

A Linear-Time Algorithm for Solving the Molecular Distance Geometry Problem with Exact Inter-Atomic Distances

Qunfeng Dong* and Zhijun Wu†

June 1, 2001

Abstract

We describe a linear-time algorithm for solving the molecular distance geometry problem with exact distances between all pairs of atoms. This problem needs to be solved in every iteration of general distance geometry algorithms for protein modeling such as the EM-BED algorithm by Crippen and Havel. However, previous approaches to the problem rely on decomposing an distance matrix or minimizing an error function and require $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ floating point operations. The linear-time algorithm will provide a much more efficient approach to the problem, especially in large-scale applications. It exploits the problem structure and hence is able to identify infeasible data more easily as well.

Keywords. 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,

*Department of Biochemistry and Biophysics and Molecular Biology, Iowa State University, Ames, IA 50010, Email: qfdong@iastate.edu

†Department of Mathematics and Graduate Program on Bioinformatics and Computational Biology, Iowa State University, Ames, IA 50010, Email: zhijun@iastate.edu

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 [2].

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 [5]. 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. The problem is in general called the molecular distance geometry problem.

Modeling protein structure with distance geometry was pioneered by Crippen and Havel, who developed the EMBED algorithm for structure determination with NMR distance data [4]. Several biochemistry groups developed similar methods for NMR structure modeling, such as Kuntz, Thomason, and Oshiro [13], and Brünger and Niles [3]. Work has also been done in developing more efficient and reliable algorithms including the graph reduction algorithm by Hendrickson [10], the alternating-projection algorithm by Glunt, Hayden, and Raydan [7], and the global smoothing and continuation algorithm by Moré and Wu [15, 16, 17].

In this paper, we study a special class of distance geometry problems when exact distances between all pairs of atoms are given. In practice, we may have only a sparse set of distances and know only their lower and upper bounds. However, the missing data can often be estimated or approximated, and a set of exact distances can be generated in between the bounds. A valid structure can then be determined by repeatedly solving a distance geometry problem with exact distances between all pairs of atoms [4, 7, 9]. This problem can be solved in $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ floating point operations by decomposing an n by n distance matrix, where n is the number of atoms. However, it is too costly for large-scale applications, especially when the problem needs to be solved many times. We describe a new algorithm for solving the problem in only $\mathcal{O}(n)$ time. The algorithm is based on a simple geometric relationship between the coordinates of the atoms and the distances among them. 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, and in other words, is in the order of n floating point operations.

The paper is organized as follows. We first describe the distance geometry problem with exact distances in greater detail in Section 2. We also review the matrix decomposition algorithm for the problem. We then present our algorithm in Section 3 and verify the computation time. The implementation of the algorithm is described in Section 4. We also discuss some of our computational results with proteins. In Section 5, we conclude the paper and discuss possible extensions of the algorithm to problems with bounds on the distances and beyond.

2 The Problem with Exact Distances

A special class of distance geometry problems is when exact distances between all pairs of atoms are given. Then, the problem can be solved by decomposing a distance matrix formed with 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 for the atoms in three-dimensional space, the distance matrix must be of rank ≤ 3 . If we can find the nonzero singular values of the matrix, we can use the corresponding singular vectors to find the coordinates of the atoms immediately [1, 4].

If the number of arithmetic operations required for solving a problem is bounded by a polynomial function of the problem size, we say that the problem can be solved in polynomial time and it is a tractable problem. Note that computing the singular values of an $n \times n$ matrix can be done in at most $\mathcal{O}(n^3)$ floating point operations [6]. So, the distance geometry problem, when all exact distances are given, can be solved in polynomial time and is a tractable problem. We now describe the matrix decomposition algorithm for this special class of distance geometry problems in greater detail in the following.

If all exact distances are given, they can be arranged into a matrix, $d = [d_{i,j}]$, with $d_{i,j}$ corresponds to the distance between atoms i and j . Suppose that we have a set of coordinates x_0, x_1, \dots, x_n , where $x_i = (u_i, v_i, w_i)^T$ specifies the coordinates for atom i . Without loss of generality, we assume that the coordinates are translated so that $x_0 = (0, 0, 0)^T$. We can make this assumption since no matter what the coordinates are, we can always translate them without changing any distances among the atoms.

We consider the problem to find x_1, \dots, x_n so that the distances between points i and j are equal to given distances $d_{i,j}$ for all i and j . The distance

constraints can be written in the following mathematical form,

$$\|x_i - x_j\| = d_{i,j}, \quad i, j = 0, 1, \dots, n,$$

or equivalently,

$$\begin{aligned} \|x_i\|^2 &= d_{i,0}^2, \\ \|x_i - x_j\|^2 &= d_{i,j}^2, \quad i, j = 1, \dots, n. \end{aligned}$$

The second set of constraints are equivalent to

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i, j = 1, \dots, n.$$

We then obtain

$$d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2 = 2x_i^T x_j, \quad i, j = 1, \dots, n.$$

Let $D_{i,j} = (d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2)/2$. We can then define a matrix $D = [D_{i,j}]$. Let X be an $n \times 3$ matrix and $X = [x_1^T; \dots; x_n^T]$. We then have

$$D = XX^T.$$

If a solution exists for this equation, matrix D must be of rank ≤ 3 . Therefore, we can make a singular value decomposition for D to obtain

$$D = U\Sigma U^T,$$

where U is an $n \times 3$ orthogonal matrix and Σ a 3×3 diagonal matrix with the diagonal elements σ_1 , σ_2 , and σ_3 being three largest singular values of D . A solution for $D = XX^T$ can then be obtained with

$$X = U\Sigma^{1/2}.$$

Note that the singular value decomposition can be done in at most $\mathcal{O}(n^3)$ floating point operations. Therefore, the solution to the distance geometry problem can be obtained in polynomial time if given all exact distances between pairs of atoms.

3 A Linear-Time Algorithm

Our algorithm is based on a simple geometric relationship between distances and coordinates. In two dimensional space, if we know the distances among three atoms, we can find the coordinates for the atoms by solving a simple algebraic equation. If the three atoms are not in the same line, the coordinates for any of the remaining atoms can then be determined uniquely with its distances to the three fixed atoms (see Figure 1).

Similarly, in three dimensional space, if we know the distances among four atoms, we can find the coordinates for the atoms immediately. If the four atoms are not in the same plane, the coordinates for any of the remaining atoms can then be determined uniquely with its distances to the fixed four atoms.

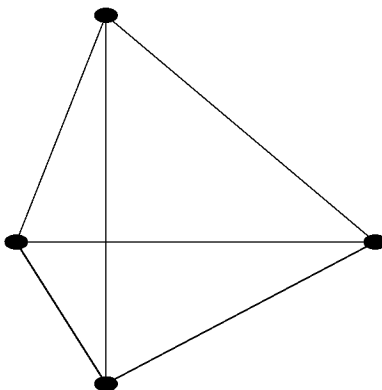


Figure 1: A 2D example: The fourth atom on the top can be determined with its distances to the other three atoms.

Mathematically, let us assume that we have found the coordinates for the first four atoms. Let the coordinates be denoted by

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

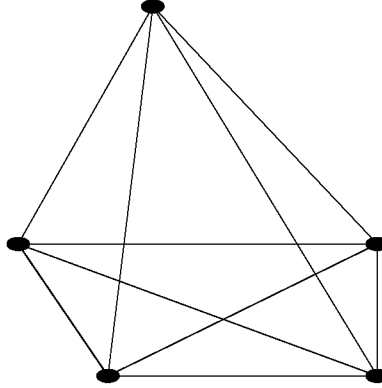


Figure 2: A 3D example: The fifth atom on the top can be determined with its distances to the other four atoms.

$$x_4 = (u_4, v_4, w_4)^T.$$

Suppose that we want to determine the coordinates $x_i = (u_i, v_i, w_i)^T$ for a certain atom i . Since we know the distances between all pairs of atoms, we certainly know the distances between atoms i and j for $j = 1, 2, 3, 4$. Let the distances be denoted by $d_{i,j}$. 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 is 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

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

Subtracting the first equation from the rest ones, we obtain

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

In matrix form, the equations are reduced 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

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

This system of equations can be solved easily in a fixed number of floating point operations. Therefore, if we have n atoms in a molecule, we only need to solve the equations for at most n times. The total number of floating point operations is $\mathcal{O}(n)$. We then obtain a linear time algorithm for determining the structure of the molecule given exact distances between all pairs of atoms.

4 Computational Results

We have implemented our algorithm and tested it with a set of protein structures. We describe our implementation procedure, the particular protein we studied, and some of our preliminary results.

We implemented the algorithm in C++. The code includes two major objects, molecule and atom, implemented as C++ classes. The molecule class is built with the atom class. It consists of an array of atoms. Each atom has a set of data including its coordinates and the distances to other atoms in the molecule. Several member functions are implemented for molecule. They include input, output, and graphics display functions, and a function for structure determination. The input function takes distance data from an input file and determines the molecule and its atoms. The structure determination function computes the coordinates of the atoms based on the input distances. The results are saved in a file in certain format and can be displayed by the graphics function, which is implemented by integrating RasMol [18] into our program.

The structure determination function determines the coordinates for the atoms using the linear time distance geometry algorithm. More specifically, it takes the first atom and puts it at the origin. Let u_1 , v_1 , and w_1 be the three coordinates for the atom. Then $u_1 = 0$, $v_1 = 0$, and $w_1 = 0$. It then fixes the second atom on one of the axes, say 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 atom then is selected among the remaining atoms if it cannot be on the same line determined by the first two atoms according to its distances to the two atoms. The atom is put into one of the planes formed by the axes, say 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:

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

and therefore,

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

Here, v_3 can either be positive or negative without affecting the final structure. We therefore choose v_3 to be positive. Finally, the fourth atom is selected if it cannot be put in the same plane formed by the first three atoms according to its distances to the three atoms. The atom can then be fixed by solving the following equations.

$$u_4^2 + v_4^2 + w_4^2 = d_{4,1}^2$$

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

and

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

Once all first four atoms are allocated, the remaining atoms can be determined by using exactly the algorithm described in the previous section. Here again, w_4 can either be positive or negative, corresponding to two mirror symmetric structures. We compute one of the structures with w_4 positive. The second one can be obtained by simply making all w_i , $i \geq 4$, to have an opposite sign.

We tested our program with a set of proteins. In particular, we worked on a HIV-1 RT protein. We describe this protein briefly in the following. Human immunodeficiency virus type 1 (HIV-1) is known to be the etiologic agent of the acquired immunodeficiency syndrome (AIDS). Extensive studies have been carried out toward understanding how the virus replicates and integrates into host genome. The reverse transcriptase (RT) of HIV-1 is responsible for converting the viral genome RNA into DNA, which is the key step for viral replication. Similar to other retrovirus RT, HIV-1 RT contains three distinct enzymatic activities: (i) RNA dependent DNA polymerase activity which uses single-strand RNA as template to synthesize minus-strand DNA; (ii) RNase H activity which digests the RNA strand in the synthesized DNA-RNA duplex and free the minus-strand DNA; (iii) DNA dependent DNA polymerase activity which then uses the minus strand DNA as template to synthesize plus-strand DNA [11, 12, 19]. The crystal structures of HIV-1 RT have been determined by several different groups [19]. HIV-1 RT forms an asymmetric heterodimer of 66- and 51-kDa subunits. The 66-kDa subunit, p66, consists of both the DNA polymerase activities and the RNase H activity while the 51-kDa subunit, p51, also contains some DNA polymerase activities but lacks the RNase H activity. Furthermore, genetic studies demonstrated that only p66 contributes directly to the polymerase activity of the HIV-1 [8]. In order to test our program, we have retrieved the X-ray structural data of the p66 (1HMV.pdb) from the Protein Data Bank (PDB). The structural data deposited in the PDB are the coordinates of each atom in the molecule. We first converted the coordinates to distances

for every pair of atoms and then used the converted distance data as the input to our program.

There are 4,200 atoms in the p66 subunit of HIV-1 RT. We ran our code with the input distance data and were able to determine the structure of the protein in only 188,859 floating point operations. Figure 3 shows the original (left) and computed (right) structures of the protein. The two structures match perfectly, and their distance matrix error (DME) is equal to zero.

We have also implemented an SVD algorithm as described in Section 2 in Matlab [14]. We ran the Matlab program on the same p66 subunit of HIV-1 RT. The program required 1,268,200,000 floating point operations to obtain a structure. Our algorithm was about 6,715 times faster than the SVD algorithm.

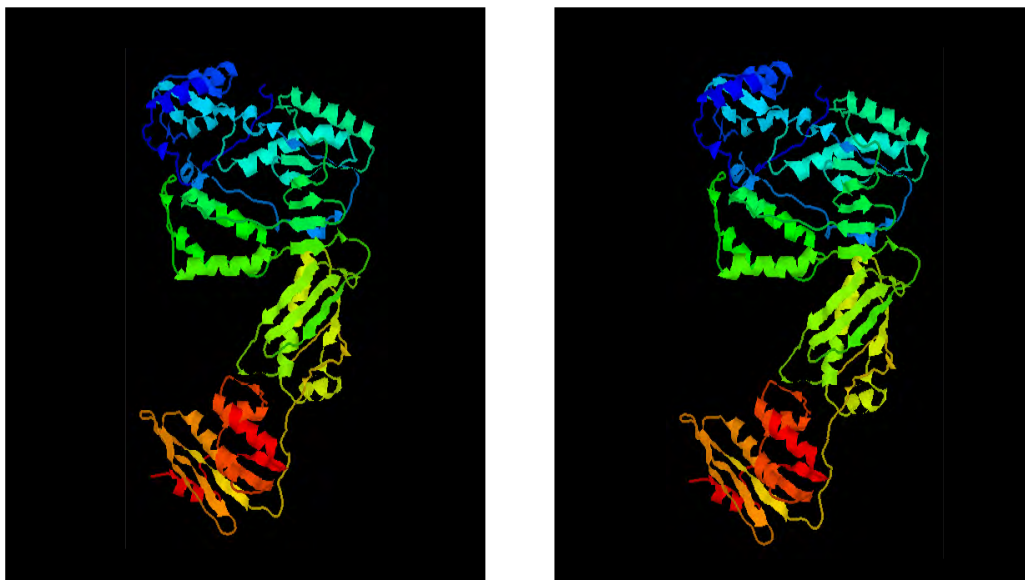


Figure 3: The original (left) and computed (right) structures for the HIV-1 RT p66 protein

5 Concluding Remarks

In this paper, we have studied a special class of distance geometry problems when exact distances between all pairs of atoms are given. In practice, we may have only a sparse set of distances and know only their lower and upper bounds. However, the missing data can often be estimated or approximated, and a set of exact distances can be generated by using the bounds. A valid structure can then be determined by repeatedly solving a distance geometry problem with exact distances between all pairs of atoms [4, 7, 9]. This problem can be solved in $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$ floating point operations by decomposing an n by n distance matrix, where n is the number of atoms. However, it is too costly for large-scale applications, especially when the problem needs to be solved many times. We describe a new algorithm for solving the problem in only $\mathcal{O}(n)$ floating point operations. The algorithm is based on a simple geometric relationship between the coordinates of the atoms and the distances among them. 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, and in other words, is in the order of n floating point operations.

We have implemented our algorithm in C++ and tested it with a set of protein structures, and in particular, a HIV-1 RT protein. The results showed that our algorithm determined the structure efficiently and outperformed the SVD algorithm by several orders of magnitude.

The linear-time algorithm is not only a great improvement over the SVD algorithm for the exact distance geometry problem, but also detects errors in the data more easily when a feasible structure cannot be found: The inconsistent distances can be identified when the system of equations for a particular atom does not have a solution. This property can be as important as determining the structure of the molecule itself in practice.

However, since only a subset of the distances are used by the algorithm, even if a structure is determined, other distances may still be inconsistent. In our implementation of the algorithm, we have a follow-up procedure to detect such possible inconsistency in the distance data: We computed all the distances from the computed structure and compare them with the given distances. The inconsistent distances are reported when any discrepancies are detected. This computation, when needed, requires $\mathcal{O}(n^2)$ floating point calculations.

Finally, the linear-time algorithm can also be extended directly to general

classes of distance geometry problems when 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

- [1] L. M. Blumenthal, *Theory and Applications of Distance Geometry*, Oxford, Clarendon Press, 1953.
- [2] C. Branden and J. Tooze, *Introduction to Protein Structure*, Garland Publishing, Inc., 1991.
- [3] A. T. Brüger and M. Niles, *Computational Challenges for Macromolecular Modeling*, in *Reviews in Computational Chemistry*, K. B. Lipkowitz and D. B. Boyd, eds., VCH Publishers, 1993, Vol. 5, pp. 299–335.
- [4] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, 1988.
- [5] T. E. Creighton, *Proteins: Structures and Molecular Properties*, W. H. Freeman and Company, 1993.
- [6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.
- [7] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular Conformations from Distance Matrices*, *J. Comput Chem*, Vol.14, No.1, pp. 114–120, 1993.
- [8] S. F. J. Le Grice, T. Naas, B. Wohlgensinger, and O. Schatz, *Subunit-Selective Mutagenesis Indicates Minimal Polymerase Activity in Heterodimer-Associated p51 HIV-1 Reverse Transcriptase*, *EMBO J* 10, 3905–3911, 1991.
- [9] T. F. Havel, *Distance Geometry*, in *Encyclopedia of Nuclear Magnetic Resonance*, D. M. Grant and R. K. Harris, eds., John Wiley & Sons, 1995, pp. 1701–1710.
- [10] B. A. Hendrickson, *The Molecular Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Cornell University, 1991.

- [11] A. Jacob-Molina and E. Arnold, *HIV Reverse Transcriptase Function Relationships*, *Biochemistry* 30, 6351–6361, 1991.
- [12] J. Jaeger, T. Restle, and T. A. Steitz, *The Structure of HIV-1 Reverse Transcriptase Complexed with an RNA Pseudoknot Inhibitor*, *EMBO J* 17, 4535–4542, 1998.
- [13] I. D. Kuntz, J. F. Thomason, and C. M. Oshiro, *Distance Geometry*, in *Methods in Enzymology*, N. J. Oppenheimer and T. L. James, eds., vol. 177, Academic Press, 1993, pp. 159–204.
- [14] The MathWorks Inc., *Matlab User's Guide*, 1992.
- [15] J. Moré and Z. Wu, ϵ -*Optimal Solutions to Distance Geometry Problems via Global Continuation*, in *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, P. M. Pardalos, D. Shalloway, and G. Xie, eds., American Mathematical Society, 1996, pp. 151–168.
- [16] J. Moré and Z. Wu, *Global Continuation for Distance Geometry Problems*, in *SIAM Journal on Optimization*, Vol. 7, No. 3, 1997, pp. 814–836.
- [17] J. Moré and Z. Wu, *Distance Geometry Optimization for Protein Structures*, *Journal of Global Optimization*, Vol. 15, 1999, pp. 219–234.
- [18] R. Sayle and E. Milner-White, *RasMol: Biomolecular Graphics for All*, *Trends in Biochemical Sciences (TIBS)*, Vol. 20, No. 9, 1995, pp. 374.
- [19] A. Telesnitsky and S. P. Goff, *Reverse Transcriptase and the Generation of Retroviral DNA*, in *Retroviruses*, J. Coffin, S. Hughes, and H. Varmus, Eds., Cold Spring Harbor Laboratory Press, 1997.