New Approaches to Protein NMR Automation

by

Babak Alipanahi Ramandi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2011

Author's Declaration

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

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

Babak Alipanahi Ramandi

Abstract

The three-dimensional structure of a protein molecule is the key to understanding its biological and physiological properties. A major problem in bioinformatics is to efficiently determine the three-dimensional structures of query proteins. Protein NMR structure determination is one of the main experimental methods and is comprised of: (*i*) protein sample production and isotope labelling, (*ii*) collecting NMR spectra, and (*iii*) analysis of the spectra to produce the protein structure. In protein NMR, the three-dimensional structure is determined by exploiting a set of distance restraints between spatially proximate atoms. Currently, no practical automated protein NMR method exists that is without human intervention. We first propose a complete automated protein NMR pipeline, which can efficiently be used to determine the structures of moderate sized proteins. Second, we propose a novel and efficient semidefinite programming-based (SDP) protein structure determination method.

The proposed automated protein NMR pipeline consists of three modules: (*i*) an automated peak picking method, called PICKY, (*ii*) a backbone chemical shift assignment method, called IPASS, and (*iii*) a protein structure determination method, called FALCON-NMR. When tested on four real protein data sets, this pipeline can produce structures with reasonable accuracies, starting from NMR spectra. This general method can be applied to other macromolecule structure determination methods. For example, a promising application is RNA NMR-assisted secondary structure determination.

In the second part of this thesis, due to the shortcomings of FALCON-NMR, we propose a novel SDP-based protein structure determination method from NMR data, called SPROS. Most of the existing prominent protein NMR structure determination methods are based on molecular dynamics coupled with a simulated annealing schedule. In these methods, an objective function representing the error between observed and given distance restraints is minimized; these objective functions are highly non-convex and difficult to optimize. Euclidean distance geometry methods based on SDP provide a natural formulation for realizing a three-dimensional structure from a set of given distance constraints. However, the complexity of the SDP solvers increases cubically with the input matrix size, i.e., the number of atoms in the protein, and the number of constraints. In fact, the complexity of SDP solvers is a major obstacle in their applicability to the protein NMR problem.

To overcome these limitations, the SPROS method models the protein molecule as a set of intersecting two- and three-dimensional cliques. We adapt and extend a technique called semidefinite facial reduction for the SDP matrix size reduction, which makes the SDP problem size approximately one quarter of the original problem. The reduced problem is solved nearly one hundred times faster and is more robust against numerical problems. Reasonably accurate results were obtained when SPROS was applied to a set of 20 real protein data sets.

Acknowledgements

I wish to express my sincere gratitude to my supervisors, Professors Ming Li and Ali Ghodsi, for their support, encouragement, and advice throughout the duration of my doctorate studies. They are both fantastic mentors and remarkable human beings. In addition to providing guidance for my research, they made arrangements for me to meet with professionals in my field of study. My supervisors believed in me and knew that I would successfully complete my degree. I would like to extend my appreciation to my committee members, Professors Dong Xu, Bin Ma, Brendan McConkey, and Brian Ingalls for their constructive comments. I thank them for reading my thesis and providing valuable feedback that improved my thesis.

I would also like to thank the faculty and the students at the David R. Cheriton School of Computer Scienece, especially the Professors at the Bioinformatics Group: Professors Forbes Burkowski and Dan Brown. I am grateful to Ms. Nicole Keshav, who kindly helped me in proofreading my thesis; her comments and notes greatly improved my thesis. I cherish every moment I spent with my collaborators: Professor Logan Donaldson, Professor Henry Wolkowicz, Dr. Nathan Krislock and Pooyan Khajehpour. I am also grateful of my co-authors for helping me in my research: Dr. Emre Karakoc, Dr. Xin Gao, Dr. Shuai-Cheng Li, Guangyu Feng, Yuzhong Zhao, Xuefeng Cui, and Michael Biggs. I will miss all of my labmates at the Bioinformatics lab.

I highly appreciate the second-to-none CSCF support staff. I am also very grateful of the David R. Cheriton Graduate Scholarship that really helped me to focus on my research, instead of worrying about financial matters.

I will never forget my wonderful memories with my amazing friends: Ghasem, Hossein, Ehsan, Mohsen, Siamak, Akbar, Javad, Amir, Hamid, Bashir, Mahdi, Mohammad, among many others.

I am thankful to my family, my beloved wife, Ms. Motahareh Armin, who has been my best friend, one who I can always count on as a wise companion. I am also grateful to my precious parents, Mr. Abbas Alipanahi and Ms. Parinaz Rahemi, for their unconditional love and supportive role during my study. My family was always there for me and taught me to have confidence in my work and abilities.

Finally, we were blessed with an amazing baby boy, Ameen, towards the end of my doctorate studies. He is the best gift that God has given to us.

Dedication

This thesis is dedicated to my beloved wife, my precious son, and my dear parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background

1.1 NMR Physics

1.1.1 Nuclear Spin

In this section, we briefly review the physics behind the nuclear magnetic resonance (NMR) phenomenon, without delving much into the details of quantum mechanics. The NMR phenomenon arises from the fact that some nuclei have a property called *nuclear spin*; nuclei with a nonzero nuclear spin are NMR active; so that these nuclei can be thought of as very tiny magnets. Nuclear spin is characterized by a quantity called *nuclear spin quantity number*, I (Keeler, 2010). If p is the number of protons and n is the number of neutrons in the nucleus, then: (i) if p and n are both even, then $I = 0$, e.g., ^{12}C; (ii) if $p + n$ is odd, then $I = k + 1/2$, for some $k \geq 0$, e.g., for ^{1}H and ^{15}N we have $I = \frac{1}{2}$; and (iii) if p and n are both odd, then nuclear spin is $I = k + 1$, for some $k \geq 0$, e.g., for ^{14}N we have $I = 1$. Nuclei with $I > 1/2$ demonstrate quadrupole moments that have a very short lifetime; therefore, in NMR we focus only on nuclei with spin-$\frac{1}{2}$ ($I = 1/2$), such as ^{1}H, ^{13}C, and ^{15}N (Cavanagh et al., 2006).

Each nucleus with a nonzero nuclear spin has a corresponding vector called *nuclear spin angular momentum*, $\boldsymbol{I}$, whose magnitude is given by[1]:

$$\|\boldsymbol{I}\| = \hbar\sqrt{I(I+1)},$$

where $\hbar$ is the Planck's constant ($h = 6.63 \times 10^{-34}$ m^2 kg s^{-1}) divided by 2π. If we assume that the initial point of $\boldsymbol{I}$ is located at the origin, then its terminal point spans a circle

[1]The italic boldface notation indicates vectors, so I is a scalar different from $\boldsymbol{I}$, which is a vector.

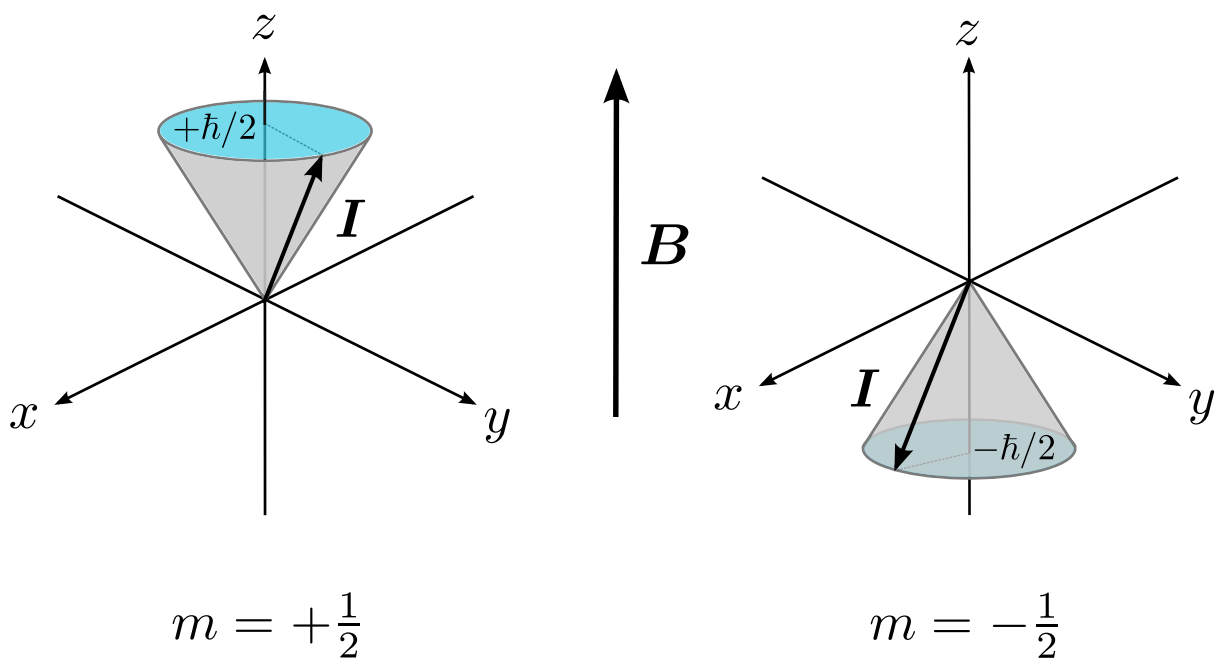


Figure 1.1: The nuclear spin angular momentum for $m = +\frac{1}{2}$ and $m = -\frac{1}{2}$.

with a known radius. That is, $\boldsymbol{I}$ can only be on the lateral surface of a right circular cone, as illustrated in Fig. 1.1. Moreover, we assume that the z-axis coincides with the axis of the aforementioned cone. We know that the projection of $\boldsymbol{I}$ on the z-axis, or I_z, can only take a discrete set of values:

$$I_z = \hbar m,$$

where $m \in \{-I, -I+1, \ldots, I-1, I\}$ is the *magnetic quantum number*. In our case, i.e., $I = 1/2$, $m \in \{-1/2, +1/2\}$. The nuclear spin angular momentum is depicted for $m = +1/2$ and $m = -1/2$ in Fig. 1.1. The magnitude of the vector $\boldsymbol{I}$ and the size of I_z are fixed, i.e., $\|\boldsymbol{I}\| = \sqrt{3}\hbar/2$ and $|I_z| = \hbar/2$ (Cavanagh et al., 2006).

Nuclear magnetic moment

Another property of nuclei with nonzero nuclear spin angular momentum, $\boldsymbol{I}$, is called the *nuclear magnetic moment* (magnetic moment), $\boldsymbol{\mu}$, which is a vector collinear with $\boldsymbol{I}$ given by

$$\boldsymbol{\mu} = \gamma \boldsymbol{I}, \tag{1.1}$$

where γ is called the gyromagnetic ratio and is different for each nuclide. The values of γ

for a few nuclides of interest are listed in Table 1.1. In the next sections we focus only on the case where $\gamma > 0$.

Table 1.1: Gyromagnetic and natural abundances of selected nuclides (Hore, 1995)

Nuclide	$\gamma/10^7$	Natural abundance (percentile)
^{1}H	26.75	99.985
^{2}H	4.11	0.015
^{13}C	6.73	1.108
^{14}N	1.93	99.63
^{15}N	-2.71	0.37

In the presence of an external homogeneous magnetic field in the z-direction of a rigid axis attached to the NMR laboratory, $\boldsymbol{B} = B_0\boldsymbol{k}$ ($\boldsymbol{k}$ is the unit vector in the z-direction), the magnetic moments of individual nuclei cannot have arbitrary orientations; their z-axis coincides with that of the NMR laboratory. The quantum spin states have energies defined as:

$$E = -\boldsymbol{\mu}^\top \boldsymbol{B} = -\gamma I_z B_0 = -\gamma m \hbar B_0. \tag{1.2}$$

The spin state with $m = 1/2$ is called the α-state and has energy $E_\alpha = -\gamma\hbar B_0/2$ and the spin state with $m = -1/2$ is called the β-state and has energy $E_\beta = +\gamma\hbar B_0/2$. The difference between the energy of two spin states is $\Delta E = E_\beta - E_\alpha = \gamma\hbar B_0$. If $\gamma > 0$, the α-state is more populated by spins than the β-state, because it has less energy and is more stable. At a thermal equilibrium, the ratio of spins in the β-state over the α-state is given by the Boltzman distribution:

$$\frac{N_\beta}{N_\alpha} = \exp\left(-\frac{\Delta E}{k_B T}\right) = \exp\left(-\frac{\gamma\hbar B_0}{k_B T}\right), \tag{1.3}$$

where $k_B = 1.38 \times 10^{-23}\ \mathrm{m^2\,kg\,s^{-2}\,K^{-1}}$ is the Boltzmann constant and T is the absolute temperature (Hore, 1995). We can simplify (1.3) and approximate the normalized difference in population at a thermal equilibrium as:

$$\frac{\Delta N_{\mathrm{eq}}}{N} = \frac{N_\alpha - N_\beta}{N} \approx \frac{\Delta E}{2k_B T}, \tag{1.4}$$

where N is the total number of spins. The overpopulation of the less energetic state is called *nuclear Zeeman splitting*, which removes the degeneracy of spin states (Levitt, 2008). For example, at room temperature ($T = 293^\circ$ K), for a typical NMR instrument magnet

($B_0 = 7$ T), and for hydrogen nuclides ($\gamma = 2.675 \times 10^8$ T^{-1} s^{-1}), $\Delta N_{\text{eq}}/N \approx 5 \times 10^{-5}$. That is, the α-state is slightly more populated than the β-state. This is due to the much smaller magnitude of ΔE compared to $k_B T$, which indicates that ΔN_{eq} is dominated by the temperature.

Bulk magnetic moment

The *bulk magnetic moment* vector, $\boldsymbol{M}$, or the net magnetic moment of a set of spins, is given by the vector sum of all individual $\boldsymbol{\mu}_i$s:

$$\boldsymbol{M} = \sum_{i=1}^{N} \boldsymbol{\mu}_i. \tag{1.5}$$

We decompose $\boldsymbol{M}$ into two components:

$$\boldsymbol{M} = \boldsymbol{M}_z + \boldsymbol{M}_{xy}, \tag{1.6}$$

where $\boldsymbol{M}_z$ is the longitudinal component or the projection on the z-axis, and $\boldsymbol{M}_{xy}$ is the traverse component or the projection of $\boldsymbol{M}$ on the xy-plane. When no magnetic field is applied, no quantum state is preferred and the individual $\boldsymbol{u}_i$s have random isotropic orientations; therefore, they sum up to zero and $\|\boldsymbol{M}\| = 0$ (Levitt, 2008). When the field is applied after equilibrium we have:

$$\|\boldsymbol{M}_{\text{eq}}\| \approx \frac{N\gamma^2\hbar^2 B_0}{4k_B T} = \frac{\gamma\hbar}{2}\Delta N_{\text{eq}}, \tag{1.7}$$

which indicates dependence of $\|\boldsymbol{M}_{\text{eq}}\|$ on γ and the strength of the magnetic field, B_0 (Cavanagh et al., 2006). Individual $\boldsymbol{\mu}_i$s' projections on the xy-plane are random and sum up to zero; therefore, $\boldsymbol{M}_{z,eq} = \boldsymbol{M}_{\text{eq}}$ and $\|\boldsymbol{M}_{xy,eq}\| = 0$. In the special case where $B_0 = 0$, then $\Delta N_{\text{eq}} = 0$, so there is no bulk magnetic moment, as expected.

Larmor frequency

We know that magnetic moments of individual spins precesses around $\boldsymbol{B}$ on the lateral surfaces of the aforementioned cones (see Fig. 1.1) with the *Larmor frequency* given by:

$$\omega_0 = -\gamma B_0; \tag{1.8}$$

the negative frequency indicates a clockwise precession (Cavanagh et al., 2006).

Looking at the spins from a different point of view, it turns out that the difference between the energy of two spin states is $\Delta E = \gamma\hbar B_0$; therefore, a photon with energy ΔE can cause low- to high-energy state transition. From Planck's law the frequency of the photon is given by:

$$\nu = \frac{\Delta E}{\hbar} = \gamma B_0. \tag{1.9}$$

Note that we can change the states of the spins by simply applying an electromagnetic radiation with frequency ν, called the *resonance frequency*, equal to the Larmor frequency. When $\boldsymbol{B}$ is present, we can apply a sufficiently strong electromagnetic radiation to make the population of both states equal, i.e., $N_\alpha \approx N_\beta$; when this happens the NMR signal fades.

1.1.2 Nuclear Magnetic Resonance

Magnetic moment vectors of all spins precess *incoherently* around $\boldsymbol{B}$ with the Laromor frequency, that is, every spin has a random different phase. Consequently, the traverse component $\boldsymbol{M}_{xy,eq} = \boldsymbol{0}$ and $\boldsymbol{M}_{z,eq} = \boldsymbol{M}_{\text{eq}}$. In Fig. 1.2.a, the bulk magnetic moment at equilibrium is depicted. The detector is placed in the xy-plane, because the powerful magnetic field in the z-direction, $\boldsymbol{B}$, would mask any signal and render it undetectable. Since $\boldsymbol{M}_{\text{eq}}$ has no traverse component, nothing is picked up by the detector.

Generally when electromagnetic radiation, $\boldsymbol{B}_r(t)$, with Laromor frequency and perpendicular to the static magnetic field is applied to the spins, all the spins precess coherently[2] (with coherent phases). Consequently, the bulk magnetic moment bends and ultimately lies on the xy-plane; therefore, $\|\boldsymbol{M}_{xy}\| \neq 0$ in a phenomenon called *nuclear magnetic resonance*. After resonance, the detector in the xy-plane can detect the precession of $\boldsymbol{M}_{xy}$, as depicted in Fig. 1.2.b.

Continuous wave NMR

There are several nuclei in the protein molecule, each with a different resonance frequency. To detect and record their corresponding frequencies, either (i) the frequency of the electromagnetic radiation, called *carrier* frequency, must sweep a frequency range or (ii) the strength of the magnetic field, B_0, must be varied to match the resonance frequency of a nucleus with the frequency of the electromagnetic radiation. This procedure is called continuous wave NMR, which was the prevalent technology in the early days of NMR

[2] An insightful animation demonstrating this phenomenon is available at `http://vam.anest.ufl.edu/forensic/nmr.html`.

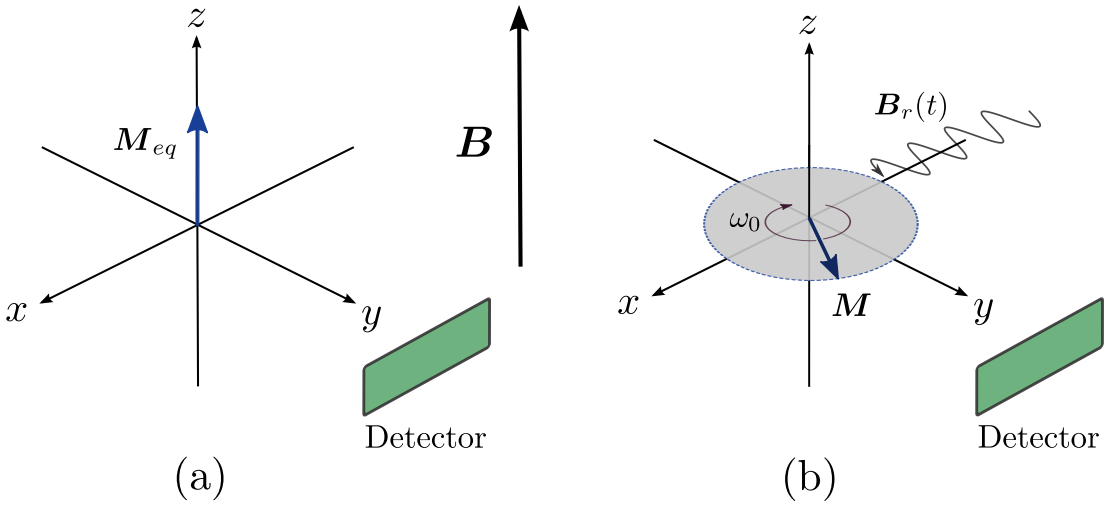


Figure 1.2: The bulk magnetic moment in the presence of $\boldsymbol{B}$ alone (a), and in the presence of $\boldsymbol{B}$ and an electromagnetic radiation $\boldsymbol{B}_r(t)$ with Laromor frequency.

(1945-1970) (Hore, 1995). Currently, continuous wave is obsolete and has been replaced by *pulse* NMR, in which short and intense pulses are applied to the sample.

The precessing bulk magnetic moment induces a current in the detector coil; this induced current is called the NMR *signal* or *free-induction decay* (FID). Detailed explanation of NMR experiments and many of the pulse techniques are described by Levitt (2008).

Relaxation

Although we have only discussed the steady state of spins, three questions arise about the transient behavior of spins: (*i*) what is the transient behavior of spins? (*ii*); what happens when spins are magnetically disturbed?; and (*iii*) how long does it take for spins to reach their equilibrium states? Two very important *relaxation* processes called *spin-lattice relaxation* and *spin-spin relaxation* address these questions. These relaxations are critical for understanding the NMR phenomenon and are described by the Bloch equations (Cavanagh et al., 2006). We focus on the solutions. Finally, to analyze the bulk magnetic moment and its components, we assume they are functions of time:

$$\boldsymbol{M}(t) = \boldsymbol{M}_z(t) + \boldsymbol{M}_{xy}(t).$$

Spin-lattice relaxation In the absence of the magnetic field, two spin states are equally populated, that is $\Delta N = 0$ and hence $\|\boldsymbol{M}(t)\| = 0$. Assume that at $t = 0$ the field is applied to the spins; N_α becomes larger and N_β becomes smaller and ultimately ΔN reaches ΔN_{eq} given in (1.4). If we assume an exponential growth, then we have:

$$\Delta N(t) = \Delta N_{\text{eq}} \left(1 - \exp(-t/T_1)\right),$$

where T_1 is called the *spin-lattice relaxation time* (Hore, 1995) and is also called the *longitudinal relaxation time* (Wüthrich, 1986). As the spins lose energy to transit to the lower energy state, they dissipate energy to the surroundings or to the "lattice". Since $\Delta N(t)$ and $\boldsymbol{M}_z(t)$ are linearly related, we have:

$$\|\boldsymbol{M}_z(t)\| = \|\boldsymbol{M}_{\text{eq}}\| \left(1 - \exp(-t/T_1)\right), \tag{1.10}$$

where $\boldsymbol{M}$ is the equilibrium bulk magnetic moment given in (1.7). Equation (1.10) is insightful; for example, if T_1 were very large, the thermal equilibrium would take very long, rendering the process of protein NMR infeasible.

Spin-spin relaxation A more important relaxation process happens when the electromagnetic radiation applied to the spins ($\boldsymbol{B}_r(t)$ in Fig. 1.2.b) is removed. The spins lose their phase coherence and after some time, they return to their equilibrium states, where $\|\boldsymbol{M}_{xy}\| = 0$, as in Fig. 1.2.a. Assume that the spins are at equilibrium and phase coherence; at $t = 0$ we remove $\boldsymbol{B}_r(t)$. The *spin-spin relaxation time*, T_2, is defined as the time constant in the exponential decay of magnitude of $\boldsymbol{M}_{xy}(t)$:

$$\|\boldsymbol{M}_{xy}(t)\| = \|\boldsymbol{M}_{xy}(0)\| \exp(-t/T_2),$$

where $\|\boldsymbol{M}_{xy}(0)\|$ is the magnitude of $\boldsymbol{M}_{xy}(t)$ at $t = 0$ (Cavanagh et al., 2006). Alternatively, T_2 is sometimes also called the *traverse relaxation time*. We know that $T_2 \leq T_1$, more importantly, T_2 increases as the molecule size increases. In fact, T_2 is a limiting factor on the maximum protein size that NMR can process (Wüthrich, 1986).

1.1.3 Chemical Shift

The Larmor frequency's relation (1.8) relates the strength of magnetic field to the frequency of the precession. For single isolated nuclei, the field strength is the same as the external field; however, in a macromolecule such as a protein, different nuclei are connected to each other through bonds and can locally *augment* or *oppose* the static magnetic field. The effect of the local electron density is called *nuclear shielding* and changes the Larmor frequency

of each nucleus as:

$$\omega = -\gamma(1-\sigma)B_0, \quad \text{or} \quad \omega = (1-\sigma)\omega_0,$$

where σ is the *average isotropic shielding constant* of the nucleus (Cavanagh et al., 2006). This phenomenon is the cornerstone of protein NMR and illustrates the fact that each nucleus experiences a different effective local magnetic field:

$$B_{\text{eff}} = (1-\sigma)B_0.$$

The strength of the magnetic field, B_0, varies from one NMR instrument to another; therefore, resonance frequency of the same nuclide varies in different machines. To facilitate comparison of the frequencies recorded on different instruments, NMR spectroscopists define a unitless normalized feature called *chemical shift* given by:

$$\delta = \frac{\omega - \omega_{\text{ref}}}{\omega_{\text{ref}}} \times 10^6 \ \ (\text{ppm}),$$

where ω_{ref} is the reference frequency for each nuclide. Chemical shift is expressed in parts per million (ppm).

Spin-spin coupling

Another phenomenon that can affect chemical shifts of nuclei is called spin-spin coupling, which is the isotropic magnetic interaction between nuclei that are connected with each other through chemical bonds; it is alternatively called *scalar coupling* or $\boldsymbol{J}$-coupling. If a nucleus A has spin-spin couplings with a set of nuclei {X}, its resonance frequency is changed to:

$$\omega_{\text{A}} = (1-\sigma_{\text{A}})\omega_0 - 2\pi \sum J_{\text{AX}} m_{\text{X}}, \tag{1.11}$$

where J_{AX} measured in Hertz is the *spin-spin coupling constant* between A and X and m_{X} is the magnetic quantum number of X (Hore, 1995). Spin-spin coupling is notable because it is independent of the field strength. Moreover, J_{AX} can be both positive or negative, i.e., spin-spin coupling can increase or decrease the resonance frequency. Spin-spin coupling is caused by the indirect magnetic interaction of electrons involved in the chemical bonds. Consequently, it is negligible when two nuclei are separated by more than three bonds.

1.1.4 The Nuclear Overhauser Effect

The *Nuclear Overhauser Effect* or *Enhancement* (NOE) is undoubtedly the most important phenomenon in the protein NMR process, because it is one of the major sources of structural information. Before introducing NOE, we need to describe *dipolar coupling*.

Dipolar coupling

Two magnetic dipoles interact with each other and can increase or decrease their energies. For example, two magnets repel each other if their north poles are close. If their opposite poles are close, they attract each other. In contrast to isotropic spin-spin coupling, dipolar coupling is the interaction between two spins that is dependent on both the distance and orientation of the two spins. Dipolar coupling does not change the resonance frequencies in liquids, because molecules in liquid collide with each other and tumble around constantly, changing the orientation and distance of the spins with dipolar coupling. As a result, the average dipolar coupling effect sums to zero. However, they are responsible for the NOE phenomenon (Hore, 1995).

1.1.5 NOE

The NOE phenomenon is the interaction between two spins without spin-spin coupling and with appreciable dipolar coupling. In NOE, the coupled spins experience *cross-relaxation*, i.e., both spins simultaneously relax to their lower (or higher) energy states (simultaneous spin-flips). To quantify NOE between two spins, a parameter η is used, which is defined as the ratio of change in the intensity of the NMR signal of a nucleus, in the presence of dipolar coupling:

$$\eta = \frac{i - i_0}{i_0},$$

where i_0 is the normal intensity and i is the intensity in the presence of the dipolar coupling (Hore, 1995). For protons i can be smaller, reversed, and even zero, such that $-1 \leq \eta \leq \frac{1}{2}$. Cross-relaxation rates have a d^{-6} dependence on the distance between two spins; therefore, NOE is only observed when two spins are close to each other, $2 \leq d \leq 5$ Å. This is the main reason for the critical importance of NOE, because in Nuclear Overhauser Enhancement Spectroscopy (NOESY) spatially proximate protons can be identified. This provides a rich source of structural information that can be used to determine the 3D structure of protein.

1.2 Protein NMR Process

In this section, we briefly review the protein NMR structure determination process, focusing mainly on the problems related to computer science. Before describing the entire pipeline, we first present definitions of some of the terms frequently used.

1.2.1 NMR Terminology

Chemical shift In this study chemical shift or interchangeably resonance frequency is the ID or signature of each one of the NMR active nuclei in the protein molecule. For example ω_ℓ is the resonance frequency of nucleus A_ℓ (the corresponding chemical shift is denoted by δ_ℓ). The resonance frequency of a nucleus is assumed to be identical in all NMR experiments.

Peak Peak is a D-dimensional vector of resonance frequencies $\boldsymbol{p} = [\omega_1, \ldots, \omega_D]^\top \in \mathbb{R}^D$ that represents a group of D nuclei $\{\mathrm{A}_d\}$ that are coupled through bonds (spin-spin coupling) or through space (dipolar coupling). It should be noted that resonance frequencies and chemical shifts are used interchangeably; for example, a peak may be denoted as $\boldsymbol{p} = [\delta_1, \ldots, \delta_D]$.

Spectrum Spectrum is determined by computing the Fourier transform of the time-domain NMR signal or FID (Levitt, 2008). For a set of peaks of the same dimension D, $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K\}$, the corresponding D-dimensional continuous spectrum is a function $S : \mathbb{R}^D \to \mathbb{R}$ defined as:

$$S(\boldsymbol{\omega}) = \sum_{k=1}^{K} a_k \; f_k(\boldsymbol{\omega} - \boldsymbol{p}_k) + \eta, \tag{1.12}$$

where a_k is the intensity of the signal associated with $\boldsymbol{p}_k$, η is the additive white Gaussian noise, and $\boldsymbol{\omega} \in \mathbb{R}^D$. We assume that $f_k(.)$ can be factorized into functions called *lineshapes*:

$$f_k(\boldsymbol{\omega}) = \prod_{d=1}^{D} f_k^{(d)}(\omega_d), \tag{1.13}$$

where $f_k^{(d)}(\cdot)$s are called the *absorption Lorentzian* lineshapes (Levitt, 2008). To illustrate lineshapes, consider a one dimensional noiseless spectrum with only one peak at ω_0; the spectrum is given by:

$$S(\omega) = \frac{\lambda}{\lambda^2 + (\omega - \omega_0)^2} \, ,$$

where $\lambda = 1/T_2$ is called the *coherence decay rate constant* and is equal to the inverse of spin-spin relaxation time (Levitt, 2008). The hypothetical spectrum $S(\omega)$ is illustrated in Fig. 1.3. Lineshapes are the same as in this spectrum, i.e., $f_k^{(d)}(\cdot) = S(\cdot)$.

Resonance *linewidth*, $\Delta\nu_{1/2}$, is an important property in NMR that determines the width of lineshapes in Hz. It is defined as the full-width at half-height (FWHH) of the

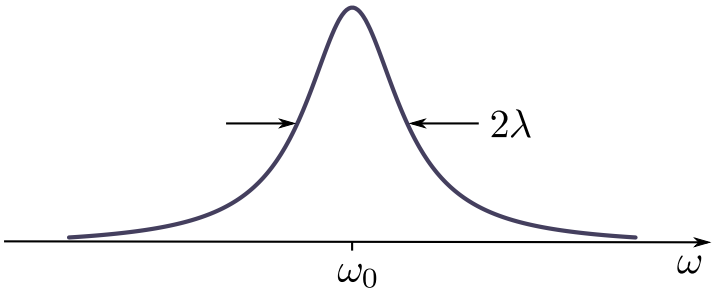


Figure 1.3: A one-dimensional spectrum with at peak at $\omega = \omega_0$. Linewidth is defined as the full-width at half-height (FWHH) of lineshape, $\Delta\nu_{1/2} = 2\lambda$.

lineshape:

$$\Delta\nu_{1/2} = 2\lambda = 2/T_2\pi.$$

As the molecule size increases, T_2 becomes smaller and $\Delta\nu_{1/2}$ becomes larger, resulting in wider lineshapes; this is one of the most crucial limiting factors for high-resolution NMR, because lineshape becomes so wide and sinks so deep that it is buried under noise.

Finally, in our model, local artifacts and noisy peaks are modeled as extra fake peaks in the peak set $\mathcal{P}$ that forms the spectrum and the variance of the noise term is assumed to be uniform throughout the spectrum.

Sampled spectrum NMR signal are sampled, because computing the Fourier transform of a continuous time-domain signal is not tractable. Because the NMR signals are sampled, the corresponding spectra are determined by computing the discrete Fourier transform of the time-domain signals. This is equivalent to sampling continuous NMR spectra at discrete frequencies with *resolution* θ_ℓ for nuclide A_ℓ (the minimum frequency spacing is θ_ℓ). The sampled spectrum, or spectrum hereinafter, is a matrix for 2D spectra and a tensor for 3D spectra whose elements are intensities. For example, for a 2D spectrum, the magnitude of (i, j)-th element is given by:

$$S_{ij} = \sum_{k=1}^{K} a_k \, f_k(\hat{\boldsymbol{\omega}}_{ij} - \boldsymbol{p}_k) + \eta_{ij},$$

where η_{ij}s are the i.i.d. sampled Gaussian noise values and the sampled frequencies are given by:

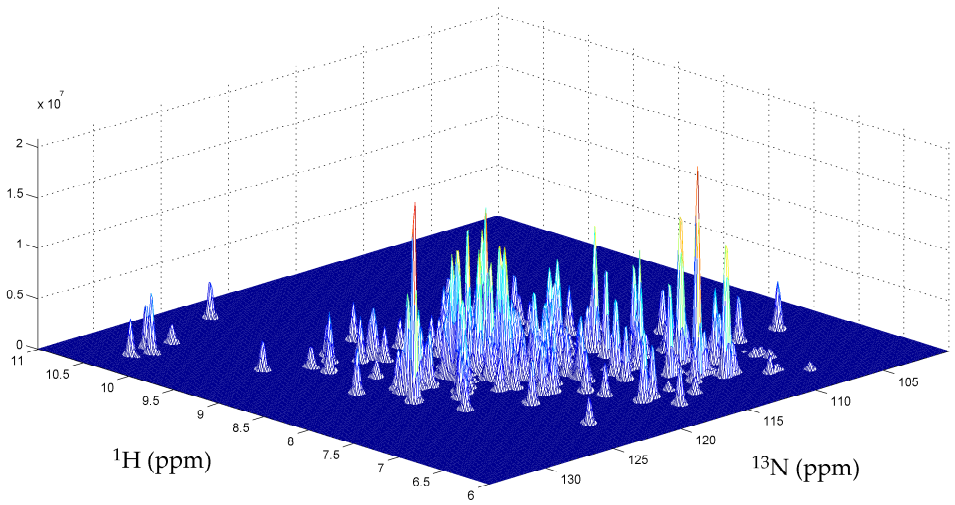


Figure 1.4: ^{15}N-HSQC spectrum of YST0336 after removal of noisy samples.

$$\hat{\boldsymbol{\omega}}_{ij} = \begin{bmatrix} \hat{\omega}_1 \\ \hat{\omega}_2 \end{bmatrix} = \begin{bmatrix} \omega s_1 \\ \omega s_2 \end{bmatrix} + \begin{bmatrix} i \cdot \theta_1 \\ j \cdot \theta_2 \end{bmatrix}. \tag{1.14}$$

In (1.14), ωs_ℓ is the start or *offset* frequency for nuclide A_ℓ and i and j are given by:

$$\begin{aligned} i &= \left\lfloor \frac{\omega_1 - \omega s_1 + \theta_1/2}{\theta_1} \right\rfloor, \\ j &= \left\lfloor \frac{\omega_2 - \omega s_2 + \theta_2/2}{\theta_2} \right\rfloor, \end{aligned} \tag{1.15}$$

where $\lfloor \cdot \rfloor$ is the floor operator. A real sampled 2D spectrum for protein YST0336 is shown in Fig. 1.4.

Uncertainty sources There are two major sources of uncertainty in the sampled spectra:

1. Sampling adds quantization errors, that is, $\xi_\ell = |\omega_\ell - \hat{\omega}_\ell|$ is the difference between actual and quantized chemical shifts. In the worst case for $\omega_\ell = \omega s_\ell + i\theta_\ell \pm \frac{1}{2}\theta_\ell$; the quantization error magnitude is $|\xi_\ell| = \frac{1}{2}\theta_\ell$. The quantization error is zero mean with a

$\theta_\ell^2/12$ variance. Consequently, the smaller the θ_ℓ, the less significant the quantization error.

2. Different NMR spectra, $\{S^{(q)}\}$, might have different offset frequencies, $\{\omega s_\ell^{(q)}\}$, and different resolutions, $\{\theta_\ell^{(q)}\}$. As a result, a nuclide might have different resonance frequencies in different spectra. To analyze this, consider a nucleus with an actual chemical shift ω that has the following sampled frequencies in two different spectra:

$$\begin{aligned} \omega &= \hat{\omega} + \xi, \quad |\xi| \leq \theta/2, \\ \omega &= \hat{\omega}' + \xi', \quad |\xi'| \leq \theta'/2, \end{aligned}$$

It can be shown that:

$$|\hat{\omega} - \hat{\omega}'| = |\xi - \xi'| \leq (\theta + \theta')/2,$$

if $\theta = \theta'$ and $\omega s = \omega s'$, then $|\xi - \xi'| = 0$. However, if $\omega s \neq \omega s'$, then $|\xi - \xi'|$ can be as large as $(\theta + \theta')/2$. For example, if $\theta = \theta'$, $\omega s - \omega s' = k\theta + \epsilon$, for some integer k and very small positive number $\epsilon > 0$, and $\omega - \omega s = p\theta + \theta/2$, we have $|\xi - \xi'| = \theta$, where p is an integer.

Due to the aforementioned uncertainties, observed chemical shifts are always noisy. Moreover, a nucleus has slightly different chemical shifts in various spectra.

1.2.2 Common NMR Spectra

There are numerous NMR experiments, each designed to present information about specific sets of nuclei. We introduce only the prevalent set of spectra that are used in our experiments and collected in most NMR labs. We use the widely adopted atom nomenclature (Markley et al., 1998). Therefore, N, HN, CA, HA, CB, and C denote chemical shifts of amide nitrogen, amide hydrogen, alpha carbon, alpha hydrogen, beta carbon, and carbonyl carbon, respectively. Moreover, HX, $\mathrm{X} \in \mathcal{X}$ denotes all the hydrogens in the set $\mathcal{X}$. The subscript attached to each of the chemical shifts corresponds to the residue in the amino acid sequence to which they belong, e.g., N_i denotes amide nitrogen chemical shift of residue i. Last, it should be noted the peaks described here are *expected* to exist in the corresponding spectra, but there is no guarantee that they actually exist. The details of these experiments are described by Cavanagh et al. (2006). Common spectra are listed below.

^{15}N-HSQC A sensitive 2D spectrum that detects the N and HN chemical shift pair, i.e. a peak at $[\mathrm{N}_i, \mathrm{HN}_i]^\top$ for residue i and is referred to as the "root pair." This spectrum is sometimes referred to as a protein signature. Nuclei involved in a sample ^{15}N-HSQC peak are depicted in Fig. 1.5.

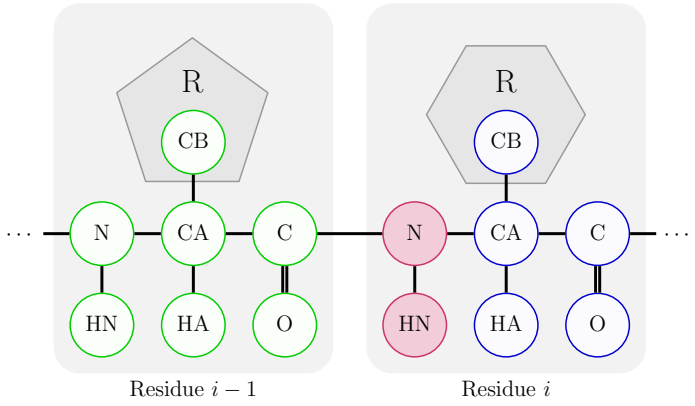


Figure 1.5: Nuclei involved in a sample ^{15}N-HSQC peak, $[\mathrm{N}_i, \mathrm{HN}_i]^\top$, are shaded.

HNCO A 3D spectrum that detects the N, HN, and C chemical shifts. Ideally, it generates one peak for each residue: $[\mathrm{N}_i, \mathrm{C}_{i-1}, \mathrm{HN}_i]^\top$. HNCO is one of the most sensitive 3D experiments. Nuclei involved in a sample HNCO peak are depicted in Fig. 1.6.

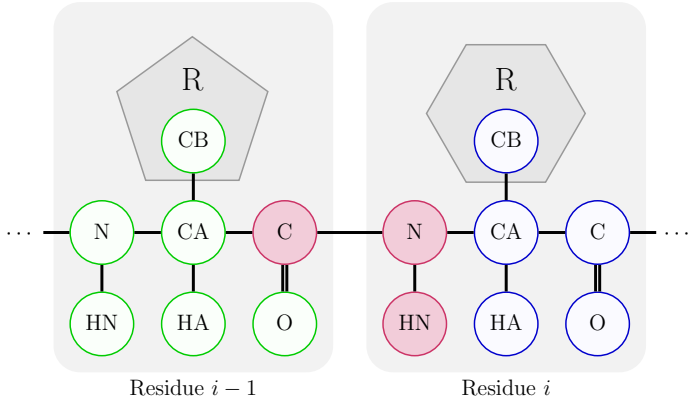


Figure 1.6: Nuclei involved in a sample HNCO peak, $[\mathrm{N}_i, \mathrm{C}_{i-1}, \mathrm{HN}_i]^\top$, are shaded.

HNCACB A 3D spectrum that detects N, HN, CA, and CB chemical shifts. In the ideal case, it generates four peaks for each residue: $[\mathrm{N}_i, \mathrm{CA}_i, \mathrm{HN}_i]^\top$, $[\mathrm{N}_i, \mathrm{CB}_i, \mathrm{HN}_i]^\top$, $[\mathrm{N}_i, \mathrm{CA}_{i-1}, \mathrm{HN}_i]^\top$, and $[\mathrm{N}_i, \mathrm{CB}_{i-1}, \mathrm{HN}_i]^\top$. The sign of the intensity values can be used to differentiate between CAs and CBs. Nuclei involved in a sample HNCACB peak are depicted in Fig. 1.7.

CBCA(CO)NH A 3D spectrum that detects N, HN, CA, and CB chemical shifts. In the ideal case, two peaks are generated for each residue: $[\mathrm{N}_i, \mathrm{CA}_{i-1}, \mathrm{HN}_i]^\top$ and $[\mathrm{N}_i, \mathrm{CB}_{i-1}, \mathrm{HN}_i]^\top$. Nuclei involved in a sample CBCA(CO)NH peak are depicted in Fig. 1.8.

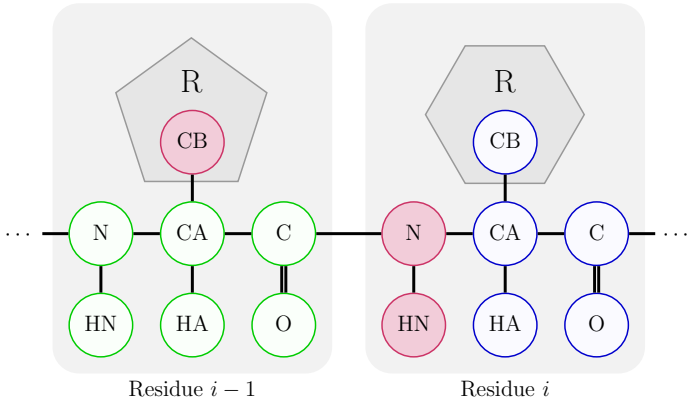


Figure 1.7: Nuclei involved in a sample HNCACB peak, $[\mathrm{N}_i, \mathrm{CB}_{i-1}, \mathrm{HN}_i]^\top$, are shaded.

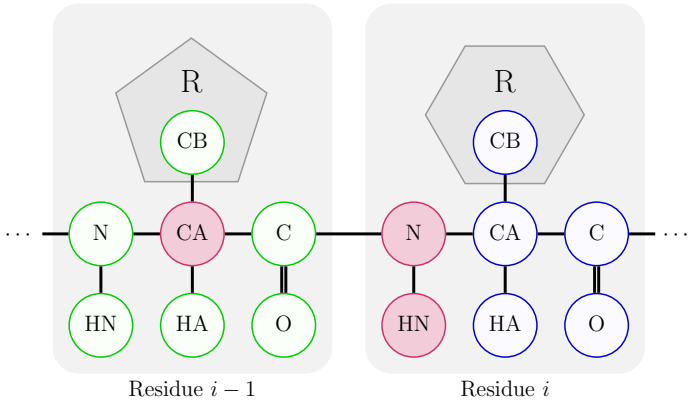


Figure 1.8: Nuclei involved in a sample CBCA(CO)NH peak, $[\mathrm{N}_i, \mathrm{CA}_{i-1}, \mathrm{HN}_i]^\top$, are shaded.

HNCA A 3D spectrum that detects N, HN, and CA chemical shifts. Ideally, it generates two peaks for each residue: $[\mathrm{N}_i, \mathrm{CA}_i, \mathrm{HN}_i]^\top$ and $[\mathrm{N}_i, \mathrm{CA}_{i-1}, \mathrm{HN}_i]^\top$. Nuclei involved in a sample HNCA peak are depicted in Fig. 1.9.

H(CCO)NH-TOCSY A 3D spectrum that detects N, HN, and HX chemical shifts, where:

$$\mathcal{X} = \{\alpha \text{ hydrogen} \cup \text{all hydrogens on the side chain}\} \text{ of residue } i-1. \tag{1.16}$$

In the ideal case, it generates peaks in $[\mathrm{N}_i, \mathrm{HX}, \mathrm{HN}_i]^\top$ format. The number of peaks for each residue depends on the type of preceding residue and the number of its side chain hydrogens. Nuclei involved in a sample H(CCO)NH-TOCSY peak are depicted in Fig. 1.10.

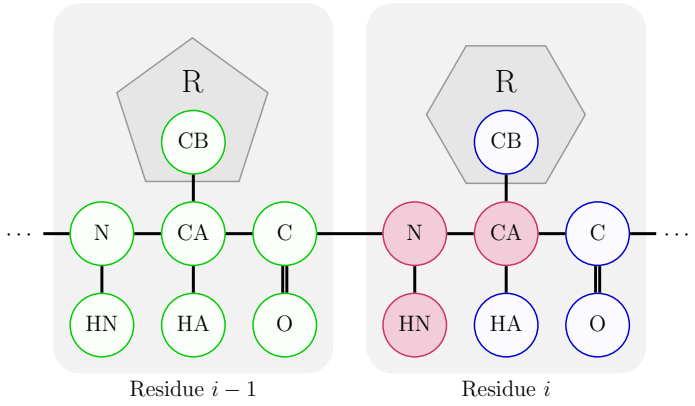


Figure 1.9: Nuclei involved in a sample HNCA peak, $[\mathrm{N}_i, \mathrm{CA}_i, \mathrm{HN}_i]^\top$, are shaded.

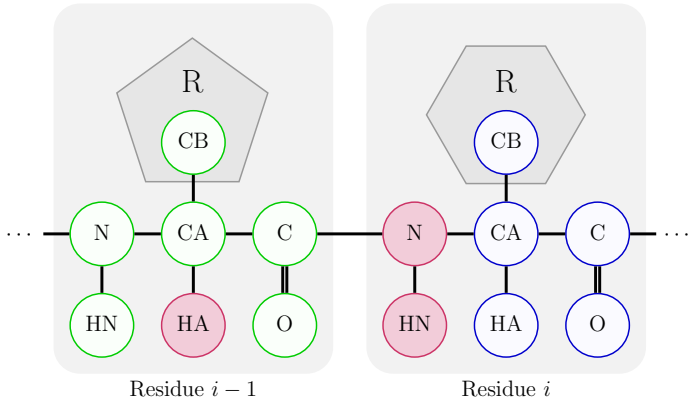


Figure 1.10: Nuclei involved in a sample H(CCO)NH-TOCSY peak, $[\mathrm{N}_i, \mathrm{HA}_{i-1}, \mathrm{HN}_i]^\top$, are shaded.

^{15}N-NOESY A 3D spectrum that detects dipolar (through space) coupling between protons. It detects N, HN, and HX chemical shifts, such that

$$\mathcal{X} = \{\mathrm{X} \mid d(\mathrm{HX}, \mathrm{HN}) \leq 5\text{Å}\}, \tag{1.17}$$

where $d(\mathrm{A}, \mathrm{B})$ is the distance between atoms A and B in the molecular structure. Ideally, it generates peaks in $[\mathrm{N}_i, \mathrm{HX}, \mathrm{HN}_i]^\top$ format for all close protons with appreciable dipolar coupling. The ^{15}N-NOESY spectrum is critical in the protein NMR process, since HX could be anywhere in the protein chain, even from a residue far from residue i in the sequence (long-range contact). The number of peaks varies from residue to residue and depends on the molecular structure of the protein. Nuclei involved in a sample ^{15}N-NOESY peak are depicted in Fig. 1.11.

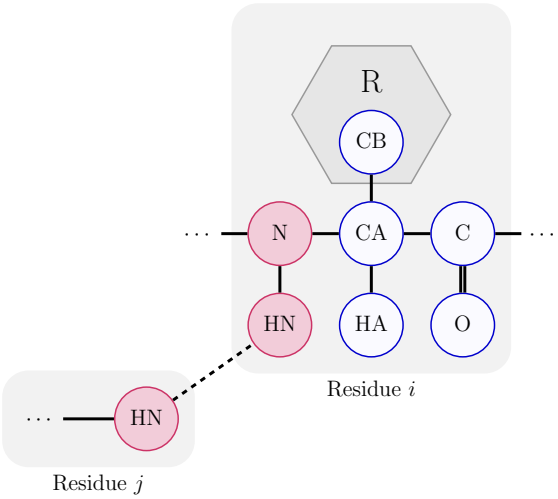


Figure 1.11: Nuclei involved in a sample ^{15}N-NOESY peak, $[\mathrm{N}_i, \mathrm{HN}_j, \mathrm{HN}_i]^\top$, are shaded. HN_i and HN_j have dipolar coupling (dashed line) and they are not chemically bonded. Only the HN of residue j is shown.

1.2.3 Protein NMR Steps

Here, we briefly review the major steps of protein NMR structure determination. The most prominent strategy used by most spectroscopists is proposed by Wüthrich (1986), which includes isotope labeling, data collection and processing, peak picking, chemical shift assignment, NOE peak assignment, and finally, structure calculation. The protein NMR process flowchart is depicted in Fig. 1.12.

Isotope labeling and collecting NMR spectra

NMR active isotopes such as ^{15}N and ^{13}C are not abundant (see Table 1.1). Therefore, most proteins contain NMR inactive/unsuitable nuclides such as ^{12}C and ^{14}N. In the absence of ^{15}N and ^{13}C isotopes, only ^{1}H is NMR active. The number of hydrogen atoms and the linewidth grow (approximately) linearly with the size of the protein, resulting in severe overlapping of peaks. Consequently, the NMR methods that only use ^{1}H are limited to proteins with the molecular masses 10–12 kDa (Cavanagh et al., 2006). To resolve degeneracy of chemical shifts, NMR inactive nuclides are replaced with their expensive NMR active isotopes in a process called *isotope labeling*, in an NMR laboratory. After labeling, the protein sample is put in an NMR instrument and a set of different spectra are recorded. This is a costly procedure that takes two to three weeks to be completed. This step is not the focus of automated NMR methods, because it is not a computer science problem.

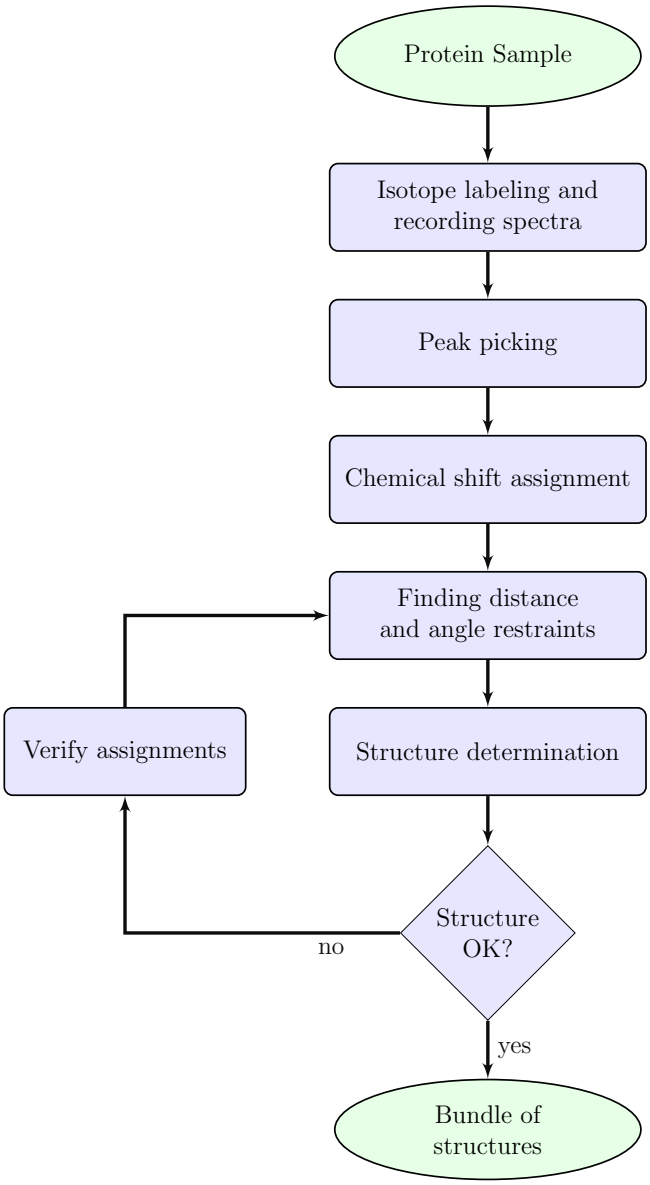


Figure 1.12: Protein NMR steps.

Peak-picking

After NMR spectra are recorded, they are analyzed by an NMR spectroscopist for identification of the corresponding peaks in a process called *peak-picking*. This is a cumbersome and labor-intensive procedure that takes around one week to be carried out, because visu-

alizing 3D and 4D spectra is not straightforward, not withstanding the problem of finding the location of peaks. In this process, multiple spectra are cross-referenced simultaneously to distinguish between fake (noisy) and real peaks.

Chemical shift assignment

Chemical shift assignment is the process of labeling NMR active nuclei with their corresponding chemical shifts. Peaks from different spectra are considered and are superimposed based on their common nuclides. Chemical shift assignment is usually divided into two steps: (*i*) backbone resonance assignment and (*ii*) side chain resonance assignment. Assignment is error-prone due to missing peaks, artifacts, and degeneracy of chemical shifts, and is mostly done by using semi-automated interactive software. Assignment is mostly done in one week; however, in some cases it could take weeks or even months.

NOE assignment and generating distance constraints

With knowledge of chemical shifts of nuclei, NOE information obtained from NOESY spectra can be used to generate distance constraints. That is, we can extract structural information from NMR experiments. Due to their intrinsically vague nature, NOE peaks usually generate ambiguous distance constraints; therefore, they should be rigorously inspected and verified by an NMR expert to discard the contradicting or false constraints. This process usually takes one to two weeks.

Structure calculation and refinement

The final step in protein NMR is structure calculation. The existing methods exploit (*i*) the set of ambiguous distance constraints, (*ii*) domain knowledge about protein structures (for example the chainlike arrangement of amino acids in the protein or the planar structure of peptides), and (*iii*) the set of approximate torsion angle restraints obtained from chemical shifts of fragments of protein. All of this information is used to generate a bundle of structures in an iterative process. For example, a hypothetical 2D chainlike molecule with seven atoms is depicted in Fig. 1.13 (left). From NOE experiments, we know that atom pairs (1,7), (2,5), and (4,6) are close to each other. These constraints can be used to generate the structure or *fold* of the chain as depicted in Fig. 1.13 (right).

Finally, if a set of structural quality constraints are not satisfied, then the assignments are inspected, the distance constraints are refined, and the procedure is repeated. This procedure usually takes one to two weeks to finish.

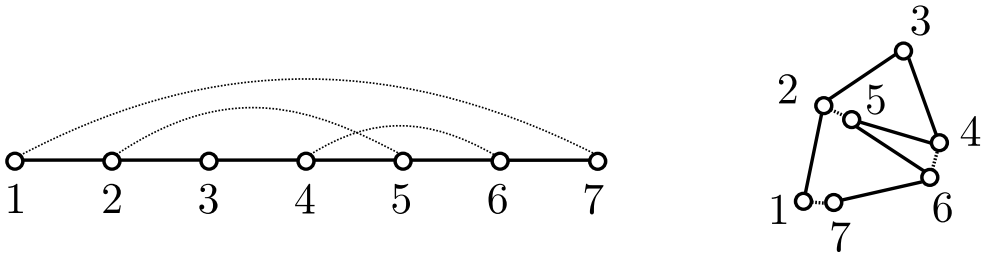


Figure 1.13: The process of using NOE constraints: Three NOE constraints in a hypothetical 2D molecule with seven atoms (left). The molecule after being folded to satisfy the constraints.

1.3 Protein NMR Automation

There are two major experimental protein structure determination methods: X-ray crystallography and protein NMR structure determination. The X-ray crystallography method has been successfully automated and the need for an "expert" is removed. Therefore, the X-ray method is faster and cheaper than before. Moreover, experts can devote their time to more challenging problems (Chandonia and Brenner, 2006). However, the X-ray method automation is more straightforward than NMR, because using X-ray, there is a single "correct" method to analyze the data and infer the structure, while in NMR, every laboratory uses its "own" data analysis and processing tools and methods (Jahnke, 2007). To make matters even worse, most of the existing methods require human intervention, which introduces irreproducible bias and subjectivity.

The protein NMR structure determination process has been widely adopted; it accounts for nearly 15% of deposited structures in the PDB (Spronk et al., 2004). Moreover, modern stable isotope labeling techniques enable collection of NMR spectra for RNAs which are longer than a hundred nucleotides. For example, NAPSS uses unassigned NMR RNA spectra to assist a dynamic RNA folding algorithm (Hart et al., 2008). Although the global fold of the structures determined by NMR is the same as the structure determined by the X-ray method, at the local level there are appreciable differences (Andrec et al., 2007). The most notable weakness of NMR is low sensitivity of spectra (Billeter et al., 2008), meaning that many peaks are missing (due to fast conformational changes or relaxation problems) or are buried in the noise (Williamson and Craven, 2009). Therefore, every attempt must be made to extract as much information as possible from spectra, while the methods involved should be designed with robustness and error-tolerance in mind.

Protein NMR automation can be very beneficial: it can make the protein NMR process fast and feasible and can remove the expert subjectivity from the process and make it "repeatable." Moreover, it can create valuable research time for NMR experts, similar to the X-ray method (Williamson and Craven, 2009). Protein NMR automation, or at least some modules of it, have received much attention during the past few years. There have been numerous research attempts to automate the protein NMR structure determination process (Jung et al., 2004; Huang et al., 2005; López-Méndez and Güntert, 2006; Shen et al., 2008, 2009a). Especially, FLYA (López-Méndez and Güntert, 2006) has succeeded in determining the structures of several small proteins fully automatically, except this method nonetheless requires manually set parameters, such as noise ratio thresholds in the peak picking step. In general, most of the existing methods require different levels of human expert intervention, which, again, incorporates expert experience and subjectivity. Some recent extensive reviews have been done by Huang et al. (2005), Güntert (2009), and Williamson and Craven (2009).

Chapter 2

Backbone Resonance Assignment

2.1 Introduction

The backbone resonance assignment, also known as the chemical shift assignment, plays a vital role in the entire NMR protein structure determination process. This step of the protein NMR process has attracted the most computational attention over the last decade (Bartels et al., 1996; Zimmerman et al., 1997; Gronwald et al., 1998; Bailey-Kellogg et al., 2000; Güntert et al., 2000; Tian et al., 2001; Xu et al., 2002; Hus et al., 2002; Erdmann and Rule, 2002; Pristovsek et al., 2002; Coggins and Zhou, 2003; Morris et al., 2004; Jung and Zweckstetter, 2004; Langmead and Donald, 2004; Langmead et al., 2004; Masse and Keller, 2005; Eghbalnia et al., 2005; Bailey-Kellogg et al., 2005; Vitek et al., 2005; Pristovsek and Franzoni, 2004; Wu et al., 2006; Wan and Lin, 2007; Lemak et al., 2008; Volk et al., 2008; Xiong and Bailey-Kellogg, 2007; Xiong et al., 2008; Stratmann et al., 2009; Apaydin et al., 2011; Donald and Martin, 2009; Crippen et al., 2010; Apaydin et al., 2008). The goal of resonance assignment is to assign the picked peaks from NMR spectra to their corresponding nuclei of the target proteins. A thorough review of assignment methods has been done by Williamson and Craven (2009) and Güntert (2009).

Typically, the backbone resonance assignment is divided into three sub-problems, as mentioned by Moseley and Montelione: (*i*) forming spin systems, (*ii*) linking spin systems into fragments, and (*iii*) mapping the fragments to the target sequence. A "spin system" denotes a group of coupled nuclei that can be observed as cross-peaks in one or more spectra. Usually, spin systems contain both inter-residue and intra-residue information. The existing methods can be classified into two groups: assignment methods that require spin systems (Coggins and Zhou, 2003; Jung and Zweckstetter, 2004; Masse and Keller, 2005; Lemak et al., 2008) and assignment methods that do not require spin systems (Zimmerman et al., 1997; Wu et al., 2006; Wan and Lin, 2007; Volk et al., 2008). However, the latter

assignment methods always require high quality peak lists with a very small number of missing or false peaks and little difference in the chemical shift of the same nucleus in different spectra. Therefore, for most cases, the experiments carried out in such studies, are based on either manually (or semi-automatically) picked and refined peak lists by spectroscopists, or on synthetic peak lists formed by assigned chemical shifts in a known protein database such as BioMagResBank (BMRB) (Seavey et al., 1991). It should be noted that most of the "spin-system-free" methods are flexible with the input spectra set, provided that the spectra contain enough CA, CB, and/or C connectivity information.

The AUTOASSIGN method (Zimmerman et al., 1997) is a multi-stage constraint-based expert system. The idea of AUTOASSIGN is the best first search, which assigns the strongest fragment matches first, and then gradually relaxes restrictions to assign weaker matches. The AUTOLINK method (Masse and Keller, 2005) is an attempt to mimic human logic by a fuzzy logic and relative hypothesis prioritization method. AUTOLINK extracts spin system connectivity information from the NOESY data. Wu et al. (2006) later proposed a weighted maximum independent set formulation for the assignment problem. They provided a comprehensive summary of the different sources of the spectra errors in the lab experiments, and further simulated these errors on perfect datasets, synthesized from BMRB database.

MARS (Jung and Zweckstetter, 2004), one of the widely acknowledged assignment methods, is different from its ancestors in that it applies the consensus idea to multiple runs of assignments, where each run is carried out to optimize different objective functions. For the local assignment, MARS uses the best first search to find the local fit of the fragments, comprising as many as five spin systems. For global assignment, however, MARS optimizes the global pseudo-energy function, which measures how well a spin system matches a residue in the target protein. The pseudo-energy is based on the likelihood of observing a certain chemical shift for an amino acid type in the BMRB database.

Lemak et al. (2008) and Volk et al. (2008) proposed two sophisticated methods to solve the resonance assignment problem on the most up-to-date NMR spectra. The ABACUS method (Lemak et al., 2008) takes unassigned peaks from NOESY, COSY (correlation spectroscopy), and TOCSY (total correlation spectroscopy), as well as database-derived likelihoods, as the input. A multi-canonical Monte Carlo procedure, Fragment Monte Carlo (FMC), is used to perform sequence-specific assignments. In the MATCH method (Volk et al., 2008), both the global and local optimization strategies merge, where the six-dimensional APSY spectrum (Hiller et al., 2005; Fiorito et al., 2006) is the input.

The SAGA method (Crippen et al., 2010), is a recent algorithm that provides three search algorithms, which can be selected by the user. It solves the assignment problem using three different techniques: (*i*) finding the largest clique, (*ii*) performing a greedy search, and (*iii*) performing a branch-and-bound method for searching the assignment space. The SAGA method needs at least a clean (well-edited) HN(CO)CA peak list and

generously-picked HNCACB and HNCA peak lists.

To fully automatically determine the structures from raw NMR data, the backbone resonance assignment method should not be designed and tested in isolation from other steps. Such an assignment algorithm must be designed to take sufficiently noisy input data (generated fully automatically) and to output the assignments with sufficient quality to allow automatic structure determination. Although most of the previously proposed assignment methods were designed to deal with noisy peaks, most were only tested on either synthetic peaks, which were simulated from the deposited chemical shifts from BMRB, or from real peaks, which were manually or semi-automatically picked and refined. Therefore, none of those methods works well on fully automatically picked peaks, which contain significantly higher levels of noise and artifacts than manually or semi-automatically picked peaks.

Here, we present an error-tolerant assignment method, IPASS (Alipanahi et al., 2011a), that processes automated peak-picking results to enable a fully automatic NMR protein structure determination process. The IPASS method is an Integer Linear Programming (ILP) based assignment method. It also contains a new spin system forming algorithm, an improved probabilistic spin system typing model, and a novel connectivity extraction method. The performance of IPASS is evaluated on the noisy automatically-picked peak lists of four real proteins by PICKY (Alipanahi et al., 2009). The success of IPASS on these real proteins is mainly due to its more robust ILP module, the more error-tolerant spin system forming module, and the more capable spin system typing module, as well as its capability to use additional but commonly available data, such as ^{15}N-NOESY cross peaks.

2.2 Proposed Method

2.2.1 Problem Formulation

Given an amino acid sequence of a protein with n residues as $r_1 r_2 \dots r_n$, define $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$. Spin systems are given as $\mathcal{S} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_m\}$, where $\boldsymbol{s}_j$ is a vector of the chemical shifts. Then, the assignment problem is finding the correct mapping between spin systems and residues, expressed as $f : \mathcal{S} \to \mathcal{R}$. Due to the imperfect NMR spectra, peak picking, and spin systems forming, the number of spin systems can be smaller, larger, or equal to the number of residues, and some spin systems might not be assigned. If $\boldsymbol{s}_j$ is assigned to r_i, then, in the ideal case it is defined as:

$$\boldsymbol{s}_j = [\mathrm{N}_i, \mathrm{HN}_i, \mathrm{CA}_i, \mathrm{CB}_i, \mathrm{CA}_{i-1}, \mathrm{CB}_{i-1}]^\top, \tag{2.1}$$

In practice, it is possible that some of the chemical shifts are missing or incorrect.

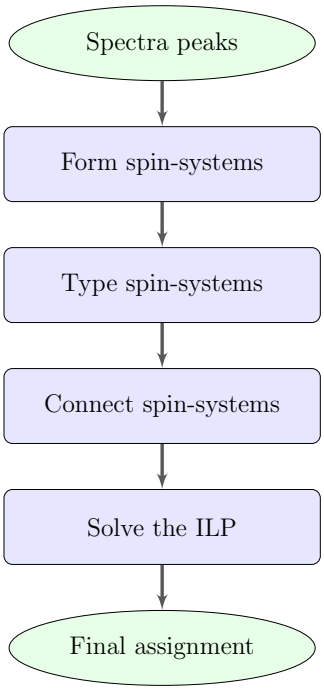


Figure 2.1: IPASS flowchart.

2.2.2 The General Strategy

IPASS has several modules which run in a sequence, as depicted in Fig. 2.1. The IPASS method can be summarized as follows:

Forming Spin Systems This is a pre-processing step for resonance assignment. A new graph-based method is developed to group chemical shifts from the peaks of different spectra into spin systems. The core idea is to apply a two-stage clustering method and use self-adapted thresholds to manage the experimental errors, chemical shift errors. The input to the spin system-forming module is the set of picked peaks of ^{15}N-HSQC, HNCACB, CBCA(CO)NH, and/or HNCA spectra, and the output is a set of spin systems.

Typing Spin Systems Estimating the potential amino acids that can generate the observed chemical shifts in a spin system is called *typing*. Spin systems are typed in a probabilistic framework by using the statistics collected from a large protein dataset. The IPASS method uses both the protein sequence and the secondary structure prediction to type spin systems. Rather than finding the probability of mapping a spin system to a single

residue, IPASS uses both the information from the current and the preceding residues, and thus computes the probability of mapping a spin system to two consecutive residues. The output of this step is a set of potential spin systems and their probabilities, associated with each residue.

Connectivity Information Extraction Two spin systems are connected if they can be mapped to two consecutive residues. The connections are detected by inter-residue and intra-residue chemical shift information. Chemical shifts within spin systems are noisy, such that a low threshold results in many undetected true connections. However, a large threshold results in many false connections, making the ILP problem intractable. In IPASS, two sets of connections are defined: a set of highly reliable connections based on the CA and CB chemical shifts and the information extracted from the ^{15}N-NOESY peaks. Furthermore, a set of less-reliable connections are detected by a larger threshold. By using reliable connections, a set of fragments is determined and their combinations are enumerated. Fixing the fragments eliminates many false connections and renders solving the ILP problem feasible.

Integer Linear Programming At this step, there are a few spin system candidates, and their probabilities for each residue. The assignment is formulated as an ILP problem to find the globally optimal assignment. Instead of solving one *large* ILP problem, IPASS solves many *small* tractable ILP problems. After fixing fragments, the ILP problem size is dramatically reduced and can be easily solved within a few seconds. The ILP is solved for all combinations of the fragments, and the one with the best score is picked as the final assignment.

2.2.3 Forming Spin Systems

The goal of this step is to group the chemical shifts that are determined from different NMR spectra into spin systems. Each spin system corresponds to nuclei within a small vicinity, usually associated with a residue of the target protein. During the spin system forming process, the chemical shifts are grouped in relation to their local environments and are not assigned to a certain residue in the protein sequence. Here, spin systems are viewed as the building blocks of the backbone resonance assignment process.

The problem of finding spin systems is modeled as a graph theoretical problem[1]. Ideally, the chemical shifts should be the same for each nucleus in the NMR spectra. In practice, a perfect peak set is not available due to experimental errors, artifacts, biases, and resolution

[1]The spin system forming module is developed by E. Karakoc and F. Balbach.

differences. Typically, an error as high as 0.5 ppm is expected to exist in the ^{15}N and ^{13}C chemical shifts, and a shift as high as 0.05 ppm in the ^{1}H chemical shifts. Therefore, an exact match algorithm is not possible for comparing the different experimental NMR data. To overcome this problem, each peak is represented as a point in the multi-dimensional space, where each dimension corresponds to a certain type of nuclei, such as ^{15}N, ^{1}H, or ^{13}C. Initially, all the peaks are represented in the nuclei space. In practice, the peaks that belong to the same residue will not coincide at the same HN and N position, but are clustered nearby.

Stage I

In the first stage, the peaks within each 3D spectrum are connected according to their N and HN (root pair) chemical shifts. Each spectrum provides multiple peaks for the same residue (see 1.2.2), and these peaks should be in the small vicinity of each other. The peaks that have similar root pairs are grouped by using a weighted Euclidean distance function. For example, the distance between root pairs of $\boldsymbol{p}_x = [\mathrm{N}_x, \mathrm{C}_x, \mathrm{HN}_x]^\top$ in CBCA(CO)NH spectrum and $\boldsymbol{p}_y = [\mathrm{N}_y, \mathrm{C}_y, \mathrm{HN}_y]^\top$ in HNCACB spectrum is defined as:

$$D_{\mathrm{rp}}(\boldsymbol{p}_x, \boldsymbol{p}_y) = \sqrt{(\mathrm{N}_x - \mathrm{N}_y)^2 + \gamma^2(\mathrm{HN}_x - \mathrm{HN}_y)^2}, \tag{2.2}$$

where γ is the scaling factor for the compensation of the difference in the resolution between HN and N. Usually, HN chemical shifts are 10 times more sensitive than the N chemical shifts, and so the default value of γ is 10. According to the distance defined in (2.2), each peak, $\boldsymbol{p}$, in a given spectrum is associated with its nearest neighbor, $\boldsymbol{p}_{nn}$. An edge is created between $\boldsymbol{p}$, and all the peaks that are closer to it than $2 \times D_{\mathrm{rp}}(\boldsymbol{p}, \boldsymbol{p}_{nn})$. Thus, IPASS does not have a preset threshold for forming spin systems. The edges between the peaks are directional, and the source is the reference peak, $\boldsymbol{p}$. The peaks connected to each other represent the peaks from the same root pair.

Stage II

The second step of generating the peak graph is to connect the peaks from different spectra. For example, the distance metric between the aforementioned peaks is defined as:

$$D(\boldsymbol{p}_x, \boldsymbol{p}_y) = \sqrt{(\mathrm{N}_x - \mathrm{N}_y)^2 + (\mathrm{C}_x - \mathrm{C}_y)^2 + \omega^2(\mathrm{HN}_x - \mathrm{HN}_y)^2}.$$

Similar to the aforementioned process, edges are created between $\boldsymbol{p}$ and its close vicinity peaks in other spectra, which are closer to than $2 \times D(\boldsymbol{p}, \boldsymbol{p}_{nn})$. All of the created edges

are directional. If there are two edges in both directions between two nodes, two edges are replaced by a non-directional edge. The remaining directed edges are removed.

After these two steps, each connected component represents a cluster that corresponds to a spin system in the resulting general peak graph. Some clusters might be incomplete as a result of missing peaks or over-crowded as a result of degenerate chemical shifts. The primary advantage of this approach is its generality. It can be applied to any set of available NMR spectra.

Last, a brute force approach is applied that searches all the possible combinations of the chemical shift values for different CA and CB nuclei in each cluster. If a unique combination of the chemical shifts exists and does not conflict with the peaks in the cluster, a spin system is generated.

2.2.4 Typing Spin Systems

The next step involves *typing* spin systems. Initially, any of the m spin systems can be mapped to any of the n residues. The objective of this step is to reduce the number of candidate spin systems for each residue, based on the chemical shift information. A statistical analysis of the deposited chemical shifts in the BMRB database reveals correlation among the chemical shifts, the amino acid types, and the secondary structure (see Table 2.1). These statistics are used to find the probability that one spin system is mapped to a certain residue.

Collecting Statistics

All the BMRB entries with a matched PDB entry were downloaded as of December 15, 2008. Then, 1168 protein sequences were clustered by using CD-HIT (Li and Godzik, 2006) with a 40% sequence identity level. From each cluster, only the longest sequence was retained, resulting in a data set of 805 non-redundant proteins. The DSSP method (Kabsch and Sander, 1993) was applied to compute the secondary structure types for all the residues. From 88,436 collected residues, for all the amino acids except Gly, 68,028 CA and CB chemical shift pairs were extracted. For Gly (which does not have a CB chemical shift) 6,577 CA chemical shifts were extracted. The mean vectors and covariance matrices were estimated for each amino acid and secondary structure type (see Table 2.1).

Probabilistic Typing

The IPASS method uses a novel probabilistic model for calculating the probability that spin system $\boldsymbol{s}_j$ can be mapped to residue r_i, $\Pr[r_i \mid \boldsymbol{s}_j]$, for the n residues and m spin systems.

Table 2.1: Mean and standard deviation values of CA and CB chemical shifts (μ_α and μ_β) divided based on amino acid and secondary structure type. $\rho_{\alpha,\beta}$ is the correlation coefficient between CA and CB chemical shifts in each case. C, S, and H denote coil, β-strand, and α-helix, respectively. The second column is the relational abundance of each amino acid type in our data set (in percentile). Distinct statistics of Ala, Gly, Ser, and Thr are shown in bold.

		μ_α			μ_β			$\rho_{\alpha,\beta}$		
A.A.	p (%)	C	S	H	C	S	H	C	S	H
Ala	7.5	52.4±1.6	50.9±1.3	54.7±1.4	**19.3±2.1**	**21.7**±2.4	**18.2**±1.4	-0.07	-0.06	-0.17
Arg	5.0	56.1±1.9	54.8±1.4	59.0±1.5	30.8±2.3	32.8±3.0	29.9±1.1	-0.14	-0.18	0.21
Asn	6.0	53.1±1.6	52.3±1.3	55.4±1.5	38.6±2.7	40.3±3.0	38.3±1.5	-0.01	0.12	0.16
Asp	4.1	54.1±1.7	53.4±1.5	56.8±1.3	41.1±2.7	42.5±1.9	40.4±1.3	0.04	0.09	0.12
Cys	1.6	57.2±2.7	56.2±1.9	62.0±3.1	33.6±6.5	33.9±6.8	30.1±5.3	-0.58	-0.43	-0.65
Glu	7.9	55.8±1.8	54.4±1.3	58.4±1.4	29.4±2.5	31.4±2.6	28.4±1.1	-0.16	-0.07	0.00
Gln	4.2	56.7±1.7	55.1±1.3	59.0±1.3	30.2±2.5	32.4±2.7	29.3±2.0	-0.13	-0.11	0.05
Gly	7.3	**45.3±1.4**	**45.0±2.0**	**46.7±1.1**	-	-	-	-	-	-
His	2.2	56.0±1.8	54.8±1.7	58.7±2.5	30.2±2.3	32.0±2.2	29.9±1.6	0.07	0.15	0.00
Ile	5.4	60.7±2.1	59.9±1.5	64.4±1.8	38.8±3.3	40.1±2.9	37.7±1.5	-0.07	-0.08	0.21
Leu	8.9	54.8±1.7	53.8±1.3	57.5±1.2	42.4±2.3	44.3±2.6	41.5±1.2	-0.06	-0.12	0.08
Lys	7.1	56.3±1.8	55.1±1.3	59.0±1.5	32.9±2.6	34.9±2.8	32.2±1.0	-0.09	-0.08	0.16
Met	2.1	55.4±1.6	54.2±1.2	57.9±1.8	33.1±2.6	35.3±3.3	32.2±1.7	0.01	0.09	0.49
Phe	3.7	57.4±2.0	56.4±1.5	60.7±2.0	39.8±2.4	41.7±2.7	38.9±1.4	-0.01	0.05	0.30
Pro	4.2	63.1±1.7	62.6±1.0	65.4±1.3	32.1±1.7	32.1±1.1	31.4±1.0	0.06	0.31	0.20
Ser	6.3	58.3±1.7	57.1±1.3	61.0±1.5	**63.9±1.4**	**65.1±1.4**	**62.9±1.4**	-0.13	0.09	-0.02
Thr	5.4	61.5±1.9	61.0±1.5	65.6±2.2	**69.8±1.7**	**70.5±1.6**	**68.5±2.4**	-0.20	-0.18	-0.09
Trp	1.2	56.9±2.1	56.1±1.6	59.9±2.0	30.2±5.3	31.5±1.9	29.1±1.3	0.00	-0.10	-0.05
Tyr	3.0	57.5±2.0	56.6±1.5	60.8±1.9	39.1±2.8	41.2±3.2	38.1±1.3	-0.09	0.05	0.33
Val	6.9	61.9±2.1	60.7±1.7	65.9±1.8	32.7±2.3	34.0±2.1	31.5±1.0	-0.23	-0.31	-0.09

Two vectors are defined for spin system $\boldsymbol{s}_j$: $\boldsymbol{c}_j := [\mathrm{CA}_i, \mathrm{CB}_i]^\top$ and $\hat{\boldsymbol{c}}_j := [\mathrm{CA}_{i-1}, \mathrm{CB}_{i-1}]^\top$. They contain the chemical shift information about the residue which $\boldsymbol{s}_j$ is mapped to, r_i, and its preceding residue, r_{i-1}, respectively. Furthermore, since the N and HN chemical shifts exhibit similar statistics for all amino acids, they are discarded. Therefore, $\Pr[r_i \,|\, \boldsymbol{s}_j]$, the probability that $\boldsymbol{c}_j$ and $\hat{\boldsymbol{c}}_j$ are mapped to r_i and r_{i-1}, respectively, can be written as in (2.3). If it is assumed that $\boldsymbol{c}_j$ and $\hat{\boldsymbol{c}}_j$ are independent[2], then (2.3) can be simplified to (2.4). By using the Bayes' rule, we can rewrite (2.4) as (2.5), where $a_p, a_q \in \mathcal{A}$, and $\mathcal{A}$ is

[2]We need to assume this independence, because otherwise there will be $(20 \times 3) \times (20 \times 3) = 3600$ different combinations of amino acid and secondary structure types, with only a few of chemical shift data pairs for each combination. The independence assumption between $\boldsymbol{c}_j$ and $\hat{\boldsymbol{c}}_j$ does not affect the accuracy noticeably.

the set of twenty amino acids.

$$\begin{aligned}\Pr[r_i \mid \boldsymbol{s}_j] &= \Pr[r_i = a_p, r_{i-1} = a_q \mid \boldsymbol{c}_j, \hat{\boldsymbol{c}}_j] && (2.3)\\ &= \Pr[r_i = a_p \mid \boldsymbol{c}_j] \times \Pr[r_{i-1} = a_q \mid \hat{\boldsymbol{c}}_j] && (2.4)\\ &= \frac{\Pr[\boldsymbol{c}_j \mid r_i = a_p]\Pr[r_i = a_p]}{\Pr[\boldsymbol{c}_j]} \times \frac{\Pr[\hat{\boldsymbol{c}}_j \mid r_{i-1} = a_q]\Pr[r_{i-1} = a_q]}{\Pr[\hat{\boldsymbol{c}}_j]} && (2.5)\end{aligned}$$

In (2.5), we assume that $\Pr[r_i = a_p]$ only depends on a_p and not on the position in the protein sequence, i. Therefore, $\Pr[r_i = a_p]$ can be estimated by the proportional abundance of amino acid a_p. In addition, by using the total probability law,

$$\begin{aligned}\Pr[\boldsymbol{c}_j] &= \textstyle\sum_{a_\ell \in \mathcal{A}} \Pr[\boldsymbol{c}_j \mid r_i = a_\ell]\Pr[r_i = a_\ell] && (2.6)\\ \Pr[\hat{\boldsymbol{c}}_j] &= \textstyle\sum_{a_\ell \in \mathcal{A}} \Pr[\hat{\boldsymbol{c}}_j \mid r_{i-1} = a_\ell]\Pr[r_{i-1} = a_\ell]\end{aligned}$$

It should be noted that in proline, N and HN do not resonate and no spin system is mapped to it, $\Pr[\boldsymbol{c}_j \mid r_i = \text{Pro}] = 0$. Furthermore, the chemical shifts depend on both the amino acid type and the secondary structure type. To incorporate the secondary structure information, the total probability law is used again and $\Pr[\boldsymbol{c}_j \mid r_i]$ is reformulated as:

$$\Pr[\boldsymbol{c}_j \mid r_i = a_\ell] = \sum_{k=1}^{3} \Pr[\boldsymbol{c}_j \mid r_i = a_\ell, \gamma_i = \sigma_k]\ \Pr[\gamma_i = \sigma_k], \tag{2.7}$$

where γ_i denotes the secondary structure state of r_i. For $k = 1, 2$, and 3, σ_k denotes random coil, β-strand, and α-helix, respectively. The PSIPRED method is used to estimate $\Pr[\gamma_i = \sigma_k]$ values (McGuffin et al., 2000). It is assumed that $\Pr[\boldsymbol{c}_j \mid r_i,\ \gamma_i]$ exhibits a joint Gaussian distribution due to the observed strong correlation between the CA and CB chemical shifts. By using the estimated covariance matrices, $\boldsymbol{\Sigma}_{\ell,k}$, and mean vectors, $\boldsymbol{\mu}_{\ell,k}$, we can write:

$$\Pr[\boldsymbol{c}_j \mid r_i = a_\ell,\ \gamma_i = \sigma_k] = \frac{1}{2\pi|\boldsymbol{\Sigma}_{\ell,k}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{c}_j - \boldsymbol{\mu}_{\ell,k})^\top \Sigma_{\ell,k}^{-1}(\boldsymbol{c}_j - \boldsymbol{\mu}_{\ell,k})\right). \tag{2.8}$$

Therefore, when one of ^{13}C chemical shifts is missing, the one-dimensional version of the Gaussian distribution is used. By substituting (2.8) in (2.7), $\Pr[\boldsymbol{c}_j \mid r_i = a_\ell]$ is computed. After computing $\Pr[\boldsymbol{c}_j]$ by using (2.6), the mappings that are very unlikely should be discarded. Therefore, if the condition in (2.9) holds for a_ℓ, $\Pr[\boldsymbol{c}_j \mid r_i = a_\ell]$ is set to zero:

$$\frac{\Pr[\boldsymbol{c}_j \mid r_i = a_\ell]\Pr[r_i = a_\ell]}{\Pr[\boldsymbol{c}_j]} < \epsilon \Rightarrow \Pr[\boldsymbol{c}_j \mid r_i = a_\ell] = 0 \tag{2.9}$$

The omission threshold, ϵ, is chosen as 0.001 similar to the approaches cited in the literature (Grishaev et al., 2005). This helps to reduce the number of candidate spin systems for each residue. After this step, the $\Pr[r_i \mid \boldsymbol{c}_j]$ values for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ are established. The next step is to find the connections among spin systems.

2.2.5 Connectivity Information Extraction

Connectivity information is extracted from the CA and CB chemical shifts, as well as from the ^{15}N-NOESY peaks information. Two sets of connections, *reliable* and *loose*, are defined. Although the spin system typing step can significantly reduce the number of candidate spin systems for each residue, this number remains large. Therefore, some highly reliable fragments are found and fixed.

Links

Consider two spin systems $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$ defined as:

$$\begin{aligned} \boldsymbol{s}_j &= [\mathrm{N}_i, \mathrm{HN}_i, \mathrm{CA}_i, \mathrm{CB}_i, \mathrm{CA}_{i-1}, \mathrm{CB}_{i-1}]^\top \\ \boldsymbol{s}_k &= [\mathrm{N}_{i'}, \mathrm{HN}_{i'}, \mathrm{CA}_{i'}, \mathrm{CB}_{i'}, \mathrm{CA}_{i'-1}, \mathrm{CB}_{i'-1}]^\top. \end{aligned} \tag{2.10}$$

These two spin systems are called *connected* or *linked* if $i' = i + 1$. In the ideal case for two linked spin systems $|\mathrm{CA}_i - \mathrm{CA}_{i'-1}| = 0$ and $|\mathrm{CB}_i - \mathrm{CB}_{i'-1}| = 0$, because $i' = i + 1$. However, CA_i and $\mathrm{CA}_{i'-1}$ may have originated from different spectra with different resolutions (see 1.2.1), so they may slightly differ. If $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$ satisfy at least one of the following two conditions without violating the other, then they are linked:

1. $|\mathrm{CA}_i - \mathrm{CA}_{i'-1}| \leq \delta_\alpha$
2. $|\mathrm{CB}_i - \mathrm{CB}_{i'-1}| \leq \delta_\beta$

In the first phase of finding connection, the goal is to find all potential links; therefore, we set $\delta_\alpha = \delta_\beta = 0.5$, which is a loose threshold.

Connectivity graph

In this step, a directed graph called a *connectivity* graph, $H = (\mathcal{W}, \mathcal{F})$, is constructed that indicates if two spin systems are linked. As mentioned earlier, two spin systems $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$ are linked if $\boldsymbol{s}_j \to r_i$ and $\boldsymbol{s}_k \to r_{i+1}$, so that they are mapped onto two consecutive residues. The graph is constructed as follows:

1. For each spin system put a node in H, i.e., $\mathcal{W} \leftarrow \mathcal{S}$
2. If $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$ are linked, then $f_{jk} \in \mathcal{F}$
3. For each edge f_{jk}, its reliability g_{jk} is computed (details to follow)
4. For two conflicting edges, e.g., f_{jk} and $f_{j'k}$, the less reliable edge is discarded. That is, if $g_{j'k} < g_{jk}$, then $\mathcal{F} \leftarrow \mathcal{F} \setminus f_{j'k}$

A sample connectivity graph is depicted in Fig. 2.2.

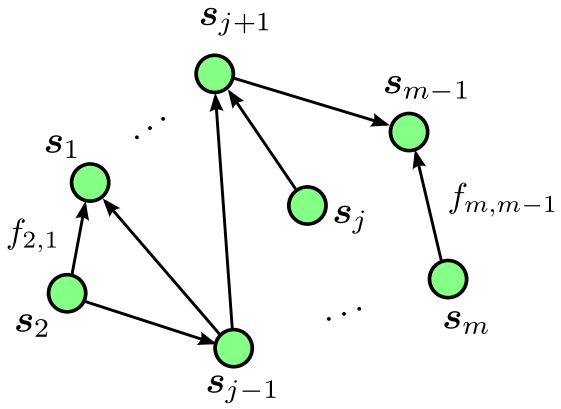


Figure 2.2: A sample connectivity graph; only a subset of nodes are shown. For clarity, only two edges are labeled.

Reliability of connections Initially, the reliability of all of the edges (links) is set to one:

$$g_{jk} = 1, \quad \forall f_{jk} \in \mathcal{F}. \tag{2.11}$$

Then, the edge reliabilities are updated as follows:

- If $|\mathrm{CA}_i - \mathrm{CA}_{i'-1}| \leq \delta'_\alpha$, then $g_{jk} \leftarrow g_{jk} + 10$.
- If $|\mathrm{CB}_i - \mathrm{CB}_{i'-1}| \leq \delta'_\beta$, then $g_{jk} \leftarrow g_{jk} + 10$.
- If $[\mathrm{N}_i, \mathrm{HN}_{i'}, \mathrm{HN}_i]^\top$ and $[\mathrm{N}_{i'}, \mathrm{HN}_i, \mathrm{HN}_{i'}]^\top$ peaks exist in the ^{15}N-NOESY spectrum, then $g_{jk} \leftarrow g_{jk} + 5$.

Since we want to elucidate more reliable connections, we set $\delta'_\alpha = \delta'_\beta = 0.05$ ppm. Moreover, if two spin systems are assigned to two adjacent residues on the target protein sequence, their hydrogen atoms in the amide groups should be close in the molecular structure, providing mutual peaks in the ^{15}N-NOESY spectrum. We assign a lower score for mutual ^{15}N-NOESY peaks, because of the degeneracy in HN chemical shifts.

After refining the connectivity graph (step 4), all the edges with $g_{jk} > 10$ are extracted and put in the *reliable* edge set, $\mathcal{F}_\mathcal{R}$. These reliables edges are used to form the fragments.

Fragment enumeration

All the reliable edges are searched for possible fragments. For example, if $f_{jk}, f_{kq} \in \mathcal{F}_\mathcal{R}$, then $\boldsymbol{s}_j$ is linked to $\boldsymbol{s}_k$ and $\boldsymbol{s}_k$ is linked to $\boldsymbol{s}_q$, so that $F = (\boldsymbol{s}_j, \boldsymbol{s}_k, \boldsymbol{s}_q)$ is a fragment of length three. After all fragments are found, we then find the possible mapping positions for each reliable fragment. Assume that p fragments, $F_1, \ldots, F_p$, are found, with lengths $\ell_1, \ldots, \ell_p$, respectively. Each fragment is shown as $F_q = (\boldsymbol{s}_{e_1}, \boldsymbol{s}_{e_2}, \ldots, \boldsymbol{s}_{e_{\ell_q}})$, where $\boldsymbol{s}_{e_j}$ is linked to $\boldsymbol{s}_{e_{j+1}}$ for $j = 1, \ldots, \ell_q - 1$. Fragments shorter than three spin systems, or fragments that are the substrings of other fragments, are discarded. For fragment F_q, a mapping score is defined for the i-th position in the target sequence, such that:

$$S_q(i) = -\sum_{k=1}^{\ell_q} \log\left(1 - \Pr[r_{i+k-1} \mid \boldsymbol{s}_{e_k}]\right), \quad 1 \le i \le n - \ell_q + 1. \tag{2.12}$$

If $S_q(i) > \ell_q \epsilon$ (ϵ in (2.9)), then i is added to the set of possible mappings of F_q, denoted $\mathcal{M}_q$. If $\mathcal{M}_q = \emptyset$, then F_q is discarded. After all the possible mappings are found, all combinations of fragments are enumerated. In each combination, no two fragments should be in conflict, i.e., they should not share any spin systems, and their mapped positions in the sequence should not overlap. Then, all the fragments within the combination are fixed. When spin system $\boldsymbol{s}_j$ is fixed to residue r_i, then:

$$\Pr[r_i \mid \boldsymbol{s}_{j'}] = 0, \quad j' \neq j, \tag{2.13}$$

$$\Pr[r_{i'} \mid \boldsymbol{s}_j] = 0, \quad i' \neq i. \tag{2.14}$$

That is, $\boldsymbol{s}_j$ is removed from all other candidate sets. After fixing all combinations, the ILP model (described in section 2.2.6) is called for each combination. From another point of view, this step can be interpreted as performing a local optimization to make the global optimization feasible.

The number of combinations is limited to 20000. In our experiments, no more than 200 combinations are discovered, because a strict threshold is used for finding the reliable fragments. If the number of combinations exceeds the upper bound, the fragments four in

length are discarded and so on. This process continues until the number of combinations becomes fewer than the predefined upper bound. After ILP is solved for each combination, the one with the highest score is selected as the final assignment.

2.2.6 Integer Linear Programming-based Assignment

Originally, all the m spin systems can be mapped to any residue r_i. However, in spin system typing and in the fragment fixing step, many $\Pr[r_i \mid \boldsymbol{s}_j]$ values are set to zero. Adapting a similar notion by Crippen et al. (2010), we introduce the *matching* graph. It is a bipartite graph, where nodes are sets of spin systems and residues. There is an edge between $\boldsymbol{s}_j$ and r_i if $\Pr[r_i \mid \boldsymbol{s}_j] \neq 0$, that is, $\boldsymbol{s}_j$ can be assigned to r_i. There are also two sets of auxiliary edges defined for better understanding of the problem. There is an edge between r_{i-1} and r_i and an edge between $\boldsymbol{s}_j$ and $\boldsymbol{s}_k$, if they are connected, i.e., $f_{jk} \in \mathcal{F}$. A sample matching graph is depicted in Fig. 2.3. The edges from the connectivity graph are depicted in Fig. 2.2 and are used to plot the auxiliary edges in the spin systems (bottom) row.

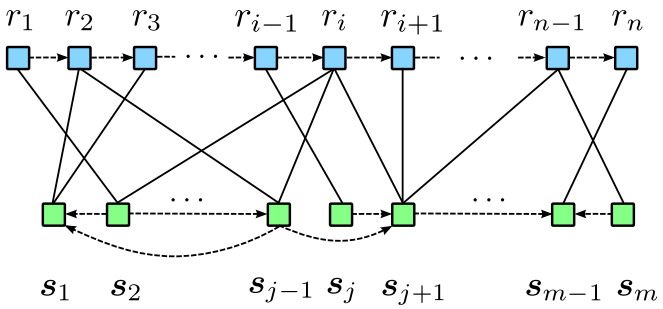


Figure 2.3: Illustration of the matching graph. Residues are put on the top row and the spin systems on the bottom row; a solid edge indicates that the corresponding spin system can be assigned to the residue it is connected to. Dashed edges are auxiliary edges not in the graph edge set.

The assignment problem is not a simple matching problem, because the edges cannot be picked independently. For example in Fig. 2.3, edges $(\boldsymbol{s}_j, r_{i-1})$ and $(\boldsymbol{s}_{j-1}, r_i)$ conflict with each other, because $\boldsymbol{s}_j$ is not connected to $\boldsymbol{s}_{j-1}$. Therefore, this is an invalid combination. To rectify this, a new graph called *assignment* graph $G = (\mathcal{V}, \mathcal{E})$ is defined. Each node in $\mathcal{V}$ corresponds to an edge in the matching graph, and the edges in $\mathcal{E}$ represent the connections between the corresponding spin systems. The graph is constructed as follows:

1. If $\Pr[r_i \mid \boldsymbol{s}_j] \neq 0$, then $v_{i,j} \in \mathcal{V}$, and variable $v_{i,j}$ is created in the ILP
2. If $v_{i,j}, v_{i+1,k} \in \mathcal{V}$ and $f_{jk} \in \mathcal{F}$ ($\boldsymbol{s}_j$ is connected to $\boldsymbol{s}_k$), then $e_{i,j,k} \in \mathcal{E}$, and variable $e_{i,j,k}$ is created in the ILP
3. All nodes with degree zero are deleted, because IPASS only assigns spin system pairs to residue pairs

Notice that a spin system can potentially be mapped to multiple residues with different probabilities, and multiple copies of the spin systems (which are differentiated according to their mapped location) exist. Fig. 2.4 illustrates the assignment graph for the matching graph in Fig. 2.3. In the depicted assignment graph, the $(\boldsymbol{s}_2, r_i)$ edge in the matching graph has no corresponding node, since it is not connected to any spin systems in the candidate set, neither r_{i-1} nor r_{i+1}.

If each residue has m_i candidate spin systems in the assignment graph, then:

$$|\mathcal{V}| = \sum_{i=1}^{n} m_i \leq m \cdot n, \quad |\mathcal{E}| \leq \sum_{i=2}^{n} m_{i-1} \cdot m_i < m^2 \cdot n, \tag{2.15}$$

where $|\mathcal{E}|$ reaches its maximum in the unlikely case that all the candidate spin systems of r_{i-1} are connected to those of r_{i-1}, for $i = 2, \ldots, n$. IPASS's assignment graph is significantly smaller than similar methods. For example, Crippen et al. define an assignment graph in their method, SAGA, that has the same nodes; however, there is an edge between every two *non-conflicting* nodes (Crippen et al., 2010). In their graph, we have:

$$|\mathcal{V}| = \sum_{i=1}^{n} m_i, \quad |\mathcal{E}| < \frac{1}{2}\left(\sum_{i=1}^{n} m_i\right)^2 \leq \frac{1}{2} m^2 \cdot n^2, \tag{2.16}$$

which indicates the dramatic large size of $|\mathcal{E}|$. Moreover, the bound in (2.16) is much tighter than the one defined in (2.15), mainly because the connectivity graph is sparse.

ILP formulation

The two defined sets of variables in the ILP are:

1. $v_{i,j} \in \{0, 1\}$, it is 1 if and only if $\boldsymbol{s}_j$ is assigned to r_i and 0 otherwise;
2. $e_{i,j,k} \in \{0, 1\}$, it is 1 if and only if $\boldsymbol{s}_j$ is assigned to r_i and $\boldsymbol{s}_k$ is assigned to r_{i+1}, and 0 otherwise

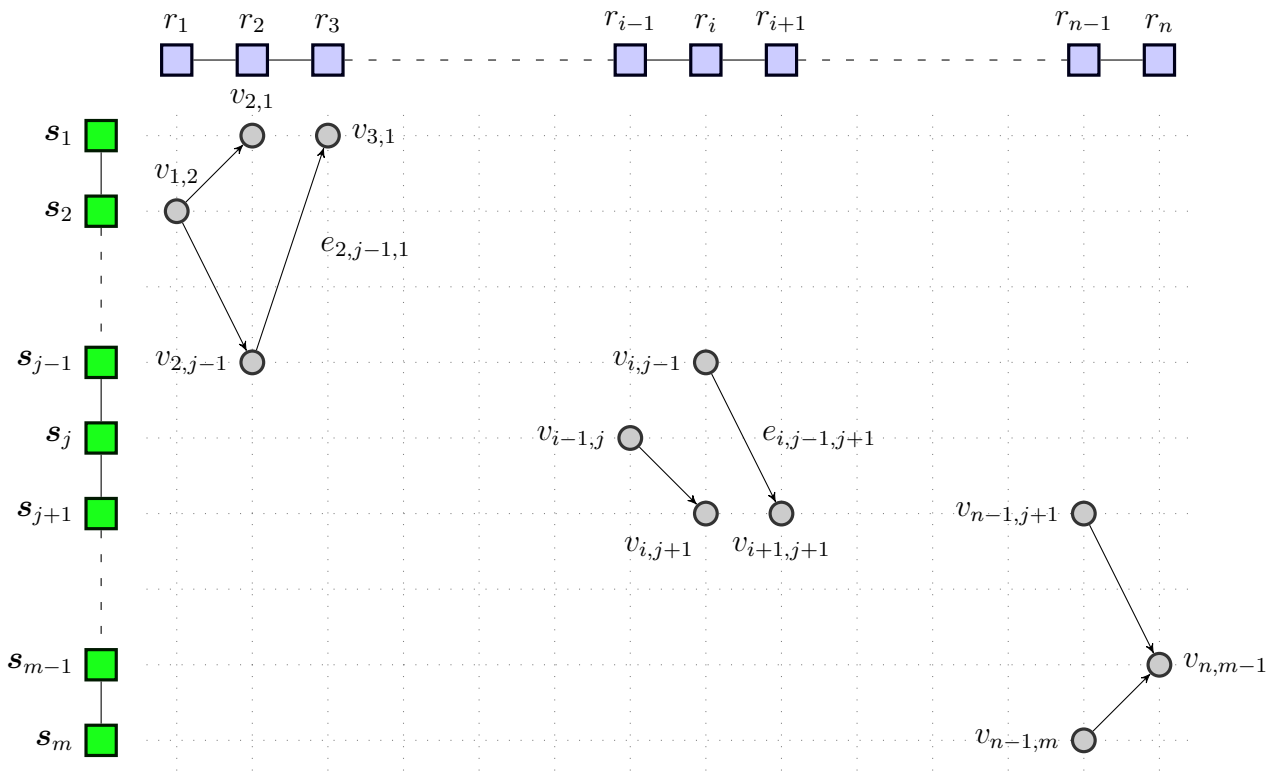


Figure 2.4: Illustration of the assignment graph for the matching graph in Fig. 2.3. There is a node $v_{i,j}$ (shown by the gray circles), corresponding to r_i and $\boldsymbol{s}_j$, only if $\Pr[r_i \mid \boldsymbol{s}_j] \neq 0$. For clarity, only two edges are labeled.

For each edge, a weight corresponding to the probability of mapping two spin systems to two adjacent residues is defined as:

$$\begin{aligned} w_{i,j,k} &= \log\left(\Pr[r_i, r_{i+1} \mid \boldsymbol{s}_j, \boldsymbol{s}_k]\right) \\ &= \log\left(\Pr[r_i \mid \boldsymbol{s}_j]\right) + \log\left(\Pr[r_{i+1} \mid \boldsymbol{s}_k]\right), \end{aligned} \tag{2.17}$$

where $\{r_i \mid \boldsymbol{s}_j\}$ and $\{r_{i+1} \mid \boldsymbol{s}_k\}$ are assumed independent. Now, the task is to find the assignment which maximizes the total weight of the chosen edges. Inherently, each spin system can be assigned to, at most, one residue in the protein sequence. For each residue, there can be, at most, one spin system assigned. After the backbone resonance assignment problem is formulated, the ILP model is written as follows:

$$\max_{e_{i,j,k}} \sum_{e_{i,j,k} \in E} (w_{i,j,k} + \lambda)\, e_{i,j,k}, \tag{2.18}$$

$$\text{subject to } \forall e_{i,j,k} \in E \quad e_{i,j,k} \leq v_{i,j};\ \ e_{i,j,k} \leq v_{i+1,k}, \tag{2.19}$$

$$\forall i \in \{1, \ldots, n\},\ \ \textstyle\sum_{j=1}^{m} v_{i,j} \leq 1, \tag{2.20}$$

$$\forall j \in \{1, \ldots, m\},\ \textstyle\sum_{i=1}^{n} v_{i,j} \leq 1, \tag{2.21}$$

$$\text{and } \ v_{i,j} \in \{0,1\}, \quad e_{i,j,k} \in \{0,1\}. \tag{2.22}$$

Since the logarithm of probabilities is non-positive, the objective function adjusts all the edge weights to positive values by adding the $\lambda = -\min_{i,j,k}(w_{i,j,k})$ term, which is equivalent to scaling the probabilities such that they are all larger or equal to one. Then, the maximization is meaningful. Constraint (2.19) ensures that an edge can be selected, only if both of its ends (the corresponding spin systems) are selected. Constraint (2.20) ensures that a spin system can be assigned to, at most, one residue, and constraint (2.21) ensures that only one spin system is assigned to a residue.

As the result of the fragment fixing step, the size of the problem, i.e., $|\mathcal{V}| + |\mathcal{E}|$ plus the number of constraints, is substantially reduced, which makes the ILP problem tractable. The ILOG CPLEX 9.1 program is used for solving the aforementioned ILP problem. Moreover, the GNU Linear Programming Kit (GLPK) is also capable of solving the formulated ILP. For each enumerated fragment combination, an ILP instance is generated and the solution is attained. The objective function of the assignment represents the score of that configuration. The assignment with the highest score is reported as the final assignment.

NP-hardness

We prove that the backbone resonance assignment problem, under our formulation, is NP-hard. Please note that NP-hardness has also been shown on different formulations of the assignment problem (Eghbalnia et al., 2005; Bailey-Kellogg et al., 2005; Xu et al., 2002).

Theorem 1 *Backbone resonance assignment problem, under the proposed graphical representation is NP-hard.*

Proof 1 *The NP-hardness of the backbone resonance assignment problem is established under the graph representation through a reduction from the* Hamiltonian path *problem which is known to be NP-hard. The Hamiltonian path problem is defined as follows: Given a graph, $H = (\mathcal{W}, \mathcal{F})$ (the connectivity graph), decide whether there exists a path in H that visits each node exactly once. For an instance of the Hamiltonian path problem, a new graph $G(\mathcal{V}, \mathcal{E})$ (assignment graph) is constructed for which $\mathcal{V} = \{1, 2, \ldots, n\} \times \mathcal{W}$, where $n = |\mathcal{W}|$. The new graph, G, has nodes of (i, w), where $w \in \mathcal{W}$ and $1 \leq i \leq n$, and edges between (i, w) and $(i + 1, w')$, if an edge between w and w' exists in H. Here, the edge weights are defined as 1 for all edges in G.*

H has a Hamiltonian path, if and only if there exists a perfect assignment for the backbone resonance assignment problem. For each i, the nodes are connected to their adjacent nodes in the graph with the weight 1. A perfect backbone resonance assignment corresponds to a mapping, where each spin system is used once, and each residue is assigned to a single spin system with a total cost of $n - 1$. Each node (i, w) in G corresponds to a spin system w assigned to residue i. As a result, the perfect assignment that visits each node once corresponds to a Hamiltonian path. Similarly, if there is a Hamiltonian path visiting nodes $w_1, w_2, ..., w_n$, it corresponds to an assignment between w_i and residue i. Consequently, this problem is NP-hard.

2.3 The Proposed Pipeline

The ultimate goal of any automated protein NMR method is to determine the protein structure when the input data is imperfect but sufficient. We propose a protein NMR pipeline which is both fully-automated and robust. The flowchart of the proposed automation pipeline is depicted in Fig. 2.5. The input of the pipeline is the NMR spectra and the protein sequence, and the output is a bundle of structures, all determined with absolutely no human intervention, whatsoever. The proposed pipeline has a fully automated peak-picking module, PICKY (Alipanahi et al., 2009), an error-tolerant robust resonance assignment method, IPASS (Alipanahi et al., 2011a), a structure determination method, FALCON-NMR, built on top of FALCON (Li et al., 2008) and RAPTOR (Xu et al., 2003), and a novel scoring scheme for choosing the best structures.

The proposed pipeline provides an alternative approach to FLYA (López-Méndez and Güntert, 2006). Given the target protein sequence and the backbone resonance assignment done by IPASS, FALCON-NMR first searches for close homologs by RAPTOR. If there are close homologs, they are used to build the initial structural models for the target protein.

If no close homolog is found, the target protein is considered as an *ab initio* target, and FALCON-Abinitio, the *ab initio* module of FALCON, is called to generate the initial structural models for the target. RAPTOR and FALCON-Abinitio generate hundreds of structures (decoys), but they are unable to identify the best decoy. Therefore, we developed a novel NOE contact-based scoring function to select the best decoys. Such decoys are selected and fed into FALCON-Refinement to conduct an all-atom level refinement. The refinement process is iterated by selecting the best decoys by the NOE contact-based scoring function at each iteration, and feeding them back to FALCON-Refinement for further model refining, until convergence, i.e., until the contact score no longer increases.

2.3.1 FALCON-NMR

The FALCON-NMR[3] method uses ideas similar to CS-ROSETTA (Shen et al., 2008) for selecting fragments using chemical shift information; while it differs from CS-Rosetta in that it uses the NOESY contact information (a more recent version of CS-Rosetta, called CS-RDC-NOE-Rosetta (Raman et al., 2010), uses only backbone NOE contacts for sampling more accurate structures). Moreover, FALCON-NMR uses the fragments to train a position specific Hidden-Markov Model (HMM) modeling the backbone torsion angle space. This enables FALCON-NMR to refine the determined structure using the structure of the last iteration.

2.3.2 Contact-based scoring function

A common bottleneck of FALCON-NMR and any other protein structure *prediction* method is that they are not able to select the best decoys from the large number of generated decoys. The most commonly used clustering-based decoy selection methods are usually trapped in popular but low-quality models. We developed an NOE contact-based scoring function to pick the best decoys. The core concept is that a "good" decoy must comply with most distance restraints inferred from NOE peaks.

For the query protein, PICKY is called to pick peaks for both the ^{15}N-NOESY and H(CCO)NH-TOCSY spectra. Proton chemical shifts, determined by H(CCO)NH-TOCSY peaks, are mapped to their corresponding residues, according to the consistency between $[\mathrm{N}, \mathrm{HN}]^\top$ values of H(CCO)NH-TOCSY peaks and those assigned by IPASS. After this process, each residue, r_i, has a corresponding proton chemical shift set, $\mathcal{H}_i$, that in the ideal case contains amide proton, alpha proton, and all side chain proton chemical shifts. Then, for each ^{15}N-NOESY peak, $\boldsymbol{p}_\ell = [\mathrm{N}_i, \mathrm{H}_j, \mathrm{HN}_i]^\top$:

[3]FALCON-NMR is designed and developed by S.C. Li and G. Feng. For more details, please refer to (Alipanahi et al., 2011a).

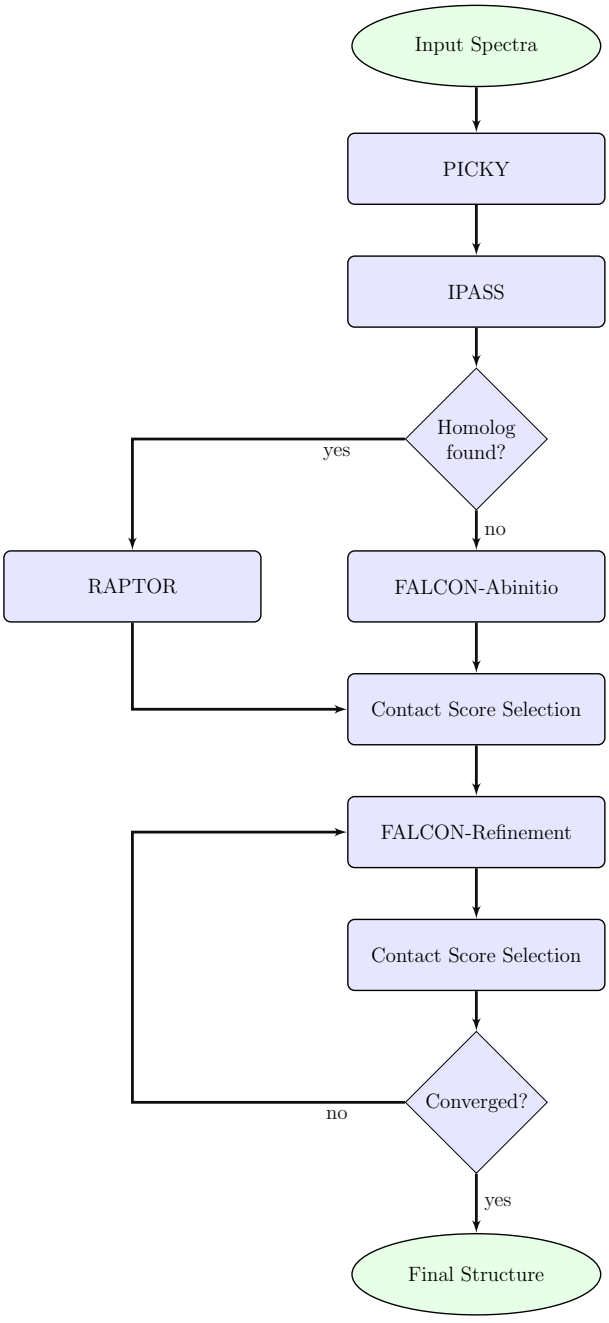


Figure 2.5: Flow chart of the proposed automated protein NMR structure determination pipeline.

1. All the residues with close N and HN chemical shifts to $\boldsymbol{p}_\ell$ are found and added to set $\mathcal{R}_\ell$

2. All residues, $r_{i'}$, such that $\exists$ H $\in \mathcal{H}_{i'}$: $|\mathrm{H} - \mathrm{H}_j| \leq 0.05$ are added to set $\mathcal{T}_\ell$

3. Set of all possible contacts for $\boldsymbol{p}_\ell$ is defined as Cartesian product of $\mathcal{R}_\ell$ and $\mathcal{T}_\ell$:

$$\mathcal{C}_\ell = \mathcal{R}_\ell \times \mathcal{T}_\ell$$

There should be at least one correct contact pair, $(r_i, r_{i'})$, in each $\mathcal{C}_\ell$. For each decoy, first the contact matrix of all residues is computed. Second, for all ^{15}N-NOESY peaks, its set of possible contacts, or $\mathcal{C}_\ell$, is compared against the decoy's contact matrix, and if the decoy satisfies *at least one* of the contacts in the set, its contact score is increased by one. All decoys are ranked according to their contact scores, and the top ranked models are selected as the input for the iterative refinement process. If the top contact score no longer increases two consecutive rounds, the refinement process is considered to be converged and the top-ranked model is output as the final structure.

2.4 Experimental Results

To evaluate the performance of IPASS, two performance measures are used: *precision* and *recall*. Precision measures the ability to reject false assignments, whereas recall measures the ability to discover true assignments. Assume that for the target protein, $\mathrm{N_{max}}$ out of n residues are manually assigned. Even in NMR labs, several residues may not be assigned, i.e, $\mathrm{N_{max}} \leq n$. This is mainly due to fast chemical exchange (conformational changes in some parts of the protein structure) that makes some parts of the protein practically invisible (Hore, 1995). In assessing the performance of assignment methods, we assume that the best any automated assignment method could do is to perform as good as a human NMR expert. Assume that the resonance assignment method assigns $\mathrm{N_o}$ residues, where $\mathrm{T_p}$ of them are assigned correctly. Then, recall and precision are defined as $\mathrm{T_P/N_{max}}$ and $\mathrm{T_P/N_o}$, respectively.

In NMR labs, the spectroscopists usually conduct the whole NMR process all together. Therefore, the final peak lists provided by NMR labs are always "almost perfect," and do not represent the original peaks picked by spectroscopists. Most of the previous assignment studies thus conducted the experiments on either manually or semi-automatically determined peak lists, or on synthetic peak lists. However, to enable automatic NMR protein structure determination, an assignment method must be able to work with imperfect automatically-picked peaks. In fact, the automatically-picked peak lists contain

various sources of errors, such as the chemical shift differences of the same nucleus in different spectra, and a large number of false or missing peaks, which are caused by the peak picking step. Therefore, an assignment method is practical only if it works on "low quality" real noisy input data sets. We evaluate the performance of IPASS on both synthetic (simulated) and real data sets.

2.4.1 Performance on Simulated Data Sets

Although our goal is to develop an error tolerant backbone resonance assignment method which works on realistic data sets, here, a comparison between IPASS and other programs is provided by using prior benchmark data. Such simulated data sets can demonstrate the performance of IPASS on manually or semi-automatically formed spin systems, and manually or semi-automatically picked peaks because manual analysis is usually of high quality.

Simulated spin sytems as input First, IPASS's performance is evaluated on a simulated data set, used by Wan and Lin (2007), which contains 12 proteins. For each protein, the spin systems are simulated, based on the BMRB deposited chemical shift assignments of proteins, and are used as the input for all of the methods. Each spin system contains N_i, HN_i, CA_i, CB_i, CA_{i-1}, and CB_{i-1} chemical shifts. Since RANDOM and CISA are not available, the precision and recall values are adapted from Wan and Lin (2007). The accuracies of RANDOM, MARS, and CISA are calculated according to two different sets of threshold values, to which these programs are sensitive. Note that in these experiments, the spin system forming step is not used. Furthermore, it should be noted that AUTOASSIGN was not available at the time and so could not be included in our experiments.

As shown in Table 2.2, IPASS performs very well and better than any other program regardless of the set of threshold settings. The average precision of IPASS is 99%, and IPASS achieves a 100% precision on seven out of the 12 target proteins. Meanwhile, IPASS can also achieve a high recall value of 96%. It is noteworthy that MARS performs well on this data set as well. However, compared to IPASS, MARS has a relatively low recall value. Table 2.2 demonstrates that RANDOM and CISA are sensitive to the threshold settings. For this simulated data set, in which all chemical shifts and the connectivity graph are perfect, a smaller threshold value can produce more accurate results. In contrast, IPASS does not rely on any parameter settings and its parameters are set without using any special data set.

Simulated peak lists as input The IPASS performance is tested on the same aforementioned data set, but with simulated peak lists. All four steps of IPASS are tested in

Table 2.2: Accuracies (precision/recall) of RANDOM, MARS, CISA, and IPASS for 12 protein data set (simulated spin systems) in percentile.

			$\delta_\alpha = 0.2$ ppm, $\delta_\beta = 0.4$ ppm			$\delta_\alpha = 0.4$ ppm, $\delta_\beta = 0.8$ ppm			
Prot. ID	n	N_{max}	RANDOM	MARS	CISA	RANDOM	MARS	CISA	**IPASS**
bmr4391	66	59	67/63	100/76	97/97	58/55	100/75	91/91	93/90
bmr4752	68	66	40/35	100/97	96/94	36/30	100/97	90/88	100/94
bmr4144	78	68	36/33	100/91	100/99	33/31	100/69	100/99	98/85
bmr4579	86	83	54/51	99/98	98/98	34/32	96/90	80/80	100/98
bmr4316	89	85	42/36	100/100	100/99	35/30	99/91	83/83	99/98
bmr4288	105	94	62/55	100/99	98/98	42/38	98/97	91/91	100/98
bmr4929	114	110	68/63	100/100	93/91	46/43	100/99	96/94	100/100
bmr4302	115	107	66/64	100/100	96/95	47/45	100/100	91/91	100/99
bmr4670	120	102	67/62	100/100	96/95	43/39	100/100	88/87	98/97
bmr4353	126	98	48/43	95/55	96/95	47/43	95/55	90/90	99/93
bmr4027	158	148	43/32	100/99	100/99	40/30	100/99	88/85	100/97
bmr4318	215	191	40/38	99/99	87/84	25/22	100/95	74/70	100/98
Average	112	101	53/48	99/93	96/95	41/37	99/89	88/87	**99/96**

These 12 proteins are selected by Wan and Lin in the CISA paper (Wan and Lin, 2007). The spin systems are simulated based on BMRB deposited chemical shift assignment of these proteins and are used as input for all of these programs.
Since RANDOM and CISA were unavailable at the time of testing, here we use precision and recall values reported in (Wan and Lin, 2007).
The accuracy of RANDOM, MARS, and CISA is calculated based on two sets of thresholds.
The third column indicates number of residues that are manually assigned in the BMRB file.

these experiments. However, Wan and Lin (2007) do not provide such a comparison on RANDOM, MARS, and CISA. Furthermore, RANDOM and CISA are unavailable. As a result, IPASS is compared with the two available programs: MARS and RIBRA. MARS takes only formed spin systems as the input and RIBRA takes the peak lists as the input. RIBRA is used directly, and IPASS's spin system forming method is applied to form spin systems for MARS. Table 2.3 shows that both MARS and IPASS perform well on the simulated peak lists and both are better than RIBRA. The MARS method achieves higher precision and lower recall values than IPASS.

2.4.2 Performance on Real Data Sets

We developed a peak picking system, PICKY (Alipanahi et al., 2009), which specializes on automatic peak picking. PICKY is tested on 32 noisy spectra, provided by our collab-

Table 2.3: Accuracies (precision/recall) of RIBRA, MARS, and IPASS on 12 protein data set (simulated peak lists) in percentile.

Prot. ID	n	N_{max}	spins	Gly/Pro	RIBRA	MARS-I	MARS-II	**IPASS**
bmr4391	66	59	55	6/1	91/76	93/43	94/46	91/85
bmr4752	68	66	65	6/1	91/90	100/94	100/94	100/92
bmr4144	78	68	63	3/5	62/45	100/58	100/41	98/85
bmr4579	86	83	80	5/2	87/67	99/87	99/83	100/94
bmr4316	89	85	80	13/3	99/88	99/83	99/73	88/79
bmr4288	105	94	93	5/10	100/97	99/95	100/97	99/97
bmr4929	114	110	108	10/2	82/78	100/83	99/68	99/98
bmr4302	115	107	107	5/2	100/92	100/96	99/97	96/95
bmr4670	120	102	92	9/5	98/86	99/87	100/87	93/79
bmr4353	126	98	97	8/10	98/93	99/90	100/91	97/90
bmr4027	158	148	146	11/8	90/82	99/94	99/92	97/94
bmr4318	215	191	188	9/12	74/63	99/93	99/86	98/90
Average	112	101	98	5/8	89/80	99/84	99/80	**96/90**

These 12 proteins are selected by CISA paper (Wan and Lin, 2007). The peak lists are simulated based on BMRB deposited chemical shift assignment of these proteins. RIBRA directly accepts peak lists whereas IPASS's spin system forming module is used to generate spin systems for MARS and IPASS.
The third column indicates the number of residues that are manually assigned in the BMRB file.
The fourth column indicates the number of correct spin systems discovered by IPASS's spin system forming module.
RIBRA's performance is measured with ^{15}N and ^{13}C threshold values of 0.5 and ^{1}H threshold value of 0.05 for both grouping and connecting. Those parameters are set according to IPASS for fair comparison.
MARS-I is run with the first set of default parameters: $\delta_\alpha = 0.2$ ppm and $\delta_\beta = 0.4$ ppm.
MARS-II is run with the second set of default parameters: $\delta_\alpha = 0.5$ ppm and $\delta_\beta = 0.5$ ppm, which is the same as IPASS.

orators. The average RCL and PRC values are 88% and 74%, respectively. As a result, the peak lists generated by PICKY are used to evaluate the performance of IPASS on real data sets.

The IPASS method is tested on the peak lists generated automatically by PICKY for four proteins, i.e., TM1112, CASKIN, VRAR, and HACS1. Protein TM1112 from *Thermotoga maritima* is provided by the Arrowsmith Lab at the University of Toronto (Xia et al., 2002) whereas CASKIN, the SH3 domain of the CASKIN neuronal signaling protein, VRAR, *S. aureus* VraR DNA binding domain (Donaldson, 2008), and HACS1, the SH3 domain of the HACS1 human myeloid/hemopoetic signaling protein, are provided by the Donaldson Lab at York University.

Table 2.4 summarizes the performance of RIBRA, MARS, and IPASS on the four real proteins. Since MARS does not contain a spin system forming step, it takes IPASS spin systems as the input. The performance of MARS and IPASS are compared on the same set of spin systems. RIBRA takes the peak lists of ^{15}N-HSQC, CBCA(CO)NH, and HNCACB as the input, so the performance of RIBRA and IPASS are compared on the same peak lists. Table 2.4 shows that IPASS performs significantly better than RIBRA and MARS on all four proteins. When the input peak list is high quality as in TM1112, IPASS can generate assignments that are almost as good as the manual assignment. In Table 2.4, the number of Gly and Pro residues are shown. The Pro residues disrupt the fragments and make the assignment more challenging. The Gly residues are favorable because they can be easily types; however, they also disrupt the fragments, because they do not have any CB chemical shifts.

Table 2.4: Performance of RIBRA, MARS, and IPASS on target proteins TM1112, CASKIN, VRAR, and HACS1.

Protein	n	N_{max}	spin systems	Gly/Pro	CB	RIBRA	MARS-I	MARS-II	IPASS
TM1112	89	83	81/85	4/5	78	40/54	6/45	55/63	71/72
CASKIN	67	54	47/48	7/4	42	12/21	23/25	23/25	29/39
VRAR	72	60	47/47	1/0	41	4/13	6/17	6/17	30/37
HACS1	74	61	48/61	7/5	37	5/11	15/16	15/16	37/50

The first and second column show the target protein name and length, respectively.
The third column shows the number of manually assigned residues by the Arrowsmith and the Donaldson labs, which is considered the upper bound for an automated method.
The Fourth column shows the number of correct/total spin systems discovered.
The Fifth column denotes the number of Pro/Gly in the sequence and the sixth column denotes number of available CB values in the spin systems.
Starting from the seventh column, for each protein, the performance of each method is shown in "number of correctly assigned residues/total number of assigned residues" format.
RIBRA's performance with ^{15}N, ^{13}C threshold values of 0.5, and ^{1}H threshold value of 0.05, for grouping and connecting thresholds. No residue can be assigned if the default values are used. The parameters are set according to IPASS, which makes the comparison fair.
MARS-I: with the first set of default parameters: $\delta_\alpha = 0.5$ ppm and $\delta_\beta = 0.5$ ppm.
MARS-II: with the second set of default parameters: $\delta_\alpha = 0.2$ ppm and $\delta_\beta = 0.4$ ppm.

2.4.3 Determined Structures

To determine the structures of four target proteins, TM1112, CASKIN, VRAR, and HACS1, FALCON-NMR method is used. For the purpose of testing, the initial struc-

tures of TM1112 and VRAR are determined by FALCON-Abinitio, whereas the initial structures of HACS1 and CASKIN are determined by the RAPTOR method, since they have close homologs in the database. For HACS1 and CASKIN, the initial models are built by RAPTOR based on the alignments between the target proteins and their homologs. For each of the two *ab initio* targets, FALCON-Abinitio generates 10,000 initial structural models (decoys) based on a fragment-HMM trained by fragments selected using various features, including chemical shifts information generated by IPASS. The NOE contact-based scoring function is applied to pick the best decoys. The best decoys are then fed into the FALCON-Refinement method to iteratively refine the structures until convergence.

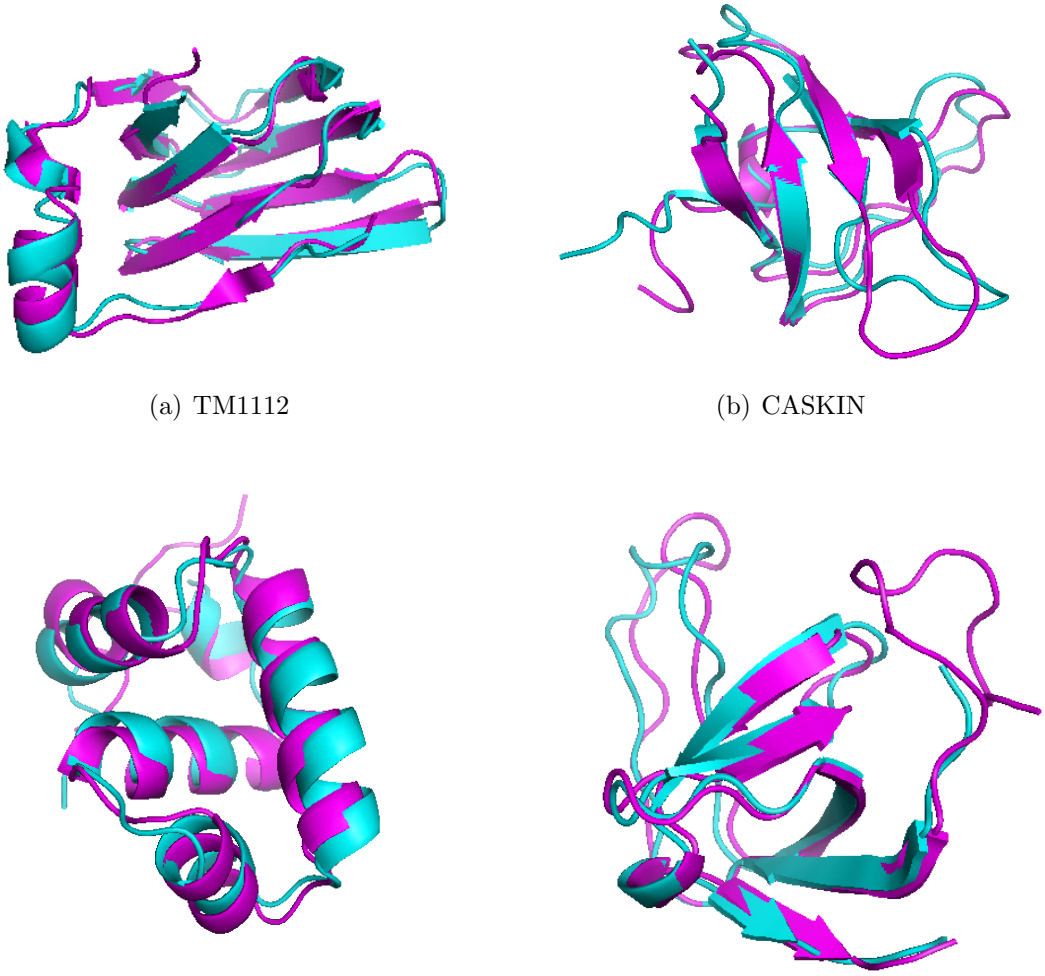
The final structures for TM1112, CASKIN, VRAR, and HACS1 have RMSDs of 1.25 Å, 0.88 Å (core regions), 1.49 Å, and 0.67 Å (core regions) to the native reference structures, respectively. Among them, only TM1112 has a solved X-ray structure, while CASKIN, VRAR, and HACS1 only have NMR structures in PDB. Since the solved NMR structures for CASKIN and HACS1 indicate that both of them have long flexible loops, we only compare our final models with the reference NMR structures on core regions for these two target proteins. The superimposition of the final models and the reference native structures for TM1112, CASKIN, VRAR, and HACS1 are depicted in Fig. 2.6. It can be seen that all the models align well to their reference structures, except in the variable loop regions, which do not have a fixed conformation.

2.5 Discussion

We now turn to the question whether even without NMR data, i.e., chemical shifts information and NOE contacts, FALCON or ROSETTA themselves could generate high-resolution structures for the target proteins. To answer this question, three more experiments were performed:

1. To verify how much the automatically assigned IPASS chemical shifts were inferior to the manually assigned chemical shifts, we ran FALCON-NMR with the manually or semi-automatically solved chemical shift assignments from the Arrowsmith Lab at the University of Toronto and the Donaldson Lab at York University. The NOE contact-based scoring function is used again for picking the best decoys.

2. To evaluate the role of the chemical shifts and the NOE contacts information, we ran FALCON without any chemical shifts information, and picked the top decoy by the default clustering-based decoy selection method of FALCON.



(a) TM1112 (b) CASKIN

(c) VRAR (d) HACS1

Figure 2.6: The superimposition between the final structures of FALCON-NMR (shown in cyan) and the experimentally determined structures (shown in magenta) of TM1112, CASKIN, VRAR, and HACS1, respectively. TM1112 has a crystal structure while CASKIN, VRAR, and HACS1 have NMR structures. The backbone RMSDs are 1.25 Å, 0.88 Å (core regions), 1.49 Å, and 0.67 Å (core regions), respectively.

3. Moreover, we ran ROSETTA 2.3.0 (Bradley et al., 2005) without any chemical shifts information, and picked the top decoy selection by the default clustering-based decoy selection method of ROSETTA.

Experimental results indicate that running FALCON with perfect assignments generates slightly better models for the target proteins, whereas simply running FALCON or ROSETTA with their default clustering-based decoy selection methods results in low-resolution models, especially for the *ab initio* targets. For example, the RMSDs for TM1112 for the three experiments are 1.17 Å, 11.84 Å, and 12.13 Å, respectively. Similar results are obtained on other targets as well.

These findings imply that: (*i*) replacing chemical shifts automatically assigned by IPASS does not affect the accuracy of the final structure noticeably and (*ii*) without the chemical shifts and NOE contacts information, neither FALCON nor ROSETTA are able to generate high quality structures.

Implemented in C++, IPASS takes less than five minutes to achieve its results for a practical noisy data set of a medium size protein (70-150 residues in length). In addition, it takes IPASS five seconds for a simulated data set with up to 215 residues. The difference in speed stems from the fact that for the simulated data sets most of the fragments are fixed and consequently the integer linear programming problem size is very small. The entire process, from peak picking by PICKY to the structure calculation by FALCON-NMR, takes one day on a cluster of 100 CPUs.

The combination of the novel spin system forming, the improved spin system typing, and the integer linear programming, results in a robust and error-tolerant protein NMR structure determination system. The ILP problems were solved efficiently, because the fragment fixing and spin system typing steps had reduced the original problem size significantly, which enables CPLEX to find the globally optimal solution.

Chapter 3

From Distance Constraints to Structure by SDP

Computing three-dimensional protein structures from their amino acid sequences has been one of the most widely studied problems in bioinformatics. Any protein's structure is a key to understanding its physical, chemical, and biological properties. Myoglobin was the first protein whose structure was solved in 1958 by the X-ray Crystallography method (Kendrew, 1958). Until 1984, X-ray method was the only experimental method for protein structure determination. The NMR method came into use 27 years later: the first protein structures determined by NMR were proteinase inhibitor IIA from bull seminal plasma (Williamson et al., 1985) and the Lac repressor headpiece (Kaptein et al., 1985). The introduction of protein NMR was a breakthrough, enabling the identification of the protein structure in the aqueous solutions, which is closer to the native states of the proteins.

We briefly review the basics of protein structures and the principles of Euclidean distance geometry and then describe the protein NMR structure determination problem, reviewing the existing methods and the history behind them. Finally, our proposed method is presented and the experimental results are demonstrated.

3.1 Basics

We review all the material necessary for describing the protein NMR problem, starting with the notation and the terminology used throughout this chapter.

3.1.1 Notation

Scalars, vectors, sets, and matrices are shown in lower case, lower case bold italic, script, and upper case italic letters, respectively. We define the following notations:

$\mathbb{R}^p$	space of real p-dimensional vectors
$\mathbb{R}^{p\times q}$	space of real $p \times q$ matrices
$\mathcal{S}^p$	the space of symmetric $p \times p$ matrices
$\mathcal{S}^p_+$	the set of symmetric positive semidefinite $p \times p$ matrices
$\mathcal{S}^p_{++}$	the set of symmetric positive definite $p \times p$ matrices
I_p	$p \times p$ identity matrix
$\mathbf{1}_p$	all-ones vector of size p
$\boldsymbol{e}_i$	the i-th column of identity matrix
$\lvert\mathcal{B}\rvert$	cardinality of set $\mathcal{B}$

For a matrix $A \in \mathbb{R}^{p\times q}$

A_{ij}	(i, j)-th entry of A
$A^\top$	transpose of A
$\mathbf{rank}(A)$	rank of A
$\mathbf{range}(A)$	range space of A
$\mathbf{null}(A)$	null space of A

For a square matrix $A \in \mathbb{R}^{p\times p}$

$\mathbf{trace}(A)$	sum of its diagonal elements
$\mathbf{diag}(A)$	a vector containing its diagonal elements

3.1.2 Protein Structure Representation

We assume the target protein has l amino acid residues $r_1, \ldots, r_l$. Moreover, we assume it has n atoms, $\{a_1, \ldots, a_n\}$, each with its van der Waals radius v_i, the so called "ball and stick" model. If a nucleus a_i is NMR active and has a corresponding assigned chemical shift, its chemical shift is denoted by ω_i. There are two major models for representing a protein molecule:

Cartesian Coordinate Space

The easiest method to represent the structure of a protein molecule is using the Cartesian coordinate space, in which the protein structure is represented by $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i \in \mathbb{R}^3$. This is the method used in the Protein Data Bank (PDB) (Bernstein et al., 1977). All information about the query protein, bond length, bond angle, torsion angles, among others, can be extracted from these coordinates.

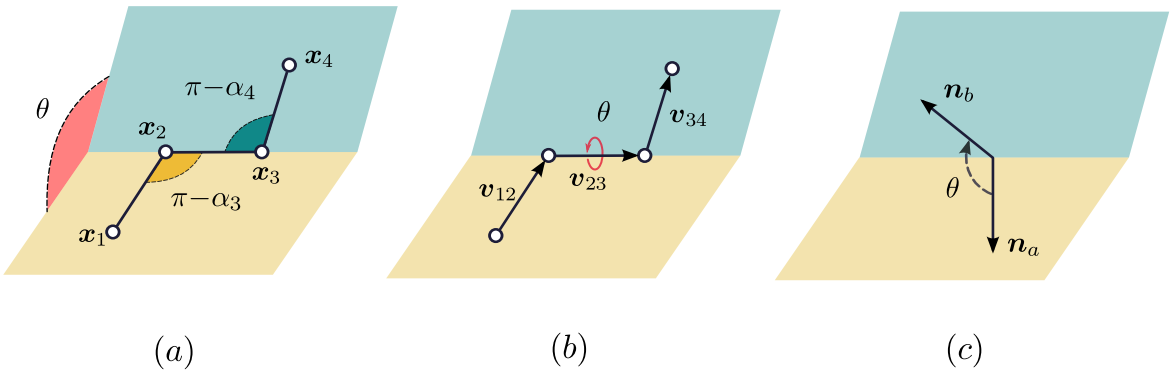


Figure 3.1: Schematic diagram of a dihedral angle.

Torsion Angle Space

If the minuscule fluctuations in the bond lengths and the bond angles of the protein molecule in the room temperature are ignored, a more efficient representation of the structure is possible in the *torsion angle space*, in which the degrees of freedom are the *rotations* about single covalent bonds. Every torsion angle is a *dihedral* angle that involves four points (atoms) in $\mathbb{R}^3$. For example, in Fig. 3.1, the dihedral angle, θ, is defined as the rotation angle about the $\boldsymbol{x}_2 - \boldsymbol{x}_3$ bond. The dihedral angle is defined as the angle between the plane passing through $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$ and the plane passing through $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$.

We define the normal vectors of the two involved planes as:

$$\boldsymbol{n}_a = (\boldsymbol{v}_{12} \times \boldsymbol{v}_{23})/(\|\boldsymbol{v}_{12} \times \boldsymbol{v}_{23}\|)$$
$$\boldsymbol{n}_b = (\boldsymbol{v}_{23} \times \boldsymbol{v}_{34})/(\|\boldsymbol{v}_{23} \times \boldsymbol{v}_{34}\|)$$

where $\boldsymbol{v}_{ii+1} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i$, $i = 1, 2, 3$ is the vector connecting $\boldsymbol{x}_i$ to $\boldsymbol{x}_{i+1}$ (see Fig. 3.1.b). The value and the sign of the dihedral angle is determined by the following relation:

$$\theta = \text{atan2}(\|\boldsymbol{v}_{23}\|\boldsymbol{v}_{12}^\top \boldsymbol{n}_\text{b},\ \|\boldsymbol{v}_{12} \times \boldsymbol{v}_{23}\|\boldsymbol{n}_\text{a}^\top \boldsymbol{n}_\text{b}),$$

which can be further simplified to

$$\theta = \text{atan2}(\boldsymbol{u}_{12}^\top \boldsymbol{n}_\text{b},\ \sin(\alpha_3)\boldsymbol{n}_\text{a}^\top \boldsymbol{n}_\text{b}),$$

where $\boldsymbol{u}_{ii+1} = \boldsymbol{v}_{ii+1}/\|\boldsymbol{v}_{ii+1}\|$ (normalized $\boldsymbol{v}_{ii+1}$) and α_3 is the angle between $\boldsymbol{v}_{12}$ and $\boldsymbol{v}_{23}$ (see

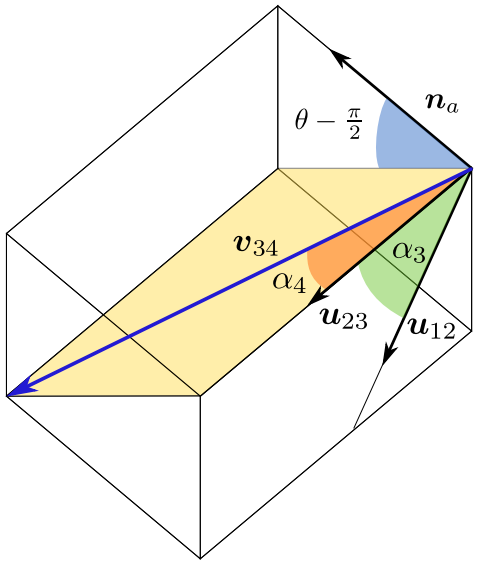


Figure 3.2: The method of finding the fourth point in a dihedral angle.

Fig 3.1.a). In the dihedral angle depicted in Fig. 3.1.c, $\theta \approx +135$ (figure is not to scale); the dihedral angle is positive, because looking in the direction of $\boldsymbol{v}_{23}$, $\boldsymbol{n}_a$ must to be rotated counterclockwise to align with $\boldsymbol{n}_b$.

Any four points in $\mathbb{R}^3$ form a dihedral angle; given the first three points, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$, the length of $\boldsymbol{v}_{34}$, and the angle between $\boldsymbol{v}_{23}$ and $\boldsymbol{v}_{34}$ (α_4 in Fig. 3.1.a) we can uniquely determine the fourth point, $\boldsymbol{x}_4$.

If we find $\boldsymbol{v}_{34}$, then we find the fourth point as $\boldsymbol{x}_4 = \boldsymbol{x}_3 + \boldsymbol{v}_{34}$. We know that $\boldsymbol{u}_{12}$, $\boldsymbol{u}_{23}$, and $\boldsymbol{n}_a$ span $\mathbb{R}^3$, as depicted in Fig. 3.2. Therefore, we can decompose $\boldsymbol{v}_{34}$ into its components as

$$\boldsymbol{v}_{34} = c_{\boldsymbol{n}_a}\boldsymbol{n}_a + c_{\boldsymbol{u}_{12}}\boldsymbol{u}_{12} + c_{\boldsymbol{u}_{23}}\boldsymbol{u}_{23},$$

where the components could be found by considering Fig. 3.2:

$$
\begin{aligned}
c_{\boldsymbol{n}_a} &= +\|\boldsymbol{v}_{34}\| \sin(\alpha_4) \sin(\theta), \\
c_{\boldsymbol{u}_{12}} &= -\|\boldsymbol{v}_{34}\| \sin(\alpha_4) \frac{\cos(\theta)}{\sin(\alpha_3)}, \\
c_{\boldsymbol{u}_{23}} &= +\|\boldsymbol{v}_{34}\| \cos(\alpha_4) + \|\boldsymbol{v}_{34}\| \frac{\sin(\theta)}{\tan(\alpha_3)}.
\end{aligned}
$$

Considering the aforementioned method, given $\{\boldsymbol{x}_i\}$, $\boldsymbol{x}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$, if we intend to add $\boldsymbol{x}_{n+1}$, we could store its Cartesian coordinates (three values), or we could compute the dihedral angle, θ_{n+1}, of $(\boldsymbol{x}_{n-2}, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n, \boldsymbol{x}_{n+1})$ and store the following three values:

1. the dihedral angle, θ_{n+1},
2. the length of the vector connecting the last two points, $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\|$, and
3. the angle between $\boldsymbol{x}_{n+1} - \boldsymbol{x}_n$ and $\boldsymbol{x}_n - \boldsymbol{x}_{n-1}$, α_{n+1}.

For an arbitrary set of points, this results in no reduction in the problem size; however, in proteins bond lengths, $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\|$, and bond angles, α_{n+1}, are already known. Consequently, the torsion angle space is more appropriate for modeling proteins.

Proteins have a unique property which makes the torsion angle space even more appealing: they consist of rigid subunits. That is, there are cliques of atoms where all the distances within each clique is fixed. Covalent bonds are very stable and for our proposes can be assumed to be rigid. For example, a tetrahedral carbon bonds to four other atoms and the group of five atoms (including the carbon itself in the center) form a three-dimensional rigid clique of size five.

Protein Structure Model

The backbone of polypeptide chains do not conform arbitrarily; it is observed that the backbone is made of *peptide planes*, as illustrated in Fig. 3.3. There are six atoms from two consecutive residues in each plane, three from the preceding and three from the proceeding residue. For example, the lower peptide plane in Fig. 3.3 contains CA_{i-1}, C_{i-1}, O_{i-1}, N_i, HN_i, and CA_i. The two connected planes cannot orient arbitrarily with respect to each other, since the distance between N_i and C_i is fixed (they are part of CA tetrahedral carbon). The two degrees of freedom are:

1. The dihedral angle Φ about the N–CA bond, consisting of (C_{i-1},N_i,CA_i,C_i)

2. The dihedral angle Ψ about the CA–C bond, consisting of (N_i,CA_i,C_i,N_{i+1})

There is also a third torsion angle ω defined within each peptide plane about the C–N bond, which consists of (CA_{i-1},C_{i-1},N_i,CA_i). In most cases, ω is very close to 180° (*trans*) and in rare cases it is close to 0° (*cis*).

Dihedral angles are defined for the amino acid side chains as well, denoted χ_ℓ, where ℓ can be as large as five, e.g, for Lysine. Since each amino acid has a different side chain, their number of side chain dihedral angles is different (see Table 3.3). Last, a vector containing all of the torsion angles, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_t]^\top$, can be used to uniquely determine the protein structure (up to a rotation and translation).

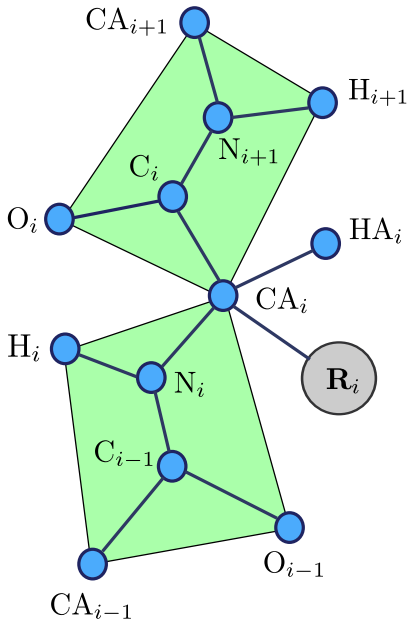


Figure 3.3: Peptide planes involving residues $i-1$, i, and $i+1$ are shown.

Ramachandran Plot Proteins are made up of atoms; atoms not covalently bonded cannot arbitrarily penetrate each other. That is, for two atoms a_i and a_j with coordinates $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and van der Waals radii v_i and v_j, respectively, the simplified *steric* constraint for the two atoms is defined as:

$$d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \geq (v_i + v_j),$$

where d_{ij} denotes the Euclidean distance between the two corresponding atoms. A large amount of energy is required to make d_{ij} smaller than $v_i + v_j$. According to the Lennard-Jones potential, the energy is proportional to $1/d_{ij}^{12}$ when the two atoms are very close. Strictly enforcing all of the steric constraints, Ramachandran et al. generated a plot which has been useful in studying protein structures, called the *Ramachandran plot* (Ramachandran et al., 1963). A Ramachandran plot is depicted in Fig. 3.4; it is observed that $\Phi > 0$ half-plane is sparsely populated. Moreover, most of the (Φ, Ψ) combinations are forbidden; they rarely occur and sometimes mistakenly occur in the structure calculation process. In fact, the compliance of a new structure's (Φ, Ψ) values to the Ramachandran plot is a popular method for verifying the correctness of the new structures (Chen et al., 2010).

With the deposition of thousands of new high resolution proteins in the PDB, more recent versions of the Ramachandran plot have been generated; for example, Lovell et al. provide a more accurate plot (Lovell et al., 2003).

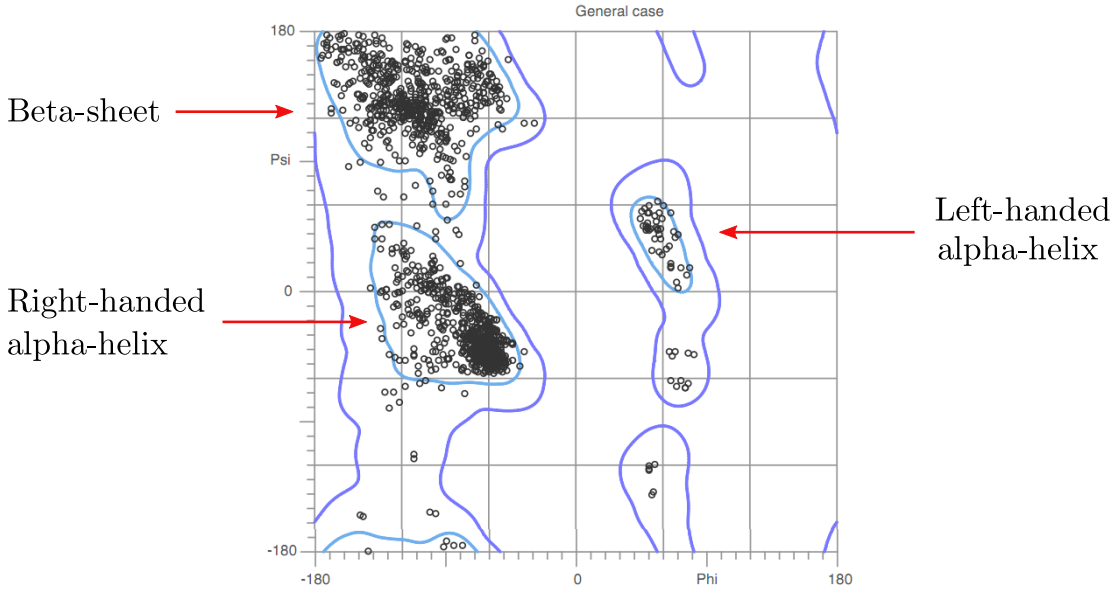


Figure 3.4: The regions enclosed by light blue lines indicate "allowed" combinations of (Φ, Ψ) that result in no steric clashes. The regions enclosed by dark blue lines indicate combinations of (Φ, Ψ) that are allowed if the steric constraints are slightly relaxed. The (Φ, Ψ) combinations for the the structure of Cellobiose phosphorylase from Clostridium thermocellum in complex with phosphate (PDB ID: `3QDE`) for non-glycine and non-proline residues are shown (plot is generated by MolProbity (Chen et al., 2010)).

Amino Acids

Amino acids differ only in their side chain groups (shown by R in Fig. 3.3). The side chains have different tendencies in interacting with water molecules and other side chains (Petsko and Ringe, 2003). Amino acids' characteristics are dominated by their side chains. For example, some amino acid side chains are *hydrophobic* and only engage in van der Waals interactions, mainly with other hydrophobic residues and are usually buried in the core of the protein structure.

In contrast, amino acids with *hydrophilic* or water-loving side chains are able to make hydrogen bonds to the other hydrophilic residues, to the surrounding water molecules, to peptide backbone, and to polar organic molecules (Petsko and Ringe, 2003). They are more abundant on the surface of the protein that is exposed to the outside environment, and that usually contains water molecules. Last, *amphipathic* side chains exhibit both polar and non-polar features, making them ideal interfaces that can connect to both of the aforementioned types of residues. More detailed information about different amino acids is given by Branden and Tooze (1999), Betts and Russell (2003), and Creighton (1992).

We use the IUPAC-IUBMB-IUPAB atom nomenclature recommendations for representation of protein structures (Markley et al., 1998).

3.1.3 Protein NMR Structural Information

The ultimate goal in *de novo* protein NMR structure determination is to compute the protein structure from the set of experimentally inferred information. We review these sources of structural information here.

NOE-derived upper bounds

The main source of structural information is in the form of NOE cross peaks. We delve deeper into the information extracted from the NOESY experiments.

NOE cross peak assignment Assume we are given a 2D $[^1\mathrm{H}, ^1\mathrm{H}]$-NOESY peak, $\boldsymbol{p}_\ell = [\omega_\ell^{(1)}, \omega_\ell^{(2)}]^\top$. For a preset *matching* tolerance of $\Delta\omega$, the assignment process is as follows:

1. Find all protons such that $|\omega_i - \omega_\ell^{(1)}| \leq \Delta\omega$, put these protons in the set $\mathcal{P}_\ell$;
2. Find all protons such that $|\omega_j - \omega_\ell^{(2)}| \leq \Delta\omega$, put these protons in the set $\mathcal{Q}_\ell$;

3. The set of all possible assignments for $\boldsymbol{p}_\ell$ is $\mathcal{C}_\ell = \mathcal{P}_\ell \times \mathcal{Q}_\ell$ (the Cartesian product of the two sets), where $\mathcal{C}_\ell = \{(\mathrm{H}_i, \mathrm{H}_j)\}$ contains the pairs of supposedly spatially proximate protons. If $|\mathcal{C}_\ell| > 1$, then the peak assignment is called *ambiguous*. That is, there is more than one assignment for the corresponding peak.

The NOE peak assignment process for 3D NOESY experiments is essentially the same as 2D NOE assignment and a brief description is given in the **Contact-based scoring function** section (see page 39).

NOE-inferred upper bounds After NOE peaks are assigned, the peak volumes are used to estimate the actual distances between involved protons. There are several packages that generate upper bounds from chemical shift assignments and NOE peaks, including CALIBA (Güntert et al., 1991), NOAH (Mumenthaler et al., 1997), CANDID (Herrmann et al., 2002), ARIA (Nilges et al., 1997; Linge et al., 2003) among others. Note that all of these methods form upper bounds of the form:

$$d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq u_{ij}, \quad (i,j) \in \mathcal{U}.$$

There are two important problems with these upper bounds:

1. *Fine noise:* the u_{ij} values are approximate and noisy. If atoms a_i and a_j are close to one another in the protein structure, their actual distance may be higher than u_{ij}. If the upper bounds cannot be slightly violated, then a feasible solution may not exist.

2. *Coarse noise:* due the ambiguity of NOE peak assignment, it is likely that the assignment is incorrect and $d_{ij} \gg u_{ij}$. If the upper bound violations are severely penalized, they can lock the structure and prevent correct folding.

In several recent methods, especially those simulating the NMR data rather than using real NMR data, the coarse noise is discarded altogether, oversimplifying the problem.

Uncertainties in the NOE peak assignment There are three major challenges with NOE peaks: (*i*) most NOE peaks cannot be uniquely assigned, (*ii*) many NOE peaks remain undetected, due to fast relaxation (spin-spin relaxation) and/or conformational exchange; these peaks are absent in the structure calculation, and (*iii*) several NOE peaks are spurious and noisy, adding additional (possibly) incorrect contacts (Güntert, 2004).

NOE peaks may not always be assigned uniquely; in fact, most of the peak assignments are ambiguous. Mumenthaler et al. proposed a simple but elegant model for estimating the percentage of ambiguously assigned NOE peaks as follows (Mumenthaler et al., 1997):

Assume we have a simple 2D $[^1\mathrm{H}, ^1\mathrm{H}]$-NOESY spectrum with a total n_p peaks, n_h hydrogen atoms in the protein, a tolerance of $\Delta\omega$ for matching chemical shifts, and with the chemical shift range of $\Delta\Omega$ for protons. If we assume that the chemical shifts are distributed uniformly in a range of width $\Delta\Omega$, the probability of finding a proton with chemical shift $\omega \in [\omega_\ell - \Delta\omega, \omega_\ell + \Delta\omega]$ around any shift ω_ℓ is

$$p = \frac{2\Delta\omega}{\Delta\Omega}.$$

For a unique assignment of a 2D $[^1\mathrm{H}, ^1\mathrm{H}]$-NOESY peak, $\boldsymbol{p}_\ell = [\omega_\ell^{(1)}, \omega_\ell^{(2)}]^\top$, there should be only one matching proton in each dimension. That is, there must be only one protein in the $[\omega_\ell^{(i)} - \Delta\omega, \omega_\ell^{(i)} + \Delta\omega]$, $i = 1, 2$, intervals. The number of such peak assignments is given by:

$$N^{(1)} = n_p \underbrace{(1-p)^{n_h-1}}_{\omega^{(1)}\text{ dimension}} \times \underbrace{(1-p)^{n_h-1}}_{\omega^{(2)}\text{ dimension}} = n_p(1-p)^{2n_h-2} \approx n_p \exp(-4n_h\Delta\omega/\Delta\Omega).$$

Using the typical values of $\Delta\Omega = 10$ and $\Delta\omega = 0.025$ and considering that n_h is in the range of a few hundreds for medium size proteins, the percentage of uniquely assigned protons is negligible. Note that, this is a lower bound because the proton chemical shifts are not truly uniformly distributed and their distribution is closer to a mixture of Gaussians.

In modern NMR spectroscopy, usually NOE peaks from ^{13}C-edited or ^{15}N-edited 3D $[^1\mathrm{H}, ^1\mathrm{H}]$-NOESY experiments are used. In 3D peaks, we assume that the ambiguity in one dimension is resolved by making use of the corresponding heterospin; therefore, $N^{(1)} \approx n_p \exp(-2n_h\Delta\omega/\Delta\Omega)$. That is, the effect is similar to running a 2D NOESY experiment on a protein half the size of the original protein. However, 3D experiments have lower resolutions and chemical shifts are more cluttered in the center of $\Delta\Omega$ range, which makes NOE peak assignment challenging.

NOE contact refinement The total number of NOE contacts, $n_c = \sum_\ell |\mathcal{C}_\ell|$ could be very large, while we know that at most n_p (number of NOE peaks) of them are correct (assuming all NOE peaks are not spurious). Several methods have been proposed for refining the contact sets and, thus, reducing n_c, making it as close to n_p as possible.

One fairly successful heuristic approach is implemented in the CANDID algorithm, which combines features from methods ARIA and NOAH, while introducing two important new features: *network anchoring* and *constraint combination* (Herrmann et al., 2002; Güntert, 2004). Network anchoring finds the set of most consistent NOE contacts. For

example, if atom a_i is in contact (close in space) with atoms a_j and a_k, there is a high probability that a_j and a_k are also in contact. Any NOE assignment must be corroborated by some other contacts in the vicinity of the atoms involved. Network anchoring can greatly reduce the number of spurious NOE contacts, however, there will always be ambiguous constraints left.

Another important concept is constraint combination, where for example two constraints (A–B and C–D) are combined into one (A–B $\cup$ C–D). If any of the constraints is satisfied, both are considered satisfied. More complex schemes are also available in CANDID (Herrmann et al., 2002). It must be noted that not all optimization methods can handle unions of constraints, which limits their usage.

Expected number of NOE peaks Assume the protein is a 3D sphere of radius R, and the n_h hydrogens are uniformly distributed in this sphere. There will be an NOE cross peak corresponding to proximate protons if their distance is less than $d_{\max}$. Therefore, the probability of two protons being closer than $d_{\max}$ or the probability of being in contact (spatially proximate) is $p_c = (d_{\max}/R)^3$. Assuming that contacts between protons are independent from each other[1], the expected number of contacts is:

$$n_c \approx \binom{n_h}{2} p_c = \frac{n_h(n_h-1)}{2} \left(\frac{d_{\max}}{R}\right)^3 .$$

Proteins are tightly packed, so we can say that the protein volume, V_p, is linearly proportional to the number of atoms, n, i.e., $V_p \propto n$. Moreover, we know $R \propto V_p^{(1/3)}$ that indicates $p_c \propto 1/n$. Since $n_h \propto n$, we conclude that $n_c \in \mathcal{O}(n)$. In practice, the number of NOE-inferred upper bounds is in the range of tens of hundreds.

Hydrogen Bonds Hydrogen bonds can be identified by looking for slow hydrogen exchange in amides (Wagner and Wüthrich, 1982). Hydrogen bonds can also be identified by carefully analyzing the NOE restraints; they may seem redundant; however, they can enforce more regular secondary structure (Güntert, 1998). Each hydrogen bond introduces two distance constraints: (i) the acceptor-hydrogen distance is bounded to 2.0 Å and (ii) the distance between the acceptor and the atom covalently bonded to the hydrogen is bounded to a maximum of 3.0 Å, as depicted in Fig. 3.5.

Disulfide and salt bridges Disulfide bridges are made after the oxidation of sulfhydryl groups on the side chains of two cysteine residues. That is, they are covalent bonds between

[1]In reality, contacts are not independent but in our approximate analysis, this model is sufficiently accurate.

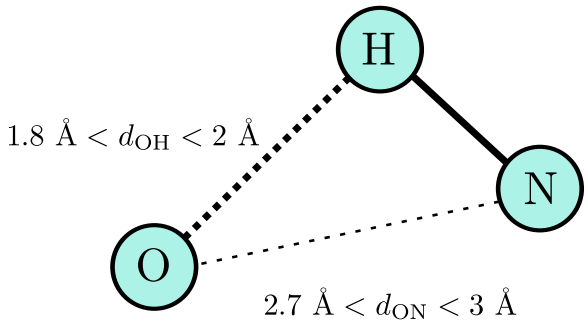


Figure 3.5: Distance restraints enforcing a hydrogen bond (picture reproduced from (Güntert, 1998)).

side chains of two cysteine residues, where two sulfide atoms are almost 2.2 Å apart (Petsko and Ringe, 2003). Salt bridges are similar to hydrogen bonds, however both the donor and acceptor are fully charged, which makes them more stable than hydrogen bonds. The distance constraints are similar to hydrogen bonds.

Torsion angle restraints

Chemical shifts depend on the local electron density and the local conformation of the molecule. There are some methods such as TALOS (Cornilescu et al., 1999) and PREDITOR (Berjanskii et al., 2006) that estimate the backbone and side chain torsion angles by searching for peptide fragments having similar sequence and chemical shifts in a database of fragments. TALOS reports a root-mean-square difference of 15° between the predicted and actual torsion angles, for 65% of residues with "good" predictions. An improved version, called TALOS+ (Shen et al., 2009b), can make "good" predictions for 88.5% of residues with a a root-mean-square difference of 13°. The PREDITOR method reports that 88% of its backbone torsion angles predictions are within 30° of the actual values.

Torsion angle restraints are in the form of $\theta^{\min} \leq \theta \leq \theta^{\max}$ for good predictions and a set of intervals for ambiguous predictions. It should be noted that although these predictions are close to reality most of the time, they are not as *dependable* as are NOE upper bounds. Consequently, they must be enforced in a flexible manner, i.e., the structure must be allowed to (slightly) violate them.

3.1.4 Protein NMR Structure Calculation Formulation

Assuming the Cartesian coordinates of the protein structure as $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i \in \mathbb{R}^3$, vector of torsion angles as $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_t]^\top$, $-\pi \leq \theta_i \leq \pi$, set of upper bounds, $d_{ij} \leq u_{ij}$, $(i,j) \in \mathcal{U}$, set of torsion angle restraints, $\boldsymbol{\theta}^{\max} = [\theta_1^{\max}, \ldots, \theta_t^{\max}]^\top$, $\boldsymbol{\theta}^{\min} = [\theta_1^{\min}, \ldots, \theta_t^{\min}]^\top$ such that $\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}$, $i \in \mathcal{T}$, we formulate the protein NMR structure calculation problem as:

$$\text{minimize}_X \quad \phi_U(X,U) + \phi_T(\boldsymbol{\theta}, \boldsymbol{\theta}^{\min}, \boldsymbol{\theta}^{\max}) + \phi_S(X), \tag{3.1}$$

where ϕ_U, ϕ_T, and ϕ_S penalize upper bounds, torsion angles, and stereochemical restraint violations, respectively. We delve more deeply into these penalty functions.

Upper Bound Violations

Upper bounds are the most important structural information that NMR experiments provide. These bounds include bounds derived from NOE peaks, hydrogen bonds, and disulfide, and salt bridges. Upper bound violations should be penalized and this is done via $\phi_U(X,U)$ that can be written as:

$$\phi_U(X,U) = \sum_{(i,j)\in\mathcal{U}} w_{ij}^{(U)} \; f\left(d_{ij}, u_{ij}\right),$$

where $w_{ij}^{(U)}$ is the weight corresponding to the bound between a_i and a_j. Weights are added, because we do not need to treat all bounds equally. For example, violations of more confident bounds are more greatly penalized. If upper bounds are treated as constraints, then we have:

$$f\left(d,u\right) = \begin{cases} \infty, & \text{if} \quad d > u \\ 0, & \text{if} \quad d \leq u \end{cases},$$

which tolerates no violation. Due to inherent noise in the protein NMR, the constrained model is highly prone to failure.

The simplest form of the penalizing function that allows violation of upper bounds can be written as:

$$f\left(d,u\right) = \begin{cases} (d-u)^2, & \text{if} \quad d > u \\ 0, & \text{if} \quad d \leq u \end{cases},$$

which may not be a proper choice, considering that some of the upper bounds are incorrect. The squared-error term penalizes erroneous bounds, preventing the formation of the cor-

rect structure. However, it is differentiable and suitable for gradient-based minimization techniques.

A more suitable approach is to penalize the absolute value of the violations:

$$f\left(d,u\right)=\begin{cases} d-u, & \text{if} \quad d>u \\ 0, & \text{if} \quad d\leq u \end{cases},$$

which is more lenient on incorrect bounds. Another compact form of this function is $f(d,u)=\max(d-u,0)$. Some alternative penalizing functions are described by Güntert (1998).

Torsion Angle Violations

The term penalizing the torsion angle violations can be written as:

$$\phi_T(\boldsymbol{\theta},\boldsymbol{\theta}^{\min},\boldsymbol{\theta}^{\max})=\sum_{i\in\mathcal{T}} w_i^{(T)}\; g\left(\theta_i,\theta_i^{\min},\theta_i^{\max}\right),$$

which, similar to the upper bounds, if $\theta_i^{\min}\leq\theta_i\leq\theta_i^{\max}$, there is no penalty. The torsion angle deviation, Δ, is defined as:

$$\Delta=\begin{cases} \theta-\theta^{\max}, & \text{if} \quad \theta^{\max}\leq\theta\leq\pi \\ \theta^{\min}-\theta, & \text{if} \quad -\pi\leq\theta\leq\theta^{\min} \\ 0, & \text{if} \quad \theta^{\min}\leq\theta\leq\theta^{\max} \end{cases} \tag{3.2}$$

Using the definition of Δ, the simplest way to penalize deviations is to minimize $\sum_i w_i^{(T)}\Delta_i^2$. Some methods use more complex penalty functions, for example, see (Güntert, 2004).

Stereochemical restraints

Proteins exhibit certain unique structural features called *stereochemical* characteristics, which should be enforced in any protein structure determination method. The two aforementioned penalty terms, vary from protein to protein and rely on the experimentally derived information; however, these restraints are protein independent. The most important set of stereochemical restraints are:

1. *Bond lengths and angles*: covalent bonds are very stable. Moreover, covalent bond length fluctuations can be assumed to be negligible, particularly because they cannot be detected in NMR experiments. Consequently, all bond lengths and angles must be set to ideal values computed from accurate X-ray structures; for example, the values reported by Engh and Huber (1991).

2. *Steric constraints*: also called van der Waals constraints prevent non-bonded atoms from penetrating each other (see page 54).

3. *Planarity of coplanar atoms*: proteins contain several coplanar atoms, such as peptide planes, phenolic rings of tyrosine and phenylalanine, carboxyl groups of aspartic acid and glutamic acid, δ-guanido group of arginine, among others. This is also important because nine amino acids have planar side chains (Hooft et al., 1996).

4. *Chirality constraints*: all amino acids have a chirality center at their CA atom except glycine. Moreover, theronine and isoleucine also have a chirality center at their CB atoms (Drenth, 2002). These constraints enforce the correct enantiomer from the two stereoisomers.

These restraints are implemented differently in various methods. For example, the protein X-ray Crystallography refinement method TNT (Tronrud et al., 1987), formulates all of the above constraints in the Cartesian coordinates space. The CORELS program (Sussman, 1985) rewrites the bond length, bond angles, and the dihedral angle restraints using only the distances between atoms, see Fig. 3.6.

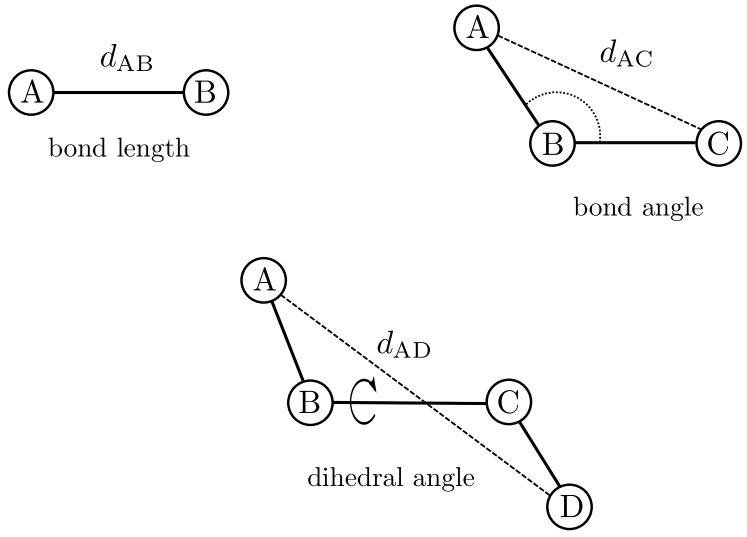


Figure 3.6: The constraints used in the CORELS method: the distance between two bonded atoms, d_{AB}, for bond length, the distance between between an atom and its second nearest neighbor, d_{AC}, for bond angle, and the distance between the first and fourth atom, d_{AD}, for dihedral angles (picture reproduced from Drenth (2002)).

In the torsion angle space, bond lengths and angles, the planarity of the coplanar atoms and the chiralities are automatically enforced, making it appealing for protein structure calculation problems.

Last, there are a few protein quality checking programs that check many more features in addition to the aforementioned restraints. For example, PROCHECK and its descendant PROCHECK-NMR (Morris et al., 1992; Laskowski et al., 1993, 1996), compare the torsion angles (especially Φ, Ψ, and χ_1 angles) with the corresponding values extracted from a dataset of proteins and search for inconsistencies.

A more recent program called MolProbity (Chen et al., 2010) performs a comprehensive analysis on protein and nucleic acid macromolecules. It uses the program REDUCE (Word et al., 1999) to add the missing hydrogens, and then the program PROBE (Word et al., 1999) finds all of the steric clashes. Moreover, torsion angle combinations are verified with an updated Ramachandran plot and the outliers are identified. Moreover, MolProbity measures a parameter called the "CB deviation" that is an indicator of a set of abnormalities in the protein structure (Chen et al., 2010).

Quality Assessment of the Computed Structures

Performance evaluation of protein structure determination methods is critical. We described some of the important stereochemical properties of the proteins; however, a method might produce a structure that satisfies all of these restraints and at the same time its structure significantly deviates from the actual structure.

Any new method, before being applied to data from unknown proteins, must be tested with some datasets for which the structure is already known. One of the most important and popular quality assessment measures is the Root Mean Square Deviation (RMSD) between the computed and the reference structures, which is a measure of disparity between the two conformations.

RMSD Let $X, \tilde{X} \in \mathbb{R}^{3\times n}$ be the reference and computed structures, respectively. Let $\mathcal{R} \subseteq 1{:}n$ be a subset of the atoms in the structure for which we want to measure the RMSD. First, we must perform *orthogonal Procrustes* to optimally align the atoms in $\mathcal{R}$. Let $P = X[:, \mathcal{R}]$, $Q = \tilde{X}[:, \mathcal{R}]$, and $\rho = |\mathcal{R}|$. In addition, let $\bar{\boldsymbol{p}} = \frac{1}{\rho}\sum_{i=1}^{\rho} \boldsymbol{p}_i$ and $\bar{\boldsymbol{q}} = \frac{1}{\rho}\sum_{i=1}^{\rho} \boldsymbol{p}_i$. Orthogonal Procrustes works as follows:

1. We move the centroid of P and Q to the origin by computing

$$P' = P - \bar{\boldsymbol{p}}\mathbf{1}_\rho^\top, \quad Q' = Q - \bar{\boldsymbol{q}}\mathbf{1}_\rho^\top.$$

2. We find a rotation matrix $R \in \mathbb{R}^{3\times 3}$ that minimizes the following objective function:

$$E(R) = \sum_{i=1}^{\rho} \|\boldsymbol{p}'_i - \boldsymbol{q}'_i\|^2.$$

   Let $T = Q'(P')^\top$ be decomposed by SVD as $T = U\Sigma V^\top$. Then the optimal rotation matrix is $R = VU^\top$ (Schönemann, 1966).

3. Last, we compute $\tilde{P}' = RQ'$.

After this procedure, the RMSD is calculated as follows:

$$\text{RMSD} = \sqrt{\frac{1}{\rho}\sum_{i=1}^{\rho} \|\boldsymbol{p}'_i - \tilde{\boldsymbol{p}}'_i\|^2} \quad \text{Å}.$$

The set of selected atoms, $\mathcal{I}$, is formed depending on which parts of the two structures are to be compared. For example, $\mathcal{I}$ can contain only the indices of CA atoms, or atoms in the backbone or it can include all of the heavy atoms in the structures. Generally, if RMSD is less than 1.5-2 Å, the structure is categorized as high-resolution, where most atomic positions are accurate. An RMSD value of around 4 Å can nonetheless be useful, because the residue positions are nearly accurate. Even if RMSD is as high as 6 Å some structural information about the protein could be obtained (Kihara et al., 2009). As a comparative measure, the histogram of resolution of protein structures determined by the X-ray method is illustrated in Fig. 3.7. For some other measures of similarity between structures, see (Wallin et al., 2003).

3.1.5 Euclidean Distance Geometry

Euclidean Spaces

As in (Krislock, 2010), we work only on real finite-dimensional *Euclidean Spaces* $\mathbb{E}$ and define an inner product operator $\langle\cdot,\cdot\rangle : \mathbb{E}\times\mathbb{E} \to \mathbb{R}$ for these spaces:

- For the space of real p-dimensional vectors, $\mathbb{R}^p$,

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle := \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{p} \boldsymbol{x}_i \boldsymbol{y}_i.$$

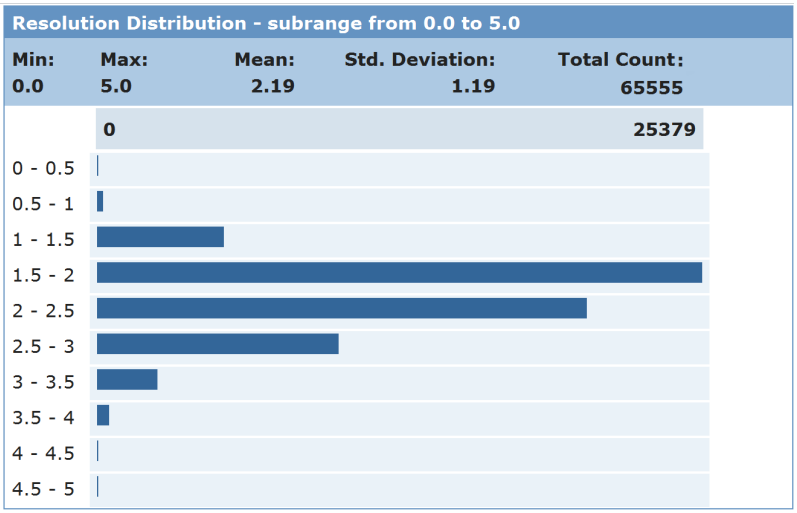


Figure 3.7: Histogram of resolution of protein structures determined by X-ray Crystallography (source: `http://www.pdb.org`).

- For the space of real $p \times q$ matrices, $\mathbb{R}^{p\times q}$,

$$\langle A, B\rangle := \mathbf{trace}(A^\top B) = \sum_{i=1}^{p}\sum_{j=1}^{q} A_{ij}B_{ij}.$$

Given the inner-product operator, for $x \in \mathbb{E}$, its norm is defined as $\|x\| := \sqrt{\langle x, x\rangle}$. For vectors, the *Euclidean* or the ℓ_2-norm is defined as $\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, which is the length of the vector. For matrices, the *Frobenius* norm is defined as $\|A\|_F := \sqrt{\mathbf{trace}(A^\top A)}$.

Linear Maps

Matrix Indexing We use the MATLAB notation for matrices and vectors:

- $1{:}n := \{1, 2, \ldots, n\}$.
- $[A; B] := \begin{bmatrix} A \\ B \end{bmatrix}$.
- For a matrix $A \in \mathbb{R}^{m\times n}$, and for two *sifting* sets $\mathcal{S} \subseteq 1{:}m$ and $\mathcal{T} \subseteq 1{:}n$, $B = A[\mathcal{S}, \mathcal{T}]$ is a $|\mathcal{S}|\times|\mathcal{T}|$ matrix formed by rows and columns of A indexed by $\mathcal{S}$ and $\mathcal{T}$, respectively.

In addition, we define:

$$A[:, \mathcal{T}] := A[1{:}m, \mathcal{T}]$$
$$A[\mathcal{S}, :] := A[\mathcal{S}, 1{:}n]$$
$$A[\mathcal{S}] := A[\mathcal{S}, \mathcal{S}]$$

Adjoint linear operator For two Euclidean subspaces $\mathbb{E}$ and $\mathbb{F}$ with inner-products $\langle \cdot, \cdot \rangle_{\mathbb{E}}$ and $\langle \cdot, \cdot \rangle_{\mathbb{F}}$, respectively, a *linear* map from $\mathbb{E}$ to $\mathbb{F}$ is defined as $\mathsf{A} \colon \mathbb{E} \to \mathbb{F}$. The *adjoint* of A is defined as $\mathsf{A}^* \colon \mathbb{F} \to \mathbb{E}$, which satisfies

$$\langle x, \mathsf{A}^* y \rangle_{\mathbb{E}} = \langle \mathsf{A} x, y \rangle_{\mathbb{F}}, \quad \forall x \in \mathbb{E}, \forall y \in \mathbb{F}.$$

Euclidean Distance Matrix

A matrix $D \in \mathcal{S}^n$ with non-negative elements and zero diagonal is called a *Euclidean Distance Matrix* (EDM), if there exists a set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{R}^r$, such that:

$$D_{ij} = d_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, \quad \forall i, j. \tag{3.3}$$

The smallest value of r is called the *embedding dimension*, denoted $\mathbf{embdim}(D) = r$. The space of all $n \times n$ EDMs is denoted $\mathcal{E}^n$. A *valid* EDM must satisfy three requirements:

1. Since $D_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \geq 0$, all of its elements must be non-negative.
2. Since $D_{ii} = \|\boldsymbol{x}_i - \boldsymbol{x}_i\|^2 = 0$, all of its diagonal elements must be zero, or it should be a *hollow* matrix. It immediately follows that $\mathcal{E}^n \subset \mathcal{S}_H^n$, where $\mathcal{S}_H^n$ is the space of hollow matrices; a matrix $A \in \mathcal{S}^n$ is hollow if $\mathbf{diag}(A) = \mathbf{0}$.
3. It must satisfy the *triangle inequality* for any three points:

   $$d_{ij} \leq d_{ik} + d_{kj}, \quad \forall i, j, k.$$

   Satisfying the triangle inequality for all possible triplet of points requires $\mathcal{O}(n^3)$ constraints, which is not trivial.

Gram Matrix If we define $X := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{r \times n}$, then the matrix of inner-products or the *Gram Matrix* is given by $G := X^\top X$, or equivalently by $G_{ij} := \boldsymbol{x}_i^\top \boldsymbol{x}_j$. It immediately follows that $G \in \mathcal{S}_+^n$, because for any $\boldsymbol{v} \in \mathbb{R}^n$,

$$\boldsymbol{v}^\top G \boldsymbol{v} = \boldsymbol{v}^\top X^\top X \boldsymbol{v} = \boldsymbol{w}^\top \boldsymbol{w} = \|\boldsymbol{w}\|^2 \geq 0,$$

where $\boldsymbol{w} = X\boldsymbol{v}$. The Gram matrix and the Euclidean distance matrix are linearly related; expanding the right hand side of (3.3) we obtain:

$$\begin{aligned} D_{ij} &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j), \\ &= \boldsymbol{x}_i^\top \boldsymbol{x}_i - 2\boldsymbol{x}_i^\top \boldsymbol{x}_j + \boldsymbol{x}_j^\top \boldsymbol{x}_j, \\ &= G_{ii} - 2G_{ij} + G_{jj}. \end{aligned}$$

In the matrix form, we can write:

$$D = \mathsf{K}(G) := \mathbf{diag}(G)\mathbf{1}^\top - 2G + \mathbf{1}\mathbf{diag}(G)^\top, \tag{3.4}$$

where, $\mathsf{K}\colon \mathcal{S}^n \to \mathcal{S}^n$ is a linear map that maps the cone of semidefinite matrices, $\mathcal{S}^n_+$, onto the cone of EDMs, $\mathcal{E}^n$, i.e., $\mathcal{E}^n = \mathsf{K}(\mathcal{S}^n_+)$.

Mapping EDMs to GMs Using the K linear map, any Gram matrix can be easily mapped to its corresponding EDM. For the reverse direction, we use the $\mathsf{K}^\dagger\colon \mathcal{S}^n \to \mathcal{S}^n$ linear map:

$$G = \mathsf{K}^\dagger(D) := -\frac{1}{2} HDH, \quad D \in \mathcal{S}^n_H,$$

where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is conventionally called the *centering* matrix. In fact, $\mathsf{K}^\dagger$ is a linear map that maps the space of EDMs, $\mathcal{E}^n$, onto the set of *centered* positive semidefinite matrices, $\mathcal{S}^n_+ \cap \mathcal{S}^n_C$. A matrix $A \in \mathcal{S}^n$ is centered if its rows (and automatically its columns) sum to zero. The space of centered matrices is defined as $\mathcal{S}^n_C := \{A \in \mathcal{S}^n : A\mathbf{1} = \mathbf{0}\}$.

Schoenberg's theorem Given a distance matrix D, its validity is in question: does the matrix have any embedding? That is, are there a set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{R}^r$, such that their EDM is D? Instead of checking the EDM validity criteria, the following well-known theorem addresses this question in a straightforward manner (Schoenberg, 1935):

Theorem 2 *A matrix $D \in \mathcal{S}^n_H$ is a Euclidean distance matrix if and only if $\mathsf{K}^\dagger(D)$ is positive semidefinite.*

Instead of verifying the triangle inequality constraints one by one, all of them can be easily verified at once by checking whether $\mathsf{K}^\dagger(D) \in \mathcal{S}^n_+$.

Embedding Dimension Assume that by using Theorem 2 we know that our distance matrix is, in fact, a valid EDM. The next question is how to determined the embedding dimension of D. The following theorem addresses this question (Alfakih et al., 1999).

Theorem 3 *Let $D \in \mathcal{E}^n$. Then*

$$\mathbf{embdim}(D) = \mathbf{rank}(\mathsf{K}^\dagger(D)) \leq n - 1. \tag{3.5}$$

While it is impossible to control the embedding dimension of D directly; Theorem 3 provides a mechanism to constrain it by $\mathbf{rank}(\mathsf{K}^\dagger(D))$.

Invariance to rotation and translation Given a set of points $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i \in \mathbb{R}^r$, a *rotation/reflection* matrix $R \in \mathbb{R}^{r \times r}$, such that $R^\top R = I$, and a translation vector $\boldsymbol{t} \in \mathbb{R}^r$, we can write the rotated and translated version of X as:

$$\hat{X} = RX + \boldsymbol{t}\mathbf{1}_n^\top.$$

The Gram matrix of $\hat{X}$ can be written as:

$$\begin{aligned} \hat{G} = \hat{X}^\top \hat{X} &= X^\top X + X^\top R^\top \boldsymbol{t}\mathbf{1}_n^\top + \mathbf{1}_n \boldsymbol{t}^\top RX + \mathbf{1}_n \boldsymbol{t}^\top \boldsymbol{t}\mathbf{1}_n^\top, \\ &= G + \mathsf{D}_{\mathbf{1}}(\boldsymbol{v}), \end{aligned} \tag{3.6}$$

where $\boldsymbol{v} := X^\top R^\top \boldsymbol{t} + \frac{\boldsymbol{t}^\top \boldsymbol{t}}{2}\mathbf{1}_n$, and the linear map $\mathsf{D}_{\mathbf{1}} : \mathbb{R}^n \to \mathcal{S}^n$ is defined as (Alfakih et al., 1999):

$$\mathsf{D}_{\mathbf{1}}(\boldsymbol{v}) := \boldsymbol{v}^\top \mathbf{1}_n^\top + \mathbf{1}_n \boldsymbol{v}^\top.$$

We conclude from (3.6) that:

1. If $\boldsymbol{t} \neq \mathbf{0}$, then $\hat{G} \neq G$. Moreover, when $\boldsymbol{t} = \mathbf{0}$ for any rotation matrix $\hat{G} = G$. That is, the Gram matrix is invariant to rotation but is variant to translation.

2. Since EDMs are invariant to both rotation and translation, we have

   $$\mathsf{K}(G) = \mathsf{K}(\hat{G}) = \mathsf{K}(G + \mathsf{D}_{\mathbf{1}}(\boldsymbol{v})),$$

   which implies that $\mathsf{D}_{\mathbf{1}}(\boldsymbol{v}) \in \mathbf{null}(\mathsf{K})$. Moreover, it can be shown that $\mathbf{null}(\mathsf{K}) = \mathbf{range}(\mathsf{D}_{\mathbf{1}})$ (Al-Homidan and Wolkowicz, 2005).

Based on these propositions, we see that $\mathsf{K}^\dagger$ is the Moore-Penrose pseudoinverse of K (Al-Homidan and Wolkowicz, 2005), because for an arbitrary Gram matrix G one of the four Moore-Penrose pseudoinverse conditions can be easily checked:

$$\mathsf{K}(\mathsf{K}^\dagger(\mathsf{K}(G))) = \mathsf{K}(\mathsf{K}^\dagger(D)) = \mathsf{K}(G_C) = \mathsf{K}(G),$$

where G_C is the corresponding centered Gram matrix; verifying the remaining three is straightforward.

Multidimensional Scaling

At this point, we know that our EDM D is (i) a valid EDM; and (ii) has an embedding dimension of r. To compute its embedding, i.e., $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i \in \mathbb{R}^r$? We can use the popular *multidimensional scaling* (MDS) method (Torgerson, 1952). It suffices to compute the eigenvalue decomposition of $\mathsf{K}^\dagger(D)$ as:

$$\mathsf{K}^\dagger(D) = V \Lambda V^\top,$$

where $V \in \mathbb{R}^{n \times r}$ is the matrix of eigenvectors of $\mathsf{K}^\dagger(D)$ and the diagonal matrix $\Lambda \in \mathcal{S}^r_{++}$ holds the eigenvalues of $\mathsf{K}^\dagger(D)$, i.e., $\mathbf{diag}(\Lambda) = [\lambda_1, \ldots, \lambda_r]^\top$, such that $\lambda_1 \geq \cdots \geq \lambda_r > 0$. The embedding is computed as:

$$X = \Lambda^{1/2} V^\top. \tag{3.7}$$

Since $\mathsf{K}^\dagger(D)$ is centered, we know that $V^\top \mathbf{1} = \mathbf{0}$ and consequently, $X\mathbf{1} = \mathbf{0}$. That is, the embedding computed by MDS is always zero-mean, i.e., $\sum_{i=1}^n \boldsymbol{x}_i = \mathbf{0}$.

3.2 Literature Review

The protein NMR method is fundamentally different with the X-ray method: it is not a "microscope with atomic resolution", rather it provides indirect structural information. For example, NMR cannot be used for measuring the bond lengths or angles. Instead, it provides a network of distance restraints between spatially proximate hydrogen atoms. Protein NMR is a *de novo* method: in *de novo* protein structure determination, the structure is computed without using any given templates. Therefore, it is used for discovering new protein folds.

Since the NMR method only provides implicit and indirect information about the protein structure, it relies heavily on complex computational algorithms and methods. The existing methods for protein NMR can be categorized into two major and two minor groups:

1. (*major*) Methods based on **Euclidean Distance Matrix Completion** (EDMC)[2];
2. (*major*) Methods based on **Molecular Dynamics** and **Simulated Annealing**;
3. (*minor*) Methods based on **Local/Global Optimization**;

[2]Mathematical Distance Geometry is used in some older papers, e.g.,(Güntert, 1998), however, considering the nature of these methods, we believe EDMC is more appropriate.

4. (*minor*) Methods originated from **Sequence-based Protein Structure Prediction** algorithms.

These first three groups are briefly reviewed in the next section. The sequence-based methods were briefly reviewed and our proposed method was presented in Chapter 2.

It should be mentioned that most of the existing methods, except for the most popular ones, make these unrealistic assumptions, which we address:

1. Assuming to know the (nearly) exact distances between atoms with small bounded perturbations,
2. Assuming to have both tight upper and lower bounds, while NOE only provides upper bounds and the steric constraints provide merely loose lower bounds,
3. Assuming to have the distances between any type of nuclei (not just hydrogens); this usually results in a staggering but unrealistic number of upper and lower bounds,
4. Constraining bond lengths and bond angles with upper and lower bounds similar to NOE restraints, while these values should vary very little from their ideal values,
5. Ignoring the fact that not all hydrogens can be uniquely assigned, e.g., the hydrogens in the methyl groups, which necessitates adding several virtual *pseudo-atoms* (for example one pseudo-atom is added at the center of three hydrogens in the methyl groups) and adjusting the corresponding upper bounds, making them even more inaccurate,
6. Using RMSD as the only figure of merit, while RMSD is a good indicator of the computed structure's quality, withouts satisfying the stereochemical restraints and passing rigorous tests of protein quality verification programs such as MolProbity (Chen et al., 2010), this can be misleading,
7. Overlooking the ambiguity in the NOE restraints. Mainly because the upper bounds are artificially synthesized from high-resolution structures in PDB, rather than being inferred from NOE peaks.

We believe that these assumptions often lead to failure of these algorithms when they are applied to real protein NMR data sets.

3.2.1 EDMC-based Methods

In the EDMC-based methods, the objective is to complete an incomplete EDM $D \in \mathcal{E}^n$, where D_{ij} is the distance between atoms a_i and a_j, i.e.,

$$D_{ij} = d_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$$

where $\boldsymbol{x}_i$, $\boldsymbol{x}_j \in \mathbb{R}^3$ are the Cartesian coordinates of atoms a_i and a_j.

In EDMC problems, the input is as follows:

1. Maximum embedding dimension, r;
2. The set of equality constraints, $\mathcal{E}$, such that $d_{ij} = e_{ij}$, $\forall (i,j) \in \mathcal{E}$;
3. The set of upper bounds, $\mathcal{U}$, such that $d_{ij} \leq u_{ij}$, $\forall (i,j) \in \mathcal{U}$; and
4. The set of lower bounds, $\mathcal{L}$, such that $d_{ij} \geq l_{ij}$, $\forall (i,j) \in \mathcal{L}$,

Any valid EDM D that satisfies the distance constraints and has $\mathbf{embdim}(D) \leq r$ is an acceptable solution.

EDMC-based methods are well-suited for the protein NMR problem, because instead of finding the protein conformation in the three-dimensional Cartesian space, the problem is solved in the space of n-dimensional EDMs, $\mathcal{E}^n$. After finding the EDM D, the embedding of the solution is found by MDS.

EDMC is appealing because in protein NMR we have: (i) a set of equality constraints $\mathcal{E}$, (ii) a set of upper bounds $\mathcal{U}$ on some of the distances, derived from NOE experiments; and (iii) a set of lower bounds $\mathcal{L}$.

Methods working directly on the EDM

In the early years of protein NMR, many EDMC-based methods were proposed that naturally directly worked on the corresponding EDM. The first method to use EDMC for protein NMR was developed by Braun et al., which was based on using MDS for finding the final embedding (Braun et al., 1981); they solved the structure of a peptide with 109 atoms and 23 distance restraints. Their proposed method includes these steps:

1. Using bond length, bond angle, van der Waals contact distances and NMR restraints, set some lower, and upper bounds on the distances between atom pairs. For atom pairs without any experimental upper bounds, set an arbitrary large bound, say 40 Å. Check for the consistency of the bounds by verifying the triangle inequality and fix possible problems.

2. Randomly sample all the distances within the allowed ranges, $l_{ij} \leq d_{ij} \leq u_{ij}$
3. Check the triangle inequality for all of the sampled distances and change the distances such that there are no violations
4. Compute the Gram matrix, $\mathsf{K}^{\dagger}(D)$, and use MDS to find the best three-dimensional embedding $\{\boldsymbol{x}_i\}_{i=1}^{n}$, $\boldsymbol{x}_i \in \mathbb{R}^3$
5. Adjust the distances, because it is possible that $\mathbf{embdim}(D) > 3$. Repeat steps three to five (usually five to 10 times)
6. Refine the structure by using a conjugate gradient-based (Powell, 1964) technique.

This method, alongside similar methods, working directly on the distance matrix, face two major drawbacks:

1. Randomly guessing the unknown distances is ineffective, it results in a large number of violated triangle inequality constraints. Moreover, there is no systematic method for fixing the violated constraints. After several iterations of distance correction, distances tend to become large (Güntert, 1998).
2. There is no way to control the embedding dimensionality. In fact, the final solution could be of any dimension up to n; therefore, even generating a perfect EDM could result in an impractical high-dimensional embedding.

In contrast to these limitations, Arseniev et al. (1984) used this method on a simplified model of protein molecule to compute the structure of a short globular protein, a scorpion insectotoxin with 35 amino acid residues.

Havel et al. (1983) proposed a similar but improved iterative EDMC method called EMBED. Havel and Wüthrich (1984) provided an improved version of this algorithm called DISGEO, which was in use for several years. The DISGEO method runs two passes of EMBED: the first runs on only on a subset of atoms computing a skeleton for the molecules and the second run computes position of the remaining atoms. Williamson et al. (1985) used DISGEO to calculate the structure of proteinase inhibitor IIA from bull seminal plasma with 58 amino acid residues and nearly 900 atoms. The emergence of new modern structure calculation techniques has made these methods obsolete.

Other methods that impose unrealistic assumptions: for example, Trosset (1998) for $d_{ij} \leq 7$ Å considers upper and lower bounds of $d_{ij} + 0.01$ Å and $d_{ij} - 0.01$ Å, respectively. Recently, Grooms et al. (2009) proposed StrainMin that was tested using *exact* distances of atoms closer than 6 Å.

Methods working on the Gram Matrix

Enforcing the triangle inequality constraints for a large set of atoms (in the range of a few thousands) is cumbersome and challenging, if not impossible. Using the Gram matrix instead of the EDM for solving EDMC problems is more straightforward and has many advantages:

1. The Gram matrix and EDM are linearly related and are converted to each other using the K and the $\mathsf{K}^\dagger$ linear maps; rendering trivial the revision of the constraints in term of the Gram matrix.

2. Based on Theorem 2 by Schoenberg (1935), instead of enforcing $\mathcal{O}(n^3)$ triangle inequality constraints, it is sufficient to enforce that the Gram matrix is positive semidefinite.

3. The embedding dimension and the rank of the Gram matrix are directly related (see Theorem 3); this provides a means to constrain the embedding dimension.

Semidefinite programming (SDP) is a natural choice for formulating the EDMC problem using the Gram matrix. SDP-based EDMC methods demonstrated great success in solving the *sensor network localization* (SNL) problem (Doherty et al., 2001; Biswas and Ye, 2004; Biswas et al., 2006; Wang et al., 2008; Kim et al., 2009; Krislock and Wolkowicz, 2010). In the SNL problem, the location of a set of sensors in $\mathbb{R}^2$ is determined, given the short-range distances between spatially proximate sensors. As a result, the SNL problem is inherently similar to the protein NMR problem.

The major obstacle in extending SNL methods to protein NMR is two fold: first, the jump from $\mathbb{R}^2$ to $\mathbb{R}^3$ makes the problem more difficult, and second, the $\mathcal{O}(n^3 + m^3)$ complexity per iteration of SDP solvers is not tractable. To overcome this limitation, Biswas et al. (2008) proposed DAFGAL, which is built on the idea of *divide-and-stitch*. Using a symmetric reverse Cuthill-McKee permutation to aggregate the nonzero distances closer to the main diagonal, the original large distance matrix is clustered into overlapping smaller quasi-diagonal blocks of variable size The smaller problems are solved and then stitched together using their commonly shared atoms, followed by a gradient-descent based post-processing.

The DAFGAL method was tested on several synthetic datasets as follows: protein structures from PDB are taken and all the distances between atom pairs with $d_{ij} \leq 6$Å are computed. Consequently, lower and upper bounds of the form $(d_{ij} \max(0, 1 - |\underline{\varepsilon}|), d_{ij}(1 + |\overline{\varepsilon}|))$ are generated for each pair, where $\underline{\varepsilon}, \overline{\varepsilon} \sim \mathcal{N}(0, \sigma^2)$. To make the tests more realistic, only a fraction of these constraints are kept. For $\sigma = 0.01$ to 0.1 and for a fraction of kept short range distances ranging from 30% to 100%, they report RMSD values in the range

0.1 to 2.5Å. Last, DAFGAL performs poorly for large proteins, if less than 50% of short range distances are given.

The DISCO method, a direct descendant of DAFGAL is proposed by Leung and Toh (2009). It is an extension of DAFGAL that can determine protein molecules with more than 10,000 atoms. They use a *divide-and-conquer* technique and break the problem into smaller subproblems. The improved methods for partitioning the partial distance matrix and iteratively aligning the solutions of the subproblems, boosts DISCOS's performance in comparison to DAFGAL. They adapt a similar experimental setup to DAFGAL and generate upper and lower bounds of the form $(\max(1,(1-|\underline{\varepsilon}|d_{ij})), d_{ij}(1+|\overline{\varepsilon}|))$ where $\underline{\varepsilon}, \overline{\varepsilon}$ are normal or uniform random variables. For 20%-30% of the short range distances with the noise ranging from 10% to 20% the core structure, RMSD is 0.6-2 Å, where the core structure is defined as "the union of the likely localizable components" (Leung and Toh, 2009).

3.2.2 Methods based on Local/Global Optimization

Protein NMR can also be formulated as an optimization problem in Cartesian coordinates space. For example, an objective function representing restraint violations can be minimized. The main assumption in these methods is that the global minimum of the optimization problem is close to the native structure of the target protein. However, the obstacle here is the severe non-convexity of the optimization problem. As a result, it is likely that the search algorithm is trapped in one of many local minima. Several different minimization protocols have been proposed to overcome the local minima problem.

Global optimization of a potential energy function based solely on the amino acid sequence is able to generate structures as close as 4.2 Å RMSD (Liwo et al., 1999). Better results can be expected if additional structural information is also available. Moré and Wu (1997, 1999) presented DGSOL that formulated the protein structure determination problem as a nonlinear least squares problem. The DGSOL method solves the problem by a global continuation algorithm. In particular, it uses Gaussian smoothing for gradually smoothing the original objective function, which is helpful because a smoother function has less local minima, so increases the likelihood of finding the global optimum. They set upper and lower bounds of the form $((1-\varepsilon)d_{ij}, (1+\varepsilon)d_{ij})$, $0 \leq \varepsilon \leq 0.16$, on all the atoms pairs within two neighbor residues. For a peptide fragment with only 200 atoms, DGSOL computed a structure with a RMSD of 2.9 Å.

Williams et al. (2001) proposed GNOMAD, which takes as input a subset, e.g., 30%, of "short-range distances" (defined as all the distances between two consecutive residues), and distances fixing bond lengths and bond angles. For a protein with 1870 atoms (1TIM) they defined 35201 short-range distances; using 30% of these exact distances, GNOMAD

computed a structure with an RMSD of 1.07 Å[3].

A major breakthrough in protein NMR was the use of torsion angles instead of Cartesian coordinates. Using torsion angles results in problem size reduction by at least one order of magnitude, compared to using Cartesian coordinates; the average number of variables are reduced from nearly $18.2 \times 3 \approx 55$ to 4.5 (see Table 3.3). The major obstacle at the time was the higher complexity of computing the gradient with regard to the torsion angles. Abe et al. (1984) presented a recursive $\mathcal{O}(t)$ method for computing the gradient in the torsion angles space with t torsion angles for tree-like structures, such as in polypeptide chains.

A widely-adapted method, called the *variable target function*, consists of a series of minimizations, each with a different objective function (Braun and Go, 1985). The distance restraints between distant atoms in the polypeptide chain are added as the optimization progresses. Combining the variable target method with the fast gradient computation technique, Braun and Go (1985) proposed DISMAN, which has proved to be effective for α-helical proteins, due to their mostly short-range restraints, however; it fails for β-sheet proteins that have many long range contacts.

3.2.3 Methods based on Molecular Dynamics

Molecular Dynamics (MD) simulation is based on classical mechanics; the molecule is modeled as a set of connected masses each with their own positions and velocities, where Newton's motion equations are numerically solved and the molecule's trajectory in the Cartesian space is computed. The MD method has a huge advantage; the molecule has both the *potential* and the *kinetic* energies; therefore, once trapped, the molecule may escape from the local minima (Güntert, 1998).

MD was originally used for modeling motions of gases, but with emerging computer technologies, it was also applied to larger molecules, such as proteins. McCammon et al. used MD to simulate BPTI at room temperature (McCammon et al., 1977). It was not until 1985 that Kaptein et al. used NMR information and MD to compute the structure of the Lac repressor headpiece. However, it should be noted that the authors had prior knowledge about the structure and that the protein was modeled as three helices (Kaptein et al., 1985).

Clore et al. developed a hybrid method; first, a roughly accurate structure was computed by an EDMC-based method; then a constrained MD was run to minimize the energy of the structure (Clore et al., 1986). The potential energy was computed using an empirical force field such as that found in CHARMM (Brooks et al., 1983) and the MD was

[3]Leung and Toh (2009) believe that the reported RMSD does not agree with the plotted structure and the correct RMSD should be about 2-3 Å.

run at room temperature. Conventional MD packages had two major drawbacks: (*i*) the exploration of the conformation space was slow, because the MD was run at room temperature, at which proteins do not change conformation drastically; (*ii*) computation of the empirical force field usually involved long-range interactions and as a result, all the pair-wise distances needed to be computed rendering MD computationally expensive.

While in MD simulations the trajectory of the molecule is important (for example for analyzing the folding pathways), in contrast, in protein NMR the objective is the final conformation and the trajectory is irrelevant as long as the final structure is close to the native structure. Another major breakthrough was the introduction of simpler and more practical potential energy functions. Nilges et al. (1988) made some improvements in the MD-based protein NMR structure determinatios: (*i*) instead of an empirical energy function, they proposed a simple *geometrical* energy function based on the NOE restraints that asymptotically linearly penalized large violations and (*ii*) they combined *simulated annealing* (SA) with MD. Since SA is the cornerstone of most of the modern protein NMR structure determination techniques, we explore it in detail.

Simulated annealing

Simulated annealing is a heuristic global optimization technique originated from Statistical Mechanics and it mimics the process by which solids attain their minimum energy conformation. For example, when making glass it is initially heated to high temperatures, the "melting" phase, and then it is slowly cooled based on an *annealing schedule*, until it 'freezes', i.e., no further change has occurred (Kirkpatrick et al., 1983; Černý, 1985).

Assume that the state of the protein molecule s is defined as the set of coordinates of its atoms, $\{\boldsymbol{x}_i\}$. For each state, an energy value is computed by the energy function $E(s)$ that represents the objective function to be minimized. If the next possible state is s' with energy $E(s') = E(s) + \Delta E$, then:

$$\Pr[s' \mid s] = \begin{cases} 1 & \text{if} \quad \Delta E < 0 \\ \exp(-\Delta E / k_B T) & \text{if} \quad \Delta E \geq 0 \end{cases}$$

where T is the absolute temperature and k_B is the Boltzman constant. The difference between SA and most other optimization methods, e.g., gradient-based methods, is that occasionally moves to a higher energy state, with probability $\exp(-\Delta E/k_B T)$, which is controlled by the temperature; this makes escaping from local minima possible in SA. When the temperature is high, $\exp(-\Delta E/k_B T)$ is close to one, so that SA can explore most of the search space without getting stuck in the local minima (coarse optimization). As SA goes on, the temperature is lowered (cooled down), then $\exp(-\Delta E/k_B T)$ becomes closer to zero, which makes jumping to higher-energy states less likely (fine tuning) (Kirkpatrick et al., 1983).

Last, being a heuristic method, SA provides no theoretical guarantee on the convergence speed or on the quality of the final solution, which is how far the final solution is from the optimal state. Johnson et al. evaluated the performance of SA on different NP-hard problems in a comprehensive study (Johnson et al., 1989, 1991); in some problems SA outperformed other methods, while in some other problems it was not meritorious.

Simulated annealing by molecular dynamics simulation

SA/MD in Cartesian coordinates space Running simulated annealing by molecular dynamics simulation was a breakthrough in protein NMR structure determination; these methods were able to search the massive conformation space without being trapped in one of the numerous local minima. The XPLOR method (Brünger, 1993; Schwieters et al., 2003, 2006) was one of the first successful and widely-adapted methods that was built on the molecular dynamics simulation package CHARMM (Brooks et al., 1983), but with a significant difference: it was modified to specifically compute and refine the structure (instead of doing simulation). A coarsely correct structure was calculated by an EDMC-based method and then refined and fine-tuned by a SA/MD protocol. In practice, the EDMC-based module was not always meritorious, since the distances randomly sampled from the $D_{ij} \in [L_{ij}\ U_{ij}]$ range tend to be large, considering the fact that U_{ij}s were usually large.

SA/MD in the torsion angles space The number of degrees of freedom in the torsion angles space is nearly 10 times smaller than in Cartesian coordinates space, while being equivalent under mild assumptions. In the torsion angles space, high-frequency bond length and angle fluctuations are absent (they are assumed to be fixed); consequently, the integration time steps are much longer and the MD can be run in higher temperatures. In MD in Cartesian coordinates space, Newton's equations of motion are solved in $\mathcal{O}(n)$; while in the torsion angles space, Lagrange equations of motion must be solved, which requires computing the Hessian matrix (the second derivative of the potential energy with regard to torsion angles) and requires solving a system of linear equations, which takes $\mathcal{O}(t^3)$, where t is the total number of torsion angles.

Molecular dynamics in the torsion angles space or *torsion angle dynamics* (TAD), became popular after (Jain et al., 1993) proposed a recursive linear time, $\mathcal{O}(t)$, method for solving Lagrange equations of motions for tree like structures. The TAD method has an appealing feature: the stereochemical restraints such as bond lengths, angles, and chiralities are automatically satisfied. The torsion angle dynamics algorithm implemented in the program CYANA (Güntert, 2004) and previously in the program DYANA (Güntert et al., 1997), is one of the fastest and most widely-used methods.

CYANA uses a complex five-stage simulated annealing protocol: initial minimization (100 CG steps with close restraints + 100 CG steps with all restraints), high-temperature phase ($N/5$ TAD steps at $T = 10,000^\circ$ K), slow cooling ($4N/5$ TAD steps, reducing temperature), low-temperature phase with individual hydrogen atoms (100 CG + 200 TAD at $T = 0^\circ$ K), and final minimization (1000 CG steps) (Güntert, 2004), where CG stands for *conjugate gradient* minimization and N is the total number of TAD steps.

3.3 The Proposed Method

We present our proposed method, called "SPROS" (Semidefinite Programming-based Protein structure determination). We review the necessary background, followed by the description of techniques used for problem size reduction. Last, we describe the different steps used in the proposed method.

3.3.1 Background

Cones

Let $\mathbb{E}$ be an Euclidean space and $K \subseteq \mathbb{E}$. The set K is called a *cone* or *nonnegative homogeneous*, if for every $x \in K$ and $\alpha \geq 0$, $\alpha x \in K$. In addition, K is a convex cone, if for any $x_1, x_2 \in K$ and $\alpha_1, \alpha_2 \geq 0$, we have $\alpha_1 x_1 + \alpha_2 x_2 \in K$ (Boyd and Vandenberghe, 2004). The *dual cone* of cone K is defined as

$$K^* = \{y \in \mathbb{E} : \langle x, y \rangle \geq 0, \forall x \in K\}.$$

The dual cone is a cone as well, and is always closed and convex, even if the original cone is not. A cone K is self-dual if $K = K^*$.

Two important sets of matrices are the set of positive semidefinite and the set of positive definite matrices:

$$\mathcal{S}^n_+ = \{X \in \mathcal{S}^n : \boldsymbol{v}^\top X \boldsymbol{v} \geq 0, \forall \boldsymbol{v} \in \mathbb{R}^n\},$$
$$\mathcal{S}^n_{++} = \{X \in \mathcal{S}^n : \boldsymbol{v}^\top X \boldsymbol{v} > 0, \forall \boldsymbol{v} \in \mathbb{R}^n, \boldsymbol{v} \neq \mathbf{0}\}.$$

Note that $\mathcal{S}^n_+$ is a self-dual cone, hence, it is closed and convex. We say $A \succeq B$ if $A - B \in \mathcal{S}^n_+$, similarly $A \succ B$ if $A - B \in \mathcal{S}^n_{++}$; this is known as the Löwner partial order.

Faces The convex cone $F \subseteq K$ is a *face* (Ramana et al., 1997) of the cone K (denoted $F \trianglelefteq K$) if:

$$x, y \in K,\ x + y \in F \Rightarrow x, y \in F.$$

Moreover, if F is a face of K and $F \neq K$, then $F \triangleleft K$.

A Simple facial reduction

Before giving the formal definition of the facial reduction, we start by presenting a simple linear program whose domain is the *nonnegative orthant* cone defined as

$$\mathbb{R}^n_+ := \{\boldsymbol{x} \in \mathbb{R}^n \colon \boldsymbol{x}_i \geq 0, i = 1, \ldots, n\}.$$

The linear program is as follows:

$$\begin{array}{ll} \text{minimize} & 2x_1 + x_2 - 5x_3 + 3x_4 \\ \text{subject to} & x_1 + x_2 + x_3 + x_4 = 1 \\ & 2x_1 - x_2 - x_3 = -1 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{array} \tag{3.8}$$

Initially, the linear program in (3.8) may seem to have four unknowns; however, it has only two unknowns. If we sum the two equality constraints we arrive at:

$$3x_1 + x_4 = 0,\ x_1, x_4 \geq 0 \Rightarrow x_1 = x_4 = 0;$$

the linear program then simplifies to:

$$\begin{array}{ll} \text{minimize} & x_2 - 5x_3 \\ \text{subject to} & x_2 + x_3 = 1 \\ & x_2, x_3 \geq 0 \end{array} \tag{3.9}$$

which is a simpler linear program. Formally, we observe that:

$$\mathcal{F} = \{\boldsymbol{x} \in \mathbb{R}^4_+ \colon \boldsymbol{a}_i^\top \boldsymbol{x} = b_i,\ \forall i\} \subseteq \{\boldsymbol{x} \in \mathbb{R}^4_+ \colon x_1 = x_4 = 0\}$$

where $\mathcal{F}$ is the feasible set of the linear program in (3.8) and $\{\boldsymbol{x} \in \mathbb{R}^4_+ \colon x_1 = x_4 = 0\}$ is a face of $\mathbb{R}^4_+$. In general, faces of nonnegative orthant are formed by setting some of the elements to zero:

$$\{\boldsymbol{x} \in \mathbb{R}^n_+ \colon x_i = 0, \forall i \in \mathcal{I} \subseteq 1{:}n\} \trianglelefteq \mathbb{R}^n_+ \ .$$

Slater's Condition

We present some background for *strong duality*. Consider the following optimization problem in the standard *primal* form:

$$\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, p \end{array} \tag{3.10}$$

where $\boldsymbol{x} \in \mathbb{R}^n$. Moreover, assume that the domain $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}(f_i) \cap \bigcap_{i=1}^{p} \mathbf{dom}(h_i)$ is nonempty and the optimal value of the optimization problem is p^*. Now, consider the *dual problem* of (3.10) as:

$$\begin{array}{ll} \text{maximize} & g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \text{subject to} & \lambda_i \geq 0, \quad i = 1, \ldots, m, \end{array} \tag{3.11}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^\top$, $\boldsymbol{\nu} = [\nu_1, \ldots, \nu_p]^\top$, and the *dual function* $g \colon \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\boldsymbol{x} \in \mathcal{D}} \left(f_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{p} \nu_i h_i(\boldsymbol{x}) \right).$$

Moreover, assume the optimal value of (3.11) is d^*. For the optimization problems in (3.11) and (3.10), note that *weak duality*, $p^* \geq d^*$, always holds. However, if $d^* = p^*$, that is both the primal and the dual problems attained their optimal values and these values are the same, the *duality gap*, $p^* - d^*$, is zero and we say that strong duality holds (Boyd and Vandenberghe, 2004).

Strong duality does not hold in general, but if the primal form in (3.10) is convex, that is:

$$\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, m \\ & A\boldsymbol{x} = \boldsymbol{b}, \end{array} \tag{3.12}$$

such that $f_0, \ldots, f_m$ are convex, then strong duality usually (but not always) holds.

One simple *constraints qualification* condition, that guarantees strong duality for convex optimization problems, is called *Slater's condition* or *strict feasibility*: there exists $\boldsymbol{x} \in \mathbf{relint}(\mathcal{D})$ such that:

$$f_i(\boldsymbol{x}) < 0, \ i = 1, \ldots, m, \qquad A\boldsymbol{x} = \boldsymbol{b},$$

such a point $\boldsymbol{x}$ is called *strictly feasible*. This is because the inequalities are held as strict inequalities. However, if $f_1, \ldots, f_m$ are affine, strict inequality is not required (Boyd and Vandenberghe, 2004).

A question arises: what happens if there is no such strictly feasible point available? *barrier* methods (also called *interior point* methods) require a strictly feasible starting point. If the given starting point is not strictly feasible, they solve the *Phase I optimization problem* to find one such point or declare infeasibility if they fail; one example of the Phase I optimization problem is formulated as:

$$\begin{array}{ll} \text{minimize} & s \\ \text{subject to} & f_i(\boldsymbol{x}) \leq s, \quad i = 1, \ldots, m \\ & A\boldsymbol{x} = \boldsymbol{b}, \end{array} \tag{3.13}$$

where s can be interpreted as a bound on the maximum possible infeasibility of the inequalities (Boyd and Vandenberghe, 2004). Let $\bar{p}^*$ be the optimal value of (3.13). If $\bar{p}^* > 0$, then (3.12) is infeasible and if $\bar{p}^* < 0$, then (3.12) is strictly feasible. It is worthwhile to note that when $\bar{p}^* = 0$, this indicates (3.12) is feasible but is not strictly feasible.

It has been shown that convex problems with convex inequalities and linear equalities are solved efficiently using barrier methods if a strictly feasible point is available. However, if the problem is close to the feasibility/infeasibility boundary, solving the problem can take an extremely long time. If the problem falls exactly on the boundary, i.e., $\bar{p}^* = 0$, then solving time is theoretically unbounded (Boyd and Vandenberghe, 2004).

In some types of convex optimization problems, when the Slater's condition is not satisfied, facial reduction can both: (i) reduce the problem size and, more importantly, (ii) make it strictly feasible and thus efficiently solvable by barrier methods.

Faces of the the semidefinite cone

In the linear programming example in (3.8) note that, in the nonnegative orthant cone, a face is formed by setting a subset of elements to zero. Here, we show that a face of the semidefinite cone is made by setting a subset of eigenvalues to zero. Finding the faces of the semidefinite cone is called *semidefinite facial reduction*. Some early studies of faces of convex cones are done by Barker (1973); Barker and Carlson (1975); Barker (1977). Later, Borwein and Wolkowicz (1981a,b,c) introduced facial reduction as a mean for regularization of convex programming problems.

We review the faces of the cone of semidefinite matrices.

Background For any arbitrary $U \in \mathbb{R}^{n \times t}$, we have (Krislock, 2010):

1. $Y \in U\mathcal{S}_+^t U^\top$ if and only if $Y \succeq 0$ and $\mathbf{range}(Y) \subseteq \mathbf{range}(U)$
2. If U is full column rank, then $Y \in U\mathcal{S}_{++}^t U^\top$ if and only if $Y \succeq 0$ and $\mathbf{range}(Y) = \mathbf{range}(U)$

These two results are important to defining the faces of $\mathcal{S}_+^n$. If we can find a full column rank matrix U, such that $\mathbf{range}(Y) = \mathbf{range}(U)$, then the $n \times n$ positive semidefinite matrix Y can be decomposed into $Y = UZU^\top$, for some $Z \in S_{++}^t$. If $t \ll n$, this results in a significant reduction in the matrix size.

Proposition 1 (Krislock (2010)) *For orthogonal $Q = [U\ V] \in \mathbb{R}^{n\times n}$, if*

$$F := U\mathcal{S}_+^t U^\top = \left\{ Q \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} Q^\top : B \in \mathcal{S}_+^t \right\},$$

then we have

1. $F \trianglelefteq \mathcal{S}_+^n$.
2. $F = \{Y \in \mathcal{S}_+^n : \mathbf{range}(Y) \subseteq \mathbf{range}(U)\} = \{Y \in \mathcal{S}_+^n : \mathbf{range}(V) \subseteq \mathbf{null}(Y)\}$.
3. $\mathbf{relint}(F) = U\mathcal{S}_{++}^t U^\top = \{Y \in \mathcal{S}_+^n : \mathbf{range}(Y) = \mathbf{range}(U)\}$.

Moreover, if $F \trianglelefteq \mathcal{S}_+^n$ and $Y \in \mathbf{relint}(F)$, then $F = U\mathcal{S}_+^t U^\top$, where $U \in \mathbb{R}^{n\times t}$ is full column rank and $\mathbf{range}(U) = \mathbf{range}(Y)$. It can be seen that the face of the semidefinite cone is made by setting $n - t$ of the eigenvalues of $Y \in \mathcal{S}_+^n$ to zero (Krislock, 2010).

Semidefinite programming

Semidefinite optimization or, more commonly, *semidefinite programming* (SDP), is a class of convex optimization methods that attracted much attention in the optimization community and has found numerous applications in different science and engineering fields. Notably, several diverse convex optimization problems can be formulated as SDPs (Vandenberghe and Boyd, 1996). While the research on SDP started long ago (Bellman and Fan, 1963; Craven, 1981; Shapiro, 1985; Fletcher, 1985; Allwright, 1988; Wolkowicz, 1981; Kojima et al., 1997), it was the break-through discovery by Nesterov and Nemirovski that made solving SDP problems feasible (Nesterov and Nemirovski, 1988). They proved that any convex optimization problem, including SDP problems, can be solved in polynomial time using the interior point methods (Nesterov and Nemirovski, 1988, 1990, 1991, 1994).

Current state of the art SDP solvers are based on *primal-dual* interior point methods, which are variations of Newton's method applied to modified KKT equations (Boyd and Vandenberghe, 2004), specifically, as in the widely-used HKM method (Helmberg et al., 1996; Kojima et al., 1997; Monteiro, 1997).

SDP Formulation In the primal form, we formulate the SDP problem as:

$$\begin{aligned} \text{minimize} \quad & \langle C, K \rangle \\ \text{subject to} \quad & \mathsf{A}(K) = \boldsymbol{b} \\ & K \in \mathcal{S}_+^n \end{aligned} \tag{3.14}$$

where $C \in \mathcal{S}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\mathsf{A}\colon \mathcal{S}^n \to \mathbb{R}^m$ is a linear map defined as:

$$\mathsf{A}(K) = \begin{bmatrix} \langle A_1, K \rangle \\ \langle A_2, K \rangle \\ \vdots \\ \langle A_m, K \rangle \end{bmatrix}, \quad A_i \in \mathcal{S}^n,\ i = 1, \ldots, m. \tag{3.15}$$

The adjoint linear map of A is $\mathsf{A}^*\colon \mathbb{R}^m \to \mathcal{S}^n$ given by

$$\mathsf{A}^*(\boldsymbol{y}) = \sum_{i=1}^{m} y_i A_i,$$

which can be used to define the dual form of (3.14) as:

$$\begin{aligned} \text{maximize} \quad & \langle \boldsymbol{b}, \boldsymbol{y} \rangle \\ \text{subject to} \quad & Q = \mathsf{A}^*(\boldsymbol{y}) - C \\ & Q \in \mathcal{S}_+^n \end{aligned} \tag{3.16}$$

SDP complexity The complexity of solving SDP problems is dominated by three factors: (i) the positive semidefinite (PSD) matrix size, n; (ii) the number of constraints, m; and (iii) the structure (sparse, dense, block diagonal, low rank, among others) of the constraint matrices, $\{A_i\}$. When analyzing the complexity of the SDP solvers, typically the complexity per iteration is considered, because the number of iterations grows slowly with the problem size for a given accuracy (Borchers and Young, 2007). In practical problems, the number of iterations is usually less than fifty and rarely more than one hundred .

In SDP solvers based on the HKM method, such as SeDuMi (Sturm, 1999), CSDP (Borchers, 1999), and SDPT3 (Tütüncü et al., 2003), the complexity is $\mathcal{O}(mn^3) + \mathcal{O}(m^2n^2)$ $\Theta(m^3) + \Theta(n^3)$ per iteration. The most complex step is computing the dense positive semidefinite Schur compliment matrix (Borchers and Young, 2007):

$$O = \left[\mathsf{A}(Q^{-1}A_1K),\ \mathsf{A}(Q^{-1}A_2K), \ldots,\ \mathsf{A}(Q^{-1}A_mK)\right], \quad O \in \mathcal{S}_+^m. \tag{3.17}$$

Finding Q^{-1} takes $\Theta(n^3)$. Moreover, for dense and high-rank $\{A_i\}$ computing O in (3.17)

takes $\mathcal{O}(mn^3 + m^2n^2)$, while for sparse matrices with $\mathcal{O}(1)$ nonzero entries or low-rank $\{A_i\}$, computing O takes $\mathcal{O}(mn^2 + m^2n)$. After O is formed, its factorization takes $\Theta(m^3)$, because it is dense.

Slater's condition for SDP For the SDP in (3.14) the Slater's condition can be stated as (Ramana et al., 1997):

$$\exists K \in \mathcal{S}_n \text{ such that } \mathsf{A}(K) = \boldsymbol{b} \text{ and } K \succ 0.$$

We use $K \succ 0$ to show the analogy with strict feasibility. We see that $K \in \mathcal{S}^n_{++}$, so that K must be full-rank. If Slater's condition is not satisfied, there is no guarantee that the HKM-based SDP solvers will succeed.

A face of the semidefinite cone Here, we present and solve a problem called Two Disjoint Cliques that is an example of semidefinite facial reduction defined as follows: Assume that for n unknown points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^r$, instead of having the coordinates, we posses only their partial (incomplete) Euclidean distance matrix:

$$D = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} n_1 & n_2 \end{array} \\ \begin{bmatrix} D_1 & \cdot \\ \cdot & D_2 \end{bmatrix} \end{array}, \quad D_\ell \in \mathcal{E}^{n_\ell},\ \ell = 1, 2,$$

where only the block diagonal elements are known (and shown). Imagine a graph $G = (\mathcal{V}, \mathcal{E})$, whose nodes are indices of the points, $\mathcal{V} = \{1, \ldots, n\}$, and its edge set is defined as

$$(i, j) \in \mathcal{E} \quad \text{if } D_{ij} \text{ is specified.}$$

It is straightforward to see that G has two *disjoint* cliques: $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $\mathcal{C}_1 \cup \mathcal{C}_2 = 1{:}n$ and $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. The first n_1 points are in $\mathcal{C}_1 = \{1, \ldots, n_1\}$, and the last n_2 points are in $\mathcal{C}_2 = \{n - n_2 + 1, \ldots, n\}$. Moreover, for $\ell = 1, 2$, all of elements of $D_\ell = D[\mathcal{C}_\ell]$ are known. We now show how to compute the face of $\{K \in \mathcal{S}^n_+ : \mathsf{K}(K[\mathcal{C}_\ell]) = D_\ell,\ \ell = 1, 2\}$.

As noted, formulating this problem in terms of the Gram matrix, $K \in \mathcal{S}^n_+$, eases handling. For $\ell = 1, 2$, let $r_\ell = \mathbf{embdim}(D_\ell)$; since we have D_ℓ, we can use MDS to find the corresponding embeddings, $P_\ell = [\boldsymbol{p}^{(\ell)}_1, \ldots, \boldsymbol{p}^{(\ell)}_{n_\ell}]$, $\boldsymbol{p}^{(\ell)}_i \in \mathbb{R}^{r_\ell}$ such that:

$$\|\boldsymbol{p}^{(\ell)}_i - \boldsymbol{p}^{(\ell)}_j\|^2 = (D_\ell)_{ij}, \quad \forall i, j \in \mathcal{C}_\ell,\ \ell = 1, 2.$$

Let $X = [X_1\ X_2]$ such that $X_1 = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_1}]$ and $X_2 = [\boldsymbol{x}_{n-n_2+1}, \ldots, \boldsymbol{x}_n]$, where

$\boldsymbol{x}_i \in \mathbb{R}^n$, $i \in \{1, \ldots, n\}$. Since X_1 and X_2 are rigid patches of points, we write:

$$X_\ell = R_\ell P_\ell + \boldsymbol{t}_\ell \mathbf{1}_{n_\ell}^\top, \quad \ell = 1, 2, \tag{3.18}$$

where $R_\ell \in \mathbb{R}^{n \times r_\ell}$ is a rotation/reflection matrix, such that $R_\ell^\top R_\ell = I_{r_\ell}$ and $\boldsymbol{t}_\ell \in \mathbb{R}^n$ is the translation vector (Khajehpour et al., 2011). Transposing (3.18) and rewriting it we have

$$X_\ell^\top = P_\ell^\top R_\ell^\top + \mathbf{1}_{n_\ell} \boldsymbol{t}_\ell^\top = \begin{bmatrix} P_\ell^\top & \mathbf{1}_{n_\ell} \end{bmatrix} \begin{bmatrix} R_\ell^\top \\ \boldsymbol{t}_\ell^\top \end{bmatrix} = Y_\ell S_\ell, \quad \ell = 1, 2,$$

where $Y_\ell \in \mathbb{R}^{n_\ell \times r_\ell + 1}$ is a known and $S_\ell \in \mathbb{R}^{r_\ell + 1 \times n}$ is an unknown matrix. Now, we can rewrite $X^\top$ as:

$$X^\top = \begin{bmatrix} X_1^\top \\ X_2^\top \end{bmatrix} = \begin{bmatrix} Y_1 & 0 \\ 0 & Y_2 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = YS,$$

where again $Y \in \mathbb{R}^{n \times k}$ is a known and $S \in \mathbb{R}^{k \times n}$ is an unknown matrix with $k = r_1 + r_2 + 2$. Now, we can form the Gram matrix:

$$G = X^\top X = YZY^\top, \quad Z = (SS^\top) \in \mathcal{S}_+^k.$$

Finally, since it is likely that $Y^\top Y \neq I$, to satisfy the requirements of Proposition 1, we need to make columns of Y orthogonal, achieved by computing the QR decomposition of Y as:

$$Y = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad U^\top U = I_k,\ V^\top V = I_{n-k},\ R \in \mathbb{R}^{k \times n}.$$

Let $Q = \begin{bmatrix} U & V \end{bmatrix}$, then the Gram matrix is defined as

$$G = YZY^\top = Q \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} Q^\top,$$

where $B = (RZR^\top) \in \mathcal{S}_+^k$. Since all requirements of Proposition 1 have been met, we see that $F = U\mathcal{S}_+^k U^\top$ is a face of $\mathcal{S}_+^n$. If $k \ll n$, this result indicates that the set of all possible Gram matrices for D is a small face of the cone of semidefinite matrices, and is termed semidefinite facial reduction.

The same facial reduction can be achieved by finding a U such that $\mathbf{range}(U) = \mathbf{range}(G)$ (Krislock, 2010; Krislock and Wolkowicz, 2010).

3.3.2 Problem Formulation

The protein structure determination problem is formulated as an instance of the EDMC problem; and due to many advantages of working on the Gram matrix, the problem is solved using semidefinite programming.

In the EDMC problem, a partial distance partial EDM $D \in \mathcal{E}^n$ is given; only a subset of elements are known and there are some upper and/or lower bounds on some other elements. In the protein NMR structure determination, the ultimate objective is to complete the missing elements of D such that (i) it is a valid EDM, (ii) $\mathbf{embdim}(D) = 3$, because ultimately, we are looking for the actual coordinates of atoms, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^3$, and (iii) it satisfies the corresponding upper and lower bounds.

We consider the protein structure determination formulation in (3.1); noting that this optimization must be solely reformulated in terms of distances between atoms, $D_{ij} = (\boldsymbol{e}_i - \boldsymbol{e}_j)^\top K (\boldsymbol{e}_i - \boldsymbol{e}_j)$, where K is the corresponding Gram matrix. The general formulation is:

$$\begin{array}{ll} \text{minimize} & \langle C, K \rangle \\ \text{subject to} & \langle A_{ij}, K \rangle = E_{ij}, \ (i,j) \in \mathcal{E} \\ & \langle A_{ij}, K \rangle \leq U_{ij}, \ (i,j) \in \mathcal{U} \\ & \langle A_{ij}, K \rangle \geq L_{ij}, \ (i,j) \in \mathcal{L} \\ & K\mathbf{1}_n = \mathbf{0} \\ & \mathbf{rank}(K) = 3 \\ & K \in \mathcal{S}^n_+ \end{array} \tag{3.19}$$

where $A_{ij} = (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top$ and $\mathcal{E}, \mathcal{U}$, and $\mathcal{L}$ are the set of equality constraints and upper and lower bounds, respectively. Moreover, the *centering* constraint $K\mathbf{1}_n = \mathbf{0}$, ensures that the embedding of K is zero-mean. That is, if $K = X^\top X$, $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{x}_i \in \mathbb{R}^p$, then $\sum \boldsymbol{x}_i = \mathbf{0}$. The optimization problem in (3.19) can also be cast as an instance of *feasibility* problem, with the goal being to find a $K \in \mathcal{S}^n_+$ that satisfies all of the constraints.

Rank constraint is NP-hard The optimization problem in (3.19) is not convex due to the rank constraint ($\mathbf{rank}(K) = 3$). In fact, adding the rank constraint makes this problem NP-complete. For example, Saxe (1979) showed that the *graph embeddability* problem (a problem similar to EDMC), with integer edge weights in one and two dimensions, is NP-complete. They proved the NP-completeness by reduction from the Partition problem, which is one of the 21 fundamental NP-complete problems introduced by Karp (1972).

Another explanation can be given by reducing the Max-Cut problem, which, again, is one of the most famous NP-complete problems (Karp, 1972). Let $G = (\mathcal{V}, \mathcal{E})$ be a graph

with the corresponding edge weights W such that $W_{ij} = 0$ if $(i, j) \notin \mathcal{E}$. A *cut* is defined as a partitioning of the $n = |\mathcal{V}|$ nodes into two disjoint sets, $\mathcal{V}_1$ and $\mathcal{V}_2 = \mathcal{V} \setminus \mathcal{V}_1$. The weight of the cut, w_c is defined as the sum of the weights of the edges crossing the cut, i.e.,:

$$w_c = \sum_{i \in \mathcal{V}_1, j \in \mathcal{V}_2} W_{ij}.$$

The MAX-CUT problem is defined as follows: Given $G = (\mathcal{V}, \mathcal{E})$ and W, determine if there is a cut such that $w_c \geq k$? We can solve MAX-CUT by solving the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \frac{1}{4} \sum W_{ij}(1 - x_i x_j) \\ \text{subject to} & x_i \in \{-1, 1\}, \quad i = 1, \ldots, n, \end{array} \tag{3.20}$$

where $x_i, i = 1, \ldots, n$ are indicator variables such that $x_i = 1$ and $x_i = -1$ indicate that node i belongs to $\mathcal{V}_1$ and $\mathcal{V}_2$, respectively. After solving the optimization problem in (3.20) and finding its optimal value w^*, we can check whether $k \geq w^*$ or not, in polynomial time.

In the matrix form, if we define $\boldsymbol{x} = [x_1, \ldots, x_n]$ and the graph Laplacian as $L := \mathbf{Diag}(W\mathbf{1}_n) - W$, the optimization problem in (3.20) can be rewritten as

$$\begin{array}{ll} \text{maximize} & \boldsymbol{x}^\top L \boldsymbol{x} \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n, \end{array}$$

which can be formulated as an instance of the SDP problem as

$$\begin{array}{ll} \text{minimize} & -\langle L, K \rangle \\ \text{subject to} & K_{ii} = 1, \ i = 1, \ldots, n \\ & \mathbf{rank}(K) = 1 \\ & K \in \mathcal{S}_+^n, \end{array}$$

because $\mathbf{rank}(K) = 1$ and $K \in \mathcal{S}_+^n$, imply that $K = \boldsymbol{x}\boldsymbol{x}^\top$ or $K_{ij} = x_i x_j$, for some $\boldsymbol{x} \in \mathbb{R}^n$. Last, it has been shown that even the *rank minimization* problem is NP-hard (Vandenberghe and Boyd, 1996).

Rank constraint relaxation The rank-constraint EDMC problem is non-convex and NP-hard. As a result, the rank constraint is relaxed, producing a convex optimization

problem as follows:

$$\begin{array}{ll} \text{minimize} & \langle C, K \rangle \\ \text{subject to} & \langle A_{ij}, K \rangle = E_{ij},\ (i,j) \in \mathcal{E} \\ & \langle A_{ij}, K \rangle \leq U_{ij},\ (i,j) \in \mathcal{U} \\ & \langle A_{ij}, K \rangle \geq L_{ij},\ (i,j) \in \mathcal{L} \\ & K \in \mathcal{S}_+^n, \end{array} \tag{3.21}$$

where the rank constraint is removed. To find X, compute the eigendecomposition of K as:

$$K = V \Lambda V^\top, \quad V \in \mathbb{R}^{n \times p},\ V^\top V = I_p,\ \Lambda \in \mathcal{S}_{++}^p,$$

where $\Lambda = \mathbf{Diag}(\lambda_1, \ldots, \lambda_p)$ is a diagonal matrix holding the eigenvalues of K, such that $\lambda_1 \geq \cdots \geq \lambda_p > 0$. The embedding is found as $X = \Lambda^{1/2} V^\top$.

Since the rank constraint is removed, there is no guarantee that the embedding dimension, p, is the same as the required embedding dimension, r. It is likely that $p \geq r$, in which case we need to project the embedding, X, onto the $\mathbb{R}^r$. We discuss this process in the post-processing section (page 119).

Rank reduction heuristics Rank of the Gram matrix determines the embedding dimension. Moreover, as we will demonstrate the projection error depends on the eigenvalues of the Gram matrix. In positive semidefinite matrices, all of the eigenvalues are non-negative; also note that for any square matrix, its trace is equal to the sum of its eigenvalues. Therefore, for a Gram matrix we have:

$$\mathbf{trace}(K) = \sum_{i=1}^{p} \lambda_i = \|\boldsymbol{\lambda}\|_1, \quad K \in \mathcal{S}_+^n,$$

where for $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_1 = \sum_{i=1}^n |x_i|$ is its ℓ_1-norm, $\mathbf{rank}(K) = p$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_p]^\top \in \mathbb{R}^p$ is a vector holding eigenvalues of K. The fact that minimizing the ℓ_1-norm tends to find *sparse* solutions, is a widely known and used heuristic (Boyd and Vandenberghe, 2004). Since $\mathbf{trace}(K)$ is the ℓ_1-norm of its eigenvalues, it appears that minimizing $\mathbf{trace}(K)$ produces a sparse $\boldsymbol{\lambda}$ vector, where most of its elements are zero, i.e., K is low-rank. This is also corroborated by the *nuclear norm* rank reduction heuristic that has received a great deal of attention in the past few years, especially for solving the *matrix completion* problem (Fazel, 2002; Recht et al., 2008; Candès and Recht, 2009; Recht et al., 2010; Candes and Plan, 2010; Ames and Vavasis, 2011).

For a matrix $A \in \mathbb{R}^{m\times n}$, the nuclear norm is defined as:

$$\|A\|_* = \sum_{i=1}^{p} \sigma_i,$$

where $\mathbf{rank}(A) = p$ and $\sigma_i > 0$ is the i-th largest singular value of A.We see that for $K \in \mathcal{S}_+^n$, we have $\|K\|_* = \sum_{i=1}^{p} \lambda_i = \mathbf{trace}(K)$. Since minimizing the nuclear norm tends to produce low-rank solutions, minimizing $\mathbf{trace}(K)$ is supposed to minimize $\mathbf{rank}(K)$ and consequently, the embedding dimension. However, it is notable that, in the EDMC problem, it has the exact opposite effect and, produces high-rank solutions instead, for example, see (Göring et al., 2011). This is an example that demonstrates that heuristic approaches do not work in every situation. We provide an informal explanation for this observation in the following.

The sum of all of elements of the EDM D can be written as:

$$\begin{aligned}\textstyle\sum_{i,j=1}^{n} D_{ij} &= \mathbf{1}^\top D \mathbf{1} \\ &= \mathbf{1}^\top (\mathbf{diag}(K)\mathbf{1}^\top - 2K + \mathbf{1}\mathbf{diag}(K)^\top)\mathbf{1} \\ &= 2(\mathbf{1}^\top \mathbf{diag}(K))(\mathbf{1}^\top \mathbf{1}) \\ &= 2n\, \mathbf{trace}(K),\end{aligned}$$

where we used the fact that $\mathbf{1}^\top K \mathbf{1} = 0$, because the Gram matrix is centered, i.e., $K\mathbf{1} = \mathbf{0}$. So, $\mathbf{trace}(K)$ is linearly proportional to the total distances between all points. However, K_{ii} is the distance of $\boldsymbol{x}_i$ from the origin and $\mathbf{trace}(K)$ is the sum of distances of all points from the origin. Therefore, minimizing the trace is equivalent to pushing all the points towards the origin and forcing them to be as close to each other as possible. Imagine that all points lie on a plane in $\mathbb{R}^2$ in the ideal solution; minimizing the trace pushes all of the points close together, analogous to crumpling a piece of paper and making it three-dimensional.

It is surprising that the counter-intuitive idea of maximizing $\mathbf{trace}(K)$, proved to be one of the most effective yet simplest heuristic techniques for rank reduction. Weinberger and Saul (2004) proposed maximizing the trace of the Gram matrix for unfolding manifolds as an effective method for dimensionality reduction. Their proposed method is called Maximum Variance Unfolding (MVU), because the eigenvalues of the centered Gram matrix and the *covariance matrix* of the points, $C \in \mathcal{S}_+^p$, are essentially the same:

$$K = X^\top X, \quad C = XX^\top, \quad \text{given} \;\; X\mathbf{1} = \mathbf{0},$$

Therefore, maximizing the trace is equivalent to maximizing the variance of the data. MVU received much attention and originated, or influenced, several similar approaches

(Weinberger and Saul, 2006; Shaw and Jebara, 2007; Weinberger et al., 2007; Shaw and Jebara, 2009). It was also successfully applied to the sensor network localization (Biswas et al., 2006) and molecular conformation (Biswas et al., 2008; Leung and Toh, 2009) problems.

An upper bound on the diameter of the structure For a set of points, the *diameter* is defined as the distance between farthest points in the set. To find an upper bound for the diameter and the sum of all distances, we reuse the idea of building a graph for the EDMC problem: Imagine a graph $G = (\mathcal{V}, \mathcal{H})$, whose nodes are indices of the points, $\mathcal{V} = \{1, \ldots, n\}$, and its edge set is defined as:

$$(i,j) \in \mathcal{H} \quad \text{if } D_{ij} \text{ is specified, or upper}-\text{bounded.}$$

Since we want to build an upper bound graph, ignoring the lower bounds, we further assume that the weight associated for the edge (i, j) is:

$$W_{ij} = \begin{cases} \sqrt{E_{ij}} & \text{if} \quad (i,j) \in \mathcal{E} \\ \sqrt{U_{ij}} & \text{if} \quad (i,j) \in \mathcal{U} \\ 0 & \text{if} \quad i = j \\ \infty & \text{o.w.} \end{cases}$$

Note that G must be connected; otherwise, we can divide the problem into two or more smaller problems (connected components of G). Let $\Delta \in \mathcal{S}^n$ be the matrix containing the squared *geodesic* distances between nodes, such that Δ_{ij} contains the squared weight of the path with least total weight from node i to j, which can be computed using Dijkstra's algorithm for example. Due to the triangle inequality, for any two points we have $\Delta_{ij} \geq D_{ij}$, where $D \in \mathcal{E}^n$ is the actual EDM. We conclude that $\delta_{\max} = \max(\Delta)$ is an upper bound on the diameter of the set of nodes in G. Moreover, if $\bar{\delta} = 1/n^2 \sum \Delta_{ij}$ is the average of Δ, then for any feasible solution we have:

$$\sum_{i,j=1}^{n} D_{ij} = 2n\, \mathbf{trace}(K) \leq n^2 \bar{\delta} \;\Rightarrow\; \mathbf{trace}(K) \leq \frac{n}{2}\bar{\delta}$$

In addition, since the distance of any point from the origin is less than the diameter, we have $K_{ii} \leq \delta_{\max}, i = 1, \ldots, n$.

Upper bounds

NOE-derived upper bounds are the most important source of structural information in the protein NMR. From the NOE cross peak assignments, hydrogen bonds, and disulfide and

salt bridges, we have upper bounds of this form:

$$d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq u_{ij}.$$

These upper bounds can easily be reformulated as constraints in (3.19) in terms of the Gram matrix:

$$D_{ij} = \langle A_{ij}, K \rangle \leq U_{ij} = u_{ij}^2.$$

However, these upper bounds could be inaccurate or even completely incorrect. Therefore, we need to allow violations, but must penalize them. In SDP solvers, inequalities are implemented by using some positive *unpenalized slacks*:

$$\langle A_{ij}, K \rangle + \xi_{ij} = U_{ij}, \quad \xi_{ij} \in \mathbb{R}_+,$$

these slacks are unpenalized because they do not appear in the objective function. We add penalized slacks and rewrite the upper bounds as:

$$\langle A_{ij}, K \rangle + \xi_{ij} = U_{ij} + \zeta_{ij}, \quad \xi_{ij}, \zeta_{ij} \in \mathbb{R}_+;$$

the objective function is changed to $\langle C, K \rangle + \gamma \sum_{ij} w_{ij} \zeta_{ij}$, where $\gamma > 0$ is the scaling factor. Since we use a linear penalty function for upper bound violations, incorrect bounds cannot "lock" the structure (preventing correct folding), or worse, make the optimization problem infeasible. In addition, looking closely at the penalized slacks we see that $\zeta_{ij} = \max(D_{ij} - U_{ij}, 0)$.

Let $\boldsymbol{\zeta} \in \mathbb{R}^{|\mathcal{U}|}$ be a vector containing all of the slacks. Since $\zeta_{ij} \in \mathbb{R}_+$, assuming that all the weights are the same, i.e., $w_{ij} = w$, we have

$$w \sum_{ij} \zeta_{ij} = w \, \|\boldsymbol{\zeta}\|_1.$$

In our problem, $\zeta_{ij} = 0$ implies no violation; consequently, SPROS tends to find a solution that violates a minimum number of upper bounds. In contrast, minimizing the ℓ_2-norm, that is $\sum_{ij} \zeta_{ij}^2 = \|\boldsymbol{\zeta}\|^2$, tends to find *dense* solutions. If only a few upper bounds are incorrect and the ℓ_2-norm of violations are minimized, to compensate for the large penalties imposed by the wrong bounds, many correct bounds will be slightly violated. If the number of incorrect bounds is relatively large, they can completely lock the structure and prevent correct folding.

Torsion Angle Restraints

Another source of structural information in protein NMR is the set of torsion angle restraints defined as $\boldsymbol{\theta}^{\max} = [\theta_1^{\max}, \ldots, \theta_t^{\max}]^\top$, $\boldsymbol{\theta}^{\min} = [\theta_1^{\min}, \ldots, \theta_t^{\min}]^\top$, such that $\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}$, $i \in \mathcal{T}$. In EDMC, we only can constrain the distances between atoms meaning that the sign of the torsion angle cannot be detected. That is, given four atoms with torsion angle θ, if we find the reflection of the atoms, their distance matrix does not change while their torsion angle becomes $-\theta$. However, we can constrain the magnitude of the torsion angles.

In the CORELS method (Sussman, 1985), the distance between the first and the fourth atom in a dihedral angle is used for fixing the torsion angles (see Fig. 3.6). We extend this idea and define upper and lower bounds on the torsion angles based on the distance between the first and the fourth atom in the torsion angle. For constraining the Φ angle (made of C_{i-1}, N_i, CA_i, and C_i), we constrain $\|\mathrm{C}_i - \mathrm{C}_{i-1}\|$. Similarly, for constraining the Ψ angle (made of N_i, CA_i, C_i, and N_{i+1}), we constrain $\|\mathrm{N}_i - \mathrm{N}_{i+1}\|$. The $\|\mathrm{C}_i - \mathrm{C}_{i-1}\|$ VS. Φ and the $\|\mathrm{N}_i - \mathrm{N}_{i-1}\|$ VS. Ψ curves are depicted in Figs. 3.8 and 3.9, respectively; it can be seen that both are even functions.

Given a range for Φ as $\Phi_{\min} = -120^\circ$ and $\Phi_{\max} = -60^\circ$, we observe the curve and see that the distances corresponding to these two values are 3.5 Å and 3 Å, respectively, as depicted in Fig. 3.8. In general, if $\theta_{\min} \times \theta_{\max} \geq 0$, i.e., both of the bounds are on the same half plane, we have one upper and one lower bound for each torsion angle restraint. If not, we only have one upper bound. To facilitate computing the bounds, we computed very close approximations for the actual distance curves (see Figs. 3.8 and 3.9):

$$\begin{aligned}
\|\mathrm{C}_i - \mathrm{C}_{i-1}\| &\approx 3.221 - 0.4866\,\cos(1.044\Phi), \quad -\pi \leq \Phi \leq +\pi, \\
\|\mathrm{N}_i - \mathrm{N}_{i+1}\| &\approx 3.157 - 0.5022\,\cos(1.046\Psi), \quad -\pi \leq \Psi \leq +\pi,
\end{aligned}$$

Similar to the upper bounds, these constraints are not always accurate. They might be slightly inaccurate or, in some cases, completely incorrect. Therefore, we add slacks for both the upper and lower bounds, but we penalize them more severely, because their range of violations is much smaller than those found NOE upper bounds (1 Å VS. 15-25 Å).

Stereochemical constraints

Proteins exhibit particular structural features that make them different from other types of molecules. Therefore, atoms in the protein molecule are not *analogous* to sensors in the sensor network localization (SNL) problem. No domain knowledge exists in SNL, any embedding of sensors that satisfies the given spatial constraints is acceptable and valid, which is not the case for proteins.

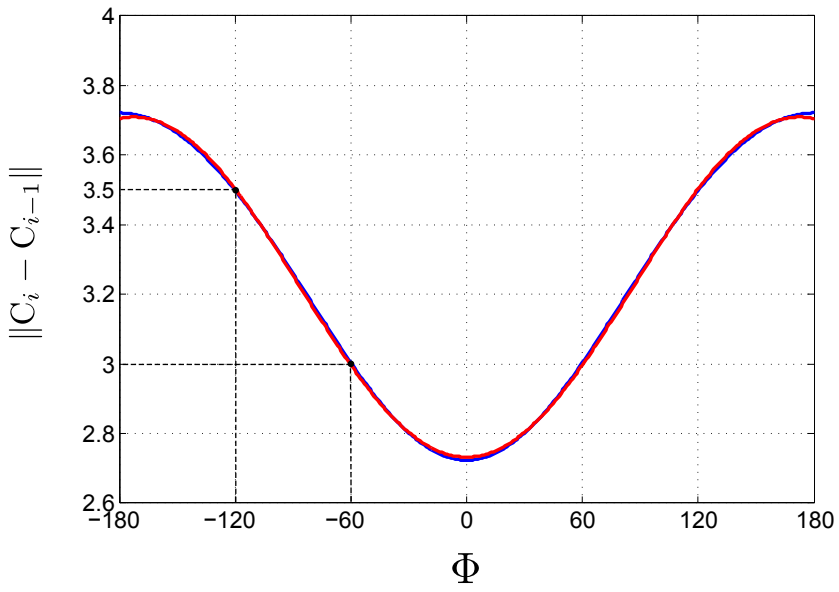


Figure 3.8: The blue curve is the actual distance, computed for two consecutive alanine residues; the red line is the approximation that is almost identical to the actual curve. Also the corresponding lower and upper distance bounds are shown for $\Phi_{\min} = -120°$ and $\Phi_{\max} = -60°$.

Recall that, the set of stereochemical constraints are added to the protein structure determination problem to enforce the validity of the protein. These constraints include: bond lengths and angles, planarity of coplanar atoms, steric constraints, and chirality constraints.

Bonds length and angle constraints are written in terms of the distance between an atom and its immediate neighbor and an atom and its second nearest neighbor, respectively (see Fig. 3.6). For preserving the planarity of the coplanar atoms, three bonded non-collinear atoms from the plane are randomly chosen and the distances between all other atoms in the plane and the selected atoms are fixed. That is, the plane is treated as if it is a rigid clique.

Steric constraints can be written as lower bounds on the distances between any two non-bonded atoms, which are not in the same clique. However, chirality constraints cannot be formulated in terms of the distances between atoms. Therefore, the chiralities are verified after the EDMC problem is solved and corrections are made in the post-processing stage.

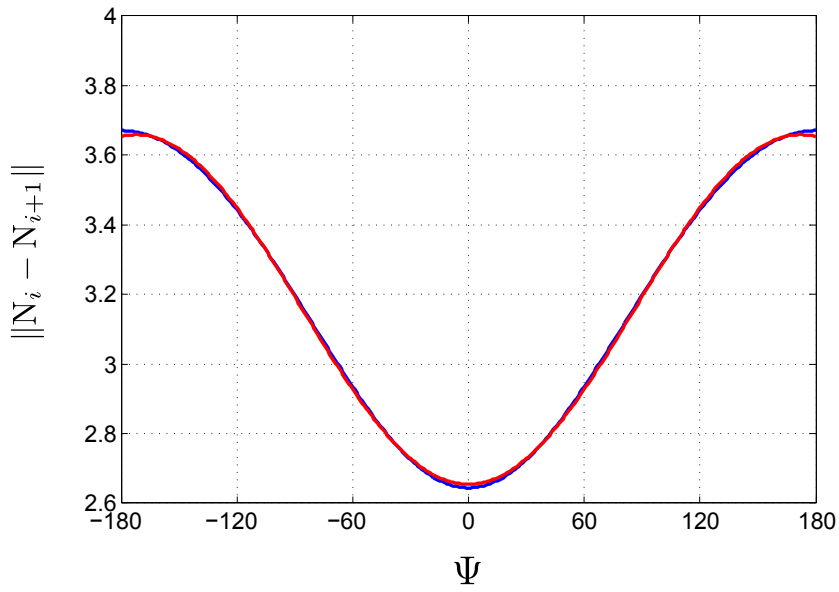


Figure 3.9: The blue curve is the actual distance, computed for two consecutive alanine residues; the red line is the approximation that is almost identical to the actual curve.

Preliminary Problem Formulation

At this point, all the constraints are formed and we can solve the EDMC problem in (3.21). The constraints are:

1. Equality constraints are the union of bond lengths (B), bond angles (A), and planarity (P) equality constraints:

$$\mathcal{E} = \mathcal{E}_B \cup \mathcal{E}_A \cup \mathcal{E}_P.$$

2. Upper bounds are the union of NOE-derived (N), hydrogen bonds (H), disulfide and salt bridges (D), and torsion angle (T) upper bounds:

$$\mathcal{U} = \mathcal{U}_N \cup \mathcal{U}_H \cup \mathcal{U}_D \cup \mathcal{U}_T.$$

3. Lower bounds are the union of steric or van der Waals (W) and torsion angle lower bounds:

$$\mathcal{L} = \mathcal{L}_W \cup \mathcal{L}_T.$$

After forming these sets, we ensure that $\mathcal{E} \cap \mathcal{U} = \mathcal{E} \cap \mathcal{L} = \emptyset$. It is possible that the set of upper and lower bounds are not disjoint, for example, when van der Waals constraints are added.

The preliminary problem formulation is given by:

$$\begin{array}{lll} \text{minimize} & -\gamma\langle I, K\rangle + \sum_{ij} W_{ij}\zeta_{ij} & \\ \text{subject to} & \langle A_{ij}, K\rangle = E_{ij}, & (i,j) \in \mathcal{E} \\ & \langle A_{ij}, K\rangle + \xi_{ij} = U_{ij} + \zeta_{ij}, & (i,j) \in \mathcal{U} \\ & \langle A_{ij}, K\rangle + \zeta_{ij} = L_{ij} + \xi_{ij}, & (i,j) \in \mathcal{L} \\ & \zeta_{ij}, \xi_{ij} \in \mathbb{R}_+, & (i,j) \in \mathcal{L} \cup \mathcal{U} \\ & K\mathbf{1}_n = \mathbf{0} & \\ & K \in \mathcal{S}^n_+, & \end{array} \tag{3.22}$$

The rank reduction heuristic term, $-\gamma\langle I, K\rangle, \gamma > 0$, in the objective function, produces lower-rank solutions. In addition, the total number of constraints is $m = |\mathcal{E}| + |\mathcal{U}| + |\mathcal{L}|$.

Solving the optimization problem in (3.22) is challenging and the reason is two fold: (i) for small to medium size proteins, the number of atoms, n, is 1,000-3,500. Current HKM-based SDP solvers cannot solve problems with $n > 2,000$ efficiently; and (ii) the optimization problem in (3.22) does not satisfy Slater's condition. As we have shown in the EDMC example, with just two cliques (see page 85), whenever there are cliques in the EDM, the Gram matrix is rank deficient; that is $\mathbf{rank}(K) < n$. Yet, it is notable that proteins are practically made of cliques, as we will discuss.

To overcome the limitations of the SDP solvers, (Biswas et al., 2008) proposed dividing the protein molecule into smaller overlapping sub-structures, solving the smaller and tractable problem, and then stitching back together the solved sub-structures. Later, (Leung and Toh, 2009) improved upon this idea by presenting more efficient dividing and stitching techniques. Both of these approaches treat bond lengths and bond angles unrealistically as loose lower and upper bounds, and consequently, they do not face rank deficiency of the Gram matrix. Moreover, there is no guarantee that the non-convex stitching method can find the optimal alignment of the sub-structures.

3.3.3 Cliques in a Protein Molecule

A protein molecule can be divided into several cliques. For example, peptide planes, or aromatic rings, are 2D cliques or tetrahedral carbons form 3D cliques. A protein with ℓ amino acid residues has $\ell + 1$ planes in the backbone: $\ell - 1$ full peptide planes and two half-planes at the N- and C-terminals:

$$
\begin{aligned}
&\mathcal{P}_0 = \{\mathrm{N}_1, \mathrm{HN}_1, \mathrm{CA}_1\},\\
&\mathcal{P}_i = \{\mathrm{CA}_i, \mathrm{C}_i, \mathrm{O}_i, \mathrm{N}_{i+1}, \mathrm{HN}_{i+1}, \mathrm{CA}_{i+1}\}, \quad 1 \le i < \ell,\\
&\mathcal{P}_\ell = \{\mathrm{CA}_\ell, \mathrm{C}_\ell, \mathrm{O}_\ell\}.
\end{aligned}
$$

For proline residues, HN is simply replaced with CD from their side chains.

Each amino acid has a different side chain with a different structure; therefore, the number of cliques in each side chain varies (see Table 3.1 for the number of cliques in each amino acids side chain). We assume that the i-th residue, r_i, has s_i cliques in its side chain, denoted by $\mathcal{S}_i^{(1)}, \ldots, \mathcal{S}_i^{(s_i)}$. For all amino acids (except glycine and proline), the first side chain clique is formed by the tetrahedral carbon CA:

$$
\mathcal{S}_i^{(1)} = \{\mathrm{N}_i, \mathrm{CA}_i, \mathrm{HA}_i, \mathrm{CB}_i, \mathrm{C}_i\},
$$

which intersects with $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ in two atoms:

$$
\begin{aligned}
\mathcal{S}_i^{(1)} \cap \mathcal{P}_{i-1} &= \{\mathrm{N}_i, \mathrm{CA}_i\},\\
\mathcal{S}_i^{(1)} \cap \mathcal{P}_i \quad &= \{\mathrm{CA}_i, \mathrm{C}_i\}.
\end{aligned}
$$

Side chain cliques for all twenty amino acids are listed in Table 3.1; see Figs. 3.10 and 3.11 for atom nomenclature.

There is a total of $q = \ell + 1 + \sum_{i=1}^{\ell} s_i$ cliques in the distance matrix of any protein. To simplify, let $\mathcal{C}_i = \mathcal{P}_{i-1}$, $1 \le i \le \ell + 1$, and $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(1)}$, $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(2)}$, $\ldots$, $\mathcal{C}_q = \mathcal{S}_\ell^{(s_\ell)}$. In addition, since for cliques $D[\mathcal{C}_i]$ is entirely known, for $i = 0, \ldots, q$, let $r_i = \mathbf{embdim}(D[\mathcal{C}_i])$. The following properties hold:

1. $\mathcal{P}_i \cap \mathcal{P}_{i'} = \emptyset$, given $|i - i'| > 1$.
2. $\mathcal{P}_i \cap \mathcal{S}_{i'}^{(j)} = \emptyset$, given $i' \neq i, i+1$.
3. $\mathcal{S}_i^{(j)} \cap \mathcal{S}_{i'}^{(j')} = \emptyset$, given $i' \neq i$.
4. $|\mathcal{C}_i| \ge r_i + 1$.
5. $3 \le |\mathcal{C}_i| \le 16$.
6. $\forall i, i'$, $|\mathcal{C}_i \cap \mathcal{C}_{i'}| \le 2$.
7. $\nexists i$ such that $\forall i' \neq i$, $\mathcal{C}_i \cap \mathcal{C}_{i'} = \emptyset$.

Table 3.1: Cliques in the side chains of amino acids. If $\mathcal{S}^{(1)}$ is not listed, it is the same as Ala; if $\mathcal{S}^{(i)}, 2 \leq i < s$ (s is the number of cliques in the side chain) is not listed, it is the same as Lys. 2D cliques are marked by an '$*$'.

A.A.	s	Side Chain Cliques
Ala	2	$\mathcal{S}^{(1)} = \{\text{N, CA, HA, CB, C}\}$ $\mathcal{S}^{(2)} = \{\text{CA, CB, QB, HB1, HB2, HB3}\}$
Arg	5	$\mathcal{S}^{(4)} = \{\text{CG, CD, HD2, HD3, QD, NE}\}$ $\mathcal{S}^{(5)} = \{\text{CD, CE, HE, CZ, NH1, HH11, HH12}\}*$
Asn	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, OD1, ND2, HD21, HD22, QD2}\}*$
Asp	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, OD1, OD2}\}*$
Cys	3	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB2, HB3, QB, SG}\}$ $\mathcal{S}^{(3)} = \{\text{CB, SG, HG}\}$
Glu	4	$\mathcal{S}^{(4)} = \{\text{CG, CD, OE1, OE2}\}*$
Gln	4	$\mathcal{S}^{(4)} = \{\text{CG, CD, OE1, NE2, HE21, HE22, QE2}\}*$
Gly	1	$\mathcal{S}^{(1)} = \{\text{N, CA, HA2, HA3, QA, C}\}$
His	1	$\mathcal{S}^{(3)} = \{\text{CB, CG, ND1, HD1, CD2, HD2, CE1, HE1, NE2}\}*$
Ile	5	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, CG1, CG2}\}$ $\mathcal{S}^{(3)} = \{\text{CB, CG2, HG21, HG22, HG23, QG2}\}$ $\mathcal{S}^{(4)} = \{\text{CB, CG1, HG12, HG13, QG1, CD1}\}$ $\mathcal{S}^{(5)} = \{\text{CG1, CD1, HD11, HD12, HD13, QD1}\}$
Leu	5	$\mathcal{S}^{(3)} = \{\text{CB, CG, HG, CD1, CD2, QQD}\}$ $\mathcal{S}^{(4)} = \{\text{CG, CD1, HD11, HD12, HD13, QD1}\}$ $\mathcal{S}^{(5)} = \{\text{CG, CD2, HD21, HD22, HD23, QD2}\}$
Lys	6	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB2, HB3, QB, CG}\}$ $\mathcal{S}^{(3)} = \{\text{CB, CG, HG2, HG3, QG, CD}\}$ $\mathcal{S}^{(4)} = \{\text{CG, CD, HD2, HD3, QD, CE}\}$ $\mathcal{S}^{(5)} = \{\text{CD, CE, HE2, HE3, QE, NZ}\}$ $\mathcal{S}^{(6)} = \{\text{CE, NZ, HZ1, HZ2, HZ3, QZ}\}*$
Met	5	$\mathcal{S}^{(3)} = \{\text{CB, CG, HG2, HG3, QG, SD}\}$ $\mathcal{S}^{(4)} = \{\text{CG, SD, CE}\}*$ $\mathcal{S}^{(5)} = \{\text{SD, CE, HE1, HE2, HE3, QE}\}$
Phe	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, CD1, HD1, CE1, HE1, CZ, HZ, CE2, HE2, CD2, HD2, QD, QE, QR}\}*$
Pro	1	$\mathcal{S}^{(1)} = \{\text{N, CD, CA, HA, CB, HB2, HB3, QB, CG, HG2, HG3, QG, HD2, HD3, QD, C}\}$
Ser	3	$\mathcal{S}^{(3)} = \{\text{CB, OG, HG}\}*$
Thr	4	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, OG1, CG2}\}$ $\mathcal{S}^{(3)} = \{\text{CB, OG1, HG1}\}*$ $\mathcal{S}^{(4)} = \{\text{CB, CG2, HG21, HG22, HG23, QG2}\}$
Trp	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, CD1, HD1, CD2, CE2, CE3, HE3, NE1, HE1, CZ2, HZ2, CZ3, HZ3, CH2, HH2}\}*$
Tyr	4	$\mathcal{S}^{(3)} = \{\text{CB, CG, CD1, HD1, CE1, HE1, CE2, HE2, CD2, HD2, CZ, OH, QD, QE, QR}\}*$ $\mathcal{S}^{(4)} = \{\text{CZ, OH, HH}\}*$
Val	4	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, CG1, CG2, QQG}\}$ $\mathcal{S}^{(3)} = \{\text{CB, CG1, HG11, HG12, HG13, QG1}\}$ $\mathcal{S}^{(4)} = \{\text{CB, CG2, HG21, HG22, HG23, QG2}\}$

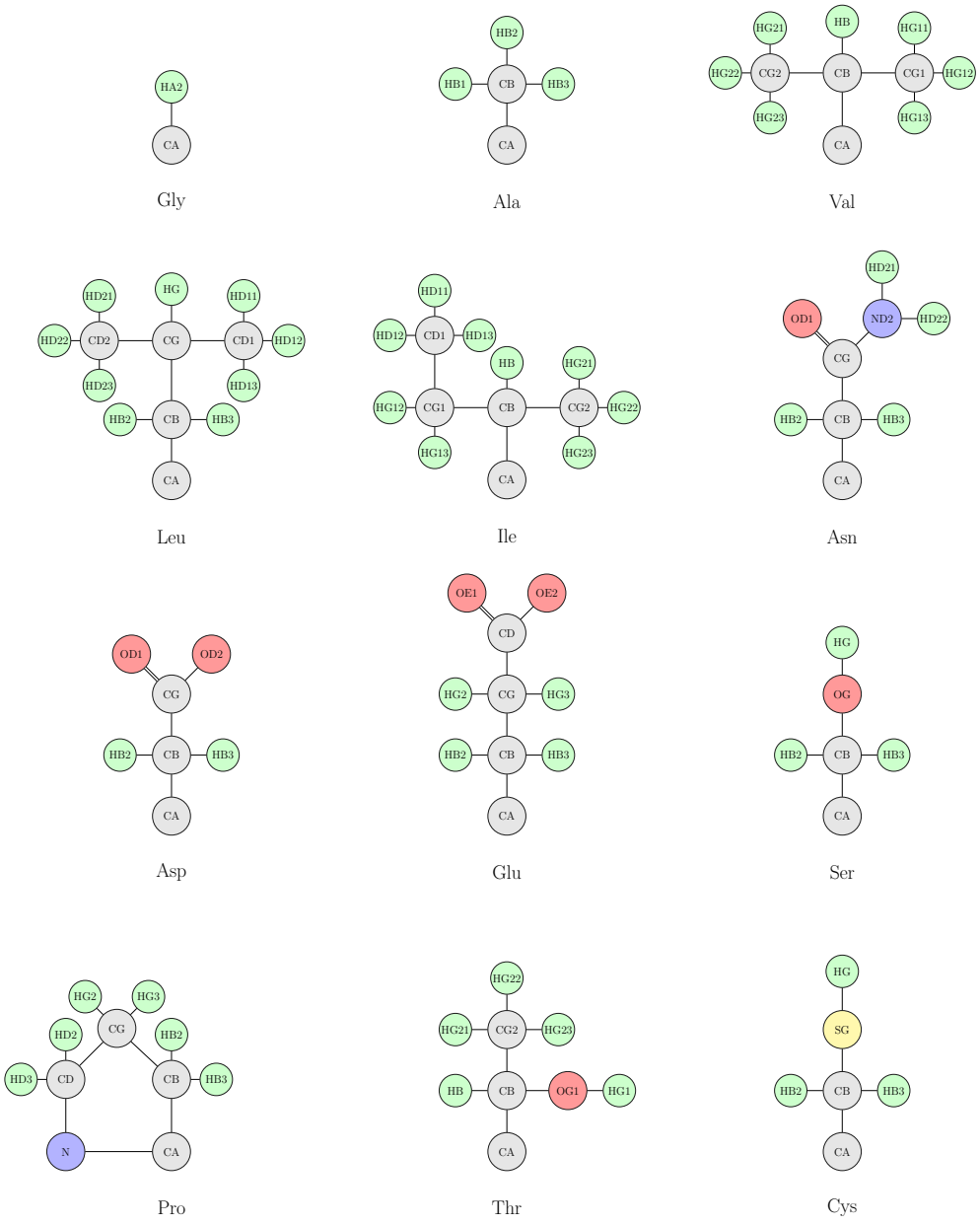


Figure 3.10: Side chains of amino acids (part I).

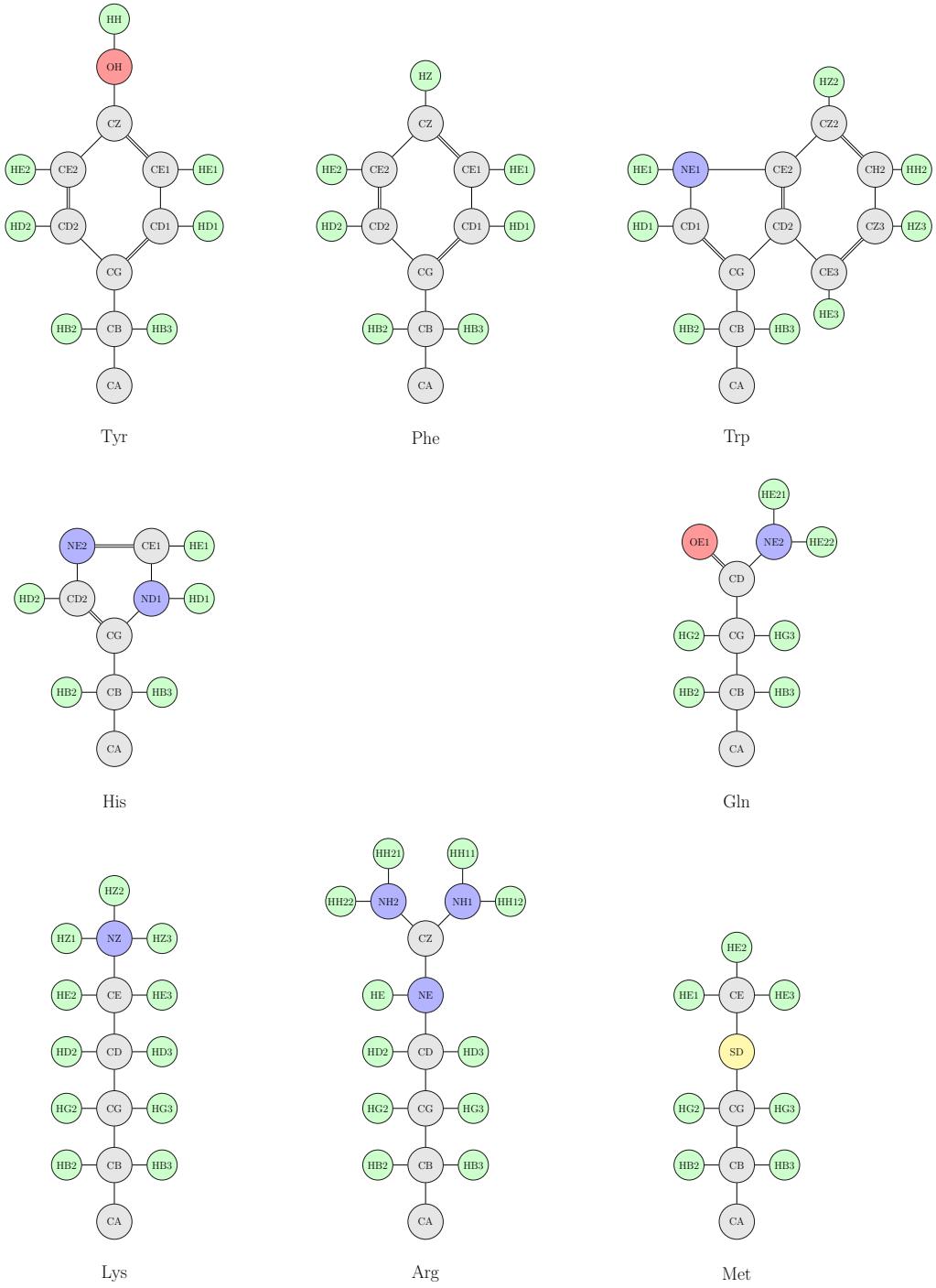


Figure 3.11: Side chains of amino acids (part II).

8. If $\mathcal{I}_i = \{i' : \mathcal{C}_i \cap \mathcal{C}_{i'} \neq \emptyset\}$, then $\forall i, |\mathcal{I}_i| \leq 4$.

9. $\bigcup_{i=1}^{q} \mathcal{C}_i = 1{:}n$.

It is clear that the protein molecule is entirely made up of cliques that if not treated properly, can cause problems (Slater's condition) and make solving the corresponding EDMC problem challenging.

Simplifying Side chains

In this section, considering the nature of the protein NMR problem, we propose a side-chain simplification scheme.

Pseudo-atoms The hydrogens in the methyl groups cannot be uniquely assigned; therefore, upper bounds involving these hydrogens are ambiguous (this is different from the ambiguity in assigning NOE peaks). To overcome this issue, *pseudo-atoms* are introduced (Güntert, 1998). They are called pseudo-atoms, because they do not really exist, rather they are virtual entities put there to facilitate handling ambiguous constraints. For example, consider the methyl group in the side chain of alanine depicted in Fig. 3.12; the pseudo atom is placed at the centroid of the hydrogens, along the CA-CB bond.

Given an ambiguous constraint between one of the hydrogens and atom A, by using the triangle inequality, we modify the constraint as follows:

$$\|\mathrm{HB}x - \mathrm{A}\| \leq b,\ x \in \{1,2,3\} \Rightarrow \|\mathrm{QB} - \mathrm{A}\| \leq b + \|\mathrm{HB}x - \mathrm{QB}\|,$$

where $\|\mathrm{HB}x - \mathrm{QB}\|$ is the same for $x = 1, 2, 3$. Pseudo-atoms are named corresponding to the hydrogens they represent; only H is changed to Q and the rightmost number is dropped. For example, in leucine, QD1 represents HD11, HD12, and HD13.

In addition to hydrogens in the methyl group, the hydrogens in the methylene groups, e.g., HB2 and HB3 in serine (see Fig. 3.10), are some times ambiguously assigned. A pseudo-atom is placed at the center of the line connecting the two hydrogens. Generally, the need for adding pseudo-atoms arises when there is symmetry in hydrogens. Asparagine, glutamine, and lysine each have one pseduo-atom in their side chain amino groups. Moreover, glycine has a pseudo-atom between its HA2 and HA3.

Leucine and valine have two similar and interesting pseudo-atoms: leucine has a pseudo-atoms, called QQD, placed between its CD1 and CD2 atoms and valine has a pseudo-atom placed between its CG1 and CG2 atoms, called QQG. These two pseudo-atoms are used if the *pro*-R and *pro*-S methyl groups are not assigned.

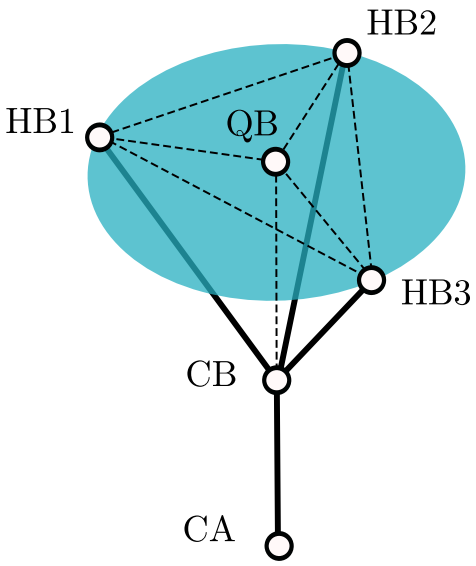


Figure 3.12: The pseudo-atom QB is shown at the centroid of the hydrogens, collinear with the CA-CB bond.

Furthermore, tyrosine and phenylalanine have a pseudo-atom, called QR, at the center of their phenolic rings. In addition, they have two more pseudo atoms, QD and QE, which are used if the hydrogens in the ring are not uniquely assigned.

Side chain simplification scheme Many hydrogens do not play an important role in structure prediction. For example, in methyl groups the C–H bond length is approximately 1.08 Å, while the van der Waals radii of carbon and hydrogen are 1.6 Å and 1.0 Å, respectively. This means that more than 75% of the volume of the hydrogen is in the carbon. In fact, hydrogens can be viewed as small bumps on the surfaces of larger atoms. This is why, in most of the protein structure determination programs, such as Cyana (Güntert, 2004), the hydrogens are discarded in the early stages of the program and are later on added.

We adapt this idea to our model of protein as a set of rigid cliques, we only discard hydrogens if they make our problem smaller. Note that we do not omit all the hydrogens. For example, the phenolic ring of tyrosine is a large two-dimensional clique and omitting the hydrogens in the ring does not proivde simplification. Omitting the hydroxyl hydrogen in tyrosine, however, removes $\mathcal{S}^{(4)} = \{\mathrm{CZ}, \mathrm{OH}, \mathrm{HH}\}$ from its side chain cliques.

In the side chain *simplification* process, we temporarily discard (i) all of the methyl

hydrogens, (*ii*) all of the methylene hydrogens, (*iii*) hydroxyl hydrogens of tyrosine and serine, (*iv*) amino hydrogens of arginine and threonine, and (*v*) sulfhydryl hydrogen of cysteine. This simplification results in a significant reduction in the problem size: the average number of side chain cliques is reduced from 3.6 to 2.7, but more importantly, the average number of three-dimensional cliques is reduced from 3.0 to 1.3, see Table 3.3 (page 116). The simplified side chain cliques are listed in Table 3.2. After the EDMC problem is solved, the omitted hydrogen atoms are put back in the protein and are present in the post-processing stage.

3.3.4 Computing the Face of Protein Molecules

Previously, we modeled the protein as a set of rigid intersecting cliques, similar to the core idea in torsion angle dynamics. We showed how to compute the face of a Gram matrix that contains two *disjoint* cliques (see page 85). While extending that method to intersecting cliques, may be intuitive, it poses challenges. Therefore, we follow a different path, that proposed by (Krislock and Wolkowicz, 2010).

The face of a cingle clique

Here, we solve the SINGLE CLIQUE problem, which is defined as follows: Let $D \in \mathcal{E}^n$ be a partial EDM, and $G = (\mathcal{V}, \mathcal{E})$ be its corresponding graph of specified distances. Suppose the first n_1 points form a clique in G, such that for $\mathcal{C}_1 = \{1, \ldots, n_1\}$, all distances are known. That is, $D_1 = D[\mathcal{C}_1]$, is completely specified. Moreover, let $r_1 = \mathbf{embdim}(D_1)$. We now show how to compute the face of the $\{K \in \mathcal{S}^n_+ : \mathsf{K}(K[\mathcal{C}_1]) = D_1\}$.

This is a simpler version of the TWO DISJOINT CLIQUES problem; however, we solve the problem using a different approach, beginning by stating the following theorem:

Theorem 4 (Krislock (2010)) *In the* SINGLE CLIQUE *problem, let* $U_1 \in \mathbb{R}^{n\times(n-n_1+r_1+1)}$ *be defined as follows:*

- *let* $V_1 \in \mathbb{R}^{n_1\times r_1}$ *be a full column rank matrix such that* $\mathbf{range}(V_1) = \mathbf{range}(\mathsf{K}^\dagger(D_1))$*;*
- *let* $\bar{U}_1 := \begin{bmatrix} V_1 & \mathbf{1} \end{bmatrix} \in \mathbb{R}^{n_1\times r_1+1}$*;*
- *let* $U := \begin{array}{c} \\ n_1 \\ n-n_1 \end{array} \begin{array}{c} \begin{array}{cc} r_1+1 & n-n_1 \end{array} \\ \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & I \end{bmatrix} \end{array} \in \mathbb{R}^{n\times(n-n_1+r_1+1)}$.

Table 3.2: Cliques in the simplified side chains of amino acids. If $\mathcal{S}^{(i)}, 2 \leq i < s'$ (s' is the number of cliques in the simplified side chain) is not listed, it is the same as Lys. 2D cliques are marked by an '$*$'.

A.A.	s'	Side Chain Cliques
Ala	1	$\mathcal{S}^{(1)} = \{\text{N}, \text{CA}, \text{HA}, \text{CB}, \text{QB}, \text{C}\}$
Arg	5	$\mathcal{S}^{(4)} = \{\text{CG}, \text{CD}, \text{NE}\}*$ $\mathcal{S}^{(5)} = \{\text{CD}, \text{CE}, \text{HE}, \text{CZ}, \text{NH1}, \text{HH11}, \text{HH12}\}*$
Asn	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{OD1}, \text{ND2}, \text{HD21}, \text{HD22}, \text{QD2}\}*$
Asp	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{OD1}, \text{OD2}\}*$
Cys	2	$\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{SG}\}*$
Glu	4	$\mathcal{S}^{(4)} = \{\text{CG}, \text{CD}, \text{OE1}, \text{OE2}\}*$
Gln	4	$\mathcal{S}^{(4)} = \{\text{CG}, \text{CD}, \text{OE1}, \text{NE2}, \text{HE21}, \text{HE22}, \text{QE2}\}*$
Gly	1	$\mathcal{S}^{(1)} = \{\text{N}, \text{CA}, \text{HA2}, \text{HA3}, \text{QA}, \text{C}\}$
His	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{ND1}, \text{HD1}, \text{CD2}, \text{HD2}, \text{CE1}, \text{HE1}, \text{NE2}\}*$
Ile	3	$\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{HB}, \text{CG1}, \text{CG2}, \text{QG2}\}$ $\mathcal{S}^{(3)} = \{\text{CB}, \text{CG1}, \text{CD1}, \text{QD1}\}*$
Leu	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{HG}, \text{CD1}, \text{CD2}, \text{QD1}, \text{QD2}, \text{QQD}\}$
Lys	5	$\mathcal{S}^{(1)} = \{\text{N}, \text{CA}, \text{HA}, \text{CB}, \text{C}\}$ $\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{CG}\}*$ $\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{CD}\}*$ $\mathcal{S}^{(4)} = \{\text{CG}, \text{CD}, \text{CE}\}*$ $\mathcal{S}^{(5)} = \{\text{CD}, \text{CE}, \text{NZ}, \text{QZ}\}*$
Met	4	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{SD}\}*$ $\mathcal{S}^{(4)} = \{\text{CG}, \text{SD}, \text{CE QE}\}*$
Phe	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{CD1}, \text{HD1}, \text{CE1}, \text{HE1}, \text{CZ}, \text{HZ}, \text{CE2}, \text{HE2}, \text{CD2}, \text{HD2}, \text{QD}, \text{QE}, \text{QR}\}*$
Pro	1	$\mathcal{S}^{(1)} = \{\text{N}, \text{CD}, \text{CA}, \text{HA}, \text{CB}, \text{HB2}, \text{HB3}, \text{QB}, \text{CG}, \text{HG2}, \text{HG3}, \text{QG}, \text{HD2}, \text{HD3}, \text{QD}, \text{C}\}$
Ser	2	$\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{OG}\}*$
Thr	2	$\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{HB}, \text{OG1}, \text{CG2}, \text{QG2}\}$
Trp	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{CD1}, \text{HD1}, \text{CD2}, \text{CE2}, \text{CE3}, \text{HE3}, \text{NE1}, \text{HE1}, \text{CZ2}, \text{HZ2}, \text{CZ3}, \text{HZ3}, \text{CH2}, \text{HH2}\}*$
Tyr	3	$\mathcal{S}^{(3)} = \{\text{CB}, \text{CG}, \text{CD1}, \text{HD1}, \text{CE1}, \text{HE1}, \text{CE2}, \text{HE2}, \text{CD2}, \text{HD2}, \text{CZ}, \text{OH}, \text{QD}, \text{QE}, \text{QR}\}*$
Val	2	$\mathcal{S}^{(2)} = \{\text{CA}, \text{CB}, \text{HB}, \text{CG1}, \text{CG2}, \text{QG1}, \text{QG2}, \text{QQG}\}$

Then U has full column rank, $\mathbf{1} \in \mathbf{range}(U)$, and

$$\mathbf{face}\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_1]) = D[\mathcal{C}_1]\} = US_+^{n-n_1+r_1+1}U^\top,$$

where K is the Gram matrix corresponding to D.

We give an informal proof for Theorem 4. Let $\bar{G}_1 = \mathsf{K}^\dagger(D_1) \in \mathcal{S}_+^{n_1} \cap \mathcal{S}_C^{n_1}$; that is, let $\bar{G}_1$ be the centered Gram matrix corresponding to D_1. In general, Gram matrices are not centered; however, we can formulate $K_1 = K[\mathcal{C}_1]$ using the centered Gram matrix as (see (3.6)):

$$K_1 = \bar{G}_1 + \mathsf{D}_{\mathbf{1}}(\boldsymbol{v}), \quad \text{for some } \boldsymbol{v} \in \mathbb{R}^{n_1},$$

where, as a reminder $\mathsf{D}_{\mathbf{1}}(\boldsymbol{v}) := \boldsymbol{v}\mathbf{1}_n^\top + \mathbf{1}_n\boldsymbol{v}^\top$. Since $\mathbf{range}(V_1) = \mathbf{range}(\bar{G}_1)$, we can conclude that:

$$\mathbf{range}(K_1) \subseteq \mathbf{range}([\bar{G}_1\ \mathbf{1}]) = \mathbf{range}([\bar{V}_1\ \mathbf{1}]) = \mathbf{range}(\bar{U}),$$

and, consequently, we see that (Krislock, 2010, Lemmas 2.14 and 4.2):

$$\mathbf{face}\{K_1\} \trianglelefteq \bar{U}_1\mathcal{S}_+^{r_1+1}\bar{U}_1^\top.$$

This indicates that the K_1 has a face, in which case, the entire Gram matrix K has also a face, defined as (Krislock, 2010, Lemma 4.3):

$$\mathbf{face}\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_1]) = D[\mathcal{C}_1]\} \trianglelefteq U\mathcal{S}_+^k U^\top,$$

where $k = n - n_1 + r_1 + 1$. The rank of any *feasible* Gram matrix is reduced from n to k. If $n_1 \gg r_1$, this increases the speed for solving an EDMC problem, for example.

Computing the V matrix In Theorem 4, it is assumed that such a V_1 is given. One way to obtain such a matrix is to use the idea mentioned in the Two Disjoint Cliques (see page 85). We must find the low-dimensional embedding of $D[\mathcal{C}_1]$, we do so by computing the eigendecomposition of $\mathsf{K}^\dagger(D_1)$ as:

$$\mathsf{K}^\dagger(D[\mathcal{C}_1]) = V_1\Lambda_1V_1^\top, \quad V_1 \in \mathbb{R}^{n_1\times r_1},\ \Lambda_1 \in \mathcal{S}_{++}^{r_1}.$$

It can be seen that V_1 has full column rank (columns are orthonormal) and also:

$$\mathbf{range}(\mathsf{K}^\dagger(D[\mathcal{C}_1])) = \mathbf{range}(V_1).$$

We form $\bar{U}_1$ as:

$$\bar{U}_1 = \begin{bmatrix} V_1 & \frac{1}{\sqrt{n_1}}\mathbf{1}_{n_1} \end{bmatrix}, \tag{3.23}$$

where we scale the all-one vector by $\frac{1}{\sqrt{n_1}}$ to make the columns of $\bar{U}_1$ orthonormal (further discussion of why this is important will follow). For comparison, in the TWO DISJOINT CLIQUES, instead of directly using V_1, we used:

$$\bar{U}_1 = \begin{bmatrix} P_1^\top & \mathbf{1}_{n_1} \end{bmatrix} = \begin{bmatrix} V_1\Lambda^{1/2} & \mathbf{1}_{n_1} \end{bmatrix},$$

which differs with the $\bar{U}_1$ in (3.23) only in the norm of its columns; their ranges are the same, because for any $A \in \mathbb{R}^{p\times q}$ and $\Gamma = \mathbf{Diag}(\gamma_1, \ldots, \gamma_q)$, $\gamma_i > 0, \forall i$, we have:

$$\mathbf{range}(A) = \mathbf{range}(A\Gamma).$$

The Intersection of two faces

To find the face of a protein molecule, we need to find the intersection of a set of faces (one face for each clique). We start by considering the problem of TWO INTERSECTING CLIQUES, which is defined as follows: for $\ell = 1, 2$, let $\mathcal{C}_\ell \subseteq 1{:}n$ be two intersecting cliques of points such that $|\mathcal{C}_1| = n_1$, $|\mathcal{C}_2| = n_2$, $|\mathcal{C}_1 \cap \mathcal{C}_2| = \alpha$, and $\mathcal{C}_1 \cup \mathcal{C}_2 = 1{:}n$; we see that $n = n_1 + n_2 - \alpha$. In addition, for $\ell = 1, 2$, let $D_\ell = D[\mathcal{C}_\ell]$ and $\mathbf{embdim}(D_\ell) = r_\ell$. We show how to find the face of $\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_\ell]) = D_\ell,\ \ell = 1, 2\}$.

For $\ell = 1, 2$, let $F_\ell = U_\ell \mathcal{S}_+^{n-n_\ell+r_\ell+1} U_\ell^\top$ be the face, computed by assuming that only $\mathcal{C}_\ell$ is present in the EDMC graph, $G = (\mathcal{V}, \mathcal{E})$. For example for $\ell = 1$, no information about the points in $\mathcal{C}_2 \setminus \mathcal{C}_1$ is specified. If U_1 and U_2 are given by:

$$U_1 = \begin{array}{c} \\ |\mathcal{C}_1| \\ |\mathcal{C}_2\setminus\mathcal{C}_1| \end{array} \begin{array}{c} \begin{array}{cc} r_1+1 & |\mathcal{C}_2\setminus\mathcal{C}_1| \end{array} \\ \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & I \end{bmatrix} \end{array}$$

$$U_2 = \begin{array}{c} \\ |\mathcal{C}_1\setminus\mathcal{C}_2| \\ |\mathcal{C}_2| \end{array} \begin{array}{c} \begin{array}{cc} |\mathcal{C}_1\setminus\mathcal{C}_2| & r_2+1 \end{array} \\ \begin{bmatrix} I & 0 \\ 0 & \bar{U}_2 \end{bmatrix} \end{array},$$

then we need to compute a $U \in \mathbb{R}^{n\times k}$ that satisfies:

$$\mathbf{range}(U) = \mathbf{range}(U_1) \cap \mathbf{range}(U_2);$$

where the value of k depends on n_1, n_2, r_1, r_2, and α (size of the intersection); we discuss finding k shortly. If we can find such a U, then $F = U\mathcal{S}_+^k U^\top$ is the intersection of two faces, i.e., $F = F_1 \cap F_2$. We find the intersection of faces by intersecting the corresponding subspaces, as follows.

Algorithm 1: Finding principal angles and vectors between two subspaces

input : $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{m\times q}$ $(p \geq q \geq 1)$
output: Orthonormal matrices $U, V \in \mathbb{R}^{m\times q}$ and $\cos(\theta_i)$

Compute the following QR factorizations

$$\begin{aligned} A &= Q_A R_A \qquad Q_A^\top Q_A = I_p, \quad R_A \in \mathbb{R}^{p\times p} \\ B &= Q_B R_B \qquad Q_B^\top Q_B = I_q, \quad R_B \in \mathbb{R}^{q\times q} \end{aligned}$$

$C = Q_A^\top Q_B$
Compute SVD of C as $C = Y\Sigma Z^\top$
$[\cos(\theta_1), \ldots, \cos(\theta_q)]^\top = \mathbf{diag}(\Sigma)$
$U = Q_A Y[:, 1{:}q]$
$V = Q_B Z$

Intersection of subspaces

For two matrices, $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{m\times q}$ $(p \geq q \geq 1)$, each with linearly independent columns, let $F = \mathbf{range}(A)$ and $G = \mathbf{range}(B)$ be two subspaces in $\mathbb{R}^m$. It can be seen that their dimensions satisfy:

$$p \;=\; \mathbf{dim}(F) \geq \mathbf{dim}(G) \;=\; q \geq 1.$$

The *principal angles* (Golub and Van Loan, 1996) between F and G, $\theta_1, \ldots, \theta_q \in [0\ \pi/2]$, are recursively determined by solving:

$$\begin{aligned} \cos(\theta_k) \;=\; \max_{\boldsymbol{u}\in F}\ \max_{\boldsymbol{v}\in G}\ & \boldsymbol{u}^\top \boldsymbol{v} \;=\; \boldsymbol{u}_k^\top \boldsymbol{v}_k \\ \text{subject to}\quad & \|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1 \\ & \boldsymbol{u}^\top \boldsymbol{u}_i = 0 \quad i = 1{:}k-1 \\ & \boldsymbol{v}^\top \boldsymbol{v}_i = 0 \quad i = 1{:}k-1. \end{aligned}$$

Due to the recursive procedure, we have $0 \leq \theta_1 \leq \cdots \leq \theta_q \leq \pi/2$. The vectors $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q\}$ and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q\}$ are called the *principal vectors* between F and G.

Golub and Van Loan (1996, Algorithm 12.4.3) proposed a systematic algorithm to compute the principal angles and vectors: Algorithm 1 computes the orthogonal matrices $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_q]$ and $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q]$ and $\cos(\theta_1), \ldots, \cos(\theta_q)$, where θ_k and $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are the corresponding principal angles and vectors, respectively, between F and G.

Using Algorithm 1, an orthonormal basis for $\mathbf{range}(A) \cap \mathbf{range}(B)$ can be found as indicated in the following theorem:

Theorem 5 (Golub and Van Loan (1996)) *Let* $\{\cos(\theta_k), \boldsymbol{u}_k, \boldsymbol{v}_k\}_{k=1}^{q}$ *be defined by Algorithm 1. If the index* s *is defined such that* $1 = \cos(\theta_1) = \cdots = \cos(\theta_s) > \cos(\theta_{s+1})$*, then we have*

$$\mathbf{range}(A) \cap \mathbf{range}(B) = \mathbf{span}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_s\} = \mathbf{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s\}.$$

Recalling the Two Intersecting Cliques problem, if U_1 and U_2 are computed, we can use Theorem 5 to compute a U satisfying:

$$\mathbf{range}(U) = \mathbf{range}(U_1) \cap \mathbf{range}(U_2),$$

which concludes our semidefinite facial reduction for the two cliques. If U_1 and U_2 have orthonormal columns, then two QR decompositions in Algorithm 1 are not needed. This led us to make some changes in forming $\bar{U}_1$ in (3.23).

Last, we use the subspace intersection method to solve the Two Disjoint Cliques problem. Let the corresponding U_1 and U_2 be defined as:

$$U_1 = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} r_1+1 & n_2 \end{array} \\ \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & I_{n_2} \end{bmatrix} \end{array}$$

$$U_2 = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} n_1 & r_2+1 \end{array} \\ \begin{bmatrix} I_{n_1} & 0 \\ 0 & \bar{U}_2 \end{bmatrix} \end{array}.$$

If Algorithm 1 is applied to these matrices, because both of them have orthonormal columns, there is no need for QR decomposition:

$$C = U_1^\top U_2 = \begin{array}{c} \\ r_1+1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} n_1 & r_2+1 \end{array} \\ \begin{bmatrix} \bar{U}_1^\top & 0 \\ 0 & \bar{U}_2 \end{bmatrix} \end{array}.$$

We find the SVD decomposition of $C = Y\Sigma Z^\top$ as:

$$Y = \begin{array}{c} \\ r_1+1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} r_1+1 & r_2+1 \end{array} \\ \begin{bmatrix} I & 0 \\ 0 & \bar{U}_2 \end{bmatrix} \end{array}$$

$$Z = \begin{array}{c} \\ n_1 \\ r_2+1 \end{array} \begin{array}{c} \begin{array}{cc} r_1+1 & r_2+1 \end{array} \\ \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & I \end{bmatrix} \end{array}$$

$$\Sigma = I_{r_1+r_2+2},$$

Therefore, as expected, the intersection of subspaces U_1 and U_2 is the range of the following matrix:

$$U = \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & \bar{U}_2 \end{bmatrix},$$

which is identical to the one obtained earlier, up to the scaling of the columns.

Dimension of the of the intersection We showed how in the the Two Intersecting Cliques problem, a proper face of $\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_\ell]) = D_\ell,\ \ell = 1, 2\}$ can be found as $U\mathcal{S}_+^k U^\top$, but no insight was provided about the value of k. Here, we present insight about k that works only for two intersecting cliques. Later, we present the general framework.

Let $P = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n_1}]$, $\boldsymbol{p}_i \in \mathbb{R}^{r_1}$ and $Q = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{n_2}]$, $\boldsymbol{q}_i \in \mathbb{R}^{r_2}$, be the low-dimensional embeddings of $D[\mathcal{C}_1]$ and $D[\mathcal{C}_2]$, respectively. Moreover, let $X_1 = X[:, \mathcal{C}_1]$ and $X_2 = X[:, \mathcal{C}_2]$ be the high-dimensional embedding of the points in the cliques, which can be written as:

$$\begin{aligned} X_1 &= R_1 P + \boldsymbol{t}_1 \mathbf{1}_{n_1} \\ X_2 &= R_2 Q + \boldsymbol{t}_2 \mathbf{1}_{n_2}, \end{aligned}$$

where for $\ell = 1, 2$, $R_\ell \in \mathbb{R}^{n \times r_\ell}$ and $\boldsymbol{t}_\ell \in \mathbb{R}^n$ are the rotation matrices and translation vectors. If the intersection of cliques is defined as $\mathcal{I} = \mathcal{C}_1 \cap \mathcal{C}_2$, then $\alpha = |\mathcal{I}|$ is the size of the intersection. Moreover, without loss of generality, assume $\mathcal{I} = \{1, \ldots, \alpha\}$ if $\alpha \geq 1$ and $\mathcal{I} = \emptyset$, if otherwise. In general, we have:

$$X_1[:, i] = X_2[:, i], \quad \forall i \in \mathcal{I}.$$

Therefore, for $i = 1$, we have

$$R_1 \boldsymbol{p}_1 + \boldsymbol{t}_1 = R_2 \boldsymbol{q}_1 + \boldsymbol{t}_2 \Rightarrow \boldsymbol{t}_2 = R_1 \boldsymbol{p}_1 - R_2 \boldsymbol{q}_1 + \boldsymbol{t}_1, \tag{3.24}$$

which indicates that the translation of the points in $\mathcal{C}_2$ is not arbitrary anymore. Similarly, for $1 < i \leq \alpha$, by substituting $\boldsymbol{t}_2$ from the above, we write:

$$R_1\boldsymbol{p}_i + \boldsymbol{t}_1 = R_2\boldsymbol{q}_i + \boldsymbol{t}_2 \Rightarrow R_1(\boldsymbol{p}_i - \boldsymbol{p}_1) = R_2(\boldsymbol{q}_i - \boldsymbol{q}_1).$$

In general, if $\tilde{P} = [\boldsymbol{p}_2, \ldots, \boldsymbol{p}_\alpha] - \boldsymbol{p}_1\mathbf{1}_{\alpha-1}^\top$ and $\tilde{Q} = [\boldsymbol{q}_2, \ldots, \boldsymbol{q}_\alpha] - \boldsymbol{q}_1\mathbf{1}_{\alpha-1}^\top$, we have

$$R_1\tilde{P} = R_2\tilde{Q} \Rightarrow \tilde{P} = R_1^\top R_2\tilde{Q},$$

which implies that R_1 and R_2 are not arbitrary and should satisfy the above constraint. In fact, for $\alpha \geq \min(r_1+1, r_2+1)$, assuming that $r_1 \geq r_2$, the above equation simplifies to $R_2 = R_1\tilde{P}\tilde{Q}^{-1}$, that is, R_2 is completely expressed in terms of R_1. Krislock and Wolkowicz (2010) call the case where $\alpha \geq \min(r_1, r_2) + 1$ a *rigid* intersection, because one clique is completely expressed in terms of the other. For rigid intersections, where the problem reduces to the *Orthogonal Procrustes* if $r_1 = r_2$, we have $k = \max(r_1, r_2) + 1$. In general, for $\alpha < \min(r_1, r_2) + 1$ we have $k = r_1 + r_2 + 2 - \alpha$, which for $\alpha = 0$ (disjoint cliques) reduces to $k = r_1 + r_2 + 2$, as expected.

Algorithm for Finding the Face of the Structure

The protein molecule is made of q cliques, $\{\mathcal{C}_1, \ldots, \mathcal{C}_q\}$, such that $D[\mathcal{C}_l]$ is known and we have $r_l = \mathbf{embdim}(D[\mathcal{C}_l])$. For $l = 1, \ldots, q$, let $n_l = |\mathcal{C}_l|$. Let $\mathcal{F}$ be the feasible set of the EDMC problem. If for each clique $\mathcal{C}_l$, we define $\mathcal{F}_l := \{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_l]) = D[\mathcal{C}_l]\}$, then

$$\mathcal{F} \subseteq \left(\bigcap_{l=1}^{q} \mathcal{F}_l\right) \cap \mathcal{S}_C^n.$$

For $l = 1, \ldots, q$, let $F_l := \mathbf{face}\{\mathcal{F}_l\}$, such that $F_l = U_l S_+^{n-n_l+r_l+1} U_l^\top$ and U_l is computed by solving the Single Clique problem, as mentioned earlier. We have (Krislock, 2010):

$$\begin{aligned}
\mathbf{face}\{\mathcal{F}\} \trianglelefteq \left(\bigcap_{l=1}^{q} F_l\right) \cap \mathcal{S}_C^n \\
&= \left(\bigcap_{l=1}^{q} U_l S_+^{n-n_l+r_l+1} U_l^\top\right) \cap \mathcal{S}_C^n \\
&= (US_+^k U^\top) \cap \mathcal{S}_C^n,
\end{aligned}$$

where $U \in \mathbb{R}^{n\times k}$ is a full column rank matrix that satisfies:

$$\mathbf{range}(U) = \bigcap_{l=1}^{q} \mathbf{range}(U_l).$$

To find such a U matrix, we need to use Theorem 5, $q-1$ times: $\mathcal{C}_1$ and $\mathcal{C}_2$ are intersected and a new clique is formed as $\mathcal{A} = \mathcal{C}_1 \cup \mathcal{C}_2$, such that the product of intersecting U_1 and U_2 is its U_A. Then, $\mathcal{A}$ is intersected with $\mathcal{C}_3$, and this process iteratively continues until all cliques are intersected. However, this process can be inefficient and numerically inaccurate for these reasons:

1. Suppose we plan to find the intersection of U_1 and U_2:

$$U_1 = \begin{array}{c} \\ |\mathcal{C}_1| \\ |\mathcal{C}_2\backslash\mathcal{C}_1| \\ n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \begin{array}{c} \begin{array}{ccc} r_1+1 & |\mathcal{C}_2\backslash\mathcal{C}_1| & n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \\ \begin{bmatrix} \bar{U}_1 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \end{array}$$

$$U_2 = \begin{array}{c} \\ |\mathcal{C}_1\backslash\mathcal{C}_2| \\ |\mathcal{C}_2| \\ n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \begin{array}{c} \begin{array}{ccc} |\mathcal{C}_1\backslash\mathcal{C}_2| & r_2+1 & n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \\ \begin{bmatrix} I & 0 & 0 \\ 0 & \bar{U}_2 & 0 \\ 0 & 0 & I \end{bmatrix} \end{array},$$

   because both of these matrices have orthonormal columns, the C matrix in Algorithm 1 is:

$$C = U_1^\top U_2 = \begin{array}{c} \\ |\mathcal{C}_1\cup\mathcal{C}_2| \\ n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \begin{array}{c} \begin{array}{cc} |\mathcal{C}_1\cup\mathcal{C}_2| & n-|\mathcal{C}_1\cup\mathcal{C}_2| \end{array} \\ \begin{bmatrix} C' & 0 \\ 0 & I \end{bmatrix} \end{array},$$

   where C' is not computed, because it is not needed for our purpose here. If we compute SVD of C, it contains $n - |\mathcal{C}_1 \cup \mathcal{C}_2|$ trivial and non-informative singular values equal to one. It can be seen that the U_1 and U_2 are unnecessarily enlarged. To avoid a large matrix SVD calculation for intersecting U_1 and U_2, it suffices to shrink the matrix sizes by deleting the rows and columns indexed by $1{:}n \setminus \mathcal{C}_1 \cup \mathcal{C}_2$, creating two new shrunk matrices U_1^S and U_2^S . A new clique is formed as $\mathcal{A} = \mathcal{C}_1 \cup \mathcal{C}_2$; however, this time the product of intersection is $\bar{U}_A$, not U_A.

2. As the intersection process progresses, $\mathcal{A}$ and $\bar{U}_A$ grow larger. Suppose $p < q$ cliques have been processed so far, i.e., $\mathcal{A} = \bigcup_{l=1}^{p} \mathcal{C}_l$. For the next clique, if p is close to q, then $|\mathcal{C}_{l+1}| \ll |\mathcal{A} \cup \mathcal{C}_{l+1}|$. Consequently, its shrunk matrix U_{l+1}^S will be very sparse and the SVD calculation of $C = U_l^\top U_A^S$ will be inaccurate. We propose the hierarchical bottom-up intersection method, listed in Algorithm 2. Any two cliques

Algorithm 2: Hierarchical bottom-up intersection

input : Set of cliques $\{\mathcal{C}_l\}$ and their matrices $\{U_l\}$, $l = 1, \ldots, q$
output: Matrix U such that $\mathbf{range}(U) = \bigcap_{l=1}^{q} \mathbf{range}(U_l)$

```
// Initialization
```
for $i \leftarrow 1$ **to** q **do**
  $Q_l^{(1)} \leftarrow U_l$ // $Q_j^{(i)}$: U of the subtree rooted at the node j, level i
  $\mathcal{A}_l^{(1)} \leftarrow \mathcal{C}_l$ // $\mathcal{A}_j^{(i)}$: points in the subtree rooted at the node j, level i
end
$v \leftarrow \lfloor \log(q) \rfloor + 1$ // number of levels in the tree
$p \leftarrow q$ // number of cliques in the current level
$p' \leftarrow p$ // number of cliques in the lower level
for $i \leftarrow 2$ **to** v **do**
  $p \leftarrow \lceil p'/2 \rceil$
  for $j \leftarrow 1$ **to** p **do**
    $\ell_1 \leftarrow 2(j-1)+1$
    $\mathcal{A}_j^{(i)} \leftarrow \mathcal{A}_{\ell_1}^{(i-1)}$
    $Q_j^{(i)} \leftarrow Q_{\ell_1}^{(i-1)}$
    if $\ell_1 < p'$ **then**
      $\ell_2 \leftarrow \ell_1 + 1$
      $\mathcal{A}_j^{(i)} \leftarrow \mathcal{A}_j^{(i)} \cup \mathcal{A}_{\ell_2}^{(i-1)}$
      $Q_j^{(i)} \leftarrow \mathrm{Intersect}(Q_j^{(i)}, Q_{\ell_2}^{(i-1)})$
    end
  end
  $p' \leftarrow p$
end
$U \leftarrow Q_1^{(v)}$ // For the root $\mathcal{A}_1^{(v)} = 1{:}n$

being intersected in this approach are of nearly equal size (at least for the cliques in the protein structure, where cliques have similar sizes).

We now have an efficient method for computing the face of the feasible set.

Constraints for preserving the structure of the cliques In the absence of semidefinite facial reduction, the set of equality constraints preserving the structure of the cliques,

i.e., preserving $D[\mathcal{C}_l]$, can be formulated as:

$$\mathcal{E} = \bigcup_{i=1}^{q} \mathcal{E}_l,$$

where $\mathcal{E}_l$ is the set of constraints fixing each clique. If we find a *base* set of points, $\mathcal{B}_l$, for each clique $\mathcal{C}_l$ such that $\mathcal{B}_l \subseteq \mathcal{C}_l$ and $\mathbf{embdim}(D[\mathcal{B}_l]) = r_l$, then by fixing the distances between points in the base set and then by fixing the distances between points in $\mathcal{C}_l \setminus \mathcal{B}_l$ and points in $\mathcal{B}_l$, the entire clique is kept rigid. Therefore, for each clique we have:

$$\mathcal{E}_l = \underbrace{\{(i,j) : i,j \in \mathcal{B}_l, i > j\}}_{\mathcal{E}_l^{(B)}} \cup \underbrace{\{(i,j) : i \in \mathcal{B}_l, j \in \mathcal{C}_l \setminus \mathcal{B}_l\}}_{\mathcal{E}_l^{(C)}};$$

$\mathcal{E}_l^{(B)}$ fixes the distances between the base points and $\mathcal{E}_l^{(C)}$ fixes the distances between all other points and the base points. We see that:

$$|\mathcal{E}_l^{(B)}| = \frac{1}{2} r_l(r_l+1),$$
$$|\mathcal{E}_l^{(C)}| = (n_l - r_l - 1)(r_l + 1).$$

By using the *rule of sum* we find the following upper bound:

$$|\mathcal{E}| \leq \sum_{l=1}^{q} \frac{1}{2} r_l(r_l+1) + \sum_{l=1}^{q} (n_l - r_l - 1)(r_l+1);$$

this is an upper bound because some cliques may share some equality constraints.

Semidefinite facial reduction removes only the redundant clique-preserving constraints, but does not remove all of them. For example, consider $\mathbf{face}\{K \in \mathcal{S}_+^n : \mathsf{K}^\dagger(K[\mathcal{C}_l]) = D[\mathcal{C}_l]\} \trianglelefteq U_l S_+^k U_l^\top$, the all-zero matrix 0 is in any face, which is apparently an incorrect solution. To keep the structure of the cliques intact, it suffices to enforce $\mathsf{K}^\dagger(K[\mathcal{B}_l]) = D[\mathcal{B}_l]$ (Alipanahi et al., 2011b). We are required to fix *only* the distances between base points (only $\mathcal{E}_l^{(B)}$ part of $\mathcal{E}_l$). After facial reduction, the total number of equality constraints is bounded by:

$$|\mathcal{E}_{\mathrm{FR}}| \leq \sum_{l=1}^{q} \frac{1}{2} r_l(r_l+1),$$

where, in comparison to the original problem, it lacks the second term in the bound. In proteins, $r_l \leq 3$, so $|\mathcal{E}_{\mathrm{FR}}| < 6q$, while for the original problem $|\mathcal{E}| < 4n$. Considering the typical values of q and n, this is nearly a three- to four-fold reduction.

A method for finding the value of k Here, we present a simple method for finding the value of k, listed in Algorithm 3. To explain this method, we use an example made of three cliques, as depicted in Fig. 3.13. The algorithm works as follows: Preserving the first clique, $\mathcal{C}_1$, requires fixing the distances between 3 atoms, of which we select CA_1, C_1, and O_1, hence $k = 3$. We move on to the next clique, $\mathcal{C}_2$ (a 3D clique), where CA_2 and N_2 are already fixed. Therefore, we only need to add two more atoms: HA_2 and C_2; k is increased by two to five. Finally, in $\mathcal{C}_3$, we add only O_2 to fix this clique, resulting in $k = 6$. The atoms added during this process are colored red.

Table 3.3 contains the increase in the value of k by adding each type of amino acid to the peptide chain. For each peptide plane k is increased by two that is on par with the increase in the problem size of torsion angle dynamics (two more torsion angles). On average, by adding each amino acid, the number of atoms, n, is increased by 18.2, while the SDP matrix size, k, is increased by 5.0, a 3.6 fold difference.

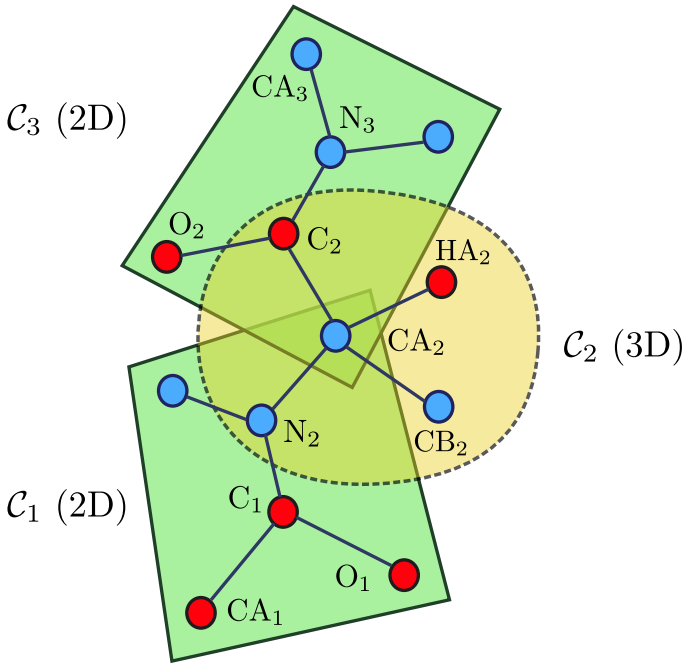


Figure 3.13: A simple example for finding the value of k. The atoms added at each step are colored red.

Algorithm 3: Finding the value of k

input : EDM D, set of cliques $\{\mathcal{C}_l\}$ and their embedding dimensions, r_l, $l = 1, \ldots, q$
output: Number of columns in the matrix U, k

$\mathcal{A} \leftarrow \mathcal{C}_1$
$k \leftarrow r_1 + 1$
for $i \leftarrow 2$ **to** q **do**
  $\mathcal{I}_l \leftarrow \mathcal{A} \cap \mathcal{C}_l$
  $k \leftarrow k + (r_l + 1 - \mathbf{embdim}(D[\mathcal{I}_l]))$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{C}_l$
end

3.3.5 Solving and Refining the Reduced SDP Problem

The SPROS method flowchart is depicted in Fig. 3.14. In it, we describe the blocks for solving the SDP problem and for refining the solution. We showed how to reduce the original Gram matrix, $K \in \mathcal{S}_+^n$, into a much smaller one by using the semidefinite facial reduction technique, such that:

$$\mathbf{face}\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_l]) = D[\mathcal{C}_l], \forall l \in 1{:}q\} \trianglelefteq U\mathcal{S}_+^k U^\top,\ U \in \mathbb{R}^{n\times k},$$

therefore, we can decompose the Gram matrix as $K = UZU^\top$, $Z \in \mathcal{S}_+^k$. It may seem that the reduced optimization problem satisfies the Slater's condition; however, it does not. The centering constraint, $K\mathbf{1}_n = \mathbf{0}$, or, equivalently $K \in \mathcal{S}_C^n$, reduces the rank by one, i.e., $\mathbf{rank}(K) \leq k-1$. Finding the new face is straightforward: If $V \in \mathbb{R}^{k\times(k-1)}$ has full column rank and satisfies $\mathbf{range}(V) = \mathbf{null}(\mathbf{1}_n^\top U)$, then we have (Krislock, 2010, Theorem 4.13)

$$\mathbf{face}\{K \in \mathcal{S}_+^n \cap \mathcal{S}_C^n : \mathsf{K}(K[\mathcal{C}_l]) = D[\mathcal{C}_l], \forall l \in 1{:}q\} \trianglelefteq (UV)\mathcal{S}_+^{k-1}(UV)^\top,$$

because $(UV)^\top\mathbf{1}_n = U^\top V^\top \mathbf{1}_n = \mathbf{0}$. We can find V by first applying the QR decomposition on $U^\top\mathbf{1}_n$ that results in $U^\top\mathbf{1}_n = QR$, $Q \in \mathbb{R}^{k\times k}$, and then forming $V = Q[:, 2{:}k]$. We define $\check{U} = UV$.

We formulate the reduced optimization problem as follows:

Table 3.3: Table summarizing properties of different amino acids: p denotes abundance of amino acids in percentile, t denotes the number of torsion angles (excluding ω), a denotes the total number of atoms and pseudo-atoms, s denotes the total number of atoms and pseudo-atoms in the side chains, q denotes the number of cliques in each side chain (the number in the parenthesis is the number of 3D cliques), and k denotes the increase in the SDP matrix size. The values in the Reduced column denote the same values in the side chain simplified case. The *weighted average* (*w.a.*) of quantity x is computed as $\sum_{i\in\mathcal{A}} p_i x_i$, where $\mathcal{A}$ is the set of twenty amino acids.

			Complete side chains				Simplified side chains			
A.A.	p	t	a	s	q	k	a	s	q	k
Ala	7.3	3	11	5	2 (2)	5	8	2	1 (1)	3
Arg	5.2	6	29	23	5 (4)	10	20	14	5 (1)	7
Asn	4.6	4	16	10	3 (2)	6	13	7	3 (1)	5
Asp	5.1	4	13	7	3 (2)	6	10	4	3 (1)	5
Cys	1.8	4	12	6	3 (2)	5	8	2	2 (1)	4
Glu	4.0	5	20	14	4 (3)	8	14	8	4 (1)	6
Gln	6.2	5	17	11	4 (3)	8	11	5	4 (1)	6
Gly	6.9	2	8	2	1 (1)	3	8	2	1 (1)	3
His	2.3	4	18	12	3 (2)	6	15	9	3 (1)	5
Ile	5.8	6	22	16	5 (5)	11	13	7	3 (2)	6
Leu	9.3	6	23	17	5 (5)	11	14	8	3 (2)	6
Lys	5.8	7	27	21	6 (6)	13	12	6	5 (1)	7
Met	2.3	6	20	14	5 (4)	10	11	5	4 (1)	6
Phe	4.1	4	24	18	3 (2)	6	21	15	3 (1)	5
Pro	5.0	1	17	12	1 (1)	3	17	12	1 (1)	3
Ser	7.4	4	12	6	3 (2)	6	8	2	2 (1)	4
Thr	5.8	5	15	9	4 (3)	8	11	5	2 (2)	5
Trp	1.3	4	25	19	3 (2)	6	22	16	3 (1)	5
Tyr	3.3	5	25	19	4 (2)	7	21	15	3 (1)	5
Val	6.5	5	19	13	4 (4)	9	13	7	2 (1)	5
w.a.	-	4.5	18.2	12.2	3.6 (3.0)	7.6	12.8	6.8	2.7 (1.3)	5.0

$$
\begin{array}{lll}
\text{minimize} & -\gamma\langle I, Z\rangle + \sum_{ij} W_{ij}\zeta_{ij} & \\
\text{subject to} & \langle \check{A}_{ij}, Z\rangle = E_{ij}, & (i,j) \in \mathcal{E}_{\text{FR}} \\
& \langle \check{A}_{ij}, Z\rangle + \xi_{ij} = U_{ij} + \zeta_{ij}, & (i,j) \in \mathcal{U} \\
& \langle \check{A}_{ij}, Z\rangle + \zeta_{ij} = L_{ij} + \xi_{ij}, & (i,j) \in \mathcal{L} \\
& \zeta_{ij}, \xi_{ij} \in \mathbb{R}_+, & (i,j) \in \mathcal{L} \cup \mathcal{U} \\
& Z \in \mathcal{S}_+^{k-1}, &
\end{array}
\tag{3.25}
$$

where for the constraint matrices we have

$$\langle A_{ij}, \check{U} Z \check{U}^\top \rangle = \langle \check{U}^\top A_{ij} \check{U}, Z\rangle \Rightarrow \check{A}_{ij} = \check{U}^\top A_{ij} \check{U}.$$

The equality constraints set, $\mathcal{E}_{\text{FR}}$, is the set of constraints fixing the structure of the cliques, hence, preserving all of the bond lengths and angles, and the splanarity of the coplanar atoms.

Weights and the regularization parameter For each type of upper and lower bound, we define a fixed penalizing weight for constraint violations. For example, for upper bounds (similarly for lower bounds) we have $\forall (i,j) \in \mathcal{U}_X, W_{ij} = w_X$. We set $w_N = 1$ and $w_H = w_D = w_T = 10$ because upper bounds from hydrogen bonds and disulfide/salt bridges are assumed to be more accurate than are NOE-derived upper bounds.

Setting the regularization parameter, γ, is a bit more challenging. If it is set too high, then the bounds can be arbitrarily violated and if it is too small, the rank reduction heuristic fails. We may use the upper bound on the trace term derived from the geodesic distances graph; or we can use this analysis:

Let R be radius of the protein, clearly $\textbf{trace}(K) \leq nR$ (protein molecule is assumed to be spherical with its center at the origin). For the upper bounds, let $0 \leq \alpha \leq 1$ be the fraction of violated constraints. The maximum violation is $2R$; therefore, for the sum of the slacks we have $\sum \zeta_{ij} \leq 2\alpha m_{\mathcal{U}} R$, where $m_{\mathcal{U}} = |\mathcal{U}|$. Discarding the role of lower bounds, with the goal of approximately balancing the two terms, a suitable γ is:

$$\gamma nR \approx 2\alpha w m_{\mathcal{U}} R \Rightarrow \gamma = \frac{2\alpha w m_{\mathcal{U}}}{n}.$$

In practice $\alpha \approx 0.01 - 0.15$, and $\bar{\gamma} \approx m_{\mathcal{U}} w/50n$ works well. Last, note that the optimization problem is not very sensitive to the value of γ and that a wide range of values, $\gamma \in [\bar{\gamma}/5\ 5\bar{\gamma}]$, produce good results.

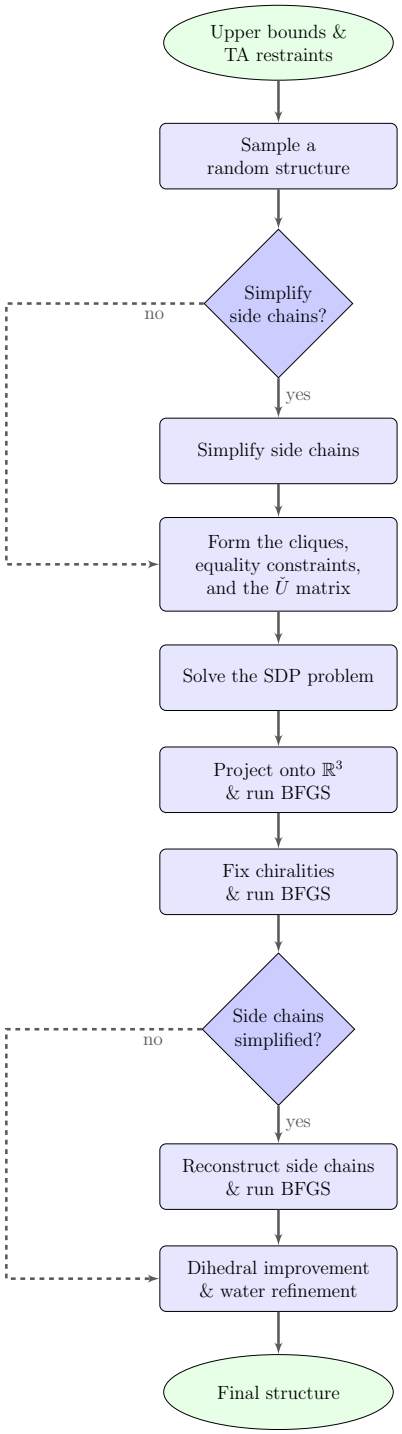


Figure 3.14: SPROS flowchart. Dashed lines indicate non-default routes.

Post-processing

Let $Z^* \in \mathcal{S}_+^{k-1}$ be the optimal positive semidefinite matrix that minimizes (3.25). Then, the optimal Gram matrix is $K^* = \check{U} Z^* \check{U}^\top$. To find the embedding, there is no need to compute the eigendecomposition of K^*; instead, it suffices to find the eigendecomposition of Z^*, which is a much smaller matrix:

$$Z^* = V_Z \Lambda V_Z, \quad \Lambda = \mathbf{Diag}(\lambda_1, \ldots, \lambda_p) \in \mathcal{S}_{++}^p,$$

where $p = \mathbf{rank}(Z^*)$. Then, we have $K^* = V \Lambda V^\top$, where $V = \check{U} V_Z$. Let $Y = \Lambda^{1/2} V^\top = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$, $\boldsymbol{y}_i \in \mathbb{R}^p$ be the embedding that corresponds to K^*; if $p > 3$; we need to project Y onto $\mathbb{R}^3$.

Let $W \in \mathbb{R}^{p \times p}$, be a rotation matrix, such that:

$$W = \begin{bmatrix} \boldsymbol{w}_1^\top \\ \vdots \\ \boldsymbol{w}_p^\top \end{bmatrix} = \begin{matrix} 3 \\ p-3 \end{matrix} \overset{p}{\begin{bmatrix} W_1 \\ W_2 \end{bmatrix}},$$

where $\boldsymbol{w}_\ell \in \mathbb{R}^p$. Since W is a rotation matrix, Y and WY have the same Gram matrix and EDM. We need to find $W_1 \in \mathbb{R}^{3 \times p}$ such that $X = W_1 Y$ contains as much information from Y as possible.

The first observation we make is that for any pair of points, $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$, because:

$$\begin{aligned} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 &= \sum_{\ell=1}^{3} (\boldsymbol{w}_\ell^\top \boldsymbol{y}_i - \boldsymbol{w}_\ell^\top \boldsymbol{y}_j)^2 \\ &\leq \sum_{\ell=1}^{p} (\boldsymbol{w}_\ell^\top \boldsymbol{y}_i - \boldsymbol{w}_\ell^\top \boldsymbol{y}_j)^2 \\ &= \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2, \end{aligned}$$

where in the last line we have used the fact that EDM is invariant to rotation. The projection always *shortens* the pair-wise distances (Mardia et al., 1980). One way to formulate the *lost* 'information content' is to measure the distortion in the distance matrices. Let $D^Y = \mathsf{K}(Y^\top Y)$ and $D^X = \mathsf{K}(X^\top X)$. We define the following loss function:

$$\psi = \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{ij}^Y - D_{ij}^X),$$

which is always nonnegative, since $D_{ij}^Y \geq D_{ij}^X$, $\forall i, j$. Because both X and Y are centered, the loss function can be simplified to:

$$\psi = 2n \left(\mathbf{trace}(Y^\top Y) - \mathbf{trace}(X^\top X) \right).$$

The above loss function is minimized if Y is projected onto its top three principal components (Mardia et al., 1980). Because rows of Y are already its principal coordinates, X is just the first three columns of Y or equivalently, $X = Y[1{:}3, :]$. For this choice of X, the loss function is:

$$\psi_{\min} = 2n \left(\sum_{i=1}^{p} \lambda_i - \sum_{i=1}^{3} \lambda_i \right) = 2n \sum_{i=4}^{p} \lambda_i,$$

where λ_i is the i-th largest eigenvalue of Z^* (Mardia et al., 1980). We define the *average normalized pair-wise distance shrinkage* as:

$$\eta = \frac{\sum_{i=4}^{p} \lambda_r}{\sum_{i=1}^{p} \lambda_i},$$

where $0 \leq \eta < 1$ is a projection loss measure; the larger the η, the more severe the distortion. In our experiments, $\eta > 0.01$ is an indication of large projection errors. To improve the quality of the protein structure, we perform a three-phase post-processing on it.

BFGS-based refinement We perform a BFGS-based refinement on the raw structure determined by the SDP solver. The BFGS method only requires the value of the objective function and its gradient at each point and is considered to be one of the best quasi-Newton nonlinear unconstrained optimization methods (Bertsekas and Bertsekas, 1999; Nocedal and Wright, 2000). Compared to the conjugate gradient method, it is generally faster and less sensitive to the accuracy of the line search.

Let $X^{(0)} = X_{\text{SDP}}$, we iteratively minimize the following objective function:

$$\begin{aligned} \phi(X) = w_E \sum_{(i,j)\in\mathcal{E}} \left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - e_{ij} \right)^2 + w_U \sum_{(i,j)\in\mathcal{U}} f\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - u_{ij} \right)^2 \\ + w_L \sum_{(i,j)\in\mathcal{L}} g\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - l_{ij} \right)^2 + w_R \sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2, \end{aligned} \tag{3.26}$$

where $f(\alpha) = \max(0, \alpha)$ and $g(\alpha) = \min(0, -\alpha)$. We set $w_E = 2$, $w_U = 1$, and $w_L = 1$. In

addition, to balance the regularization term, we set w_R as follows:

$$w_R = \alpha \frac{\phi(X^{(0)})|_{w_R=0}}{25\ \phi(X^{(0)})|_{w_R=1, w_E=w_U=w_L=0}} = \alpha \frac{\phi(X^{(0)})|_{w_R=0}}{25 \sum_{i=1}^{n} \|\boldsymbol{x}_i^{(0)}\|^2}.$$

where $-1 \leq \alpha \leq 1$ is a parameter controlling the regularization. If $\alpha < 0$, the distances between atoms are maximized, because, after projection, some of the distances have been shortened. In the process of projecting data to $\mathbb{R}^3$, this term helps to compensate for that error. However, if $\alpha > 0$, the distances between atoms are minimized, resulting in better packing of atoms in the protein molecule. In practice, different values for α can be used to generate slightly different structures, thus creating a bundle of structures.

Finally, computing the gradient of $\phi(X)$ is straightforward:

$$\begin{aligned}\nabla_{\boldsymbol{x}_i}\phi(X) = 2w_E \sum_{(i,j)\in\mathcal{E}} (\boldsymbol{x}_i - \boldsymbol{x}_j)\left(1 - \frac{e_{ij}}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}\right) + 2w_U \sum_{(i,j)\in\mathcal{U}} (\boldsymbol{x}_i - \boldsymbol{x}_j) f\left(1 - \frac{u_{ij}}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}\right) \\ + 2w_L \sum_{(i,j)\in\mathcal{L}} (\boldsymbol{x}_i - \boldsymbol{x}_j) g\left(1 - \frac{e_{ij}}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}\right) + 2w_R \sum_{i=1}^{n} \boldsymbol{x}_i, \qquad (3.27)\end{aligned}$$

which in turn is used to update $\boldsymbol{x}_i$ by the BFGS method. In our experiments, running the gradient descent for 250 iterations was enough and virtually no improvement is gained for more than 1,000 iterations. Last, it should be noted that: (*i*) a large majority of the terms in the objective function are zero, because their corresponding constraints are not *active* (especially lower and upper bounds), (*ii*) this step can be extensively parallelized to make it run much faster.

Fixing incorrect chiralities After termination of the gradient descent, we check the sign of Φ torsion angles, because as can be seen from the Ramachandran plot (see Fig. 3.4), the large majority of Φ torsion angles lie in the $\Phi \leq 0$ half-plane. It is possible that the distances in the structure are perfect, but that the structure is the reflection of the actual one. One method for verifying this is to calculate the fraction of residues with positive Φ angles as:

$$\eta_\Phi = \frac{\sum_{i=2}^{\ell} 0.5(1 + \mathbf{sign}(\Phi_i))}{\ell},$$

if $\eta_\Phi < 0.5$, we flip the structure by negating the first row of X.

Chirality constraints cannot be enforced using only distances. Consequently, some chiral centers may have the incorrect enantiomer. In this step, SPROS checks the chiral centers and if it finds any problems, fixes it. For example, all amino acids except glycine have a chiral center at their CAs. The correct enantiomer called L-form (the incorrect

one is called the D-form) is found by the CO-R-N rule: as depicted in Fig. 3.15, looking through the HA-CA bond, the order of atoms clockwise should be CO, R, and N.

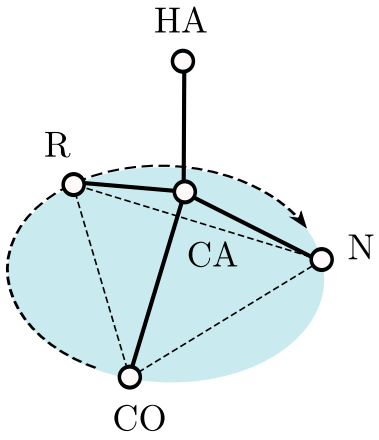


Figure 3.15: The CO-R-N rule: looking through the HA-CA bond, the order of atoms clockwise should be CO, R, and N.

One method for checking the CO-R-N rule is to measure the virtual dihedral angle made by (CA, N, C, CB) (Morris et al., 1992). If the angle is positive, chirality is correct, if not, it should be fixed. To fix an incorrect enantiomer, we flip the side chain and HA with regard to the plane, $\mathcal{P}$, which passes through N, CA, and C. The method is illustrated in Fig. 3.16: Let $\boldsymbol{v} = \boldsymbol{x}_2 - \boldsymbol{x}_1$, we are looking for the reflection of $\boldsymbol{x}_1$ with regard to $\mathcal{P}$ or $\boldsymbol{x}_2'$. Decompose $\boldsymbol{v}$ as $\boldsymbol{v} = \boldsymbol{v}' + \boldsymbol{v}''$. Then, we write:

$$\boldsymbol{x}_2' = \boldsymbol{x}_1 + \boldsymbol{w} = \boldsymbol{x}_1 + \boldsymbol{v}' - \boldsymbol{v}''.$$

Using this technique, we fix all incorrect chiralities.

Improving the stereochemical quality Williamson and Craven (2009) have described the effectiveness of explicit solvent refinement of NMR structures and suggest that it should be a standard procedure. For protein structures that have regions of high mobility/uncertainty due to few or no NOE observations, we have successfully employed a hybrid protocol in XPLOR-NIH that incorporates thin-layer water refinement (Linge et al., 2003) and a multidimensional torsion angle database (Kuszewski et al., 1996, 1997). This final step typically takes two to five minutes to complete.

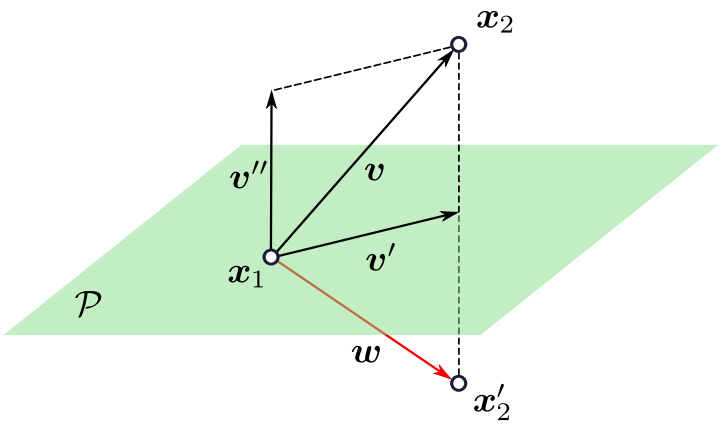


Figure 3.16: The method for fixing an incorrect chirality; the $\boldsymbol{x}_2$ is reflected with regard to the plane $\mathcal{P}$.

3.4 Results

We tested the performance of SPROS on 20 proteins: 16 protein data sets from the DOCR database in the NMR Restraints Grid (Doreleijers et al., 2003, 2005) and four real protein data sets from Donaldson's lab at York University. We picked proteins with different sizes and topologies, as listed in Table 3.4. From the empirical CDF of all protein entries in PDB, solved by NMR with sequence similarities less than 90%, as of September 20, 2011, nearly 98% of these proteins are shorter than 200 amino acids (see Fig. 3.17). Moreover, only less than 0.4 of proteins are longer than 300 amino acids. Therefore, we select proteins that are shorter than 300 amino acids.

In Table 3.5, we provide a comprehensive analysis of test proteins:

1. The total number of atoms and pseudo-atoms, n, this is the original PSD matrix size, and the original number of equality constraints, $m_{\mathcal{E}}$
2. The reduced PSD matrix size, n', and the number of equality constraints, $m'_{\mathcal{E}}$, in the reduced problem
3. The total number of cliques with the number of 2D and 3D cliques
4. The total number of upper bounds, $m_{\mathcal{U}}$
5. The fraction of different upper bound types in percentile, that is, intra-residue, $|i-j|=0$, sequential, $|i-j|=1$, medium range, $1<|i-j|\leq 4$, and long range, $|i-j|>4$

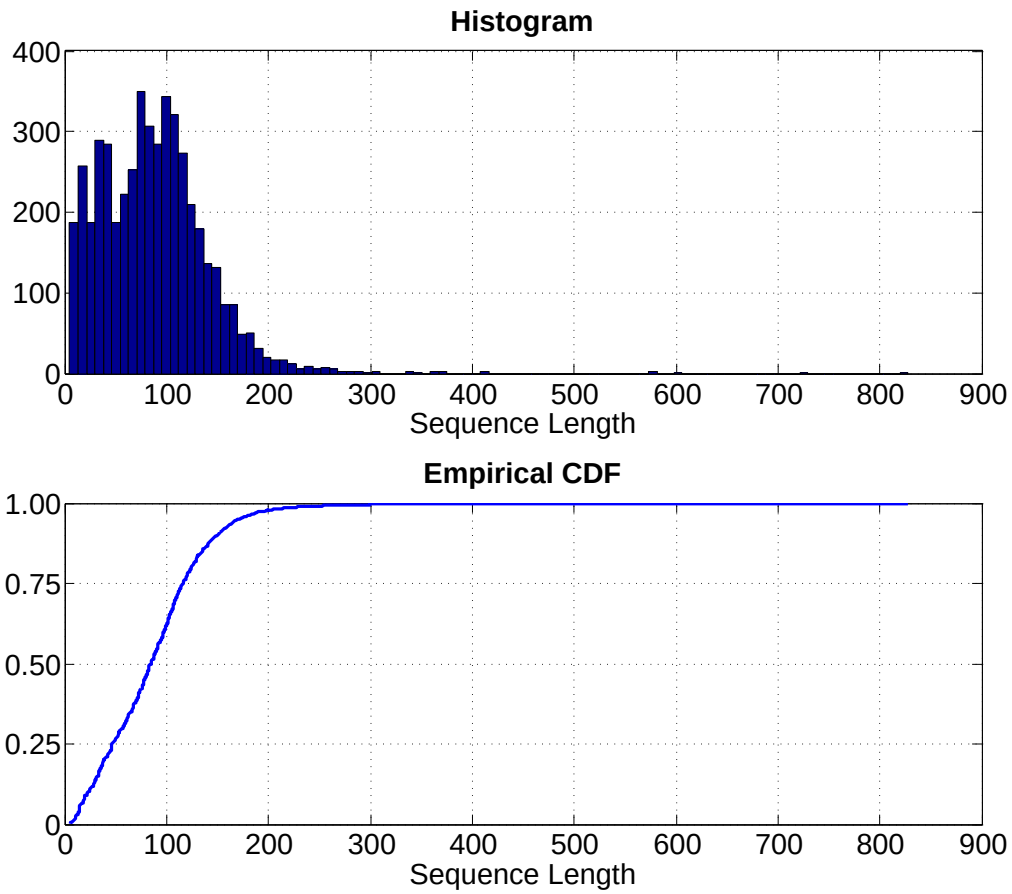


Figure 3.17: Histogram and CDF of sequence length of proteins solved by NMR. Nearly 98% of proteins are shorter than 200 amino acids.

6. The mean number of upper bounds, per residue, together with the corresponding standard deviation, note that $\bar{m}_{\mathcal{U}} = 2m_{\mathcal{U}}/\ell$, where ℓ is the number of residues

7. The number of upper bounds derived from NOE peaks, torsion angle restraints, and hydrogen bonds, disulfide and salt bridges

8. The ratio of the pseudo-atoms' presence in upper bounds. Pseudo-atoms add to the uncertainty of the bounds (they can add up to 4 Å to the bounds if both atoms involved are pseudo-atoms)

Table 3.4: Proteins selected from the DOCR database from the NMR Restraint Grid and proteins from Donaldson's lab at York University (last four rows). Molecular weights, fifth column, are in kDa.

ID	Description	topology	residues	size
1G6J	Ubiquitin	a+b	76	8.58
1B4R	PKD domain 1 from human polycystein-1	B	80	7.96
2KT6	Outer membrane usher protein papC	B	85	9.07
2E8O	SAM domain	A	103	11.40
1CN7	Yeast ribosomal protein L30	a/b	104	11.30
2KTS	Heat shock protein hslJ	a/b	117	12.85
2K49	UPF0339 protein SO3888	a+b	118	13.10
2K62	Liver fatty acid-binding protein	B	125	15.10
2L3O	Murine interleukin-3	A	127	14.30
2GJY	Tensin 1 PTB Domain	a+b	144	15.67
2KTE	Bacillus subtilis	a/b	152	17.21
1XPW	Human protein HSPCO34	B	153	17.44
2K7H	Stress-induced protein SAM22	a/b	157	16.66
2KVP	Vinculin Binding Site	A	165	17.28
2YTO	Amyloid beta A4 protein	a+b	176	19.17
2L7B	Apolipoprotein E	A	307	35.30
1Z1V	STE50 protein (Kwan et al., 2006)	A	80	9.31
HACS1	SH3 domain of HACS1 (unpublished)	B	87	9.63
2KIV	SAM1 and SAM2 domains (Kurabi et al., 2009)	A	148	16.67
2LJG	PTB domain of AIDA-1 (unpublished)	a+b	153	17.03

3.4.1 Implementation

The SPROS method has been implemented and tested in MATLAB 7.13 (apart from the water refinement, which is done by XPLOR-NIH). For solving the SDP problem, we used the SDPT3 method (Tütüncü et al., 2003), because it is arguably numerically more stable and it accepts rank-one constraint matrices. The SDPT3 solver uses MATLAB's built-in Cholesky decomposition and LU decomposition functions, which are not as fast as the state-of-the-art implementations (such as those using GotoBLAS2[4] routines). For minimizing the post-processing objective function (3.26), we implemented the functions for calculating objective and the gradient (3.27) in MATLAB's C MEX (functions written in C that are called from MATLAB). We used the BFGS implementation by Lewis and Overton (2009), which we found to be more stable and efficient than are other implementations. All the experiments were carried out on an Ubuntu 11.04 Linux PC with a 2.8 GHz Quad-Core processor and 8 GB of memory.

3.4.2 Determined Structures

The SPROS method was run on the test proteins and the final structures were generated. The superimposition of the determined structures and the reference structures are depicted in Figs. 3.19, 3.20, 3.21, 3.22, and 3.23. In Table 3.6, we list them:

1. **Run times:** run times of SDP solver, water refinement, and the total run time
2. **RMSD:** the mean and the standard deviation of the backbone and the heavy atoms RMSD. For each structure, we computed the mean and standard deviation of RMSDs between the SPROS structure and all structures in the reference bundle
3. **CB deviation:** the total number of residues with a CB deviation larger than 0.25 Å; "CB deviation" is an indicator of a set of abnormalities in the protein structure (Chen et al., 2010)
4. **Upper bound violations:** upper bound violations larger than 0.1 Å and 1.0 Å for the determined and reference structures in percentile
5. **Ramachandran:** the fraction of residues with torsion angles in the favored and allowed regions, together with the outliers, in percentile

From the 20 test proteins, nine of them have backbone RMSDs less than or equal to 1 Å, and 17 have backbone RMSDs less than 1.5 Å. Only two structures have backbone RMSDs

[4]Available from `http://www.tacc.utexas.edu/tacc-projects/gotoblas2/`.

Table 3.5: Information about the proteins used in testing SPROS. The second and third columns, n and n', list the original and reduced SDP matrix sizes, respectively. The fourth column lists the number of cliques in the protein. The fifth and sixth column, $m_{\mathcal{E}}$ and $m'_{\mathcal{E}}$, list the number of equality constraints in the original and reduced problems, respectively. The seventh column, $m_{\mathcal{U}}$, lists the total number of upper bounds for each protein. The eighth column, bound types, lists intra-residue, $|i-j|=0$, sequential, $|i-j|=1$, medium range, $1<|i-j|\leq 4$, and long range, $|i-j|>4$, respectively, in percentile. The ninth column, $\bar{m}_{\mathcal{U}} \pm s_{\mathcal{U}}$, lists the average number of upper bounds per residue, together with the standard deviation. The tenth column, $m_{\mathcal{N}}$, lists the number of NOE-inferred upper bounds. The eleventh column, $p_{\mathcal{U}}$, lists the fraction of pseudo-atoms in the upper bounds in percentages. The last two columns, $m_{\mathcal{T}}$ and $m_{\mathcal{H}}$, list the number of upper bounds inferred from torsion angle restraints, and hydrogen bonds, disulfide and salt bridges, respectively.

ID	n	n'	cliques (2D/3D)	$m_{\mathcal{E}}$	$m'_{\mathcal{E}}$	$m_{\mathcal{U}}$	bound types	$\bar{m}_{\mathcal{U}} \pm s_{\mathcal{U}}$	$m_{\mathcal{N}}$	$p_{\mathcal{U}}$	$m_{\mathcal{T}}$	$m_{\mathcal{H}}$
1G6J	1434	405	304 (201/103)	5543	1167	1354	21/29/17/33	31.9±15.3	1291	32	63	0
1B4R	1281	346	248 (145/103)	4887	1027	787	26/25/ 6 /43	17.1±10.8	687	30	22	78
2KT6	1492	413	307 (196/111)	5767	1328	1131	38/24/ 6 /31	25.3±17.0	1131	16	119	78
2E8O	1523	419	317 (212/105)	5846	1214	3157	19/29/26/26	71.4±35.4	3070	24	87	0
1CN7	1927	532	393 (253/140)	7399	1540	1560	46/24/12/18	23.1±13.4	1418	31	80	62
2KTS	2075	593	448 (299/149)	7968	1719	2279	22/28/14/36	34.6±17.4	2276	25	0	3
2K49	2017	574	433 (291/142)	7710	1657	2612	22/27/18/38	40.9±21.1	2374	27	146	92
2K62	2328	655	492 (327/165)	8943	1886	2367	21/32/15/32	33.9±18.6	2187	32	180	0
2L3O	1867	512	393 (269/124)	7143	1492	1270	24/38/20/18	22.5±12.7	1055	25	156	59
2GJY	2337	639	474 (302/172)	8919	1875	1710	7 /30/19/44	25.0±16.6	1536	29	98	76
2KTE	2576	717	542 (360/182)	9861	2089	1899	17/31/22/30	24.3±20.8	1669	30	124	106
1XPW	2578	723	541 (355/186)	9837	2081	1206	0 /31/11/58	17.0±10.8	934	37	210	62
2K7H	2710	756	563 (363/200)	10452	2196	2768	29/33/13/25	30.3±11.3	2481	19	239	48
2KVP	2533	722	535 (344/191)	9703	2094	5204	31/26/23/20	59.2±25.0	4972	22	232	0
2YT0	2940	828	627 (419/208)	11210	2404	3357	23/28/14/35	34.9±22.3	3237	30	120	0
2L7B	5603	1567	1205 (836/369)	21421	4521	4355	10/30/44/16	27.6±14.4	3459	23	408	488
1Z1V	1259	362	272 (181/91)	4836	1046	1261	46/24/18/13	28.6±16.3	1189	15	0	72
HACS1	1150	315	237 (156/81)	4401	923	828	46/21/ 5 /27	20.2±14.2	828	20	0	36
2KIV	2479	695	520 (342/178)	9553	2015	910	0 /27/35/38	13.5±11.9	786	31	0	124
2LJG	2343	662	495 (327/168)	9009	1909	1347	40/29/ 8 /22	16.4±11.9	1065	28	204	78

larger than 2 Å: the largest test protein, 2L7B, which has 307 residues, and 2KIV, which has the least mean number of upper bounds per residue and no torsion angle restraints.

We define a *residue-level contact map* as a matrix $C \in \mathbb{R}^{\ell \times \ell}$, $C_{ij} \in \{0, 1\}$, such that $C_{ij} = 1$ if there is at least one upper bound between an atom in residue i and an atom in residue j, and $C_{ij} = 0$ otherwise. We compare the residue-level contact maps of the protein with the highest mean number of contacts per residue, 2KVP, with the contact map of 2KIV in Fig 3.18. It can be seen that the contact map of 2KIV is much sparser, and in the linker of 2KIV, there are virtually no upper bounds, which makes finding the correct fold practically impossible.

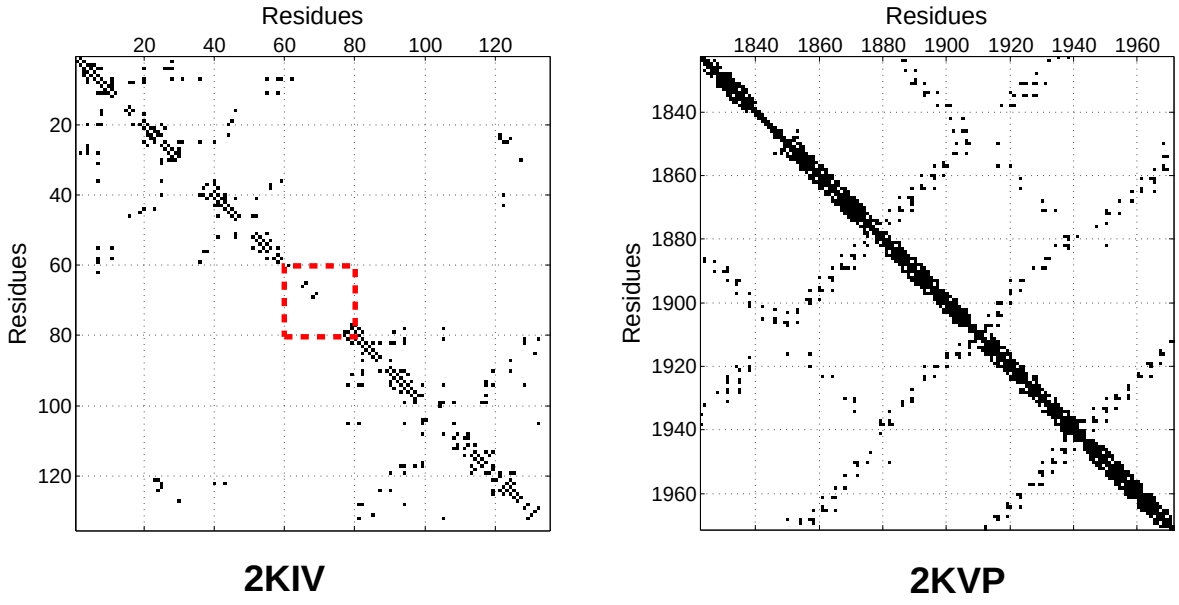


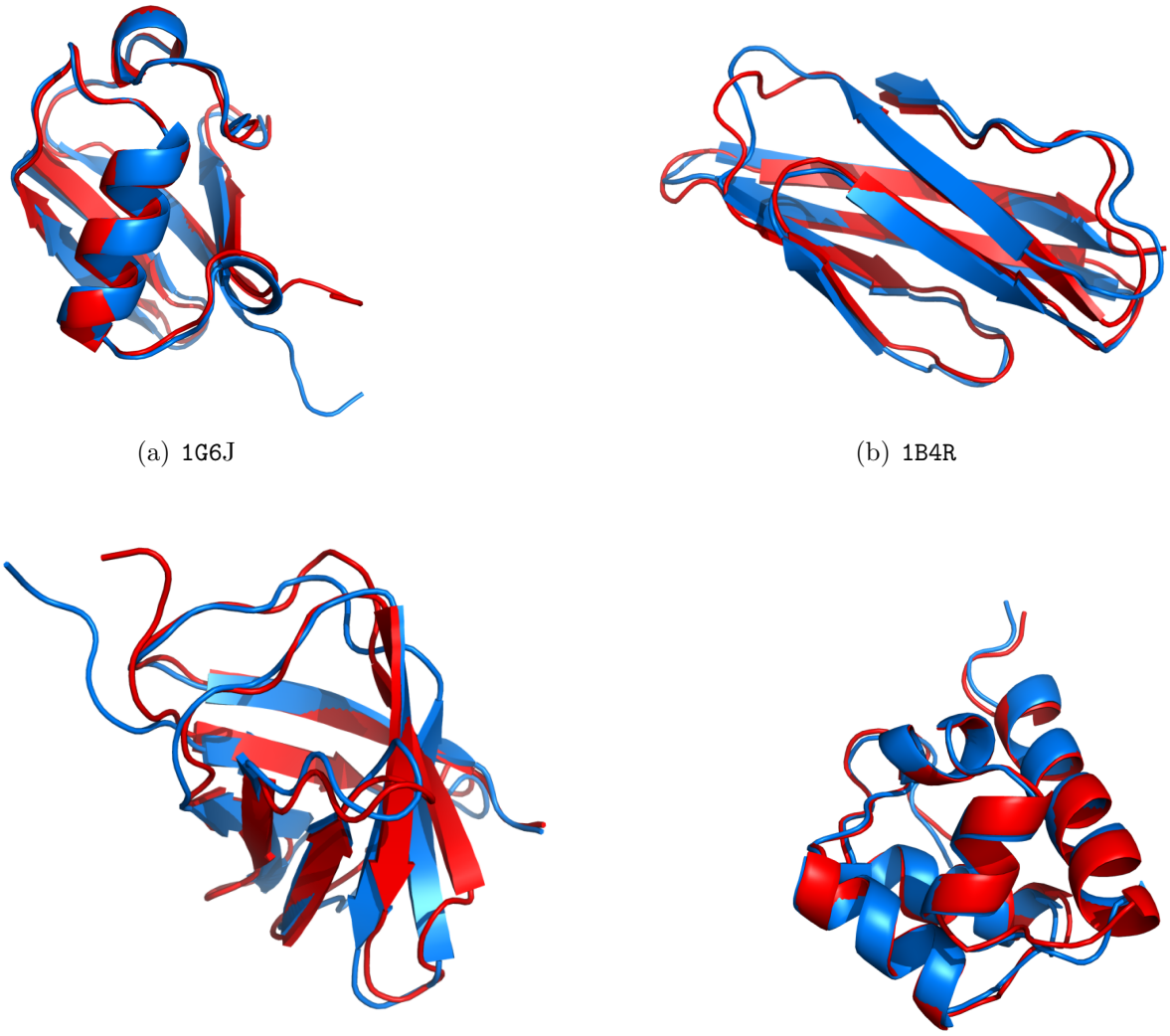
Figure 3.18: Comparison of residue-level contact maps of 2KIV and 2KVP. The linker in 2KIV has virtually no upper bounds (shown by a red dashed square in the contact map).

3.4.3 SPROS Analysis

To evaluate the role that different factors play in the quality of determined structures, we have performed a series of experiments.

Table 3.6: Information about determined structures of the test proteins. The second, third, and fourth columns list SDP time, water refinement time, and total time, respectively. For the backbone and heavy atom RMSD columns, the mean and standard deviation between the determined structure and the reference structures is reported (backbone RMSDs less than 1.5 Å are shown in bold). The seventh column, CBd, lists the number of residues with "CB deviations" larger than 0.25 Åcomputed by MolProbity, as defined by Chen et al. (2010). The eighths and ninth columns list the percentage of upper bound violations larger than 0.1 Å and 1.0 Å, respectively (the numbers for the reference structures are in parentheses). The last three columns, list the percentage of residues with favorable and allowed backbone torsion angles and outliers, respectively.

				RMSD			violations		Ramachandran		
ID	t_s	t_w	t_t	backbone	heavy atoms	CBd.	0.1 Å	1.0 Å	fav.	alw.	out.
1G6J	44.5	175.5	241.0	**0.68±0.05**	0.90±0.05	0	4.96 (0.08±0.07)	0.85 (0)	100	100	0
1B4R	21.4	138.0	179.0	**0.85±0.06**	1.06±0.06	0	20.92 (13.87±0.62)	6.14 (2.28±0.21)	80.8	93.6	6.4
2KT6	45.5	183.4	252.0	**1.46±0.10**	1.89±0.12	4	15.80 (11.72±0.40)	4.55 (3.53±0.6)	62.7	81.9	18.1
2E8O	129.8	181.3	340.9	**0.58±0.02**	0.68±0.01	0	31.33 (31.93±0.14)	9.98 (10.75±0.13)	96.2	100	0
1CN7	75.0	230.1	339.7	1.53±0.11	1.80±0.10	0	10.27 (7.63±0.80)	3.18 (2.11±0.52)	96.1	99.0	1.0
2KTS	116.7	231.0	398.5	**0.92±0.06**	1.13±0.06	0	25.36 (27.44±0.58)	6.49 (10.36±0.68)	86.1	95.7	4.3
2K49	140.7	240.7	422.7	**0.99±0.14**	1.24±0.16	0	13.75 (15.79±0.67)	2.80 (4.94±0.46)	93.8	97.3	2.7
2K62	156.1	259.0	464.2	**1.40±0.08**	1.72±0.08	1	33.74 (42.92±0.95)	10.79 (21.20±1.20)	87.8	95.9	4.1
2L3O	61.7	212.0	310.0	**1.28±0.15**	1.59±0.15	0	21.53 (19.81±0.58)	7.33 (7.61±0.31)	80.4	92.8	7.2
2GJY	113.7	285.9	455.7	**0.99±0.07**	1.29±0.09	0	11.67 (8.36±0.59)	0.36 (0.49±0.12)	85.4	92.3	7.7
2KTE	139.9	297.7	503.2	**1.39±0.17**	1.85±0.16	1	35.55 (31.97±0.46)	11.94 (11.96±0.40)	79.4	90.8	9.2
1XPW	124.8	297.1	489.7	**1.30±0.10**	1.68±0.10	0	9.74 (0.17±0.09)	1.20 (0.01±0.02)	87.9	97.9	2.1
2K7H	211.7	312.0	591.0	**1.24±0.07**	1.49±0.07	0	17.60 (16.45±0.30)	4.39 (4.92±0.35)	92.3	96.1	3.9
2KVP	462.0	282.4	814.8	**0.94±0.08**	1.05±0.09	0	15.15 (17.43±0.29)	4.01 (5.62±0.21)	96.6	100	0
2YTO	292.1	421.5	800.1	**0.79±0.05**	1.04±0.06	1	29.04 (28.9±0.36)	6.64 (6.60±0.30)	90.5	97.6	2.4
2L7B	1101.1	593.0	1992.1	2.15±0.11	2.55±0.11	3	19.15 (21.72±0.36)	4.23 (4.73±0.23)	79.2	91.6	8.4
1Z1V	30.6	158.8	209.2	**1.44±0.17**	1.74±0.15	0	3.89 (2.00±0.25)	0.62 (0)	90.9	98.5	1.5
HACS1	17.4	145.0	176.1	**1.00±0.07**	1.39±0.10	0	20.29 (15.68±0.43)	4.95 (3.73±0.33)	83.6	96.7	3.3
2KIV	79.9	136.8	435.8	2.31±0.20	2.57±0.18	7	45.49 (31.84±0.88)	23.62 (11.03±0.59)	73.7	88.0	12.0
2LJG	94.7	280.4	426.3	**1.24±0.09**	1.70±0.10	1	28.35 (25.3±0.51)	10.76 (8.91±0.49)	80.6	90.7	9.3



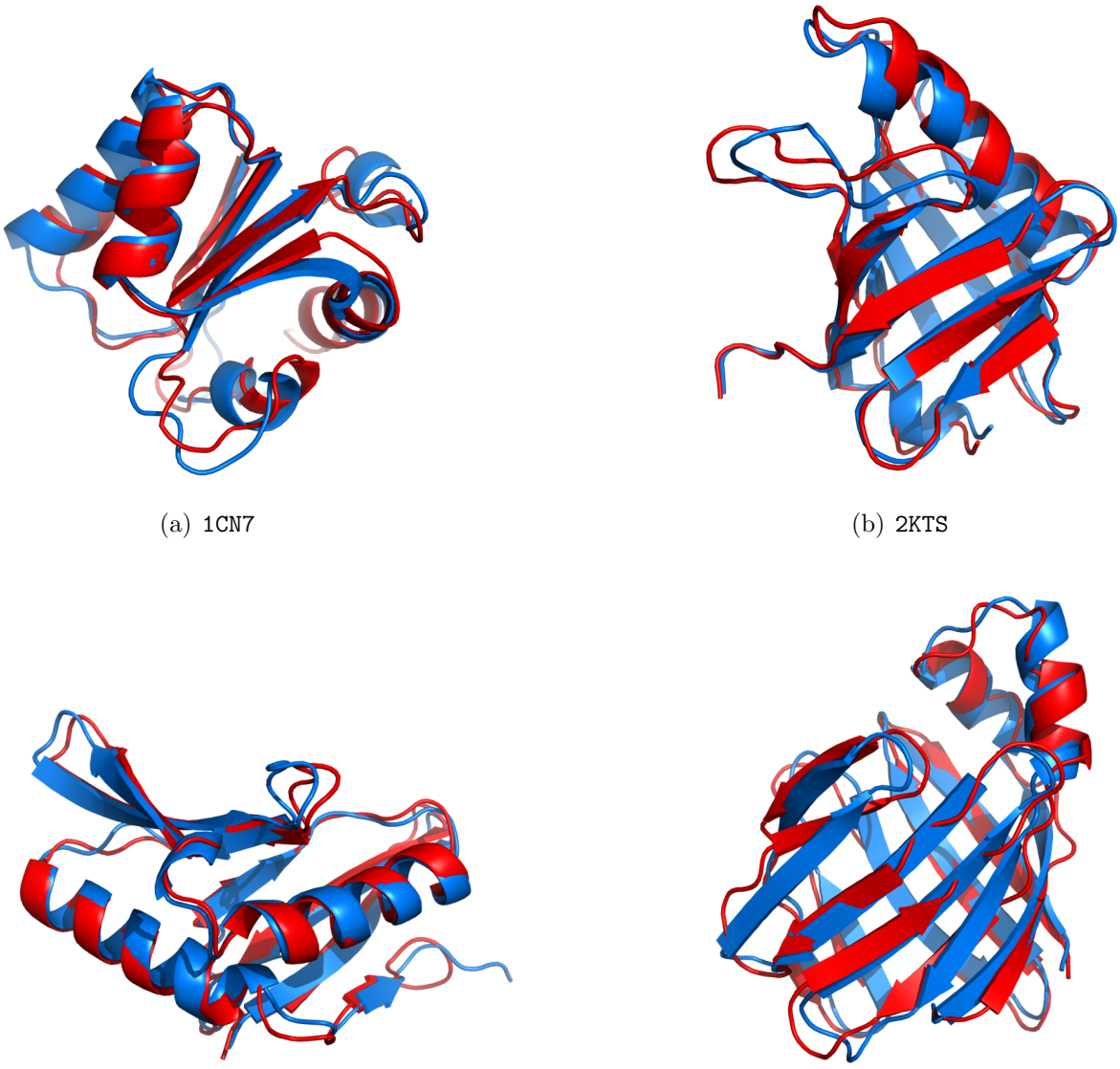
(a) 1G6J

(b) 1B4R

(c) 2KT6

(d) 2E8O

Figure 3.19: Superimposition of structures determined by SPROS in blue and the reference structures in red.



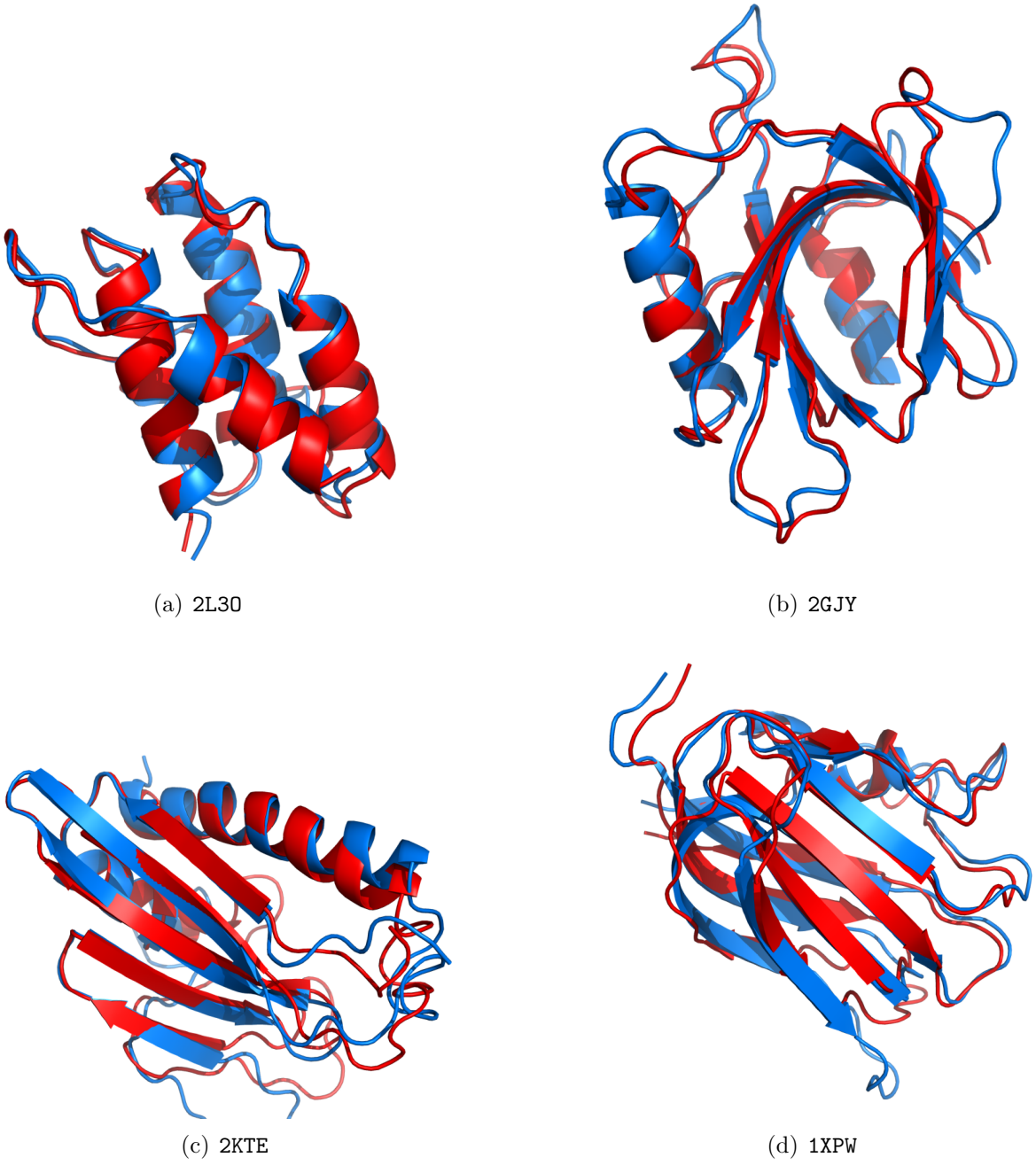
(a) 1CN7

(b) 2KTS

(c) 2K49

(d) 2K62

Figure 3.20: Superimposition of structures determined by SPROS in blue and the reference structures in red.

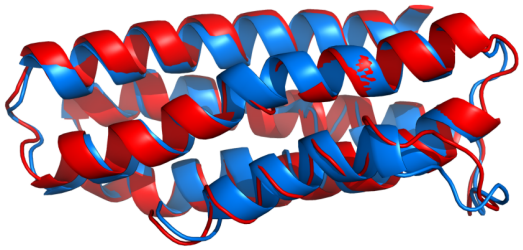


(a) 2L3O (b) 2GJY

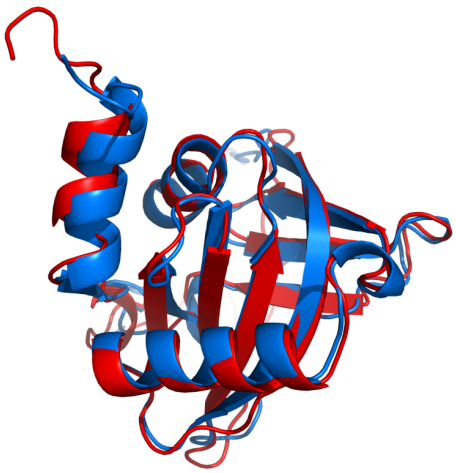
(c) 2KTE (d) 1XPW

Figure 3.21: Superimposition of structures determined by SPROS in blue and the reference structures in red.

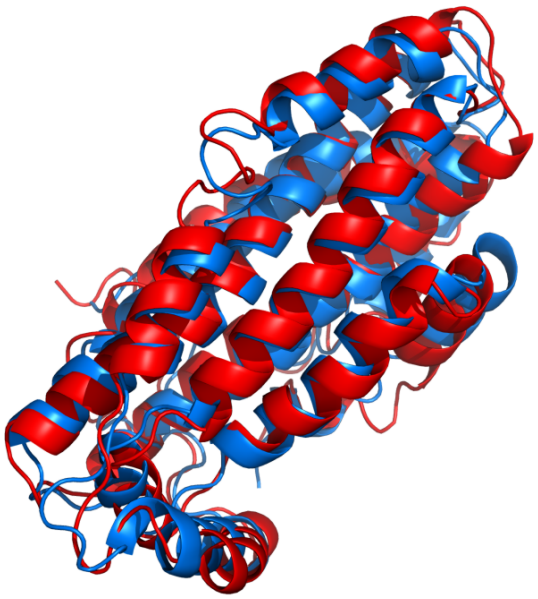
(a) 2K7H



(b) 2KVP

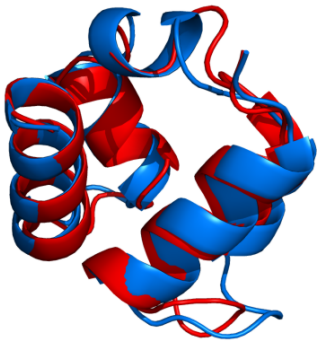


(c) 2YT0

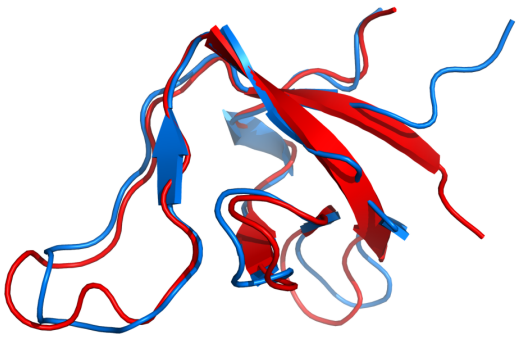


(d) 2L7B

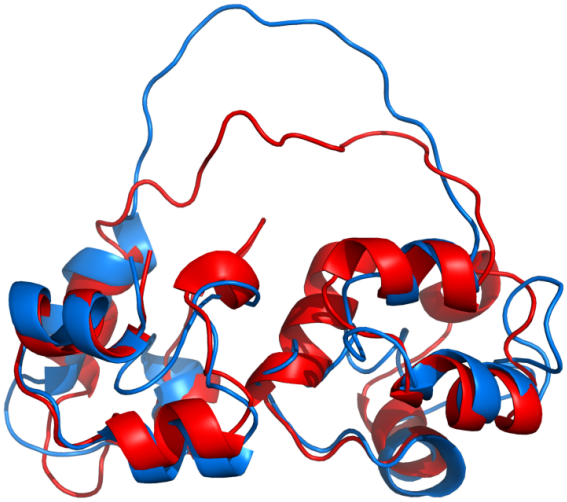
Figure 3.22: Superimposition of structures determined by SPROS in blue and the reference structures in red.



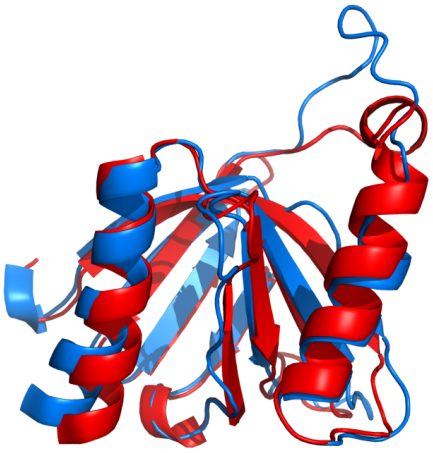
(a) 1Z1V



(b) HACS1



(c) 2KIV



(d) 2LJG

Figure 3.23: Superimposition of structures determined by SPROS in blue and the reference structures in red.

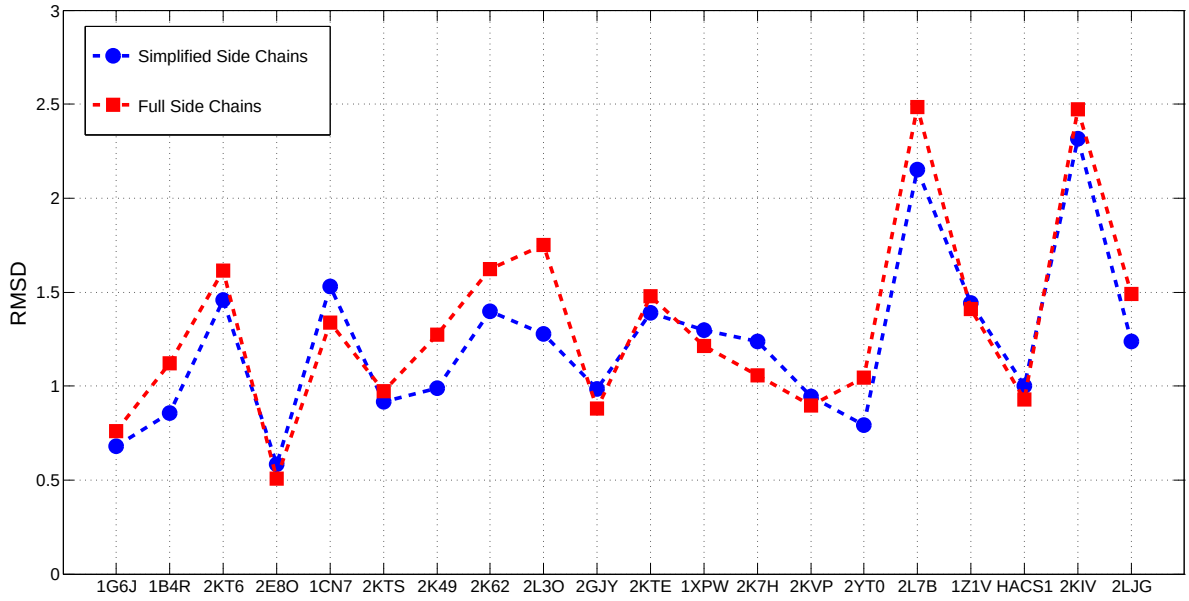


Figure 3.24: RMSD curves for full and simplified side chain cases.

Side chain simplifications

Simplifying side chains reduces SDP matrix size by a factor of two thirds. We solved the structures with full side chains to evaluate the effect of side chain simplification. The RMSDs in two different scenarios is illustrated in Fig. 3.24. It can be seen that the side chain simplification not only speeds up the process, but also generates more accurate structures in most cases. This may be due to the fact that smaller SDP problems are solved more accurately.

Sensitivity to weights

To test the sensitivity of SPROS to the weights used, we ran two experiments: (i) we set all of the weights equal to one; the resultant RMSDs are plotted in Fig. 3.25, and (ii) we varied the α parameter of the regularization weight (see page 120) from -10 to 10. Note that SPROS was designed with $\alpha \in [-1 \ \ +1]$. The resultant RMSD curves are depicted in Fig. 3.26. It can be seen that SPROS is not sensitive to the weights.

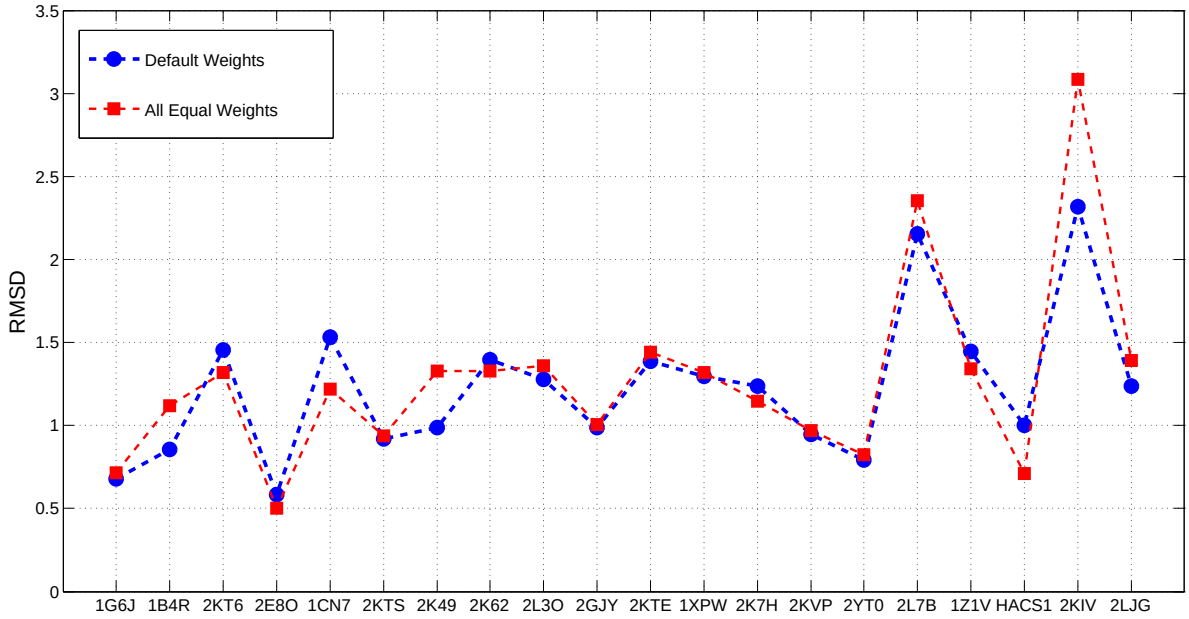


Figure 3.25: RMSD curves when all weights are equal to one.

Constraints

Finally, we tested the roles of torsion angle restraints and intra-residue upper bounds. In Fig. 3.27, the RMSD curves with and without inclusion of torsion angle restraints is plotted. Note that except for a few proteins, there are no noticeable differences between the two scenarios.

In Fig. 3.28, the RMSD curves with and without inclusion of intra-residue upper bounds are plotted. In most cases, there are no notifiable differences in the accuracy of the determined structures. That is, these constraints could be eliminated, which would, on average, cut the SDP solving time in half.

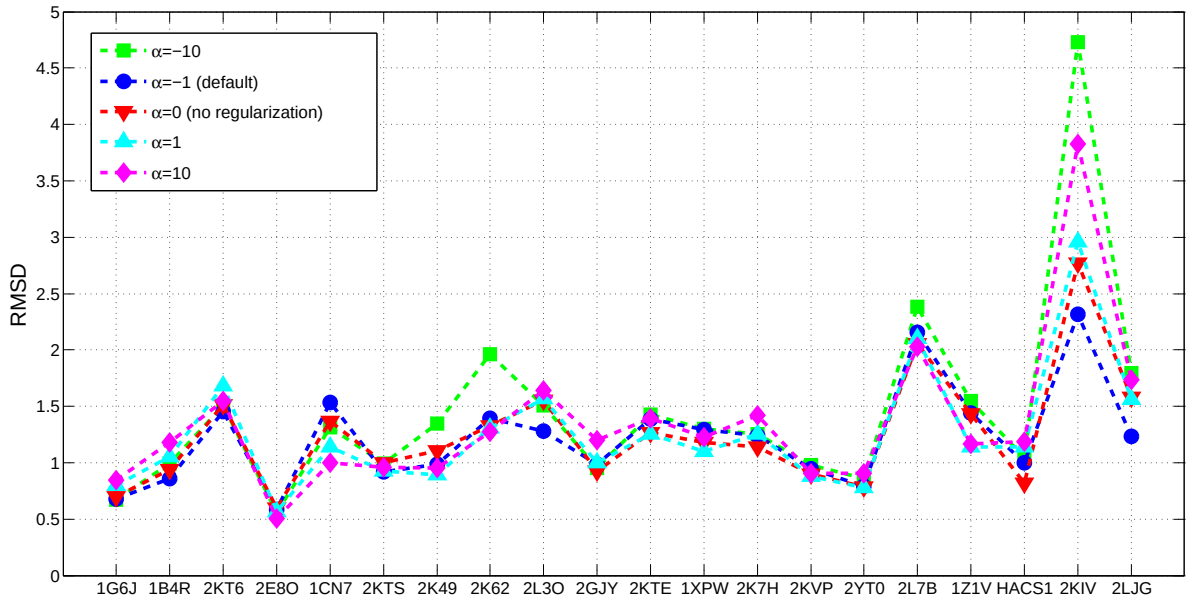


Figure 3.26: RMSD curves for different values of α (w_R coefficient).

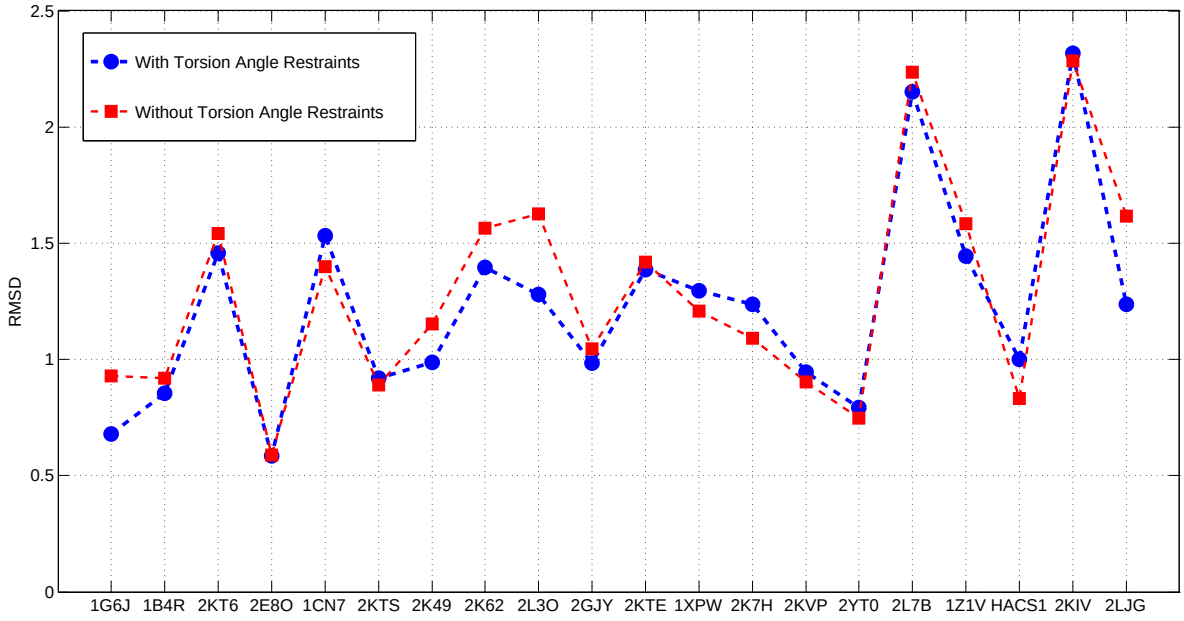


Figure 3.27: RMSD curves for the regular and torsion angle restraints omitted cases.

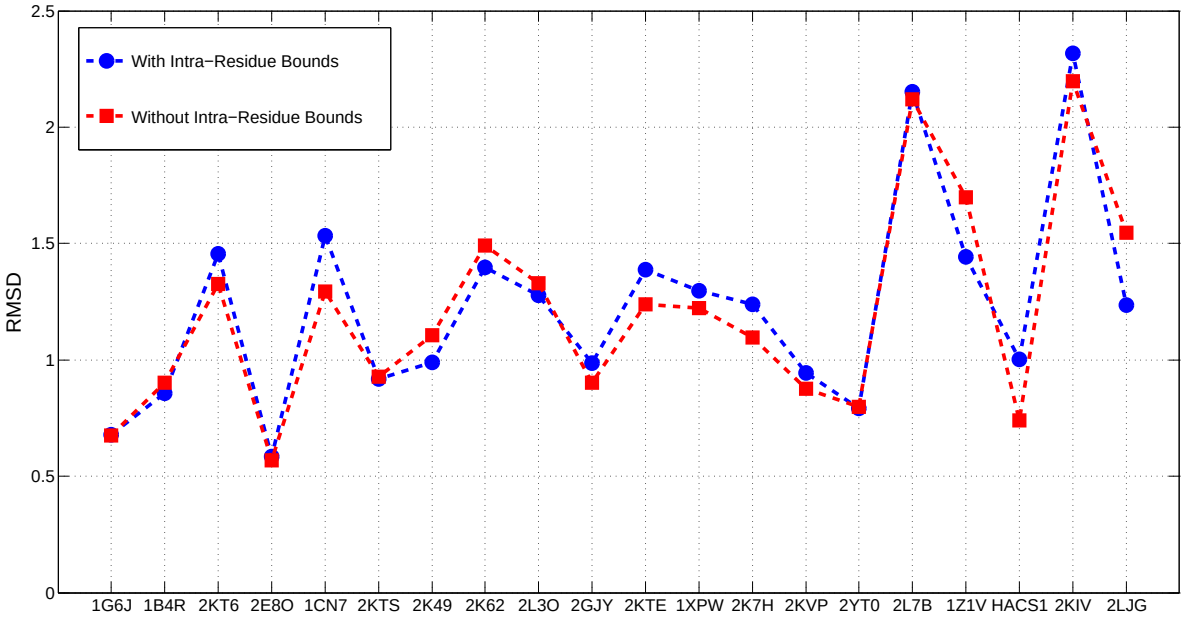


Figure 3.28: RMSD curves for the regular and intra-residue bounds-omitted cases.

Chapter 4

Conclusions and Future Work

We have designed and developed: (*i*) a fully-automated protein NMR structure determination pipeline that can, in less than one day, generate the final structure, starting from NMR spectra and (*ii*) an SDP-based protein structure determination method, called SPROS, that works on real noisy data and generates accurate structures which are on par with, or better than, existing methods.

The combination of a robust peak-picking method, PICKY, the novel spin system forming, the improved spin system typing, the integer linear programming in IPASS, and the novel contact scoring scheme implemented in FALCON-NMR, together result in a robust and error-tolerant protein NMR structure determination system. The automated pipeline was tested on four real protein data sets and it generated reasonably accurate results.

We adapted and extended the semidefinite facial reduction technique, which not only significantly reduces problem size, but also facilitates solving the SDP problem by enforcing strong duality. The reduced SDP problem size is nearly a quarter of the original problem and is solved more efficiently. The post-processing and the water refinement steps fix minor problems in the raw structure, determined by SDP, and improve its stereochemical quality.

We believe that no synthetic protein NMR data set is capable of capturing the intrinsic complexities and uncertainties of real data sets. Therefore, the SPROS method was only tested on real protein NMR data sets of amino acid sequence lengths ranging from 76-307 (weights ranging from 8 to 35 KDa) and, for most proteins generated accurate results. Note that existing programs have been under continuous development for more than twenty years and contain complex iterative protocols. The SPROS method, in its current state can manage many incorrect upper bounds and can efficiently generate structures in only one iteration. We tested SPROS on 20 real protein data sets and it determined structures of the test proteins in a few minutes (the majority in the 3-7 minutes range). Note that water refinement must be run on proteins solved by other methods as well.

4.1 Future Work

1. The SPROS method must be re-implemented in C++ and the SDP problem should be solved by using advanced BLAS packages, such as GotoBLAS2. Moreover, the SPROS must be parallelized and must make use of all available CPU cores. We estimate the speedup factor to be in the range of 50-500, which makes the SPROS method of wider appeal.

2. The SPROS method must be made iterative and, more importantly, it must be merged in the entire protein NMR pipeline. Currently, no NOE peak assignment verification and reassignment, or upper bound calibration is performed. Moreover, all NOE-inferred upper bound violations are weighted equally, while some bounds are more accurate than others. An adaptive violation weight mechanism must be added to SPROS.

3. Because SPROS can manage a large number of incorrect upper bounds, it is an efficient and suitable alternative to FALCON-NMR in the automated protein NMR pipeline. However, SPROS needs reasonably accurate side chain proton chemical shifts; therefore, a side chain assignment method and an NOE contact refinement method should be developed and added to the pipeline. That is, these two methods connect IPASS to SPROS, making a fully-automated protein NMR pipeline.

4. The reduction methodology developed for SPROS is an ideal choice for ligand docking applications. For example, in semi-flexible docking most of the molecular structure of the target protein is assumed to be rigid, and only the side chains of the residues in the active site are flexible and can change conformation. Applying the semidefinite reduction technique results in an SDP matrix size of nearly less than 100, which can be solved in only a few seconds. Therefore, a modification of SPROS can be used for fast, automated screening of ligand libraries.

References

H. Abe, W. Braun, T. Noguti, and N. Go. Rapid calculation of first and second derivatives of conformational energy with respect to dihedral angles for proteins general recurrent equations. *Computers & Chemistry*, 8(4):239–247, 1984. 76

Suliman Al-Homidan and Henry Wolkowicz. Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.*, 406:109–141, 2005. 69

Abdo Y. Alfakih, Amir Khandani, and Henry Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.*, 12 (1-3):13–30, 1999. 68, 69

B. Alipanahi, X. Gao, E. Karakoc, L. Donaldson, and M. Li. PICKY: a novel SVD-based NMR spectra peak picking method. *Bioinformatics*, 25(12):i268–275, 2009. 24, 38, 43

B. Alipanahi, X. Gao, E. Karakoc, S.C. Li, F. Balbach, G. Feng, L. Donaldson, and M. Li. Error tolerant NMR backbone resonance assignment and automated structure generation. *Journal of Bioinformatics and Computational Biology*, 0(1):1–26, 2011a. 24, 38, 39

B. Alipanahi, N. Krislock, and A. Ghodsi. Manifold learning by semidefinite facial reduction. Unpublished manuscript (in preparation), 2011b. 113

J. C. Allwright. Positive semidefinite matrices: characterization via conical hulls and least-squares solution of a matrix equation. *SIAM J. Control Optim.*, 26:537–556, April 1988. 83

Brendan Ames and Stephen Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming*, pages 1–21, May 2011. 89

Michael Andrec, David A. Snyder, Zhiyong Zhou, Jasmine Young, Gaetano T. Montelione, and Ronald M. Levy. A large data set comparison of protein structures determined by crystallography and NMR: statistical test for structural differences and the effect of crystal packing. *Proteins*, 69(3):449–465, November 2007. 20

S. M. Apaydin, V. Conitzer, and B.R. Donald. Structure-based protein NMR assignments using native structural ensembles. *Journal of Biomolecular NMR*, 40:263–276, 2008. 22

S. M. Apaydin, B. Catay, N. Patrick, and B. Donald. NVR-BIP: Nuclear vector replacement using binary integer programming for NMR Structure-Based assignments. *The Computer Journal*, 54(5):708–716, May 2011. 22

A. Arseniev, V. Kondakov, V. Maiorov, and V. Bystrov. NMR solution spatial structure of 'short' scorpion insectotoxin I5A. *FEBS Letters*, 165(1):57–62, 1984. 73

C. Bailey-Kellogg, A. Widge, J. Kelly, J. Brushweller, and B.R. Donald. The NOESY jigsaw: automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. *Journal of Computational Biology*, 7:537–558, 2000. 22

C. Bailey-Kellogg, S. Chainraj, and G. Pandurangan. A random graph approach to NMR sequential assignment. *Journal of Computational Biology*, 12(6):569–583, 2005. 22, 37

G.P. Barker. The lattice of faces of a finite dimensional cone. *Linear Algebra and Its Applications*, 7:71–82, 1973. 82

G.P. Barker. Faces and duality in convex cones. *Linear and Multilinear Algebra*, 6:161–169, 1977. 82

G.P. Barker and D. Carlson. Cones of diagonally dominant matrices. *Pacific J. of Math.*, 57:15–32, 1975. 82

C. Bartels, M. Billeter, P. Güntert, and K. Wüthrich. Automated sequence-specific NMR assignment of homologous proteins using the program garant. *Journal of Biomolecular NMR*, 7:207–213, 1996. 22

R. Bellman and K. Fan. On systems of linear inequalities in hermitian matrix variables. In *Proceedings of Symposia in Pure Mathematics*, volume 7. AMS, 1963. 83

M.V. Berjanskii, S. Neal, and D.S. Wishart. PREDITOR: a web server for predicting protein torsion angle restraints. *Nucleic Acids Research*, 34:63–69, 2006. 60

F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *Journal of molecular biology*, 112(3):535–542, May 1977. 50

Dimitri P. Bertsekas and Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999. 120

Matthew J. Betts and Robert B. Russell. Amino acid properties and consequences of substitutions. pages 289–316, 2003. 56

Martin Billeter, Gerhard Wagner, and Kurt Wüthrich. Solution NMR structure determination of proteins revisited. *Journal of biomolecular NMR*, 42(3):155–158, 2008. 20

Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 26–27 April 2004. ACM. 74

Pratik Biswas, Tzu-Chen Liang, Kim-Chuan Toh, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3:360–371, 2006. 74, 91

Pratik Biswas, Kim-Chuan Toh, and Yinyu Ye. A distributed sdp approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM J. Sci. Comput.*, 30:1251–1277, March 2008. 74, 91, 96

Brian Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999. 84

Brian Borchers and Joseph G. Young. Implementation of a primal-dual method for SDP on a shared memory parallel architecture. *Comput. Optim. Appl.*, 37(3):355–369, 2007. 84

J.M. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Austral. Math. Soc. Ser. A*, 30(3):369–380, 1981a. 82

J.M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981b. 82

J.M. Borwein and H. Wolkowicz. Characterization of optimality for the abstract convex program with finite-dimensional range. *J. Austral. Math. Soc. Ser. A*, 30(4):390–411, 1981c. ISSN 0263-6115. 82

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 79, 81, 82, 83, 89

P. Bradley, K.M.S. Misura, and D. Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309:1868–1871, 2005. 48

Carl Branden and John Tooze. *Introduction to Protein Structure*. Garland Science, 2 edition, January 1999. 56

W. Braun and N. Go. Calculation of protein conformations by proton-proton distance constraints. a new efficient algorithm. *Journal of molecular biology*, 186(3):611–626, 1985. 76

W. Braun, C. Bösch, L. R. Brown, N. Go, and K. Wüthrich. Combined use of proton-proton overhauser enhancements and a distance geometry algorithm for determination of polypeptide conformations. application to micelle-bound glucagon. *Biochimica et biophysica acta*, 667(2):377–396, February 1981. 72

Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, FebFeb 1983. 76, 78

Axel T. Brünger. *X-PLOR Version 3.1: A System for X-ray Crystallography and NMR*. Yale University Press, 1993. 78

E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936, June 2010. 89

Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 89

John Cavanagh, Wayne J. Fairbrother, Arthur G. Palmer, Nicholas J. Skelton, and Mark Rance. *Protein NMR Spectroscopy, Second Edition: Principles and Practice*. Academic Press, 2 edition, 2006. 1, 2, 4, 6, 7, 8, 13, 17

John-Marc M. Chandonia and Steven E. Brenner. The impact of structural genomics: expectations and outcomes. *Science*, 311(5759):347–351, 2006. 20

Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel a Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta crystallographica. Section D, Biological crystallography*, 66(Pt 1):12–21, January 2010. 55, 64, 71, 126, 129

G. M. Clore, M. Nilges, D. K. Sukumaran, A. T. Brünger, M. Karplus, and A. M. Gronenborn. The three-dimensional structure of alpha1-purothionin in solution: combined use of nuclear magnetic resonance, distance geometry and restrained molecular dynamics. *The EMBO journal*, 5(10):2729–2735, October 1986. 76

B. Coggins and P. Zhou. PACES: protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR*, 26:93–111, 2003. 22

G. Cornilescu, F. Delaglio, and A. Bax. Protein backbone angle restraints from searching a database for chemical shift and sequence homology. *Journal of Biomolecular NMR*, 13:289–302, 1999. 60

B. Craven. Linear programming with matrix variables. *Linear Algebra and its Applications*, 38(1):73–80, June 1981. 83

Thomas E. Creighton. *Proteins: Structures and Molecular Properties*. W. H. Freeman, second edition edition, August 1992. 56

Gordon M Crippen, Aikaterini Rousaki, Matthew Revington, Yongbo Zhang, and Erik R P Zuiderweg. SAGA: rapid automatic mainchain NMR assignment for large proteins. *Journal of biomolecular NMR*, 46(4):281–98, 2010. 22, 23, 34, 35

L. Doherty, K. S. J. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663 vol.3, 2001. 74

B. Donald and J. Martin. Automated NMR assignment and protein structure determination using sparse dipolar coupling constraints. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 55(2):101–127, 2009. 22

L. W. Donaldson. The NMR structure of the *Staphylococcus aureus* response regulator VraR DNA binding domain reveals a dynamic relationship between it and its associated receiver domain. *Biochemistry*, 47(11):3379–3388, 2008. 44

Jurgen F. Doreleijers, Steve Mading, Dimitri Maziuk, Kassandra Sojourner, Lei Yin, Jun Zhu, John L. Markley, and Eldon L. Ulrich. BioMagResBank database with sets of experimental NMR constraints corresponding to the structures of over 1400 biomolecules deposited in the protein data bank. *Journal of biomolecular NMR*, 26(2):139–146, June 2003. 123

Jurgen F. Doreleijers, Aart J. Nederveen, Wim Vranken, Jundong Lin, Alexandre M. Bonvin, Robert Kaptein, John L. Markley, and Eldon L. Ulrich. BioMagResBank databases DOCR and FRED containing converted and filtered sets of experimental NMR restraints and coordinates from over 500 protein PDB structures. *Journal of biomolecular NMR*, 32(1):1–12, May 2005. 123

Jan Drenth. *Principles of Protein X-ray Crystallography (Springer Advanced Texts in Chemistry)*. Springer, 2nd edition, 2002. 63

H. R. Eghbalnia, A. Bahrami, L. Wang, A. Assadi, and J. L. Markley. Probabilistic identification of spin systems and their assignments including coil-helix inference as output (PISTACHIO). *Journal of Biomolecular NMR*, 32(3):219–233, 2005. 22, 37

R. a. Engh and R. Huber. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallographica Section A Foundations of Crystallography*, 47(4):392–400, 1991. 62

M.A. Erdmann and G.S. Rule. Rapid protein structure detection and assignment using residual dipolar couplings. *Technical Report CMU-CS-02-195, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA*, 2002. 22

M Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002. 89

F. Fiorito, S. Hiller, G. Wider, and K. Wüthrich. Automated resonance assignment of protein: 6D APSY-NMR. *Journal of Biomolecular NMR*, 35:27–37, 2006. 23

R Fletcher. Semi-definite matrix constraints in optimization. *SIAM Journal on Control and Optimization*, 23(4):493–513, 1985. 83

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, 3rd edition, 1996. 107, 108

Frank Göring, Christoph Helmberg, and Markus Wappler. The rotational dimension of a graph. *J. Graph Theory*, 66(4):283–302, 2011. 90

A. Grishaev, C. A. Steren, B. Wu, A. Pineda-Lucena, C.H. Arrowsmith, and M. Llinás. ABACUS, a direct method for protein NMR structure computation via assembly of fragments. *Proteins*, 61:36–43, 2005. 31

W. Gronwald, L. Willard, T. Jellard, R.F. Boyko, K. Rajarathnam, D.S. Wishart, F.D. Sönnichsen, and B.D. Sykes. Camra: chemical shift based computer aided protein NMR assignments. *Journal of Biomolecular NMR*, 12:395–405, 1998. 22

Ian G. Grooms, Robert Michael Lewis, and Michael W. Trosset. Molecular embedding via a second order dissimilarity parameterized approach. *SIAM J. Scientific Computing*, 31 (4):2733–2756, 2009. 73

P. Güntert. Structure calculation of biological macromolecules from NMR data. *Quarterly reviews of biophysics*, 31(2):145–237, 1998. 59, 60, 62, 70, 73, 76, 101

P. Güntert. Automated NMR structure calculation with CYANA. *Methods in Molecular Biology*, 278:353–378, 2004. 57, 58, 62, 78, 79, 102

P. Güntert, W. Braun, and K. Wüthrich. Efficient computation of three-dimensional protein structures in solution from nuclear magnetic resonance data using the program DIANA and the supporting programs CALIBA, HABAS and GLOMSA. *Journal of molecular biology*, 217(3):517–530, February 1991. 57

P. Güntert, C. Mumenthaler, and K. Wüthrich. Torsion angle dynamics for NMR structure calculation with the new program DYANA. *Journal of molecular biology*, 273:283–298, 1997. 78

P. Güntert, M. Salzmann, D. Braun, and K. Wüthrich. Sequence-specific NMR assignment of proteins by global fragment mapping with the program MAPPER. *Journal of Biomolecular NMR*, 18:129–137, 2000. 22

Peter Güntert. Automated structure determination from NMR spectra. *European biophysics journal : EBJ*, 38(2):129–43, 2009. 21, 22

James M. Hart, Scott D. Kennedy, David H. Mathews, and Douglas H. Turner. NMR-assisted prediction of RNA secondary structure: identification of a probable pseudoknot in the coding region of an R2 retrotransposon. *Journal of the American Chemical Society*, 130(31):10233–10239, 2008. 20

Timothy F. Havel and K. Wüthrich. A Distance Geometry Program for Determining the Structures of Small Proteins and Other Macromolecules From Nuclear Magnetic Resonance Measurements of Intramolecular H-H Proxmities in Solution. *Bulletin of Mathematical Biology*, 46(4):673–698, 1984. 73

Timothy F. Havel, Irwin D. Kuntz, and Gordon M. Crippen. The theory and practice of distance geometry. *Bull. Math. Biol.*, 45(5):665–720, 1983. 73

C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361, 1996. 83

Torsten Herrmann, Peter Güntert, and Kurt Wüthrich. Protein NMR structure determination with automated NOE assignment using the new software CANDID and the torsion angle dynamics algorithm DYANA. *Journal of molecular biology*, 319(1):209–227, May 2002. 57, 58, 59

S. Hiller, F. Fiorito, K. Wüthrich, and G. Wider. Automated projection spectroscopy (APSY). *Proceedings of the National Academy of Sciences*, 102:10876–10881, 2005. 23

R. W. W. Hooft, C. Sander, and G. Vriend. Verification of Protein Structures: Side-Chain Planarity. *Journal of Applied Crystallography*, 29(6):714–716, 1996. 63

P. J. Hore. *Nuclear Magnetic Resonance (Oxford Chemistry Primers, 32)*. Oxford University Press, USA, 1 edition, 1995. 3, 6, 7, 8, 9, 41

Yuanpeng Janet J. Huang, Hunter N. Moseley, Michael C. Baran, Cheryl Arrowsmith, Robert Powers, Roberto Tejero, Thomas Szyperski, and Gaetano T. Montelione. An integrated platform for automated analysis of protein NMR structures. *Methods in enzymology*, 394:111–141, 2005. 21

J.C. Hus, J. Prompers, and R. Brüschweiler. Assignment strategy for proteins with known structure. *Journal of Magnetic Resonance*, 157:119–123, 2002. 22

Wolfgang Jahnke. Perspectives of biomolecular NMR in drug discovery: the blessing and curse of versatility. *Journal of Biomolecular NMR*, 39(2):87–90, 2007. 20

A. Jain, N. Vaidehi, and G. Rodriguez. A fast recursive algorithm for molecular dynamics simulation. *J. Comput. Phys.*, 106:258–268, June 1993. 78

David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research*, 37(6):865–892, November 1989. 78

David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. *Operations Research*, 39(3):378–406, May 1991. 78

Y. Jung and M. Zweckstetter. Mars - robust automatic backbone assignment of proteins. *Journal of Biomolecular NMR*, 30:11–23, 2004. 22, 23

Y. S. Jung, M. Sharma, and M. Zweckstetter. Simultaneous assignment and structure determination of protein backbones by using NMR dipolar couplings. *Angewandte Chemie (International ed. in English)*, 43(26):3479–3481, 2004. 21

W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1993. 28

R. Kaptein, E.R.P. Zuiderweg, Scheek R.M., R. Boelens, and W.F. van Gunsteren. A protein structure from nuclear magnetic resonance data lac repressor headpiece. *Journal of Molecular Biology*, 182(1):179–182, 1985. 49, 76

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. 87

James Keeler. *Understanding NMR Spectroscopy*. Wiley, 2 edition, 2010. 1

J. C. Kendrew. Architecture of a protein molecule. *Nature*, 182:764–767, 1958. 49

P. Khajehpour, B. Alipanahi, and A. Ghodsi. Tesseramap: Unfolding manifolds by tessellation of linear subspaces. Unpublished manuscript (in preparation), 2011. 86

Daisuke Kihara, Hao Chen, and Yifeng David D. Yang. Quality assessment of protein structure models. *Current protein & peptide science*, 10(3):216–228, June 2009. 65

Sunyoung Kim, Masakazu Kojima, and Hayato Waki. Exploiting sparsity in SDP relaxation for sensor network localization. *SIAM Journal on Optimization*, 20(1):192–215, 2009. 74

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983. 77

M. Kojima, S. Kojima, and S. Hara. Linear algebra for semidefinite programming. Technical Report 1004, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1997. Linear matrix inequalities and positive semidefinite programming (Japanese) (Kyoto, 1996). 83

Nathan Krislock. *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion*. PhD thesis, University of Waterloo, 2010. 65, 82, 83, 86, 103, 105, 110, 115

Nathan Krislock and Henry Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM J. on Optimization*, 20:2679–2708, 2010. 74, 86, 103, 110

Arwa Kurabi, Stacey Brener, Mehdi Mobli, Jamie J. Kwan, and Logan W. Donaldson. A nuclear localization signal at the SAM-SAM domain interface of AIDA-1 suggests a requirement for domain uncoupling prior to nuclear import. *Journal of molecular biology*, 392(5):1168–1177, 2009. 125

J. Kuszewski, A. M. Gronenborn, and G. M. Clore. Improving the quality of NMR and crystallographic protein structures by means of a conformational database potential derived from structure databases. *Protein Sci*, 5(6):1067–1080, 1996. 122

J. Kuszewski, A. M. Gronenborn, and G. M. Clore. Improvements and extensions in the conformational database potential for the refinement of NMR and x-ray structures of proteins and nucleic acids. *Journal of magnetic resonance*, 125(1):171–177, March 1997. 122

Jamie J. Kwan, Neil Warner, Joban Maini, Kelvin W. Chan Tung, Hoshang Zakaria, Tony Pawson, and Logan W. Donaldson. Saccharomyces cerevisiae ste50 binds the MAPKKK ste11 through a head-to-tail SAM domain interaction. *Journal of molecular biology*, 356 (1):142–154, February 2006. 125

C.J. Langmead and B.R. Donald. An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *Journal of Biomolecular NMR*, 29:111–138, 2004. 22

C.J. Langmead, A. Yan, R. Lilien, L. Wang, and B.R. Donald. A polynomial-time nuclear vector replacement algorithm for automated NMR resonance assignment. *Journal of Computational Biology*, 11:277–298, 2004. 22

R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26(2):283–291, April 1993. 64

R. A. Laskowski, J. A. Rullmannn, M. W. MacArthur, R. Kaptein, and J. M. Thornton. AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR. *Journal of biomolecular NMR*, 8(4):477–486, December 1996. 64

A. Lemak, C.A. Steren, C.H. Arrowsmith, and M. Llinás. Sequence specific resonance assignment via Multicanonical Monte Carlo search using an ABACUS approach. *Journal of Biomolecular NMR*, 41:29–41, 2008. 22, 23

Ngai-Hang Z. Leung and Kim-Chuan Toh. An sdp-based divide-and-conquer algorithm for large-scale noisy anchor-free graph realization. *SIAM J. Sci. Comput.*, 31:4351–4372, 2009. 75, 76, 91, 96

Malcolm H. Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance*. Wiley, 2nd edition, 2008. 3, 4, 6, 10

A.S. Lewis and M.L. Overton. Nonsmooth optimization via bfgs. *Submitted to SIAM J. Optimization*, 2009. 126

S. C. Li, D. Bu, J. Xu, and M. Li. Fragment-HMM: a new approach to protein structure prediction. *Protein Science*, 2008. 38

W. Li and A. Godzik. CD-HIT: a fast program for clustering and comparing large sets of protein for nucleotide sequences. *Bioinformatics*, 22:1658–1659, 2006. 28

Jens P. Linge, Michael Habeck, Wolfgang Rieping, and Michael Nilges. ARIA: automated NOE assignment and NMR structure calculation. *Bioinformatics*, 19(2):315–316, January 2003. 57, 122

Adam Liwo, Jooyoung Lee, Daniel R. Ripoll, Jaroslaw Pillardy, and Harold A. Scheraga. Protein structure prediction by global optimization of a potential energy function. *Proceedings of the National Academy of Sciences*, 96(10):5482–5485, May 1999. 75

B. López-Méndez and P. Güntert. Automated protein structure determination from NMR spectra. *Journal of the American Chemical Society*, 128(40):13112–13122, 2006. 21, 38

Simon C. Lovell, Ian W. Davis, W. Bryan Arendall, Paul I. W. de Bakker, J. Michael Word, Michael G. Prisant, Jane S. Richardson, and David C. Richardson. Structure validation by C_α geometry: ϕ, ψ and C_β deviation. *Proteins: Structure, Function, and Genetics*, 50(3):437–450, February 2003. 55

Kanti V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis (Probability and Mathematical Statistics)*. Academic Press, first edition, second impression edition, February 1980. 119, 120

John L. Markley, Ad Bax, Yoji Arata, C. W. Hilbers, Robert Kaptein, Brian D. Sykes, Peter E. Wright, and Kurt Wüthrich. Recommendations for the presentation of NMR structures of proteins and nucleic acids. *European Journal of Biochemistry*, 256(1):1–15, 1998. 13, 56

J.E. Masse and R. Keller. Autolink: automated sequential resonance assignment of biopolymers from NMR data by relative-hypothesis-prioritization-based simulated logic. *Journal of Magnetic Resonance*, 174:133–151, 2005. 22, 23

J. Andrew McCammon, Bruce R. Gelin, and Martin Karplus. Dynamics of folded proteins. *Nature*, 267(5612):585–590, June 1977. 76

L. J. McGuffin, K. Bryson, and D. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000. 30

R.D.C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. 7 (3):663–678, 1997. 83

Jorge J. Moré and Zhijun Wu. Global continuation for distance geometry problems. *SIAM J. on Optimization*, 7:814–836, March 1997. 75

Jorge J. Moré and Zhijun Wu. Distance geometry optimization for protein structures. *J. of Global Optimization*, 15:219–234, October 1999. ISSN 0925-5001. 75

A.L. Morris, M.W. MacArthur, E.G. Hutchinson, and J.M. Thornton. Stereochemical quality of protein structure coordinates. *Proteins: Structure, Function, and Bioinformatics*, 12(4):345–364, 1992. ISSN 1097-0134. 64, 122

Laura C. Morris, Homayoun Valafar, and James H. Prestegard. Assignment of protein backbone resonances using connectivity, torsion angles and 13c alpha chemical shifts. *Journal of biomolecular NMR*, 29(1):1–9, May 2004. 22

H. Moseley and G. Montelione. Automated analysis of NMR assignments and structures for proteins. *Current Opinion in Structural Biology*, 9:635–642, 1999. 22

C. Mumenthaler, P. Güntert, W. Braun, and K. Wüthrich. Automated combined assignment of NOESY spectra and three-dimensional protein structure determination. *Journal of biomolecular NMR*, 10(4):351–362, December 1997. 57

Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 1994. 83

Y.E. Nesterov and A.S. Nemirovski. A general approach to polynomial-time algorithms design for convex programming. Technical report, Centr. Econ. & Math. Inst., USSR Acad. Sci., USSR Academy of Sciences, Moscow, USSR, 1988. 83

Y.E. Nesterov and A.S. Nemirovski. *Optimization over positive semidefinite matrices: Mathematical background and user's manual*. USSR Acad. Sci. Centr. Econ. & Math. Inst., 32 Krasikova St., Moscow 117418 USSR, 1990. 83

Y.E. Nesterov and A.S. Nemirovski. Conic formulation of a convex programming problem and duality. Technical report, Centr. Econ. & Math. Inst., USSR Academy of Sciences, Moscow, USSR, 1991. 83

M. Nilges, G. M. Clore, and A. M. Gronenborn. Determination of three-dimensional structures of proteins from interproton distance data by hybrid distance geometry-dynamical simulated annealing calculations. *FEBS letters*, 229(2):317–324, March 1988. 77

M. Nilges, M. J. Macias, S. I. O'Donoghue, and H. Oschkinat. Automated NOESY interpretation with ambiguous distance restraints: the refined NMR solution structure of the pleckstrin homology domain from beta-spectrin. *Journal of molecular biology*, 269(3): 408–422, June 1997. 57

Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2000. 120

Gregory A. Petsko and Dagmar Ringe. *Protein Stucture and Function (Primers in Biology)*. New Science Press, Ltd., 1 edition, 2003. 56, 60

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, January 1964. 73

P. Pristovsek and L. Franzoni. Stereospecific assignments of protein NMR resonances based on the tertiary structure and 2D/3D NOE data. *Journal of Computational Chemistry*, 27:791–797, 2004. 22

P. Pristovsek, H. Rüterjans, and R. Jerala. Semiautomatic sequence-specific assignment of proteins based on the tertiary structure - the program st2nmr. *Journal of Computational Chemistry*, 23:335–340, 2002. 22

G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99, July 1963. 55

Srivatsan Raman, Oliver F. Lange, Paolo Rossi, Michael Tyka, Xu Wang, James Aramini, Gaohua Liu, Theresa A. Ramelot, Alexander Eletsky, Thomas Szyperski, Michael A. Kennedy, James Prestegard, Gaetano T. Montelione, and David Baker. NMR structure determination for larger proteins using Backbone-Only data. *Science*, 327(5968):1014–1018, February 2010. 39

Motakuri V. Ramana, Levent Tunçel, and Henry Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997. 80, 85

B. Recht, Weiyu Xu, and B. Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. pages 3065–3070, 2008. 89

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed Minimum-Rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52:471–501, 2010. 89

J.B. Saxe. Embeddability of weighted graphs in k-space is strongly NP-hard. In *Proceedings of 17th Allerton Conference in Communications, Control and Computing, Allerton*, pages 480–489, 1979. 87

I. J. Schoenberg. Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espace distanciés vectoriellement applicable sur l'espace de Hilbert". *Ann. of Math. (2)*, 36(3):724–732, 1935. 68, 74

Peter Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 65

C.D. Schwieters, J.J. Kuszewski, N. Tjandra, and G.M. Clore. The Xplor-NIH NMR molecular structure determination package. *Journal of Magnetic Resonance*, 160:65–73, 2003. 78

C.D. Schwieters, J.J. Kuszewski, and G.M. Clore. Using Xplor-NIH for NMR molecular structure determination. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 48:47–62, 2006. 78

B.R. Seavey, E.A. Farr, W.M. Westler, and J. Markley. A relational database for sequence-specific protein NMR data. *Journal of Biomolecular NMR*, 1:217–236, 1991. 23

A. Shapiro. Extremal problems on the set of nonnegative definite matrices. *Linear Algebra and its Applications*, 67:7–18, June 1985. 83

B. Shaw and T. Jebara. Minimum volume embedding. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics March 21-24, 2007, San Juan, Puerto Rico*, volume 2 of JMLR: W&CP, pages 460–467, March 2007. 91

Blake Shaw and Tony Jebara. Structure preserving embedding. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 937–944, Montreal, June 2009. Omnipress. 91

Y. Shen, O. Lange, F. Delaglio, P. Rossi, J. M. Aramini, G. Liu, A. Eletsky, Y. Wu, K. K. Singarapu, A. Lemak, A. Ignatchenko, C. H. Arrowsmith, T. Szyperski, G. T. Montelione, D. Baker, and A. Bax. Consistent blind protein structure generation from NMR chemical shift data. *Proceedings of the National Academy of Sciences of the United States of America*, 105(12):4685–4690, 2008. 21, 39

Y. Shen, R. Vernon, D. Baker, and A. Bax. De novo protein structure generation from incomplete chemical shift assignments. *Journal of Biomolecular NMR*, 43(2):63–78, 2009a. 21

Yang Shen, Frank Delaglio, Gabriel Cornilescu, and Ad Bax. TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. *Journal of biomolecular NMR*, 44(4):213–223, August 2009b. 60

C. Spronk, S. Nabuurs, E. Krieger, G. Vriend, and G. Vuister. Validation of protein structures derived by NMR spectroscopy. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 45(3-4):315–337, 2004. 20

D. Stratmann, C. Heijenoort, and E. Guittet. NOEnet - use of NOE networks for NMR resonance assignment of proteins with known 3D structure. *Bioinformatics*, 25(4):474–481, 2009. 22

Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999. 84

Joel L. Sussman. *Constrained-restrained least-squares (CORELS) refinement of proteins and nucleic acids*, volume 115, pages 271–303. Elsevier, 1985. 63, 93

F. Tian, H. Valafar, and J. H. Prestegard. A dipolar coupling based strategy for simultaneous resonance assignment and structure determination of protein backbones. *Journal of the American Chemical Society*, 123(47):11791–11796, 2001. 22

W.S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4): 401–419, 1952. 70

D. E. Tronrud, L. F. Ten Eyck, and B. W. Matthews. An efficient general-purpose least-squares refinement program for macromolecular structures. *Acta Crystallographica Section A*, 43(4):489–501, 1987. 63

M. W. Trosset. Applications of multidimensional scaling to molecular conformation. *Computing Science and Statistics*, 29:148–152, 1998. 73

R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95(2, Ser. B):189–217, 2003. 84, 126

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38 (1):49–95, 1996. 83, 88

V. Černý. A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, January 1985. 77

Olga Vitek, Chris Bailey-Kellogg, Bruce Craig, Paul Kuliniewicz, and Jan Vitek. Reconsidering complete search algorithms for protein backbone NMR assignment. *Bioinformatics*, 21(suppl_2):ii230–236, 2005. 22

J. Volk, T. Herrmann, and K. Wüthrich. Automated sequence-specific protein NMR assignment using the memetic algorithm MATCH. *Journal of Biomolecular NMR*, 41: 127–138, 2008. 22, 23

G. Wagner and K. Wüthrich. Amide proton exchange and surface conformation of the basic pancreatic trypsin inhibitor in solution studies with two-dimensional nuclear magnetic resonance. *Journal of Molecular Biology*, 160(2):343–361, September 1982. 59

Stefan Wallin, Jochen Farwer, and Ugo Bastolla. Testing similarity measures with continuous and discrete protein models. *Proteins*, 50(1):144–157, January 2003. 65

X. Wan and G. Lin. CISA: combined NMR resonance connectivity information determination and sequential assignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:336–348, 2007. 22, 42, 43, 44

Zizhuo Wang, Song Zheng, Yinyu Ye, and Stephen Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008. 74

Kilian Q. Weinberger and Lawrence K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:988–995, 2004. 90

Kilian Q. Weinberger and Lawrence K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vision*, 70:77–90, October 2006. ISSN 0920-5691. 91

K.Q. Weinberger, F. Sha, Q. Zhu, and L.K. Saul. Graph Laplacian regularization for large-scale semidefinite programming. *Advances in neural information processing systems*, 19: 1489, 2007. 91

G. A. Williams, J. M. Dugan, and R. B. Altman. Constrained global optimization for estimating molecular structure from atomic distances. *Journal of computational biology*, 8(5):523–547, 2001. 75

M. P. Williamson, T. F. Havel, and K. Wüthrich. Solution conformation of proteinase inhibitor IIA from bull seminal plasma by 1H nuclear magnetic resonance and distance geometry. *Journal of molecular biology*, 182(2):295–315, March 1985. 49, 73

Mike P Williamson and C Jeremy Craven. Automated protein structure calculation from NMR data. *Journal of biomolecular NMR*, 43(3):131–43, 2009. 20, 21, 22, 122

Henry Wolkowicz. Some applications of optimization in matrix theory. *Linear Algebra and its Applications*, 40:101–118, October 1981. 83

J. M. Word, S. C. Lovell, J. S. Richardson, and D. C. Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of molecular biology*, 285(4):1735–1747, January 1999. 64

K. Wu, J. Chang, J. Chen, C. Chang, W. Wu, T. Huang, T. Sung, and W. Hsu. RIBRA: an error-tolerant algorithm for the NMR backbone assignment problem. *Journal of Computational Biology*, 13(2):229–244, 2006. 22, 23

Kurt Wüthrich. *NMR of Proteins and Nucleic Acids*. John Wiley & Sons, 1986. 7, 17

Y. Xia, A. Yee, A. Semesi, and C.H. Arrowsmith. Solution structure of hypothetical protein TM1112. *PDB Database*, 2002. 44

F. Xiong and C. Bailey-Kellogg. A hierarchical grow-and-match algorithm for backbone resonance assignments given 3D structure. In *The 7th IEEE International Conference on Bioinformatics and Bioengineering*, pages 403–410, Boston, MA, 2007. 22

F. Xiong, G. Pandurangan, and C. Bailey-Kellogg. Contact replacement for NMR resonance assignment. *Bioinformatics*, 24:i205–i213, 2008. 22

J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1:95–117, 2003. 38

Y. Xu, D. Xu, D. Kim, V. Olman, J. Razumovskaya, and T. Jiang. Automated assignment of backbone NMR peaks using constrained bipartite matching. *Computing in Science and Engineering*, 4(1):50–62, 2002. 22, 37

D.E. Zimmerman, C.A. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. Shimotakahara, C. Chien, R. Powers, and G.T. Montelione. Automated analysis of protein NMR assignments using methods from artificial intelligence. *Journal of Molecular Biology*, 269: 592–610, 1997. 22, 23