Journal of Global Optimization **26:** 321–333, 2003.
© 2003 *Kluwer Academic Publishers. Printed in the Netherlands.*

A Geometric Build-Up Algorithm for Solving the Molecular Distance Geometry Problem with Sparse Distance Data

QUNFENG DONG[1] and ZHIJUN WU[2]
[1]*Department of Zoology and Genetics, Iowa State University, Ames, IA 50010, USA. (e-mail: qfdong@iastate.edu)*
[2]*Department of Mathematics and Graduate Program on Bioinformatics and Computational Biology, Iowa State University, Ames, IA 50010, USA. (e-mail: zhijun@iastate.edu)*

(Received 25 April 2002; accepted in revised form 29 November 2002)

Abstract. Nuclear magnetic resonance (NMR) structure modeling usually produces a sparse set of inter-atomic distances in protein. In order to calculate the three-dimensional structure of protein, current approaches need to estimate all other "missing" distances to build a full set of distances. However, the estimation step is costly and prone to introducing errors. In this report, we describe a geometric build-up algorithm for solving protein structure by using only a sparse set of inter-atomic distances. Such a sparse set of distances can be obtained by combining NMR data with our knowledge on certain bond lengths and bond angles. It can also include confident estimations on some "missing" distances. Our algorithm utilizes a simple geometric relationship between coordinates and distances. The coordinates for each atom are calculated by using the coordinates of previously determined atoms and their distances. We have implemented the algorithm and tested it on several proteins. Our results showed that our algorithm successfully determined the protein structures with sparse sets of distances. Therefore, our algorithm reduces the need of estimating the "missing" distances and promises a more efficient approach to NMR structure modeling.

Key words: Molecular distance geometry, Protein structure determination, Numerical linear algebra and optimization

1. Introduction

Many of the research subjects in biology focus on properties and activities of cells that are primarily determined by proteins. Proteins are biopolymers made up of twenty different amino acids, each having an acid group, an amino group, and a side chain. The order of the amino acids and the properties of their side chains in a protein determine a three-dimensional structure. The structure specifies the function of the protein (Branden and Tooze, 1991).

The structure of a protein may be determined experimentally via NMR spectroscopy or X-ray crystallography or theoretically through potential energy minimization or molecular dynamics simulation (Creighton, 1993). We study a problem related to the NMR approach to structure determination. More specifically, we consider the problem of determining the structure of a protein with a set of distances between pairs of atoms in the protein. The distances are either obtained with our knowledge on certain bond lengths and bond angles or estimated through NMR experiments. Solving protein structure based on the distance data is generally called molecular distance geometry problem (Crippen and Havel, 1988. A particular case of molecular distance geometry problem is when exact distances between all pairs of atoms are given. The problem can then be solved by factorizing a distance matrix formed by the given distances. More specifically, we can define a special matrix with the given distances. If the distances are consistent in the sense that we can indeed find a set of feasible points in three-dimensional space, the distance matrix must be of rank less than or equal to three. If we can find the three non-zero eigenvalues of the matrix, we can use the eigenvectors to find the coordinates of the points (Blumenthal, 1953; Crippen and Havel, 1988). Note that computing the eigenvalues of an n by n matrix can be done in $O(n^2)$ to $O(n^3)$ floating point operations by using the singular value decomposition (Golub and Van Loan, 1989). So, the molecular distance geometry problem, when all exact distances are given, can be solved in polynomial time and is a tractable problem. However in practice, we usually cannot obtain the exact distances between all pairs of atoms in protein. For example, in NMR experiments, usually only the distances between certain close-range hydrogen atoms can be measured (Creighton, 1993). Therefore, the real challenge in NMR structure determination is to solve a molecular distance geometry problem with only sparse sets of distance data resulted from NMR experiments in addition to our knowledge on certain bond lengths and bond angles. Two major heuristic approaches have been used to solve the sparse distance geometry problem. One approach, represented by the EMBED algorithm of Crippen and Havel, first estimates the “missing” distances to build a full set of distances. The problem is then converted to one with all distances and solved with singular value decomposition as described above followed by necessary error minimization (Crippen and Havel, 1988; Glunt et al., 1993; Havel, 1995). However, the estimation step is costly and prone to introducing errors. The other approach directly solves the problem as a global least-squares problem (Hendrickson, 1991; Moré and Wu, 1996, 1997, 1999;). The least-squares function is defined with respect to only the given distances, for example,

$$f(x_1, \ldots, x_n) = \sum_{(i,j)\in S} [(\|x_i - x_j\|^2 - d_{i,j}^2)]^2,$$

where S is the set of given distances, and $d_{i,j}$ the distance between atoms i and j. Given this f, it is easy to see that a set of coordinates $x_1, \ldots, x_n$ is a solution to the molecular distance geometry problem if and only if it is the global minimizer of f with the global minimum equal to zero. However, the global minimizer has been proved to be difficult to find, even for some simple problem instances (Moré and Wu, 1997, 1999). In this report, we describe a geometric build-up algorithm for

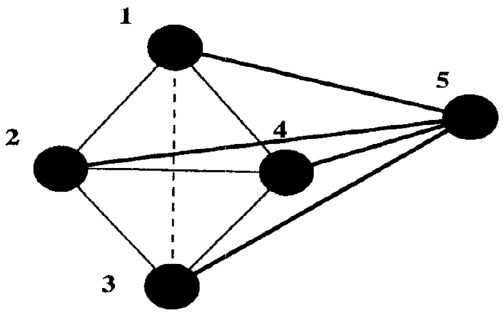


Figure 1. An atom determined with its distances to four base atoms.

the molecular distance geometry problem with sparse distance data. The algorithm does not require recovering the "missing" distances and works with only the sparse distance data. It utilizes a simple geometric relationship between coordinates and distances. The coordinates for each atom are calculated by using the coordinates of previously determined atoms and their distances.

2. The Algorithm for Full Sets of Distances

Our geometric build-up algorithm is based on the same idea for the linear-time algorithm we have previously reported for solving the molecular distance geometry problem with full sets of distances. We describe the idea of the algorithm briefly in this section. For more details, readers are referred to (Dong and Wu., 2002). Some related work can also be found in a recent paper in (Huang et al., 2001).

Consider a molecule of n atoms. A simple idea for determining the coordinates of the atoms with the distances among them is the following. First, if there are four atoms not in the same plane, the coordinates of the atoms can be calculated easily by using the distances among the atoms. Then, the four atoms, we call the base atoms, can be used to determine uniquely the coordinates of any of the remaining atoms in the molecule, given the distances among the base atoms and the atom to be determined (see Figure 1). For each of the atoms, the algorithm determines the coordinates for it by solving a small and simple system of algebraic equations. The amount of computation is proportional to the number of the atoms in the molecule.

3. The Algorithm for Sparse Sets of Distances

For problems with full sets of distances, the same set of base atoms can be used repeatedly to fix all other atoms in the molecule, and the structure of the molecule can be determined in n computational steps, where n is the number of atoms in the molecule. However, for problems with sparse sets of distances, a set of base atoms may not be used to fix all other atoms since some distances from the base

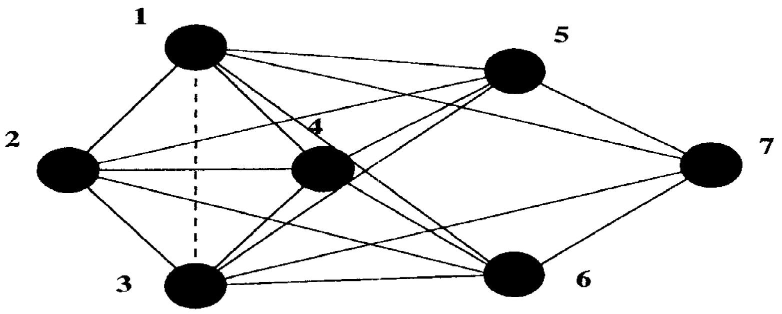
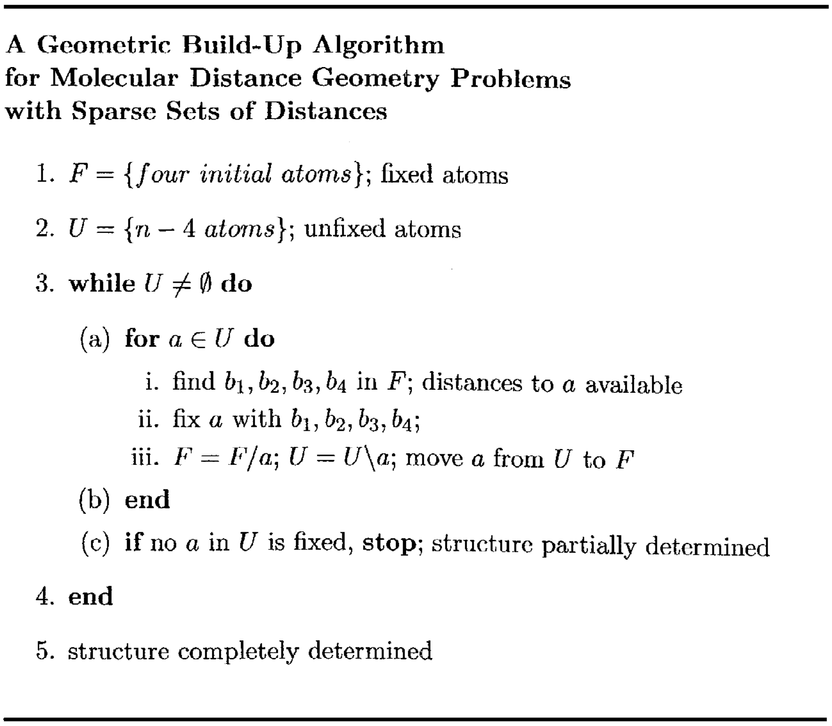


Figure 2. A set of atoms with sparse distances determined with different bases.

atoms to the atom to be determined may not be provided. Thus we cannot directly use the previous algorithm for problems with sparse sets of distances. However, we can still use the same idea to determine the coordinates for an atom with any four determined ones whenever the distances among the atoms are available. More specifically, given a sparse set of distances, assume that we can first fix the coordinates for at least four atoms. We can then examine each of the remaining unfixed atoms to find four fixed atoms such that the distances between any of the four fixed atoms and the unfixed one are known. If such four atoms are found, the coordinates for the unfixed atom can immediately be determined. The algorithm may continue until all the atoms are fixed. Figure 2 illustrates how such an algorithm can be used to find the coordinates of seven atoms with a sparse set of distances given among them. Atoms 1, 2, 3 and 4 can be used as the first set of base atoms. The coordinates of 1, 2, 3 and 4 can be calculated using the distances among them. Then, the coordinates of atoms 5 and 6 can be calculated with their distances to the base atoms by using our algorithm. Now atoms 1, 3, 5 and 6 can be used as a new set of base atoms to calculate the coordinates of atom 7 as long as atoms 1, 3, 5, and 6 are not in the same plane. Obviously, such a construction can be extended to any given number of atoms with a sparse set of distances among them.

An outline of our algorithm for sparse sets of distances is given in the diagram in Figure 3. Note that the algorithm does not guarantee to solve any distance geometry problem. In every loop, the algorithm requires that at least one of the unfixed atoms can be determined by using four of the fixed atoms. Otherwise the algorithm will stop and report only a partial structure, which is composed of the coordinates of the atoms being fixed so far. For this reason, the **while** loop in the algorithm will be executed at most n times. The **for** loop in the algorithm examines all unfixed atoms and takes at most n steps. Then for each unfixed atom, the algorithm searches for four fixed atoms to form a set of base atoms. This will need at most n steps since there are at most n fixed atoms. Here for each of the fixed atoms, the algorithm basically needs to find out if there is a distance from the fixed atom to the unfixed one. This will require at most n steps of calculations. Once four base atoms are found, the coordinates for the unfixed atom can be determined by using our ba-



A Geometric Build-Up Algorithm for Molecular Distance Geometry Problems with Sparse Sets of Distances

```
1. $F = \{four\ initial\ atoms\}$; fixed atoms
2. $U = \{n - 4\ atoms\}$; unfixed atoms
3. while $U \neq \emptyset$ do
   (a) for $a \in U$ do
       i. find $b_1, b_2, b_3, b_4$ in $F$; distances to $a$ available
       ii. fix $a$ with $b_1, b_2, b_3, b_4$;
       iii. $F = F/a$; $U = U\backslash a$; move $a$ from $U$ to $F$
   (b) end
   (c) if no $a$ in $U$ is fixed, stop; structure partially determined
4. end
5. structure completely determined
```

Figure 3. A geometric build-up algorithm for problems with sparse distances.

sic distance-coordinate build-up algorithm. Together, the algorithm requires $O(n^4)$ steps of calculations to complete. Further reduction in the total computation steps is possible but depends on specific implementation.

4. Computational Issues

4.1. FIXING THE FOUR BASE ATOMS

We can use any atom, i.e., the first atom in the molecule, as the first base atom. Let u_1, v_1, and w_1 be the three coordinates for the atom. We can set $u_1 = 0$, $v_1 = 0$, and $w_1 = 0$. Then the second atom in the molecule can be used as the second base atom and be fixed on one of the axes, i.e., the first axis, by setting $u_2 = d_{1,2}$, $v_2 = 0$, and $w_2 = 0$, where $d_{1,2}$ is the distance between atoms 1 and 2. The third base atom is put into one of the planes formed by the axes, i.e., the one by the first and second axes. Therefore, the third coordinate for the atom w_3 is set to zero. The other two coordinates are determined by using the distances of the atom to the first two atoms:

$$
\begin{aligned}
u_3^2 + v_3^2 &= d_{3,1}^2 \\
(u_3 - u_2)^2 + v_3^2 &= d_{3,2}^2,
\end{aligned}
$$

and therefore,

$$
\begin{aligned}
u_3 &= (d_{3,1}^2 - d_{3,2}^2 + u_2^2)/(2u_2) \\
v_3 &= \pm(d_{3,1}^2 - u_3^2)^{1/2}.
\end{aligned}
$$

Here, v_3 cannot be zero in order to avoid being on the same line determined by the first two atoms. Therefore, the third base atom can be selected among the remaining atoms if its calculated v_3 is nonzero. Since v_3 can be either positive or negative without affecting the final structure, we always choose v_3 to be positive. Finally, the fourth base atom can be fixed by solving the following equations.

$$
\begin{aligned}
u_4^2 + v_4^2 + w_4^2 &= d_{4,1}^2 \\
(u_4 - u_2)^2 + v_4^2 + w_4^2 &= d_{4,2}^2 \\
(u_4 - u_3)^2 + (v_4 - v_3)^2 + w_4^2 &= d_{4,3}^2,
\end{aligned}
$$

and

$$
\begin{aligned}
u_4 &= (d_{4,1}^2 - d_{4,2}^2 + u_2^2)/(2u_2) \\
v_4 &= (d_{4,2}^2 - d_{4,3}^2 - (u_4 - u_2)^2 + (u_4 - u_3)^2 + v_3^2)/(2v_3) \\
w_4 &= \pm(d_{4,1}^2 - u_4^2 - v_4^2)^{1/2}.
\end{aligned}
$$

Obviously, w_4 cannot be zero in order to avoid being in the same plane determined by the first three atoms. Therefore, the fourth base atom can be selected among the remaining atoms if its calculated w_4 is nonzero. Here w_4 can either be positive or negative, corresponding to two mirror symmetric structures. We can compute one of the structures with w_4 positive. The second one can be obtained by simply making all w_i, $i = 4$, to have an opposite sign.

4.2. DETERMINING THE REMAINING ATOMS

Let us assume that we have found the coordinates for the four base atoms. Let the coordinates be denoted by

$$
\begin{aligned}
x_1 &= (u_1, v_1, w_1)^T \\
x_2 &= (u_2, v_2, w_2)^T \\
x_3 &= (u_3, v_3, w_3)^T \\
x_4 &= (u_4, v_4, w_4)^T.
\end{aligned}
$$

Suppose that we want to determine the coordinates $x_i = (u_i, v_i, w_i)^T$ for some atom i and we know the distances among atom i and the four base atoms. Let the distances be denoted by $d_{i,j}$ for $j = 1, 2, 3, 4$. We then have the following equations.

$$\begin{aligned}
\|x_i - x_1\| &= d_{i,1}\\
\|x_i - x_2\| &= d_{i,2}\\
\|x_i - x_3\| &= d_{i,3}\\
\|x_i - x_4\| &= d_{i,4},
\end{aligned}$$

which are equivalent to

$$\begin{aligned}
\|x_i - x_1\|^2 &= \|x_i\|^2 - 2x_i^T x_1 + \|x_1\|^2 = d_{i,1}^2\\
\|x_i - x_2\|^2 &= \|x_i\|^2 - 2x_i^T x_2 + \|x_2\|^2 = d_{i,2}^2\\
\|x_i - x_3\|^2 &= \|x_i\|^2 - 2x_i^T x_3 + \|x_3\|^2 = d_{i,3}^2\\
\|x_i - x_4\|^2 &= \|x_i\|^2 - 2x_i^T x_4 + \|x_4\|^2 = d_{i,4}^2,
\end{aligned}$$

and

$$\begin{aligned}
\|x_i\|^2 - 2u_i u_1 - 2v_i v_1 - 2w_i w_1 + \|x_1\|^2 &= d_{i,1}^2\\
\|x_i\|^2 - 2u_i u_2 - 2v_i v_2 - 2w_i w_2 + \|x_2\|^2 &= d_{i,2}^2\\
\|x_i\|^2 - 2u_i u_3 - 2v_i v_3 - 2w_i w_3 + \|x_3\|^2 &= d_{i,3}^2\\
\|x_i\|^2 - 2u_i u_4 - 2v_i v_4 - 2w_i w_4 + \|x_4\|^2 &= d_{i,4}^2.
\end{aligned}$$

Twelve different systems of linear equations can be derived from the above four non-linear equations by subtracting them each other. For example, we can subtract the first equation from the rest ones to obtain

$$\begin{aligned}
2u_i(u_1-u_2)+2v_i(v_1-v_2)+2w_i(w_1-w_2) &= (\|x_1\|^2-\|x_2\|^2)-(d_{i,1}^2-d_{i,2}^2)\\
2u_i(u_1-u_3)+2v_i(v_1-v_3)+2w_i(w_1-w_3) &= (\|x_1\|^2-\|x_3\|^2)-(d_{i,1}^2-d_{i,3}^2)\\
2u_i(u_1-u_4)+2v_i(v_1-v_4)+2w_i(w_1-w_4) &= (\|x_1\|^2-\|x_4\|^2)-(d_{i,1}^2-d_{i,4}^2).
\end{aligned}$$

Mathematically, it's easy to solve the three unknowns u_i, v_i, and w_i by solving the above equations. However, there are 12 different systems of linear equations that can be derived from the non-linear equations. Even though they are equivalent mathematically, they can be quite different computationally. In matrix form, the equations are equivalent to

$$Ax_i = b_i,$$

where

$$A = 2\begin{pmatrix} u_1 - u_2 & v_1 - v_2 & w_1 - w_2\\ u_1 - u_3 & v_1 - v_3 & w_1 - w_3\\ u_1 - u_4 & v_1 - v_4 & w_1 - w_4 \end{pmatrix},$$

and

$$b_i \;=\; \begin{pmatrix} (\|x_1\|^2 - \|x_2\|^2) - (d_{i,1}^2 - d_{i,2}^2) \\ (\|x_1\|^2 - \|x_3\|^2) - (d_{i,1}^2 - d_{i,3}^2) \\ (\|x_1\|^2 - \|x_4\|^2) - (d_{i,1}^2 - d_{i,4}^2) \end{pmatrix}.$$

The system of equations has a unique solution if the coefficient matrix A is nonsingular, or in other words the determinant of A is not equal to zero. Furthermore, for numerical stability, it will be better if the determinant of A is not even close to zero. Geometrically, it means that the volume of the simplex formed by the three column vectors of A must not be equal to or close to zero. For this reason, we choose among twelve possible ones a system that has the biggest absolute value of the determinant of A and then use it to determine the coordinates u_i, v_i, and w_i.

4.3. SOLVING THE LINEAR EQUATIONS

As we have described above, to determine the coordinates for an atom by using a set of base atoms, a system of linear equations needs to be solved. To solve such a system of equations, a special procedure similar to Gauss elimination can be used as can be illustrated in the following example.

Given three linear equations with three unknown, u, v, and w,

$$\begin{aligned} a_1u + b_1v + c_1w &= f_1 && (1)\\ a_2u + b_2v + c_2w &= f_2 && (2)\\ a_3u + b_3v + c_3w &= f_3, && (3) \end{aligned}$$

if a_1 is not smaller than a_2 and a_3 and is not equal to zero, then u can be eliminated:

$$\begin{aligned} (1) * (a_2/a_1) - (2) : &\quad (b_1 * a_2/a_1 - b_2)v + (c_1 * a_2/a_1 - c_2)w = f_1 * a_2/a_1\\ (1) * (a_3/a_1) - (3) : &\quad (b_1 * a_3/a_1 - b_3)v + (c_1 * a_3/a_1 - c_3)w = f_1 * a_3/a_1. \end{aligned}$$

Let the coefficients in the equations be denoted by B_i, C_i, and F_i, $i = 1, 2$. The equations can be written in the following form with two unknowns $(v,\ w)$:

$$\begin{aligned} B_1v + C_1w &= F_1 && (4)\\ B_2v + C_2w &= F_2. && (5) \end{aligned}$$

Here, if B_1 is not smaller than B_2 and is not equal to zero, then v can be eliminated:

$$(4) * B_2/B_1 - (5) : \qquad (C_1 * B_2/B_1 - C_2)w = F_1 * B_2/B_1 - F_2.$$

Thus, w can be easily solved. By substituting w into (4), $v = (F_1 - C_1w)/B_1$, and v and w into (1), $u = (f_1 - b_1v - c_1w)/a_1$. The system of linear equations is then solved.

Table 1. Results for different proteins at different cutoff distances

Proteins	No. of atoms	8 Å	10 Å	12 Å	14 Å	16 Å
1PTQ:	402	2.84e-08	2.03e-10	5.65e-12	5.65e-12	4.31e-12
1HOE:	558	9.38e-06	1.28e-09	5.69e-08	2.07e-09	1.49e-11
1LFB:	641	———	1.28e-07	1.33e-08	5.22e-09	1.43e-08
1F39A:	767	2.28e-06	4.11e-07	3.61e-08	8.08e-10	3.05e-09
1PHT:	814	4.39e-05	5.25e-07	6.69e-08	5.38e-09	1.38e-08
1POA:	914	———	3.13e-07	7.30e-08	2.18e-09	1.31e-11
1AX8:	1003	1.45e-06	2.63e-06	6.00e-08	1.34e-07	1.51e-08
1RGS:	2015	———	5.64e-05	4.52e-07	1.07e-06	1.10e-06
1BPM:	3672	———	2.99e-04	1.82e-04	1.45e-06	1.74e-06
1HMV:	4200	———	1.11e-04	3.49e-04	1.94e-04	7.86e-06

5. Computational Results

We have implemented our algorithm in C++ and tested it with a set of model problems on a UNIX workstation. The distance data was generated by using the structural data for a set of proteins downloaded from the Protein Data Bank (PDB) (Berman et al., 2000). For each of the proteins, the distances were calculated with certain cutoff value so that only a subset of distances were obtained. Table 1 shows the results of using our algorithm to find the coordinates for a set of proteins given their generated distances. Note that we measured the distance matrix error (DME) between the original and calculated structures. The DME values are listed in the table for different proteins with different sets of distances each obtained by using a specific cutoff distance. From this table, we can see that our algorithm solved the model problems reasonably well. The problem sizes ranged from about four hundreds to four thousands of atoms. The distances were cut off at 8, 10, 12, 14, and 16 Å. Usually, the distance data obtained from NMR experiments are less than or equal to 5 Å. We used longer distances not only because that we wanted to see how the algorithm performs for a spectra of cutoff distances, but also because that the PDB files we used do not include hydrogen atoms which otherwise would add more short-range distances. Because of the absence of the hydrogen atoms, only partial structures were obtained for some of the proteins with an 8 Å cutoff distance. However, complete structures were obtained for all of the proteins with increasing the cutoff distance to 10 Å and beyond.

As an example, we give more detailed information for the last test problem, 1HMV, which is related to an important protein called HIV-1 RT. Human immunodeficiency virus type (HIV-1) is known to be the etiological agent of the acquired immunodeficiency syndrome (AIDS). The reverse transcriptase (RT) of HIV-1 is responsible for converting the viral genome RNA into DNA, which is the key step

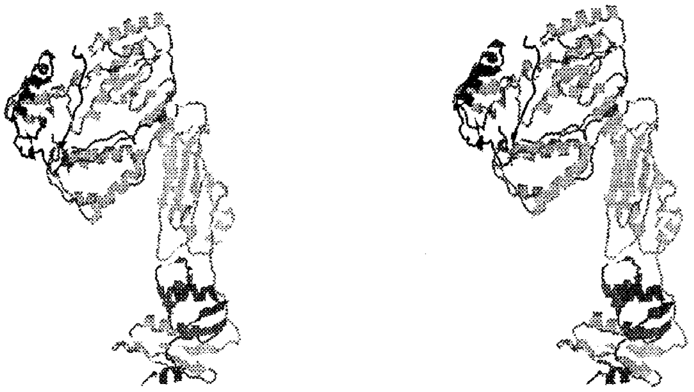


Figure 4. The original (left) and computed (right) structures for the HIV-1 RT p66 protein.

for viral replication. The 66-kDa subunit of the protein, p66, consists of both the DNA polymerase domain and the RNase H domain (Le Grice et al., 1991; Telesnitsky and Goff, 1997). In order to test our algorithm, we have retrieved the X-ray structural data for p66 (1HMV.pdb) from the PDB. The structural data deposited in the PDB contains the coordinates of the atoms in the molecule. Using the retrieved coordinates, we first calculated a full set of distances between all pairs of atoms. The PDB file for HIV-1 RT p66 subunit contains the coordinates for 4200 atoms. The distances range from smaller than 2 Åto larger than 90 Å, depending on the relative positions of the atoms, and they can be put into a 4200 × 4200 matrix. We then generated subsets of distances by removing large distances with certain cutoffs. For example, if the cutoff is 10 Å, a subset of distances was obtained which contained only distances less than 10 Å. We applied our algorithm to different sets of distances and obtained a structure for the protein based on each set of distances. We then compared the structures with the original one. Figure 4 shows the structure we obtained for a set of distances with a 10 Åcutoff. The structures are displayed by the graphic function, which is implemented by integrating RasMol (Sayle and Milner-White, 1995) into our program. The original (left) and computed (right) structures match successfully. Their distance matrix error (DME) is equal to 1.1e-04.

6. Discussion

The distance data derived from the atomic interaction measured by NMR can be used to calculate the three-dimensional structure of protein. If the distances between all pairs of atoms are available, a unique protein structure can then be computed in polynomial time. Previously, we reported a linear time algorithm for solving such a problem. Our linear time algorithm was based on a simple geometric relationship between coordinates and distances. However, usually the

NMR method can only provide a sparse set of distances among atoms in protein. Even though we can combine the NMR data with our knowledge on certain bond lengths and angles, we still cannot obtain all the distances. Heuristic algorithms, such as the EMBED algorithm, have been developed to estimate the "missing" distances. Then a full set of distances can be used to compute the protein structure. Unfortunately, such estimation is usually costly and prone to introduce errors since some "missing" distances, especially those long distances between atoms spatially away from each other, may be very hard to derive from available distances. For example, the EMBED algorithm uses a technique called 'Bound Smoothing', in general with the computational cost of $O(n^4)$, to estimate "missing distances" (Crippen and Havel, 1988). Bound smoothing relies on geometric rules such as triangular inequalities, which at best can only estimate an upper and lower distance bounds. It's hard to derive the 'exact' distance from the bounds. Then in order to find all the "missing distances", the derived 'error' distances are likely used to estimate other missing distances, propagating errors and making the estimation unreliable. A better approach would be directly solving the structure by using the sparse distances. In that case, instead of estimating every "missing" distance, we can just use reliable sparse distances obtained either from NMR experiments and our knowledge on certain bond length and bond angles or from some confident estimations. In this paper, we reported a geometric build-up algorithm for protein structure determination using only sparse sets of distances. Our algorithm was based on the same idea in our previously reported linear time algorithm for full-sets of distances. The difference is that the current algorithm uses a set of base atoms to build only a partial or local structure of protein. The base is changed if the distance data does not suffice to fix all the atoms. We implemented and tested our algorithm for proteins with sparse sets of distances and obtained the protein structures correctly. An example is given in the report, for which only distances shorter than 10 Åare assumed available. The total number of available distances in this particular test case is only about 3% of all distances in the molecule, but they were enough to be used to determine the protein structure correctly as demonstrated in our results. In our algorithm, the coordinates of unfixed atoms are determined by four distances from the unfixed atoms to their four neighboring fixed atoms given the condition that the four neighbors form a good non-planar geometric shape. Once an unfixed atom is fixed, the distances from this atom to the rest of its neighbors may not be used any more. Therefore, in practice, not all given distances will be used in determining a structure. To determine the minimum number of distances required to solve a structure is probably not very meaningful since the distribution of the distances is more important than the total number of available distances. Consider the example we have given in the report. It is possible that some unfixed atoms may not have enough neighbors while others have more than enough. Then even though the total number of available distances remains the same, the protein structure will not be solvable.

One problem we have encountered in developing our algorithm for sparse sets of distances was the error accumulation in the geometric build-up procedure. For example, consider the structure shown in Figure 2. If the calculated coordinates for atoms 5 and 6 had some small errors, the coordinates for atom 7 may get the errors accumulated since they depend on atoms 5 and 6. So on and so forth, the errors will get aggregated and the accuracy of the structure will be lost. In fact, we have seen in our test cases that small proteins with several hundreds of atoms were not affected much by such errors, but big proteins with more than one thousand atoms were affected severely. This problem can be resolved by choosing a proper system of linear equations for fixing the coordinates and by using a stable numerical method for solving the system, as we have discussed in the report. However, a complete analysis on these issues may need to be done in order to make the algorithm more robust for all practical applications. Overall, we have developed a new algorithm for the molecular distance geometry problem with sparse sets of distances. The algorithm does not require estimating all "missing" distances and is built upon simple mathematical and geometrical calculations. It could also be extended to more general and practical classes of molecular distance geometry problems when only lower and upper bounds on the distances are given. In those cases, the coordinates for each of the atoms can be determined as a set of intervals that specify a region for the location of the atom. Mathematically, this can be achieved by solving a system of interval equations. Work along this direction is being underway and will be reported elsewhere.

References

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000), The protein data bank, *Nucleic Acids Res.* 28: 235–242.

Blumenthal, L.M. (1953), *Theory and Applications of Distance Geometry*, Clarendon Press, Oxford.

Branden, C. and Tooze, J. (1991), *Introduction to Protein Structure*, Garland Publishing, Inc.

Crippen, G.M. and Havel, T.F. (1988), *Distance Geometry and Molecular Conformation*, John Wiley & Sons, New York.

Creighton, T.E. (1993), *Proteins: Structures and Molecular Properties*, W.H. Freeman and Company, New York.

Dong, Q. and Wu, Z. (2002), A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances, *Journal of Global Optimization* 22: 365–375.

Golub, G.H. and Van Loan, C.F. (1989), *Matrix Computations*, Johns Hopkins University Press.

Glunt, W., Hayden, T.L. and Raydan, M. (1993), Molecular conformations from distance matrices, *J. Comput Chem* 14: 114–120.

Le Grice, S.F.J., Naas, T., Wohlgensinger, B. and Schatz, O. (1991), Subunit-selective mutagenesis indicates minimal polymerase activity in heterodimer-associated p51 HIV-1 reverse transcriptase, *EMBO J.* 10: 3905–3911.

Havel, T.F. (1995), Distance Geometry. In: *Encyclopedia of Nuclear Magnetic Resonance*, Grant, D.M. and Harris, R.K. (eds.), John Wiley & Sons, pp. 1701–1710.

Hendrickson, B.A. (1991), *The Molecular Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Cornell University.

Huang, H., Liang, Z. and Pardalos, P.M. (2001), *Some Properties on Euclidean Distance Matrix and Positive Semi-Definite Matrix Completion Problems*, Department of Industrial and Systems Engineering, University of Florida, Gainsville, FL.

Moré, J. and Wu, Z. (1996), ϵ-Optimal solutions to distance geometry problems via global continuation. In: *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, Pardalos, P.M., Shalloway, D. and Xie, G. (eds.), American Mathematical Society, pp. 151–168.

Moré, J. and Wu, Z. (1997), Global continuation for distance geometry problems, *SIAM Journal on Optimization* 7: 814–836.

Moré, J. and Wu, Z. (1999), Distance geometry optimization for protein structures, *Journal of Global Optimization* 15: 219–234.

Sayle, R. and Milner-White, E. (1994), RasMol: biomolecular graphics for all, *Trends Biochem. Sci. (TIBS)* 20: 374.

Telesnitsky, A. and Goff, S.P. (1997), Reverse transcriptase and the generation of retroviral DNA. In: *Retroviruses*, Coffin, J., Hughes, S. and Varmus, H. (eds.), Cold Spring Harbor Laboratory Press.