure could be further refined by including other available experimental data (PATIDOCK+). Moreover, the presence of extended unstructured/flexible parts (e.g. tails) in a molecule can potentially affect the solution by more than 2Å,

depending on which structure/conformation of such parts is chosen. We propose removal of the flexible tails as a potential solution to this problem.

The PATIDOCK methods are extremely fast, and therefore we do not foresee a need for a faster, but less accurate, method for prediction of the alignment tensor than PATI. Potential improvements in the prediction of the alignment tensor will most likely involve (i) representing individual molecular components as structural ensembles rather than single structures and (ii) using a weight function inside the integrals in equation (2.15), to account for possible non-steric interactions with the aligning medium. For example, such a function could weigh η differently, or introduce charge potentials in case of non-neutral alignment media (see e.g., [86]). We foresee such an addition as being easily adapted into our docking method.

The PATIDOCK approach presented in this chapter can potentially be used in several ways. First, it provides a quick rigid-body docking method whose solutions can be utilized to significantly limit the search space of a more complicated flexible-docking algorithm. For example, we know that using PATIDOCK we are able to place the second domain to within 10Å or less of its actual position. We can then constrain the search of a flexible-docking algorithm (e.g., HADDOCK [26], XPLOR-NIH [63]) to within that radius.

Second, our energy functions can be included as an additional term into a more general energy function that utilizes more complicated constraints such as geometric/structural restraints, electrostatic and van der Waals potentials, etc. We have partially done this in this chapter by combining alignment tensor with CSP-like constraints. In a similar manner our energy function can be combined with a more

complicated energy function in HADDOCK and other programs.

Third, PATIDOCK can be used as the main method for driving molecular docking in the situation where there is a lack of unambiguous inter-domain structural information, like NOEs. This last application will become more practical as methods for prediction of the alignment tensor improve. The computational efficiency of our approach makes it feasible to perform RDC-based docking at each iteration step of a more complicated flexible docking algorithm, for example by analyzing docking of multiple conformers at each minimization iteration. Note also that the energy function designed here could potentially be used to evaluate and refine protein structures, including those for single-domain proteins, based on how well the 3D shape of the molecule agrees with experimental RDC data.

The fact that our docking method is extremely fast for two-domain complexes opens up the possibility of extending the PATIDOCK approach to three or more domains. Even though each additional domain gives rise to an exponential increase in complexity and time, it is still possible to quickly evaluate our energy function for several domains.

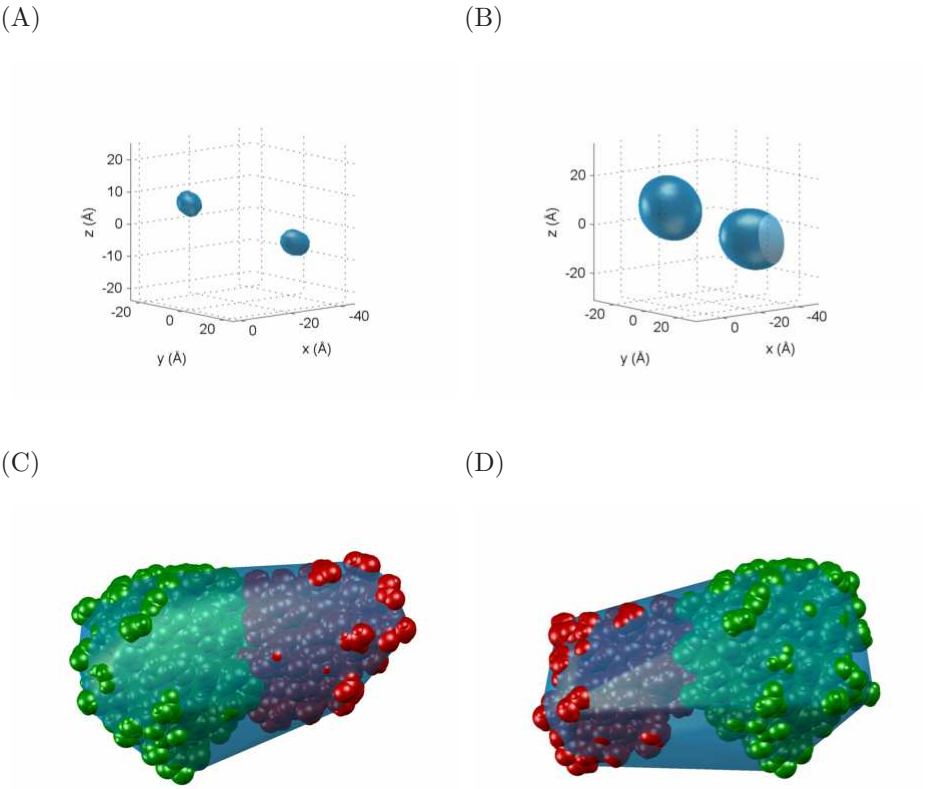

Figure 3.4: The results of RDC-guided docking for the tailless Ub/UBA complex (2jy6-II) using PATIDOCK-t. Shown are (A-B) isosurface plots of the $\chi^2(\mathbf{x})$ function and (C-D) the associated van der Waals surfaces (wrapped by their convex hulls) of the two solutions corresponding to the two local minima of $\chi^2(\mathbf{x})$. The isosurfaces correspond to (A) $\min_{\mathbf{x}} \chi^2(\mathbf{x}) + 0.1\sigma$ and (B) $\min_{\mathbf{x}} \chi^2(\mathbf{x}) + 0.6\sigma$, for all $\mathbf{x}$ inside the grid, where σ is the standard deviation of the values of χ^2 in the grid. The isosurface data were collected on a $100 \times 100 \times 100$ Å grid around $\mathbf{0}$. (C) The best (closest) solution with the UBA domain positioned to the right of Ub, with $\chi^2 = 2.01 \times 10^{-7}$ at the solution. (D) The incorrect solution where the UBA domain is to the left of Ub, with $\chi^2 = 1.24 \times 10^{-7}$ at the solution. In these van der Waals surface plots Ub is colored green and UBA is red. Both solutions have a very similar convex hull, hence similar predicted alignment tensor. The camera angle relative to Ub's orientation is the same in both figures. Note that the best solution has a higher χ^2 value.

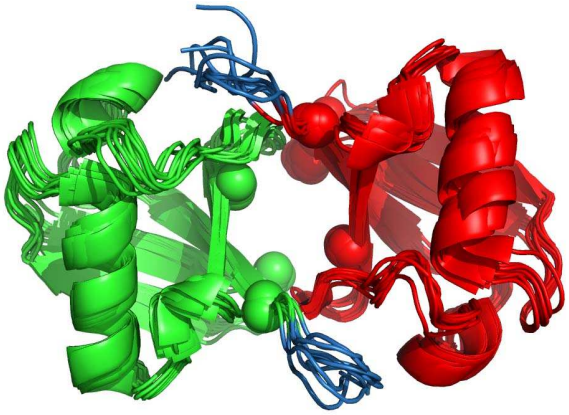

Figure 3.5: A cartoon representation of the ensemble of 10 models for the di-Ubiquitin complex (Structure 2bgf-I). Proximal domain is colored green, distal domain is in red, the flexible tails are colored blue, and the CSP-active residues are represented by spheres around their C_α atoms.

(A) (B)

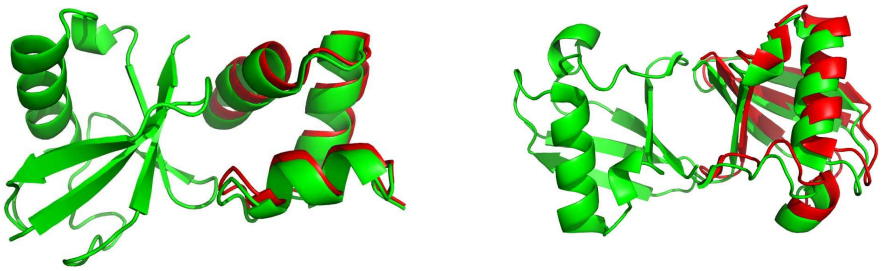

Figure 3.6: A cartoon representation of the actual structure (green) vs. the docked structure (red) for the (A) Ub/UBA complex and (B) Ub_2 molecule based on minimization of χ^2_F. Only the adjusted domain (S_2) is shown for the docked structures, the other domain (S_1) superimposes exactly with the corresponding domain in the actual structure.

Chapter 4

Docking Based on the Diffusion Tensor (ELMDOCK)

In this Chapter we present and evaluate ELMDOCK, a rigid molecular docking method for a two-domain complex. ELMDOCK relies solely on the three-dimensional structure of the individual components and the experimentally derived diffusion tensor that is derived directly from NMR relaxation data. We show that, given an accurate *ab initio* predictor of the diffusion tensor from protein structure, it is possible to accurately assemble a protein-protein complex by leveraging the NMR relaxation data's sensitivity to molecular shape in our docking method. The proposed docking method is robust against experimental errors in the NMR relaxation data and is computationally efficient. We analyze the accuracy and efficiency of this method using synthetic data for a large variety of protein-protein complexes as well as actual experimental data for three protein systems for which the structure of the complex and diffusion data is available. Additionally, we analyze the effect of flexible unstructured tails on the outcome of docking for a complex of ubiquitin and a ubiquitin-associated domain. The results demonstrate that it is possible to quickly assemble a protein-protein complex based solely on experimental NMR relaxation data for a wide variety of complexes.

4.1 Introduction

In Chapter 3 we introduced PATIDOCK, a method for using the alignment tensor as a global constraint for rigid docking of multi-domain proteins. However, PATIDOCK is limited by the physical ability to study a molecule of interest in a solution filled with an alignment medium for which we are able to predict an alignment tensor (bicelles, PEG/hexanol, and other alignment media that can be modeled by steric restrictions introduced by a planar barrier). For a variety of molecules, measuring a molecule in that type of medium might not be physically possible.

In this chapter we introduce ELMDOCK, a rigid docking method that is analogous to PATIDOCK, but that uses the diffusion tensor [13] instead of the alignment tensor for docking. ELMDOCK is named for the *ELlipsoidal Model* it uses to approximate the shape of a domain. ELMDOCK has an advantage over PATIDOCK in that it does not require any alignment medium, and therefore can be applied to a larger variety of complexes. ELMDOCK utilizes the sensitivity of the rotational diffusion tensor to molecular shape to dock a two-domain complex based solely on the three-dimensional structure of each domain and the experimental diffusion tensor of the complex. This idea of using the diffusion tensor as a primary guide for rigid docking of multi-domain proteins was introduced in Ryabov et al. [57] and further explored in Ryabov et al. [59]. Similar to PATIDOCK, ELMDOCK uses the difference between the experimental and the predicted diffusion tensors to find the proper positioning of the second domain of the complex relative to the first.

Docking using the diffusion tensor requires three components. The first component is a method that will determine the experimental diffusion tensor from the NMR relaxation data. The equivalent method for the alignment tensor was simple: a linear least squares problem involving the RDCs (see kbequation (2.31)). In the case of the diffusion tensor the method is significantly more complex. We describe such a method, called ROTDIF [33, 37, 81], for the computation of the experimental diffusion tensor from the NMR relaxation data in Section 4.2. We also introduce an improvement to the algorithm for ROTDIF which results in an order of magnitude speedup of the method.

The second required component for ELMDOCK is a method for predicting the diffusion tensor given a three-dimensional structure of a molecule. The two known methods are HYDRONMR [15, 23] and ELM [58]. We present both of these methods in Section 4.3, but will use only ELM in ELMDOCK.

The final required component for ELMDOCK is a method that will efficiently find the optimal positioning of the second domain relative to the first based on the difference between the experimental diffusion tensor (computed by ROTDIF) and the predicted diffusion tensor (computed by ELM). We present this docking method in Section 4.4.

Having fully described all components of ELMDOCK in Sections 4.2, 4.3, and 4.4, in Section 4.5 we demonstrate that ELMDOCK can deterministically and efficiently perform rigid-body docking based on the diffusion tensor, analyze the robustness of ELMDOCK under certain types of experimental errors, and examine its performance in applications to real experimental data.

4.2 Computing Experimental Diffusion Tensor (ROTDIF)

In this section we present ROTDIF [33, 37, 81], a method for determining the experimental diffusion tensor $\mathbf{D}_{exp}$ from NMR relaxation data.

The (rotational) diffusion tensor $\mathbf{D}$ is a symmetric positive definite 3×3 matrix that represents the anisotropic overall tumbling of a molecule [13]. Tumbling refers to the random reorientation of a molecule around its axes in a solvent. Anisotropy refers to the case when the tumbling rates around each axis are different. We label the sorted eigenvalues of $\mathbf{D}$ as $D_x \leq D_y \leq D_z$.

We can visualize the diffusion tensor as a set of three orthogonal vectors oriented in the molecule's coordinate space. The orientation of the vectors (the eigenvectors of the diffusion tensor) describes the axes around which the molecule is rotating. The length of each vector (the eigenvalues of the diffusion tensor) describes the rate of rotation around the associated axis. See Section 1.3.1 for mathematical properties of the matrix $\mathbf{D}$.

To compute the experimental diffusion tensor from the experimentally measured NMR relaxation data, nonlinear least squares is used to find the diffusion tensor that minimizes the difference between the experimentally measured NMR relaxation data and the NMR relaxation data predicted by a physical model. The physical model predicts the NMR relaxation data given a diffusion tensor and a set of unit vectors of the complex's NH bonds (normalized vector between the positions of the N and H atoms). The algorithm for finding the experimental diffusion tensor

can be expressed as

$$\mathbf{D}_{exp} = \arg\min_{\mathbf{D}} \chi_R^2(\mathbf{v}, \mathbf{D}), \tag{4.1}$$

where

$$\chi_R^2(\mathbf{v}, \mathbf{D}) = \sum_{i=1}^{n} \left[\rho_i^{exp} - \rho_i^{pred}(\mathbf{v}^i, \mathbf{D})\right]^2, \tag{4.2}$$

n is the number of NH bonds in the molecule, $\mathbf{v}^i = [v_x^i, v_y^i, v_z^i]$ is the unit vector between the N and H atoms in the i-th NH bond, ρ_i^{exp} is the ratio of experimentally measured transverse and longitudinal relaxation rates for bond i, and $\rho_i^{pred}(\mathbf{v}_i, \mathbf{D})$ is the predicted ratio of transverse and longitudinal relaxation rates for bond i.

Given the longitudinal relaxation rate r_{1i}, the transverse relaxation rate r_{2i}, and the steady state NOE r_{3i} for ith NH bond, the experimental ratio ρ_i^{exp} (adjusted for high frequency components) was derived in [35, 32] and is computed as

$$\rho_i^{exp} = \frac{4r'_{1i}}{6r_{2i} - 3r_{1i} - 13.624H_i}, \tag{4.3}$$

where

$$r'_{1i} = r_{1i} - 6.246H_i, \tag{4.4}$$

$$H_i = -r_{1i}\frac{\gamma_N}{5\gamma_H}(1 - r_{3i}), \tag{4.5}$$

and γ_N,γ_H are the gyromagnetic ratios of N and H. See Cavanagh et al. [16] for the values of the gyromagnetic ratios.

To compute $\rho_i^{pred}(\mathbf{v}_i, \mathbf{D})$ three different physical models can be used depending on how similar we expect the eigenvalues of the experimental diffusion tensor to be. In Section 4.2.1 we present the three diffusion tensor models for the cases when none, two, or all of the eigenvalues are equal. Then, in Section 4.2.2 we

present the associated algorithms that deterministically solve all three models for the experimental diffusion tensor, and are faster than the algorithms proposed in Walker et al. [81].

4.2.1 Experimental Diffusion Tensor Models

There are three diffusion tensor models that can be used to model ρ^{pred}. The most general (and the most complicated) diffusion tensor model is the *fully anisotropic* model, where all three eigenvalues of the experimental diffusion tensor are assumed to be different. We describe this model in Section 4.2.1.1. In the case when two eigenvalues of the experimental diffusion tensor are assumed to be equal, we can simplify the fully anisotropic model to an *axially symmetric* model, which we describe in Section 4.2.1.2. Finally, in the simplest case, when all three eigenvalues are assumed to be equal, a simple *isotropic* model is used, and is presented in Section 4.2.1.3.

4.2.1.1 Fully Anisotropic Diffusion Tensor Model

We start with the most general, fully anisotropic, diffusion tensor model, when all three eigenvalues of the experimental diffusion tensor are assumed to be different. The eigenvalues do not have to be sorted. Then for the i-th bond, ρ_i^{pred} for the fully anisotropic diffusion model, derived in Woessner [84] and slightly reformulated in Ghose et al. [37], is computed as

$$\rho_i^{pred}(\mathbf{v}^i, \mathbf{D}) = \frac{J(\mathbf{v}^i, \omega_N, \mathbf{D})}{J(\mathbf{v}^i, 0, \mathbf{D})}, \tag{4.6}$$

where ω_N is the resonance frequency of the ^{15}N spin (which is dependent on the spectrometer that is used for the experiment),

$$J(\mathbf{v}^i, \omega, \mathbf{D}) = \frac{2}{5} \sum_{k=1}^{5} \frac{d_k(\mathbf{D}) a_k(\mathbf{v}^i, \mathbf{D})}{d_k^2(\mathbf{D}) + \omega^2}, \tag{4.7}$$

the components independent of the NH bonds are

$$\begin{aligned}
d_1(\mathbf{D}) &= 4D_x + D_y + D_z, \\
d_2(\mathbf{D}) &= D_x + 4D_y + D_z, \\
d_3(\mathbf{D}) &= D_x + D_y + 4D_z, \\
d_4(\mathbf{D}) &= 6e_3 + 2e_4, \\
d_5(\mathbf{D}) &= 6e_3 - 2e_4, \\
e_1 &= D_y - D_x, \\
e_2 &= D_z - D_x, \\
e_3 &= (D_x + D_y + D_z)/3, \\
e_4 &= \sqrt{e_1^2 - e_1 e_2 + e_2^2},
\end{aligned} \tag{4.8}$$

the components dependent on the NH bonds are

$$
\begin{aligned}
a_1(\mathbf{v}^i, \mathbf{D}) &= 3\bar{v}_2^2\bar{v}_3^2, \\
a_2(\mathbf{v}^i, \mathbf{D}) &= 3\bar{v}_1^2\bar{v}_3^2, \\
a_3(\mathbf{v}^i, \mathbf{D}) &= 3\bar{v}_1^2\bar{v}_2^2, \\
a_4(\mathbf{v}^i, \mathbf{D}) &= p_1 - p_2, \\
a_5(\mathbf{v}^i, \mathbf{D}) &= p_1 + p_2, \\
p_1 &= \frac{1}{4}[3(\bar{v}_1^4 + \bar{v}_2^4 + \bar{v}_3^4) - 1], \\
p_2 &= \frac{1}{12}[\delta_1(3\bar{v}_1^4 + 2a_1 - 1) + \delta_2(3\bar{v}_2^4 + 2a_2 - 1) + \delta_3(3\bar{v}_3^4 + 2a_3 - 1)], \\
\bar{\mathbf{v}} &= \mathbf{V}^T\mathbf{v}^i,
\end{aligned}
\tag{4.9}
$$

and the shared components are

$$
\begin{aligned}
\delta_1 &= (-e_1 - e_2)/e_4, \\
\delta_2 &= (2e_1 - e_2)/e_4, \\
\delta_3 &= (2e_2 - e_1)/e_4,
\end{aligned}
\tag{4.10}
$$

Note that we reformulated how d_4 and d_5 are calculated in Ghose et al. [37] to increase numerical stability.

4.2.1.2 Axially Symmetric Diffusion Tensor Model

If two eigenvalues of $\mathbf{D}_{exp}$ are assumed to be equal, then an axially symmetric diffusion tensor model can be used for the computation of $\mathbf{D}_{exp}$. We label the two equal eigenvalues as $D_\perp$, and the unique eigenvalue as $D_\parallel$. The expression ρ_i^{exp} can be simplified greatly from the case of fully anisotropic diffusion model and is given in [84, 37].

Without loss of generality, we simplify the fully anisotropic model for the case when $D_\perp = D_x = D_y$ and $D_\parallel = D_z$. Observe that $e_1 = 0$, $e_2 = e_4 = D_\parallel - D_\perp$, $d_1 = d_2 = 5D_\perp + D_\parallel$, $d_3 = d_4 = 2D_\perp + 4D_\parallel$, and $d_5 = 6D_\perp$. Equation 4.7 simplifies to:

$$J(\mathbf{v}^i, \omega, \mathbf{D}) = \frac{2}{5} \sum_{k=1}^{3} \frac{\hat{d}_k(\mathbf{D})\hat{a}_k(\mathbf{v}^i, \mathbf{D})}{\hat{d}_k^2(\mathbf{D}) + \omega^2}, \tag{4.11}$$

where $\hat{d}_1 = 5D_\perp + D_\parallel$, $\hat{d}_2 = 2D_\perp + 4D_\parallel$, $\hat{d}_3 = 6D_\perp$, $\hat{a}_1 = a_1 + a_2$, $\hat{a}_2 = a_3 + a_4$, and $\hat{a}_3 = a_5$.

Since $\bar{v}$ is normalized, we simplify $\hat{a}_2$:

$$\hat{a}_1 = a_1 + a_2 = 3\bar{v}_2^2\bar{v}_3^2 + 3\bar{v}_1^2\bar{v}_3^2 = 3\bar{v}_3^2(1 - \bar{v}_3^2). \tag{4.12}$$

We observe that:

$$\begin{aligned} \delta_1 &= (-e_2)/e_4 = -1, \\ \delta_2 &= (-e_2)/e_4 = -1, \\ \delta_3 &= (2e_2)/e_4 = 2. \end{aligned} \tag{4.13}$$

We now simplify p_2:

$$\begin{aligned} p_2 &= \frac{1}{12}[\delta_1(3\bar{v}_1^4 + 2a_1 - 1) + \delta_2(3\bar{v}_2^4 + 2a_2 - 1) + \delta_3(3\bar{v}_3^4 + 2a_3 - 1)], \\ &= \frac{1}{12}[-3\bar{v}_1^4 - 3\bar{v}_2^4 + 6\bar{v}_3^4 + 2(-a_1 - a_2 + 2a_3)], \\ &= \frac{1}{12}[-3\bar{v}_1^4 - 3\bar{v}_2^4 + 6\bar{v}_3^4 - 6\bar{v}_2^2\bar{v}_3^2 - 6\bar{v}_1^2\bar{v}_3^2 + 12\bar{v}_1^2\bar{v}_2^2], \\ &= \frac{1}{4}[-\bar{v}_1^4 - \bar{v}_2^4 + 2\bar{v}_3^4 - 2\bar{v}_2^2\bar{v}_3^2 - 2\bar{v}_1^2\bar{v}_3^2 + 4\bar{v}_1^2\bar{v}_2^2]. \end{aligned} \tag{4.14}$$

Using the simplification of p_2 in (4.14), we simplify $\hat{a}_2$:

$$\begin{aligned}
\hat{a}_2 &= a_3 + a_4 = p_1 - p_2 + 3\bar{v}_1^2\bar{v}_2^2 \\
&= \frac{1}{4}[3\bar{v}_1^4 + 3\bar{v}_2^4 + 3\bar{v}_3^4 - 1 + \bar{v}_1^4 + \bar{v}_2^4 - 2\bar{v}_3^4 + 2\bar{v}_2^2\bar{v}_3^2 + 2\bar{v}_1^2\bar{v}_3^2 \\
&\quad - 4\bar{v}_1^2\bar{v}_2^2 + 12\bar{v}_1^2\bar{v}_2^2] \\
&= \frac{1}{4}[4\bar{v}_1^4 + 4\bar{v}_2^4 + \bar{v}_3^4 - 1 + 2\bar{v}_2^2\bar{v}_3^2 + 2\bar{v}_1^2\bar{v}_3^2 + 8\bar{v}_1^2\bar{v}_2^2] \\
&= \frac{1}{4}[4(\bar{v}_1^2 + \bar{v}_2^2)^2 + \bar{v}_3^4 - 1 + 2\bar{v}_2^2\bar{v}_3^2 + 2\bar{v}_1^2\bar{v}_3^2] \\
&= \frac{1}{4}[4(1 - \bar{v}_3^2)^2 - \bar{v}_3^4 - 1 + 2(1 - \bar{v}_3^2)\bar{v}_3^2] \\
&= \frac{1}{4}[4 - 8\bar{v}_3^2 + 3\bar{v}_3^4 - 1 + 2\bar{v}_3^2 - 2\bar{v}_3^4] \\
&= \frac{1}{4}[3 - 6\bar{v}_3^2 + \bar{v}_3^4] \\
&= \frac{3}{4}(1 - \bar{v}_3^2)^2.
\end{aligned} \tag{4.15}$$

Again, using the simplification of p_2 in (4.14), we simplify $\hat{a}_3$:

$$\begin{aligned}
\hat{a}_3 &= a_5 = p_1 + p_2 \\
&= \frac{1}{4}[3\bar{v}_1^4 + 3\bar{v}_2^4 + 3\bar{v}_3^4 - 1 - \bar{v}_1^4 - \bar{v}_2^4 + 2\bar{v}_3^4 - 2\bar{v}_2^2\bar{v}_3^2 - 2\bar{v}_1^2\bar{v}_3^2 + 4\bar{v}_1^2\bar{v}_2^2] \\
&= \frac{1}{4}[2\bar{v}_1^4 + 2\bar{v}_2^4 + 5\bar{v}_3^4 - 1 - 2\bar{v}_2^2\bar{v}_3^2 - 2\bar{v}_1^2\bar{v}_3^2 + 4\bar{v}_1^2\bar{v}_2^2] \\
&= \frac{1}{4}[2\bar{v}_1^4 + 2\bar{v}_2^4 + 5\bar{v}_3^4 - 1 - 2(1 - \bar{v}_3^2)\bar{v}_3^2 + 4\bar{v}_1^2\bar{v}_2^2] \\
&= \frac{1}{4}[2\bar{v}_1^4 + 4\bar{v}_1^2\bar{v}_2^2 + 2\bar{v}_2^4 + 5\bar{v}_3^4 - 1 - 2\bar{v}_3^2 + 2\bar{v}_3^4] \\
&= \frac{1}{4}[2(\bar{v}_1^2 + \bar{v}_2^2)^2 + 7\bar{v}_3^4 - 2\bar{v}_3^2 - 1] \\
&= \frac{1}{4}[2(1 - \bar{v}_3^2)(1 - \bar{v}_3^2) + 7\bar{v}_3^4 - 2\bar{v}_3^2 - 1] \\
&= \frac{1}{4}[1 - 6\bar{v}_3^2 + 9\bar{v}_3^4] \\
&= \frac{1}{4}(3\bar{v}_3^2 - 1)^2.
\end{aligned} \tag{4.16}$$

The axially symmetric diffusion model is therefore:

$$\rho_i^{pred}(\mathbf{v}^i, \mathbf{D}) = \frac{J(\mathbf{v}^i, \omega_N, \mathbf{D})}{J(\mathbf{v}^i, 0, \mathbf{D})}, \tag{4.17}$$

where ω_N is the resonance frequency of the ^{15}N spin,

$$J(\mathbf{v}^i, \omega, \mathbf{D}) = \frac{2}{5}\sum_{k=1}^{3} \frac{\hat{d}_k(\mathbf{D})\hat{a}_k(\mathbf{v}^i, \mathbf{D})}{\hat{d}_k^2(\mathbf{D}) + \omega^2}, \tag{4.18}$$

the components independent of the NH bonds are

$$\begin{aligned} \hat{d}_1(\mathbf{D}) &= 5D_\perp + D_\parallel, \\ \hat{d}_2(\mathbf{D}) &= 2D_\perp + 4D_\parallel, \\ \hat{d}_3(\mathbf{D}) &= 6D_\perp, \end{aligned} \tag{4.19}$$

the components dependent on the NH bonds are

$$\begin{aligned} \hat{a}_1(\mathbf{v}_i, \mathbf{D}) &= 3\bar{v}_3^2(1 - \bar{v}_3^2), \\ \hat{a}_2(\mathbf{v}_i, \mathbf{D}) &= \frac{3}{4}(1 - \bar{v}_3^2)^2, \\ \hat{a}_3(\mathbf{v}_i, \mathbf{D}) &= \frac{1}{4}(3\bar{v}_3^2 - 1)^2, \\ \bar{\mathbf{v}} &= \mathbf{V}^T\mathbf{v}^i, \end{aligned} \tag{4.20}$$

and $\mathbf{V}$ is an orthonormal matrix of the eigenvectors of $\mathbf{D}$.

4.2.1.3 Isotropic Diffusion Tensor Model

If all three eigenvalues of the experimental diffusion tensor are assumed to be equal then a simple isotropic diffusion tensor model can be used. We label the eigenvalue as D_c.

Observe that now $\hat{d}_1 = \hat{d}_2 = \hat{d}_3 = 6D_c$. From equation (4.17) we have:

$$\begin{aligned}\rho_i^{pred}(\mathbf{v}^i, \mathbf{D}) &= \frac{J(\mathbf{v}^i, \omega_N, \mathbf{D})}{J(\mathbf{v}^i, 0, \mathbf{D})} \\ &= \frac{\dfrac{6D_c(\hat{a}_1 + \hat{a}_2 + \hat{a}_3)}{36D_c^2 + \omega_N^2}}{\dfrac{\hat{a}_1 + \hat{a}_2 + \hat{a}_3}{6D_c}} \\ &= \frac{36D_c^2}{36D_c^2 + \omega_N^2}.\end{aligned} \tag{4.21}$$

The isotropic model is therefore:

$$\begin{aligned}\rho_i^{pred}(\mathbf{D}) &= \frac{1}{1 + (\omega_N \tau_c)^2}, \\ \tau_c &= 1/(6D_c),\end{aligned} \tag{4.22}$$

where ω_N is resonance frequency of the ^{15}N spin. Note that the isotropic model does not dependent on the orientation of the NH bonds.

We can solve equation (4.22) for D_c, which gives:

$$D_c = \left| \frac{\omega_n}{6} \sqrt{\frac{\rho_i^{pred}}{1 - \rho_i^{pred}}} \right|. \tag{4.23}$$

4.2.2 Algorithms for Solving the Three Diffusion Models

In this section we present three minimization algorithms that solve for the experimental diffusion tensor $\mathbf{D}_{exp}$ for each of the three models. We first solve for the diffusion tensor model in the isotropic case, and then use the solution as the initial guess for the other two models. Note that our algorithms use a nonlinear

least squares function "lsqnonlin" that we define in Section A.3.2, and which solves the problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{f}(\mathbf{x})\|^2, \tag{4.24}$$

where $\mathbf{x}^*$ is a local minimizer of $\mathbf{f}$.

In Algorithm 4.1 we give the algorithm for computing $\mathbf{D}_{exp}$ for the isotropic diffusion tensor model. The algorithm first uses (4.23) to get an initial estimate for D_x, and then uses a nonlinear least squares solver to fully solve (4.1).

Algorithm 4.1 rotdiffIso

Input: ρ^{exp} – defined in equation (4.3).

Output: $\mathbf{D}_{exp}$ – the experimental diffusion tensor.

1: **for** all bonds **do**

2: $d_i \leftarrow \left| \frac{\omega_N}{6} \sqrt{\frac{\rho_i^{exp}}{1-\rho_i^{exp}}} \right|$

3: **end for**

4: $x_0 \leftarrow <\mathbf{d}>$ {$<\mathbf{d}>$ is the mean of $\mathbf{d}$.}

5: $\mathbf{D}_{exp} \leftarrow \begin{bmatrix} x_0 & 0 & 0 \\ 0 & x_0 & 0 \\ 0 & 0 & x_0 \end{bmatrix}$

6: $\mathbf{D}_{exp} \leftarrow \text{lsqnonlin}(\chi_R^2(\emptyset, \mathbf{x}), \mathbf{D}_{exp})$ {$\emptyset$ represents the fact that the first parameter $\mathbf{v}$ to χ_R^2 in (4.2) is not used in the isotropic model.}

7: **return** $\mathbf{D}_{exp}$

We now proceed to describe an algorithm for solving the axially symmetric model for $\mathbf{D}_{exp}$ using the solution from the isotropic model as our initial guess. Recall from Definition 1.9 that we can express $\mathbf{V}$, the orthogonal matrix of the

eigenvectors of the diffusion tensor, using three Euler angles α, β, and γ. Since two of the eigenvalues are equal in the case of the axially symmetric model, the orientation of the diffusion tensor can be described by the orientation of the unique eigenvalue $D_{\|}$. Therefore, we can express the orientation using only α and β angles and set $\gamma = 0$.

Due to the eight-fold ambiguity of an eigendecomposition, equation (4.17) is π periodic in the two Euler angles. We take a similar approach to minimizing our equation as Walker et al. [81], but rather than randomly sampling a large number of angles for initial guesses to the nonlinear least squares solver, we only make four initial guesses for α and β: $[0, 0]$, $[0, \pi/2]$, $[\pi/2, 0]$, and $[\pi/2, \pi/2]$. Additionally, we alternate between the last and the first two eigenvalues being equal to handle the prolate and oblate case. We therefore perform nonlinear least squares for eight initial guesses. The complete algorithm is shown in Algorithm 4.2. Applying the algorithm to real and randomly generated synthetic data empirically confirms that we are able to correctly find the minimizer every time.

Algorithm 4.2 rotdifAxi

Input: ρ^{exp} – defined in equation (4.3), $\mathbf{v}$ – array of the normalized NH vectors, where $\mathbf{v}_i$ is associated with ρ_i^{exp}.

Output: $\mathbf{D}_{exp}$, the experimental diffusion tensor.

1: $\mathbf{D}_{iso} \leftarrow \text{rotdifIso}(\rho^{exp})$

2: $\hat{\mathbf{D}} \leftarrow \mathbf{D}_{iso}$

3: $\mathbf{D}_{exp} \leftarrow \mathbf{D}_{iso}$

4: **for** $j = 1, 2$ **do**

5: $\hat{D}_{jj} \leftarrow .5\hat{D}_{jj}$ {To switch between the prolate and oblate cases. The first eigenvalue changes from being $D_{\parallel}$ to $D_{\perp}$.}

6: **for** $\alpha = 0, \pi/2$ **do**

7: **for** $\beta = 0, \pi/2$ **do**

8: $\mathbf{x}_0 \leftarrow \mathbf{R}(\alpha, \beta, 0)\hat{\mathbf{D}}\mathbf{R}^T(\alpha, \beta, 0)$

9: $\mathbf{x}^* \leftarrow \text{lsqnonlin}(\chi_R^2(\mathbf{v}, \mathbf{x}), \mathbf{x}_0)$

10: **if** $\|\rho^{pred}(\mathbf{v}, \mathbf{x}^*) - \rho^{exp}\| < \|\rho^{pred}(\mathbf{v}, \mathbf{D}_{exp}) - \rho^{exp}\|$ **then**

11: $\mathbf{D}_{exp} \leftarrow \mathbf{x}^*$

12: **end if**

13: **end for**

14: **end for**

15: **end for**

16: **return** $\mathbf{D}_{exp}$

Finally, we describe the algorithm for solving the fully anisotropic diffusion

model for $\mathbf{D}_{exp}$. Again, we use the solution from the isotropic model as our initial guess for the solution. We make an observation similar to that for the axially symmetric case, that equation (4.17) is $\pi/2$ periodic for α, β, and γ. We therefore take eight initial guesses for the Euler angles: $[0, 0, 0]$, $[0, 0, \pi/4]$, $[0, \pi/4, 0]$, $[0, \pi/4, \pi/4]$, $[\pi/4, \pi/4, 0]$, and $[\pi/4, \pi/4, \pi/4]$. The complete algorithm is shown in Algorithm 4.3. Applying the algorithm to real and randomly generated synthetic data empirically confirms that we are able to correctly find the minimizer every time.

Algorithm 4.3 rotdifAni

Input: ρ^{exp} – defined in equation (4.3), $\mathbf{v}$ – array of the normalized NH vectors, where $\mathbf{v}_i$ is associated with ρ_i^{exp}.

Output: $\mathbf{D}_{exp}$, the experimental diffusion tensor.

1: $\mathbf{D}_{iso} \leftarrow \text{rotdifIso}(\rho_{exp})$

2: $\hat{\mathbf{D}} \leftarrow \mathbf{D}_{iso}$

3: $\mathbf{D}_{exp} \leftarrow \mathbf{D}_{iso}$

4: $\hat{D}_{11} \leftarrow .5\hat{D}_{11}$, $\hat{D}_{33} \leftarrow 1.5\hat{D}_{33}$ {Move away from the isotropic case, which causes division by 0.}

5: **for** $\alpha = 0, \pi/4$ **do**

6: **for** $\beta = 0, \pi/4$ **do**

7: **for** $\gamma = 0, \pi/4$ **do**

8: $\mathbf{x}_0 \leftarrow \mathbf{R}(\alpha, \beta, \gamma)\hat{\mathbf{D}}\mathbf{R}^T(\alpha, \beta, \gamma)$

9: $\mathbf{x}^* \leftarrow \text{lsqnonlin}(\chi_R^2(\mathbf{v}, \mathbf{x}), \mathbf{x}_0)$

10: **if** $\|\rho^{pred}(\mathbf{v}, \mathbf{x}^*) - \rho^{exp}\| < \|\rho^{pred}(\mathbf{v}, \mathbf{D}_{exp}) - \rho^{exp}\|$ **then**

11: $\mathbf{D}_{exp} \leftarrow \mathbf{x}^*$

12: **end if**

13: **end for**

14: **end for**

15: **end for**

16: **return** $\mathbf{D}_{exp}$

4.3 Predicting the Diffusion Tensor from Three-dimensional Structure

Having described methods for computing the experimental diffusion tensor we now present two different methods for predicting the diffusion tensor *ab initio* from the three-dimensional structure of a molecule.

Physically, the diffusion tensor represents how fast an object re-orients in a solvent. There are several forces that act upon the molecule in a solvent that affect its rotation. By far the most dominant force is the frictional force of the molecule as it grinds against the solvent during rotation. Therefore, the diffusion tensor is heavily related to how the surface of the object interacts with the solvent. As a consequence, the internal mass distribution can be ignored for the purposes of the calculation [15].

4.3.1 HYDRONMR

HYDRONMR is a well known method for computing the diffusion tensor from a three-dimensional structure of a molecule [15, 23]. HYDRONMR computes the diffusion tensor by modeling the molecule with spheres (beads) along its surface. Figure 4.1B shows the beads representation of the lysozyme molecule. The hydrodynamic properties of the beads can then be computed using the theoretical method described in Carrasco and Garcia de la Torre [15].

HYDRONMR requires that the interaction between each individual pair of beads be computed. Assuming that we have N beads, the computation is $O(N^3)$

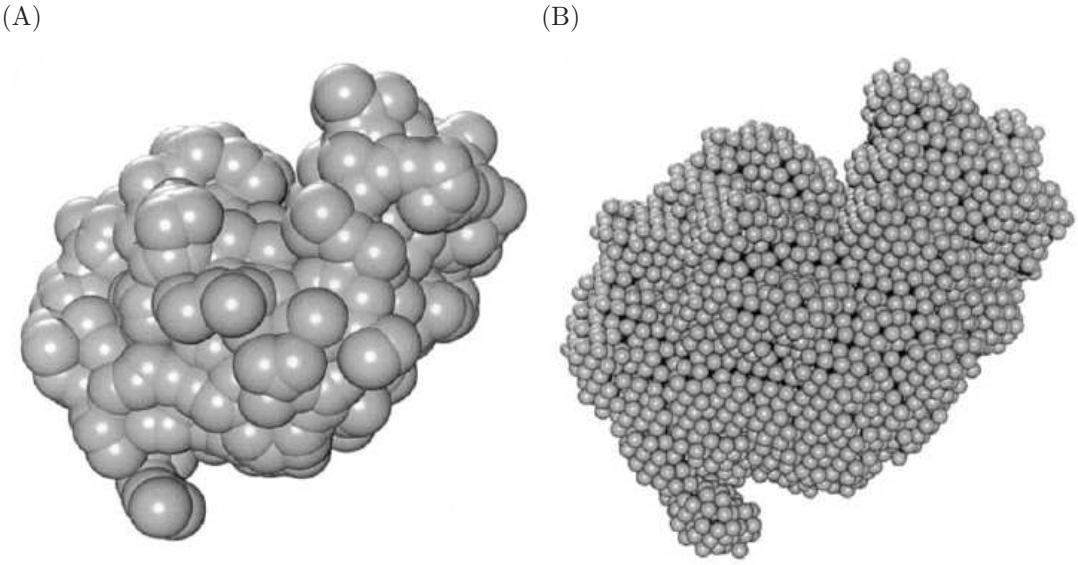

Figure 4.1: (A) Primary hydrodynamic model of lysozyme. (B) HYDRONMR shell model with bead radius $\sigma = 0.8$. [23]

for HYDRONMR. For a newer version of HYDRONMR, called *FAST-HYDRONMR* [15], which uses approximations to speed up the calculation, computation drops to $O(N^2)$. The smaller the radii of the beads the more accurate the representation of the molecule's shape; however more beads are then required to represent the shape of the molecule. This means that computing the diffusion tensor more accurately requires additional computation time. To overcome the expense of the computation, HYDRONMR starts out with beads of large radius σ, decreasing the radius several times, and then extrapolates the results to $\sigma \to 0$.

4.3.2 Equivalent Ellipsoid Method

An alternative method for computing the diffusion tensor is to represent the arbitrarily shaped molecule by a simpler shape for which the diffusion tensor can be

computed using known equations. We choose the shape to be an ellipsoid, since it is a simple geometric shape that can also be represented by a 3×3 positive definite symmetric matrix, just like the diffusion tensor.

Computing the diffusion tensor for a molecule is a two-step process: We first find an equivalent ellipsoid for the molecule; then we use the known equations to compute the diffusion tensor from the equivalent ellipsoid.

We tried two approaches for finding an equivalent ellipsoid. In the first approach we used the MVE (see Section 1.3.2.1) of a molecule. We found that this method gave inaccurate results for non-elliptically shaped molecules. In the second approach we used the PCAE (see Section 1.3.2.3) of the molecule. We found that this method produced better results and is the only method used in the rest of the Chapter.[1]

Having shown how to compute an equivalent ellipsoid $\mathcal{E}(\mathbf{A}, \mathbf{c})$ of a molecule, we now present Perrin's equations for computing the diffusion tensor from the computed equivalent ellipsoid [53].

Intuitively, Perrin's equations express the idea that molecules re-orients faster around the longer axis of an ellipsoid than around shorter axes. This physical behavior is similar to the behavior of a log of wood that rotates in water: The log rotates much easier around its length than in any other direction. Perrin's equations also show that the orientation of the principal axes (eigenvectors) of the diffusion tensor and the ellipsoid are the same.

Given the lengths of the equivalent ellipsoid's principal semi-axes, ℓ_1, ℓ_2, and

[1] During the computation of all the PCAE in this chapter we set HLT=2.8Å.

ℓ_3, and the ellipsoid's orientation matrix $\mathbf{V}$, the predicted diffusion tensor of the ellipsoid is:

$$\mathbf{D}_{pred} = \mathbf{V} \begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{bmatrix} \mathbf{V}^T, \tag{4.25}$$

where

$$\begin{aligned} D_1(\ell_1, \ell_2, \ell_3) &= \frac{k_b t}{I_1}, \\ D_2(\ell_1, \ell_2, \ell_3) &= \frac{k_b t}{I_2}, \\ D_3(\ell_1, \ell_2, \ell_3) &= \frac{k_b t}{I_3}, \end{aligned} \tag{4.26}$$

t is the temperature (° Kelvin), k_b is the Boltzmann constant,

$$\begin{aligned} I_1 &= \frac{16\pi\upsilon(\ell_2^2 + \ell_3^2)}{\ell_2^2 Q_2 + \ell_3^2 Q_3}, \\ I_2 &= \frac{16\pi\upsilon(\ell_1^2 + \ell_3^2)}{\ell_1^2 Q_1 + \ell_3^2 Q_3}, \\ I_3 &= \frac{16\pi\upsilon(\ell_1^2 + \ell_2^2)}{\ell_1^2 Q_1 + \ell_2^2 Q_2}, \end{aligned} \tag{4.27}$$

υ is the solvent viscosity, and

$$\begin{aligned} Q_1 &= \int_0^\infty \frac{ds}{\sqrt{(\ell_1^2 + s)^3(\ell_2^2 + s)(\ell_3^2 + s)}}, \\ Q_2 &= \int_0^\infty \frac{ds}{\sqrt{(\ell_2^2 + s)^3(\ell_1^2 + s)(\ell_3^2 + s)}}, \\ Q_3 &= \int_0^\infty \frac{ds}{\sqrt{(\ell_3^2 + s)^3(\ell_1^2 + s)(\ell_2^2 + s)}}. \end{aligned} \tag{4.28}$$

Thus, given a molecule the steps to predicting its diffusion tensor are: Compute the molecule's PCAE; compute the eigendecomposition of the PCAE; find the lengths of PCAE's axes using equation (1.8); and finally, predict the diffusion tensor using Perrin's equations.

4.4 Docking Method

Having derived the methods for computing the experimental diffusion tensor and predicting the diffusion tensor of a molecule, we now present ELMDOCK, our docking method for determining domain position of a molecule made up of two domains for which the individual three-dimensional structure and the associated experimental diffusion tensors are known.

Just like in PATIDOCK, we first need to align the two domains based on their experimental diffusion tensor. Let M be a molecule made up of two domains, A and B, with experimentally measured ratio of transverse and longitudinal relaxation rates ρ^{exp}, and the associated experimental diffusion tensors $\mathbf{D}_A$ and $\mathbf{D}_B$ (computed by ROTDIF using the fully anisotropic model). The sorted eigendecompositions of the experimental diffusion tensors for A and B are

$$\mathbf{D}_A = \mathbf{V}\hat{\mathbf{D}}_A\mathbf{V}^T, \tag{4.29}$$

$$\mathbf{D}_B = \mathbf{V}\hat{\mathbf{D}}_B\mathbf{V}^T. \tag{4.30}$$

We assume that the diffusion tensors have unique principal components and so there are only four possible sorted eigendecompositions for $\mathbf{D}_A$ and $\mathbf{D}_B$. We also assume that A and B tumble together in a solution. When the two domains tumble as one unit, $\mathbf{D}_A \approx \mathbf{D}_B$ [36]. Similar to the procedure in PATIDOCK, we align the two domains based on their diffusion tensors and recompute the overall experimental diffusion tensor $\mathbf{D}_{exp}$ for this newly aligned structure M using ROTDIF.[2] We refer

[2]See Section 3.2.1 on how to align two domains using their alignment tensors. The procedure for the diffusion tensor is identical.

to the newly computed overall experimental diffusion tensors as $\mathbf{D}_{exp}$.

Note that there is still a four-fold ambiguity in the alignment of the two domains, and that our docking algorithm should be repeated for each of the four alignments, and another method should be used to evaluate which of the four possible orientations is correct.

Having just given the procedure for aligning M using the individual diffusion tensors, in the rest of the section we will assume that A and B, and hence M, are already properly aligned. We refer to the method for finding the optimal translation between two domains when they are already aligned as ELMDOCK-t.

Let $B + \mathbf{x}$ represent a shift in the position of each atom of B by a vector $\mathbf{x} \in \mathbb{R}^3$. We define $M(\mathbf{x})$ to be the positions of all the atoms from A and $B + \mathbf{x}$.

The goal of ELMDOCK is to find a shift $\mathbf{x}^*$ in the position of the B molecule so that the combined molecule $M(\mathbf{x}^*)$ has the same diffusion tensor as the experimental diffusion tensor $\mathbf{D}_{exp}$. Specifically, we find $\mathbf{x}^*$ such that

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \chi^2_D(\mathbf{x}), \tag{4.31}$$

$$\chi^2_D(\mathbf{x}) = \sum_{i=1}^{3}\sum_{j=1}^{3}[F_{ij}(M(\mathbf{x})) - (D_{exp})_{ij}]^2, \tag{4.32}$$

where $\mathbf{F}(M)$ is a function that predicts the diffusion tensor of a molecule M. For example $\mathbf{F}(M)$ could be HYDRONMR [23] or ELM [58].

Solving equation (4.31) directly will be slow for two reasons: First, the diffusion tensor needs to be recalculated for each iteration of the minimization. Since computing the diffusion tensor involves computation of the Richards' smooth molecular surface, this computation is expensive. Second, a nonlinear least squares method

will be slow because of the need to approximate the Jacobian for $\mathbf{F}(M)$ using finite differences. The finite differences approximation leads to further problems since we expect the function to not be perfectly smooth due to the sudden changes in the surface points as the two domains collide. In addition, finite differences will require us to compute $M(\mathbf{x})$ three additional times for each minimization iteration.

To explain how we solve equation (4.31) in a more efficient way we dissect the ELM method. Recall from Section 4.3.2 that the steps for computing the predicted diffusion tensor using ELM for any molecule M are

$$M(\mathbf{X}) \xrightarrow{\text{SURF}} S \xrightarrow{\text{PCA}} \mathbf{C} \to \mathcal{E} \xrightarrow{\text{Perrin's equations}} \mathbf{D}_{pred}, \tag{4.33}$$

where S is the set of sample points from Richards' smooth surface for molecule M, $\mathbf{C}$ is the covariance matrix of S, and $\mathcal{E}$ is the associated PCAE. When $\mathbf{X} = \mathbf{x}^*$ we expect that $\mathbf{D}_{pred} \approx \mathbf{D}_{exp}$.

The goal of our docking algorithm is to reverse these steps in an efficient manner so that given D_{exp}, we find the best fitting molecule $M(x^*)$, and hence x^*. We accomplish this in two separate steps:

$$\mathbf{D}_{exp} \xrightarrow{1} \mathbf{C}^* \xrightarrow{2} M(\mathbf{x}^*). \tag{4.34}$$

Since we are given $\mathbf{D}_{exp}$ for our input, we set $\mathbf{D}_{pred} = \mathbf{D}_{exp}$. If our problem is well conditioned, small errors in the prediction of the diffusion tensor or $\mathbf{D}_{exp}$ will result in a small difference between the true solution and $\mathbf{x}^*$. We test if ELMDOCK is well conditioned in Section 4.5.2.

In step 1 we find $\mathbf{C}^*$ by solving the equation

$$\mathbf{C}^* = \arg\min_{\mathbf{C}} \chi^2_C(\mathbf{x}), \tag{4.35}$$

where

$$\chi^2_C(\mathbf{x}) = \sum_{i=1}^{3}\sum_{j=1}^{3}(L_{ij}(\mathbf{C}) - (D_{exp})_{ij})^2, \tag{4.36}$$

and $\mathbf{L}(\mathbf{C})$ is the function that returns the diffusion tensor of a covariance matrix $\mathbf{C}$. $\mathbf{L}$ computes the diffusion tensor by first computing the ellipsoid using Theorem C.1, and then uses Perrin's equations to compute the diffusion tensor of this ellipsoid. We present a detailed description of step 1 in Section 4.4.1.

In step 2 we efficiently find $\mathbf{x}^*$ by solving the equation

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \chi^2_G(\mathbf{x}), \tag{4.37}$$

where

$$\chi^2_G(\mathbf{x}) = \sum_{i=1}^{3}\sum_{j=1}^{3}(G_{ij}(\mathbf{x}) - C^*_{ij})^2, \tag{4.38}$$

and $\mathbf{G}(\mathbf{x})$ is a function, described in Section 1.3.2.3, that returns the covariance matrix of the surface of a molecule $M(\mathbf{x})$. In order to describe the minimization method for χ^2_G, we first present two methods for approximating $\mathbf{G}(\mathbf{x})$ in Section 4.4.2. We then use these approximation methods to efficiently minimize χ^2_G in Section 4.4.3.

We present the outline of our complete docking method in Algorithm 4.4. The relevant references are presented in the comment section of each line, and are explained in detail in the rest of the chapter.

Algorithm 4.4 Docking Algorithm

Input: $\mathbf{D}_{exp}$ – three-dimensional structure of A and B that are already aligned, $\mathbf{G}(\mathbf{x})$ – a function that computes the covariance matrix of $M(\mathbf{x})$.

Output: $\mathbf{x}^*$ – the translation of B that yields the best docking solution as measured by our energy function.

1: Compute the covariance matrix $\mathbf{C}^*$ from $\mathbf{D}_{exp}$ {See Section 4.4.1.}

2: $\mathbf{x}^* \leftarrow \infty$

3: **for** every initial guess x_0 {See Section 4.4.3.1.} **do**

4: $k \leftarrow 0$

5: $\mathbf{x}_k \leftarrow x_0$

6: **while** stopping condition not reached {See Section 4.4.3.3}. **do**

7: Compute a descent direction $\mathbf{p} \in \mathbb{R}^3$ for $\chi^2_G(\mathbf{x}_k)$ {See Section 4.4.3.2}.

8: Set $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \mathbf{p}$

9: Set $k \leftarrow k + 1$

10: **end while**

11: **if** $\chi^2_G(\mathbf{x}_k) < \chi^2_G(\mathbf{x}^*)$ **then**

12: $\mathbf{x}^* \leftarrow \mathbf{x}_k$

13: **end if**

14: **end for**

15: **return** $\mathbf{x}*$

4.4.1 Step 1: Diffusion Tensor to Covariance Matrix

In this section we describe step 1 of our docking method, where we solve equation (4.35) by finding a covariance matrix of an ellipsoid that has the diffusion tensor value $\mathbf{D}_{exp}$. Then, given the covariance matrix it is much easier to find $\mathbf{x}^*$ since the covariance matrix is directly proportional to the surface points of the domain, while the relationship between $\mathbf{x}^*$ and the diffusion tensor is much harder to quantify.

Recall from Section 4.3.2 that the orientation of the diffusion tensor $\mathbf{D}_{exp}$ and of the associated covariance matrix $\mathbf{C}^*$ is the same. That means that the eigendecompositions of $\mathbf{D}_{exp}$ and $\mathbf{C}^*$ are

$$\mathbf{D}_{exp} = \mathbf{V} \begin{bmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_3 \end{bmatrix} \mathbf{V}^T, \tag{4.39}$$

and

$$\mathbf{C}^* = \mathbf{V} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{V}^T. \tag{4.40}$$

By performing an eigendecomposition on $\mathbf{D}_{exp}$, we get the values for $\mathbf{V}$, D_x, D_y, and D_z. Given D_x, D_y, and D_z, we now solve equation (4.26) (Perrin's equations) for the lengths of the ellipsoid's principal semi-axes ℓ_1, ℓ_2, and ℓ_3 that yield the diffusion tensor values D_x, D_y, and D_z.

Once we have gotten the lengths of the ellipsoid's principal semi-axes $[\ell_1, \ell_2, \ell_3]$, and its orientation $\mathbf{V}$, by equation (1.8) and Theorem C.1, the covariance matrix

$\mathbf{C}^*$ of the ellipsoid is:

$$\mathbf{C}^* = \mathbf{V} \begin{bmatrix} \ell_1^2/3 & 0 & 0 \\ 0 & \ell_2^2/3 & 0 \\ 0 & 0 & \ell_3^2/3 \end{bmatrix} \mathbf{V}^T. \tag{4.41}$$

We observe that we can compute the Jacobian of equation (4.26), and solve for $[\ell_1, \ell_2, \ell_3]$ by using nonlinear least squares method given a proper initial guess for the values. We need to be careful when selecting the initial guess, since the solution for $[\ell_1, \ell_2, \ell_3]$ is not necessarily unique.

In Figure 4.2 we show the mapping of lengths of ellipsoid's principal semi-axes $[\ell_1, \ell_2, \ell_3]$, where $0 < \ell_1 \leq \ell_2 \leq \ell_3$, sampled at 2Å intervals, into the diffusion tensor space using Perrin's equations. To better visually spread out the points we adjust each eigenvalue of the diffusion tensor using the function T, where

$$T(D_i) = \log(\log(\log(\log(\log(D_i + 1) + 1) + 1) + 1) + 1), \tag{4.42}$$

for $i = x, y, z$.

Observe that the color gradient in Figure 4.2 is fairly smooth. This implies that in Perrin's equations, neighboring values in the domain map into neighboring values in the range. To confirm this observation, we split the cube of the diffusion tensor space (from 0 to $\max(T)$) into $20 \times 20 \times 20$ cubes, and for each cube observe which triples of $[\ell_1, \ell_2, \ell_3]$ are mapped into that cube. We performed hierarchical clustering on the triples based on their Euclidean distances and recorded the number of clusters.[3] This shows how many disconnected parts of the domain are being mapped

[3]We measure the distance between two clusters as the Euclidean distance between the two

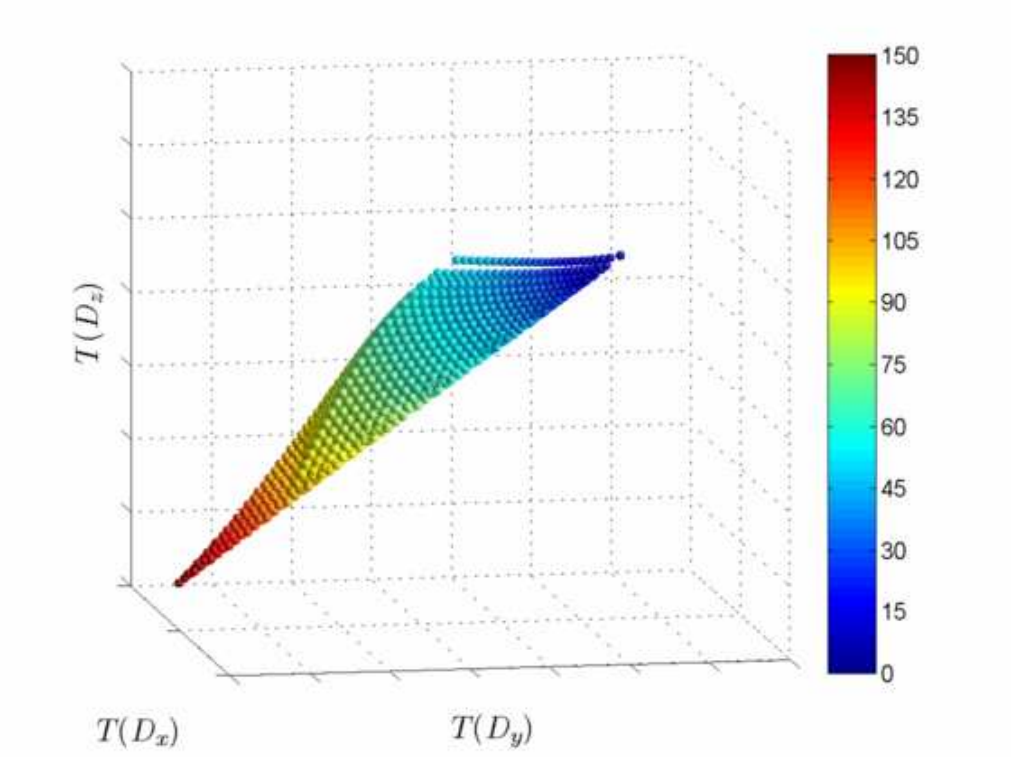

Figure 4.2: A sample of all possible triples of the principal semi-axes lengths $\ell_1 \leq \ell_2 \leq \ell_3$ of an ellipsoid from 1Å to 50Å sampled at 2Å intervals mapped into the diffusion tensor principal components using Perrin's equations. The colors represent the value $\ell_1 + \ell_2 + \ell_3$ of each sample point.

into a connected range (the cube). The maximum number of clusters in any cube was two, and the majority of the occupied cubes contain only one cluster. Therefore, we expect at most two distinct triplets of $[\ell_1, \ell_2, \ell_3]$ to have the same diffusion tensor. Since there are only two solutions, we can try to find both of the solutions by simply trying eight different starting points, $[1, 1, 1]$, $[1, 1, 1000]$, $[1, 1000, 1]$, $[1, 1000, 1000]$, $[1000, 1, 1]$, $[1000, 1, 1000]$, $[1000, 1000, 1]$, and $[1000, 1000, 1000]$ in the nonlinear least squares algorithm. In practice, we are able to eliminate all but

closest points of the clusters. We set the cluster cutoff at 3.7Å, a value that is smaller than 4Å, the shortest possible distance between two non-adjacent sample points.

one of the solutions by simply checking if the computed lengths make sense given the known shape of the domains.

Having gotten the ellipsoid's principal semi-axes lengths $[\ell_1, \ell_2, \ell_3]$, we compute covariance matrix $\mathbf{C}^*$ by (4.41).

We have now solved equation (4.35), and can therefore move to step 2 of our docking method.

4.4.2 Estimating the Covariance Matrix of a Molecule

Before we can present step 2 of our docking method, we need to describe two methods for approximating $\mathbf{G}(\mathbf{x})$, a function that computes the covariance matrix of $M(\mathbf{x})$. Since each iteration of a Newton-like minimization requires an evaluation of the target function and a computation of a descent step (see Appendix A), we first derive two algorithms that provide fast approximations to the function $\mathbf{G}(\mathbf{x})$, and by extension an approximation for a descent step.

The first algorithm allows us to quickly compute the descent step for our minimization algorithm by finding a quadratic approximation of the covariance matrix around the current value $\mathbf{x}$. The minimizer of this quadratic function can be efficiently computed by a Newton-like method. We can express the approximation as

$$\mathbf{G}(\mathbf{x}+\mathbf{p}) \approx \mathbf{G}(\mathbf{x}) + \mathbf{Q}(\mathbf{p}), \tag{4.43}$$

where

$$Q_{ij}(\mathbf{p}) = \kappa p_i p_j + K_{ij} p_i + K_{ji} p_j, \tag{4.44}$$

$i, j = 1, 2, 3$, $\mathbf{p} \in \mathbb{R}^3$, κ is a constant, and $\mathbf{K}$ is a constant 3×3 matrix. We derive the formulae for the constants in Section 4.4.2.1.

The second algorithm, $\mathbf{G}^{fast}$, is a more accurate method for approximating $\mathbf{G}(\mathbf{x})$ but is computationally slower. It provides a method for estimating the covariance matrix of a molecule by only computing the Richards' molecular surface initially, and quickly adjusting it for different values of $\mathbf{x}$. We describe $\mathbf{G}^{fast}$ in Section 4.4.2.2.

4.4.2.1 Quadratic Approximation of a Molecule's Covariance Matrix

In this section we derive the quadratic approximation $\mathbf{Q}$ of the function $\mathbf{G}$ around a point $\mathbf{x}$. The approximation will allow us to quickly approximate the descent step for our minimization of χ^2_G.

Let $\mathbf{a}^1, \ldots, \mathbf{a}^{n_a}$ be the surface points for $M(\mathbf{x})$ that come from domain A and let $\mathbf{b}^1, \ldots, \mathbf{b}^{n_b}$ be the surface points for $M(\mathbf{x})$ that come from domain B. Observe that the set of surface points does not change much as the position of B is perturbed by $\mathbf{p}$. The majority of the change in the covariance matrix comes from the fact that $\mathbf{b}^i$ points are shifted and not from the actual change in the surface points. The larger $\|\mathbf{p}\|$ is, the more we expect the set of the surface points to change, but at the same time the translation of points that remain on the surface also contributes a greater weight. Thus, we expect that we can estimate the covariance matrix well at $\mathbf{x}+\mathbf{p}$ by simply adjusting the points $\mathbf{b}$ by $\mathbf{p}$ and recomputing the covariance matrix.

We now write out the equation for approximating $G_{ij}(\mathbf{x}+\mathbf{p})$ by simply com-

puting the covariance matrix of set **a** and the adjusted set **b** using the equation (1.15):

$$
\begin{aligned}
G_{ij}(\mathbf{x}+\mathbf{p}) &\approx \frac{\sum_{v=1}^{n_a} a_i^v a_j^v + \sum_{v=1}^{n_b} (b_i^v + p_i)(b_j^v + p_j)}{n_a + n_b} \\
&\quad - \frac{\left[\sum_{v=1}^{n_a} a_i^v + \sum_{v=1}^{n_b} (b_i^v + p_i)\right]\left[\sum_{v=1}^{n_a} a_j^v + \sum_{v=1}^{n_b} (b_j^v + p_j)\right]}{(n_a+n_b)^2} \\
&= \frac{\sum_{v=1}^{n_a} a_i^v a_j^v + \sum_{v=1}^{n_b} b_i^v b_j^v}{n_a + n_b} \\
&\quad - \frac{\left[\sum_{v=1}^{n_a} a_i^v + \sum_{v=1}^{n_b} b_i^v\right]\left[\sum_{v=1}^{n_a} a_j^v + \sum_{v=1}^{n_b} b_j^v\right]}{(n_a+n_b)^2} \\
&\quad + \frac{\sum_{v=1}^{n_b} (b_i^v p_j + b_j^v p_i + p_i p_j)(n_a+n_b)}{(n_a+n_b)^2} \\
&\quad - \frac{\left(\sum_{v=1}^{n_a} a_i^v + \sum_{v=1}^{n_b} b_i^v\right) n_b p_j + \left(\sum_{v=1}^{n_a} a_j^v + \sum_{v=1}^{n_b} b_j^v\right) n_b p_i + n_b^2 p_i p_j}{(n_a+n_b)^2} \\
&= G_{ij}(\mathbf{x}) + \frac{(n_a+n_b)\sum_{v=1}^{n_b} b_i^v p_j + (n_a+n_b)\sum_{v=1}^{n_b} b_j^v p_i + (n_a+n_b) n_b p_i p_j}{(n_a+n_b)^2} \\
&\quad - \frac{n_b p_j \sum_{v=1}^{n_a} a_i^v + n_b p_j \sum_{v=1}^{n_b} b_i^v + n_b p_i \sum_{v=1}^{n_a} a_j^v + n_b p_i \sum_{v=1}^{n_b} b_j^v + n_b^2 p_i p_j}{(n_a+n_b)^2} \\
&= G_{ij}(\mathbf{x}) + Q_{ij}(\mathbf{p}),
\end{aligned}
\tag{4.45}
$$

where

$$
Q_{ij}(\mathbf{p}) = \kappa p_i p_j + K_{ij} p_i + K_{ji} p_j, \tag{4.46}
$$

and

$$
\kappa = \frac{n_a n_b}{(n_a+n_b)^2}, \tag{4.47}
$$

$$
K_{ij} = \frac{(n_a+n_b)\sum_{v=1}^{n_b} b_j^v - n_b(\sum_{k=1}^{n_a} a_j^v + \sum_{v=1}^{n_b} b_j^v)}{(n_a+n_b)^2}, \tag{4.48}
$$

for $i, j = 1, 2, 3$.

Observe that if the two sets of points do not change during the translation **p** (i.e. the two domains never collide, either before or after) our approximation yields

an exact value, and that the analytical formula for the Jacobian of $\mathbf{Q}$ is trivially computed.

4.4.2.2 Geometric Approximation of a Molecule's Covariance Matrix

Computing the approximation $\mathbf{Q}$ around $\mathbf{x}$ requires that we first compute $\mathbf{G}(\mathbf{x})$. Recomputing $\mathbf{G}$ is computationally expensive, and we would like to avoid it as much as possible. In this section we derive a method, called $\mathbf{G}^{fast}$, for approximating $\mathbf{G}$ that is more accurate than the quadratic approximation derived in Section 4.4.2.1, but computationally slower, because it redetermines the set of surface points.

Recall from Section 1.3.2.3 the steps to computing the covariance matrix of a molecule. The method has been shown to be relatively fast when calculating covariance matrices for different molecules. However, in the case of rigid docking, the shape of the domains does not change, so it is computationally wasteful to fully recompute the surface of the domains every time we want to evaluate $\mathbf{G}(\mathbf{x})$.

Since we assumed that the three-dimensional structure of the domains does not change as the molecules come closer together, we compute the surfaces of the two molecules initially and figure out how to adjust their surfaces as the molecules move closer and start colliding. We label the set of surface points of molecule A as S_A, and the surface points of B as S_B. The surface points of $B + \mathbf{x}$ are therefore written as $S_B + \mathbf{x}$, representing the fact that the surface points of B are shifted by $\mathbf{x}$. The goal is to determine which surface points in S_A and S_B remain as part of the overall surface of the combined molecule, and which are no longer on the surface.

To figure out what surface points disappear in a collision we need to use a collision detection algorithm. Figure 4.3 shows how as two domains come closer together the surface points of one domain start colliding with the PCA ellipsoid of the second domain, thus no longer participating in the definition the combined surface. We approximate the surfaces of our two domains by ellipsoids, find which points are colliding, and then remove these points from our calculation.

First, we find ellipsoids that provide a good representation of the surfaces of the A and B molecules. We have been using the PCAE to describe the surface for the diffusion tensor computation and we use it here, too. Let the PCAE for molecule A be $\mathcal{E}_A$, and for molecule $B + \mathbf{x}$ be $\mathcal{E}_B^{\mathbf{x}}$. We find all points in S_a that do not collide with $\mathcal{E}_B^{\mathbf{x}}$ and all points in $S_B + \mathbf{x}$ that do not collide with $\mathcal{E}_A$[4], and compute the covariance matrix of these points.

Let $\mathbf{a}^1, \ldots, \mathbf{a}^{n_a}$ be the set of points in S_A that do not collide with $\mathcal{E}_B^{\mathbf{x}}$, and let $\mathbf{b}^1, \ldots, \mathbf{b}^{n_b}$ be the set of points in $S_B + \mathbf{x}$ that do not collide with $\mathcal{E}_A$. Using equation (1.15), the covariance matrix for the set $\mathbf{a}$ and $\mathbf{b}$ is computed as

$$G_{i,j}^{fast}(\mathbf{x}) = \frac{\sum_{v=1}^{n_a} a_i^v a_j^v + \sum_{v=1}^{n_b} b_i^v b_j^v}{n_a + n_b} - \frac{\left(\sum_{v=1}^{n_a} a_i^v + \sum_{v=1}^{n_b} b_i^v\right)\left(\sum_{v=1}^{n_a} a_j^v + \sum_{v=1}^{n_b} b_j^v\right)}{(n_a + n_b)^2}, \tag{4.49}$$

for $i, j = 1, 2, 3$.

The major source of error in $\mathbf{G}^{fast}$, comes from the fact that the collisions are approximately computed. If the shape of the domain is not approximated well by

[4]We simply check each surface point to see if it is inside or outside the ellipsoid. Note that this is not equivalent to recomputing Richards' smooth molecular surface on $M(\mathbf{x})$, unless the two domains do not intersect when B is shifted by $\mathbf{x}$.

(A) (B)

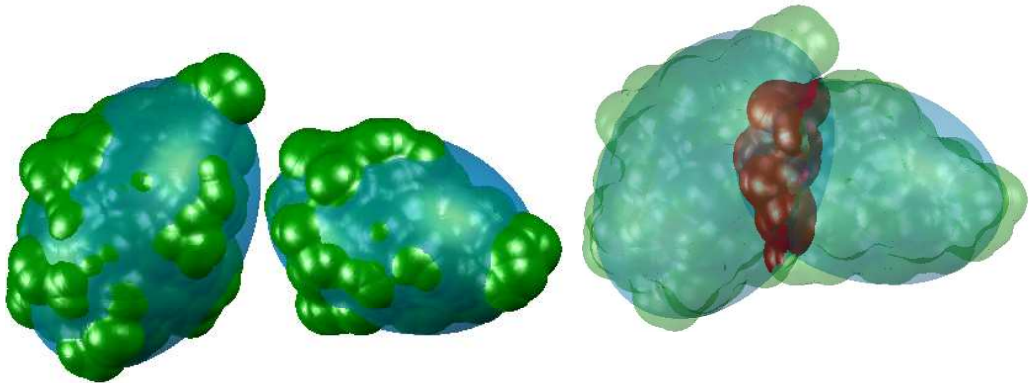

Figure 4.3: Two domains of Ub/UBA complex coming closer together, with the individual PCAE drawn around the surface points computed with HLT=2.8Å. (A) The domains are apart so all the surface points contribute to the overall PCAE. (B) The domains come closer together, and some of the previously surface points no longer contribute to the overall PCAE (colored red).

an ellipsoid, we expect $\mathbf{G}^{fast}$ to not be very accurate. We analyze the accuracy of using this approximation in Section 4.5.

4.4.3 Step 2: Equivalent Ellipsoid to Domain Position

Having discussed computation of the covariance matrix $\mathbf{C}^*$ in Section 4.4.2, we now describe step 2, where we find $\mathbf{x}^*$ such that the covariance matrix of the surface points of $M(\mathbf{x}^*)$ is equal to $\mathbf{C}^*$.

We use a Newton-like method to minimize χ^2_G. In Section 4.4.3.1 we describe how we choose an initial starting point $\mathbf{x}_0$; in Section 4.4.3.2 we show how to compute a good descent direction; and in Section 4.4.3.3 we give the stopping conditions. See Algorithm 4.4 for an outline of our Newton-like method.

4.4.3.1 Computing the Initial Starting Point

Recall that every Newton-like minimization method starts out with the initial starting point of $\mathbf{x}_0$. Choosing the initial starting position $\mathbf{x}_0$ is important because due to the symmetry inherent in the covariance matrix, just like in PATIDOCK, there are multiple local minimizers of χ^2_G. Figure 4.4 shows two local minimizers for Ub/UBA complex; both have similar covariance matrices of the surface points.

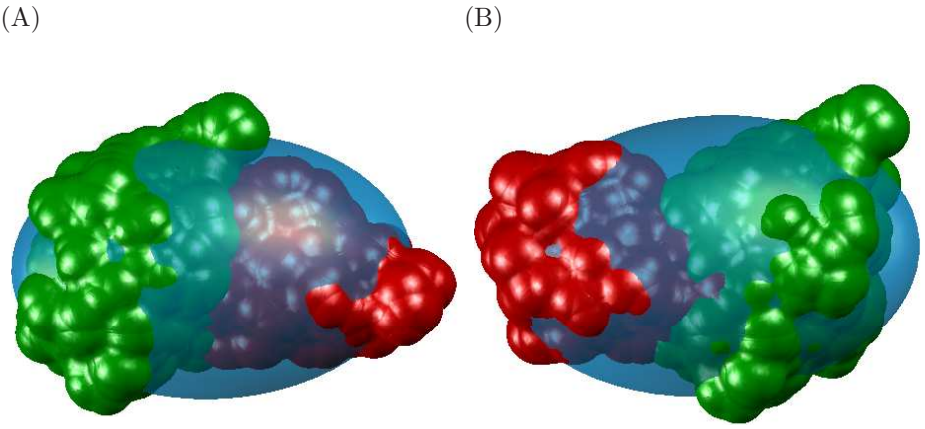

Figure 4.4: Two equivalent docking solutions for the Ub/UBA complex; both have similar covariance matrices of the surface points. The surface of the complex with HLT=2.8Å is drawn along with the equivalent PCAE for the specific solution. Domain A is drawn in green and domain B is drawn in red. (A) The solution with the correct positioning of the second domain. (B) The solution with a similar covariance matrix to the first solution, but with an incorrect domain placement.

In order to solve for the global solution using the method described in Algorithm A.2, we need to choose starting points $\mathbf{x}_0$ close to each of the local minimizers in order to make sure that we find the correct overall minimizer. To compute such a set of $\mathbf{x}_0$, we replace the minimization problem given in equation (4.38) by an

approximation where we only look at the diagonal elements:

$$\chi_g^2(\mathbf{x}) = \sum_{i=1}^{3}(G_{ii}(\mathbf{x}) - C_{ii}^*)^2. \tag{4.50}$$

Minimizing this new target function should yield a good initial guess, since if we have good model then

$$\chi_G^2(\mathbf{x}) \approx \mathbf{0} \Leftrightarrow \chi_g^2(\mathbf{x}) \approx \mathbf{0}. \tag{4.51}$$

χ_g^2 is too complicated to easily be solved analytically. We therefore approximate it using equation (4.43) and (4.44).

$$\begin{aligned}\chi_g^2(\mathbf{x}) &\approx \sum_{i=1}^{3}(G_{ii}(\mathbf{0}) + Q_{ii}(\mathbf{x}) - C_{ii}^*)^2 \\ &= (\kappa x_1^2 + 2K_{11}x_1 + \nu_1)^2 + \\ &\qquad (\kappa x_2^2 + 2K_{22}x_2 + \nu_2)^2 + \\ &\qquad (\kappa x_3^2 + 2K_{33}x_3 + \nu_3)^2,\end{aligned} \tag{4.52}$$

where

$$\nu_i = G_{ii}(\mathbf{0}) - C_{ii}^*. \tag{4.53}$$

We now analytically minimize equation (4.52). Since minimization of this equation is a minimization of three independent quadratic equations, each equation can be solved separately for its minimizer. The minimization of each of the three quadratic equations gives a maximum of eight initial guesses for $\mathbf{x}_0$. Therefore,

$$\mathbf{x}_0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \end{bmatrix}, \tag{4.54}$$

where

$$x_i^0 = \begin{cases} \frac{-2K_{ii} \pm \sqrt{4K_{ii}^2 - 4\kappa\nu_i}}{2\kappa} & \text{if } K_{ii}^2 - 4\kappa\nu_i > 0, \\ \\ \frac{-K_{ii}}{\kappa} & \text{otherwise.} \end{cases} \tag{4.55}$$

In practice we only end up with two initial values for $\mathbf{x}_0$.

4.4.3.2 Approximating the Descent Step

Having computed the initial guess $\mathbf{x}_0$ in Section 4.4.3.1, we now show how to efficiently guide our iterative minimization. Recall from Appendix A that the most important step in a minimization is finding a descent step.

At each step k, we would like to find the value for $\mathbf{p}$ such that $\mathbf{x}_k + \mathbf{p}$ minimizes χ_G^2:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \arg\min_{\mathbf{p}} \chi_G^2(\mathbf{x}_k + \mathbf{p}). \tag{4.56}$$

However, finding the true minimizer of χ_G^2 directly is too complicated.

We can approximate $\chi_G^2(\mathbf{x}_k + \mathbf{p})$ by using our quadratic function approximation derived in Section 4.4.2.1:

$$\chi_G^2(\mathbf{x}_k + \mathbf{p}) \approx \tilde{\chi}_G^2(\mathbf{p}) = \sum_{i=1}^{3} \sum_{j=1}^{3} (G_{ij}(\mathbf{x}_k) + Q_{ij}(\mathbf{p}) - C_{ij}^*)^2. \tag{4.57}$$

Observe that the Jacobian of $\mathbf{Q}$ can be trivially computed, and we can very quickly solve for the value of $\mathbf{p}$ that minimizes $\tilde{\chi}_G$.

Therefore, the equation for our next step in each iteration becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \arg\min_{p} \tilde{\chi}_G^2(\mathbf{p}). \tag{4.58}$$

We now iteratively converge to the true minimizer of χ_G^2.

To evaluate equation (4.58) we need to evaluate $\mathbf{G}(\mathbf{x}_k)$. We speedup the minimization by using $\mathbf{G}^{fast}(\mathbf{x}_k)$ (equation (4.49)) instead of $\mathbf{G}(\mathbf{x}_k)$ in the first few iterations of the minimization, and then switch to the computationally more expensive $\mathbf{G}(\mathbf{x}_k)$ when our step length drops below 0.5Å.

4.4.3.3 Stopping Conditions

There are three conditions which terminate our algorithm: The first case is when we are close enough to the solution

$$\|\mathbf{G}^{fast}(\mathbf{x}_k) - \mathbf{C}^*\|_F^2 < \epsilon_1. \tag{4.59}$$

The second case is when we are not making enough progress:

$$\|\mathbf{G}^{fast}(\mathbf{x}_k) - \mathbf{G}^{fast}(\mathbf{x}_{k-1})\|_F^2 < \epsilon_2. \tag{4.60}$$

And the last case is when the step size is small enough:

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_F^2 < \epsilon_3. \tag{4.61}$$

4.5 Results

In this section we present the results for ELMDOCK. Due to the four-fold ambiguity of relative orientation of S_2 relative to S_1 and the existence of two symmetrical local minimizers for each orientation, there usually will be at least eight potential solutions. Similar to the analysis of PATIDOCK in Section 3.3, we measure the distance between the *correct minimizer* $\tilde{\mathbf{x}}$, and the best predicted minimizer

$\mathbf{x}^*$. The experimental setup is identical to PATIDOCK, and the overall approach is almost identical to the Results section in PATIDOCK.

We implemented ELMDOCK in MATLAB 7.8.0 and performed all calculations and timing on a single 1.7 GHz Pentium M processor with 1.5 GB RAM, running Windows XP. In the current implementation we use only the last stopping condition, $\epsilon_3 = .2$Å, and set $\epsilon_1 = \epsilon_2 = 0$.

We run the algorithms on two distinct datasets. The first dataset, which we refer to as COMPLEX, is a set of 76[5] protein-protein complexes described in Mintseris et al. [48]. The COMPLEX dataset provides a wide variety of protein-protein complexes, but it contains no experimental diffusion tensor data. For each complex we generate a *synthetic diffusion tensor* $\mathbf{D}_{syn}$ by predicting the diffusion tensor on the already known complex structure using ELM. This allows us to test our method under ideal experimental conditions, when we are able to accurately predict the diffusion tensor for an arbitrary molecule.

The second dataset is made of three proteins for which we have experimental diffusion tensor data: HIV-1 protease; Maltose-binding protein; and Ubiquitin/UBA complex. We use this dataset to measure the accuracy of the algorithm under real experimental conditions and the inaccuracies inherent in ELM's prediction of the diffusion tensor.

[5]Due to technical issues with SURF [80, 79] we removed eight complexes from the original set of 84.

4.5.1 Docking Using Ideal Synthetic Data

We first demonstrate the feasibility of docking based solely on the diffusion tensor by docking the COMPLEX dataset based on the synthetic diffusion tensor. For each complex we generate the synthetic diffusion tensor $\mathbf{D}_{syn}$ using ELM. We then dock the complex using ELMDOCK-t, where we use $\mathbf{D}_{syn}$ instead of $\mathbf{D}_{exp}$. The detailed results for the docking algorithm using the synthetic diffusion tensor are presented in Figure 4.5.

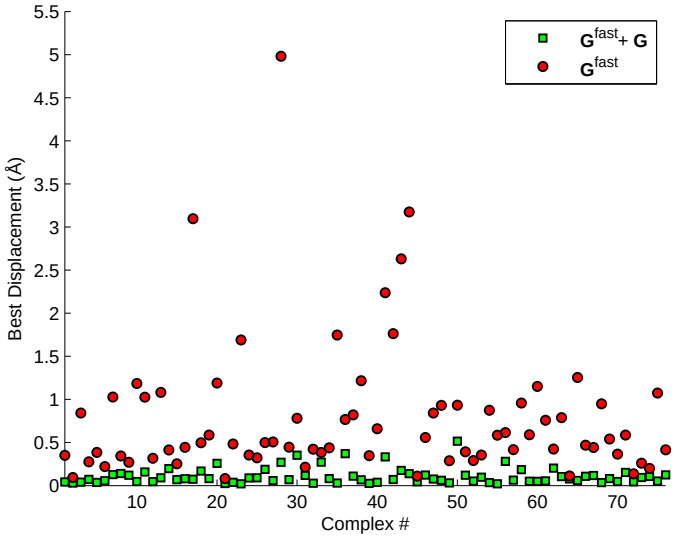

Figure 4.5: Docking results for the 76 complexes with no errors in ρ^{syn}. Circles denote results using $\mathbf{G}^{fast}$ approximation at each iteration, and the squares denote results of the full algorithm that uses $\mathbf{G}^{fast}$ and then $\mathbf{G}$.

From Figure 4.5 we can conclude that we are able to effectively dock two domains together given that we have a good prediction of the diffusion tensor. For most proteins just using the fast approximation $\mathbf{G}^{fast}$ yields a solution accurate to within 1.5Å. Further refinement using full computation of $\mathbf{G}$ yields a completely accurate solution. These results further support that it is possible, under ideal conditions,

to accurately assemble a molecular complex based solely on its diffusion tensor. In addition, we have shown that our approach of minimizing using a quadratic approximation (Newton's method), presented in equation (4.57), can be used to efficiently minimize χ^2_G.

4.5.2 Robustness of Diffusion Tensor Docking to Experimental Noise

In an experimental setting, ρ values usually have experimental error around $2-5\%$. To simulate the effects of these errors on the quality of the solution, we added normally distributed noise to ρ^{syn} with a standard deviation of 2.5% or 5%. Using the NH vectors of the complex and ρ^{syn} we computed the synthetic diffusion tensor $\mathbf{D}_{syn}$ using ROTDIF, and docked the complex. Figure 4.6 shows the docking results with the described errors in ρ^{syn}.

From Figure 4.6 we can see that in most cases, even with the errors in ρ^{exp} values, we are still able to converge a correct solution within 1Å. The large errors for some complexes are due to the fact that one domain is larger than the other; as a result the larger number of surface points in the larger domain makes the overall computation of the covariance matrix insensitive to small variations in the position of the smaller domain.

4.5.3 Application to Real Dual-Domain Systems

Finally, we test our method on two-domain complexes for which we have an overall experimental diffusion tensor: HIV-1 protease, *Structure 1bvg*; Maltose-

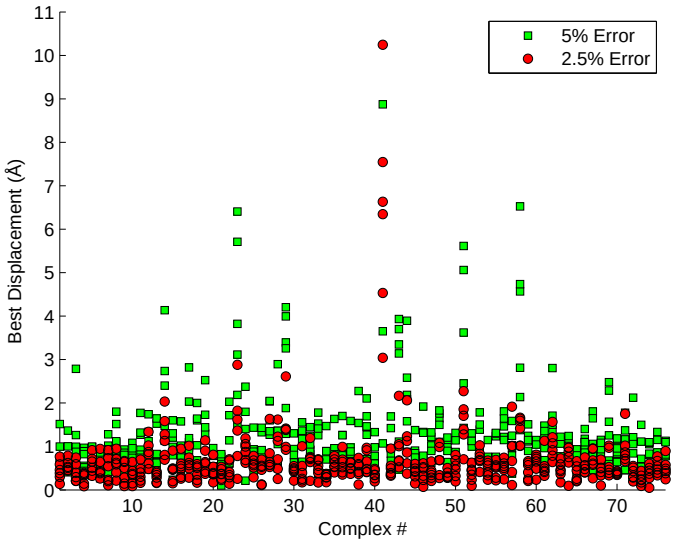

Figure 4.6: Docking results for the 76 complexes with 2.5% and 5% normal errors in ρ^{syn} that uses $\mathbf{G}^{fast}$ and then $\mathbf{G}$ is presented. Docking of each complex was performed six times, with individual errors in ρ_i^{exp} randomly selected from the normal distribution. For the purposes of visualization a few outliers for complex #41 are not shown. The large error in the solution for a few of the complexes is due to the significant difference in size of the two domains.

binding protein, *Structure 1ezp*; and Ubiquitin/UBA complex, *Structure 2jy6*. The cartoon representation of HIV-1 protease is shown in Figure 4.7A and Maltose-binding protein is shown in Figure 4.7B. For the Ubiquitin/UBA we have complete relaxation data and therefore use the complete method ELMDOCK, where the two domains are first aligned and then optimally translated relative to each other. Identical to Section 3.3.6, we create *Structures 2jy6-I and 2jy6-II*, the modified structures of 2jy6, to test the effect of the tails on our docking results. In our current implementation we did not recalculate the experimental diffusion tensor after alignment, but simply took the diffusion tensor of the Ubiquitin domain as the value for the overall experimental diffusion tensor of the complex. We compare our method to

the one proposed in Ryabov and Fushman [57]. Since in Ryabov and Fushman no initial guess was specified for the minimization, we will use the method derived in Section 4.4.3.1. The results for the three proteins are presented in Table 4.1.

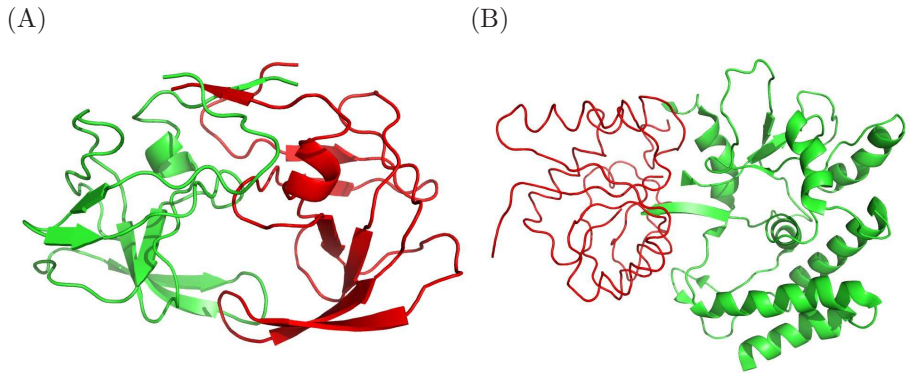

Figure 4.7: A cartoon representation of the HIV-1 protease and the Maltose-binding protein. (A) HIV-1 protease homodimer, with the first domain colored red and the scond domain green. (B) The first model of the Maltose-binding protein with the C domain colored in green and the N domain colored in red.

We see from Table 4.1 that ELMDOCK-t gives about 5Å error in displacement. The HIV-1 protease is a very rigid structure and as expected gives the best results. For a structure with large tails like the Ubiquitin/UBA complex the solution on average can change by around 2.4Å depending on which tail is chosen, therefore picking the right tail orientation, just like for the alignment tensor docking, is important. Removing the tails increased the $RMSD_2$ from 6.30Å to 10Å. This suggests that removing the tail might not be an effective strategy for the diffusion tensor although it was for the alignment tensor. The tail contributes to the overall tumbling of the individual molecule, and it is very plausible that its effect does not average out in the solution. The alignment of two domains based on their diffusion

Table 4.1: The results of diffusion-tensor-guided docking using ELMDOCK-t and ELMDOCK for the Ubiquitin/UBA Complex.

Struct.	Method[a]	RMSD[b]	$RMSD_2$[c]	Time[d]	Ryabov[e]	#[f]
1bvg	ELMDOCK-t	0.84	2.47	29	1275	2
1ezp	ELMDOCK-t	1.52	4.53	43	2010	2
2jy6-I	ELMDOCK-t	2.47[g] [1.02][h]	6.30[g] [2.39][h]	22[g] [2.31][h]	-	2[g]
2jy6-II	ELMDOCK-t	4.59	7.71	27	935	2
2jy6-II	ELMDOCK	4.10	9.61	84	4220	8

[a] The method that was used to dock the complex.

[b] The RMSD (in Å) between the original complex structure and the predicted complex. The structures are optimally rotated and centered using the center of mass [47].

[c] The RMSD (in Å) between the coordinates of atoms of the second domain for the original and predicted complex.

[d] The elapsed time (in seconds) required for docking.

[e] The elapsed time (in seconds) for the method proposed in Ryabov and Fushman [57], with the initial guess provided by the algorithm developed in Section 4.4.3.1.

[f] The number of possible solutions, all of which have a very similar overall alignment tensor.

[g] Values are the means of the individual values for the best solution of each of the 100 models.

[h] Values in the brackets are the standard deviations of the individual values for the best solution of each of the 100 models.

tensor does not significantly affect the $RMSD_2$ of the optimal solution, suggesting that just as in the case of the alignment tensor, it is not a significant contributor of error. Overall, ELMDOCK is about forty times faster than the method proposed in

Ryabov and Fushman [57].

4.6 Conclusion

In this chapter we presented an efficient minimization method for docking two-domain complexes based on their diffusion tensor. We first improved the performance of ROTDIF, a method for computing experimental diffusion tensor, by introducing a faster, non-stochastic, algorithm. We then combined the new ROTDIF algorithm with a novel two step minimization method that provides the first complete deterministic method for docking two-domain proteins based on the experimental NMR relaxation data and three-dimensional structure of the individual domains. This is the first method developed that gives a formula for quickly determining the initial guess for a convex minimization method. Given an initial guess, our method finds the solution about forty times faster than the method developed in Ryabov and Fushman [57] (which provides no method for determining the initial guess) and is significantly more computationally efficient than the simulated annealing method developed in Ryabov et al. [59] (which is not guaranteed to converge to the correct solution and has no clear stopping condition).

We show that we are able to correctly dock a large variety of two-domain proteins using a synthetic experimental diffusion tensor, with or without expected experimental errors. Using a real experimental diffusion tensor we are able to dock within about 5Å.

We foresee the same type of integration with other docking methods for ELM-

DOCK as for PATIDOCK. See Section 3.4 for the variety of ways that PATIDOCK can be combined with other docking methods. In particular, ELMDOCK can be used to produce an initial guess for a more expensive docking algorithm.

Chapter 5

Conclusion

In this thesis we have presented three main contributions in the field of protein structure determination. The first main contribution is an *ab initio* method called PATI for efficient prediction of the alignment tensor of a molecule. We developed formulas and methods for using numerical integration to reduce the dimensionality of the problem from four to two, improved the speed, and introduced a way to control the trade-off between speed and accuracy of the computation. Additionally, we introduced and developed the novel idea of using a convex hull instead of molecular shape to further reduce the complexity of the computation. We compared our method to three other methods and showed that our method is just as accurate or more accurate than other methods for prediction. We further analyzed the errors in all the prediction methods and showed that inaccurate prediction of orientation is the major cause of error in all the methods.

Building upon PATI, we introduced a novel idea for docking a two-domain complex based on its overall alignment tensor. This new docking method, PATIDOCK, uses the PATI method as one of its main components. We expanded on PATI by showing how it can be adapted to quickly recalculate the alignment tensor of a two-domain complex, where the second domain is experiencing translational motion. We then used this new method in a docking method PATIDOCK, which is

able to dock a two-domain molecule in seconds. Based on extensive benchmarking, we determine that we are able to dock two domains under heavy experimental error, assuming accurate prediction of the alignment tensor. Using real experimental data we expect to align the two domains and dock the two domains to within 4.3Å. To further improve the docking results we introduced a new method that combines the alignment tensor results with additional experimental data (in the form of CSPs).

Finally, similar to PATIDOCK, we developed a docking method called ELMDOCK based on the overall diffusion tensor of a molecule. Computational efficiency is achieved by separating the problem into two distinct steps and then approximating the covariance matrix. We analyzed and showed how robust ELMDOCK is to common experimental errors. Using real experimental data we expect to align and dock the two domains to within 4.3Å.

5.1 Future Work

Moving forward, we would like to integrate our methods into a more complete docking software package such as HADDOCK [25]. In PATIDOCK we started toward that goal by adding additional constraints like CSPs. However, we feel that it is better to integrate our energy function into an already established software package rather than to try to build a software package from the ground up.

Meanwhile some further improvements can be added to the current algorithms. Currently, in ELMDOCK we recalculate the surface of the molecule at each iteration of the optimization. If this method is integrated in a more general docking method

the current implementation of ELM might be too slow. Instead of recomputing ELM we could attempt to readjust the value of the covariance matrix by adjusting only the affected surface points.

Fundamentally, the accuracy of our docking methods is limited by our prediction methods, PATI and ELM. We would like to see if it is possible to further improve the accuracy of these methods by basing them on more complicated physical models. Specifically in the case of ELM, we would like to move away from the ellipsoidal approximation of a molecule, to a model that uses the shape of the molecule directly.

Appendix A

Minimization

Here we describe basic algorithms for solving minimization problems. For further review on minimization see Nash and Sofer [51].

Minimization is a process for finding the minimum value of a function.

Definition A.1 (Global Minimizer)**.** $\mathbf{x}^* \in \mathbb{R}^n$ *is the global minimizer of the function* $f(\mathbf{x}) \in \mathbb{R}$ *if*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in \Psi,$$

where $\Psi \subseteq \mathbb{R}^n$ *is the region on which* f *is defined. We say that* $f(\mathbf{x}^*)$ *is the global minimum (minimum value) of* f*.*

Local minimization is a process for finding the minimum value in a neighborhood of the domain of a function.

Definition A.2 (Local Minimizer)**.** $\mathbf{x}^*_{loc} \in \mathbb{R}^n$ *is a local minimizer of the function* $f(\mathbf{x}) \in \mathbb{R}$ *if there exists an* ϵ *such that*

$$f(\mathbf{x}^*_{loc}) \leq f(\mathbf{x}),$$

when $\|\mathbf{x}^*_{loc} - \mathbf{x}\| < \epsilon$*. We say that* $f(\mathbf{x}^*_{loc})$ *is the local minimum (local minimum value) of* f*.*

We observe that if the function is strictly convex then it only has one local minimum that is also the global minimum.

A.1 General Local Minimization

The basic principle behind most local minimization method is to continue stepping to lower function values until one hits a local minimum. The key to finding a lower value is to figure out in what direction the function decreases (a descent direction) and to make sure your step size in that direction is large enough to decrease the function value $f(\mathbf{x})$, but not too large that it would jump over $\mathbf{x}^*_{loc}$. Algorithm A.1 outlines the general minimization algorithm.

Algorithm A.1 Calculating the local minimum $\mathbf{x}^*_{loc}$

1: $k \leftarrow 0$

2: $\mathbf{x}_k \leftarrow$ initial guess for $\mathbf{x}^*_{loc}$

3: **while** Stopping condition not reached **do**

4: Compute a descent direction $\mathbf{p} \in \mathbb{R}^n$

5: Compute a step size $\alpha_k \in \mathbb{R}$

6: Set $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}$

7: Set $k \leftarrow k + 1$

8: **end while**

9: **return** $\mathbf{x}_k$

A.1.1 Newton Step

Because $f(\mathbf{x})$ is complicated, it might be difficult to directly solve for the optimal descent direction and step size. Therefore we approximate $f(\mathbf{x})$ by a quadratic

Taylor series expansion around our current value $\mathbf{x}_k$;

$$f(\mathbf{x}_k + \mathbf{p}) \approx f(\mathbf{x}_k) + \mathbf{p}^T \mathbf{g}(\mathbf{x}_k) + \frac{1}{2}\mathbf{p}^T \mathbf{H}(\mathbf{x}_k)\mathbf{p} = \tilde{f}(\mathbf{x}_k + \mathbf{p}), \tag{A.1}$$

where $\mathbf{g}(\mathbf{x})$ is the gradient of $\tilde{f}(\mathbf{x})$, and $\mathbf{H}(\mathbf{x})$ is the Hessian of $f(\mathbf{x})$.

$\tilde{f}(\mathbf{x}_k + \mathbf{p})$ is a fairly accurate approximation to $f(\mathbf{x}_k + \mathbf{p})$ if $\|\mathbf{p}\|$ is small, and can be minimized by setting the gradient of $f(\mathbf{x}_k + \mathbf{p})$ equal to 0:

$$\nabla \tilde{f}(\mathbf{x}_k + \mathbf{p}) = \mathbf{g}(\mathbf{x}_k) + \mathbf{H}(\mathbf{x}_k)\mathbf{p} = \mathbf{0}. \tag{A.2}$$

We then solve directly for the minimizing solution:

$$\mathbf{p} = -\mathbf{H}(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k). \tag{A.3}$$

This is known as the *Newton step*, and is the basis for most minimization methods where the gradient and the Hessian are known or can be estimated.

A.1.2 Step Length

Since the Newton step $\mathbf{p}$ is based on a quadratic approximation of $f(\mathbf{x})$ it will be a bad estimate unless the step size remains small. The simplest way to keep the step size small is to perform a line search. In a line search, the Newton step is probed along $\mathbf{p}$ in order to ensure that one does not overstep the minimum value.

An alternative method is to constrain the length of $\mathbf{p}$ while computing the minimizer of $\tilde{f}(\mathbf{x}_k + \mathbf{p})$, thus keeping $\mathbf{p}$ small enough so that $\tilde{f}(\mathbf{x}_k + \mathbf{p}) \approx f(\mathbf{x}_k + \mathbf{p})$. This is referred to as a creation of a trust region in which $\tilde{f}(\mathbf{x}_k + \mathbf{p})$ is believed to be a good estimate of $f(\mathbf{x}_k + \mathbf{p})$.

A.1.3 Alternatives to Newton's Method

In cases when the function is complicated, it is not feasible to compute its gradient and Hessian. However in some cases, it is possible to calculate a good estimate of the function with a simpler function whose gradient and Hessian can be computed. This estimated gradient and Hessian can be used to compute an approximation to the actual Newton step. We will use this idea to construct an approximation to our energy function, from which we will obtain a step in our minimization of the original function.

A.2 Global Minimization

Following the direction of descent with appropriate choice of step length will lead to a local minimum value. However this value is not necessarily the global minimum if the function is not convex. Since most energy functions, including the energy function we will use, are not convex we need a method to find the global minimizer of a non-convex function.

While finding a global minimum of a general $f(x)$ is an area of open research, the problem can sometimes be efficiently solved for a specific $f(x)$ given some insight into how it behaves. If the approximate locations of the local minimizers can be determined, one can solve the global optimization problem. The search space is split into regions, such that it becomes feasible to find the local minimizer in each region. The smallest of the local minimizers then becomes the global minimizer. Algorithm A.2 presents an outline of this global minimization method.

Algorithm A.2 Global Minimization

1: Choose $\mathbf{x}^* \in \Psi$ arbitrarily.

2: Split $\mathbf{\Psi}$ into m regions, $\{\Psi_1, \cdots, \Psi_m\}$.

3: **for** $i = 1$ to $i = m$ **do**

4: $\quad \hat{\mathbf{x}}_i^* \leftarrow \arg\min_{x \in \Psi_i} f(\mathbf{x})$

5: $\quad$ **if** $f(\hat{\mathbf{x}}_i^*) < f(\mathbf{x}_i^*)$ **then**

6: $\quad\quad \mathbf{x}^* \leftarrow \hat{\mathbf{x}}_i^*$

7: $\quad$ **end if**

8: **end for**

9: **return** $\mathbf{x}^*$

A.3 Least Squares Problems

Least squares is a specific type of local minimization problem where $\mathbf{x}_{loc}^*$ is a value that minimizes the sum of squares of a set of functions $\mathbf{f}(\mathbf{x})$. The global minimizer is usually close to 0, since the parameters $\mathbf{x}$ determine the fit of a model to data, and the fit is poor unless $f(\mathbf{x}_{loc}^*)$ is small.

Definition A.3 (Least squares)**.** *Given $f_i(\mathbf{x}) \in \mathbb{R}$ for $i = 1, \ldots, m$, least squares is a process for finding $\mathbf{x}_{loc}^* \in \Psi$ such that*

$$\mathbf{x}_{loc}^* = \arg\min_{x \in \Psi} \sum_{i=1}^{m} f_i(\mathbf{x})^2. \tag{A.4}$$

Observe that this problem can be restated as

$$\mathbf{x}_{loc}^* = \arg\min_{x \in \Psi} \|\mathbf{f}(\mathbf{x})\|_2. \tag{A.5}$$

A.3.1 Linear Least Squares

If $\mathbf{f}(\mathbf{x})$ is a linear function such that

$$\begin{aligned} f_1(\mathbf{x}) &= x_1\phi_1(t_1) + \ldots + x_i\phi_i(t_1) + \ldots + x_n\phi_n(t_1) - b_1, \\ &\ldots \\ f_m(\mathbf{x}) &= x_1\phi_1(t_m) + \ldots + x_i\phi_i(t_m) + \ldots + x_n\phi_n(t_m) - b_m. \end{aligned} \tag{A.6}$$

then the least squares problem can be stated as

$$\mathbf{x}^*_{loc} = \arg\min_{\mathbf{x}\in\Psi} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2. \tag{A.7}$$

where

$$\mathbf{A} = \begin{bmatrix} \phi_1(t_1) & \ldots & \phi_n(t_1) \\ \vdots & & \vdots \\ \phi_1(t_m) & \ldots & \phi_n(t_m) \end{bmatrix}, \tag{A.8}$$

and $\mathbf{x}^*_{loc}$ is also the global solution $\mathbf{x}^*$. For the purposes of our thesis, the rank of $\mathbf{A}$ is assumed to be n.

Let

$$\mathbf{A} = \mathbf{Q}\mathbf{R} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \tag{A.9}$$

be the QR decomposition of $\mathbf{A}$, where $\mathbf{Q}_1$ is $m \times n$ matrix with orthonormal columns, $\mathbf{Q}_2$ is $m \times (m-n)$ matrix with orthonormal columns, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, and $\mathbf{R}_1$ is an upper triangular $n \times n$ matrix. Then

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 = \|\mathbf{Q}_1^T\mathbf{b} - \mathbf{R}_1\mathbf{x}\|_2 + \|\mathbf{Q}_2^T\mathbf{b}\|_2. \tag{A.10}$$

Since only the first norm is dependent on $\mathbf{x}$, we now present a simple algorithm to solve linear least squares using QR decomposition.

Algorithm A.3 QR Linear Least Squares

1: Compute QR decomposition $\mathbf{A} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \\ \mathbf{0} \end{bmatrix}$

2: Solve $\mathbf{R}_1\mathbf{x} = \mathbf{Q}_1^T\mathbf{b}$ for $\mathbf{x}$ using back-substitution

Alternatively, the SVD decomposition can be used to solve linear least squares, which is slower than QR but more numerically stable.[52]

A.3.2 Nonlinear Least Squares

In the case when $\mathbf{f}(\mathbf{x})$ is nonlinear, the problem is a nonlinear least squares problem. When solving a nonlinear least squares problem the Hessian matrix $\mathbf{H}$ can often be efficiently approximated, and a good nonlinear least squares solver will take advantage of that. In the most popular least squares algorithm, Levenberg-Marquardt, the Hessian is approximated by

$$\mathbf{H}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x}) + \lambda\mathbf{I}, \tag{A.11}$$

where λ is a damping factor that is adjusted at each iteration of the algorithm, $\mathbf{I}$ is the identity matrix, and $\mathbf{J}(\mathbf{x})$ is the Jacobian of $\mathbf{f}(\mathbf{x})$:

$$J(\mathbf{x})_{ij} = \frac{\partial f_i(\mathbf{x})}{\partial x_j}. \tag{A.12}$$

In cases when $\mathbf{J}$ is too difficult to analytically compute it can be approximated using finite differences.

We define the syntax "$\mathbf{x}^* = \text{lsqnonlin}(\text{fun}(\mathbf{x}), \mathbf{x}_0)$" as a function that returns the local minimizer of the function "fun(x)".

Appendix B

Numerical Integration

Here we present some techniques for numerically integrating a function. We provide only a cursory overview of the topic. For more in depth discussion see [76].

Let $f(x)$ be a smooth function on the interval $[a, b]$. Then the integral of $f(x)$ can be approximated using Simpson's rule such that

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right]. \tag{B.1}$$

Furthermore, if $c \in (a, b)$ then we expect

$$\begin{aligned}\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx \approx &\frac{c-a}{6}\left[f(a) + 4f(\frac{a+c}{2}) + f(c)\right] \\ &+ \frac{b-c}{6}\left[f(c) + 4f(\frac{c+b}{2}) + f(b)\right]\end{aligned} \tag{B.2}$$

to be even a better approximation of the integral. The intervals $[a, c]$ and $[c, b]$ can be recursively subdivided further until we get an approximation that is good enough.

Note that Simpson's rule is just one of many possible ways of approximating the integral. Depending on the type of function that is being integrated other approximations could work better.

B.1 Adaptive Integration

Adaptive Integration is a method for efficient recursive subdivision of the integration region. Since integrals over regions in which f is well-approximated by

a quadratic polynomial are better approximated by Simpson's rule than regions in which its behavior is highly nonlinear, it is more efficient to approximate smooth regions using a smaller number of function evaluations. The recursive subdivision method for integration that does this is referred to as adaptive integration. A basic adaptive integration algorithm is presented in Algorithm B.1.

Algorithm B.1 adaptivelyIntegrate

Input: $f(x)$- a smooth function, $[a, b]$- an interval on which $f(x)$ is integrated. ϵ- the bound on the absolute error in the result of the integration.

Output: Q, the value of the integral.

1: $c \leftarrow \frac{a-b}{2}$

2: $Q \leftarrow \frac{b-a}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right]$

3: $Q_1 \leftarrow \frac{c-a}{6}\left[f(a) + 4f(\frac{a+c}{2}) + f(c)\right]$

4: $Q_2 \leftarrow \frac{b-c}{6}\left[f(c) + 4f(\frac{c+b}{2}) + f(b)\right]$

5: **if** $|Q - Q_1 - Q_2| < \epsilon$ **then**

6: **return** $Q_1 + Q_2$

7: **else**

8: $Q_1 \leftarrow$ adaptivelyIntegrate$(f, [a, c], \epsilon/2)$

9: $Q_2 \leftarrow$ adaptivelyIntegrate$(f, [c, b], \epsilon/2)$

10: **return** $Q_1 + Q_2$

11: **end if**

The advantage of adaptive integration is that we can tune the desired error tolerance of the integration vs. number of function evaluations, and efficiently distribute the function evaluations on the interval such that it gives the best accuracy

for the number of evaluations.

B.2 Improper Numerical Integration

In one part of ELMDOCK we have to numerically integrate an improper integral such as

$$\int_0^\infty f(x)dx. \tag{B.3}$$

We can numerically evaluate this integral by performing a change of variable $u = \frac{1}{1+x}$. Our integration problem now becomes

$$\int_0^1 f(\frac{1-u}{u})\frac{1}{u^2}du, \tag{B.4}$$

which can be numerically integrated using adaptive integration.

Appendix C

Covariance of an Ellipsoid

In this section we derive a formula for the covariance matrix of an ellipsoid. We use the formula for the covariance matrix of an ellipsoid to derive a method for finding an equivalent ellipsoid representation of a molecule in Section 1.3.2.3.

Let $\mathbf{X}$ be a three-dimensional random variable; then the covariance matrix $\mathbf{C}$ for $\mathbf{X}$ is defined as

$$\mathbf{C} = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \mathrm{Cov}(X_2, X_3) \\ \mathrm{Cov}(X_3, X_1) & \mathrm{Cov}(X_3, X_2) & \mathrm{Cov}(X_3, X_3) \end{bmatrix}, \tag{C.1}$$

where

$$\begin{aligned} \mathrm{Cov}(X_i, X_j) &= E((X_i - E(X_i))(X_j - E(X_j))) \\ &= E(X_i \cdot X_j) - E(X_i)E(X_j), \end{aligned} \tag{C.2}$$

and

$$\mathrm{Cov}(X_i, X_i) = \mathrm{Var}(X_i) = E(X_i^2) - E(X_i)^2. \tag{C.3}$$

We now derive the formula for the covariance matrix of an ellipsoid using integration.

Theorem C.1. *Given the ellipsoid $\mathcal{E}(\mathbf{A}, \mathbf{c})$, and the sorted eigendecomposition*

$$\mathbf{A} = \mathbf{V} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{V}^T,$$

the covariance matrix of $\mathcal{E}$ is

$$\mathbf{C}_{\mathcal{E}} = \mathbf{V} \begin{bmatrix} 1/(3\lambda_1) & 0 & 0 \\ 0 & 1/(3\lambda_2) & 0 \\ 0 & 0 & 1/(3\lambda_3) \end{bmatrix} \mathbf{V}^T.$$

Proof. Let $\mathcal{E}(\mathbf{A}, \mathbf{c})$ be an ellipsoid and

$$\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^T = \mathbf{V} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \mathbf{V}^T.$$

Since the covariance matrix is independent of the position of the ellipsoid we let $\mathbf{c} = \mathbf{0}$.

Changing into the coordinate space of $\mathbf{A}$, the ellipsoid can be rewritten using the Cartesian axes x, y, z:

$$\lambda_1 x^2 + \lambda_2 y^2 + \lambda_3 z^2 = 1. \tag{C.4}$$

Due to the symmetry of the ellipsoid along the coordinate space axes, $\mathrm{Cov}(\mathcal{E}_i, \mathcal{E}_j) = 0$ when $i \neq j$. Therefore the covariance matrix of $\mathcal{E}$ is

$$\begin{bmatrix} \mathrm{Var}(\mathcal{E}_x) & 0 & 0 \\ 0 & \mathrm{Var}(\mathcal{E}_y) & 0 \\ 0 & 0 & \mathrm{Var}(\mathcal{E}_z) \end{bmatrix}, \tag{C.5}$$

where $\mathrm{Var}(\mathcal{E}_x)$ is the variance along the x axis, $\mathrm{Var}(\mathcal{E}_y)$ is the variance along the y axis, and $\mathrm{Var}(\mathcal{E}_z)$ is the variance along the z axis.

The variance along the z axis is

$$\mathrm{Var}(\mathcal{E}_z) = \frac{\int_{\mathcal{E}} z^2 \, dx \, dy \, dz}{\int_{\mathcal{E}} dx \, dy \, dz}. \tag{C.6}$$

Performing the change of variables into

$$\begin{aligned} x &= \frac{1}{\sqrt{\lambda_1}} \sin\phi \cos\theta, \\ y &= \frac{1}{\sqrt{\lambda_2}} \sin\phi \sin\theta, \\ z &= \frac{1}{\sqrt{\lambda_3}} \cos\phi, \end{aligned} \tag{C.7}$$

where θ is the azimuthal angle, ϕ is the polar angle, and the Jacobian determinant is

$$|\mathbf{J}| = (\lambda_1\lambda_2\lambda_3)^{-1/2} \sin\phi, \tag{C.8}$$

we get

$$\begin{aligned} \mathrm{Var}(\mathcal{E}_z) &= \frac{\int_0^{2\pi}\int_0^{\pi} (\lambda_3^{-1/2}\cos\phi)^2 \, |\mathbf{J}| \; d\phi \, d\theta}{\int_0^{2\pi}\int_0^{\pi} |J| \; d\phi \, d\theta} \\ &= \frac{4/3\lambda_3^{-1}\pi(\lambda_1\lambda_2\lambda_3)^{-1/2}}{4\pi(\lambda_1\lambda_2\lambda_3)^{-1/2}} \\ &= \frac{1}{3\lambda_3}. \end{aligned} \tag{C.9}$$

Similarly, $\mathrm{Var}(\mathcal{E}_x) = 1/(3\lambda_1)$, and $Var(\mathcal{E}_y) = 1/(3\lambda_2)$.

Changing back to the original coordinate system,

$$\mathbf{C}_{\mathcal{E}} = \mathbf{V} \begin{bmatrix} 1/(3\lambda_1) & 0 & 0 \\ 0 & 1/(3\lambda_2) & 0 \\ 0 & 0 & 1/(3\lambda_3) \end{bmatrix} \mathbf{V}^T. \tag{C.10}$$

□

Appendix D

PATI: Supplementary Information

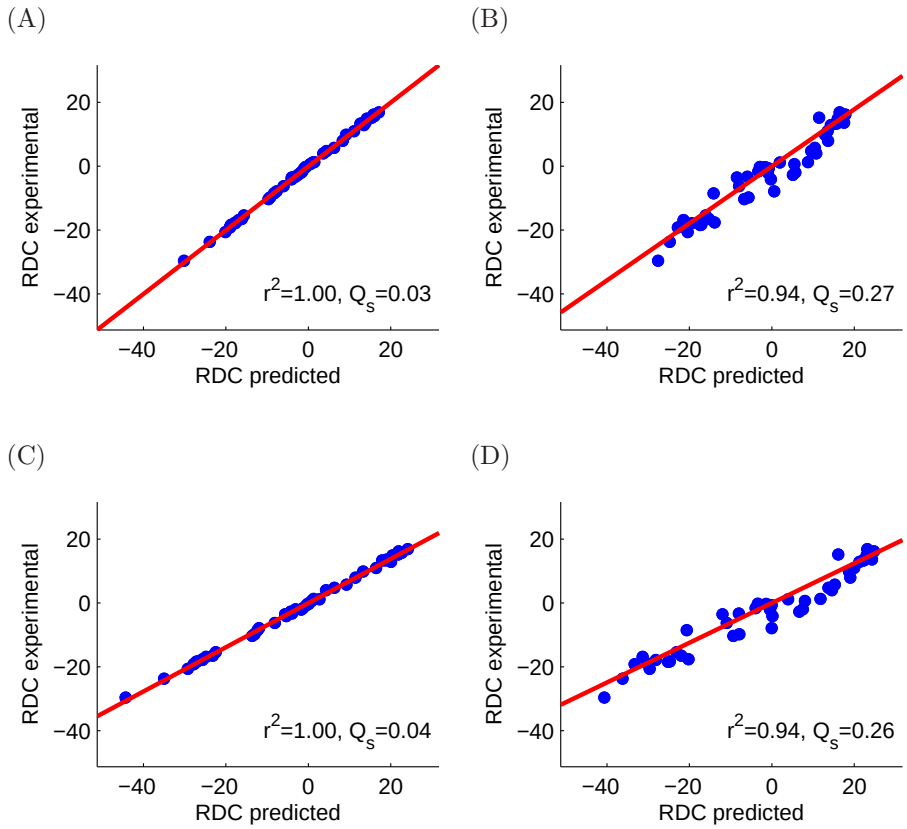

Figure D.1: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the Cellular factor BAF, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

Table D.1: Unscaled Quality Factors and the Scaling for *ab initio* Methods

PDB[a]	PATI[b,c]	PALES[b,c]	PATI-E[b,c,d]
2ezx	0.64 (0.62)	0.70 (0.60)	1.55 (0.39)
3gb1	0.21 (1.19)	0.14 (1.08)	0.29 (1.11)
2oed	0.25 (1.09)	0.21 (1.10)	0.35 (0.77)
1b4c	0.47 (1.72)	0.48 (1.77)	0.65 (2.20)
2ezm	0.47 (0.90)	0.48 (0.90)	0.69 (0.65)
1cmz	0.32 (0.98)	0.30 (1.05)	0.38 (0.96)
1d3z	0.32 (0.80)	0.33 (0.81)	0.53 (0.71)
1e8l[e]	0.47 (1.60)	0.46 (1.55)	0.47 (1.33)
1yjj[e]	0.66 (1.87)	0.69 (1.73)	0.61 (1.43)
Mean	0.42 (1.20)	0.42 (1.18)	0.61 (1.06)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the (unscaled) quality factor Q between the predicted and experimental data.

[c] MVE was used.

[d] Values in the parentheses represent the scaling constant ρ defined in Equation (2.30).

[e] The experimental values were multiplied by -1 to make the sign of experimental data consistent.

Table D.2: Quality of RDC Prediction for *ab initio* Methods using GE Model

PDB[a]	PATI-E[b,c]	Almond[b,c]	PROLFIT[b,c]
2ezx	0.13 (0.99)	0.13 (0.99)	0.14 (0.98)
3gb1	0.21 (0.99)	0.21 (0.99)	0.28 (0.98)
2oed	0.32 (0.97)	0.31 (0.96)	0.41 (0.96)
1b4c	0.56 (0.77)	0.56 (0.78)	0.99 (0.09)
2ezm	0.54 (0.54)	0.55 (0.52)	0.49 (0.61)
1cmz	0.30 (0.91)	0.31 (0.90)	0.28 (0.93)
1d3z	0.48 (0.72)	0.49 (0.71)	0.42 (0.70)
1e8l	0.28 (0.92)	0.28 (0.92)	0.30 (0.91)
1yjj	0.87 (0.29)	0.87 (0.29)	0.94 (0.23)
Mean	0.41 (0.79)	0.41 (0.78)	0.47 (0.71)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the scaled quality factor Q_s between the predicted and experimental data.

[c] Values in the parentheses represent squared Pearson's correlation coefficient (r^2).

Table D.3: Quality of RDC Prediction for *ab initio* Methods using PCAE Model

PDB[a]	PATI-E[b,c]	Almond[b,c]	PROLFIT[b,c]
2ezx	0.14 (0.98)	0.15 (0.98)	0.15 (0.98)
3gb1	0.11 (0.99)	0.10 (0.99)	0.23 (0.98)
2oed	0.26 (0.98)	0.25 (0.98)	0.35 (0.97)
1b4c	0.45 (0.82)	0.45 (0.83)	0.84 (0.26)
2ezm	0.50 (0.60)	0.51 (0.58)	0.45 (0.67)
1cmz	0.33 (0.89)	0.34 (0.88)	0.33 (0.90)
1d3z	0.34 (0.83)	0.36 (0.82)	0.31 (0.80)
1e8l	0.28 (0.92)	0.28 (0.92)	0.30 (0.92)
1yjj	0.86 (0.30)	0.86 (0.30)	0.96 (0.19)
Mean	0.37 (0.81)	0.37 (0.81)	0.43 (0.74)

[a] The RCSB Protein Data Bank code for protein coordinates. First model from the ensemble of NMR structures was used for the calculations. See Table 2.1 for the names of the proteins.

[b] Values represent the scaled quality factor Q_s between the predicted and experimental data.

[c] Values in the parentheses represent squared Pearson's correlation coefficient (r^2).

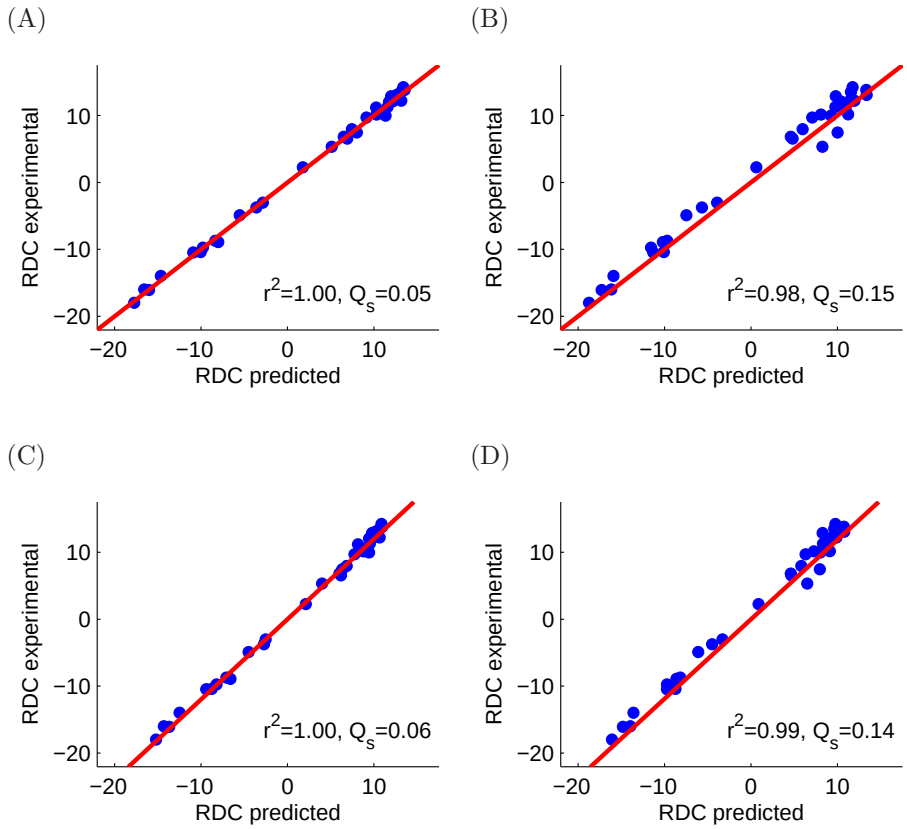

Figure D.2: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the B1 domain of protein G, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

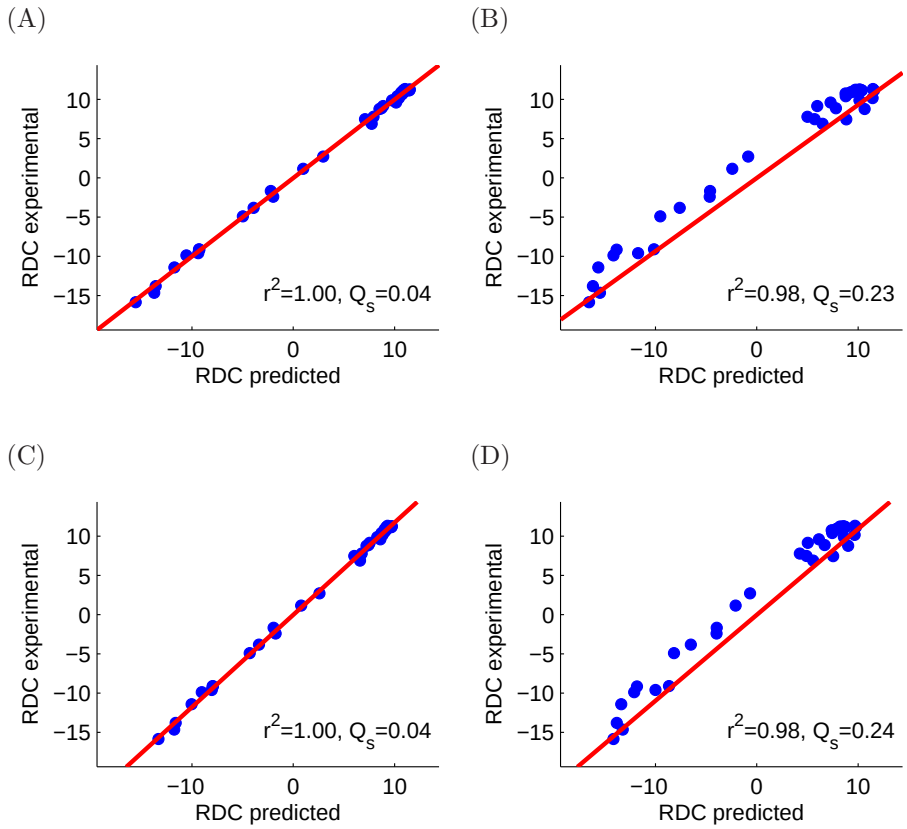

Figure D.3: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the B3 domain of protein G, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

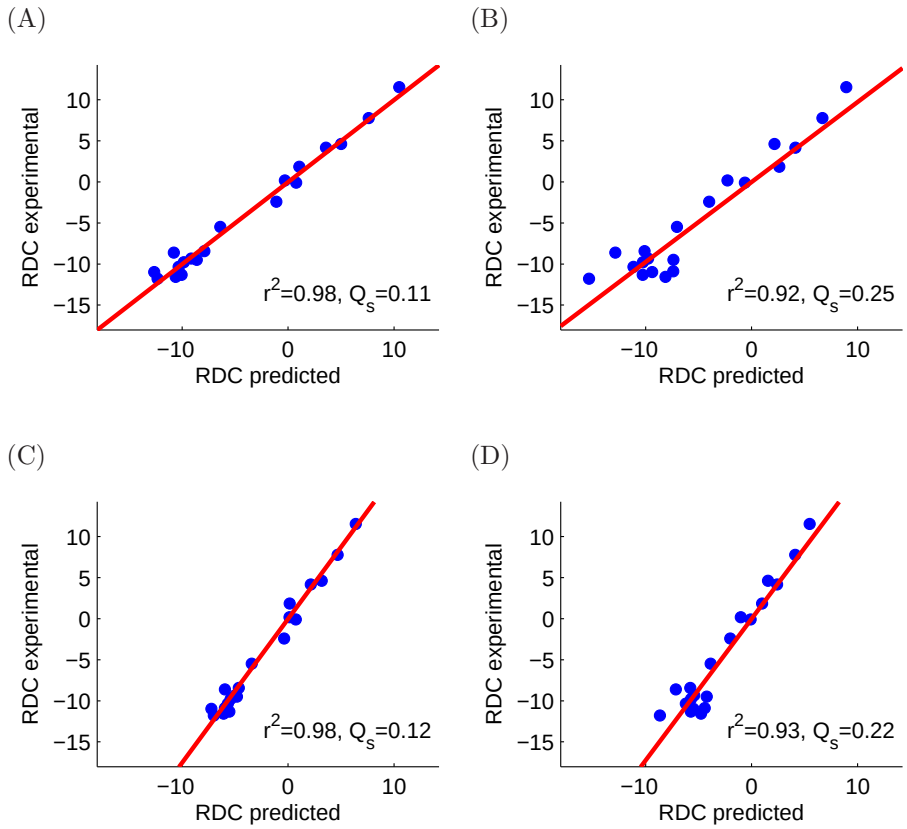

Figure D.4: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the rat apo-S100B protein, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

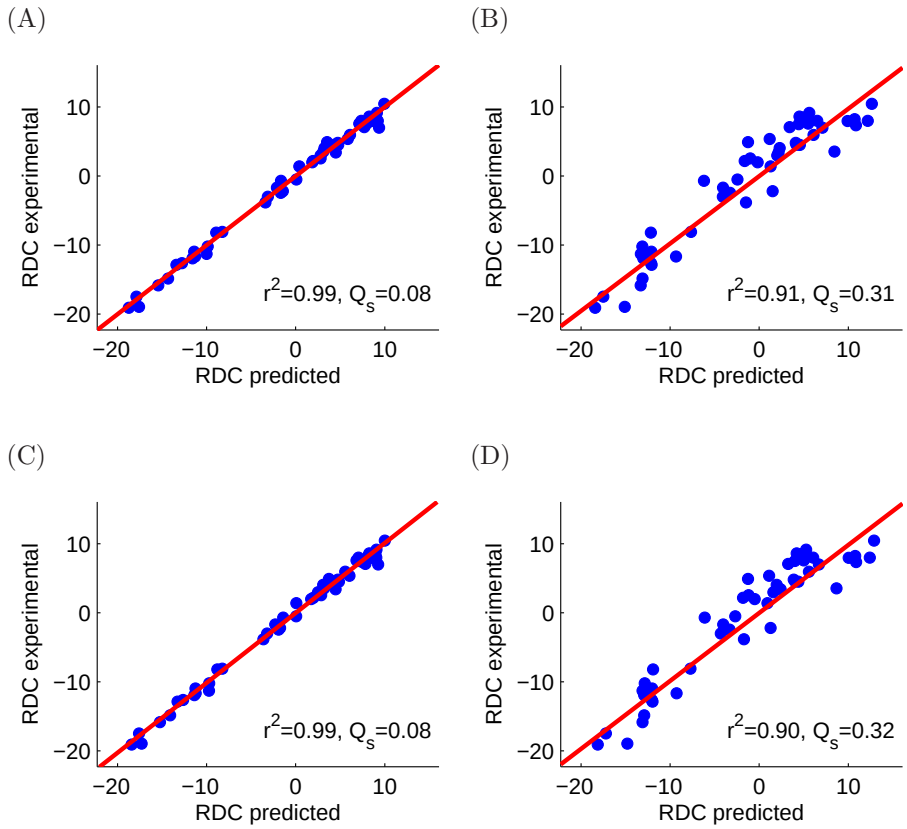

Figure D.5: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the G_{α} interacting protein, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

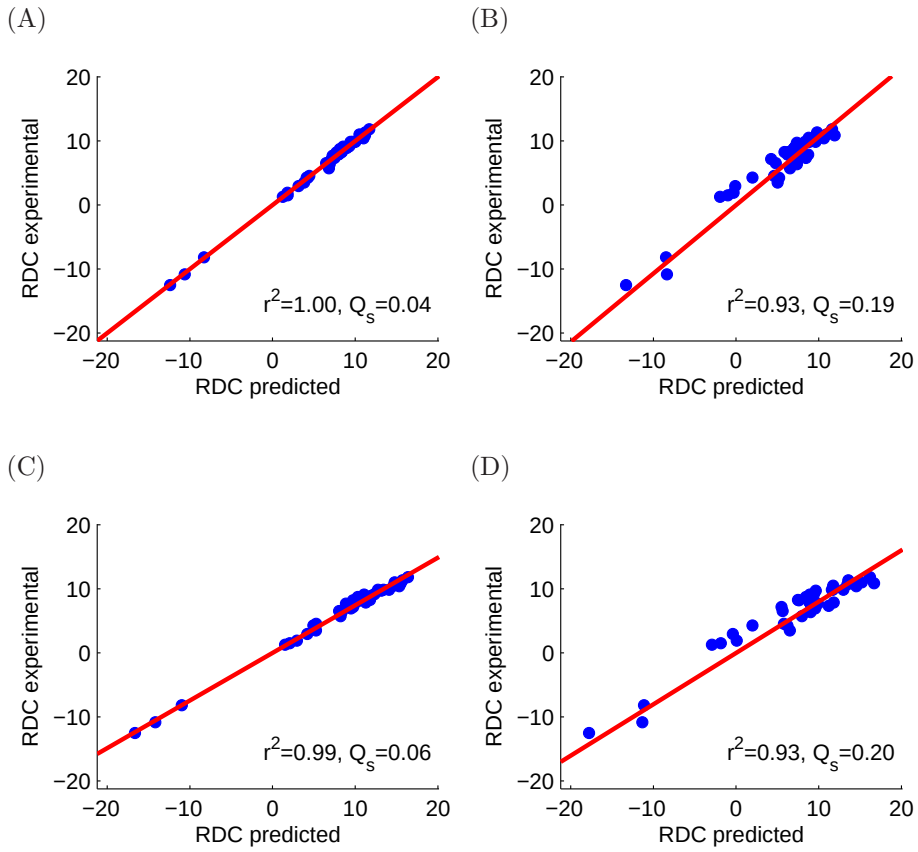

Figure D.6: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in Ubiquitin, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated.

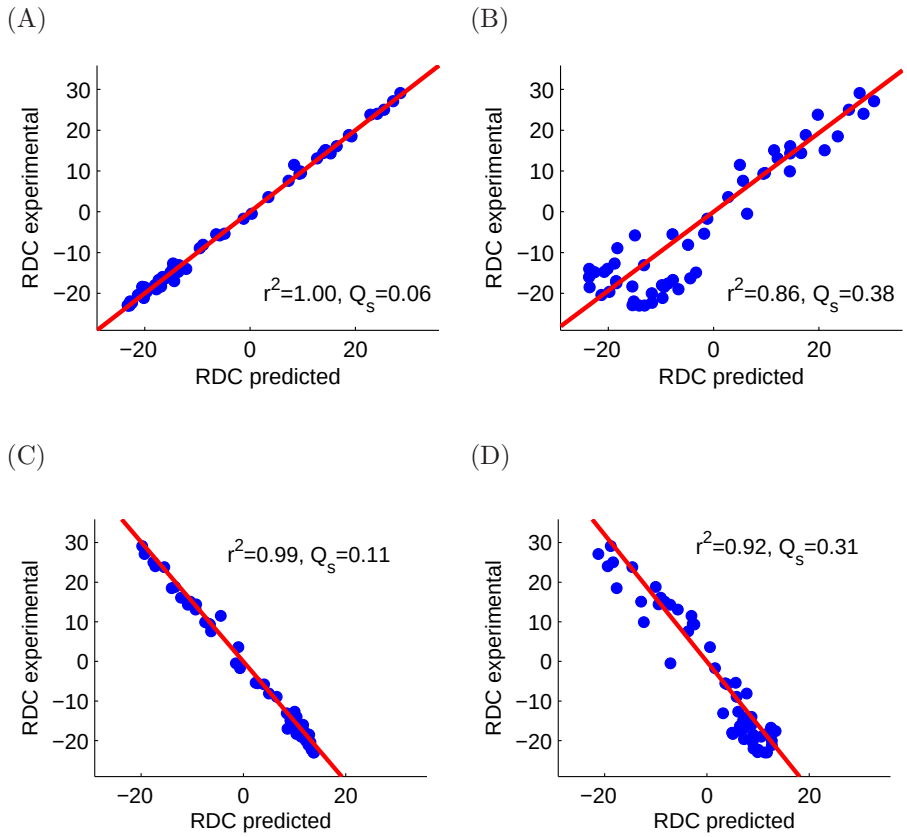

Figure D.7: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in hen Lysozyme, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated. The negative slope in panels (C) and (D) reflects the fact that the reported experimental RDCs and the corresponding predicted values are of opposite sign.

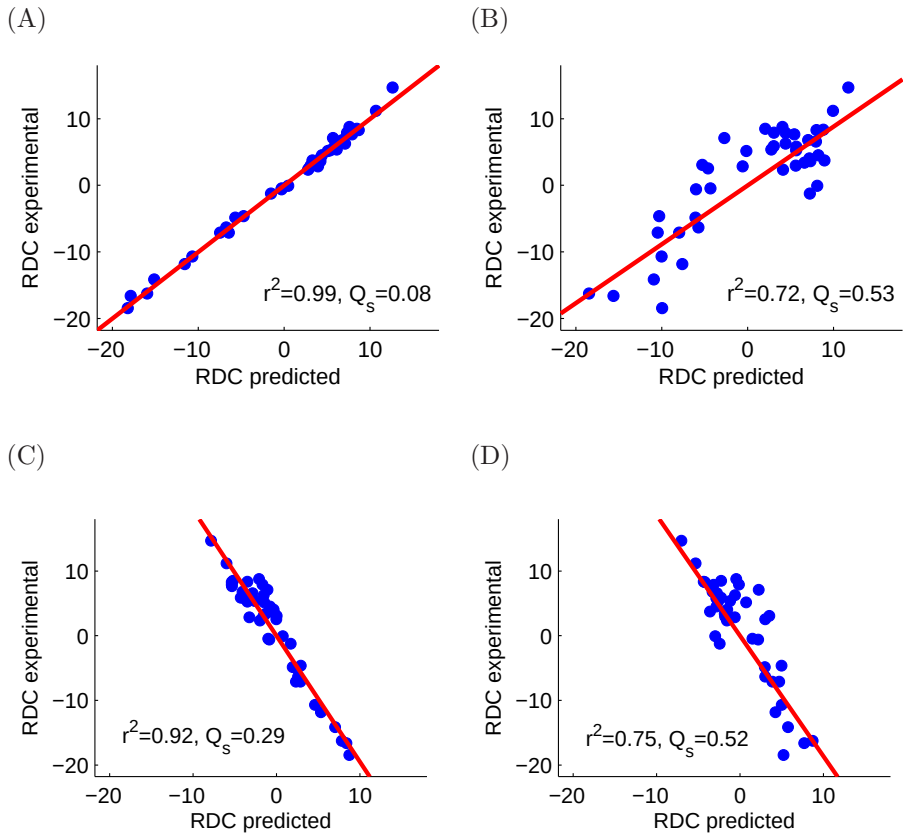

Figure D.8: Comparison of the predicted vs. experimental $^{1}H^{15}N$ RDC values for the backbone amides in the oxidized Putidaredoxin, using various versions of the molecular alignment tensor derived from PATI. (A) The experimental alignment tensor was derived directly from the experimental data using least squares. (B) The alignment tensor was constructed using the magnitude (eigenvalues) of the experimental alignment tensor and the tensor orientation predicted using PATI program. (C) The alignment tensor was constructed using the orientation (eigenvectors) of the experimental alignment tensor but PATI-predicted magnitude (eigenvalues) of the tensor. (D) The alignment tensor was fully predicted from PATI simulation. The values of the squared Pearson's correlation coefficient, r^2, and the scale-insensitive quality factor, Q_s, are indicated. The negative slope in panels (C) and (D) reflects the fact that the reported experimental RDCs and the corresponding predicted values are of opposite sign.

Appendix E

PATIDOCK: Supplementary Information

The complete synthetic RDC results for the COMPLEX dataset for 0Hz, 1Hz, and 3Hz errors are presented in Table E.1.

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs.

Name[a]	#[b]	0 Hz[c]	1 Hz[c,d]	3 Hz[c,d]
1A2K	1	0.02	0.09	0.28
1ACB	2	0.05	0.10	0.28
1AHW	3	0.03	0.11	0.17
1AK4	4	0.05	0.09	0.29
1AKJ	5	0.01	0.07	0.29
1ATN	6	0.00	0.06	0.19
1AVX	7	0.07	0.11	0.30
1AY7	8	0.07	0.16	0.37
1B6C	9	0.05	0.11	0.27
1BGX	10	0.02	0.06	0.09

[a] The first 4 letters of the file name. Bound versions of the domains were used in docking.

[b] Index of the complex.

[c] Best Displacement (in Å), computed as the smallest Euclidean norm between all the computed translations (solutions) and the known correct translation. The values in brackets represent the RMSD (in Hz) between the synthetic RDCs and the predicted RDCs at the solution. The column labels represent the size of the standard deviation of the normally distributed noise added to synthetic RDCs. "0 Hz" corresponds to no noise added to synthetic RDCs.

[d] The values represent an average of six independent runs.

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs. (continued)

Name	#	0 Hz	1 Hz	3 Hz
1BJ1	11	0.02	0.08	0.31
1BUH	12	0.03	0.09	0.22
1BVK	13	0.03	0.12	0.37
1BVN	14	0.03	0.09	0.21
1CGI	15	0.03	0.12	0.34
1D6R	16	0.02	0.10	0.29
1DE4	17	0.01	0.06	0.14
1DFJ	18	0.09	0.12	0.19
1DQJ	19	0.05	0.12	0.24
1E6E	20	0.02	0.18	0.42
1E6J	21	0.05	0.09	0.20
1E96	22	0.02	0.06	0.26
1EAW	23	0.07	0.13	0.33
1EER	24	0.21	0.23	0.89
1EWY	25	0.07	0.16	0.41

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs. (continued)

Name	#	0 Hz	1 Hz	3 Hz
1EZU	26	0.05	0.11	0.20
1F34	27	0.01	0.10	0.29
1F51	28	0.01	0.11	0.34
1FAK	29	0.03	0.12	0.34
1FC2	30	0.14	0.16	0.35
1FQ1	31	0.02	0.08	0.34
1FQJ	32	0.02	0.10	0.23
1FSK	33	0.01	0.11	0.20
1GCQ	34	0.05	0.13	0.59
1GHQ	35	0.03	0.07	0.21
1GP2	36	0.03	0.07	0.18
1GRN	37	0.05	0.14	0.35
1H1V	38	0.02	0.06	0.26
1HE1	39	0.02	0.11	0.28

[e] Values in the parentheses are standard deviations of the values in the column.

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs. (continued)

Name	#	0 Hz	1 Hz	3 Hz
1HE8	40	0.03	4.65	7.68
1HIA	41	0.10	0.11	0.35
1I2M	42	0.05	0.07	0.25
1I4D	43	0.04	0.73	1.09
1I9R	44	0.03	0.09	0.27
1IB1	45	0.05	0.15	0.52
1IBR	46	0.16	0.61	0.84
1IJK	47	0.05	0.10	0.21
1IQD	48	0.01	0.09	0.23
1JPS	49	0.04	0.07	0.21
1K4C	50	0.00	0.08	0.20
1K5D	51	0.02	0.09	0.24
1KAC	52	0.10	0.09	0.30
1KKL	53	0.02	0.73	0.57
1KLU	54	0.02	0.05	0.24

[e] Values in the parentheses are standard deviations of the values in the column.

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs. (continued)

Name	#	0 Hz	1 Hz	3 Hz
1KTZ	55	0.01	0.07	0.43
1KXP	56	0.02	0.06	0.17
1KXQ	57	0.01	0.06	0.25
1M10	58	0.06	0.09	0.23
1MAH	59	0.02	0.11	0.34
1ML0	60	2.13	1.82	1.28
1MLC	61	0.20	0.16	0.31
1N2C	62	0.08	0.11	0.15
1NCA	63	0.01	0.09	0.19
1NSN	64	0.02	0.07	0.22
1PPE	65	0.03	0.25	0.53
1QA9	66	0.01	0.09	0.41
1QFW	67	0.03	0.13	0.43
1RLB	68	0.02	0.08	0.26
1SBB	69	0.02	0.07	0.30

[e] Values in the parentheses are standard deviations of the values in the column.

Table E.1: Docking results for COMPLEX dataset using synthetic RDCs. (continued)

Name	#	0 Hz	1 Hz	3 Hz
1TMQ	70	0.02	0.06	0.24
1UDI	71	0.02	0.14	0.32
1VFB	72	0.05	0.13	0.37
1WEJ	73	0.00	1.73	3.69
1WQ1	74	0.03	0.11	0.21
2BTF	75	0.02	0.07	0.18
2HMI	76	0.03	0.07	0.19
2JEL	77	0.04	0.10	0.29
2MTA	78	0.02	0.10	0.25
2PCC	79	0.02	0.11	0.20
2QFW	80	0.01	0.07	0.28
2SIC	81	0.01	0.09	0.34
2SNI	82	0.02	0.08	0.37
2VIS	83	0.01	0.08	0.19
7CEI	84	0.05	0.10	0.33
Mean		0.06 (0.23)[e]	0.22 (0.56)[e]	0.45 (0.90)[e]

[e] Values in the parentheses are standard deviations of the values in the column.

Bibliography

[1] A. Almond and J. Axelsen. Physical interpretation of residual dipolar couplings in neutral aligned media. *Journal of the American Chemical Society*, 124(34):9986–9987, 2002.

[2] H. Azurmendi, M. Martin-Pastor, and C. Bush. Conformational studies of Lewis X and Lewis A trisaccharides using NMR residual dipolar couplings. *Biopolymers*, 63(2):89–98, 2002.

[3] H. F. Azurmendi and C. A. Bush. Conformational studies of blood group A and blood group B oligosaccharides using NMR residual dipolar couplings. *Carbohydrate Research*, 337(10):905–915, 2002.

[4] A. Bax. Weak alignment offers new NMR opportunities to study protein structure and dynamics. *Protein Science*, 12(1):1–16, 2003.

[5] A. Bax, G. Kontaxis, and N. Tjandra. Dipolar couplings in macromolecular structure determination. In V. D. Thomas L. James and U. Schmitz, editors, *Part B: Nuclear Magnetic Resonance of Biological Macromolecules*, volume 339 of *Methods in Enzymology*, pages 127–174. Academic Press, 2001.

[6] K. Berlin, D. P. O'Leary, and D. Fushman. Improvement and analysis of computational methods for prediction of residual dipolar couplings. *Journal of Magnetic Resonance*, 201(1):25–33, 2009.

[7] K. Berlin, D. P. O'Leary, and D. Fushman. Structural assembly of molecular complexes based on residual dipolar couplings. preprint (2010), 2010.

[8] P. Bernadó, L. Blanchard, P. Timmins, D. Marion, R. Ruigrok, and M. Blackledge. A structural model for unfolded proteins from residual dipolar couplings and small-angle X-ray scattering. *Proceedings of the National Academy of Sciences*, 102(47):17002–17007, 2005.

[9] C. Bewley and G. Clore. Determination of the relative orientation of the two halves of the domain-swapped dimer of Cyanovirin-N in solution using dipolar couplings and rigid body minimization. *Journal of the American Chemical Society*, 122(25):6009–6016, 2000.

[10] C. Bewley, K. Gustafson, M. Boyd, D. Covell, A. Bax, G. Clore, and A. Gronenborn. Solution structure of Cyanovirin-N, a potent HIV-inactivating protein. *Nature Structural Biology*, 5(7):571–578, 1998.

[11] C. A. Bewley. Rapid validation of the overall structure of an internal domain-swapped mutant of the anti-HIV protein Cyanovirin-N using residual dipolar couplings. *Journal of the American Chemical Society*, 123(5):1014–1015, 2001.

[12] M. Blackledge. Recent progress in the study of biomolecular structure and dynamics in solution from residual dipolar couplings. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 46(1):23–61, 2005.

[13] R. Bruschweiler, X. Liao, and P. Wright. Long-range motional restrictions in a multidomain zinc-finger protein from anisotropic tumbling. *Science*, 268(5212):886–889, 1995.

[14] M. Cai, Y. Huang, R. Zheng, S. Wei, R. Ghirlando, M. Lee, R. Craigie, A. Gronenborn, and G. Clore. Solution structure of the cellular factor BAF responsible for protecting retroviral DNA from autointegration. *Nature Structural Biology*, 5(10):903–909, 1998.

[15] B. Carrasco and J. G. de la Torre. Hydrodynamic properties of rigid particles: Comparison of different modeling and computational procedures. *Biophysical Journal*, 76(6):3044–3057, 1999.

[16] J. Cavanagh, W. Fairbrother, A. Palmer III, N. Skelton, and M. Rance. *Protein NMR Spectroscopy: Principles and Practice*. Academic Press, San Diego, CA, USA, 1996.

[17] P. Chacn, F. Morn, J. Daz, E. Pantos, and J. Andreu. Low-resolution structures of proteins in solution retrieved from X-ray scattering with a genetic algorithm. *Biophysical Journal*, 74(6):2760–2775, 1998.

[18] R. Chen, L. Li, and Z. Weng. ZDOCK: An initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Genetics*, 52(1):80–87, 2003.

[19] G. Cornilescu, J. L. Marquardt, M. Ottiger, and A. Bax. Validation of protein structure from anisotropic carbonyl chemical shifts in a dilute liquid crystalline phase. *Journal of the American Chemical Society*, 120(27):6836–6837, 1998.

[20] E. de Alba, J. L. Baber, and N. Tjandra. The use of residual dipolar coupling in concert with backbone relaxation rates to identify conformational exchange by NMR. *Journal of the American Chemical Society*, 121(17):4282–4283, 1999.

[21] E. de Alba, L. De Vries, M. Farquhar, and N. Tjandra. Solution structure of human GAIP (Gα interacting protein): A regulator of G protein signaling. *Journal of Molecular Biology*, 291(4):927–939, 1999.

[22] M. De Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, New York, NY, 2000.

[23] J. G. de la Torre, M. L. Huertas, and B. Carrasco. Hydronmr: Prediction of nmr relaxation of globular proteins from atomic-level structures and hydrodynamic calculations. *Journal of Magnetic Resonance*, 147(1):138 – 146, 2000.

[24] S. de Vries, A. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar, and A. Bonvin. HADDOCK versus HADDOCK: New features and performance of HADDOCK2.0 on the CAPRI targets. *Proteins: Structure, Function, and Bioinformatics*, 69(4):726–733, 2007.

[25] C. Dominguez, R. Boelens, and A. Bonvin. HADDOCK: A protein-protein docking approach based on biochemical or biophysical data. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003.

[26] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin. HADDOCK: A protein–protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125:1731–1737, 2003.

[27] P. Dosset, J. Hus, D. Marion, and M. Blackledge. A novel interactive tool for rigid-body modeling of multi-domain macromolecules using residual dipolar couplings. *Journal of Biomolecular NMR*, 20(3):223–231, 2001.

[28] A. Drohat, N. Tjandra, D. Baldisseri, and D. Weber. The use of dipolar couplings for determining the solution structure of rat apo-S100B ($\beta\beta$). *Protein Science*, 8(4):800?–809, 1999.

[29] F. Eisenhaber, P. Lijnzaad, P. Argos, C. Sander, and M. Scharf. The double cubic lattice method: Efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *Journal of Computational Chemistry*, 16(3):273–284, 1995.

[30] M. X. Fernandes, P. Bernado, M. Pons, and J. Garcia de la Torre. An analytical solution to the problem of the orientation of rigid particles by planar obstacles. application to membrane systems and to the calculation of dipolar couplings in protein NMR spectroscopy. *Journal of the American Chemical Society*, 123(48):12037–12047, 2001.

[31] M. Fischer, J. Losonczi, J. Weaver, and J. Prestegard. Domain orientation and dynamics in multidomain proteins from residual dipolar couplings. *Biochemistry*, 38(28):9013–9022, 1999.

[32] D. Fushman and D. Cowburn. Model-independent analysis of 15N chemical shift anisotropy from NMR relaxation data. ubiquitin as a test example. *Journal of the American Chemical Society*, 120(28):7109–7110, 1998.

[33] D. Fushman and D. Cowburn. Nuclear magnetic resonance relaxation in determination of residue-specific 15N chemical shift tensors in proteins in solution: Protein dynamics, structure, and applications of transverse relaxation optimized spectroscopy. *Methods in Enzymology*, 339:109–122, 2001.

[34] D. Fushman, R. Ghose, and D. Cowburn. The effect of finite sampling on the determination of orientational properties: A theoretical treatment with application to interatomic vectors in proteins. *Journal of the American Chemical Society*, 122(43):10640–10649, 2000.

[35] D. Fushman, N. Tjandra, and D. Cowburn. Direct measurement of 15N chemical shift anisotropy in solution. *Journal of the American Chemical Society*, 120(42):10947–10952, 1998.

[36] D. Fushman, R. Varadan, M. Assfalg, and O. Walker. Determining domain orientation in macromolecules by using spin-relaxation and residual dipolar coupling measurements. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 44(3-4):189–214, 2004.

[37] R. Ghose, D. Fushman, and D. Cowburn. Determination of the rotational diffusion tensor of macromolecules in solution from NMR relaxation data with a combination of exact and approximate methods–application to the determination of interdomain orientation in multidomain proteins. *Journal of Magnetic Resonance*, 149(2):204–217, 2001.

[38] W. Hu and L. Wang. Residual dipolar couplings: Measurements and applications to biomolecular studies. *Annual Reports of NMR Spectroscopy*, 58:232, 2006.

[39] N. U. Jain, E. Tjioe, A. Savidor, and J. Boulie. Redox-dependent structural differences in putidaredoxin derived from homologous structure refinement via residual dipolar couplings. *Biochemistry*, 44(25):9067–9078, 2005.

[40] D. E. Krane and M. L. Raymer. *Fundamental concepts of bioinformatics*. Benjamin Cummings, San Francisco, CA, 2003.

[41] J. Kuszewski, A. M. Gronenborn, and G. M. Clore. Improving the packing and accuracy of NMR structures with a pseudopotential for the radius of gyration. *Journal of the American Chemical Society*, 121(10):2337–2338, 1999.

[42] E. Lawler and D. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.

[43] P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. In *VIS '98: Proceedings of the Conference on Visualization '98*, pages 279–286, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[44] P. Lindstrom and G. Turk. Evaluation of memoryless simplification. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):98–115, 1999.

[45] J. A. Losonczi, M. Andrec, M. W. F. Fischer, and J. H. Prestegard. Order matrix analysis of residual dipolar couplings using singular value decomposition. *Journal of Magnetic Resonance*, 138(2):334–342, 1999.

[46] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[47] A. McLachlan. Gene duplications in the structural evolution of chymotrypsin. *Journal of Molecular Biology*, 128(1):49–79, 1979.

[48] J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin, and Z. Weng. Protein–protein docking benchmark 2.0: An update. *Proteins*, 60:214–216, 2005.

[49] M. Mukrasch, P. Markwick, J. Biernat, M. von Bergen, P. Bernado, C. Griesinger, E. Mandelkow, M. Zweckstetter, and M. Blackledge. Highly populated turn conformations in natively unfolded tau protein identified from residual dipolar couplings and molecular simulation. *Journal of the American Chemical Society*, 129(16):5235–5243, 2007.

[50] F. Murnaghan. The element of volume of the rotation group. *Proceedings of the National Academy of Sciences*, 36(11):670–672, 1950.

[51] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, NY, 1996.

[52] D. O'Leary. *Scientific Computing with Case Studies*. Society for Industrial Mathematics, 2009.

[53] F. Perrin. Mouvement Brownien d'un ellipsoide (ii). rotation libre et dpolarisation des fluorescences. translation et diffusion de molcules ellipsoidales. *Le Journal de Physique*, 7:1–11, 1936.

[54] P. K. Redington. Molfit: A computer program for molecular superposition. *Computers & Chemistry*, 16(3):217–222, 1992.

[55] F. Richards. Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering*, 6(1):151–176, 1977.

[56] M. Ruckert and G. Otting. Alignment of biological macromolecules in novel nonionic liquid crystalline media for NMR experiments. *Journal of the American Chemical Society*, 122(32):7793–7797, 2000.

[57] Y. Ryabov and D. Fushman. Structural assembly of multidomain proteins and protein complexes guided by the overall rotational diffusion tensor. *Journal of the American Chemical Society*, 129(25):7894–7902, 2007.

[58] Y. Ryabov, C. Geraghty, A. Varshney, and D. Fushman. An efficient computational method for predicting rotational diffusion tensors of globular proteins using an ellipsoid representation. *Journal of the American Chemical Society*, 128(48):15432–15444, 2006.

[59] Y. Ryabov, J.-Y. Suh, A. Grishaev, G. M. Clore, and C. D. Schwieters. Using the experimentally determined components of the overall rotational diffusion tensor to restrain molecular shape and size in NMR structure determination of globular proteins and protein-protein complexes. *Journal of the American Chemical Society*, 131(27):9522–9531, 2009.

[60] H. Sass, G. Musco, S. Stahl, P. Wingfield, and S. Grzesiek. Solution NMR of proteins within polyacrylamide gels: Diffusional properties and residual alignment by mechanical stress or embedding of oriented purple membranes. *Journal of Biomolecular NMR*, 18(4):303–309, 2000.

[61] A. Saupe and G. Englert. High-resolution nuclear magnetic resonance spectra of orientated molecules. *Physical Review Letters*, 11(10):462–464, 1963.

[62] H. Schwalbe, S. Grimshaw, A. Spencer, M. Buck, J. Boyd, C. Dobson, C. Redfield, and L. Smith. A refined solution structure of hen lysozyme determined using residual dipolar coupling data. *Protein Science*, 10(4):677–688, 2001.

[63] C. Schwieters, J. Kuszewski, N. Tjandra, and G. Marius Clore. The Xplor-NIH NMR molecular structure determination package. *Journal of Magnetic Resonance*, 160(1):65–73, 2003.

[64] N. Skrynnikov, N. Goto, D. Yang, W. Choy, J. Tolman, G. Mueller, and L. Kay. Orienting domains in proteins using dipolar couplings measured by liquid-state NMR: Differences in solution and crystal forms of maltodextrin binding protein loaded with β-cyclodextrin. *Journal of Molecular Biology*, 295(5):1265–1273, 2000.

[65] G. Smith and M. Sternberg. Prediction of protein–protein interactions by docking methods. *Current Opinion in Structural Biology*, 12(1):28–35, 2002.

[66] H. Stuhrmann. Interpretation of small-angle scattering functions of dilute solutions and gases. a representation of the structures related to a one-particle scattering function. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 26(3):297–306, 1970.

[67] N. Tjandra and A. Bax. Direct measurement of distances and angles in biomolecules by NMR in a dilute liquid crystalline medium. *Science*, 278(5340):1111–1114, 1997.

[68] M. J. Todd and E. A. YIldIrIm. On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007.

[69] J. Tolman, J. Flanagan, M. Kennedy, and J. Prestegard. Nuclear magnetic dipole interactions in field-oriented proteins: Information for structure determination in solution. *Proceedings of the National Academy of Sciences*, 92(20):9279–9283, 1995.

[70] R. Tycko, F. J. Blanco, and Y. Ishii. Alignment of biopolymers in strained gels: A new way to create detectable dipole-dipole couplings in high-resolution biomolecular NMR. *Journal of the American Chemical Society*, 122(38):9340–9341, 2000.

[71] T. S. Ulmer, B. E. Ramirez, F. Delaglio, and A. Bax. Evaluation of backbone proton positions and dynamics in a small protein by liquid crystal NMR spectroscopy. *Journal of the American Chemical Society*, 125(30):9179–9191, 2003.

[72] E. Ulrich, H. Akutsu, J. Doreleijers, Y. Harano, Y. Ioannidis, J. Lin, M. Livny, S. Mading, D. Maziuk, Z. Miller, E. Nakatani, C. F. Schulte, D. E. Tolmie, R. K. Wenger, H. Yao, and J. L. Markley. BioMagResBank. *Nucleic Acids Research*, 36(Database issue):D402–D408, 2008.

[73] A. van Dijk, D. Fushman, and A. Bonvin. Various strategies of using residual dipolar couplings in NMR-driven protein docking: Application to lys48-linked di-ubiquitin and validation against 15N-relaxation data. *Proteins: Structure, Function, and Bioinformatics*, 60(3):367–381, 2005.

[74] A. D. van Dijk, D. Fushman, and A. M. Bonvin. Various Strategies of Using Residual Dipolar Couplings in NMR-Driven Protein Docking: Application to Lys48-Linked Di-Ubiquitin and Validation Against 15 N-Relaxation Data. *PROTEINS: Structure, Function, and Bioinformatics*, 60(3):367–381, 2005.

[75] A. D. J. van Dijk, R. Kaptein, R. Boelens, and A. M. J. J. Bonvin. Combining NMR relaxation with chemical shift perturbation data to drive protein–protein docking. *Journal of Biomolecular NMR*, 34(4):237–244, 2006.

[76] C. F. Van Loan. *Introduction to Scientific Computing: A Matrix-Vector Approach Using MATLAB*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[77] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

[78] R. Varadan, O. Walker, C. Pickart, and D. Fushman. Structural properties of polyubiquitin chains in solution. *Journal of Molecular Biology*, 324(4):637–647, 2002.

[79] A. Varshney and F. Brooks Jr. Fast analytical computation of Richard's smooth molecular surface. In *IEEE Visualization'93 Proceedings*, pages 300–307, 1993.

[80] A. Varshney, F. Brooks Jr, and W. V. Wright. Linearly scalable computation of smooth molecular surfaces. *IEEE Computer Graphics and Applications*, 14(5):19–25, 1994.

[81] O. Walker, R. Varadan, and D. Fushman. Efficient and accurate determination of the overall rotational diffusion tensor of a molecule from 15N relaxation data using computer program ROTDIF. *Journal of Magnetic Resonance*, 168:336–345, 2004.

[82] J. Warren and P. M. Moore. Application of dipolar coupling data to the refinement of the solution structure of the Sarcin-Ricin loop RNA. *Journal of Biomolecular NMR*, 20(4):311–323, 2001.

[83] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, volume 555, pages 359–370. Springer-Verlag, 1991.

[84] D. Woessner. Nuclear spin relaxation in ellipsoids undergoing rotational Brownian motion. *The Journal of Chemical Physics*, 37:647, 1962.

[85] D. Zhang, S. Raasi, and D. Fushman. Affinity makes the difference: Nonselective interaction of the uba domain of ubiquilin-1 with monomeric ubiquitin and polyubiquitin chains. *Journal of Molecular Biology*, 377(1):162–180, 2008.

[86] M. Zweckstetter. NMR: Prediction of molecular alignment from structure using the PALES software. *Nature Protocols*, 3(4):679–690, 2008.

[87] M. Zweckstetter and A. Bax. Prediction of sterically induced alignment in a dilute liquid crystalline phase: Aid to protein structure determination by NMR. *Journal of the American Chemical Society*, 122(15):3791–3792, 2000.