Determining Protein Structures from NOESY Distance Constraints by Semidefinite Programming

Babak Alipanahi[1*], Nathan Krislock[2], Ali Ghodsi[3], Henry Wolkowicz[4], Logan Donaldson[5], and Ming Li[1**]

[1] David R. Cheriton School of Computer Science, University of Waterloo,
Waterloo, Ontario, Canada

[2] INRIA Grenoble Rhône-Alpes, France

[3] Department of Statistics and Actuarial Science, University of Waterloo,
Waterloo, Ontario, Canada

[4] Department of Combinatorics and Optimization, University of Waterloo,
Waterloo, Ontario, Canada

[5] Department of Biology, York University,
Toronto, Ontario, Canada

Abstract. All practical contemporary protein NMR structure determination methods use molecular dynamics coupled with a simulated annealing schedule. The objective of these methods is to minimize the error of deviating from the NOE distance constraints. However, this objective function is highly nonconvex and, consequently, difficult to optimize. Euclidean distance geometry methods based on semidefinite programming (SDP) provide a natural formulation for this problem. However, complexity of SDP solvers and ambiguous distance constraints are major challenges to this approach. The contribution of this paper is to provide a new SDP formulation of this problem that overcomes these two issues for the first time. We model the protein as a set of intersecting two- and three-dimensional cliques, then we adapt and extend a technique called semidefinite facial reduction to reduce the SDP problem size to approximately one quarter of the size of the original problem. The reduced SDP problem can not only be solved approximately 100 times faster, but is also resistant to numerical problems from having erroneous and inexact distance bounds.

Key words: Molecular structural biology, nuclear magnetic resonance, semidefinite programming

[*] Current address: Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada

[**] All correspondence should be addressed to mli@uwaterloo.ca.

1 Introduction

Computing three-dimensional protein structures from their amino acid sequences has been one of the most widely studied problems in bioinformatics because knowing the structure of protein structure is key to understanding its physical, chemical, and biological properties. The protein nuclear magnetic resonance (NMR) method is fundamentally different from the X-ray method: It is not a "microscope with atomic resolution"; rather it provides a network of distance measurements between spatially proximate hydrogen atoms (Güntert, 1998). As a result, the NMR method relies heavily on complex computational algorithms. The existing methods for protein NMR can be categorized into four major groups: (*i*) methods based on Euclidean distance matrix completion (EDMC) (Braun et al., 1981; Havel and Wüthrich, 1984; Biswas et al., 2008; Leung and Toh, 2009), (*ii*) methods based on molecular dynamics and simulated annealing (Nilges et al., 1988; Brünger, 1993; Schwieters et al., 2003; Güntert et al., 1997; Güntert, 2004), (*iii*) methods based on local/global optimization (Braun and Go, 1985; Moré and Wu, 1997; Williams et al., 2001), and (*iv*) methods originating from sequence-based protein structure prediction algorithms (Shen et al., 2008; Raman et al., 2010; Alipanahi et al., 2011).

In the early years of protein NMR, many EDMC-based methods directly worked on the corresponding Euclidean distance matrix (EDM). The first method to use EDMC for protein NMR was developed by Braun et al. (Braun et al., 1981). Other notable methods include EMBED (Havel et al., 1983) and DISGEO (Havel and Wüthrich, 1984). These methods face two major drawbacks: Randomly guessing the unknown distances is ineffective and after several iterations of distance correction, distances tend to become large (Güntert, 1998). In addition, there is no way to control the embedding dimensionality.

A major breakthrough came by combining simulated annealing with molecular dynamics (MD) simulation. Nilges et al. made some improvements in the MD-based protein NMR structure determination (Nilges et al., 1988): Instead of an empirical energy function, they proposed a simple *geometrical* energy function based on the NOE restraints that penalized large violations and they also combined simulated annealing (SA) with MD. These methods were able to search the massive conformation space without being trapped in one of 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). The number of degrees of freedom in torsion angle space is nearly 10 times smaller than in Cartesian coordinates space, while being equivalent under mild assumptions. 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.

1.1 Gram Matrix Methods

Using the Gram matrix, or the matrix of inner products, has many advantages: (*i*) The Gram matrix and Euclidean distance matrix (EDM) are linearly related to each other. (*ii*) Instead of enforcing all of the triangle inequality constraints, it is sufficient to enforce that the Gram matrix is positive semidefinite. (*iii*) The embedding dimension and the rank of the Gram matrix are directly related.

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 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 the complexity of SDP solvers. To overcome this limitation Biswas et al. proposed DAFGAL, which is built on the idea of *divide-and-stitch* (Biswas et al., 2008). Leung and Toh proposed the DISCO method (Leung and Toh, 2009). It is an extension of

DAFGAL that can determine protein molecules with more than 10,000 atoms using a *divide-and-conquer* technique. The improved methods for partitioning the partial distance matrix and iteratively aligning the solutions of the subproblems, boost the performance of DISCO in comparison to DAFGAL.

1.2 Contributions of the Proposed SPROS Method

Most of the existing methods make some of the following assumptions: (*i*) assuming to know the (nearly) exact distances between atoms, (*ii*) assuming to have the distances between any type of nuclei (not just hydrogens), (*iii*) ignoring the fact that not all hydrogens can be uniquely assigned, and (*iv*) overlooking the ambiguity in the NOE cross-peak assignments. In order to automate the NMR protein structure determination process, we need a robust structure calculation method that tolerates more errors. We give a new SDP formulation that does not assume (*i*–*iv*) above. Moreover, the new method, called “SPROS” (Semidefinite Programming-based Protein structure determination), models the protein molecule as a set of intersecting two- and three-dimension cliques. We adapt and extend a technique called semidefinite facial reduction which makes the SDP problem strictly feasible and reduces its size to approximately one quarter the size of the original problem. The reduced problem is more numerically stable to solve and can be solved nearly 100 times faster.

2 The SPROS Method

We have divided the presentation of the SPROS method into providing the necessary background, followed by giving a description of techniques used for problem size reduction, and finally, showing the performance of the method on experimentally derived data.

2.1 Euclidean Distance Geometry

Scalars, vectors, sets, and matrices are shown in lower case, lower case bold italic, script, and upper case italic letters, respectively. 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: (i) for the space of real p-dimensional vectors, $\mathbb{R}^p$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{p} \boldsymbol{x}_i \boldsymbol{y}_i$, and ($ii$) for the space of real $p \times q$ matrices, $\mathbb{R}^{p\times q}$, $\langle A, B \rangle := \textbf{trace}(A^\top B) = \sum_{i=1}^{p} \sum_{j=1}^{q} A_{ij} B_{ij}$. The Euclidean distance norm of $\boldsymbol{x} \in \mathbb{R}^p$ is defined as $\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. We use the Matlab notation that $1{:}n := \{1, 2, \ldots, n\}$. For a matrix $A \in \mathbb{R}^{n\times n}$ and an *index set* $\mathcal{I} \subseteq 1{:}n$, $B = A[\mathcal{I}]$ is the $|\mathcal{I}| \times |\mathcal{I}|$ matrix formed by rows and columns of A indexed by $\mathcal{I}$. Finally, we let $\mathcal{S}^p$ the space of symmetric $p \times p$ matrices.

Euclidean Distance Matrix A symmetric matrix D 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:

$$D_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, \quad \forall i, j. \tag{1}$$

The smallest value of r is called the *embedding dimension* of D, and is denoted $\textbf{embdim}(D)$. The space of all $n \times n$ EDMs is denoted $\mathcal{E}^n$.

The 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 *Gram Matrix*, is given by $G := X^\top X$. It immediately follows that $G \in \mathcal{S}^n_+$, where $\mathcal{S}^n_+$ is the set of symmetric positive semidefinite $n \times n$ matrices. The Gram matrix and the Euclidean distance matrix are linearly related:

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

where $\mathbf{1}$ is the all-ones vector of the appropriate size. To go from the EDM to the Gram matrix, we use the $\mathsf{K}^\dagger \colon \mathcal{S}^n \to \mathcal{S}^n$ linear map:

$$G = \mathsf{K}^\dagger(D) := -\tfrac{1}{2} HDH, \quad D \in \mathcal{S}^n_H, \tag{3}$$

where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ is the *centering* matrix, $\mathcal{S}^n$ is the space of symmetric $n \times n$ matrices, and $\mathcal{S}^n_H := \{A \in \mathcal{S}^n : \mathbf{diag}(A) = \mathbf{0}\}$, is the set of symmetric matrices with zero diagonal.

Schoenberg's Theorem Given a matrix D, we can determine if it is an EDM with the following well-known theorem (Schoenberg, 1935):

Theorem 1. *A matrix $D \in \mathcal{S}^n_H$ is a Euclidean distance matrix if and only if $\mathsf{K}^\dagger(D)$ is positive semidefinite. Moreover,* $\mathbf{embdim}(D) = \mathbf{rank}(\mathsf{K}^\dagger(D))$ *for all $D \in \mathcal{E}^n$.*

2.2 The SDP Formulation

Semidefinite optimization or, more commonly, semidefinite programming is a class of convex optimization problems that has 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 SDP problems (Vandenberghe and Boyd, 1996). Current state-of-the-art SDP solvers are based on *primal-dual interior-point* methods.

Preliminary Problem Formulation There are three types of constraints in our formulation: (i) equality constraints, which are the union of equality constraints preserving bond lengths $(\mathcal{B})$, bond angles $(\mathcal{A})$, and planarity of the coplanar atoms $(\mathcal{P})$, giving $\mathcal{E} = \mathcal{E}_\mathcal{B} \cup \mathcal{E}_\mathcal{A} \cup \mathcal{E}_\mathcal{P}$; (ii) upper bounds, which are the union of NOE-derived $(\mathcal{N})$, hydrogen bonds ($\mathcal{H}$), disulfide and salt bridges ($\mathcal{D}$), and torsion angle ($\mathcal{T}$) upper bounds, giving $\mathcal{U} = \mathcal{U}_{\mathcal{N}} \cup \mathcal{U}_{\mathcal{H}} \cup \mathcal{U}_{\mathcal{D}} \cup \mathcal{U}_{\mathcal{T}}$; *(iii)* lower bounds, which are the union of steric or van der Waals ($\mathcal{W}$) and torsion angle ($\mathcal{T}$) lower bounds, giving $\mathcal{L} = \mathcal{L}_{\mathcal{W}} \cup \mathcal{L}_{\mathcal{T}}$. We assume the target protein has n atoms, $a_1, \ldots, a_n$. The preliminary problem formulation is given by:

$$\begin{array}{ll} \text{minimize} & \gamma\langle I, K\rangle + \sum_{ij} w_{ij}\xi_{ij} + \sum_{ij} w'_{ij}\zeta_{ij} \\ \text{subject to} & \langle A_{ij}, K\rangle = e_{ij},\ (i,j) \in \mathcal{E} \\ & \langle A_{ij}, K\rangle \le u_{ij} + \xi_{ij},\ (i,j) \in \mathcal{U} \\ & \langle A_{ij}, K\rangle \ge l_{ij} - \zeta_{ij},\ (i,j) \in \mathcal{L} \\ & \xi_{ij} \in \mathbb{R}_+,\ (i,j) \in \mathcal{U}, \quad \zeta_{ij} \in \mathbb{R}_+,\ (i,j) \in \mathcal{L} \\ & K\mathbf{1} = \mathbf{0}, \quad K \in \mathcal{S}^n_+, \end{array} \tag{4}$$

where $A_{ij} = (\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top$ and $\boldsymbol{e}_i$ is the ith column of the identity matrix. The *centering* constraint $K\mathbf{1} = \mathbf{0}$, ensures that the embedding of K is centered at the origin. Since both upper bounds and lower bounds may be inaccurate and noisy, non-negative penalized slacks, ζ_{ij}'s and ξ_{ij}'s, are included to prevent infeasibility and manage ambiguous upper bounds. The heuristic rank reduction term, $\gamma\langle I, K\rangle$, with $\gamma < 0$, in the objective function, produces lower-rank solutions (Weinberger and Saul, 2004).

Bond lengths and angles Covalent bonds are very stable, and since their fluctuations cannot be detected in NMR experiments, all bond lengths and angles must be set to ideal values computed from accurate X-ray structures; see (Engh and Huber, 1991). 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.

Planarity constraints Proteins contain several coplanar atoms, from HCON in the peptide planes, and from side chain in moieties found in nine amino acids (Hooft et al., 1996). We have enforced planarity by preserving the distances between all coplanar atoms.

Torsion angle constraints Another source of structural information in protein NMR is the set of torsion angle restraints, defined as $\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}$, $i \in \mathcal{T}$. We extend the idea proposed in (Sussman, 1985) 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. Thus, for example, we can constrain the Φ_i angle by constraining the distance between the C_i and C_{i-1} atoms.

Penalizing incorrect bounds Let $\boldsymbol{\xi} \in \mathbb{R}^{|\mathcal{U}|}$ be a vector containing all of the slacks for upper bounds. Since $\xi_{ij} \in \mathbb{R}_+$, assuming that all the weights are the same, i.e., $w_{ij} = w$, we have $w \sum_{ij} \xi_{ij} = w \|\boldsymbol{\xi}\|_1$, where $\|\boldsymbol{x}\|_1$ is the ℓ-1 norm of vector $\boldsymbol{x}$. The fact that minimizing the ℓ_1-norm finds *sparse* solutions is a widely known and used heuristic (Boyd and Vandenberghe, 2004). In our problem, $\xi_{ij} = 0$ implies no violation; consequently, SPROS tends to find a solution that violates a minimum number of upper bounds.

Pseudo-atoms Not all hydrogens can be uniquely assigned, such as the hydrogens in the methyl groups; therefore, upper bounds involving these hydrogens are ambiguous. To overcome this problem, *pseudo-atoms* are introduced (Güntert, 1998). Given an ambiguous constraint between one of the hydrogens and atom A, by using the triangle inequality, we modify the constraint as follows:

$$\|\mathrm{HB}_i - \mathrm{A}\| \leq b,\ i \in \{1,2,3\} \quad \Rightarrow \quad \|\mathrm{QB} - \mathrm{A}\| \leq b + \|\mathrm{HB}_i - \mathrm{QB}\|, \tag{5}$$

where $\|\mathrm{HB}_i - \mathrm{QB}\|$ is the same for $i = 1, 2, 3$. 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. We adapt the pseudo-atoms used in CYANA (Güntert, 2004).

Side chain simplification In CYANA, hydrogens that do not participate directly in the structural solution are discarded initially and then added at later stages (Güntert, 2004). We have adapted this approach by discarding hydrogens only if they make our problem smaller. 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. After the SDP problem is solved, the omitted hydrogen atoms are replaced and remain in all post-processing stages.

Challenges in Solving the SDP Problem Solving the optimization problem in (4) can be challenging: For small to medium sized proteins, the number of atoms, n, is 1,000-3,500, and current primal-dual interior-point SDP solvers cannot solve problems with $n > 2,000$ efficiently. Moreover, the optimization problem in (4) does not satisfy *strict feasibility*, causing numerical problems; see (Wei and Wolkowicz, 2010).

It can be observed that the protein contains many small intersecting cliques. For example, peptide planes or aromatic rings, are 2D cliques, and tetrahedral carbons form 3D cliques. As we show later, whenever there is a clique in the protein, the corresponding Gram matrix, K, can never be full-rank, which violates strict feasibility. By adapting and extending a technique called *semidefinite facial reduction*, not only do we obtain an equivalent problem that satisfies strict feasibility, but we also significantly reduce the SDP problem size.

2.3 Cliques in a Protein Molecule

A protein molecule with ℓ amino acid residues has $\ell + 1$ planes in its backbone. Moreover, each amino acid has a different side chain with a different structure; therefore, the number of cliques in each side chain varies (see Table 4 in Appendix B for the number of cliques in each amino acid 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 around 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 two peptide planes $\mathcal{P}_{i-1}$ and $\mathcal{P}_i$ in two atoms: $\mathcal{S}_i^{(1)} \cap \mathcal{P}_{i-1} = \{\mathrm{N}_i, \mathrm{CA}_i\}$ and $\mathcal{S}_i^{(1)} \cap \mathcal{P}_i = \{\mathrm{CA}_i, \mathrm{C}_i\}$. Side chain cliques for all twenty amino acids are listed in Table 4 (see Appendix B). 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 $\mathcal{C}_i = \mathcal{P}_{i-1}$, $1 \leq i \leq \ell + 1$, and $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(1)}$, $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(2)}$, $\ldots$, $\mathcal{C}_q = \mathcal{S}_\ell^{(s_\ell)}$. For properties of the cliques in the protein molecule, see Appendix A.

2.4 Algorithm for Finding the Face of the Structure

For $t < n$ and $U \in \mathbb{R}^{n \times t}$, the set of matrices $U\mathcal{S}_+^t U^\top$ is a face of $\mathcal{S}_+^n$ (in fact every face of $\mathcal{S}_+^n$ can be described in this way); see, e.g., (Ramana et al., 1997). We let $\mathbf{face}(\mathcal{F})$ represent the smallest face containing a subset $\mathcal{F}$ of $\mathcal{S}_+^n$; then we have the important property that $\mathbf{face}(\mathcal{F}) = U\mathcal{S}_+^t U^\top$ if and only if there exists $Z \in \mathcal{S}_{++}^t$ such that $UZU^\top \in \mathcal{F}$. Furthermore, in this case, we have that every $Y \in \mathcal{F}$ can be decomposed as $Y = UZU^\top$, for some $Z \in \mathcal{S}_+^t$, and the reduced feasible set $\{Z \in \mathcal{S}_+^t : UZU^\top \in \mathcal{F}\}$ has a strictly feasible point, giving us a problem that is more numerically stable to solve (problems that are not strictly feasible have a dual optimal set that is unbounded and therefore can be difficult to solve numerically; for more information, see (Wei and Wolkowicz, 2010)). Moreover, if $t \ll n$, this results in a significant reduction in the matrix size.

The Face of a Single Clique Here, we solve the Single Clique problem, which is defined as follows: Let D be a partial EDM of a protein. Suppose the first n_1 points form a clique in the protein, such that for $\mathcal{C}_1 = \{1, \ldots, n_1\}$, all distances are known. That is, the matrix $D_1 = D[\mathcal{C}_1]$ is completely specified. Moreover, let $r_1 = \mathbf{embdim}(D_1)$. We now show how to compute the smallest face containing the feasible set $\{K \in \mathcal{S}_+^n : \mathsf{K}(K[\mathcal{C}_1]) = D_1\}$.

Theorem 2 (Single Clique, **(Krislock and Wolkowicz, 2010)).** *Let the matrix* $U_1 \in \mathbb{R}^{n \times (n - n_1 + r_1 + 1)}$ *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* $\bar{U}_1 := \begin{bmatrix} V_1 \, \mathbf{1} \end{bmatrix}$ *and* $U_1 := \begin{array}{c@{}c} & \begin{array}{cc} \scriptstyle r_1+1 & \scriptstyle n-n_1 \end{array} \\ \begin{array}{c} \scriptstyle n_1 \\ \scriptstyle n-n_1 \end{array} & \begin{bmatrix} \bar{U}_1 & 0 \\ 0 & I \end{bmatrix} \end{array} \in \mathbb{R}^{n \times (n - n_1 + r_1 + 1)}$.

Then U_1 *has full column rank,* $\mathbf{1} \in \mathbf{range}(U)$, *and*

$$\mathbf{face}\{K \in \mathcal{S}^n_+ : \mathsf{K}(K[\mathcal{C}_1]) = D[\mathcal{C}_1]\} = U_1 \mathcal{S}_+^{n-n_1+r_1+1} U_1^\top.$$

Computing the V_1 Matrix In Theorem 2, we can find V_1 by computing the eigendecomposition of $\mathsf{K}^\dagger(D[\mathcal{C}_1])$ as follows:

$$\mathsf{K}^\dagger(D[\mathcal{C}_1]) = V_1 \Lambda_1 V_1^\top, \quad V_1 \in \mathbb{R}^{n_1 \times r_1},\ \Lambda_1 \in \mathcal{S}^{r_1}_{++}. \tag{6}$$

It can be seen that V_1 has full column rank (columns are orthonormal) and also that $\mathbf{range}(V_1) = \mathbf{range}(\mathsf{K}^\dagger(D_1))$.

2.5 The Face of a Protein Molecule

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 $r_l = \mathbf{embdim}(D[\mathcal{C}_l])$, and $n_l = |\mathcal{C}_l|$. Let $\mathcal{F}$ be the feasible set of the SDP 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

$$\mathcal{F} \subseteq \left(\bigcap_{l=1}^{q} \mathcal{F}_l \right) \cap \mathcal{S}^n_C, \tag{7}$$

where $\mathcal{S}^n_C := \{K \in \mathcal{S}^n : K\mathbf{1} = \mathbf{0}\}$ are the *centered* symmetric matrices. For $l = 1, \ldots, q$, let $F_l := \mathbf{face}(\mathcal{F}_l) = U_l \mathcal{S}_+^{n-n_l+r_l+1} U_l^\top$, where U_l is computed as in Theorem 2. We have (Krislock and Wolkowicz, 2010):

$$\left(\bigcap_{l=1}^{q} \mathcal{F}_l \right) \cap \mathcal{S}^n_C \subseteq \left(\bigcap_{l=1}^{q} U_l \mathcal{S}_+^{n-n_l+r_l+1} U_l^\top \right) \cap \mathcal{S}^n_C = (U \mathcal{S}^k_+ U^\top) \cap \mathcal{S}^n_C, \tag{8}$$

where $U \in \mathbb{R}^{n \times k}$ is a full column rank matrix that satisfies $\mathbf{range}(U) = \bigcap_{l=1}^{q} \mathbf{range}(U_l)$.

We now have an efficient method for computing the face of the feasible set $\mathcal{F}$. To have better numerical accuracy, we developed a bottom-up algorithm for intersecting subspaces (see Algorithm 1 in Appendix C).

After computing U, we can decompose the Gram matrix as $K = UZU^\top$, for $Z \in \mathcal{S}_+^k$. However, by exploiting the centering constraint, $K\mathbf{1} = \mathbf{0}$, we can reduce the matrix size one more. If $V \in \mathbb{R}^{k\times(k-1)}$ has full column rank and satisfies $\mathbf{range}(V) = \mathbf{null}(\mathbf{1}^\top U)$, then we have (Krislock and Wolkowicz, 2010):

$$\mathcal{F} \subseteq (UV)\mathcal{S}_+^{k-1}(UV)^\top. \tag{9}$$

For more details on facial reduction for Euclidean distance matrix completion problems, see (Krislock, 2010).

Constraints for Preserving the Structure of Cliques If we find a *base* set of points $\mathcal{B}_l$ in each clique $\mathcal{C}_l$ such that $\mathbf{embdim}(D[\mathcal{B}_l]) = r_l$, then by fixing the distances between points in the base set and 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, we need to fix *only* the distances between base points (Alipanahi et al., 2012), resulting in a three- to four-fold reduction in the number of equality constraints. We call the reduced set of equality constraints $\mathcal{E}_{\mathrm{FR}}$.

2.6 Solving and Refining the Reduced SDP Problem

The SPROS method flowchart is depicted in Appendix D (see Fig. 2). In it, we describe the blocks for solving the SDP problem and for refining the solution. From equation (9), we can formulate the reduced SDP problem as follows:

$$\begin{array}{ll} \text{minimize} & \gamma\langle I, Z\rangle + \sum_{ij} w_{ij}\xi_{ij} + \sum_{ij} w'_{ij}\zeta_{ij} \\ \text{subject to} & \langle A'_{ij}, Z\rangle = e_{ij},\ (i,j) \in \mathcal{E}_{\mathrm{FR}} \\ & \langle A'_{ij}, Z\rangle \le u_{ij} + \xi_{ij},\ (i,j) \in \mathcal{U} \\ & \langle A'_{ij}, Z\rangle \ge l_{ij} - \zeta_{ij},\ (i,j) \in \mathcal{L} \\ & \xi_{ij} \in \mathbb{R}_+,\ (i,j) \in \mathcal{U}, \quad \zeta_{ij} \in \mathbb{R}_+,\ (i,j) \in \mathcal{L} \\ & Z \in \mathcal{S}_+^{k-1}, \end{array} \tag{10}$$

where $A'_{ij} = (UV)^\top A_{ij}(UV)$.

Weights and the regularization parameter For each type of upper and lower bound, we define a fixed penalizing weight for violations. For example, for upper bounds (similarly for lower bounds) we have $\forall (i,j) \in \mathcal{U}_{\mathcal{X}}, w_{ij} = w_{\mathcal{X}}$. We set $w_{\mathcal{N}} = 1$ and $w_{\mathcal{H}} = w_{\mathcal{D}} = w_{\mathcal{T}} = 10$ because upper bounds from hydrogen bonds and disulfide/salt bridges are assumed to be more accurate than are NOE-derived upper bounds. Moreover, the range of torsion angle violations is ten times smaller than NOE violations.

Let $m_{\mathcal{U}} = |\mathcal{U}|$ and R be the radius of the protein. Then, the maximum upper bound violation is $2R$. Moreover, $\langle I, Z \rangle \leq nR$. Discarding the role of lower bound violations, with the goal of approximately balancing the two terms, a suitable γ is:

$$\gamma n R \approx 2\varepsilon w m_{\mathcal{U}} R \quad \Rightarrow \quad \gamma = \frac{2\varepsilon w m_{\mathcal{U}}}{n}, \tag{11}$$

where $0 \leq \varepsilon \leq 1$ is the fraction of violated upper bounds. In practice $\varepsilon \approx 0.01 - 0.30$, and $\bar{\gamma} \approx w m_{\mathcal{U}}/50n$ works well.

Post-Processing We perform a refinement on the raw structure determined by the SDP solver. For this refinement we use a BFGS-based quasi-Newton method (Lewis and Overton, 2009) that only requires the value of the objective function and its gradient at each point. Letting $X^{(0)} = X_{\text{SDP}}$, we iteratively minimize the following objective function:

$$\begin{aligned} \phi(X) = {} & w_{\mathcal{E}} \sum_{(i,j)\in\mathcal{E}} \left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - e_{ij}\right)^2 + w_{\mathcal{U}} \sum_{(i,j)\in\mathcal{U}} f\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - u_{ij}\right)^2 \\ & + w_{\mathcal{L}} \sum_{(i,j)\in\mathcal{L}} g\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - l_{ij}\right)^2 + w_{\mathcal{R}} \sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2, \end{aligned} \tag{12}$$

where $f(\alpha) = \max(0, \alpha)$ and $g(\alpha) = \min(0, -\alpha)$. We set $w_{\mathcal{E}} = 2$, $w_{\mathcal{U}} = 1$, and $w_{\mathcal{L}} = 1$. In addition, to balance the regularization term, we set $w_{\mathcal{R}} = \alpha \phi(X^{(0)})|_{w_R=0}/25 \sum_{i=1}^{n} \|\boldsymbol{x}_i^{(0)}\|^2$, 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, 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.

Fixing incorrect chiralities 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 resolves any problems.

Improving the stereochemical quality Williamson and Craven have described the effectiveness of explicit solvent refinement of NMR structures and suggest that it should be a standard procedure (Williamson and Craven, 2009). For protein structures that have regions of high mobility/uncertainty due to few or no NOE observations, we have successfully employed a hybrid protocol from XPLOR-NIH that incorporates thin-layer water refinement (Linge et al., 2003) and a multidimensional torsion angle database (Kuszewski et al., 1996, 1997).

3 Results

We tested the performance of SPROS on 18 proteins: 15 protein data sets from the DOCR database in the NMR Restraints Grid (Doreleijers et al., 2003, 2005) and three protein data sets from Donaldson's laboratory at York University. We chose proteins with different sizes and topologies, as listed in Table 1. Finally, the input to the SPROS method is exactly the same as the input to the widely-used CYANA method.

3.1 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). For minimizing the post-processing objective function (12), we used the BFGS-based quasi-Newton method implementation by Lewis and Overton (Lewis and Overton, 2009). All the experiments were carried out on an Ubuntu 11.04 Linux PC with a 2.8 GHz Intel Core i7 Quad-Core processor and 8 GB of memory.

3.2 Determined Structures

From the 18 test proteins, 9 of them were calculated with backbone RMSDs less than or equal to 1.0 Å, and 16 have backbone RMSDs less than 1.5 Å. Detailed analysis of calculated structures is listed in Table 2. The superimposition of the SPROS and reference structures for three of the proteins are depicted in Figure 1. More detailed information about the determined structures can be found in (Alipanahi, 2011).

To further assess the performance of SPROS, we compared the SPROS and reference structures for 1G6J, Ubiquitin, and 2GJY, PTB domain of Tensin, with their corresponding X-ray structures, 1UBQ and 1WVH, respectively. For 1G6J, the backbone (heavy atoms) RMSDs for SPROS and the reference structures are 0.42 Å (0.57 Å) and 0.73±0.04 Å (0.98±0.04 Å), respectively. For 2GJY, the backbone (heavy atoms) RMSDs for SPROS and the reference structures are 0.88 Å (1.15 Å) and 0.89 ± 0.08 Å (1.21 ± 0.06 Å), respectively.

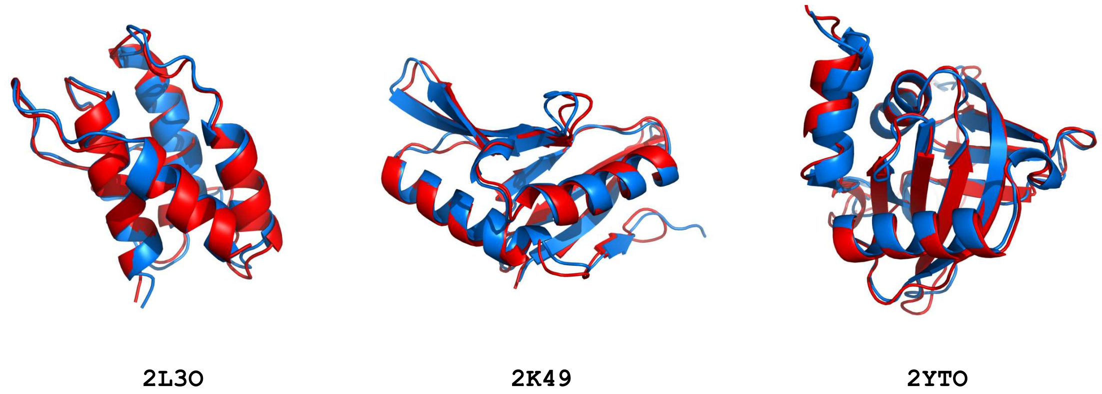

Fig. 1. Superimposition of structures determined by SPROS in blue and the reference structures in red.

3.3 Discussion

The SPROS method was tested on 18 experimentally derived protein NMR data sets of sequence lengths ranging from 76 to 307 (weights ranging from 8 to 35 KDa). Calculation times were in the order of a few minutes per structure. Accurate results were obtained for all of the data sets, although with some variability in precision. The best attribute of the SPROS method is its tolerance for, and efficiency at, managing many incorrect distance constraints (that are typically defined as upper bounds).

The reduction methodology developed for SPROS is an ideal choice for protein-ligand docking. If the side chains participating at the interaction surface are only declared to be flexible, it has the effect of reducing the SDP matrix size to less than 100. Calculations under these specific parameters can be achieved in a few seconds thereby making SPROS a worthwhile choice for automated, high-throughput screening.

Our final goal is a fully automated system for NMR protein structure determination, from peak picking (Alipanahi et al., 2009) to resonance assignment (Alipanahi et al., 2011), to protein structure determination. An automated system, without the laborious human intervention will have to tolerate more errors than usual. This was the initial motivation of designing SPROS. The key is to tolerate more errors. Thus, we are working towards incorporating an adaptive violation weight mechanism to identify the most significant outliers in the set of distance restraints automatically.

Acknowledgments

This work was supported in part by NSERC Grant OGP0046506, Canada Research Chair program, an NSERC Collaborative Grant, David R. Cheriton Graduate Scholarship, OCRiT, Premier's Discovery Award, and Killam Prize.

Author Disclosure Statement

No competing financial interests exist.

Bibliography

Alipanahi, B. 2011. *New Approaches to Protein NMR Automation* [Ph.D. dissertation]. University of Waterloo, Waterloo, ON.

Alipanahi, B., Gao, X., Karakoc, E., et al. 2009. PICKY: a novel SVD-based NMR spectra peak picking method. *Bioinformatics*, 25(12):i268–275.

Alipanahi, B., Gao, X., Karakoc, E., et al. 2011. Error tolerant NMR backbone resonance assignment and automated structure generation. *J. Bioinform. Comput. Bioly.*, 0(1):1–26.

Alipanahi, B., Krislock, N., and Ghodsi, A. 2012. Large-scale Manifold learning by semidefinite facial reduction. Unpublished manuscript (in preparation).

Biswas, P., Liang, T.C., Toh, K.C., et al. 2006. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Trans. Autom. Sci. Eng.*, 3:360–371.

Biswas, P., Toh, K.C., and Ye, Y. 2008. A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM. J. Sci. Comp.*, 30:1251–1277.

Biswas., P., and Ye, Y. 2004. 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.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization.* Cambridge University Press.

Braun, W., Bösch, C., Brown, L.R., et al. 1981. Combined use of proton-proton overhauser enhancements and a distance geometry algorithm for determination of polypeptide conformations. application to micelle-bound glucagon. *Biochim. Biophys. Acta*, 667(2):377–396.

Braun, W., and Go, N. 1985 Calculation of protein conformations by proton-proton distance constraints. a new efficient algorithm. *J. Mol. Biol.*, 186(3):611–626.

Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., et al. 1983. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–

217.

Brünger, A.T. 1993. *X-PLOR Version 3.1: A System for X-ray Crystallography and NMR.* Yale University Press.

Chen, V.B., Arendall, W.B., Headd, J.J., et al. 2010. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D*, 66(Pt 1):12–21.

Doherty, L., Pister, K.S.J., and El Ghaoui, L. 2001. 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.

Doreleijers, J.F., Mading, S., Maziuk, D., et al. 2003. BioMagResBank database with sets of experimental NMR constraints corresponding to the structures of over 1400 biomolecules deposited in the protein data bank. *J. Biomol. NMR*, 26(2):139–146.

Doreleijers, J.F., Nederveen, A.J., Vranken, W., et al. 2005. BioMagResBank databases DOCR and FRED containing converted and filtered sets of experimental NMR restraints and coordinates from over 500 protein PDB structures. *J. Biomol. NMR*, 32(1):1–12.

Engh, R.A., and Huber, R. 1991. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta. Crystallogr. A*, 47(4):392–400.

Güntert, P. 1998. Structure calculation of biological macromolecules from NMR data. *Q. Rev. Biophys.*, 31(2):145–237.

Güntert, P. 2004. Automated NMR structure calculation with CYANA. *Methods in Molecular Biology*, 278:353–378.

Güntert, P., Mumenthaler, C., and Wüthrich, K. 1997. Torsion angle dynamics for NMR structure calculation with the new program DYANA. *J. Mol. Biol.*, 273:283–298.

Havel, T.F., Kuntz, I.D., and Crippen, G.M. 1983. The theory and practice of distance geometry. *Bull. Math. Biol.*, 45(5):665–720.

Havel, T.F., and Wüthrich, K. 1984. 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. *B. Math. Biol.*, 46(4):673–698.

Hooft, R.W.W., Sander, C., and Vriend, G. 1996. Verification of Protein Structures: Side-Chain Planarity. *J. Appl. Crystallogr.*, 29(6):714–716.

Kim, S., Kojima, M., and Waki, H. 2009. Exploiting sparsity in SDP relaxation for sensor network localization. *SIAM J. Optimiz.*, 20(1):192–215.

Krislock, N. 2010. *Semidefinite Facial Reduction for Low-Rank Euclidean Distance Matrix Completion* [Ph.D. dissertation]. University of Waterloo, Waterloo, ON.

Krislock, N., and Wolkowicz, H. 2010. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM J. Optimiz.*, 20:2679–2708.

Kuszewski, J., Gronenborn, A.M., and Clore, G.M. 1996. 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.

Kuszewski, J., Gronenborn, A.M., and Clore, G.M. 1997. Improvements and extensions in the conformational database potential for the refinement of NMR and x-ray structures of proteins and nucleic acids. *J. Magn. Reson.*, 125(1):171–177.

Leung, N.H.Z., and Toh, K.C. 2009. An SDP-based divide-and-conquer algorithm for large-scale noisy anchor-free graph realization. *SIAM J. Sci. Comput.*, 31:4351–4372.

Lewis, A.S., and Overton, M.L. 2009. Nonsmooth optimization via BFGS. *Submitted to SIAM J. Optimiz.*

Linge, J.P., Habeck, M., Rieping, W., et al. 2003. ARIA: automated NOE assignment and NMR structure calculation. *Bioinformatics*, 19(2):315–316.

Moré, J.J., and Wu, Z. 1997 Global continuation for distance geometry problems. *SIAM J. Optimiz.*, 7:814–836.

Nilges, M., Clore, G.M., and Gronenborn, A.M. 1998. Determination of three-dimensional structures of proteins from interproton distance data by hybrid distance geometry-dynamical simulated annealing calculations. *FEBS lett.*, 229(2):317–324.

Raman, S., Lange, O.F., Rossi, P., et al. 2010. NMR structure determination for larger proteins using Backbone-Only data. *Science*, 327(5968):1014–1018.

Ramana, M.V., and Tunçel, L., Wolkowicz, H. 1997. Strong duality for semidefinite programming. *SIAM J. Optimiz.*, 7(3):641–662.

Schoenberg, I.J. 1935. 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.

Schwieters, C.D., Kuszewski, J.J., and Clore, G.M. 2006. Using Xplor-NIH for NMR molecular structure determination. *Prog. Nucl. Mag. Res. Sp.*, 48:47–62.

Schwieters, C.D., Kuszewski, J.J., Tjandra, N., et. al. 2003. The Xplor-NIH NMR molecular structure determination package. *J. Magn. Reson.*, 160:65–73.

Shen, Y., Lange, O., Delaglio, F., et al. 2008. Consistent blind protein structure generation from NMR chemical shift data. *P. Natl. Acad. Sci. USA*, 105(12):4685–4690.

Sussman, J.L. 1985. *Constrained-restrained least-squares (CORELS) refinement of proteins and nucleic acids*, volume 115, pages 271–303. Elsevier.

Tütüncü, R.H., Toh, K.C., and Todd, M.J. 2003. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95(2, Ser. B):189–217.

Vandenberghe, L., and Boyd, S. 1996. Semidefinite programming. *SIAM Rev.*, 38(1):49–95.

Wang, Z., Zheng, S., Ye, Y., et al. 2008. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM J. Optimiz.*, 19(2):655–673.

Wei, H., and Wolkowicz, H. 2010. Generating and measuring instances of hard semidefinite programs. *Math. Program.*, 125:31–45.

Weinberger, K.Q., and Saul, L.K. 2004. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, 2:988–995.

Williamson, M.P., and Craven, C.J. 2009. Automated protein structure calculation from NMR data. *J. Biomol. NMR*, 43(3):131–43.

Williams, G.A., Dugan, J.M., and Altman, R.B. 2001. Constrained global optimization for estimating molecular structure from atomic distances. *J. Comput. Biol.*, 8(5):523–547.

Table 1. Information about the proteins used in testing SPROS. The second, third, and fourth columns, list the topologies, sequence lengths, and molecular weight of the proteins, the fifth and sixth columns, n and n', list the original and reduced SDP matrix sizes, respectively. The seventh column lists the number of cliques in the protein. The eights and ninth columns, $m_{\mathcal{E}}$ and $m'_{\mathcal{E}}$, list the number of equality constraints in the original and reduced problems, respectively. The 10^{th} column, $m_{\mathcal{U}}$, lists the total number of upper bounds for each protein. The 11^{th} 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 12^{th} column, $\bar{m}_{\mathcal{U}} \pm s_{\mathcal{U}}$, lists the average number of upper bounds per residue, together with the standard deviation. The 13^{th} column, $m_{\mathcal{N}}$, lists the number of NOE-inferred upper bounds. The 14^{th} column, $p_{\mathcal{U}}$, lists the fraction of pseudo-atoms in the upper bounds in percentile. 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	topo.	len.	weight	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	a+b	76	8.58	1434	405	304 (201/103)	5543	1167	1354	21/ 29/ 17/ 33	31.9±15.3	1291	32	63	0
1B4R	B	80	7.96	1281	346	248 (145/103)	4887	1027	787	26/ 25/ 6 / 43	17.1±10.8	687	30	22	78
2E8O	A	103	11.40	1523	419	317 (212/105)	5846	1214	3157	19/ 29/ 26/ 26	71.4±35.4	3070	24	87	0
1CN7	a/b	104	11.30	1927	532	393 (253/140)	7399	1540	1560	46/ 24/ 12/ 18	23.1±13.4	1418	31	80	62
2KTS	a/b	117	12.85	2075	593	448 (299/149)	7968	1719	2279	22/ 28/ 14/ 36	34.6±17.4	2276	25	0	3
2K49	a+b	118	13.10	2017	574	433 (291/142)	7710	1657	2612	22/ 27/ 18/ 38	40.9±21.1	2374	27	146	92
2K62	B	125	15.10	2328	655	492 (327/165)	8943	1886	2367	21/ 32/ 15/ 32	33.9±18.6	2187	32	180	0
2L3O	A	127	14.30	1867	512	393 (269/124)	7143	1492	1270	24/ 38/ 20/ 18	22.5±12.7	1055	25	156	59
2GJY	a+b	144	15.67	2337	639	474 (302/172)	8919	1875	1710	7 / 30/ 19/ 44	25.0±16.6	1536	29	98	76
2KTE	a/b	152	17.21	2576	717	542 (360/182)	9861	2089	1899	17/ 31/ 22/ 30	24.3±20.8	1669	30	124	106
1XPW	B	153	17.44	2578	723	541 (355/186)	9837	2081	1206	0 / 31/ 11/ 58	17.0±10.8	934	37	210	62
2K7H	a/b	157	16.66	2710	756	563 (363/200)	10452	2196	2768	29/ 33/ 13/ 25	30.3±11.3	2481	19	239	48
2KVP	A	165	17.28	2533	722	535 (344/191)	9703	2094	5204	31/ 26/ 23/ 20	59.2±25.0	4972	22	232	0
2YTO	a+b	176	19.17	2940	828	627 (419/208)	11210	2404	3357	23/ 28/ 14/ 35	34.9±22.3	3237	30	120	0
2L7B	A	307	35.30	5603	1567	1205 (836/369)	21421	4521	4355	10/ 30/ 44/ 16	27.6±14.4	3459	23	408	488
1Z1V	A	80	9.31	1259	362	272 (181/91)	4836	1046	1261	46/ 24/ 18/ 13	28.6±16.3	1189	15	0	72
HACS1	B	87	9.63	1150	315	237 (156/81)	4401	923	828	46/ 21/ 5 / 27	20.2±14.2	828	20	0	36
2LJG	a+b	153	17.03	2343	662	495 (327/168)	9009	1909	1347	40/ 29/ 8 / 22	16.4±11.9	1065	28	204	78

Table 2. 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 Å computed by MolProbity, as defined by (Chen et al., 2010). The eighth 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
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
2YT0	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
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

Appendix A: Properties of Cliques

Let $\mathcal{C}_i = \mathcal{P}_{i-1}$, $1 \le i \le \ell + 1$, and $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(1)}$, $\mathcal{C}_{\ell+2} = \mathcal{S}_1^{(2)}$, $\ldots$, $\mathcal{C}_q = \mathcal{S}_\ell^{(s_\ell)}$. Let $r_i = \textbf{embdim}(D[\mathcal{C}_i])$. The following properties hold for cliques in a protein molecule:

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. $|\mathcal{C}_i| \geq r_i + 1$.
5. $3 \leq |\mathcal{C}_i| \leq 16$.
6. $\forall i, i'$, $|\mathcal{C}_i \cap \mathcal{C}_{i'}| \leq 2$.
7. $\nexists i$ such that $\forall i' \neq i$, $\mathcal{C}_i \cap \mathcal{C}_{i'} = \emptyset$.
8. If $\mathcal{I}_i = \{i' : \mathcal{C}_i \cap \mathcal{C}_{i'} \neq \emptyset\}$, then $\forall i, |\mathcal{I}_i| \leq 4$.
9. $\bigcup_{i=1}^{q} \mathcal{C}_i = 1{:}n$.

Appendix B: Additional Tables

Table 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

Table 4. 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	$\mathcal{S}^{(1)} = \{\text{N, CA, HA, CB, QB, C}\}$
Arg	5	$\mathcal{S}^{(4)} = \{\text{CG, CD, NE}\}*$ $\mathcal{S}^{(5)} = \{\text{CD, CE, HE, CZ, NH1, HH11, HH12}\}*$
Asn	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, OD1, ND2, HD21, HD22, QD2}\}*$
Asp	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, OD1, OD2}\}*$
Cys	2	$\mathcal{S}^{(2)} = \{\text{CA, CB, SG}\}*$
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	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, ND1, HD1, CD2, HD2, CE1, HE1, NE2}\}*$
Ile	3	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, CG1, CG2, QG2}\}$ $\mathcal{S}^{(3)} = \{\text{CB, CG1, CD1, QD1}\}*$
Leu	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, HG, CD1, CD2, QD1, QD2, QQD}\}$
Lys	5	$\mathcal{S}^{(1)} = \{\text{N, CA, HA, CB, C}\}$ $\mathcal{S}^{(2)} = \{\text{CA, CB, CG}\}*$ $\mathcal{S}^{(3)} = \{\text{CB, CG, CD}\}*$ $\mathcal{S}^{(4)} = \{\text{CG, CD, CE}\}*$ $\mathcal{S}^{(5)} = \{\text{CD, CE, NZ, QZ}\}*$
Met	4	$\mathcal{S}^{(3)} = \{\text{CB, CG, SD}\}*$ $\mathcal{S}^{(4)} = \{\text{CG, SD, CE QE}\}^{*}$
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	2	$\mathcal{S}^{(2)} = \{\text{CA, CB, OG}\}*$
Thr	2	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, OG1, CG2, QG2}\}$
Trp	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, CD1, HD1, CD2, CE2, CE3, HE3, NE1, HE1, CZ2, HZ2, CZ3, HZ3, CH2, HH2}\}*$
Tyr	3	$\mathcal{S}^{(3)} = \{\text{CB, CG, CD1, HD1, CE1, HE1, CE2, HE2, CD2, HD2, CZ, OH, QD, QE, QR}\}*$
Val	2	$\mathcal{S}^{(2)} = \{\text{CA, CB, HB, CG1, CG2, QG1, QG2, QQG}\}$

Appendix C: Efficient Subspace Intersection Algorithm

Algorithm 1: 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
    $Q_l^{(1)} = U_l$                                   // $Q_j^{(i)}$: $U$ of the subtree rooted at the node $j$, level $i$
    $\mathcal{A}_l^{(1)} = \mathcal{C}_l$               // $\mathcal{A}_j^{(i)}$: 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
            $\ell_2 \leftarrow \ell_1 + 1$
            $\mathcal{A}_j^{(i)} \leftarrow \mathcal{A}_j^{(i)} \cup \mathcal{A}_{\ell_2}^{(i-1)}$
            $Q_j^{(i)} \leftarrow \text{Intersect}(Q_j^{(i)}, Q_{\ell_2}^{(i-1)})$
        end
    end
    $p' \leftarrow p$
end
$U \leftarrow Q_1^{(v)}$                                // For the root $\mathcal{A}_1^{(v)} = 1{:}n$
```

Appendix D: SPROS Flow Chart

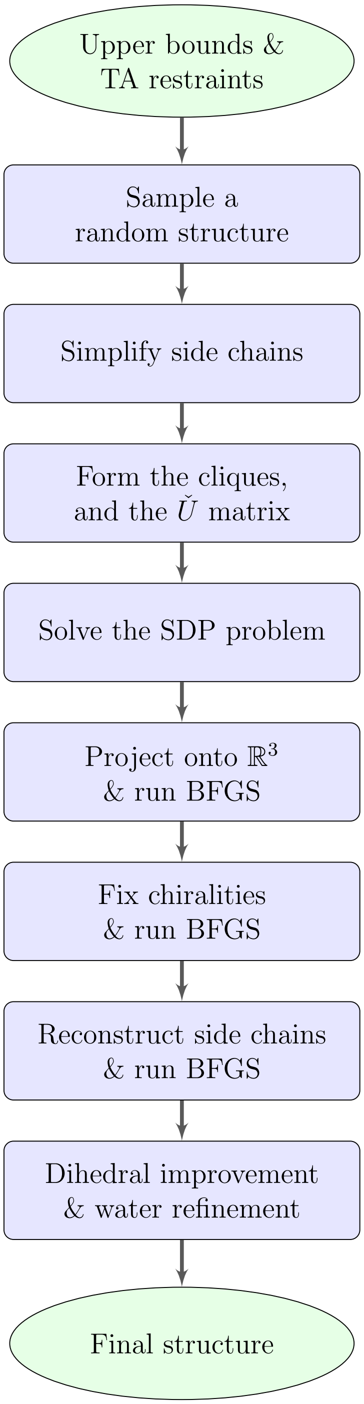

Fig. 2. SPROS method flowchart.