On Provable Algorithms for Determination of Continuous Protein Interdomain Motions from Residual Dipolar Couplings

by

Yang Qi

Department of Computer Science
Duke University

Date: ____________________________
Approved:

____
Bruce R. Donald, Supervisor

____
Mauro Maggioni

____
Terrence G. Oas

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Department of Computer Science
in the Graduate School of Duke University
2016

Abstract

Dynamics of biomolecules over various spatial and time scales are essential for biological functions such as molecular recognition, catalysis and signaling. However, reconstruction of biomolecular dynamics from experimental observables requires the determination of a conformational probability distribution. Unfortunately, these distributions cannot be fully constrained by the limited information from experiments, making the problem an ill-posed one in the terminology of Hadamard. The ill-posed nature of the problem comes from the fact that it has no unique solution. Multiple or even an infinite number of solutions may exist. To avoid the ill-posed nature, the problem needs to be regularized by making assumptions, which inevitably introduce biases into the result.

Here, I present two continuous probability density function approaches to solve an important inverse problem called the RDC trigonometric moment problem. By focusing on interdomain orientations we reduced the problem to determination of a distribution on the 3D rotational space from residual dipolar couplings (RDCs). We derived an analytical equation that relates alignment tensors of adjacent domains, which serves as the foundation of the two methods. In the first approach, the ill-posed nature of the problem was avoided by introducing a continuous distribution model, which enjoys a smoothness assumption. To find the optimal solution for the distribution, we also designed an efficient branch-and-bound algorithm that exploits the mathematical structure of the analytical solutions. The algorithm is guaranteed to find the distribution that best satisfies the analytical relationship. We observed good performance of the method when tested under various levels of experimental noise and when applied to two protein systems. The second approach avoids the use of any model by employing maximum entropy principles. This ‘model-free’ approach delivers the least biased result which presents our state of knowledge. In this approach, the solution is an exponential function of Lagrange multipliers. To determine the multipliers, a convex objective function is constructed. Consequently, the maximum entropy solution can be found easily by gradient descent methods. Both algorithms can be applied to biomolecular RDC data in general, including data from RNA and DNA molecules.

To my parents.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathcal{L}$	Lagrangian.
H	Hessian matrix.
$\langle f \rangle$	Expectation of function f given a distribution.
D_{KL}	Kullback-Leibler divergence
JSD	Jensen-Shannon divergence

Abbreviations

BnB	Branch-and-bound
CaM	Calmodulin
DOF	Degree of freedom
DoS	Disk-on-Sphere
NMR	Nuclear magnetic resonance
PCS	Pseudo contact shift
RDC	Residual dipolar coupling
SpA-N	N terminal domains of Staphylococcal protein A
SVD	Singular value decomposition
OLC	Orthogonal linear combination

Acknowledgements

The path towards this thesis is a fascinating journey, mixed with joyful ups and downs. I would like to express my sincere gratitude to my advisor, Dr. Bruce Donald for his introduction to a mathematically sound formulation of the problem and his guidance along developing these provable algorithms. I also thank my committee members, Dr. Terrence Oas and Dr. Mauro Maggioni for their guidance and support over the years.

I'm very grateful that I didn't work alone on the path. I couldn't make this far without the help from the Donald lab members. I would like to thank Dr. Mark Hallen, Dr. Swati Jain, Dr. Pablo Gainza, Hunter Nisonoff and all other members for their helpful and stimulating discussions. I especially wish to thank Dr. Jeffrey Martin, Dr. Anthony Yan and François Thélot. Jeff introduced me to quaternion rotations and worked with me to write down the relationship between Saupe tensors, which is the foundation of the thesis. Tony's introduction to ill-posed problem gave me such a deep impression that I can still remember it vividly today. I really enjoyed working with François Thélot on the pseudo-contact shift problem and on the maximum entropy method. Besides the Donald lab, I thank my friend, Dr. Tingran Gao and Dr. Shiwen Zhao for their helpful discussions and mathematical insights.

Finally, I would like to thank my supportive friends and family. I wish to give my best regards to my friends Weiwei Li, Hui Kang, Xiao Yan, Dr. Yu Jiang, Ruo

He, Dr. Shiwen Zhao, Rujie Yin and Dr. Tingran Gao, who share their life moments with me.

1

Introduction

1.1 Interdomain motions and residual dipolar couplings (RDCs)

Dynamics of biological macromolecules are essential for biological functions such as molecular recognition, catalysis and signaling. Biomolecular dynamics occur over various spatial and time scales and include motional modes such as local bond vibration, sidechain rearrangement, interdomain motion and global tumbling. Among them, interdomain motions are important for molecular recognition, as demonstrated in multiple studies [1, 2, 3]. Protein and RNA molecules can rearrange interdomain orientations upon binding, thereby enabling adaptive conformational changes that are accompanied by contributions to the free energy, internal energy and entropy of binding. In addition, different binding geometries were observed in the presence of different binding partners [2]. The observations suggest the interdomain motions of biopolymers direct the transitions between different conformations, contributing to their functional plasticity.

Residual dipolar couplings were first introduced to biomolecular systems as structural constraints. A RDC between two spins is a function of angle θ as in Eq. (1.1),

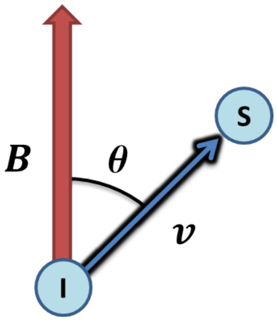

FIGURE 1.1: The angle θ between the bond vector v and the direction of the magnetic field B. I and S are two spins forming a bond vector.

which is the angle between the bond vector from one spin to the other and the direction of the magnetic field in Fig. 1.1. Compared to other NMR local constraints, RDCs restraint the direction of bond vector and serve as global constraints. Because the direction of bond vector was determined independently, its error does not accumulate over the distance. As global constraints, RDCs provide complementary information to constrain the overall fold of biomolecular structure, especially the relative orientation between secondary structures [4]. In addition to structural information, RDCs also contain dynamic information. Because the bond vector is rotating when the molecule is in solution, there is a distribution of angles θ instead of one unique angle. RDCs are scalar averages over the angle distributions and thus have a rich information content of biomolecular dynamics. The dynamics observed by RDCs cover a wide range of timescales, ranging from picosecond to millisecond. The combined structural and dynamics information over the wide range of timescales makes RDCs an exceptional powerful technique to probe the spatial nature of conformational fluctuations of biomolecules.

$$D_{IS} = -\frac{\mu_0 \gamma_I \gamma_S \hbar}{4\pi^2 \langle r_{IS}^3 \rangle} \langle \frac{3\cos^2\theta - 1}{2} \rangle. \tag{1.1}$$

Here, we focus on interdomain motions and study systems with relative little

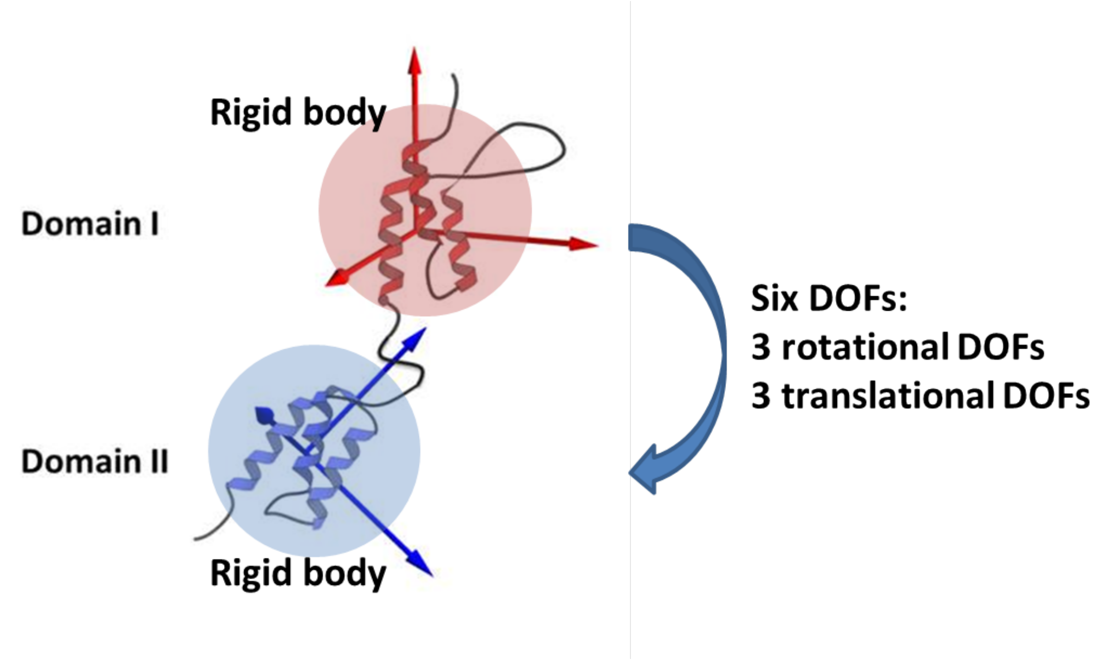

Figure 1.2: The domains are considered as rigid bodies by ignoring intradomain motions. Six degrees of freedom (DOFs), three translational DOFs and three rotational DOFs exist between the two domains.

intradomain dynamics. As a result, the domains are treated as rigid bodies and the interdomain linker are considered flexible as in Fig. 1.2. In order words, we only consider six degrees of freedom (DOFs), three translational DOFs and three rotational DOFs between the domains. Under the assumption, the structure of the two domains are static. A rotational motion tensor can be calculated for each domain given the static structures of the two domains. The tensor, also known as Saupe tensor, contains the structural dynamic information in RDCs. So for the interdomain motion problem, all the information we have are in the Saupe tensors. Intuitively, the Saupe tensor of domain I informs on the global tumbling of the whole molecule and the Saupe tensor of domain II depends on both global tumbling and interdomain motions. If we can separate the global tumbling component from the second Saupe tensor, we extract the dynamic information of interdomain motions.

1.2 Reconstructing interdomain motions from RDCs is an ill-posed inverse problem

Although RDCs deliver the desired information, using the piece of information to reconstruct interdomain motions poses a big computational challenge. Because the observation of RDCs from experiments is a forward process, the problem of reconstruction is an inverse problem (Fig. 1.3). A simulation of the forward process is straight forward, because the forward simulation is well-posed. However, the inverse problem is ill-posed due to the lack of information. As noted previously, RDCs contain rich information about interdomain motions. Obtaining independent RDC datasets can further increase the information content [5]. But RDCs and the derived Saupe tensors only contain average information. In the two approaches discussed in the following chapters, the interdomain motions are represented as a probability distribution. The average information in RDCs are moments of the distributions. As most moment problems [6], reconstructing interdomain motions from RDCs is also an ill-posed problem.

Following the definition of Hadamard [7, 8], the solution of an ill-posed problem may not be unique, or the solution does not exist, or the solution does not depend continuously on the data. Any violation of the solution existence, uniqueness or continuity makes the problem an ill-posed one. In our case, the data are collected through physical experiments, solution existence and continuity are not much of a concern. However, multiple or an infinite number of solutions can satisfy the data, so the solution is not unique. Because a fraction of information is lost in the forward process, a recovery of the ground truth is impossible. However, approximate and stable solutions for ill-posed problems can be determined with the help of regularization methods. Regularizations generally select one solution from the set of all possible solutions. The selection is based on prior information or reasonable assumptions. The

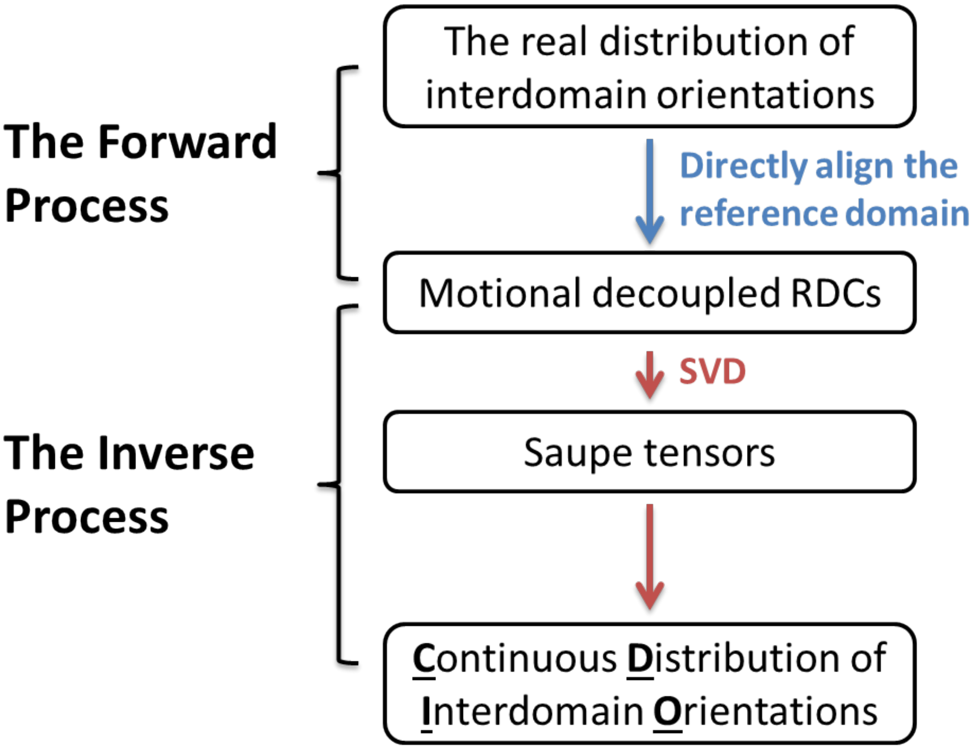
The Forward Process
The Inverse Process
The real distribution of interdomain orientations
Directly align the reference domain
Motional decoupled RDCs
SVD
Saupe tensors
Continuous Distribution of Interdomain Orientations

FIGURE 1.3: The forward process is to convert the ground truth distribution to RDCs either by experiments or simulations. The inverse process is to reconstruct the distribution from RDCs.

smoothness assumption is a reasonable assumption for many problems. In our case, because a biomolecular domain rotate through the space continuously, the probability distribution should also be continuous. In addition, the probability distribution is dictated by the molecule's surrounding force field. Given no evidence of abrupt change in the force field, the distribution should be smooth. As discussed in the following chapters, the smoothness assumption is used to regularize the problem.

1.3 Previous approaches

Several methods have been developed to solve the ill-posed problem, including maximum allowed probability (MAP) [1], sample-and-select (SAS) [9] and sparse ensemble selection (SES) [3]. All three methods use a discrete finite ensemble to describe interdomain motions. They select an ensemble of discrete conformers from a preconfigured conformational pool. When conformations in the pool are generated by molecular dynamics simulation or selected by an energy function, the conformations are restricted to regions where the empirical energy is favorable. Because of discrete nature and the use of energy function, all the methods suffer from three disadvantages. First, although the energy function offers regularization, it also introduces empirical assumptions into the result. Unfortunately, the energy function does not have a good ability to predict salient features of unfolded states or unstructured regions, raising doubts concerning the associated assumptions [10]. Second, for large amplitude interdomain motions, the number of discrete conformers required to represent the broad conformational distribution is enormous, which increases the risk of over-fitting. Last but not least, a discrete ensemble assigns certain probabilities to the conformations in the ensemble and assigns zero probability to the rest of the conformational space. Although the ensemble description captures several characteristics of biomolecular motions, the void of probability between structures is physically unreasonable.

The use of discrete conformers also weakens the maximum entropy claim. In order to compute the solution in a realistic amount of time and avoid over-fitting, previous maximum entropy methods restrict the number of discrete conformers [11, 12]. Although the derived solution by those methods has the maximum entropy given the number of conformers, the solution is certainly not the maximum entropy solution among all solutions satisfying the constraints. On the other hand, the maximum entropy solution is supposed to be the least biased solution. By selecting certain number of conformers from a preconfigured conformational pool, additional assumptions are made and biases are introduced.

2

The Bingham model and the branch-and-bound algorithm

2.1 The information about interdomain motions is summarized in the Q matrix

Besides Eq. (1.1), there is an alternative expression for RDC. The averaging in RDCs can be summarized by using a tensor, which is a 3 × 3 symmetric matrix with zero trace. The matrix is formally named as a Saupe tensor. Thus, the RDC equation can be formulated in the following way [13]:

$$D = \frac{K}{2}\mathbf{v}^T S\mathbf{v}. \tag{2.1}$$

In Eq. (2.1), K is the dipolar coupling constant, v is a normalized unit bond vector and S is a Saupe tensor. The dipolar coupling constant K is calculated as:

$$K = -\frac{\mu_0 \gamma_I \gamma_S \hbar}{4\pi^2 r_{IS}^3} \tag{2.2}$$

The Saupe tensor is a 3 × 3 traceless symmetric matrix:

$$S = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{xy} & S_{yy} & S_{yz} \\ S_{xz} & S_{yz} & S_{zz} \end{bmatrix}, \tag{2.3}$$

which has 5 DOFs: axiality, rhombicity and orientation of the principle axes in a molecular frame. Saupe tensors can be calculated from RDCs by an SVD method [14].

The interdomain motions can be described by a probability distribution. If each interdomain orientation can be assigned a probability, the interdomain dynamics are completely quantified. Thus, we can convert the problem of determining interdomain motions to determining a probability distribution. Furthermore, each interdomain orientation can be represented as a rotation (Fig. 2.1). In the double domain scenario, we have Domain I and Domain II. Suppose we attach a set of axes to each domain. Then an interdomain orientation can be represented by the orientational difference between the two sets of axes, in other words, a rotation. The rotation can be parameterized either by a rotation matrix or a quaternion. Consequently, we reduce the problem to determining a probability distribution over the rotation space $SO(3)$.

Following the above framework, the two Saupe tensors determined from their own molecular frame can be related by a rotation if the two domains have a fixed orientation:

$$D = \frac{K}{2}\mathbf{v}_{\mathrm{I}}{}^{T} S_{\mathrm{I}} \mathbf{v}_{\mathrm{I}} = \frac{K}{2}\mathbf{v}_{\mathrm{II}}{}^{T} R^{T} S_{\mathrm{I}} R \mathbf{v}_{\mathrm{II}} \tag{2.4}$$

$$D = \frac{K}{2}\mathbf{v}_{\mathrm{II}}{}^{T} S_{\mathrm{II}} \mathbf{v}_{\mathrm{II}} \tag{2.5}$$

where $\mathbf{v}_{\mathrm{I}}$ is a normalized unit bond vector in the molecular frame of the first domain and $\mathbf{v}_{\mathrm{II}}$ is a normalized bond vector in the molecular frame of the second domain. $R \in SO(3)$ and can be parameterized as a 3×3 rotation matrix, which has orthogonal

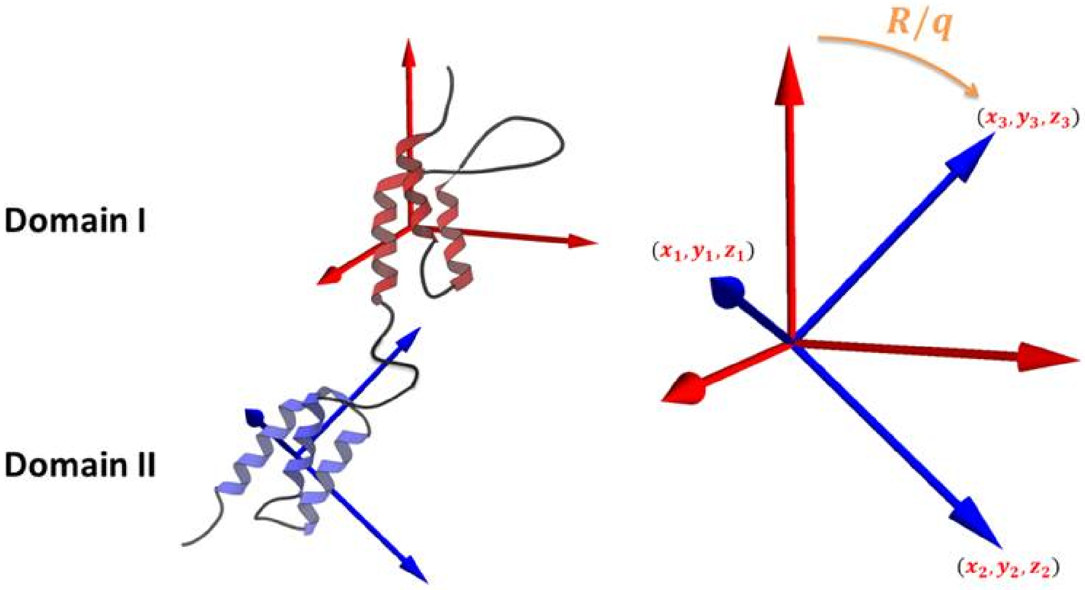

FIGURE 2.1: The interdomain orientation of two domains can be represented by a rotation.

rows and columns, and a determinant of +1:

$$R = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}. \tag{2.6}$$

Hence,

$$S_{\text{II}} = R^T S_{\text{I}} R. \tag{2.7}$$

When there are interdomain motions, the second Saupe tensor S_{II} is an average over all possible orientations:

$$S_{\text{II}} = \langle R^T S_{\text{I}} R \rangle. \tag{2.8}$$

In the current experimental setting, Domain I is aligned directly and Domain II is aligned indirectly through the linker. So the interdomain motions do not influence the alignment of the first domain. In other words, the rotational motion of Domain I is decoupled from the interdomain motions (Fig. 2.2). Because of the motional

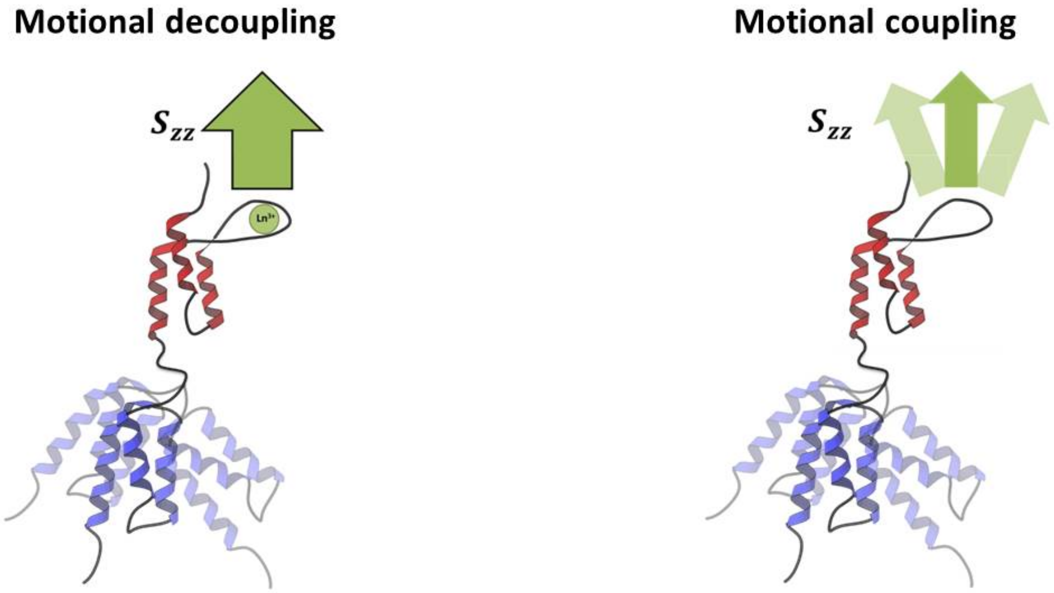

FIGURE 2.2: Motional decoupling is achieved when Domain I is experimentally set as a reference.

decoupling, S_{I} remains a constant. Thus, we can take S_{I} out of the average. In order to do so, first we rewrite the rotation matrix R in the following form:

$$R = \left[\begin{array}{ccc} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{array}\right], \tag{2.9}$$

where $\mathbf{x} = [x_1, x_2, x_3]^T$, $\mathbf{y} = [y_1, y_2, y_3]^T$ and $\mathbf{z} = [z_1, z_2, z_3]^T$. As a result, Eq. (2.8) becomes:

$$S_{\mathrm{II}} = \langle \left[\begin{array}{c} \mathbf{x}^T \\ \mathbf{y}^T \\ \mathbf{z}^T \end{array}\right] S_{\mathrm{I}} \left[\begin{array}{ccc} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{array}\right] \rangle. \tag{2.10}$$

From Eq. (2.10), we have:

$$S_{\mathrm{II},ab} = \langle \mathbf{a}^T S_{\mathrm{I}} \mathbf{b} \rangle, \tag{2.11}$$

where $(a, \mathbf{a}) \in \{(x, \mathbf{x}), (y, \mathbf{y}), (z, \mathbf{z})\}$ and $(b, \mathbf{b}) \in \{(x, \mathbf{x}), (y, \mathbf{y}), (z, \mathbf{z})\}$. This subscript convention allows us to conveniently define each element of the Saupe tensor S_{II}. It should be noted that on the left hand side, $S_{\mathrm{II},ab}$ is an element in S_{II} and on the right hand side, S_I is a 3 × 3 matrix. The right hand side of Eq. (2.11) can be written as a scalar product of tensors [13]:

$$S_{\mathrm{II},ab} = T_{ab} \odot S_{\mathrm{I}}, \tag{2.12}$$

where $\odot$ is the Frobenius inner product, T_{ab} is a 3 × 3 matrix in the following form:

$$T_{ab} = \langle \mathbf{a}\mathbf{b}^T \rangle = \begin{bmatrix} \langle a_1 b_1 \rangle & \langle a_1 b_2 \rangle & \langle a_1 b_3 \rangle \\ \langle a_2 b_1 \rangle & \langle a_2 b_2 \rangle & \langle a_2 b_3 \rangle \\ \langle a_3 b_1 \rangle & \langle a_3 b_2 \rangle & \langle a_3 b_3 \rangle \end{bmatrix}. \tag{2.13}$$

Again, $(a, \mathbf{a}) \in \{(x, \mathbf{x}), (y, \mathbf{y}), (z, \mathbf{z})\}$ and $(b, \mathbf{b}) \in \{(x, \mathbf{x}), (y, \mathbf{y}), (z, \mathbf{z})\}$. By writing out the elements of Saupe tensor S_{I}, we have:

$$S_{\mathrm{II},ab} = \langle a_1 b_1 \rangle S_{\mathrm{I},xx} + \langle a_2 b_2 \rangle S_{\mathrm{I},yy} + \langle a_3 b_3 \rangle S_{\mathrm{I},zz} + \langle a_1 b_2 + a_2 b_1 \rangle S_{\mathrm{I},xy} + \langle a_1 b_3 + a_3 b_1 \rangle S_{\mathrm{I},xz} + \langle a_2 b_3 + a_3 b_2 \rangle S_{\mathrm{I},yz}. \tag{2.14}$$

Because Saupe tensor is traceless, we have $S_{xx} + S_{yy} + S_{zz} = 0$ and:

$$S_{\mathrm{II},ab} = \langle a_1 b_1 - a_3 b_3 \rangle S_{\mathrm{I},xx} + \langle a_2 b_2 - a_3 b_3 \rangle S_{\mathrm{I},yy} + \langle a_1 b_2 + a_2 b_1 \rangle S_{\mathrm{I},xy} + \langle a_1 b_3 + a_3 b_1 \rangle S_{\mathrm{I},xz} + \langle a_2 b_3 + a_3 b_2 \rangle S_{\mathrm{I},yz}, \tag{2.15}$$

where $a \in \{x, y, z\}$ and $b \in \{x, y, z\}$. If we summarize the above result in Eq. (2.15) by vectorizing the Saupe tensors, we have:

$$\mathbf{s}_{\mathrm{II}} = \mathbf{E}_{R \sim P(R)}[Q(R)] \cdot \mathbf{s}_{\mathrm{I}} \tag{2.16}$$

where

$$Q(R) = \begin{bmatrix} x_1^2 - x_3^2 & x_2^2 - x_3^2 & 2x_1x_2 & 2x_1x_3 & 2x_2x_3 \\ y_1^2 - y_3^2 & y_2^2 - y_3^2 & 2y_1y_2 & 2y_1y_3 & 2y_2y_3 \\ x_1y_1 - x_3y_3 & x_2y_2 - x_3y_3 & x_1y_2 + x_2y_1 & x_1y_3 + x_3y_1 & x_2y_3 + x_3y_2 \\ x_1z_1 - x_3z_3 & x_2z_2 - x_3z_3 & x_1z_2 + x_2z_1 & x_1z_3 + x_3z_1 & x_2z_3 + x_3z_2 \\ y_1z_1 - y_3z_3 & y_2z_2 - y_3z_3 & y_1z_2 + y_2z_1 & y_1z_3 + y_3z_1 & y_2z_3 + y_3z_2 \end{bmatrix} \tag{2.17}$$

and

$$\mathbf{E}_{R\sim P(R)}[Q(R)] = \begin{bmatrix} \langle x_1^2 - x_3^2 \rangle & \langle x_2^2 - x_3^2 \rangle & \langle 2x_1x_2 \rangle & \langle 2x_1x_3 \rangle & \langle 2x_2x_3 \rangle \\ \langle y_1^2 - y_3^2 \rangle & \langle y_2^2 - y_3^2 \rangle & \langle 2y_1y_2 \rangle & \langle 2y_1y_3 \rangle & \langle 2y_2y_3 \rangle \\ \langle x_1y_1 - x_3y_3 \rangle & \langle x_2y_2 - x_3y_3 \rangle & \langle x_1y_2 + x_2y_1 \rangle & \langle x_1y_3 + x_3y_1 \rangle & \langle x_2y_3 + x_3y_2 \rangle \\ \langle x_1z_1 - x_3z_3 \rangle & \langle x_2z_2 - x_3z_3 \rangle & \langle x_1z_2 + x_2z_1 \rangle & \langle x_1z_3 + x_3z_1 \rangle & \langle x_2z_3 + x_3z_2 \rangle \\ \langle y_1z_1 - y_3z_3 \rangle & \langle y_2z_2 - y_3z_3 \rangle & \langle y_1z_2 + y_2z_1 \rangle & \langle y_1z_3 + y_3z_1 \rangle & \langle y_2z_3 + y_3z_2 \rangle \end{bmatrix}. \tag{2.18}$$

In Eqs. (2.16-2.18), $\mathbf{s}_{\text{I}}$ and $\mathbf{s}_{\text{II}}$ are vectorized Saupe tensors with 5 elements in the format $s = [S_{xx}, S_{yy}, S_{xy}, S_{xz}, S_{yz}]^T$. $R \in SO(3)$ is parameterized as in Eq. (2.6). It is important to see that the scalars x_i, y_i and z_i, $i = (1, 2, 3)$ in Eqs. (2.17-2.18) are precisely the nine elements of the rotation matrix R given in Eq. (2.6). The Q matrix is a function of the interdomain orientation R, shown in Eq. (2.17). $\mathbf{E}_{R\sim P(R)}[Q(R)]$ is the expectation of the Q matrix over the rotation space $SO(3)$. $P(R)$ is a distribution of R over the rotation space $SO(3)$. Define matrix $G \equiv \mathbf{E}_{R\sim P(R)}[Q(R)]$. Then each element in the matrix G can be calculated by an integration:

$$G_{ij} = \int_{SO(3)} Q_{ij} P(R) dR. \tag{2.19}$$

In Eq. (2.16), $\mathbf{s}_{\text{I}}$ and $\mathbf{s}_{\text{II}}$ are directly derived from experimental observables. With one or multiple alignments, we can obtain the full Q matrix or linear combinations of its elements. Because Eq. (2.16) expresses a relationship between the experimental observables $\mathbf{s}_{\text{I}}$ and $\mathbf{s}_{\text{II}}$, and the underlying interdomain orientational distribution, the full Q matrix or linear combinations of its elements contain information about the orientational distribution, and the information can serve as geometric constraints.

2.2 The Q matrix can be decomposed into three matrices

The family of Bingham distributions [15] are widely used to describe circular distributions on the 2D sphere S^2 and 3D rotational space $SO(3)$ [16]. Previous studies demonstrated the Bingham model's ability to represent salient features of a broad spectrum of orientational distributions [16]. Consequently, we model the interdomain orientational distribution as the Bingham distribution on $SO(3)$, which takes the following form [15, 16]:

$$P(\tilde{\mathbf{q}};\, X) = c^{-1}(X)\exp(\tilde{\mathbf{q}}^T X \tilde{\mathbf{q}}). \tag{2.20}$$

In Eq. (2.20), $c^{-1}(X)$ is the normalization factor. $P(\tilde{\mathbf{q}};\, X)$ is the probability of $\tilde{\mathbf{q}}$ given X. $q \in SO(3)$ is a rotation and $\tilde{\mathbf{q}}$ is the 4D unit quaternion representation of q. X is a symmetric 4×4 matrix with a constant trace and thus 9 degrees of freedom (DOFs). The meaning of the 9 DOFs becomes more clear if we decompose X in the following way:

$$X = M^T \Lambda M. \tag{2.21}$$

Here, M is a rotation matrix belonging to the group $SO(4)$ and Λ is a diagonal matrix with a constant trace specifying the variances along the four principle directions. M contains 6 DOFs and Λ contains 3 DOFs. The rotation matrix $M \in SO(4)$ can be further decomposed into a left isoclinic rotation M^L and a right isoclinic rotation M^R:

$$M = M^L M^R. \tag{2.22}$$

The left isoclinic rotation M^L corresponds to a left quaternion rotation $q_{_L}$, the right isoclinic rotation M^R corresponds to a right quaternion rotation $q_{_R}$. Consequently, the 9 DOFs in the Bingham distribution can be separated into three matrices, Λ, M^L and M^R.

When the Bingham distribution is used to model an interdomain distribution, the Q matrix can be decomposed into three matrices. The decomposition of the Q matrix is proved by using the matrix von Mises Fisher (vMF) distribution. The 3×3 matrix vMF distribution is a distribution of 3×3 orthogonal matrices with determinants of +1 on $SO(3)$ [17]. The probability density function of the vMF distribution takes the following form:

$$P(R\,;\,F) = c^{-1}(F)\exp(\mathrm{Tr}(FR)) \tag{2.23}$$

In Eq. (2.23), $c^{-1}(F)$ is a normalization factor. $P(R\,;\,F)$ is the probability of R given F. $R \in SO(3)$ is a rotation matrix and F is a 3×3 matrix containing 9 parameters. F can be decomposed in the following way:

$$F = \Theta D \Gamma, \tag{2.24}$$

where $\Theta \in SO(3)$ and $\Gamma \in SO(3)$ are two rotation matrices. $D = \mathrm{Diag}(\phi_1, \phi_2, \phi_3)$ is a 3×3 diagonal matrix with 3 concentration parameters. The matrix vMF distribution is equivalent to the Bingham distribution [17] with the following relationships:

$$\Lambda = \begin{bmatrix} \phi_1 + \phi_2 + \phi_3 & 0 & 0 & 0 \\ 0 & \phi_1 - \phi_2 - \phi_3 & 0 & 0 \\ 0 & 0 & \phi_2 - \phi_1 - \phi_3 & 0 \\ 0 & 0 & 0 & \phi_3 - \phi_1 - \phi_2 \end{bmatrix} \tag{2.25}$$

$$\Gamma = \mathrm{RotM}(\tilde{\mathbf{q}}_{\mathbf{L}}) \tag{2.26}$$

$$\Theta = \mathrm{RotM}(\tilde{\mathbf{q}}_{\mathbf{R}}), \tag{2.27}$$

where RotM is a function mapping from a quaternion $\tilde{\mathbf{q}}$ to its corresponding rotation matrix. If we construct a new matrix $Y = \Gamma R \Theta$, we can rewrite Eq. (2.16) as

$$\mathbf{s_{II}} = \mathbf{E}_{Y \sim P(Y\,;\,D)}[Q(\Gamma^T Y \Theta^T)] \cdot \mathbf{s_I}, \tag{2.28}$$

where $Y \in SO(3)$ and $Q(R)$ is a function converting the rotation matrix $R \in SO(3)$ into the Q matrix form as shown in Eq. (2.17). We observe the following property of $Q(R)$ when the rotation matrix R is a product of two rotations R_1 and R_2:

$$S_{II} = R^T S_I R = R_2^T R_1^T S_I R_1 R_2. \tag{2.29}$$

Correspondingly, we have:

$$\mathbf{s_{II}} = Q(R) \cdot \mathbf{s_I} = Q(R_2) \cdot Q(R_1) \cdot \mathbf{s_I} \tag{2.30}$$

and

$$Q(R) = Q(R_2) \cdot Q(R_1). \tag{2.31}$$

Using the above property in Eq. (2.31), we could write Eq. (2.28) in the following form:

$$\mathbf{s_{II}} = Q(\Theta^T) \cdot \mathbf{E}_{Y \sim P(Y\,;\,D)}[Q(Y)] \cdot Q(\Gamma^T) \cdot \mathbf{s_I}, \tag{2.32}$$

As a result, the following is true for the Bingham distribution:

$$\mathbf{s}_{\text{II}} = Q(\tilde{\mathbf{q}}_{\mathbf{R}}^{-1}) \cdot \mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda)}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})] \cdot Q(\tilde{\mathbf{q}}_{\mathbf{L}}^{-1}) \cdot \mathbf{s}_{\text{I}}. \tag{2.33}$$

In Eq. (2.33), $\tilde{\mathbf{q}}_{\mathbf{R}}$ and $\tilde{\mathbf{q}}_{\mathbf{L}}$ are two quaternions corresponding to rotations specified by rotation matrices $\Theta \in SO(3)$ and $\Gamma \in SO(3)$. The probability density function used to calculate the expectation $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda)}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]$ is

$$P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda) = c^{-1}(\Lambda)\exp(\tilde{\mathbf{q}}_{\mathbf{Y}}^T \Lambda \tilde{\mathbf{q}}_{\mathbf{Y}}) \tag{2.34}$$

where

$$q_{_Y} = q_{_L} q q_{_R}. \tag{2.35}$$

In Eq. (2.33), $Q(\tilde{\mathbf{q}})$ is a function converting the quaternion $\tilde{\mathbf{q}}$ into the Q matrix form. Essentially, $Q(\tilde{\mathbf{q}}) = Q(\text{RotM}(\tilde{\mathbf{q}}))$.

In addition, the distribution $P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda)$ is symmetric with respect to xyz, xyw, xzw and yzw hyperplanes in the 4D space. Let $\tilde{\mathbf{q}}_{\mathbf{Y}} = (q_{_1}, q_{_2}, q_{_3}, q_{_4})$. Because of the 4 hyperplane symmetries, most expectations of quartic terms are zero. There are only ten non-zero expectations of quartic terms, $\langle q_{_i}^4 \rangle$ and $\langle q_{_i}^2 q_{_j}^2 \rangle$, where $i, j = 1, 2, 3, 4$. The expectation matrix of $Q(\tilde{\mathbf{q}}_{\mathbf{Y}})$ contains a large number of zero entries. If we define

$$a_{_{ij}} = \frac{1}{3}\langle q_{_i}^4 + q_{_j}^4 - 6 q_{_i}^2 q_{_j}^2 \rangle \tag{2.36}$$

Then

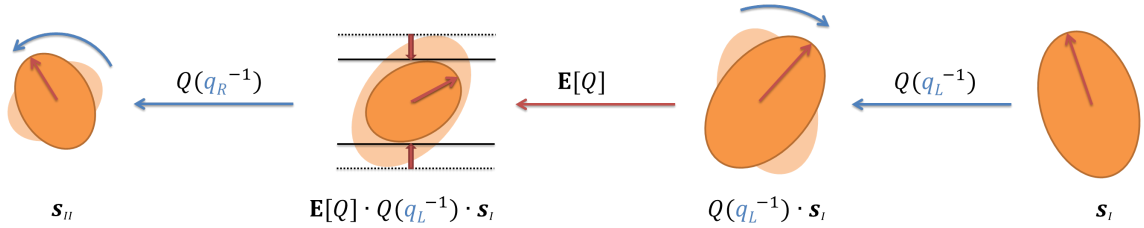
$Q(q_R{}^{-1})$
$\mathbf{E}[Q]$
$Q(q_L{}^{-1})$
$\boldsymbol{s}_{II}$
$\mathbf{E}[Q]\cdot Q(q_L{}^{-1})\cdot \boldsymbol{s}_I$
$Q(q_L{}^{-1})\cdot \boldsymbol{s}_I$
$\boldsymbol{s}_I$

FIGURE 2.3: The three matrices can be viewed as operators transforming Saupe tensors.

$$\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}}\sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\Lambda)}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})] = \begin{bmatrix} -a_{12} - a_{34} + 3a_{13} + 3a_{24} + a_{14} + a_{23} & 2a_{13} + 2a_{24} - 2a_{14} - 2a_{23} & 0 & 0 & 0 \\ 2a_{12} + 2a_{34} - 2a_{14} - 2a_{23} & 3a_{12} + 3a_{34} - a_{13} - a_{24} + a_{14} + a_{23} & 0 & 0 & 0 \\ 0 & 0 & 3a_{14} - 3a_{23} & 0 & 0 \\ 0 & 0 & 0 & 3a_{13} - 3a_{24} & 0 \\ 0 & 0 & 0 & 0 & 3a_{12} - 3a_{34} \end{bmatrix} \tag{2.37}$$

Eq. (2.36) and Eq. (2.37) are evaluated by efficiently approximating the matrix hypergeometric function [18] (Appendix A).

2.3 A three-step branch and bound algorithm to determine a unimodal distribution

Following Eq. (2.33), Q matrix can be decomposed into three matrices. Each of the matrices has three DOFs, living in the spaces S^3, $\mathbb{R}^3$ and S^3, respectively. In order to distinguish the two rotation spaces, the space corresponding to a left rotation is notated as S^3_L and the one corresponding to a right rotation is notated as S^3_R. When the matrices are viewed as operators, the Saupe tensors get rotated but the magnitude of principle components of a Saupe tensor remain the same under rotation operations (Fig. 2.3). Consequently, the first left rotation and the second averaging operation transform the principle components of $\mathbf{s}_{\text{I}}$ into those of $\mathbf{s}_{\text{II}}$. Based on this, we focus on $\mathbb{R}^3 \times S^3_L$ in the first two steps of search, thus reducing the dimensionality, and find a subspace in $\mathbb{R}^3 \times S^3_L$ which transforms the largest principle component $D_{\text{I},a}$ of $\mathbf{s}_{\text{I}}$ into $D_{\text{II},a}$ of $\mathbf{s}_{\text{II}}$.

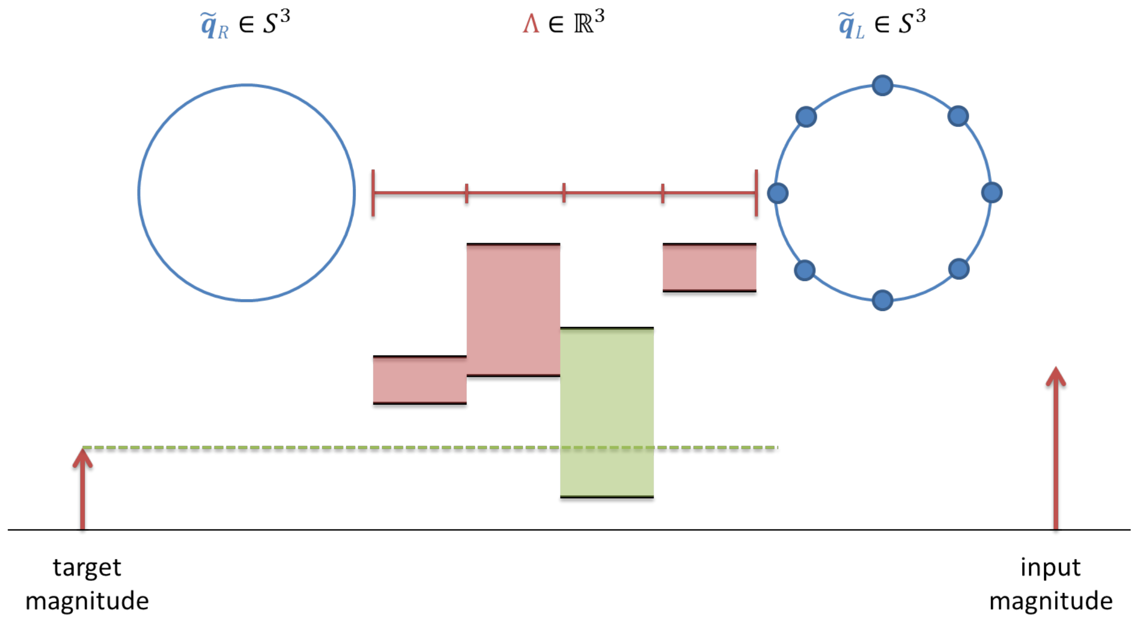

FIGURE 2.4: The first step in the branch-and-bound algorithm. For illustration, the $\mathbb{R}^3$ space is branched into 4 regions. The upper and lower bound for each region are calculated by sampling the right-side S^3 space. The red arrow on the right represents the magnitude of the largest principle component of the first Saupe tensor S_{I}. The red arrow on the left represents the magnitude of the largest principle component of the second Saupe tensor S_{II}. Regions (green) covering the target magnitude is saved for the next round of branch-and-bound. Other regions (red) are pruned.

Step 1. The first step is to prune the $\mathbb{R}^3$ space. Before pruning, bounds of the largest principle component D_a on a region of $\mathbb{R}^3$ are built. If $D_{\text{II},a}$ does not fall into the range between the bounds, the region can not contain the solution to Eq. (2.33) and will be consequently pruned (Fig. 2.4). Detailed description of building the bounds is shown in the following.

In Eq. (2.20), M is a rotation matrix in $SO(4)$ and $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is a diagonal matrix with four concentration parameters. Without ordering the concentration parameters, there are multiple pairs of Λ and M corresponding to the same distribution. To resolve the ambiguity, the concentration parameters are defined in the order of $\lambda_1 > \lambda_4 > \lambda_3 > \lambda_2$. $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 0$ is also enforced because for any scalar a, Λ and $\Lambda - aI$ correspond to the same distribution. With these two conventions, most of the ambiguity has been resolved. Although the column vector in M can flip signs, the remaining sign ambiguity hardly raises a concern.

Because the right rotation does not change the magnitude of principle components, the resulting D_a is a function of only Λ and q_L^{-1}. For each q_L^{-1}, bounds of D_a can be calculated easily based on the range of variables in Λ. Intuitively, Λ defines the averaging operation on a Saupe tensor. When Λ corresponds to a broad distribution and the averaging operation is intensive, D_a is small and vice versa. So D_a reaches its maximum and minimum when there is the least and the most averaging, respectively. Suppose the range of variables in Λ is as:

$$\underline{\lambda_2} \leqslant \lambda_2 \leqslant \overline{\lambda_2} \tag{2.38}$$

$$\underline{\lambda_3} \leqslant \lambda_3 \leqslant \overline{\lambda_3} \tag{2.39}$$

$$\underline{\lambda_4} \leqslant \lambda_4 \leqslant \overline{\lambda_4}. \tag{2.40}$$

In Eqs. (2.38-2.40), an underlined symbol represents the parameter's lower limit and an overlined one indicates its upper limit. Following the definition, it can be proved that the averaging operation is least intensive when $(\lambda_2, \lambda_3, \lambda_4) = (\underline{\lambda}_2, \underline{\lambda}_3, \underline{\lambda}_4)$ and it is most intensive when $(\lambda_2, \lambda_3, \lambda_4) = (\overline{\lambda}_2, \overline{\lambda}_3, \overline{\lambda}_4)$. Suppose we have a $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, if λ_2 is increased by Δ, then by definition $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 0$, λ_1 is decreased by Δ. The resulted Λ is $\text{Diag}(\lambda_1 - \Delta, \lambda_2 + \Delta, \lambda_3, \lambda_4)$. In addition, because λ_1 is larger than λ_2, the scaling constant $c^{-1}(\Lambda)$ increases. If we rescale the increase into Λ, the resulted Λ is $\text{Diag}(\lambda_1 - \Delta + \Delta_1, \lambda_2 + \Delta + \Delta_2, \lambda_3 + \Delta_3, \lambda_4 + \Delta_4)$. If λ_2 increases, the absolute value of λ_2 component increases, so does that of λ_3 and λ_4, and the averaging is more intensive. The reverse is also true, when λ_2 decreases, the averaging is less intensive. The same is true for λ_3 and λ_4. Consequently, for each q_L^{-1}, D_a takes its maximum when $(\lambda_2, \lambda_3, \lambda_4) = (\underline{\lambda}_2, \underline{\lambda}_3, \underline{\lambda}_4)$ and it takes its minimum when $(\lambda_2, \lambda_3, \lambda_4) = (\overline{\lambda}_2, \overline{\lambda}_3, \overline{\lambda}_4)$.To fully solve the problem, we also need to know which $q_{_L}^{-1} \in S^3$ makes D_a reach its maximum and minimum. A systematic sampling on one half sphere of S^3 with 4608 samples was used.

In order to strictly follow the definition $\lambda_1 > \lambda_4 > \lambda_3 > \lambda_2$, the branched region in the $\mathbb{R}^3$ space follows the rules below.

$$\overline{\lambda}_2 \leqslant \underline{\lambda}_3 \tag{2.41}$$

$$\overline{\lambda}_3 \leqslant \underline{\lambda}_4 \tag{2.42}$$

$$\overline{\lambda}_4 \leqslant \underline{\lambda}_1. \tag{2.43}$$

Because the above rules need to be followed, some regions violating the rules can not be bounded. We name those regions as *illegal regions*. However, as the branching goes to a deeper level, the total volume of the illegal regions gets smaller(Fig. 2.5). Empirically, the volume of illegal regions can be reduced below a desired value after 6 levels of branching. Consequently, the existence of illegal regions does not break down the provable guarantee nor decrease the accuracy, although it reduces the pruning efficiency to some extent. The 6 levels of branching algorithm is equivalent to a systematic sampling on $\mathbb{R}^3$ with 262144 samples.

Step 2. The 2nd step is very similar to the first step, building bounds over a region in $\mathbb{R}^3 \times S^3_{_L}$ where the bounds are for the largest principle component D_a (Fig. 2.6). The spherical space S^3 is divided into 6912 regions. For a given region in $\mathbb{R}^3 \times S^3_{_L}$, the maximum and minimum are reached when $(\lambda_2, \lambda_3, \lambda_4) = (\underline{\lambda}_2, \underline{\lambda}_3, \underline{\lambda}_4)$ and when $(\lambda_2, \lambda_3, \lambda_4) = (\overline{\lambda}_2, \overline{\lambda}_3, \overline{\lambda}_4)$. For a given region in S^3, the quaternions, $q_{_L}^{max}$ and $q_{_L}^{min}$, that give maximum and minimum D_a can not be determined analytically, so a systematic sampling with 8 samples on the region is used. Because each one of the 6912 subregions on S^3 is small, the systematic sampling with 8 samples is sufficient. After bounds are calculated, regions that do not contain $D_{\mathrm{II},a}$ will be pruned.

At least two independent alignments are necessary to determine a unimodal Bingham distribution. In the first and second pruning steps described in the previous

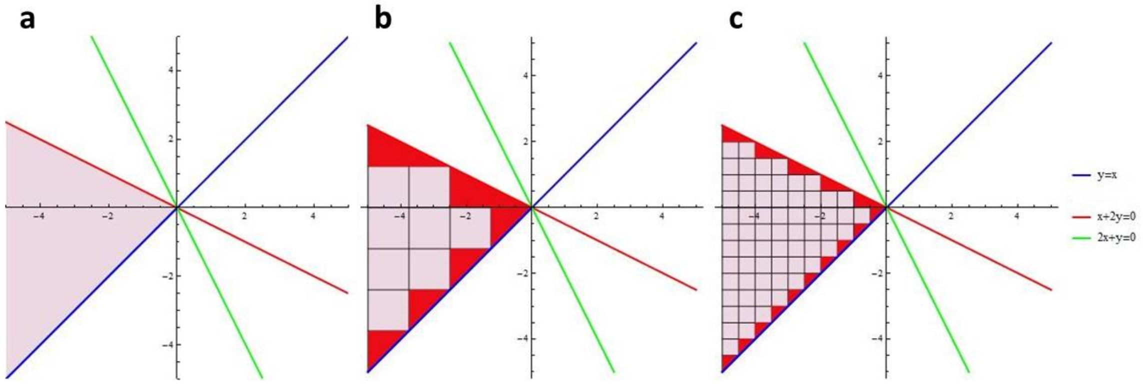

Figure 2.5: Branching the real space decrease the volume of illegal regions. The figure shows an example of the branching. x axis is λ_3, y axis is λ_4, in this example, $\lambda_2 = 0$ and $\lambda_1 = -\lambda_4 - \lambda_3$. (a) the shaded area is allowed by definition $\lambda_1 > \lambda_4 > \lambda_3 > \lambda_2$. (b) the first branching in the allowed area, illegal regions are labeled red. (c) the second branching in the allowed area, area of the illegal regions is smaller.

paragraphs, the magnitudes of principle components of Saupe tensors serve as two constraints. They are applied to the parameter space in $\mathbb{R}^3 \times S^3_{L}$, reducing the number of DOFs from six to four.

Step 3. The third step is a systematic sampling on the remaining four dimensional space. For each sampled point, the resulting Saupe tensors after the left rotation operation and the averaging operation are calculated and rotation matrices of the Saupe tensors are calculated. Suppose the calculated rotation matrices are $R_{q,1}$ and $R_{q,2}$ for the two alignments and suppose the rotation matrices for the Saupe tensors of the second domain are $R_{\text{II},1}$ and $R_{\text{II},2}$. Hence, the right rotation matrices $R_{R,1}$ and $R_{R,2}$ can be calculated using the following equation:

$$R_R = R_q \cdot R_{\text{II}}^T, \tag{2.44}$$

where $R_R = \text{RotM}(\tilde{\mathbf{q}}_{R})$ Two right rotation matrices can be calculated, one for each alignment. If the two rotation matrices agree with each other within a numerical tolerance, the sampled point in $\mathbb{R}^3 \times S^3_{L}$ along with the calculated rotation in S^3_{R} is stored (Fig. 2.7). The stored solutions are sorted by the objective function in

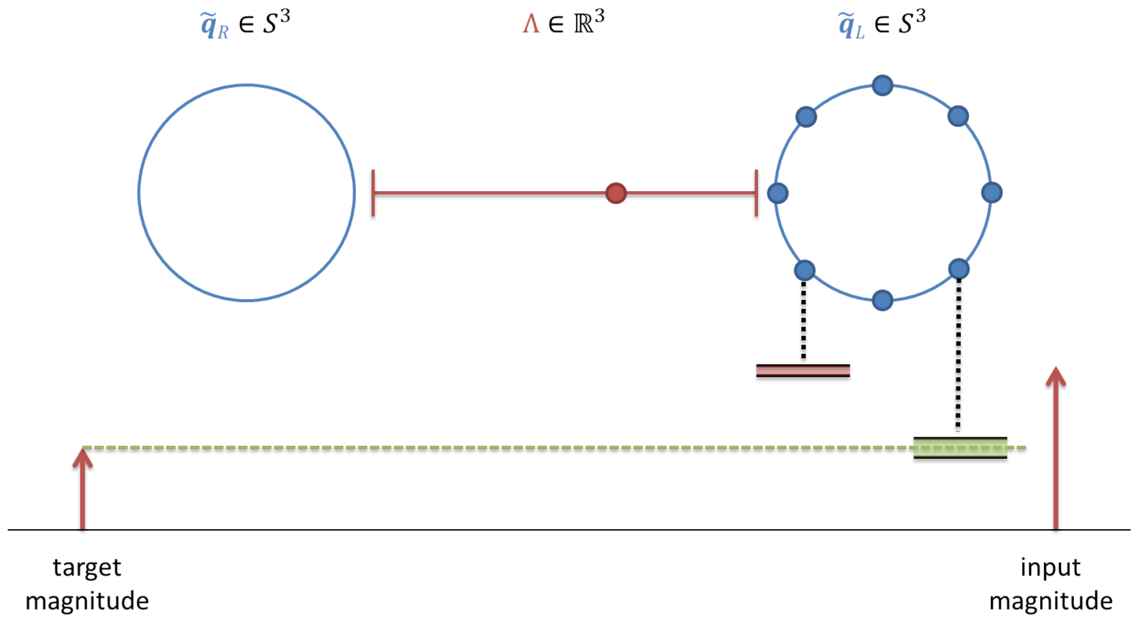
$\widetilde{q}_R \in S^3$
$\Lambda \in \mathbb{R}^3$
$\widetilde{q}_L \in S^3$
target magnitude
input magnitude

FIGURE 2.6: The second step in the branch-and-bound algorithm. For illustration, the S^3 space (on the right) is branched into 8 regions. The upper and lower bound for each region are calculated for each pair of points on the $\mathbb{R}^3$ space and the S^3 space . The red arrow on the right represents the magnitude of the largest principle component of the first Saupe tensor S_{I}. The red arrow on the left represents the magnitude of the largest principle component of the second Saupe tensor S_{II}. Regions (green) covering the target magnitude is saved for the next round of branch-and-bound. Other regions (red) are pruned.

Eq. (2.45) to find the best Bingham distribution satisfying the data:

$$f(\mathbf{x}) = \sum_i (\mathbf{E}_{R\sim P(R\,;\,X)}[Q(R)] \cdot \mathbf{s}_{\text{I,i}} - \mathbf{s}_{\text{II,i}})^2. \qquad (2.45)$$

The whole method is summarized in Fig. 2.8.

2.4 Generalization to the case when the reference domain is switched

In the previous discussions, we assume the reference domain is always Domain I in different alignments. However, experimental settings could be different from case to case. For example, in the TAR RDC datasets, Domain I is the reference in the first alignment while Domain II is the reference in the second alignment. In this section,

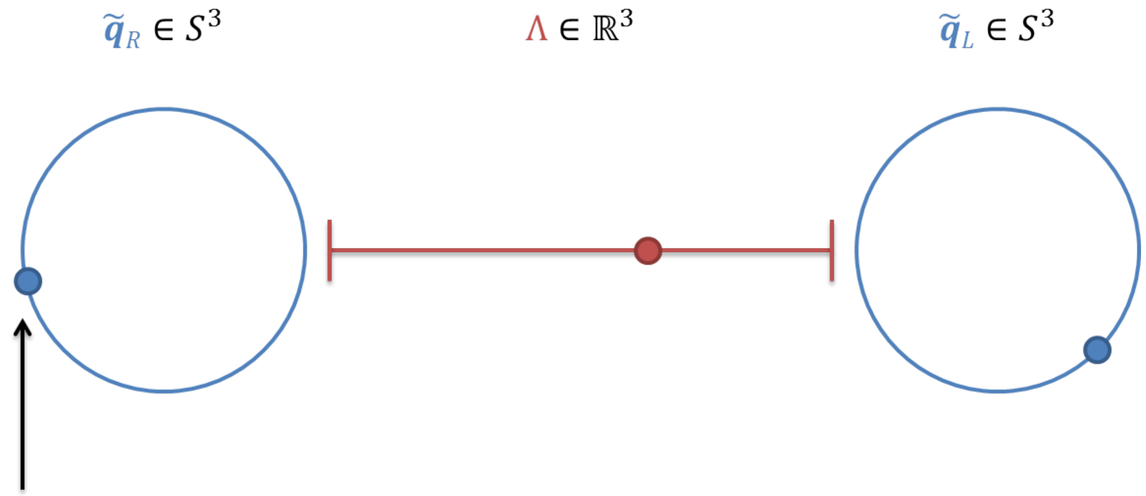

FIGURE 2.7: The third step in the branch-and-bound algorithm. The point on the S^3 space of $\tilde{\mathbf{q}}_\mathbf{R}$ can be directly calculated.

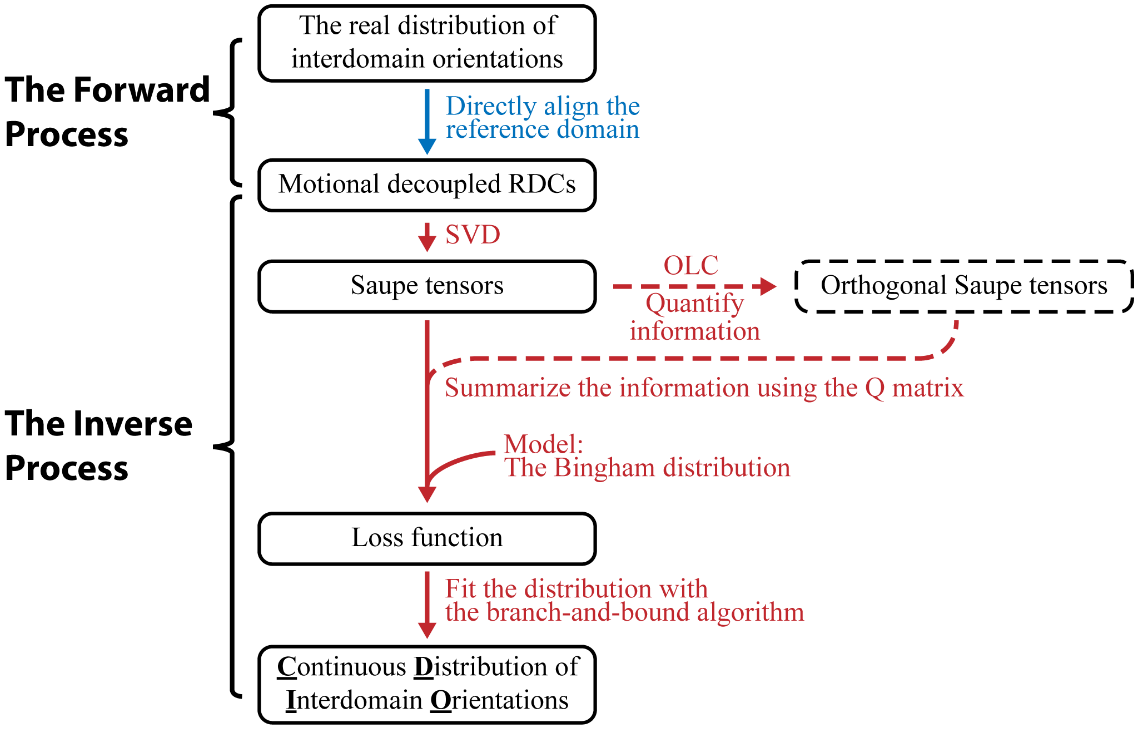
The Forward Process
The real distribution of interdomain orientations
Directly align the reference domain
Motional decoupled RDCs
The Inverse Process
SVD
Saupe tensors
OLC
Quantify information
Orthogonal Saupe tensors
Summarize the information using the Q matrix
Model: The Bingham distribution
Loss function
Fit the distribution with the branch-and-bound algorithm
Continuous Distribution of Interdomain Orientations

FIGURE 2.8: A flow chart of the experimental and computational steps used to calculate the continuous distribution of interdomain orientations. The loss function is the objective function in Eq. (2.45)

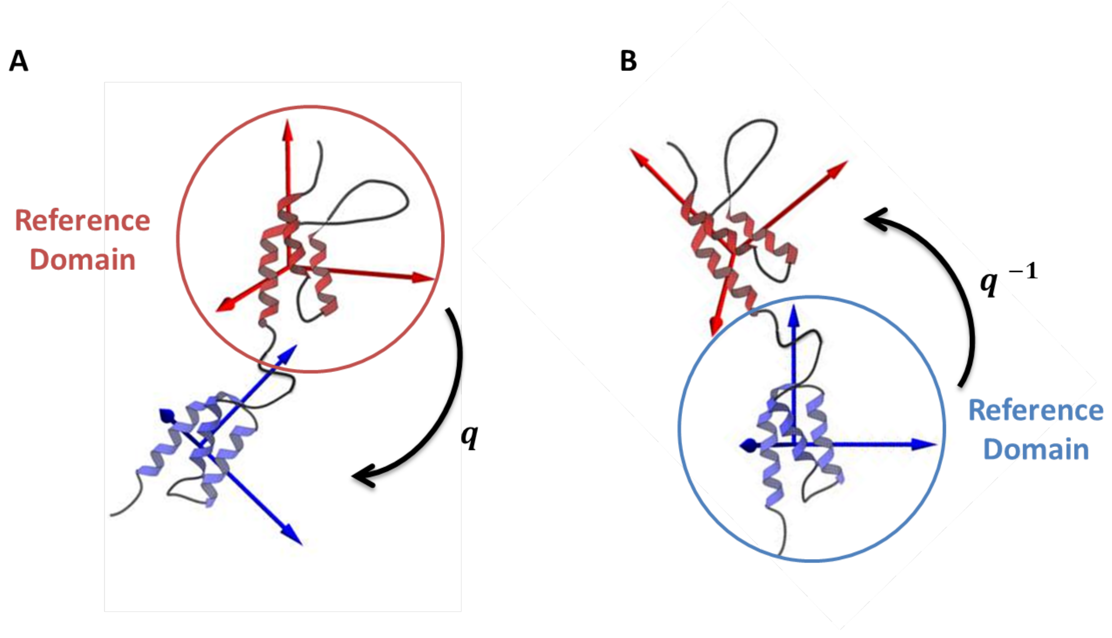

Figure 2.9: The interdomain conformation of A and B are the same. A. the reference domain is the top red domain and the rotation from the reference domain to the other domain is q. B. the reference domain is the bottom blue domain and the rotation from the reference domain to the other domain is q^{-1}.

I present a generalization of the method which can deal with the above experimental settings. Although parameterization of the interdomain orientation distributions depends on the choice of reference domain, a simple relationship between the two sets of parameters can be derived, which enables the branch-and-bound algorithm to work for this dataset as well.

Because the interdomain motions are the same in the two alignments, the probabilities corresponding to one particular interdomain orientation must equal each other. Assuming an interdomain orientation is represented by a rotation R in the first alignment, the orientation is represented by its inverse rotation R^T in the second alignment, because of the change of reference domain, as shown in Fig. 2.9. Consequently, probability $P_I(R)$ in the first distribution equals to $P_{II}(R^{-1})$ in the second distribution for any given rotation R. Because rotations can also be represented by quaternions, we have the following equation:

$$P_{_I}(\tilde{\mathbf{q}}) = P_{_{II}}(\tilde{\mathbf{q}}^{-1}). \tag{2.46}$$

The above equation generally holds without any assumption. When Bingham distribution is introduced to model the interdomain motions [15, 16], we have:

$$P_{_I}(\tilde{\mathbf{q}}) = c^{-1}(X_I)\exp(\tilde{\mathbf{q}}^T X_I \tilde{\mathbf{q}}), \tag{2.47}$$

$$P_{_{II}}(\tilde{\mathbf{q}}^{-1}) = c^{-1}(X_{II})\exp((\tilde{\mathbf{q}}^{-1})^T X_{II} \tilde{\mathbf{q}}^{-1}). \tag{2.48}$$

Based on Eq. (2.46), we know the second distribution is an axially inverted image of the first one. So the scaling factors $c^{-1}(X_I)$ and $c^{-1}(X_{II})$ are the same. Consequently, we have:

$$\tilde{\mathbf{q}}^T X_I \tilde{\mathbf{q}} = (\tilde{\mathbf{q}}^{-1})^T X_{II} \tilde{\mathbf{q}}^{-1}. \tag{2.49}$$

From Eq. (2.21), X can be decomposed as $M^T \Lambda M$, we have:

$$\tilde{\mathbf{q}}^T M_I^T \Lambda_I M_I \tilde{\mathbf{q}} = (\tilde{\mathbf{q}}^{-1})^T M_{II}^T \Lambda_{II} M_{II} \tilde{\mathbf{q}}^{-1}. \tag{2.50}$$

If we define $\tilde{\mathbf{q}}$ as $\tilde{\mathbf{q}} = [q_1, q_2, q_3, q_4]^T$, then $\tilde{\mathbf{q}}^{-1} = [q_1, -q_2, -q_3, -q_4]^T$. If we further define M as $M = [m_1, m_2, m_3, m_4]$ where each m_i is a column vector, the above equation can be written as:

$$\tilde{\mathbf{q}}^T M_I^T \Lambda_I M_I \tilde{\mathbf{q}} = \tilde{\mathbf{q}}^T [m_{_{II,1}}, -m_{_{II,2}}, -m_{_{II,3}}, -m_{_{II,4}}]^T \Lambda_{II} [m_{_{II,1}}, -m_{_{II,2}}, -m_{_{II,3}}, -m_{_{II,4}}] \tilde{\mathbf{q}}. \tag{2.51}$$

Consequently, we have:

$$M_I^T \Lambda_I M_I = [m_{_{II,1}}, -m_{_{II,2}}, -m_{_{II,3}}, -m_{_{II,4}}]^T \Lambda_{II} [m_{_{II,1}}, -m_{_{II,2}}, -m_{_{II,3}}, -m_{_{II,4}}]. \tag{2.52}$$

From the above equation, it seems that we could derive relationships between Λ_I and Λ_{II} and between M_I and M_{II}. Indeed, we could get the relationships after straighting out a minor issue. In Eq. (2.52), after changing the signs of three row vectors in rotation matrix M_{II}, the determinant becomes -1 and the resulted matrix is no longer a rotation matrix. In order to change the determinant back to 1, we could change the signs of the last three column vectors. The resulted matrix is

$$M'_{II} = \begin{bmatrix} m_{II,11} & -m_{II,12} & -m_{II,13} & -m_{II,14} \\ -m_{II,21} & m_{II,22} & m_{II,23} & m_{II,24} \\ -m_{II,31} & m_{II,32} & m_{II,33} & m_{II,34} \\ -m_{II,41} & m_{II,42} & m_{II,43} & m_{II,44} \end{bmatrix}. \tag{2.53}$$

In addition, the following equation still holds:

$$M_I \Lambda_I M_I^T = M'_{II} \Lambda_{II} (M'_{II})^T. \tag{2.54}$$

From the above equation, we can derive some simple relationships: $\Lambda_{II} = \Lambda_I$ and $M'_{II} = M_I$. Consequently, we have:

$$M_{II} = \begin{bmatrix} m_{II,11} & m_{II,12} & m_{II,13} & m_{II,14} \\ m_{II,21} & m_{II,22} & m_{II,23} & m_{II,24} \\ m_{II,31} & m_{II,32} & m_{II,33} & m_{II,34} \\ m_{II,41} & m_{II,42} & m_{II,43} & m_{II,44} \end{bmatrix} = \begin{bmatrix} m_{I,11} & -m_{I,12} & -m_{I,13} & -m_{I,14} \\ -m_{I,21} & m_{I,22} & m_{I,23} & m_{I,24} \\ -m_{I,31} & m_{I,32} & m_{I,33} & m_{I,34} \\ -m_{I,41} & m_{I,42} & m_{I,43} & m_{I,44} \end{bmatrix}. \tag{2.55}$$

As discussed in section 2.2, matrix M in the Bingham distribution can be decomposed as a product of one left rotation and one right rotation [19]. Both left and right rotations are isoclinic rotations in 4D space and can be represented as functions of a quaternion. The decomposition of M_I is shown in the following equation:

$$M_I = M_L(\tilde{\mathbf{q}}_{\mathrm{I},L}) M_R(\tilde{\mathbf{q}}_{\mathrm{I},R}), \tag{2.56}$$

where $M_L(\tilde{\mathbf{q}})$ and $M_R(\tilde{\mathbf{q}})$ are two 4×4 matrix generated by the unit quaternion $\tilde{\mathbf{q}}$. If we define $\tilde{\mathbf{q}} \equiv [q_1, q_2, q_3, q_4]^T$, we have:

$$M_L(\tilde{\mathbf{q}}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix}, \tag{2.57}$$

$$M_L(\tilde{\mathbf{q}}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix}. \tag{2.58}$$

Assuming $\tilde{\mathbf{q}}_{\mathrm{I},L} = [a, b, c, d]^T$ and $\tilde{\mathbf{q}}_{\mathrm{I},R} = [p, q, r, s]^T$, then according to Mebius [19], we have:

$$\begin{aligned} M_I &= \begin{bmatrix} m_{I,11} & m_{I,12} & m_{I,13} & m_{I,14} \\ m_{I,21} & m_{I,22} & m_{I,23} & m_{I,24} \\ m_{I,31} & m_{I,32} & m_{I,33} & m_{I,34} \\ m_{I,41} & m_{I,42} & m_{I,43} & m_{I,44} \end{bmatrix} \\ &= \begin{bmatrix} ap-bq-cr-ds & -aq-bp+cs-dr & -ar-bs-cp+dq & -as+br-cq-dp \\ aq+bp+cs-dr & ap-bq+cr+ds & as-br-cq-dp & -ar-bs+cp-dq \\ ar-bs+cp+dq & -as-br-cq+dp & ap+bq-cr+ds & aq-bp-cs-dr \\ as+br-cq+dp & ar-bs-cp-dq & -aq+bq-cs-dr & ap+bq+cr-ds \end{bmatrix}, \end{aligned} \tag{2.59}$$

$$\begin{aligned} M_{II} &= \begin{bmatrix} m_{I,11} & -m_{I,12} & -m_{I,13} & -m_{I,14} \\ -m_{I,21} & m_{I,22} & m_{I,23} & m_{I,24} \\ -m_{I,31} & m_{I,32} & m_{I,33} & m_{I,34} \\ -m_{I,41} & m_{I,42} & m_{I,43} & m_{I,44} \end{bmatrix} \\ &= \begin{bmatrix} ap-bq-cr-ds & aq+bp-cs+dr & ar+bs+cp-dq & as-br+cq+dp \\ -aq-bp-cs+dr & ap-bq+cr+ds & as-br-cq-dp & -ar-bs+cp-dq \\ -ar+bs-cp-dq & -as-br-cq+dp & ap+bq-cr+ds & aq-bp-cs-dr \\ -as-br+cq-dp & ar-bs-cp-dq & -aq+bq-cs-dr & ap+bq+cr-ds \end{bmatrix}. \end{aligned} \tag{2.60}$$

A general method to decompose 4D rotation matrix M is to calculate its associate matrix A as shown in the following equation [19]:

$$A = \tfrac{1}{4} \begin{bmatrix} m_{11}+m_{22}+m_{33}+m_{44} & m_{21}-m_{12}-m_{43}+m_{34} & m_{31}+m_{42}-m_{13}-m_{24} & m_{41}-m_{32}+m_{23}-m_{14} \\ m_{21}-m_{12}+m_{43}-m_{34} & -m_{11}-m_{22}+m_{33}+m_{44} & m_{41}-m_{32}-m_{23}+m_{14} & -m_{31}-m_{42}-m_{13}-m_{24} \\ m_{31}-m_{42}-m_{13}+m_{24} & -m_{41}-m_{32}-m_{23}-m_{14} & -m_{11}+m_{22}-m_{33}+m_{44} & m_{21}+m_{12}-m_{43}-m_{34} \\ m_{41}+m_{32}-m_{23}-m_{14} & m_{31}-m_{42}+m_{13}-m_{24} & -m_{21}-m_{12}-m_{43}-m_{34} & -m_{11}+m_{22}+m_{33}-m_{44} \end{bmatrix}. \tag{2.61}$$

We know M_I is constructed from two rotations and we know the decomposition. As an example, we could use Eq. (2.61) to obtain the decomposition and to get the left rotation quaternion and the right rotation quaternion.

$$A_I = \begin{bmatrix} ap & aq & ar & as \\ bp & bq & br & bs \\ cp & cq & cr & cs \\ dp & dq & dr & ds \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \begin{bmatrix} p & q & r & s \end{bmatrix}. \tag{2.62}$$

Similarly, we could decompose M_{II} using Eq. (2.61).

$$A_{II} = \begin{bmatrix} pa & -pb & -pc & -pd \\ -qa & qb & qc & qd \\ -ra & rb & rc & rd \\ -sa & sb & sc & sd \end{bmatrix} = \begin{bmatrix} p \\ -q \\ -r \\ -s \end{bmatrix} \begin{bmatrix} a & -b & -c & -d \end{bmatrix}. \tag{2.63}$$

and

$$M_{II} = M_L(\tilde{\mathbf{q}}_{\mathrm{I},R}^{-1}) M_R(\tilde{\mathbf{q}}_{\mathrm{I},L}^{-1}). \tag{2.64}$$

Consequently,

$$\tilde{\mathbf{q}}_{\mathrm{II},L} = \tilde{\mathbf{q}}_{\mathrm{I},R}^{-1} \tag{2.65}$$

and

$$\tilde{\mathbf{q}}_{\mathrm{II},R} = \tilde{\mathbf{q}}_{\mathrm{I},L}^{-1}. \tag{2.66}$$

Following the discussions in section 2.2, the Q matrix could be decomposed based on the decomposition of X and M. For the first alignment with the reference Domain I, we have the following relationship:

$$\mathbf{s}_{\mathrm{II}} = Q(\tilde{\mathbf{q}}_{\mathrm{I},R}^{-1}) \cdot \mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\mathrm{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})] \cdot Q(\tilde{\mathbf{q}}_{\mathrm{I},L}^{-1}) \cdot \mathbf{s}_{\mathrm{I}}. \tag{2.67}$$

For the second alignment with reference Domain II, we have:

$$\mathbf{s}_{\mathrm{I}}{}' = Q(\tilde{\mathbf{q}}_{\mathrm{II},R}^{-1}) \cdot \mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\mathrm{II}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})] \cdot Q(\tilde{\mathbf{q}}_{\mathrm{II},L}^{-1}) \cdot \mathbf{s}_{\mathrm{II}}{}'. \tag{2.68}$$

From Eq. (2.52), we know $\Lambda_{II} = \Lambda_I$. From Eq. (2.65) and Eq. (2.66), we have $\tilde{\mathbf{q}}_{\mathrm{II},L} = \tilde{\mathbf{q}}_{\mathrm{I},R}^{-1}$ and $\tilde{\mathbf{q}}_{\mathrm{I},R} = \tilde{\mathbf{q}}_{\mathrm{II},L}^{-1}$. Consequently, Eq. (2.68) can be written as:

$$\mathbf{s}_{\mathrm{I}}{}' = Q(\tilde{\mathbf{q}}_{\mathrm{I},L}) \cdot \mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\mathrm{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})] \cdot Q(\tilde{\mathbf{q}}_{\mathrm{I},R}) \cdot \mathbf{s}_{\mathrm{II}}{}'. \tag{2.69}$$

In the above two equations, $\mathbf{s}_{\text{I}}$ and $\mathbf{s}_{\text{II}}$ are vectorized Saupe tensor of Domain I and Domain II in the first alignment, $\mathbf{s}_{\text{I}}{}'$ and $\mathbf{s}_{\text{II}}{}'$ are vectorized Saupe tensor of Domain I and Domain II in the second alignment. Based on Eq. (2.69), we could easily derive

$$\mathbf{s}_{\text{II}}{}' = Q(\tilde{\mathbf{q}}_{\text{I},R}^{-1}) \cdot \mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]^{-1} \cdot Q(\tilde{\mathbf{q}}_{\text{I},L}^{-1}) \cdot \mathbf{s}_{\text{I}}{}'. \tag{2.70}$$

From Eq. (2.67) and Eq. (2.70), the mathematical expression for switching the reference domain is almost the same as using a different alignment with the same reference domain. The only difference is the middle matrix $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]^{-1}$, which is the inverse of the averaging matrix $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]$. Consequently, the middle matrix acts like a concentrating operator and increases the eigenvalues of a Saupe tensor. The increase on eigenvalues is expected because $\mathbf{s}_{\text{I}}{}'$ is an averaging product of $\mathbf{s}_{\text{II}}{}'$ and $\mathbf{s}_{\text{II}}{}'$ should have larger eigenvalues. Additional functions have been implemented to calculate the inverse $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]^{-1}$. According to Eq. (2.37), the matrix $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]$ only have seven non-zero entries. If we define the element in position i, j in matrix $\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]$ as m_{ij}, we have:

$$\mathbf{E}_{\tilde{\mathbf{q}}_{\mathbf{Y}} \sim P(\tilde{\mathbf{q}}_{\mathbf{Y}}\,;\,\Lambda_{\text{I}})}[Q(\tilde{\mathbf{q}}_{\mathbf{Y}})]^{-1} = \begin{bmatrix} \frac{m_{22}}{m_{11}m_{22}-m_{12}m_{21}} & -\frac{m_{12}}{m_{11}m_{22}-m_{12}m_{21}} & 0 & 0 & 0 \\ -\frac{m_{21}}{m_{11}m_{22}-m_{12}m_{21}} & \frac{m_{11}}{m_{11}m_{22}-m_{12}m_{21}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m_{33}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m_{44}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{m_{55}} \end{bmatrix}. \tag{2.71}$$

With the minor revision, the brand and bound algorithm is able to find the best solution satisfying Eq. (2.67) and Eq. (2.70).

3

Results of the branch-and-bound method

3.1 Performance of the method with respect to noise

In order to test the performance of the method, I first observed the stability of the solution under various levels of noise. The RDC data is simulated from an arbitrary Bingham distribution. The Bingham distribution is considered as the ground truth. The simulation is a forward process as shown in Fig. 1.3. Given a Bingham distribution and two arbitrary orthogonal Saupe tensors of Domain I, the corresponding Saupe tensors of Domain II can be calculated using Eq. 2.33. Using the Saupe tensors of Domain I and Domain II, the RDCs can be calculated by Eq. 2.1. The bond vector $\mathbf{v}$ for each RDC is sampled from a uniform distribution on the sphere S^2. Then different levels of noise are added into the simulated RDCs. As in any real experiment, only the RDCs with noise are available to us after the forward simulation.

Because we assume we have no access to the noise-free Saupe tensors and the ground truth distribution, Saupe tensors are calculated from the noisy data using the SVD method mentioned in section 2.1. The level of noise is quantified by the Q factor:

$$Q = \frac{\sqrt{\sum_{i=1}^{N}(D_i^{cal} - D_i^{exp})^2}}{\sqrt{\sum_{i=1}^{N}(D_i^{exp})^2}}. \tag{3.1}$$

Here, D_i^{exp} and D_i^{cal} are supposed to be the i-th experimental RDC and the i-th RDC back-calculated from the calculated Saupe tensors, respectively. In our case, D_i^{exp} is the i-th RDC generated by the forward simulation. The Q factor associated with each dataset reflects the level of noise in the RDC data. The Q factor can be considered as a measure of the noise to signal ratio. The noise level is generally considered low when the Q factor is below 0.3 [20].

The solution corresponding to the calculated Saupe tensors is calculated by the branch-and-bound algorithm. The solution will be compared to the ground truth. The difference between the calculated solution and the ground truth is measured by the square root of the Jensen-Shannon Divergence $\sqrt{JSD}$, which is a metric satisfying the triangular inequality [21]. Jensen-Shannon Divergence JSD is calculated by the following equation:

$$JSD(p(x)||q(x)) = \frac{1}{2}D_{KL}(p(x)||m(x)) + \frac{1}{2}D_{KL}(q(x)||m(x)), \tag{3.2}$$

where $m(x) = \frac{1}{2}(p(x) + q(x))$ and D_{KL} is the Kullback-Leibler divergence in Eq. 4.6.

Although the branch-and-bound algorithm is deterministic, the forward simulation is stochastic. In addition, the performance of the method should be tested with a variety of ground truth distributions. Consequently, the forward simulation was repeated ten times for each noise level. The test results is summarized in Fig. 3.1.

In Fig. 3.1, the maximum, upper quartile, median, lower quartile and minimum of $\sqrt{JSD}$ are reported for each noise level. The noise level ranges from $Q = 0$ to $Q = 0.5$. In the ideal case, when no noise is present, all the calculated solutions are nearly the same as their corresponding ground truth. The result indicates the

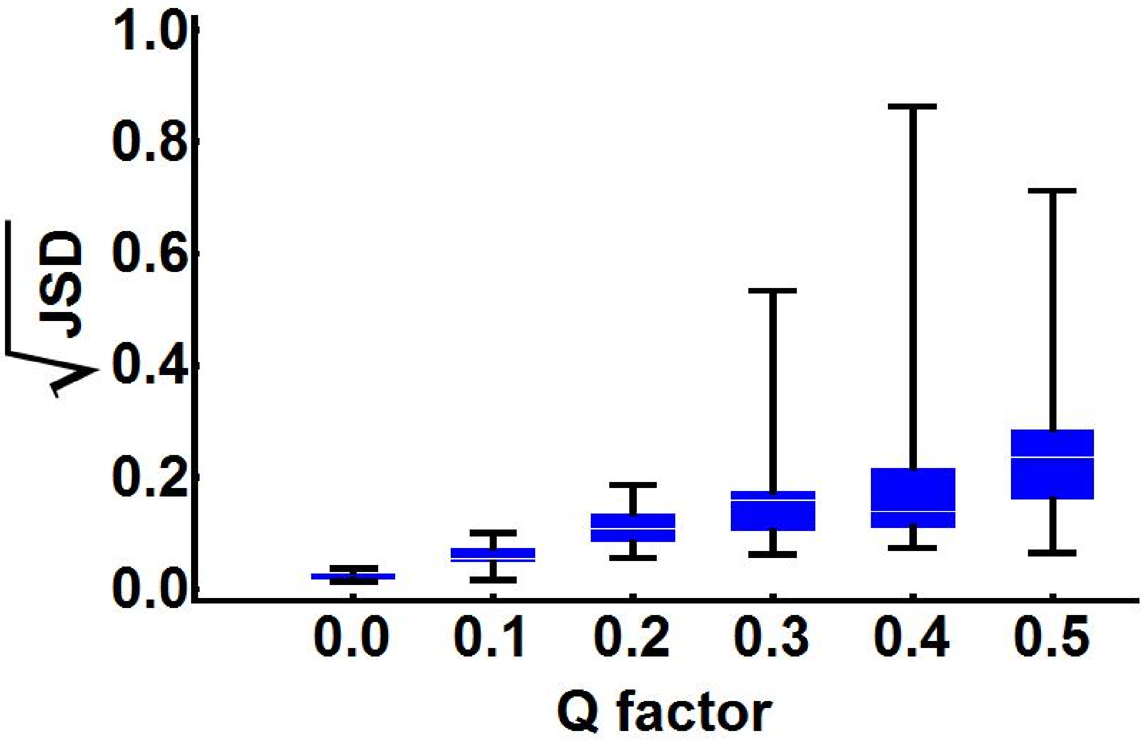

FIGURE 3.1: The performance of the branch-and-bound method with respect to various noise levels in a box-whisker plot.

method works as expected. With the increase of the noise, the calculated solution deviates more significantly from the ground truth. Although the $\sqrt{JSD}$ maximums are above 0.5 when $Q = 0.3, 0.4, 0.5$, the upper quartile remains around 0.2. It indicates the method may still be able to find a good solution from a very noisy dataset. In the noise level range from $Q = 0$ to $Q = 0.2$, the method can find a good solution very close to the ground truth. As a guideline, datasets with noise level below or around $Q = 0.2$ should be used.

3.2 Application to SpA-N

Staphylococcal protein A (SpA) is a key virulence factor that supports the invasion of *Staphylococcus aureus* into the human body. SpA binds to an array of targets in the host to disarm the immune system, facilitate the colonization and consequently contribute to the pathogenicity of *S. aureus*. The emergence of antibiotic-resistant *S. aureus* strains has driven the search for vaccines with high efficacy to counteract several virulence factors, including SpA [22]. Targeting virulence as opposed to bacterial cell growth may be a more effective way to avoid the emergence of resistant strains. Structural studies of SpA and its interaction with host proteins would support rational design of improved vaccines or other therapeutics to diminish *S. aureus* virulence. The binding targets of SpA include the $\mathrm{F_c}$ region of antibodies, the $\mathrm{F_{ab}}$ region of $\mathrm{V_H}$3 antigen receptors (e.g., IgM) on B cells, TNFR1 and EGFR [23, 24, 25].

There are five tandem functional domains in the N-terminal half of SpA (SpA-N). The five domains share a high sequence identity and they are structurally and functionally similar [26, 27, 24]. Recent studies indicate a correlation between the functional plasticity and the structural flexibility of SpA-N. Conformational heterogeneity of individual domains was observed by high-resolution X-ray crystallography [26, 28]. The backbone and side-chain dynamics presumably help SpA-N adapt to the binding interfaces of multiple partners. The interdomain dynamics were measured by small-angle X-ray scattering (SAXS) experiments [29]. However, SAXS experiments provide limited information about the distance distribution and they are insensitive to the orientational change during interdomain motions. Consequently, the interdomain motion, especially its orientational component, remains poorly understood.

I applied the branch-and-bound method the motional decoupled RDC data of SpA-N and found the best-fit solutions to the data. To describe the solutions, it is first necessary to define the molecular frame of each domain. In the coordinate frame of either Z or C domain, the z-axis of the domain is parallel to the helices and points toward the N-termini of helices 1 and 3 , the y-axis is in the plane of helix II and III, and the x-axis is perpendicular to the plane (Fig. 3.2). The 3D joint interdomain orientational distribution can be visualized in a *disk-on-sphere (DoS)* representation (Fig. 3.2). In this representation, the x, y, z-axes in Fig. 3.2 represent

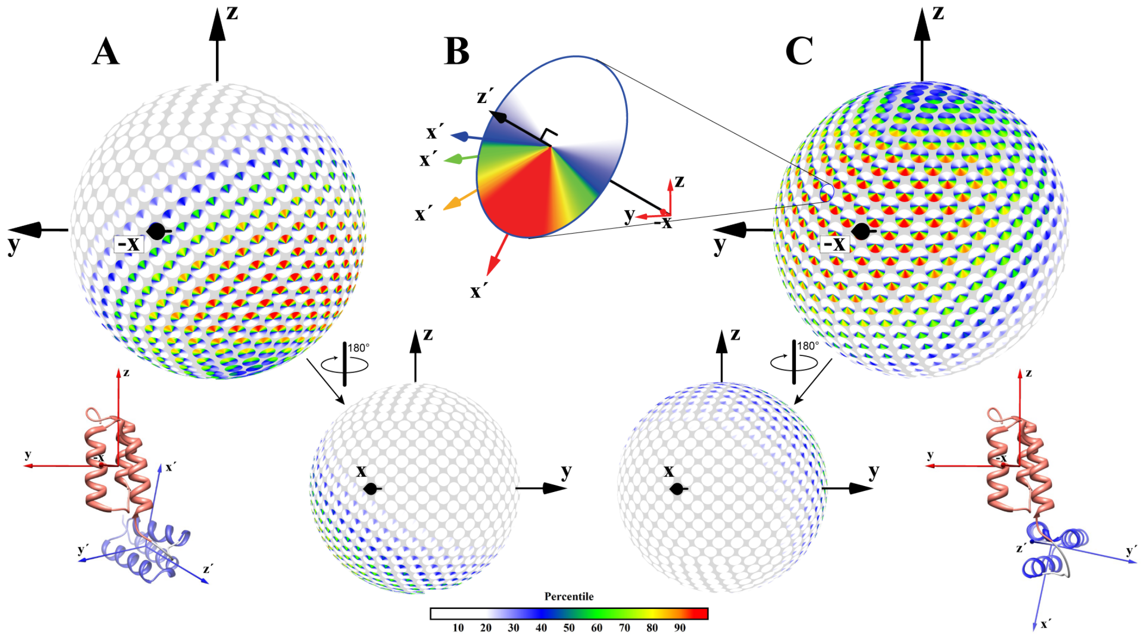

FIGURE 3.2: Continuous distribution of interdomain orientations models for ZLBT-C shown in disk-on-sphere (DoS) views. Two solutions (shown in panels A and C) give equivalently good fits to the data. Each panel shows the distribution from a front view (top) and a back view (bottom). Also shown are atomistic models whose interdomain orientation is the most probable in each solution. The linker conformation and interdomain distance are arbitrary. (B) An example disk showing the joint probabilities of 4 different different interdomain orientations, all with the same z' axis orientation but different rotations around z'.

the coordinate frame of the reference domain (Z domain) and each radial line on a disk corresponds to a unique interdomain orientation. The location of the disk on the sphere corresponds to the orientation of the z-axis of the C domain, while the direction of the radial line corresponds to the x-axis of the C domain. The representation is color coded, with red the highest probability and white the lowest probability.

There are two solutions that can reproduce the RDC data nearly equally well (Fig. 3.3). A priori, it is possible that any linear combination of the two solutions could also fit the data. However, when we simulated a structural ensemble of the B-C di-domain molecule in the absence of any binding partner using the RANCH component of the EOM package [30], a strong disagreement was found between the

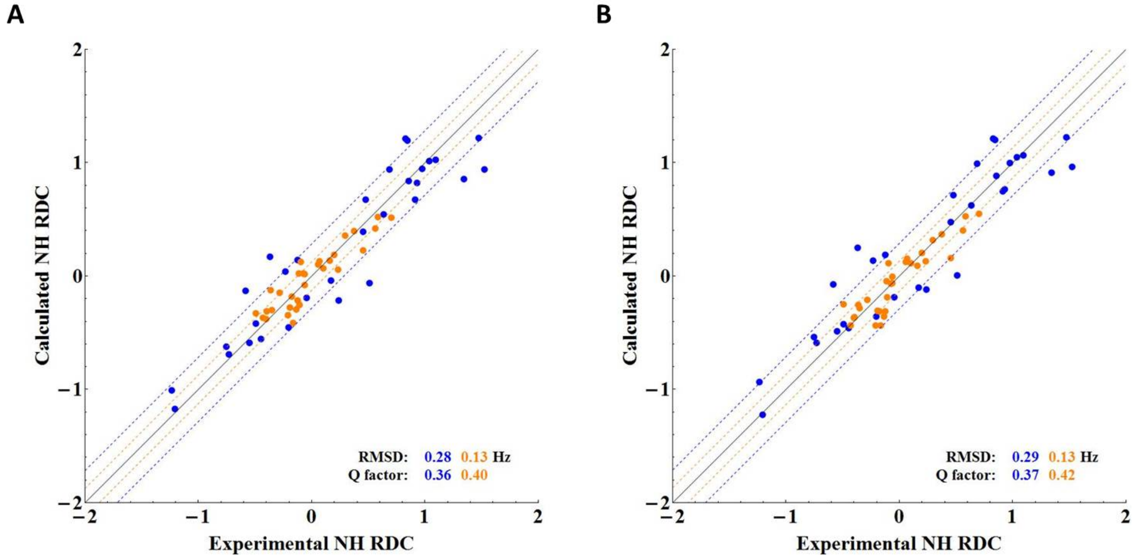

FIGURE 3.3: RDC correlation plots. The correlation plot of the first OLC dataset(blue) and the second OLC dataset(yellow) of Z domain and the RDCs back-calculated from solution 1(A) and solution 2(B).

first solution described above and the structural ensembles generated by RANCH. The DoS plot of the simulation shows a large void of probability in the white area (Fig. 3.4), suggesting that most conformers in this area have steric clashes. The distribution mode of the first solution largely coincides with this low probability region and 37.3% of its probability falls into the region (Fig. 3.4). Quantitatively, we defined a clash score based on the probability of a distribution in the low probability region and calculated clash scores for both solutions. The clash score of the first solution is more than two fold higher than that of the second solution. By thermodynamic criteria, the simulated ensemble is also in better agreement with the second solution than the first solution or any linear combination of the two. These results suggest that the first solution is a so-called "ghost" solution [1], but further studies with additional orthogonal alignments are necessary to definitively rule it out.

In the second solution, the most probable interdomain orientation has the z-axis of C domain close to the minus x-axis of Z domain and the x-axis of C domain close to the minus z-axis of Z domain. It should be noted that the marginal distribution of

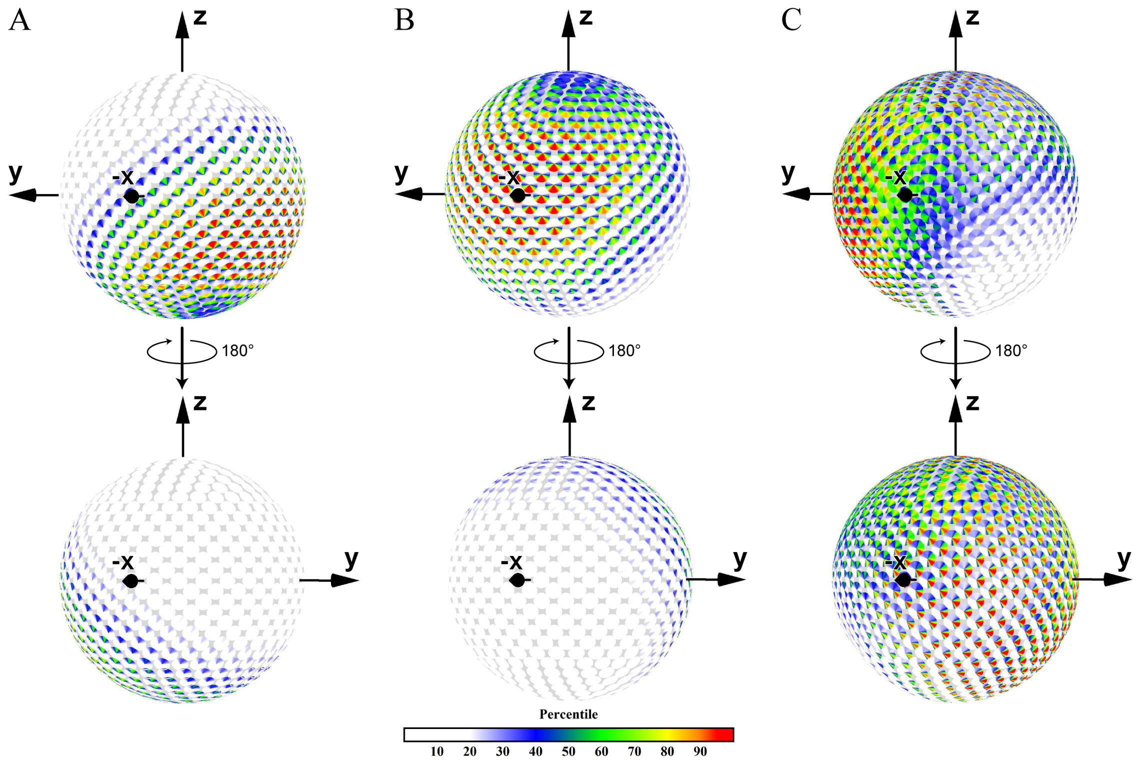

FIGURE 3.4: Comparing calculated interdomain orientational distribution with the simulated di-domain orientational distribution. The first(A) and the second(B) best calculated interdomain orientational distributions and the simulated interdomain orientational distribution(C) in the disk representation. The bottom of each panel shows the distribution from an opposite view. The color goes from red, yellow, green, blue to white as probability decreases.

the z-axis is relatively broad but the marginal distribution of the x-axis is narrow. In Fig. 3.2, the region bounded by the green color takes 9.7% of the entire interdomain orientational space and it has a probability of 40%. The joint distribution suggests that the flexible linker enables the two domains to sample a relatively large range of the interdomain orientational space but somewhat restricts the orientation of the C domain's x-axis.

3.3 Application to calmodulin

Using the NMR data of calmodulin [1], I calculated the alignment tensors of both domains, which were subsequently fed into the branch-and-bound algorithm to search for optimal solutions. The first four optimal solutions were compared to the four conformations with highest maximum allowed probabilities(MAPs) in the 2007 JACS paper [1]. More specifically, the inter-domain orientation corresponding to the maximum point of each optimal distribution was used for comparison. The four highest-probability inter-domain orientations generally agree with the four high-MAP conformations, suggesting a consistency between the two methods.

In the 2007 JACS paper, an approach was presented to determine the maximum allowed probabilities(MAPs) of inter-domain orientations [1]. The MAP value is not the probability of an inter-domain orientation, instead, it represents the maximum probability of that orientation in any distribution that satisfies the experimental constraints. In other words, the MAP value is a measure that represents the information derived from experimental observables, but without a probabilistic interpretation. In the calmodulin case, the distribution of MAP value has four modes. The inter-domain conformations corresponding to local maxima of the four modes are shown in Fig.3.5.

In the RDC and PCS dataset of calmodulin [1], PCSs were measured for the N-terminal domain and RDCs were measured for the C-terminal domain. Although PCS and RDC are different physical observables, both were induced by the magnetic alignment of lanthanide ions and thus they share the same alignment tensor. Alignment tensors of the N-terminal domain were calculated using the PCSs while alignment tensors of the C-terminal domain were calculated using the RDCs. All tensors were calculated using a SVD method following the protocol described in [14]. Axial and rhombic components of the alignment tensors are summarized in Table 3.1 and Table 3.2. The determined alignment tensors are consistent with the reported alignment tensors in the 2004 PNAS paper and the 2007 JACS paper [31, 1]. Experimental RDCs/PCSs were plotted against the back-calculated ones in Fig.3.6 and Fig.3.7.

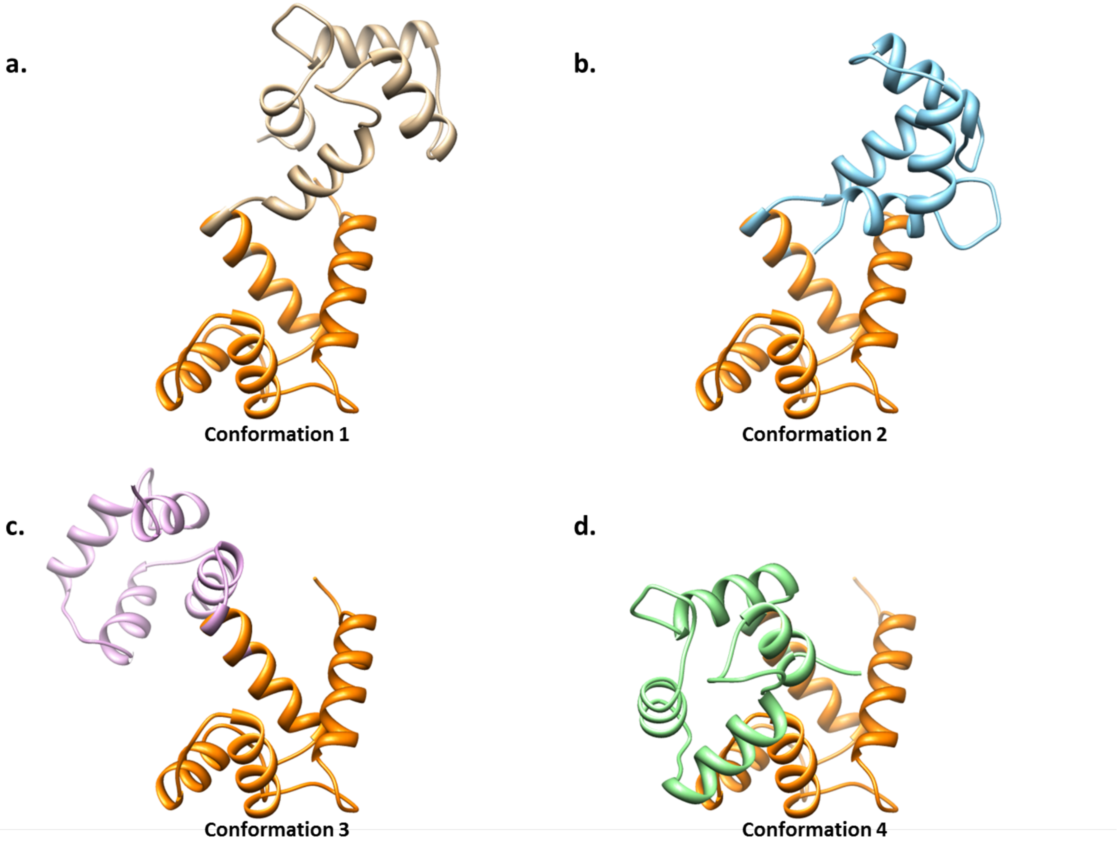

FIGURE 3.5: N-terminal domains of the four conformations are shown as orange ribbons. The C-terminal domains are shown as brown, cyan, purple and green ribbons, respectively. A-D. the four high-MAP conformations.

Table 3.1: Axial and rhombic components of N-terminal domain of calmodulin

	Dy^{3+}	Tb^{3+}	Tm^{3+}	
$\Delta\chi_{ax}$	34.78	35.54	24.75	$\times 10^{-32} m^3$
$\Delta\chi_{rh}$	-15.10	-12.42	-8.42	$\times 10^{-32} m^3$

Table 3.2: Axial and rhombic components of C-terminal domain of calmodulin

	Dy^{3+}	Tb^{3+}	Tm^{3+}	
$\Delta\chi_{ax}$	-2.05	-1.85	-3.74	$\times 10^{-32} m^3$
$\Delta\chi_{rh}$	0.85	0.93	2.46	$\times 10^{-32} m^3$

The calculated alignment tensors were subsequently fed into the fitting algorithm

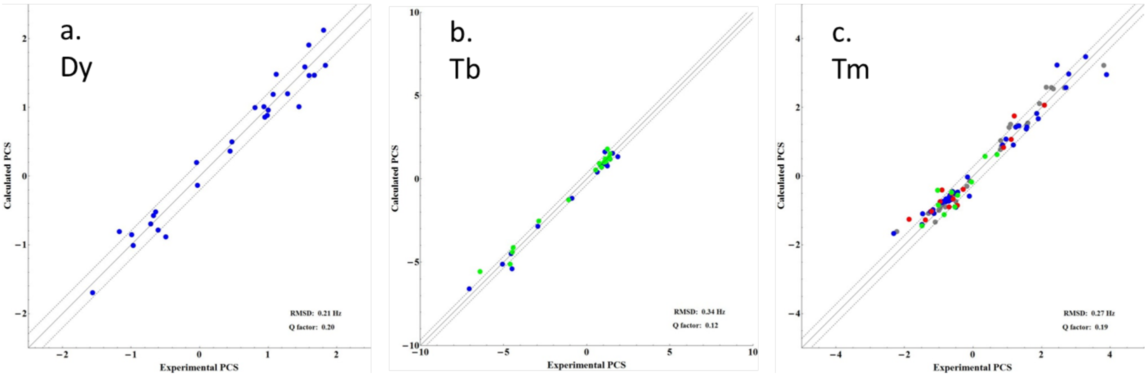

FIGURE 3.6: Experimental PCSs of the N-terminal domain are plotted against back-calculated PCSs in three alignments(Dy/Tb/Tm). RMSD and Q factor are reported in the bottom right corner.

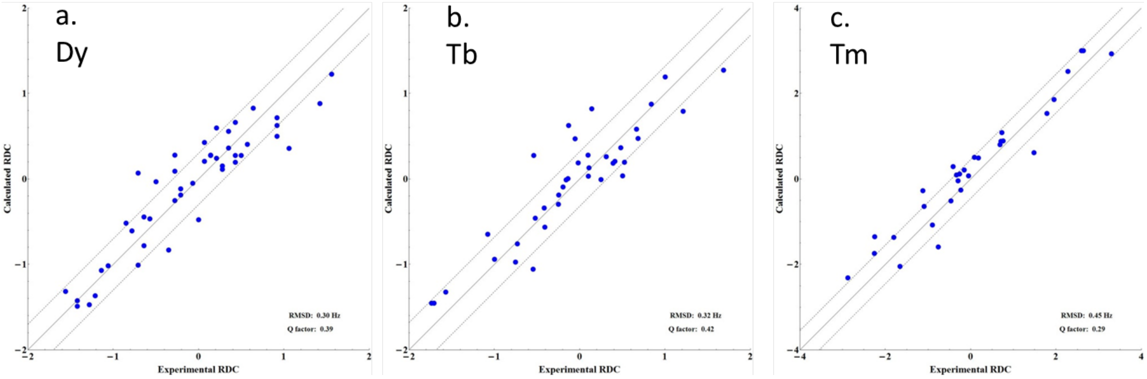

FIGURE 3.7: Experimental RDCs of the C-terminal domain are plotted against back-calculated RDCs in three alignments(Dy/Tb/Tm). RMSD and Q factor are reported in the bottom right corner.

to find the optimal solution for the distribution. The objective function used in the fitting algorithm is Eq. 2.45.

The global minimum has a objective function value of 8.014. After analyzing all solutions with objective function values under 15, another three different solutions were found. They have objective function values of 8.017, 9.894 and 11.56, respectively. The value of the objective function does not mean much, but solutions with objective function values above 15 barely satisfy the experimental constraints. The four solutions are different from each other because the Jensen-Shannon divergences(JSD) between each pair of their corresponding distributions are above 0.2. The divergences between each pair are summarized in Table 3.3.

Table 3.3: JSD between the four solutions of CaM

JSD	Dist 1	Dist2	Dist3	Dist4
Dist1	—	0.55	0.94	0.95
Dist2		—	0.96	0.95
Dist3			—	0.55
Dist4				—

Inter-domain orientations are expressed as quaternions in the distribution. From the first four optimal solutions, I derived the quaternion with highest probability in each solution for distribution. From the four conformations shown in Fig.3.5, I also calculated the quaternion corresponding to each conformation. By comparing the two sets of quaternions, it can be shown easily that the highest probability orientation in solution 1 is nearly identical to conformation 3, the orientation in solution 2 is nearly identical to conformation 1, the orientation in solution 3 is nearly identical to conformation 4 and the orientation in solution 4 is nearly identical to conformation 2. The similarities between the highest probability orientations and conformations are first quantified by the inner product of their corresponding quaternions. The inner product range from 0 to 1, indicating least to highest similarity. The result is summarized in Table 3.4. I also applied another similarity measure called FAA percentile on the inter-domain orientations [32]. The percentile presents the fraction of all orientations that have a larger geometric difference from one orientation in the pair than the difference between the two in the pair. It ranges from 0 to 100, indicating least to highest similarity. In addition, inter-domain conformations corresponding to the quaternion pairs are shown in Fig. 3.8 to provide a more intuitive comparison.

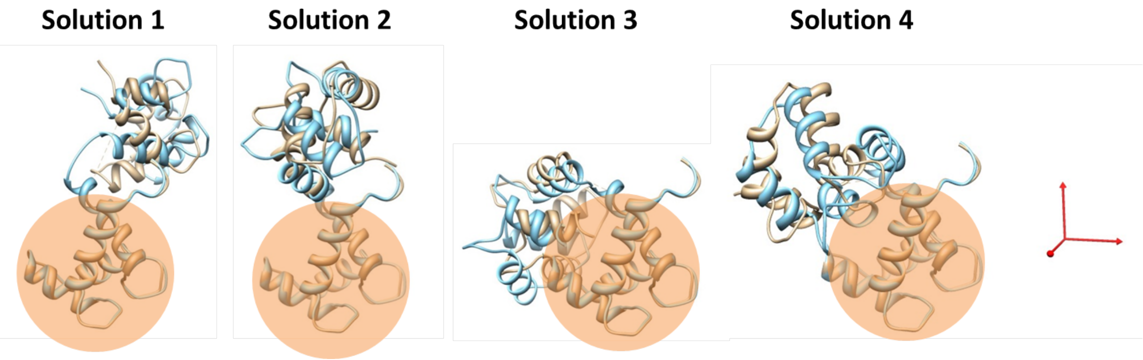

FIGURE 3.8: The N-terminal domain of each panel is shaded in orange. Cyan structures represent the MAP conformations. Brown structures represent the highest probability orientations calculated by the branch-and-bound method.

Table 3.4: Similarity between the quaternions representing the maximum probability orientation of the four solutions of CaM

	orientation1 and conformer3	orientation2 and conformer1	orientation3 and conformer4	orientation4 and conformer2
Inner product	0.94	0.95	0.96	0.93
FAA percentile	98.2	99.1	98.9	98.6

The continuous probabilistic approach and the MAP method agree with each other in general. Four solutions were observed when fitting for distributions and four MAP maxima were observed in the 2007 JACS paper. The existence of multiple solutions or multiple modes could indicate a degeneracy in the RDC and PCS data of calmodulin, which does not have the information content to distinguish the solutions. Alternatively, all the modes could be real motional modes of calmodulin. The fact of finding multiple solutions indicates that the continuous probabilistic approach has the ability to detect multiple modes even though the algorithm uses a unimodal model for fitting.

4

The maximum entropy method

4.1 The reasons to choose the maximum entropy solution

Following the discussion in section 1.2, our problem is an ill-posed inverse problem which may have an infinite number of solutions. Because all the solutions satisfy the experimental observables, we need some other criteria to select one from the others. One good criterion could be the one which Boltzmann used to construct the Boltzmann distribution. When constructing the Boltzmann distribution, Boltzmann faced a very similar problem. Given the total energy of a system E_{total} and the total number of particles N, what is the probability of a state with certain state energy E. The problem is clearly another ill-posed inverse problem and it does not have a unique solution. At the end, Boltzmann chose to find the most probable distribution, which can be realized in the maximum number of ways. In the discrete case, the energy level is partitioned into s small intervals. Suppose the i-th interval is occupied with N_i particles. The number of ways to be realized is calculated by the following equation:

$$W = \frac{N!}{N_1!N_2!\ldots N_s!}. \tag{4.1}$$

When N is large enough, Eq. 4.1 can be calculated using the Stirling approximation, which is:

$$\ln W = -N \sum_{i=1}^{s} \frac{N_i}{N} \ln \frac{N_i}{N} = -N \sum_{i=1}^{s} p_i \ln p_i, \tag{4.2}$$

where $p_i = \frac{N_i}{N}$. Eq. 4.2 indicates that the entropy is a measure of the number of ways to be realized. In order to get the most probable distribution, we need to maximize the entropy of the distribution. For the continuous case, the entropy measure could be differential entropy:

$$S(p(x)) = -\int p(x) \ln p(x) dx. \tag{4.3}$$

Consequently, differential entropy is a reasonable criterion and we could use it as a regularization to solve our inverse problem.

In addition, maximizing the entropy enforces smoothness to the distribution. In other words, we are reconstructing a function with most low frequency components and with little high frequency components. It is reasonable to exclude high frequency oscillations in the reconstructed distribution because of what we observe in the $\mathbf{E}[Q]$ matrix (Eq. 2.18). The $\mathbf{E}[Q]$ matrix carries all the information content we can use for the reconstruction. However, the matrix only contains information about second-order circular moments. High frequency oscillations are invisible to these low-order moments, so including high frequency oscillations introduces additional assumptions and possibly biases into the reconstruction. As a result, it is reasonable to exclude them and thus avoid biases in the result.

At last but not least, the maximum entropy solution is the correct solution from the Bayesian point of view. Bayesian probability does not represent the frequency of an event over an infinite amount of time, instead, it represents our state of knowledge. A probability distribution is inferred from observables. It may change when additional information is introduced. Nonetheless, the probability distribution represents our state of knowledge given the limited information we currently have. Within the Bayesian system, we have the Bayes' law:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4.4}$$

Using Eq. 4.4, we can infer a probability distribution given a prior distribution $P(A)$ and current evidences B. However, we lack the prior distribution to start with in our problem. The problem of obtaining the prior distribution dates back to the time of James Bernoulli, when he proposed the principle of insufficient reasons. The principle of maximum entropy is a modern replacement, first expounded by Edwin T. Jaynes [33, 34]. According to the principle, the distribution with the maximum entropy best represents our current state of knowledge. The maximum entropy solution also agrees with the Bayes law. In our problem, it is physically reasonable to assign equal probability to each state/conformation without any prior information. It can be proved that the maximum entropy solution is the same as the Bayesian inference result obtained by updating a uniform distribution [35].

4.2 The maximum entropy solution takes the exponential form

Before going into the details of the maximum entropy method, we first need to have a good definition of entropy. For discrete problems, we can use the Shannon entropy:

$$S = -\sum_i p_i \ln p_i. \tag{4.5}$$

For continuous problems, we can extend the concept of Shannon entropy into the continuous domain and use the differential entropy in Eq. 4.3. However, there are problems associated with this simple extension. First, the differential entropy is not always non-negative. Second, the differential entropy is not invariant with respect to change of variables. In order to overcome these intrinsic problems associated with the differential entropy, we can use the Kullback-Leibler(KL) divergence between our target distribution $p(x)$ and a reference distribution $q(x)$. The KL divergence is also known as the relative entropy:

$$D_{KL}(p(x)||q(x)) = \int p(x) \ln \frac{p(x)}{q(x)} dx. \tag{4.6}$$

Here, we choose the uniform distribution $u(x)$ as the reference distribution. So the relative entropy becomes:

$$S(p(x)) = \int p(x) \ln \frac{p(x)}{u(x)} dx. \tag{4.7}$$

The relative entropy overcomes the previous problems. It is always non-negative and it is invariant under parameter transformations. Although the differential entropy has certain problems, I still use the differential entropy in the following derivations for simplicity. The conclusion holds for relative entropy as well.

Based on the discussion in section 2.1, the elements in the Q matrix are circular moments of interdomain motions. Consequently, one goal is to find a reasonable orientation distribution satisfying the circular moments constraints. By following the reasons stated in section 4.1, the other goal is to maximize the entropy of the distribution. In summary, we have the following optimization problem:

$$\begin{aligned} \underset{p(x)}{\text{maximize}} \quad & S(p(x)) = -\int p(x) \ln p(x) dx \\ \text{subject to} \quad & \int p(x) dx = 1; \;\; \langle r_i(x) \rangle = \int r_i(x) p(x) dx = m_i, \; i = 1, \ldots, n. \end{aligned} \tag{4.8}$$

where $p(x)$ is the probability density function of the orientation distribution. $S(p(x))$ is the differential entropy of the continuous function. $\langle r_i(x) \rangle$ is a circular moment which should equal to the experimental observable m_i. The distribution is also enforced to be normalized.

For our problem, we could use the method of Lagrange multipliers to find the local maximum of entropy function $S(p(x))$ subject to the equality constraints. For this purpose, a Lagrangian is constructed as:

$$\mathcal{L}(p(x), \lambda_0, \ldots, \lambda_n) = -\textstyle\int p(x) \ln p(x) dx + \lambda_0(\int p(x) dx - 1) + \sum_{i=1}^{n} \lambda_i (\int r_i(x) p(x) dx - m_i). \tag{4.9}$$

Because all the terms in the Lagrangian are integrals over x, the Lagrangian is not a function of x, but a functional. So we vary the probability distribution $p(x)$ instead of x to change the Lagrangian. Nonetheless, the method of Lagrangian multipliers still work. If a maximum entropy solution exists, then the solution corresponds to a stationary point of the Lagrangian $\mathcal{L}(p(x), \lambda_0, \ldots, \lambda_n)$. In the stationary point, all the equality constraints are satisfied:

$$\frac{\partial \mathcal{L}}{\partial \lambda_0} = \int p(x) dx - 1 = 0, \tag{4.10}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = \int r_i(x) p(x) dx - m_i = 0, i = 1, \ldots, n. \tag{4.11}$$

The entropy of the probability distribution also reach one of its extremes. It should be noted that the method of Lagrange multipliers yield a necessary but not a sufficient condition for optimality.

Because the Lagrangian is a functional, the entropy attains an extreme when the functional derivative equals zero:

$$\frac{\partial \mathcal{L}}{\partial p(x)} = -\ln p(x) - 1 + \lambda_0 + \sum_{i=1}^{n} \lambda_i r_i(x) = 0. \tag{4.12}$$

Consequently, we have our maximum entropy solution in the form of:

$$p(x) = \exp(\lambda_0 - 1 + \sum_{i=1}^{n} \lambda_i r_i(x)). \tag{4.13}$$

If we redefine $\lambda_0 \equiv \lambda_0 - 1$, we have a more compact form:

$$p(x) = \exp(\lambda_0 + \sum_{i=1}^{n} \lambda_i r_i(x)). \tag{4.14}$$

Because the probability distribution needs to be normalized, we have the following relationship:

$$\int \exp(\lambda_0 + \sum_{i=1}^{n} \lambda_i r_i(x))dx = 1, \tag{4.15}$$

$$\exp(-\lambda_0) = \int \exp(\sum_{i=1}^{n} \lambda_i r_i(x))dx. \tag{4.16}$$

Conventionally, we name the right-hand side of Eq. 4.16 as the partition function Z. So:

$$Z = \int \exp(\sum_{i=1}^{n} \lambda_i r_i(x))dx, \tag{4.17}$$

$$\lambda_0 = -\ln Z. \tag{4.18}$$

Assume a solution exists and the solution has the Lagrange multipliers as $\mathbf{\Lambda}^* = (\lambda_0^*, \lambda_1^*, \ldots, \lambda_n^*)$. The probability density function of the solution is:

$$p^*(x) = \exp(\lambda_0^* + \sum_{i=1}^{n} \lambda_i^* r_i(x)). \tag{4.19}$$

The solution achieves the maximum entropy among all solutions satisfying the constraints. If we define any distribution satisfying the constraints as $q(x)$, then we always have:

$$S(q(x)) \leqslant S(p^*(x)) \tag{4.20}$$

4.3 The Lagrange multipliers can be solved through a dual problem

In our optimization problem, the goal is to find the set of Lagrange multipliers that maximize the entropy and satisfy the constraints. However, we could also view the

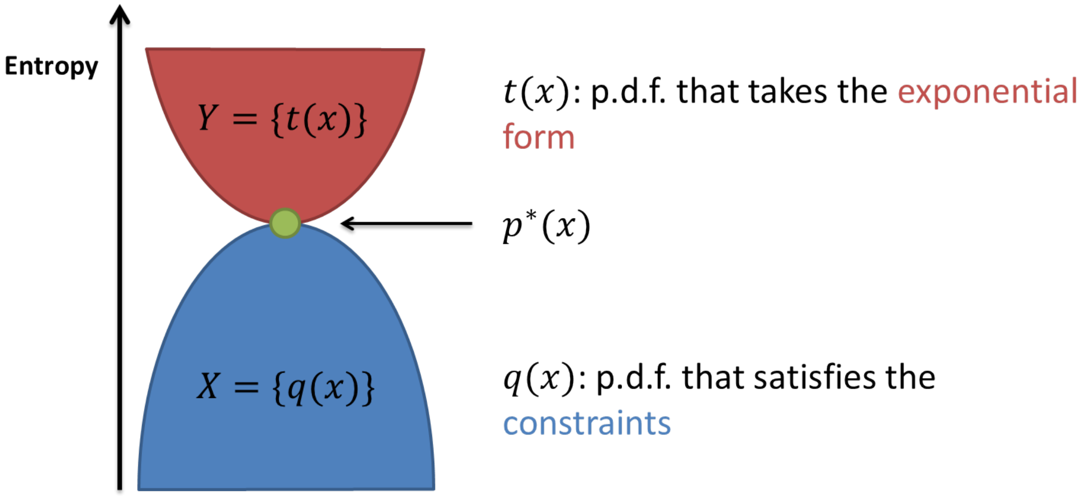
Entropy
$Y = \{t(x)\}$
$X = \{q(x)\}$
$t(x)$: p.d.f. that takes the exponential form
$p^*(x)$
$q(x)$: p.d.f. that satisfies the constraints

FIGURE 4.1: The primal and the dual problem. The primal problem is to find the maximum in the set of X. The dual problem is to find the minimum in the set of Y.

problem from a different point of view. If we have a function which provides an upper bound of the entropy, the other way to solve the problem is to find the minimum of the upper-bound function. The minimum is always larger than or equal to entropy of any distribution $q(x)$. When the minimum equals the maximum entropy $S(p^*(x))$, we can solve the Lagrange multipliers by finding the minimum of the upper-bound function. In this perspective, the original maximum entropy problem is the primal problem, searching for the minimum of the upper-bound function is the dual problem (Fig. 4.1). Next, I will show the construction of the upper-bound function and how to search for its global minimum.

As discussed in section 4.2, KL divergence D_{KL} is a measure of the difference between two probability distributions. For any two probability distributions, the measure D_{KL} is always non-negative. Suppose we have two distributions. The first one is $p^*(x)$ in Eq. 4.19. The second one is any probability distribution that takes the following exponential form but not necessarily satisfies the moment constraints in Eq. 4.8:

$$t(x) = \exp(\lambda_0 + \sum_{i=1}^{n} \lambda_i r_i(x)). \tag{4.21}$$

Because $D_{KL}(p^*(x)||t(x)) \geqslant 0$, we have:

$$\int p^*(x) \ln \frac{p^*(x)}{t(x)} \geqslant 0, \tag{4.22}$$

$$-\int p^*(x) \ln p^*(x) \leqslant -\int p^*(x) \ln t(x). \tag{4.23}$$

From $S(p^*(x)) = -\int p^*(x) \ln p^*(x)$ and Eq. 4.21, we have:

$$S(p^*(x)) \leqslant -\int p^*(x)(\lambda_0 + \sum_{i=1}^{n} \lambda_i r_i(x)) = -\lambda_0 - \sum_{i=1}^{n} \lambda_i m_i, \tag{4.24}$$

and

$$S(q(x)) \leqslant S(p^*(x)) \leqslant -\lambda_0 - \sum_{i=1}^{n} \lambda_i m_i. \tag{4.25}$$

Eq. 4.25 provides the upper bound we need to generate the dual problem. We can also conclude the duality gap is zero because the equality of Eq. 4.24 can be achieved when $(\lambda_0, \lambda_1, \ldots, \lambda_n) = (\lambda_0^*, \lambda_1^*, \ldots, \lambda_n^*)$.

Based on the above discussions, a new objective function can be constructed as:

$$W = -\lambda_0 - \sum_{i=1}^{n} \lambda_i m_i. \tag{4.26}$$

From Eq. 4.18, we also have:

$$W = \ln Z - \sum_{i=1}^{n} \lambda_i m_i. \tag{4.27}$$

Now, to solve Lagrange multipliers of the maximum entropy solution, we only need to find the minimum of Eq. 4.27. We also observe the followings in a stationary point of the objective function 4.27:

$$\frac{\partial W}{\partial \lambda_i} = \langle r_i(x) \rangle - m_i = 0, i = 1, \ldots, n. \tag{4.28}$$

The resulted probability distribution is also normalized by construction, so the distribution indeed satisfies all constraints.

4.4 The dual problem is a convex optimization problem

The purpose of converting the primal problem to its dual problem is to exploit a special property of the new objective function 4.27. As it turns out, the function is strictly convex. The strict convexity gives us two guarantees. First, there is only one unique solution assuming any solution exists. Second, the global minimum can be found provably by applying gradient descent methods. The rest of the section presents a proof of the convexity [36, 6].

Given the objective function 4.27, we can construct its Hessian matrix. Using Eq. 4.27, the first order partial derivative with respect to λ_i is:

$$\frac{\partial W}{\partial \lambda_i} = \frac{1}{Z}\frac{\partial Z}{\partial \lambda_i} - m_i. \tag{4.29}$$

Then the second order partial derivative with respect to λ_i and λ_j is:

$$\frac{\partial W}{\partial \lambda_i \partial \lambda_j} = \frac{1}{Z^2}(\frac{\partial Z}{\partial \lambda_i \partial \lambda_j} Z - \frac{\partial Z}{\partial \lambda_i}\frac{\partial Z}{\partial \lambda_j}), \tag{4.30}$$

$$\frac{\partial W}{\partial \lambda_i \partial \lambda_j} = \frac{1}{Z}\frac{\partial Z}{\partial \lambda_i \partial \lambda_j} - \frac{1}{Z}\frac{\partial Z}{\partial \lambda_i}\frac{1}{Z}\frac{\partial Z}{\partial \lambda_j}. \tag{4.31}$$

As Z is defined in Eq. 4.17, we have:

$$\begin{aligned}\frac{1}{Z}\frac{\partial Z}{\partial\lambda_i\partial\lambda_j} &= \frac{1}{Z}\frac{\partial(\int\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx)}{\partial\lambda_i\partial\lambda_j}\\ &= \frac{1}{Z}\int r_i(x)r_j(x)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx\\ &= \langle r_i(x)r_j(x)\rangle,\end{aligned} \tag{4.32}$$

$$\begin{aligned}\frac{1}{Z}\frac{\partial Z}{\partial\lambda_i} &= \frac{1}{Z}\frac{\partial(\int\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx)}{\partial\lambda_i}\\ &= \frac{1}{Z}\int r_i(x)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx\\ &= \langle r_i(x)\rangle.\end{aligned} \tag{4.33}$$

So

$$\frac{\partial W}{\partial\lambda_i\partial\lambda_j} = \langle r_i(x)r_j(x)\rangle - \langle r_i(x)\rangle\langle r_j(x)\rangle, \tag{4.34}$$

$$\frac{\partial W}{\partial\lambda_i\partial\lambda_j} = \langle r_i(x)r_j(x)\rangle - 2\langle r_i(x)\rangle\langle r_j(x)\rangle + \langle r_i(x)\rangle\langle r_j(x)\rangle, \tag{4.35}$$

$$\begin{aligned}\frac{\partial W}{\partial\lambda_i\partial\lambda_j} = &\frac{1}{Z}\int r_i(x)r_j(x)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx - \frac{1}{Z}\int r_i(x)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx\langle r_j(x)\rangle\\ &- \langle r_i(x)\rangle\frac{1}{Z}\int r_i(x)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx + \langle r_i(x)\rangle\langle r_j(x)\rangle\frac{1}{Z}\int\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx,\end{aligned} \tag{4.36}$$

$$\tfrac{\partial W}{\partial\lambda_i\partial\lambda_j} = \tfrac{1}{Z}\int(r_i(x)r_j(x) - r_i(x)\langle r_j(x)\rangle - \langle r_i(x)\rangle r_j(x) + \langle r_i(x)r_j(x)\rangle)\exp(\textstyle\sum_{i=1}^{n}\lambda_i r_i(x))dx. \tag{4.37}$$

$$\begin{aligned}\frac{\partial W}{\partial\lambda_i\partial\lambda_j} &= \frac{1}{Z}\int(r_i(x) - \langle r_i(x)\rangle)(r_j(x) - \langle r_j(x)\rangle)\exp(\sum_{i=1}^{n}\lambda_i r_i(x))dx\\ &= \langle(r_i(x) - \langle r_i(x)\rangle)(r_j(x) - \langle r_j(x)\rangle)\rangle.\end{aligned} \tag{4.38}$$

Define function $B_i(x)$ as:

$$B_i(x) = r_i(x) - \langle r_i(x) \rangle, \tag{4.39}$$

so we have more compact form of the second order derivative:

$$\frac{\partial W}{\partial \lambda_i \partial \lambda_j} = \langle B_i(x) B_j(x) \rangle. \tag{4.40}$$

The Hessian matrix of the objective function 4.27 takes the following form:

$$H(W) = \begin{bmatrix} \langle B_1(x)B_1(x)\rangle & \langle B_1(x)B_2(x)\rangle & \dots & \langle B_1(x)B_n(x)\rangle \\ \langle B_1(x)B_2(x)\rangle & \langle B_2(x)B_2(x)\rangle & \dots & \langle B_2(x)B_n(x)\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle B_1(x)B_n(x)\rangle & \langle B_2(x)B_n(x)\rangle & \dots & \langle B_n(x)B_n(x)\rangle \end{bmatrix}. \tag{4.41}$$

where $H(W)_{i,j} = \frac{\partial W}{\partial \lambda_i \partial \lambda_j} = \langle B_i(x) B_j(x) \rangle$. To prove the objective function is convex, we need to prove the Hessian matrix $H(W)$ is positive semi-definite. For any vector $\mathbf{v} = (v_1, v_2, \dots, v_n)$, we have:

$$\begin{aligned} \mathbf{v}^T H(W) \mathbf{v} &= \mathbf{v}^T \begin{bmatrix} \langle B_1(x)B_1(x)\rangle & \langle B_1(x)B_2(x)\rangle & \dots & \langle B_1(x)B_n(x)\rangle \\ \langle B_1(x)B_2(x)\rangle & \langle B_2(x)B_2(x)\rangle & \dots & \langle B_2(x)B_n(x)\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle B_1(x)B_n(x)\rangle & \langle B_2(x)B_n(x)\rangle & \dots & \langle B_n(x)B_n(x)\rangle \end{bmatrix} \mathbf{v} \\ &= \sum_{i,j=1}^{n} v_i v_j \langle B_i(x) B_j(x) \rangle \\ &= \sum_{i,j=1}^{n} \langle v_i B_i(x) v_j B_j(x) \rangle \\ &= \langle (\sum_{i=1}^{n} v_i B_i(x))^2 \rangle \\ &\geqslant 0. \end{aligned} \tag{4.42}$$

The equality in Eq. 4.42 obtains when

$$\sum_{i=1}^{n} v_i B_i(x) = 0, \tag{4.43}$$

which means functions in $\sum_{i=1}^{n} v_i B_i(x)$ are linear dependent. The functions are the moment functions $r_i(x)$ and a constant function $c_0 = \sum_{i=1}^{n} \langle r_i(x) \rangle$. The next section will show the functions in our problem are orthogonal to each other, so the objective function 4.27 is strictly convex.

4.5 Linear combinations of the functions in the $\tilde{Q}$ matrix are orthonormal functions

We have seen the Q matrix in Eq. 2.18. However, the circular moment functions are not orthogonal to each other. One of the problem is that the vectorized Saupe tensors $s = [S_{xx}, S_{yy}, S_{xy}, S_{xz}, S_{yz}]^T$ do not have coordinates based on orthogonal unit vectors. Because of the constraint $S_{xx} + S_{yy} + S_{zz} = 0$, a Saupe tensor has six different elements but only five degrees of freedom (DOFs). Consequently, variables S_{xx}, S_{yy} and S_{zz} live in a plane of the three dimensional space. Suppose a vector $\mathbf{v}$ on the plane is:

$$\mathbf{v} = S_{xx}\mathbf{i} + S_{yy}\mathbf{j} + S_{zz}\mathbf{k}, \tag{4.44}$$

where $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are three orthogonal base vectors. We can construct a new pair of base vectors:

$$\mathbf{i}' = \frac{\sqrt{2}}{2}(\mathbf{i} - \mathbf{j}), \tag{4.45}$$

$$\mathbf{j}' = \frac{\sqrt{6}}{6}(-\mathbf{i} - \mathbf{j} + 2\mathbf{k}). \tag{4.46}$$

The new base vectors are orthonormal and they are on the plane $S_{xx} + S_{yy} + S_{zz} = 0$. If we write vector $\mathbf{v}$ regarding to the new bases, we have:

$$\begin{aligned} \mathbf{v} &= (\frac{\sqrt{2}}{2}S_{xx} - \frac{\sqrt{2}}{2}S_{yy})\frac{\sqrt{2}}{2}(\mathbf{i} - \mathbf{j}) + \frac{\sqrt{6}}{2}S_{zz}\frac{\sqrt{6}}{6}(-\mathbf{i} - \mathbf{j} + 2\mathbf{k}) \\ &= (\frac{\sqrt{2}}{2}S_{xx} - \frac{\sqrt{2}}{2}S_{yy})\mathbf{i}' + \frac{\sqrt{6}}{2}S_{zz}\mathbf{j}'. \end{aligned} \tag{4.47}$$

As a result, we could format the vectorized Saupe tensor in a different way:

$$\tilde{\mathbf{s}} = [\frac{\sqrt{6}}{2}S_{zz}, \frac{\sqrt{2}}{2}S_{xx} - \frac{\sqrt{2}}{2}S_{yy}, \sqrt{2}S_{xy}, \sqrt{2}S_{xz}, \sqrt{2}S_{yz}]^T, \tag{4.48}$$

where the coordinates are based on orthogonal unit vectors. In order to use the format, Q matrix has to be reformatted as well. The new $\tilde{Q}$ matrix takes the following form:

$$\tilde{Q} =$$

$$\frac{\sqrt{5}}{4\sqrt[4]{2}\pi}\begin{bmatrix} \frac{1}{2}(2z_3^2 - z_1^2 - z_2^2) & \frac{\sqrt{3}}{2}(z_1^2 - z_2^2) & \sqrt{3}z_1z_2 & \sqrt{3}z_1z_3 & \sqrt{3}z_2z_3 \\ \frac{\sqrt{3}}{6}(2x_3^2 - x_1^2 - x_2^2 - 2y_3^2 + y_1^2 + y_2^2) & \frac{1}{2}(x_1^2 - x_2^2 - y_1^2 + y_2^2) & x_1x_2 - y_1y_2 & x_1x_3 - y_1y_3 & x_2x_3 - y_2y_3 \\ \frac{\sqrt{3}}{3}(2x_3y_3 - x_1y_1 - x_2y_2) & x_1y_1 - x_2y_2 & x_1y_2 + x_2y_1 & x_1y_3 + x_3y_1 & x_2y_3 + x_3y_2 \\ \frac{\sqrt{3}}{3}(2x_3z_3 - x_1z_1 - x_2z_2) & x_1z_1 - x_2z_2 & x_1z_2 + x_2z_1 & x_1z_3 + x_3z_1 & x_2z_3 + x_3z_2 \\ \frac{\sqrt{3}}{3}(2y_3z_3 - y_1z_1 - y_2z_2) & y_1z_1 - y_2z_2 & y_1z_2 + y_2z_1 & y_1z_3 + y_3z_1 & y_2z_3 + y_3z_2 \end{bmatrix} \tag{4.49}$$

The elements in the $\tilde{Q}$ matrix are functions. It can be proved that the functions in Eq. 4.49 are orthonormal functions and that they are orthogonal to a constant function (Appendix B). Based on the discussion in the previous section, given the expectation of the 25 functions, we can construct a strictly convex objective function and we can guarantee to find the global optimum if one exists. However, getting all 25 expectations can be hard through experiments. Most of the time, we only have a couple linear combinations of the moments. Previous studies have derived methods to orthogonalize the linear combination coefficients [20]. Basically, the method uses a SVD algorithm to find orthogonal principle components out of the experimentally determined Saupe tensors. The following is going to prove that the conclusion holds if the linear combination coefficient vectors are orthonormal vectors.

Using the $\tilde{Q}$ matrix, we can establish the following relationship between $\tilde{\mathbf{s}}_{\text{I}}$ and $\tilde{\mathbf{s}}_{\text{II}}$:

$$\tilde{\mathbf{s}}_{\text{II}} = \frac{4\sqrt[4]{2}\pi}{\sqrt{5}}\mathbf{E}[\tilde{Q}] \cdot \tilde{\mathbf{s}}_{\text{I}}, \tag{4.50}$$

where $\mathbf{E}[\tilde{Q}]$ is an element-wise expectation of the $\tilde{Q}$ matrix in the following form

$$\mathbf{E}[\tilde{Q}] =$$

$$\frac{\sqrt{5}}{4\sqrt[4]{2}\pi}\left[\begin{array}{ccccc} \frac{1}{2}\langle 2z_3^2 - z_1^2 - z_2^2\rangle & \frac{\sqrt{3}}{2}\langle z_1^2 - z_2^2\rangle & \sqrt{3}\langle z_1 z_2\rangle & \sqrt{3}\langle z_1 z_3\rangle & \sqrt{3}\langle z_2 z_3\rangle \\ \frac{\sqrt{3}}{6}\langle 2x_3^2 - x_1^2 - x_2^2 - 2y_3^2 + y_1^2 + y_2^2\rangle & \frac{1}{2}\langle x_1^2 - x_2^2 - y_1^2 + y_2^2\rangle & \langle x_1 x_2 - y_1 y_2\rangle & \langle x_1 x_3 - y_1 y_3\rangle & \langle x_2 x_3 - y_2 y_3\rangle \\ \frac{\sqrt{3}}{3}\langle 2x_3 y_3 - x_1 y_1 - x_2 y_2\rangle & \langle x_1 y_1 - x_2 y_2\rangle & \langle x_1 y_2 + x_2 y_1\rangle & \langle x_1 y_3 + x_3 y_1\rangle & \langle x_2 y_3 + x_3 y_2\rangle \\ \frac{\sqrt{3}}{3}\langle 2x_3 z_3 - x_1 z_1 - x_2 z_2\rangle & \langle x_1 z_1 - x_2 z_2\rangle & \langle x_1 z_2 + x_2 z_1\rangle & \langle x_1 z_3 + x_3 z_1\rangle & \langle x_2 z_3 + x_3 z_2\rangle \\ \frac{\sqrt{3}}{3}\langle 2y_3 z_3 - y_1 z_1 - y_2 z_2\rangle & \langle y_1 z_1 - y_2 z_2\rangle & \langle y_1 z_2 + y_2 z_1\rangle & \langle y_1 z_3 + y_3 z_1\rangle & \langle y_2 z_3 + y_3 z_2\rangle \end{array}\right]. \tag{4.51}$$

If we define each element in the $\tilde{Q}$ matrix as $\widetilde{Q_{i,j}}(R)$, the expectation matrix becomes:

$$\mathbf{E}[\tilde{Q}] = \left[\begin{array}{ccccc} \langle\tilde{Q}_{1,1}\rangle & \langle\tilde{Q}_{1,2}\rangle & \langle\tilde{Q}_{1,3}\rangle & \langle\tilde{Q}_{1,4}\rangle & \langle\tilde{Q}_{1,5}\rangle \\ \langle\tilde{Q}_{2,1}\rangle & \langle\tilde{Q}_{2,2}\rangle & \langle\tilde{Q}_{2,3}\rangle & \langle\tilde{Q}_{2,4}\rangle & \langle\tilde{Q}_{2,5}\rangle \\ \langle\tilde{Q}_{3,1}\rangle & \langle\tilde{Q}_{3,2}\rangle & \langle\tilde{Q}_{3,3}\rangle & \langle\tilde{Q}_{3,4}\rangle & \langle\tilde{Q}_{3,5}\rangle \\ \langle\tilde{Q}_{4,1}\rangle & \langle\tilde{Q}_{4,2}\rangle & \langle\tilde{Q}_{4,3}\rangle & \langle\tilde{Q}_{4,4}\rangle & \langle\tilde{Q}_{4,5}\rangle \\ \langle\tilde{Q}_{5,1}\rangle & \langle\tilde{Q}_{5,2}\rangle & \langle\tilde{Q}_{5,3}\rangle & \langle\tilde{Q}_{5,4}\rangle & \langle\tilde{Q}_{5,5}\rangle \end{array}\right], \tag{4.52}$$

where R is the rotation matrix in Eq. 2.6. Similar to Eq. 2.19, we also have:

$$\tilde{G}_{ij} = \int_{SO(3)} \tilde{Q}_{ij} P(R) dR. \tag{4.53}$$

Suppose we have one pair of normalized Saupe tensors $\tilde{\mathbf{s}}_{\text{I}} = [\tilde{s}_{\text{I},1}, \tilde{s}_{\text{I},2}, \tilde{s}_{\text{I},3}, \tilde{s}_{\text{I},4}, \tilde{s}_{\text{I},5}]$ and $\tilde{\mathbf{s}}_{\text{II}} = [\tilde{s}_{\text{II},1}, \tilde{s}_{\text{II},2}, \tilde{s}_{\text{II},3}, \tilde{s}_{\text{II},4}, \tilde{s}_{\text{II},5}]$. We have five linear relationships:

$$\tilde{s}_{\text{II},i} = \tilde{s}_{\text{I},1}\langle\tilde{Q}_{i,1}\rangle + \tilde{s}_{\text{I},2}\langle\tilde{Q}_{i,2}\rangle + \tilde{s}_{\text{I},3}\langle\tilde{Q}_{i,3}\rangle + \tilde{s}_{\text{I},4}\langle\tilde{Q}_{i,4}\rangle + \tilde{s}_{\text{I},5}\langle\tilde{Q}_{i,5}\rangle;\ i = 1, \ldots, 5. \tag{4.54}$$

Define new functions $O_i (i = 1, \ldots, 5)$ as:

$$O_i = \tilde{s}_{\text{I},1}\tilde{Q}_{i,1} + \tilde{s}_{\text{I},2}\tilde{Q}_{i,2} + \tilde{s}_{\text{I},3}\tilde{Q}_{i,3} + \tilde{s}_{\text{I},4}\tilde{Q}_{i,4} + \tilde{s}_{\text{I},5}\tilde{Q}_{i,5};\ i = 1, \ldots, 5. \tag{4.55}$$

We have:

$$\tilde{s}_{\text{II},i} = \langle O_i\rangle;\ i = 1, \ldots, 5. \tag{4.56}$$

Here, O_i is a linear combination of the 25 functions. It can be proved that O_i is normalized.

$$\begin{aligned}\int O_i^2 dR &= \int (\sum_{j=1}^{5} \tilde{s}_{\mathrm{I},j} \tilde{Q}_{i,j})^2 dR \\ &= \sum_{j=1}^{5} \tilde{s}_{\mathrm{I},j}^2 \int \tilde{Q}_{i,j}^2 dR + \sum_{j \neq k} \tilde{s}_{\mathrm{I},j} \tilde{s}_{\mathrm{I},k} \int \tilde{Q}_{i,j} \tilde{Q}_{i,k} dR\end{aligned} \tag{4.57}$$

Because $\tilde{Q}_{i,j}$ are orthonormal functions, $\int \tilde{Q}_{i,j}^2 dR = 1$ and $\int \tilde{Q}_{i,j} \tilde{Q}_{i,k} dR = 0$. We have:

$$\int O_i^2 dR = \sum_{j=1}^{5} \tilde{s}_{\mathrm{I},j}^2 = 1. \tag{4.58}$$

It can also be proved that $O_i (i = 1, \ldots, 5)$ are orthogonal to each other.

$$\begin{aligned}\int O_i O_j dR &= \int (\sum_{k=1}^{5} \tilde{s}_{\mathrm{I},k} \tilde{Q}_{i,k})(\sum_{k=1}^{5} \tilde{s}_{\mathrm{I},k} \tilde{Q}_{j,k}) dR \\ &= \sum_{k,l=1}^{5} \tilde{s}_{\mathrm{I},k} \tilde{s}_{\mathrm{I},l} \int \tilde{Q}_{i,k} \tilde{Q}_{j,l} dR\end{aligned} \tag{4.59}$$

Because $\int \tilde{Q}_{i,k} \tilde{Q}_{j,l} dR = 0$, we have:

$$\int O_i O_j dR = 0 \tag{4.60}$$

Now, the linear combinations $O_i (i = 1, \ldots, 5)$ weighted by one Saupe tensor are orthonormal functions. Suppose we have more than one pair of normalized Saupe tensors. We name the i-th pair as $\tilde{\mathbf{s}}_{\mathrm{I}}^{\mathbf{i}}$ and $\tilde{\mathbf{s}}_{\mathrm{II}}^{\mathbf{i}}$. Define new functions $O_i^j (i = 1, \ldots, 5;)$ as:

$$O_i^j = \tilde{s}_{\mathrm{I},1}^j \tilde{Q}_{i,1} + \tilde{s}_{\mathrm{I},2}^j \tilde{Q}_{i,2} + \tilde{s}_{\mathrm{I},3}^j \tilde{Q}_{i,3} + \tilde{s}_{\mathrm{I},4}^j \tilde{Q}_{i,4} + \tilde{s}_{\mathrm{I},5}^j \tilde{Q}_{i,5};\ i = 1, \ldots, 5;\ j = 1, \ldots, 5. \tag{4.61}$$

Now we want to prove O_i^k and O_j^l $(k \neq l)$ are orthogonal to each other. If $i \neq j$, the proof is similar to Eq. 4.60. When $i = j$, we have:

$$\begin{aligned}\int O_i^k O_i^l dR &= \int (\sum_{j=1}^{5} \tilde{s}_{_{\mathrm{I},j}}^k \tilde{Q}_{i,j})(\sum_{j=1}^{5} \tilde{s}_{_{\mathrm{I},j}}^l \tilde{Q}_{i,j}) dR \\ &= \sum_{j=1}^{5} \tilde{s}_{_{\mathrm{I},j}}^k \tilde{s}_{_{\mathrm{I},j}}^l \int \tilde{Q}_{i,j}^2 dR + \sum_{j1 \neq j2} \tilde{s}_{_{\mathrm{I},j1}}^k \tilde{s}_{_{\mathrm{I},j2}}^l \int \tilde{Q}_{i,j1} \tilde{Q}_{i,j2} dR \qquad (4.62) \\ &= \sum_{j=1}^{5} \tilde{s}_{_{\mathrm{I},j}}^k \tilde{s}_{_{\mathrm{I},j}}^l\end{aligned}$$

Because $\mathbf{\tilde{s}}_{_\mathrm{I}}^{\mathbf{k}}$ and $\mathbf{\tilde{s}}_{_\mathrm{I}}^{\mathbf{l}}$ are orthogonal vectors, we have:

$$\int O_i^k O_i^l dR = \sum_{j=1}^{5} \tilde{s}_{_{\mathrm{I},j}}^k \tilde{s}_{_{\mathrm{I},j}}^l = 0 \qquad (4.63)$$

In summary, all the linear combinations $O_i^j (i = 1, \ldots, 5;\ j = 1, \ldots, 5)$ are orthonormal functions. We can build the strictly convex objective function in Eq. 4.27 given any number of Saupe tensors. The method can be applied to datasets with one to five orthogonal alignments.

5

Conclusions

In this work, I have presented two provable algorithms to determine the interdomain motions from RDCs. We have observed good performance of the branch-and-bound algorithm both on simulated data and experimental data. The branch-and-bound method calculates close-to-ground-truth solutions when the noise level is around $Q = 0.2$ or under. The method also generates a result consistent with the MAP method for the calmodulin data. Although the branch-and-bound method and the MAP method differs from each other, the consistency suggests both methods capture certain key features in the RDC data. In addition, the branch-and-bound method offers a probabilistic interpretation of the result, which MAP fails to deliver. It should be noted that the Bingham model is a unimodal distribution. The uni-modality is the most biased assumption in the model. When the ground truth distribution is a bimodal distribution, the method may fail to find the second mode. Interesting, the method found two solutions from the SpA-N data and four solutions from the calmodulin data. Although the multiple solutions could be due to the degeneracy in the dataset and hence they are the ghost solutions, the solutions could also be modes in a multi-modal distribution. If it is true, the branch-and-bound method has the ability to detect multiple modes. The branch-and-bound algorithm is provable in the sense that it guarantees to find the global minimum of the objective function in Eq. 2.45. However, we don't have bound on the run time of algorithm. In the worst case, the algorithm takes exponential time. Nonetheless, the run time of algorithm depends on the pruning criteria in practice. If the pruning is efficient, the algorithm usually takes much less time than the exponential time. In practice, the branch-and-bound method runs very efficiently. Some runs take around 8 hours, and most runs take less than 24 hours. Despite the good design of the branch-and-bound algorithm, there is another factor significantly contributing to the run time. The algorithm takes a huge amount of operations on evaluating the quartic moments of a Bingham distribution. Instead of systematically sampling the $SO(3)$ space and summing up the samples, I converted the numerical integration problem into evaluating a hypergeometric function and its second derivatives (Appendix A). The evaluations are approximated by an established method. The efficiency of the method contributes significantly to the run time of the algorithm. One limitation of the branch-and-bound method is that it only works with motionally decoupled data. It certainly raises the bar for experimental design. However, in motionally coupled data, the global tumbling information and the interdomain motion information are convolved with each other. Without making assumptions about the coupling, it is impossible to separate them. Unless we have a solid method to predict the coupling mechanism, motionally decoupled data is required to make correct interpretations. On the other hand, I have extended the branch-and-bound method to incorporate data with different reference domains. Consequently, the method can work with any kind of motionally decoupled data.

The second algorithm is a maximum entropy algorithm. There are extensive studies on the maximum entropy problem [33, 34, 36, 6]. Our problem fits perfectly to the existing theoretical framework. There are two reasons for developing the maximum entropy method. First, as we have discussed, the branch-and-bound method uses the Bingham model. Although the model is reasonable, it introduces assumptions and biases into the result. Consequently, it is reasonable to avoid specific models and use a weaker assumption, maximum entropy. Maximum entropy solution is the least biased solution representing our state of knowledge. The maximum entropy method also has the ability to present multiple modes in the solution if the data suggests. Second, the current maximum entropy method can be extended to solve bigger problems. The immediate next step is to determine the joint distribution of both interdomain distance and orientation from RDCs and PCSs. We have some preliminary work on PCSs and we know they are moments of the joint distance and orientation distribution. PCSs certainly contain the information to derive the joint distribution and we need to a method to deal with PCSs. Although the maximum entropy method seems a very reasonable choice on the problem, there are still two questions to be answered before we have a solid method. First, we don't have a succinct formalism like the Q matrix to summarize the information content in PCSs. So the moment functions of PCSs are clearly linearly dependent. Using the same theoretical framework in Chapter 4, we can conclude that the objective function for the PCSs problem is still convex. However, the linearly dependent functions generate redundant Lagrange multipliers, increasing the dimensionality of the search problem significantly. The gradient descent search in this high dimensional space may still be efficient, but we need to test it before giving any conclusion. Second, without the Q matrix, or even the Saupe tensor to summarize the information content in PCSs, the input will contain more noise. The Saupe tensor is usually determined from dozens of RDCs. The five elements in the Saupe tensors are calculated from a SVD. The majority of noise is eliminated in the SVD process. However, without the SVD process, raw PCSs may contain a large amount of noise. The presence of significant noise may threaten the existence of a solution. In summary, the maximum entropy method delivers the ideal answers for problems associated with RDCs and PCSs. Future work needs to be done in order to incorporate PCSs into the current framework.

Appendix A

Numerical integration of the Bingham probability density function

The appendix provides an efficient way to evaluate the quartic moments $\langle q_i^4 \rangle$ and $\langle q_i^2 q_j^2 \rangle$ over S^3 given a Bingham distribution. We first observe that the normalization constant $c(\Lambda)$ in Eq. 2.34 can be expressed as the confluent hypergeometric function of matrix argument:

$$
{}_1F_1(\frac{1}{2}; 2; \Lambda) = \int \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}}. \tag{A.1}
$$

As a result, we can rewrite Eq. 2.34 in the following form:

$$
\begin{aligned}
P(\tilde{\mathbf{q}}\,;\,\Lambda) &= c^{-1}(\Lambda) \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) \\
&= \frac{\exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}})}{\int \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}}} \qquad \text{(A.2)} \\
&= \frac{\exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}})}{{}_1F_1(\frac{1}{2}; 2; \Lambda)}.
\end{aligned}
$$

Define $\Lambda \equiv \mathrm{Diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ and $\tilde{\mathbf{q}} \equiv [q_1, q_2, q_3, q_4]^T$. We also observe that for the numerator on the right hand side of Eq. A.2, its derivative regarding to λ_i is:

$$\frac{\partial \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}})}{\partial \lambda_i} = q_i^2 \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}). \tag{A.3}$$

From the two observations, the first kind of quartic moments $\langle q_i^4 \rangle$ can be written as:

$$\begin{aligned} \langle q_i^4 \rangle &= c^{-1}(\Lambda) \int q_i^4 \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}} \\ &= \frac{\int q_i^4 \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}}}{{}_1F_1(\frac{1}{2}; 2; \Lambda)} \\ &= \frac{1}{{}_1F_1(\frac{1}{2}; 2; \Lambda)} \frac{\partial^2 {}_1F_1(\frac{1}{2}; 2; \Lambda)}{\partial \lambda_i^2} \end{aligned} \tag{A.4}$$

The other kind of quartic moments $\langle q_i^2 q_j^2 \rangle$ can be written as:

$$\begin{aligned} \langle q_i^2 q_j^2 \rangle &= c^{-1}(\Lambda) \int q_i^2 q_j^2 \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}} \\ &= \frac{\int q_i^2 q_j^2 \exp(\tilde{\mathbf{q}}^T \Lambda \tilde{\mathbf{q}}) d\tilde{\mathbf{q}}}{{}_1F_1(\frac{1}{2}; 2; \Lambda)} \\ &= \frac{1}{{}_1F_1(\frac{1}{2}; 2; \Lambda)} \frac{\partial^2 {}_1F_1(\frac{1}{2}; 2; \Lambda)}{\partial \lambda_i \partial \lambda_j} \end{aligned} \tag{A.5}$$

Now we only need to evaluate the hypergeometric function ${}_1F_1(\frac{1}{2}; 2; \Lambda)$ and its second derivatives to calculate the quartic moments. Fortunately, the hypergeometric function ${}_1F_1(\frac{1}{2}; 2; \Lambda)$ can be evaluated very efficiently by an established method [18]. Its derivatives are calculated by numerical differentiation.

Appendix B

The functions in the $\tilde{Q}$ matrix are orthonormal

In this appendix, I give a proof that the functions in the $\tilde{Q}$ matrix are orthonormal. It should be noted that the elements of the $\tilde{Q}$ matrix are functions of a rotation matrix. We could parameterize the rotation matrix with Euler angles α, β and γ.

$$\begin{aligned} R &= \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\sin(\gamma) & -\cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta)\cos(\gamma) + \cos(\alpha)\sin(\gamma) & -\sin(\alpha)\cos(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta) \\ -\sin(\beta)\cos(\gamma) & \sin(\beta)\sin(\gamma) & \cos(\beta) \end{bmatrix}. \end{aligned} \tag{B.1}$$

Although Euler angles are generally avoided because of the singularity and ambiguity, they are used here for the convenience of integration calculation. Because Euler angles are used, the volume element of the transformation dV needs to be calculated. First, we calculate the Jacobian as:

$$J = [\frac{\partial R}{\partial \alpha}, \frac{\partial R}{\partial \beta}, \frac{\partial R}{\partial \gamma}]. \tag{B.2}$$

Then,

$$dV = \sqrt{\det(J^T J)} d\alpha d\beta d\gamma \tag{B.3}$$

I first evaluated the orthogonality between the 25 functions $\tilde{Q}_{i,j}$ and a constant function c:

$$\int c\tilde{Q}_{i,j} dV \tag{B.4}$$

The integrations were evaluated by Mathematica. All 25 integrations equal zero, indicating the 25 functions are orthogonal to a constant function. Then I evaluated the orthogonality between the 25 functions $\tilde{Q}_{i,j}$:

$$\int \tilde{Q}_{i,j}\tilde{Q}_{k,l} dV \tag{B.5}$$

Again, the integrations were evaluated by Mathematica. Out of the 325 integrations, 300 integrations equal zero. They are the integrations involving two different functions. So the functions are orthogonal to each other. In addition, the other 25 integrations result in the same constant, indicating they are all normalized.

Bibliography

[1] Ivano Bertini, Yogesh K Gupta, Claudio Luchinat, Giacomo Parigi, Massimiliano Peana, Luca Sgheri, and Jing Yuan. Paramagnetism-based nmr restraints provide maximum allowed probabilities for the different conformations of partially independent protein domains. *Journal of the American Chemical Society*, 129(42):12786–12794, 2007.

[2] Qi Zhang, Andrew C Stelzer, Charles K Fisher, and Hashim M Al-Hashimi. Visualizing spatially correlated dynamics that directs rna conformational transitions. *Nature*, 450(7173):1263–1267, 2007.

[3] Konstantin Berlin, Carlos A Castaneda, Dina Schneidman-Duhovny, Andrej Sali, Alfredo Nava-Tudela, and David Fushman. Recovering a representative conformational ensemble from underdetermined macromolecular structural data. *Journal of the American Chemical Society*, 135(44):16595–16609, 2013.

[4] D. Zheng, J. M. Aramini, and G. T. Montelione. Validation of helical tilt angles in the solution nmr structure of the z domain of staphylococcal protein a by combined analysis of residual dipolar coupling and noe data. *Protein Sci*, 13(2):549–54, 2004.

[5] Joel R Tolman. A novel approach to the retrieval of structural and dynamic information from residual dipolar couplings using several oriented media in biomolecular nmr spectroscopy. *Journal of the American Chemical Society*, 124(40):12020–12030, 2002.

[6] Lawrence R Mead and Nikos Papanicolaou. Maximum entropy in the problem of moments. *Journal of Mathematical Physics*, 25(8):2404–2417, 1984.

[7] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52):28, 1902.

[8] Andreǐ Nikolaevich Tikhonov and Vasiliǐ IAkovlevich Arsenin. *Solutions of ill-posed problems*. New York: Winston, 1977.

[9] Yiwen Chen, Sharon L Campbell, and Nikolay V Dokholyan. Deciphering protein dynamics from nmr data using explicit structure sampling and selection. *Biophysical journal*, 93(7):2300–2306, 2007.

[10] João Henriques, Carolina Cragnell, and Marie Skepö. Molecular dynamics simulations of intrinsically disordered proteins: force field evaluation and comparison with experiment. *Journal of Chemical Theory and Computation*, 11(7):3420–3431, 2015.

[11] Bartosz Rozycki, Young C Kim, and Gerhard Hummer. Saxs ensemble refinement of escrt-iii chmp3 conformational transitions. *Structure*, 19(1):109–116, 2011.

[12] Andrea Cavalli, Carlo Camilloni, and Michele Vendruscolo. Molecular dynamics simulations with replica-averaged structural restraints generate structural ensembles according to the maximum entropy principle. *The Journal of chemical physics*, 138(9):094112, 2013.

[13] Bruce R Donald. *Algorithms in structural molecular biology.* MIT Press Cambridge, MA:, 2011.

[14] Judit A Losonczi, Michael Andrec, Mark WF Fischer, and James H Prestegard. Order matrix analysis of residual dipolar couplings using singular value decomposition. *Journal of Magnetic Resonance*, 138(2):334–342, 1999.

[15] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, 2(6):1201–1225, 1974.

[16] Karsten Kunze and Helmut Schaeben. The bingham distribution of quaternions and its spherical radon transform in texture analysis. *Mathematical Geology*, 36(8):917–943, 2004.

[17] MJ Prentice. Orientation statistics without parametric assumptions. *Journal of the Royal Statistical Society. Series B. Methodological*, 48(2):214–222, 1986.

[18] Plamen Koev and Alan Edelman. The efficient evaluation of the hypergeometric function of a matrix argument. *Mathematics of Computation*, 75(254):833–846, 2006.

[19] Johan Ernest Mebius. A matrix-based proof of the quaternion representation theorem for four-dimensional rotations. *arXiv preprint math/0501249*, 2005.

[20] Ke Ruan and Joel R Tolman. Composite alignment media for the measurement of independent sets of nmr residual dipolar couplings. *Journal of the American Chemical Society*, 127(43):15032–15033, 2005.

[21] Shan Yang, Loïc Salmon, and Hashim M Al-Hashimi. Measuring similarity between dynamic ensembles of biomolecules. *Nature methods*, 11(5):552–554, 2014.

[22] H. K. Kim, A. G. Cheng, H. Y. Kim, D. M. Missiakas, and O. Schneewind. Nontoxigenic protein a vaccine for methicillin-resistant staphylococcus aureus infections in mice. *J Exp Med*, 207(9):1863–70, 2010.

[23] Johann Deisenhofer. Crystallographic refinement and atomic models of a human fc fragment and its complex with fragment b of protein a from staphylococcus aureus at 2.9 and 2.8 ang. resolution. *Biochemistry*, 20(9):2361–2370, 1981.

[24] Marisa I. Gómez, Maghnus O'Seaghdha, Mariah Magargee, Timothy J. Foster, and Alice S. Prince. Staphylococcus aureus protein a activates tnfr1 signaling through conserved igg binding domains. *Journal of Biological Chemistry*, 281(29):20190–20196, 2006.

[25] Marisa I Gómez, Maghnus O Seaghdha, and Alice S Prince. Staphylococcus aureus protein a activates tace through egfrdependent signaling. *The EMBO journal*, 26(3):701–709, 2007.

[26] Lindsay N Deis, Charles W Pemble, Yang Qi, Andrew Hagarman, David C Richardson, Jane S Richardson, and Terrence G Oas. Multiscale conformational heterogeneity in staphylococcal protein a: Possible determinant of functional plasticity. *Structure*, 22(10):1467–1477, 2014.

[27] B. Jansson, M. Uhlen, and P. A. Nygren. All individual domains of staphylococcal protein a show fab binding. *FEMS Immunol Med Microbiol*, 20(1):69–78, 1998. Jansson, B Uhlen, M Nygren, P A Research Support, Non-U.S. Gov't Netherlands FEMS immunology and medical microbiology FEMS Immunol Med Microbiol. 1998 Jan;20(1):69-78.

[28] Lindsay N Deis, Qinglin Wu, You Wang, Yang Qi, Kyle G Daniels, Pei Zhou, and Terrence G Oas. Suppression of conformational heterogeneity at a proteinprotein interface. *Proceedings of the National Academy of Sciences*, 112(29):9028–9033, 2015.

[29] Jo A Capp, Andrew Hagarman, David C Richardson, and Terrence G Oas. The statistical conformation of a highly flexible protein: Small-angle x-ray scattering of s. aureus protein a. *Structure*, 22(8):1184–1195, 2014.

[30] Pau Bernadó, Efstratios Mylonas, Maxim V Petoukhov, Martin Blackledge, and Dmitri I Svergun. Structural characterization of flexible proteins using small-angle x-ray scattering. *Journal of the American Chemical Society*, 129(17):5656–5664, 2007.

[31] Ivano Bertini, Cristina Del Bianco, Ioannis Gelis, Nikolaus Katsaros, Claudio Luchinat, Giacomo Parigi, Massimiliano Peana, Alessandro Provenzani, and Maria Antonietta Zoroddu. Experimentally exploring the conformational space

sampled by domain reorientation in calmodulin. *Proceedings of the National Academy of Sciences of the United States of America*, 101(18):6841–6846, 2004.

[32] Anthony K Yan, Christopher J Langmead, and Bruce Randall Donald. A probability-based similarity measure for saupe alignment tensors with applications to residual dipolar couplings in nmr structural biology. *The International Journal of Robotics Research*, 24(2-3):165–182, 2005.

[33] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[34] Edwin T Jaynes. Information theory and statistical mechanics. ii. *Physical review*, 108(2):171, 1957.

[35] Adom Giffin and Ariel Caticha. Updating probabilities with data and moments. *arXiv preprint arXiv:0708.1593*, 2007.

[36] Y Alhassid, N Agmon, and RD Levine. An upper bound for the entropy and its applications to the maximal entropy problem. *Chemical Physics Letters*, 53(1):22–26, 1978.