

Protein Comparison Methods for Structure Determination

Jeff Martin

December 8, 2008

Abstract

In this paper, we present a brief review of methods for protein structure comparison. We also present a novel idea for comparing homo-oligomeric protein complexes by using a simplified representation of the protein that takes advantage of the symmetry present in the protein. We compare this method to other similar protein comparison methods in computational experiments.

1 Introduction

Structure determination is the process by which one attempts to solve the structure (i.e. provide coordinates in \mathbb{R}^3 for every atom) of a protein that agrees with experimentally measured data. For instance, the Nuclear Overhauser Effect (NOE) from Nuclear Magnetic Resonance (NMR) can be exploited to derive distance restraints on pairs of protons (e.g. hydrogen atoms) in a protein by analyzing a NOESY spectrum [2]. These distance restraints are simply referred to as NOEs. In their unassigned form, they are merely scalar values giving a maximum separation distance in angstroms between two (unknown) protons. They must be assigned to pairs of protons in the protein before they can be used in determination algorithms. This is referred to as the assignment problem for NOEs and also a difficult problem [1]. Given only this information, the problem of structure determination then becomes one of graph embedding in \mathbb{R}^3 based on distance geometry which is proven strongly NP-Hard [4] and thus does not provide an efficient solution to the problem. Typical protocols for protein structure determination involve heavy reliance on simulation of molecular dynamics. The NOEs can be input to the simulation as restraints which in many cases can cause the simulation to converge to a realistic model of the protein.

However, in protein complexes demonstrating symmetry among its subunits (e.g. C_n or D_n), the simulation protocols can fail to converge. Since the convergence criteria of the simulation are difficult to determine, many tricks are employed to encourage convergence, but these rarely shed light on why the simulation failed to converge before. The main source of confusion for the simulations in this case is the nature of the experimentally-measured NOESY spectra. For symmetric proteins, a NOESY spectrum can still be used to derive distance restraints for two pairs of protons in the protein, but the assignment problem is much more difficult. It is not possible to determine (from just the NOESY spectrum) from which subunits the pairs of protons originate for a given observed NOE. For structure determination to be successful in this setting, one must explicitly model the symmetry of the protein complex.

1.1 Complete Systematic Structure Determination for Symmetrical Protein Complexes

Another approach to the structure determination problem attempts to find all possible structures for a protein that can be shown to satisfy the experimental restraints (e.g. NOEs). Such an approach is said to be complete if it can guarantee this property. In contrast to simulation-based approaches, systematic approaches to structure determination define a search configuration space (SCS) that represents all configurations of a protein (depending on the parameterization) and then searches that space to report the satisfying configurations. Since proteins are continuously flexible, the reported set of satisfying configurations is also continuous and represents a subset of the original SCS. To generate discrete configurations of the protein, the set of satisfying configurations can be uniformly sampled at a desired resolution. In practice, the resolution is chosen such that the minimum distance between any two conformations is 1 \AA RMSD.

One such approach that is both complete and systematic is an algorithm by Potluri et.al. [9] called SYMBRANE that parameterizes conformations of a C_n symmetrical protein complex in terms of the axes of symmetry. Let P be the perfect set of conformations of the protein complex that satisfy the experimental data. Given experimental NOEs and subunit conformation as input, SYMBRANE returns an approximation A that is guaranteed to be a superset of P . In addition, SYMBRANE allows the user to control the volume (in terms of the SCS) of the set of false positives $P \setminus A$. More computational time can be spent to bring A closer to P without discarding any satisfying configurations. In [9], Potluri et.al. derived a SCS that was four-dimensional and resulted in an ensemble of tens of thousands of structures sampled uniformly from A for the human phospholamban pentamer protein. Each of these structures was then compared to a previously-published reference structure for phospholamban to determine if SYMBRANE was able to recover it. Indeed, SYMBRANE was able to recover the published structure and also identify other structures with higher satisfaction scores, but doing so required comparing a large number of putative structures. Thus, a fast and efficient method for comparing protein structures was needed.

New work in progress hopes to extend SYMBRANE to solve structures that have D_n symmetry by using a parameterization derived in a similar way. Unsurprisingly, a more complicated form of symmetry leads to a more difficult problem. In this work, the SCS is six-dimensional and a search for structures satisfying experimental restraints for the DNA binding domain of the human P53 tumor suppressor protein (see Fig. 1) resulted in millions of putative structures sampled from the set of satisfying configurations. Thus, millions of protein structure comparisons must be performed to evaluate the success of the search and creates a need for even faster comparison methods.

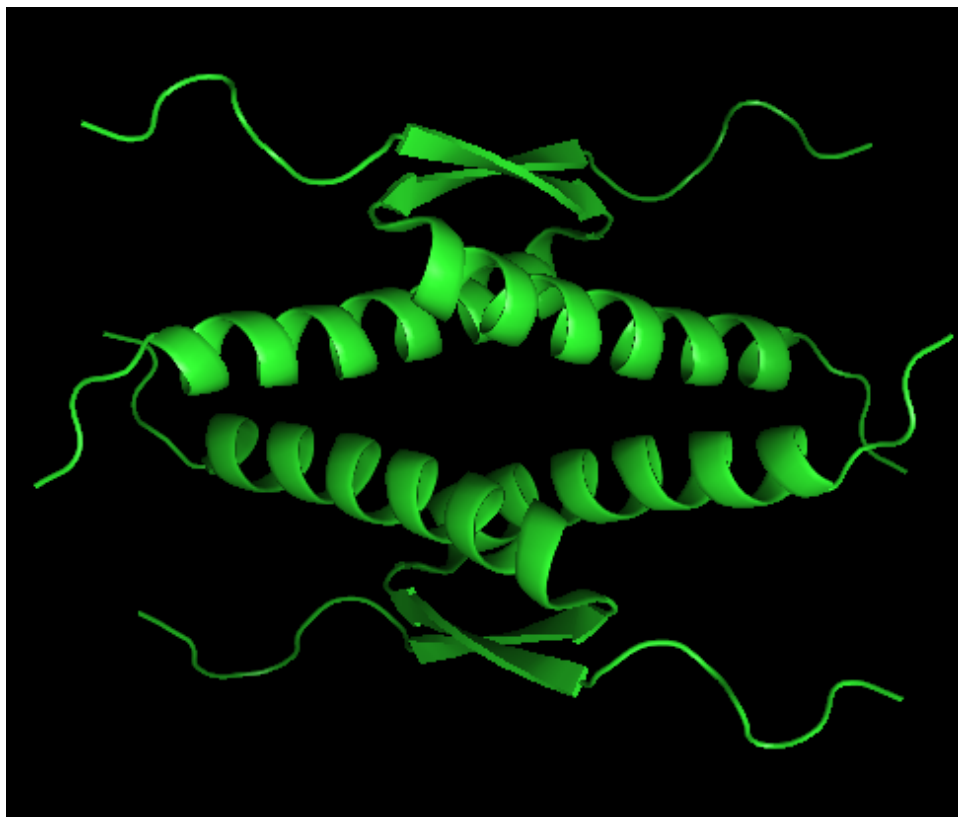


Figure 1: DNA binding domain of the human P53 tumor suppressor protein

2 Protein Structure Comparison Methods

When comparing protein structures, one wishes to quantify the distance between two structures in a way that is fast and reliable. Depending on the reason for comparing the proteins, one may wish to focus on different aspects of the comparison. For instance, in applications where one wishes to perform a clustering on all known proteins [3], a similarity measure needs to consider structures that may have wildly different folds while not being too expensive to compute.

Proteins also come in various shapes and sizes. Before any kind of comparison can be done involving interatomic distances, one must first determine which atoms are appropriate to compare. Once such a correspondence map between atoms can be determined, it is relatively straightforward to perform an alignment of the two structures and then compute a distance measure (discussed in more detail below). However, once an alignment has been made, it is possible to refine the correspondence map to obtain a better comparison. Iterating between correspondences and alignments leads to the Iterative Closest Point (ICP) algorithm [10], recently improved by Phillips et.al. [8].

3 Protein Structure Comparison Under Perfect Correspondence

In structure determination, one often wishes to compare putative protein structures to published reference models. These putative structures have exactly the same atoms as the reference, but they may be in different positions. In this situation, the atom correspondence problem is easily solved by ordering the atoms in the reference structure and using the same ordering in the putative structure. Thus, we can consider comparison without having to search over correspondences and have no need for the iterative approaches described above.

Two methods for reporting protein distances are widely used, cRMS and dRMS [6]. cRMS simply reports the root mean square of the sum of *interatomic* distances and is given by the formula below.

$$cRMS = \sqrt{\frac{1}{m} \sum_{i=1}^m \|\mathbf{a}_i - \mathbf{b}_i\|^2}$$

where \mathbf{a}_i and \mathbf{b}_i are the coordinates of atoms in two different proteins each having m atoms. On the other hand, dRMS measures the distances between the two distance matrices of the two structures.

$$dRMS = \sqrt{\frac{1}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m (d_{ij}^A - d_{ij}^B)^2}$$

where d_{ij}^A and d_{ij}^B are the distances between atoms i and j in proteins A and B respectively. Thus, dRMS represents comparisons of *intra-atomic* distances. Because dRMS distances have a quadratic dependence on m , they are expensive to compute. In the case of structure determination, we will focus only on cRMS distances and refer to them simply as RMSD.

In this special case of perfect correspondence, the optimal alignment of two proteins (i.e. translation and rotation) given an all-atom RMSD measure can be solved exactly and in closed form [5]. However, computing all-atom RMSD scores can be time-consuming with a large number of structures. By taking advantage of the geometry of the protein, one is able to compute other distance measures in much less time. For example, it is common practice to compute RMSD scores for structures based only on backbone atoms (e.g. C, C_α, N). The polypeptide nature of protein geometry ensures that the set of backbone atoms is a reasonable approximation to the set of all atoms and, by extension, backbone RMSD scores reasonably approximate all-atom RMSD scores.

3.1 Rigid Body Comparison

Proteins that form symmetric complexes, however, provide extra geometric properties that are useful. Each subunit of such proteins is identical to the other subunits in the configuration, but differ by only a translation and a rotation. It is therefore possible to use a simplified representation of the protein using rigid bodies

for the purposes of calculating similarity. The rigid body representation for a protein subunit is defined as follows.

Consider a protein with n subunits. Let $S_i = \{s_1^{(i)}, \dots, s_m^{(i)}\}$ be a set of m points representing atoms in subunit i . Let R_i be a rigid body that represents subunit i . R_i has a position $\mathbf{t}_{R_i} \in \mathbb{R}^3$ and orientation $\mathbf{r}_{R_i} \in SO(3)$. The position element of R_i is calculated to be the centroid of the subunit i and is given by the following equation.

$$\mathbf{t}_{R_i} = \frac{1}{m} \sum_{j=1}^m s_j^{(i)}$$

Let R_0 be the rigid body representation of the ‘‘origin’’ subunit of the protein. The rotation components for R_1, \dots, R_{n-1} are measured relative to R_0 . These rotations can be calculated using optimal alignment techniques [5], but it is much more efficient to exploit the symmetry of the protein complex to obtain them. To explain this, we will need to introduce a parameterization for D_n symmetric protein complexes.

3.2 Parameterization for D_n Symmetric Proteins

D_n symmetry can be characterized as having two axes of symmetry. One vector \mathbf{f} defines a rotation axis to transform one subunit into its dimeric pair. This rotation is always a 180° rotation. The other vector \mathbf{r} defines a rotation axis to transform each dimer around the ring of the protein complex (See Fig. 2). These axes are also constrained such that they are perpendicular to each other and always intersect. This leads to the following six-dimensional parameterization of \mathbf{r} and \mathbf{f} .

$$\begin{aligned} \mathbf{r} &= R_{\hat{\mathbf{z}}}(\theta)R_{\hat{\mathbf{y}}}(\phi)\hat{\mathbf{z}} \\ \mathbf{f} &= R_{\hat{\mathbf{z}}}(\theta)R_{\hat{\mathbf{y}}}(\phi)R_{\hat{\mathbf{z}}}(\rho)\hat{\mathbf{x}} \\ \mathbf{t} &= \langle x, y, z \rangle \end{aligned}$$

Here the parameters of the axes are $x, y, z, \theta, \phi, \rho$ and $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}$ are the unit axes. Additionally, let $R_{\mathbf{a}}(\theta)$ represent a rotation of θ about an axis \mathbf{a} . Figure 3 illustrates the parameterization of the axes and its degrees of freedom.

3.3 More Rigid Body Comparison

We can now use this parameterization to define the rotations needed to represent the orientations of R_1, \dots, R_n with respect to R_0 . P53, our reference protein, has 4 identical subunits and shows D_4 symmetry. Thus, $n = 4$ and we can define the rotations for R_1, \dots, R_{n-1} in terms of rotations about the axes of symmetry \mathbf{r}, \mathbf{f} and R_0 .

$$\begin{aligned} \mathbf{r}_{R_1} &= R_{\mathbf{f}}(180^\circ)\mathbf{r}_{R_0} \\ \mathbf{r}_{R_2} &= R_{\mathbf{r}}(180^\circ)\mathbf{r}_{R_0} \\ \mathbf{r}_{R_3} &= R_{\mathbf{r}}(180^\circ)R_{\mathbf{f}}(180^\circ)\mathbf{r}_{R_0} \end{aligned}$$

Since rigid bodies contain a rotational component as well as a translational component, the standard Euclidean distance will not correctly describe their similarity. Thus, we define the distance between two rigid bodies R, S as a weighted sum of the distances between their translational and rotational components.

$$d(R, S) = d_t(\mathbf{t}_R, \mathbf{t}_S) + \omega d_r(\mathbf{r}_R, \mathbf{r}_S)$$

where ω is a parameter to be determined later. As one might expect, it makes sense to use Euclidean distance to compare the translational components. However, defining rotational distance is less straightforward. When using quaternion math to represent rotations, Park and Ravani provide a natural similarity measure [7]. Let p, q be quaternions. Their distance $d_t(p, q)$ is defined as follows.

$$d_r(p, q) = \|\log(p^*q)\|$$

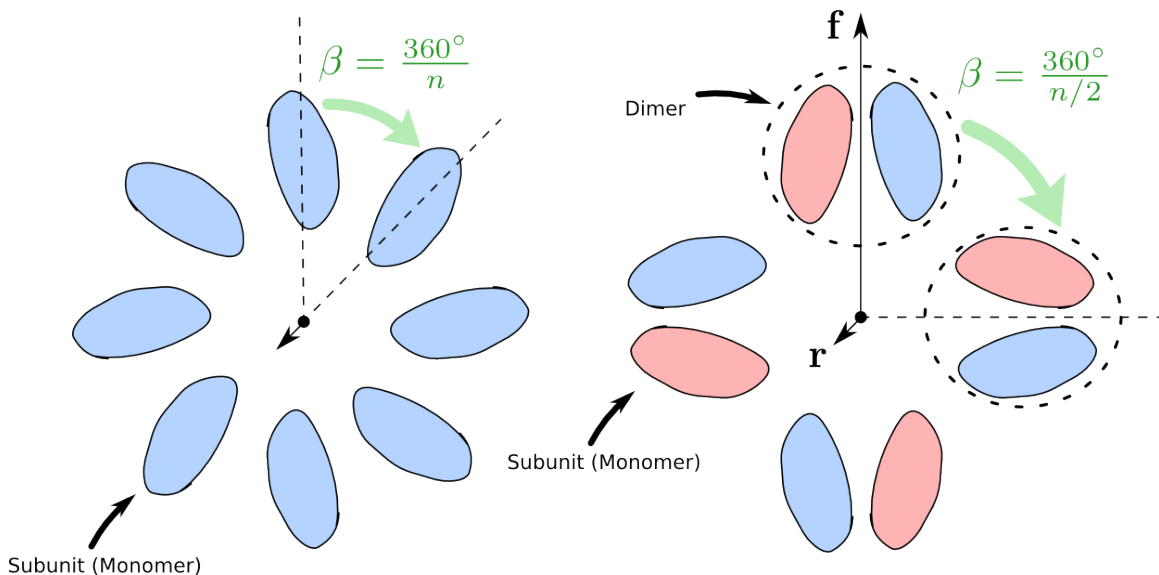


Figure 2: C_n symmetry (left): This symmetry is defined by a single rotation axis (coming out of the page). D_n symmetry (right): can be viewed as an oligomer of dimers. The dimer (marked with the dashed circle) is composed of identical subunits which are labeled red and blue to highlight their differences in orientation. The two axes of D_n symmetry \mathbf{r} , \mathbf{f} are also shown. \mathbf{r} is shown coming out of the page and is the central rotation axis. The \mathbf{f} axis defines the conformation of the dimer by specifying an axis of rotation to transform the red subunit into the blue subunit and vice versa. β in both cases represents the angle used to rotate subunits about the \mathbf{r} axis.

Intuitively, this represents the length of the short arc on the great circle of S^3 containing p and q . It can also be thought of as the shortest (curved) distance between two unit quaternions on S^3 . This formula can be simplified to avoid computing logarithms of quaternions.

$$d_r(p, q) = \arccos(p \cdot q)$$

This formulation suggests an interpretation where p, q are unit vectors in \mathbb{R}^4 and the distance is the angle between them measured in radians. However, since q and $-q$ represent the same rotation one must only consider positive values of the dot product $p \cdot q$ resulting in a final rotation distance equation.

$$d_r(p, q) = \arccos(|p \cdot q|)$$

4 Results

To test the effectiveness and speed of perfect correspondence protein structure comparison methods, we present complexity analysis and experimental results of five different approaches. Let P and Q be two proteins with the same number of atoms m and the same number of subunits n . Let p_i denote the i^{th} atom in P . Similarly, let $p_i^{(j)}$ denote the i^{th} atom in the j^{th} subunit of P .

All-Atom RMSD: This is the typical distance measure used for proteins under perfect correspondence. Each atom in the reference structure is compared to exactly one atom in the compared structure.

$$d_{AA}(P, Q) = \sqrt{\frac{1}{m} \sum_{i=1}^m \|p_i - q_i\|^2}$$

Backbone Atom RMSD: Backbone RMSD is computed in the same way as all-atom RMSD with the exception that only backbone atoms of the protein are compared. This typically includes only C, C_α

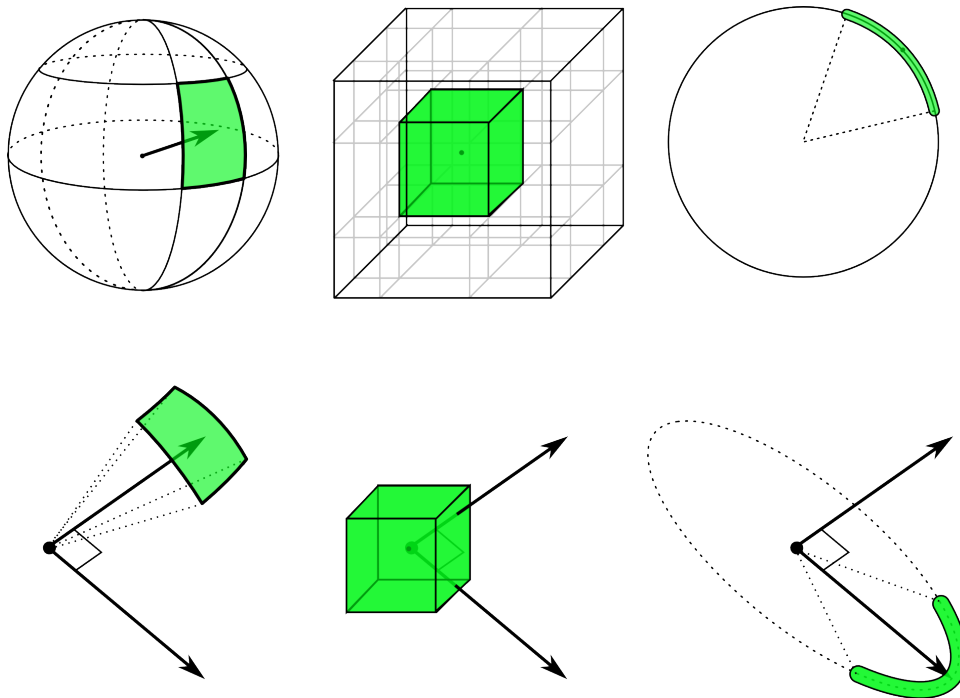


Figure 3: Highlighted in green are subsets of the configuration space for the D_n symmetry axes \mathbf{r} and \mathbf{f} . The top row shows these subsets in their originating spaces. The bottom row shows the effect each subset has on the position of the axes. The left column shows the parameters $(\theta, \phi) \in S^2$. Together they define an orientation for \mathbf{r} relative to $\hat{\mathbf{z}}$. The middle column shows the parameters $(x, y, z) \in \mathbb{R}^3$ and they define a position for the intersection of \mathbf{r} and \mathbf{f} . The right column shows the parameter $\rho \in S^1$. ρ defines the orientation of \mathbf{f} relative to \mathbf{r} and $\hat{\mathbf{x}}$.

and N atoms on the backbone. Side chain atoms and hydrogens are excluded from the measure. Backbone RMSDs are faster to compute than all atom RMSDs and still reasonably approximate all-atom RMSDs. Let $\{b_1, \dots, b_s\}$ be the set of backbone atom indices for the subunit. This distance formula is given by

$$d_{BA}(P, Q) = \sqrt{\sum_{j=1}^n \frac{1}{s} \sum_{i=1}^s \|p_{b_i} - q_{b_i}\|^2}$$

3 Atom RMSD: This method chooses three arbitrary (non-colinear) atoms from each subunit as a simplified representation of that subunit. In contrast to the previous two methods, this method depends on only the number of subunits and not the number of atoms in the protein. Since three atom are not enough to fully represent the position and orientation of the subunit, this method does not perform as well as the all-atom and backbone RMSD methods. This method is not used in practice and serves only as a negative control to compare against other methods. Let x, y, z represent the index of the arbitrarily-chosen atoms for a the subunit. The distance formula is then given by the following.

$$d_{3A}(P, Q) = \sqrt{\sum_{j=1}^n \frac{1}{3} \left(\|p_x^{(j)} - q_x^{(j)}\|^2 + \|p_y^{(j)} - q_y^{(j)}\|^2 + \|p_z^{(j)} - q_z^{(j)}\|^2 \right)}$$

1 Atom RMSD: This measure is calculated in the same way as the 3 Atom RMSD with the exception that only one atom is chosen from each subunit. Again, one atom is not enough to represent the orientation of the subunit and this method performs poorly. Let x represent the index of the arbitrarily-chosen

atom from the subunit.

$$d_{1A}(P, Q) = \sqrt{\sum_{j=1}^n \|p_x^{(j)} - q_x^{(j)}\|^2}$$

Rigid Body RMSD: This method is described in more detail in Sec. 3.1. The distance formula is as follows. Let R_i^P be the rigid-body representation for subunit i in protein P .

$$d_{RB}(P, Q) = \sqrt{\sum_{j=1}^n d(R_j^P, R_j^Q)^2}$$

For the weight parameter ω , a value of 2 was chosen. Since the range of d_r is $[0, \pi]$, choosing a higher weight can result in higher overall RMSD scores. However, this parameter had little effect on the penalty function used to evaluate the quality of the scores (See Sec.4.2).

4.1 Complexity

While all of the methods described above have linear asymptotic running time, it will still be beneficial to analyze the constants hidden behind the big-O notation and their relationship with the input parameters. The complexities for each method are given in Table 1.

RMSD Method	Complexity
All-Atom	$c_1 m$
Backbone Atom	$c_2 m$
3 Atom	$d_1 n$
1 Atom	$d_2 n$
Rigid Body	$c_3 m + d_3 n$

Table 1: Complexity analyses for the five distances measures described in Sec. 4.

The All-atom method obviously only depends on the number of atoms in the protein. However, the backbone method natively depends on the number of residues in the protein – from which a constant number of atoms are taken. Rather than introduce a third variable into the complexity analysis, we can simply treat the backbone method as operating on a reduced number of atoms in comparison to the all-atom method. Interestingly, the 3 and 1 atom methods do not rely on the number of atoms in the protein at all and result in fast execution times, but suffer from poor performance. The rigid body method seeks to find a balance between the dependence on m and n and compute a fast distance measure while still maintaining reliability.

Before running any experiments, one expects the constants above to have the following relationships.

$$c_1 > c_2 > c_3 \tag{1}$$

$$d_1 > d_2 \tag{2}$$

c_1 represents the amount of “work” per atom the all-atom method needs to do to compute a distance. Since the backbone method examines a smaller number of atoms, one expects it to do less work per atom in the protein. In contrast, the rigid body method only trivially examines each atom to compute centroids for each subunit. Thus, one expects that to be the smallest dependence on m possible. Simplified subunit representation methods also introduce a dependence on n , the number of subunits in the protein complex. While it is easy to see that the 3 atom method needs to do more work per subunit than the 1 atom method, a relationship between these two methods and the rigid body method is unclear.

4.2 Performance

To test the performance of these five methods, the following experiments were performed. From a set of 242 putative structures for P53, one structure was chosen arbitrarily from the set to be the reference structure. The published NMR structure for P53 could not be used because it was obtained using a simulated

annealing protocol and does not strictly adhere to the symmetry of the protein complex. Thus, it violates the assumptions of the simplified subunit representation methods. The remaining 241 structures were compared to the reference structure under three different alignment methods: no alignment, translational alignment¹, and translational and rotational alignment¹.

When comparing the performance of these methods, one should note that the actual distance value computed by the method is less important than the ranking (in order of increasing distance) of the compared structures. To capture that idea, the results for each method are normalized against the all-atom distance method in the following way. Let S be a sequence of k structures ranked in order of increasing all-atom distance score to the reference structure. By definition, a plot of the distance score curve of S must be monotonic increasing. For each other distance method, compute the distance scores in the same order as S . For that method to be successful, its score curve should also be monotonic increasing. To quantify the monotonicity of the score curves, the following penalty function $p(d)$ is used on a distance method d on proteins P, Q .

$$p(d) = \sqrt{\frac{1}{k} \sum_{i=1}^k (d(P_{i-1}, Q_{i-1}) - d(P_i, Q_i))^2 \cdot \mathbf{1}_{d(P_i, Q_i) < d(P_{i-1}, Q_{i-1})}} \quad (3)$$

The notation $\mathbf{1}_a$ defines a function $\mathbf{1}$ that returns 1 if a is true and 0 otherwise. This penalty function captures the RMS change in successive scores where the second score decreased relative to the all-atom scores. Due to the normalization, $p(d_{AA})$ is exactly zero. Penalties for other methods are positive where a smaller penalty is desired. Distance scores and penalties for atom-based methods are listed in angstroms. Distances and penalties for rigid bodies have no simple unit, but instead can be thought of as a combination of angstroms and radians.

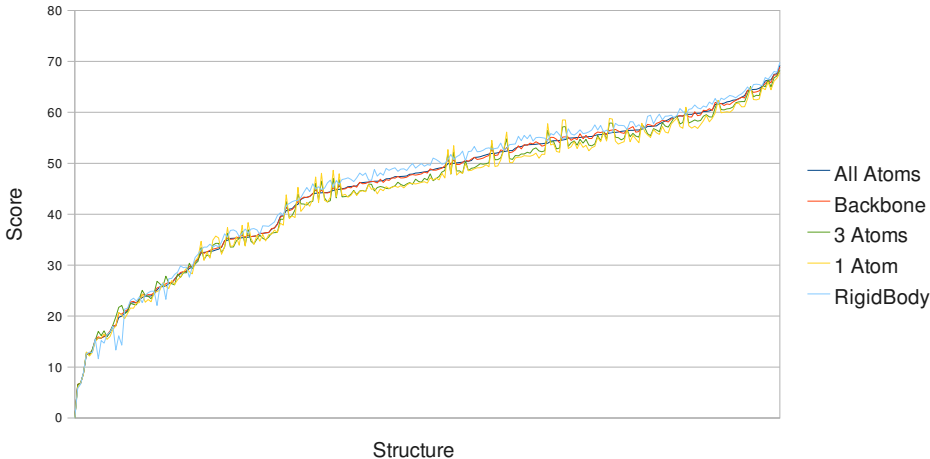


Figure 4: No Alignment: Distance score curves for all five methods in order of increasing all-atom distance score. Protein structures were not aligned before distance computation.

Without aligning the structures before computing distances, all methods perform reasonably well. Their score curves (See Fig. 4) are close to the all-atom curve. The backbone method has the smallest penalty (See Table 2) and is the closest approximation to the all-atom score. Despite not being able to represent subunit orientation, the 1 and 3 atom methods still perform surprisingly well. The rigid body method outperforms the 3 and 1 atom methods, but falls behind the backbone method.

Under a translational alignment (See Fig. 5), we begin to see the 3 and 1 atom methods fall behind the all-atom method. The backbone distance still remains the best approximation to the all-atom distance, but the rigid-body method also suffers here.

¹For the atom-based methods, this alignment is optimal [5]. However, for the rigid body method, it is sub-optimal. An optimal alignment method for rigid bodies would depend on the weight parameter ω and is not discussed here, but is mentioned briefly in Sec. 5.

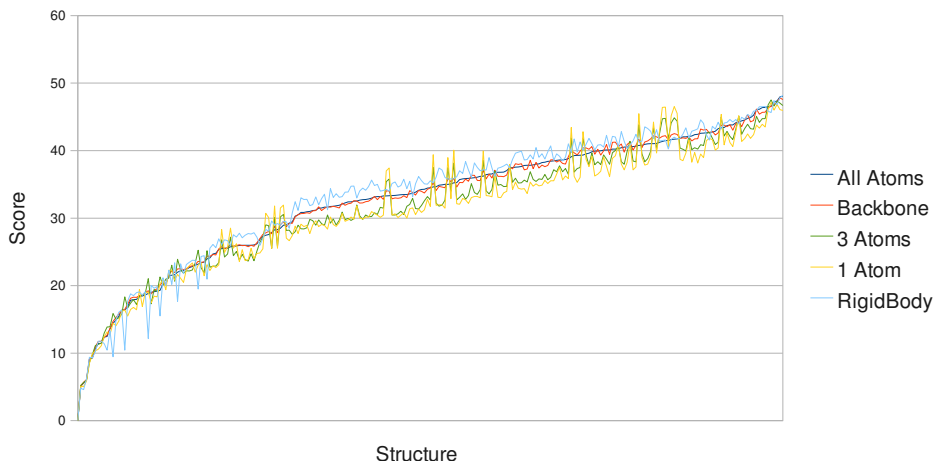


Figure 5: Translational Alignment: Distance score curves for all five methods in order of increasing all-atom distance score. Protein structures were aligned before distance computation using the centroid superposition method [5].

Method	No Alignment	Translational Alignment	Translational and Rotational Alignment
Backbone	0.25	0.34	0.46
3 Atom	0.96	1.75	1.43
1 Atom	1.39	1.24	3.70
Rigid Body	0.65	1.05	2.48

Table 2: Penalties for the distance methods: By definition, the all-atom penalty is exactly zero and is omitted from these results. A lower penalty is better. The penalty function is defined by Eqn. 3.

Adding in rotational alignment (See Fig. 5) causes the 1 atom method to fail spectacularly (as was expected). The position of a single atom per subunit is not enough information to encode its orientation. The 3 atom method also suffers, but less so than the 1 atom method because 3 atom positions allow this subunit representation to partially encode the orientation. Surprisingly, the rigid body method also suffers from poor performance despite modeling subunit orientation explicitly.

In terms of execution time (See Table 3), the rigid body method outperforms the other methods and is even 2 orders of magnitude faster than the all-atom method. The 3 and 1 atom methods are also 2 orders of magnitude faster than the all-atom method, but not quite as fast as the rigid body method. While not as fast as the simplified subunit representation methods, the backbone method is one order of magnitude faster than the all-atom method.

Method	No Alignment	Translational Alignment	Translational and Rotational Alignment
All Atom	74.157	90.083	156.133
Backbone	8.645	10.699	19.097
3 Atom	0.488	0.568	1.261
1 Atom	0.299	0.334	0.829
Rigid Body	0.209	0.215	0.900

Table 3: Execution times for the distance methods: All times are listed in seconds and report the time needed to perform 24,200 comparisons. Experiments were run on a single-core AMD 3700+ processor clocked at 1GHz.

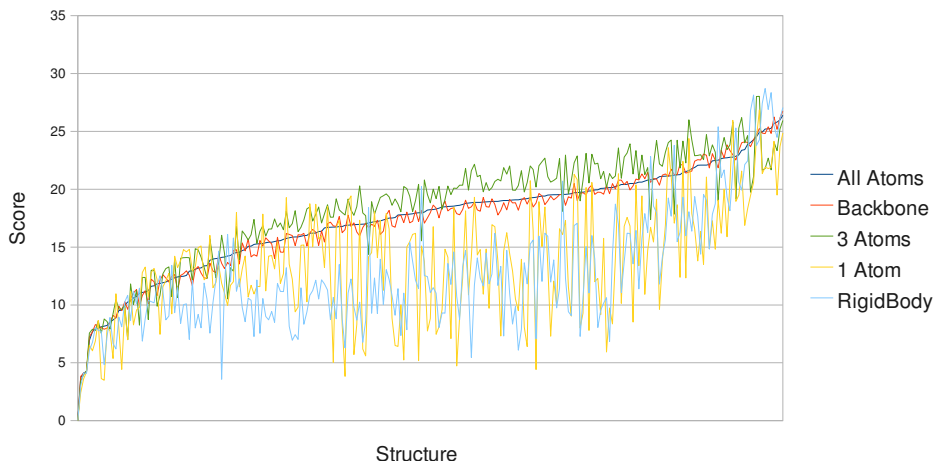


Figure 6: Translational and Rotational Alignment: Distance score curves for all five methods in order of increasing all-atom distance score. Protein structures were aligned before distance computation using the centroid superposition method along with the optimal rotation for point sets [5].

5 Conclusions

Even though the rigid body method enjoyed very fast computational speeds, it suffered from poor accuracy in comparison to the all-atom method under any type of alignment. However, under the no alignment situation, the rigid body method performed as well as the other methods. While optimal alignment only helps the all-atom, backbone, and 3-atom methods, it appears to hurt the 1-atom and the rigid body method. In the case of the 1-atom method, this is most likely due to the inability of that method to represent the orientation of the subunits. However, in the case of the rigid body method (where subunit orientation is explicitly modeled), this failure is likely due to the sub-optimality of the alignment algorithm.

Stated formally, the alignment problem for rigid bodies is as follows. Given two sets of rigid bodies $P = \{p_1, \dots, p_n\}$, $Q = \{q_1, \dots, q_n\}$ and a weight parameter ω , find the translation \mathbf{a} and rotation \mathbf{b} that minimizes the distance function

$$d_{RB}(P, Q) = \sqrt{\frac{1}{n} \sum_{i=1}^n (||t_{p_i} - (\mathbf{b}t_{q_i} + \mathbf{a})|| + \omega \arccos(|\mathbf{r}_{p_i} \cdot \mathbf{b}\mathbf{r}_{q_i}|))^2}$$

However, the rigid body method is the fastest method presented here. This is largely because of a minimal dependence on the number of atoms in the protein complex. If centroids for the subunits in the protein complex could be supplied to the rigid body method a priori, it would depend entirely on the number of subunits without any dependence on the number of atoms. While this may be interesting theoretically, in practice, proteins aren't able to reach a size where computing centroids becomes prohibitively expensive.

In conclusion, the rigid body method is a fast, but poor approximation to the all-atom method in realistic situations (i.e. under optimal alignment). At the expense of a marginal amount of computational time, the 3 atom method provides higher quality distance measures than the rigid body method. If allowed to represent subunits with 4 (non-coplanar) atoms, that method would likely be further improved with little extra computational cost. If accuracy is needed, the backbone method provides an order-of-magnitude-faster computation with little reduction in quality. If the accuracy of the rigid body method could be improved, it would serve as a fast approximation to the all-atom RMSD for protein complexes.

References

- [1] M.S. Apaydin, V. Conitzer, and B.R. Donald. Structure-based protein NMR assignments using native structural ensembles. *J. Biomol. NMR*, 40:263–276, Apr 2008.

- [2] Muhandiram DR, Farrow NA, Xu G-Y, Smallcombe SH, and Kay LE. A gradient ^{13}C NOESY-HSQC experiment for recording NOESY spectra of ^{13}C -labeled proteins dissolved in H_2O . *J Magn Res*, 102B:317–321, 1993.
- [3] J.F. Gibrat, T. Madej, and S.H. Bryant. Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, 6:377–385, Jun 1996.
- [4] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput*, 21:65–84, 1992.
- [5] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices, 1988.
- [6] P. Koehl. Protein structure similarities. *Curr. Opin. Struct. Biol.*, 11:348–353, Jun 2001.
- [7] F. C. Park and Bahram Ravani. Smooth invariant interpolation of rotations. *ACM Trans. Graph.*, 16(3):277–295, 1997.
- [8] Jeff M. Phillips, Ran Liu, and Carlo Tomasi. Outlier robust icp for minimizing fractional rmsd. *CoRR*, abs/cs/0606098, 2006.
- [9] S. Potluri, A.K. Yan, J.J. Chou, B.R. Donald, and C. Bailey-Kellogg. Structure determination of symmetric homo-oligomers by a complete search of symmetry configuration space, using NMR restraints and van der Waals packing. *Proteins*, 65:203–219, Oct 2006.
- [10] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.