

Linear Algebra in Biomolecular Modeling*

Zhijun Wu

Department of Mathematics

Program on Bioinformatics and Computational Biology

Iowa State University, Ames, Iowa, USA

Email: zhijun@iastate.edu

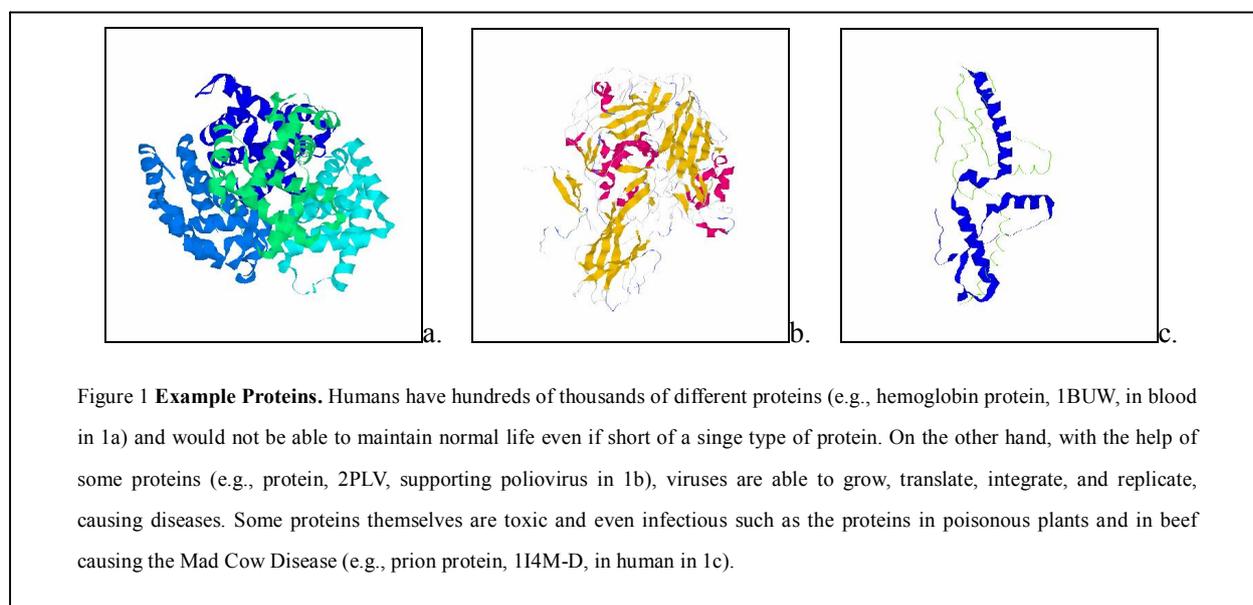
Abstract. Biomolecular modeling is an active research area in computational biology. It studies the structures and functions of biomolecules such as proteins via computer modeling and simulation. As in other types of scientific computing, linear algebra is one of the most powerful mathematical tools for biological computing. Here we review several subjects in biomolecular modeling, where linear algebra has played a major role, including mapping from distances to coordinates in NMR structure determination, solving the Procrustes problem for structural comparison, exploiting the structure of the Karle-Hauptman matrix in protein X-ray crystallography, computing the fast and slow modes of protein motions, and solving the flux balancing equations in metabolic network simulation. The last subject actually involves the modeling of a large biological system, something beyond conventional biomolecular modeling, yet of increased research interests in computational systems biology.

Key words. Biomolecular modeling, metabolic network simulation, matrix computation, Fourier transform and convolution, convex analysis

* To be published in the Handbook of Linear Algebra, edited by Leslie Hogben, Chapman/Hall CRC Press (2006)

1. Introduction Biomolecular modeling is an active research area in computational biology. It studies the structures and functions of biomolecules by using computer modeling and simulation [1]. Proteins are an important class of biomolecules. They are encoded in genes and produced in cells through genetic translation. Proteins are life supporting (or sometimes, destructing) ingredients (Figure 1) and are indispensable for almost all biological processes [2]. In order to understand the diverse biological functions of proteins, the knowledge of the three-dimensional structures of proteins are essential. Several structure determination techniques have been used, including X-ray crystallography, nuclear magnetic resonance spectroscopy (NMR), and homology modeling. They all require intensive mathematical computing, ranging from data analysis to model building [3].

As in all other types of scientific computing, linear algebra is one of the most powerful mathematical tools for biological computing. Here we review several subjects in biomolecular modeling, where linear algebra has played a major role, including mapping from distances to coordinates in NMR structure determination (Section 2), solving the Procrustes problem for structural comparison (Section 3), exploiting the structure of the Karle-Hauptman matrix in protein X-ray crystallography (Section 4), computing the fast and slow modes of protein motions (Section 5), and solving the flux balancing equations in metabolic network simulation (Section 6). The last subject actually involves the modeling of a large biological system, something beyond conventional biomolecular modeling, yet of increased research interests in computational systems biology [4].



2. Mapping from Distances to Coordinates: NMR Protein Structure Determination Let n be the number of atoms in a given protein and x_1, \dots, x_n be the coordinate vectors for the atoms, where $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ and $x_{i,1}, x_{i,2}$, and $x_{i,3}$ are the first, second, and third coordinates of atom i . Then the distance $d_{i,j}$ between atoms i and j can be computed with $d_{i,j} = \|x_i - x_j\|$, where $\|\cdot\|$ is the Euclidean norm. Define the coordinate and distance matrices for the protein by

$$X^\circ = \{x_{i,j} : i = 1, \dots, n, j = 1, 2, 3\} \quad \text{and}$$

$$D^\circ = \{d_{i,j} : i, j = 1, \dots, n\},$$

respectively. Then, if the protein structure and hence X° is known, D° can immediately be computed from X° . Conversely, if D° is known or even partially known, X° can also be obtained from D° , but the

computation is not as straightforward. The latter is known mathematically as the distance geometry problem [5] and proved to be *NP*-complete for general sparse distance matrices [6]. The solution of a distance geometry problem with incomplete and inexact distances has been an important component of NMR protein structure computing, where the coordinates of the atoms in a protein need to be determined by using a set of inter-atomic distances or their ranges obtained mainly from NMR experiments [7]. Here we consider a simple case when a complete set of exact distances is given for all the pairs of atoms in a protein.

Assume that a solution X° does exist for a given D° . Then, $\|x_i - x_j\| = d_{i,j}$ for all $i, j = 1, \dots, n$, and

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

Since the structure is invariant under any translation or rotation, we set a reference system so that the origin is located at the last atom or in other words, $x_n = (0, 0, 0)^T$. It follows that

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

Define a new set of coordinate and distance matrices,

$$X = \{x_{i,j} : i = 1, \dots, n-1, j = 1, 2, 3\} \quad \text{and}$$

$$D = \{(d_{i,n}^2 - d_{i,j}^2 + d_{j,n}^2)/2 : i, j = 1, \dots, n-1\}.$$

Then, $XX^T = D$ and D must be of maximum rank 3 (Theorem 2.1). Solving this equation for X given D would yield a solution X° to the distance geometry problem for distance matrix D° .

Theorem 2.1 The matrix D induced from a distance matrix D° in R^k is of maximum rank k . [5]

The equation $XX^T = D$ can be solved using singular-value decomposition (SVD) [7]. Let $D = U\Sigma U^T$ be the singular-value decomposition of D , where U is an orthogonal matrix and Σ a diagonal matrix with the squares of the singular values of D along the diagonal. If D is a matrix of rank less than or equal to 3, the decomposition can be obtained with U being $(n-1) \times 3$ and Σ being 3×3 . Then, $X = U\Sigma^{1/2}$ solves the equation $XX^T = D$. Here the singular-value decomposition requires at least $O(n^2)$ floating-point operations [8].

Recently, Dong and Wu [9] have developed a more efficient algorithm called the geometric build-up algorithm, which can find a solution for the above problem in $O(n)$ floating-point operations. The algorithm first finds four atoms that cannot be in the same plane and determine the coordinates for the four atoms using, say, the SVD method just described with all the distances among them. Let $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$, $i = 1, 2, 3, 4$, be the coordinate vectors already determined for the four atoms. Then, the coordinates $x_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$ for any remaining atom $j = 5, \dots, n$ can be determined by using the distances $d_{i,j}$ from atoms $i = 1, 2, 3, 4$ to atom j . Indeed, x_j can be obtained from the solution of the following system of equations,

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2, \quad i = 1, 2, 3, 4,$$

where x_i and $d_{i,j}$, $i = 1, 2, 3, 4$ are known. By subtracting equation i from equation $i+1$ for $i = 1, 2, 3$, we can eliminate the quadratic terms for x_j to obtain

$$-2(x_{i+1} - x_i)^T x_j = (d_{i+1,j}^2 - d_{i,j}^2) - (\|x_{i+1}\|^2 - \|x_i\|^2), \quad i = 1, 2, 3.$$

Let A be a matrix and b a vector, and

$$A = -2 \begin{bmatrix} (x_2 - x_1)^T \\ (x_3 - x_2)^T \\ (x_4 - x_3)^T \end{bmatrix},$$

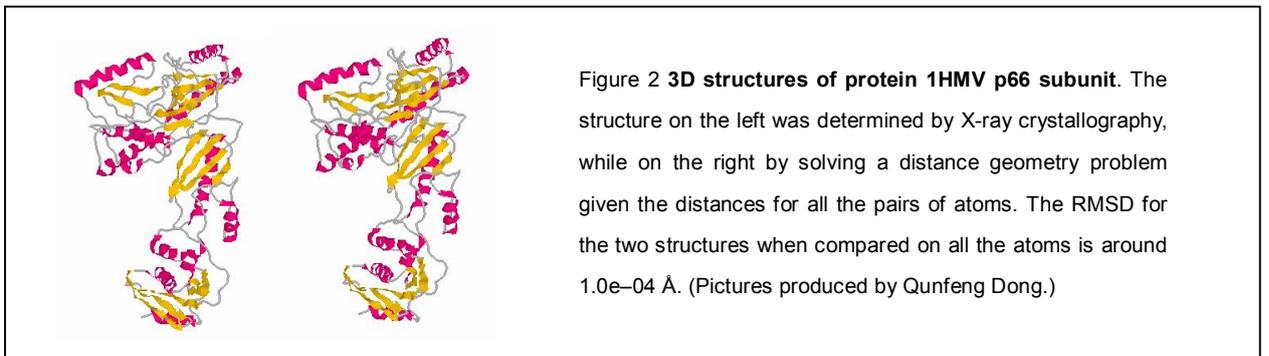
$$b = \begin{bmatrix} (d_{2,j}^2 - d_{1,j}^2) - (\|x_2\|^2 - \|x_1\|^2) \\ (d_{3,j}^2 - d_{2,j}^2) - (\|x_3\|^2 - \|x_2\|^2) \\ (d_{4,j}^2 - d_{3,j}^2) - (\|x_4\|^2 - \|x_3\|^2) \end{bmatrix}.$$

We then have $Ax_j = b$. Since x_1, x_2, x_3, x_4 are not in the same plane, A must be nonsingular and we can therefore solve the linear system to obtain a unique solution for x_j . Here, solving the linear system requires only constant time. Since we only need to solve $n-4$ such systems for $n-4$ coordinate vectors x_j , the total computation time is proportional to n (see [9] for more details).

The advantage of using the geometric build-up algorithm is that it is not only more efficient than the SVD method, but also requires a smaller number of distances and is easier to extend to the problems with sparse sets of distances [10]. The SVD method, on the other hand, requires all the distances, but it can be used to obtain a better approximate solution to the problem if the distances contain some errors or are not consistent [11]. The solution from the geometric build-up algorithm for such a case may or may not be a good approximation, depending on the choice of the initial four atoms and hence the distances it uses to build the structure.

Theorem 2.2 Let $D = U\Sigma U^T$ be the singular-value decomposition of D . Let $V = U(:,1:3)$ and $A = \Sigma(1:3,1:3)$. Then, $X = VA^{1/2}$ minimizes $\|XX^T - D\|_F$, where $\|\cdot\|_F$ is the matrix Frobenius norm. [11]

Figure 2 shows two 3D structures of the p66 subunit of the HIV-1 retrotranscriptase (1HMV), one determined experimentally by X-ray crystallography [12] and another computationally by solving a distance geometry problem given the distances for all the pairs of atoms in the protein [9]. The RMSD (see description in Section 3) for the two structures when compared on all the atoms is around $1.0e-04$ Å, showing that the two structures are almost identical. The subunit of the protein has 4,200 atoms. The geometric build-up algorithm took only 188,859 floating-point operations to build the structure, while an SVD method implemented using Matlab required 1,268,200,000 floating-point operations [9]. The geometric build-up algorithm was 6,715 times faster.



3. The Procrustes Problem for Protein Structure Comparison Let X and Y be two $n \times 3$ coordinate matrices for two lists of atoms in proteins A and B , respectively, where $x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T$ is the coordinate vector of the i th atom selected from protein A to be compared with $y_i = (y_{i,1}, y_{i,2}, y_{i,3})^T$, the coordinate vector of the i th atom selected from protein B . Assume that X and Y have been translated so that their centers of

geometry are located at the same position, say, at the origin. Then, the structural similarity between the two proteins can be measured by using the root-mean-square deviation (RMSD) of the structures,

$$\text{RMSD}(X, Y) = \min_Q \|X - YQ\|_F / \sqrt{n},$$

where Q is a 3×3 rotation matrix and $QQ^T = I$, and $\|\cdot\|_F$ is the matrix Frobenius norm. The RMSD is basically the smallest average coordinate errors of the structures for all possible rotations Q of structure Y to fit structure X . It is called the Procrustes Problem for its analogy to the Greek story about cutting a person's legs to fit a fixed sized iron bed [8]. Note that X and Y may be the coordinate matrices for the same ($A = B$) or different ($A \neq B$) proteins and therefore, each pair of corresponding atoms do not have to be of the same type (when $A \neq B$). However, the number of atoms selected to compare must be the same from A and B (# rows of $X = \#$ rows of Y).

RMSD calculation has been widely used in structural computing. A straightforward application is for comparing and validating the structures obtained from different (X-ray crystallography, NMR, or homology modeling) sources for the same protein [13]. Even from the same source, such as NMR, multiple structures are often obtained, and the average RMSD for the pairs of the multiple structures has been calculated as an indicator for the consistency and sometimes the flexibility of the structures [14]. It has also been an important tool for structural classification, motif recognition, and structure prediction, where a large number of different proteins need to be aligned and compared [15].

In any case, the RMSD calculation seems requiring the solution of an optimization problem, as suggested in its definition. The optimization problem is not so trivial to solve if a conventional optimization method is to be used (such as a Newton or steepest descent method). Fortunately, an analytical solution to the problem can actually be obtained with some simple linear algebraic calculations. To see this, we first need a simple fact from standard linear algebra:

Theorem 3.1 Let A and B be two matrices. Suppose that A is similar to B , then $\text{trace}(A) = \text{trace}(B)$. In particular, $\text{trace}(A) = \text{trace}(V^T A V)$, for any orthogonal matrix V .

Now, note that

$$\|X - YQ\|_F^2 = \text{trace}(X^T X) + \text{trace}(Y^T Y) - 2\text{trace}(Q^T Y^T X).$$

Therefore, minimizing $\|X - YQ\|_F$ is equivalent to maximizing $\text{trace}(Q^T Y^T X)$. Let $C = Y^T X$ and $C = U\Sigma V^T$ be the singular-value decomposition of C . Then, by Theorem 3.1,

$$\text{trace}(Q^T Y^T X) = \text{trace}(Q^T C) = \text{trace}(Q^T U\Sigma V^T) = \text{trace}(V^T Q^T U\Sigma) \leq \text{trace}(\Sigma),$$

and $Q = UV^T$ maximizes $\text{trace}(Q^T Y^T X)$ [8].

Theorem 3.2 Let $C = Y^T X$ and $C = U\Sigma V^T$ be the singular-value decomposition of C . Then, $Q = UV^T$ minimizes $\|X - YQ\|_F$. [8]

Based on the above analysis, RMSD(X, Y) for any given X and Y can be computed in the following steps.

1. Compute the geometric centers of X and Y :

$$x_c(j) = [\sum_{i=1}^n X(i, j)] / n, \quad j = 1, 2, 3;$$

$$y_c(j) = [\sum_{i=1}^n Y(i, j)] / n, \quad j = 1, 2, 3.$$

2. Translate X and Y to the origin:

$$X = X - e_n x_c^T, \quad Y = Y - e_n y_c^T,$$

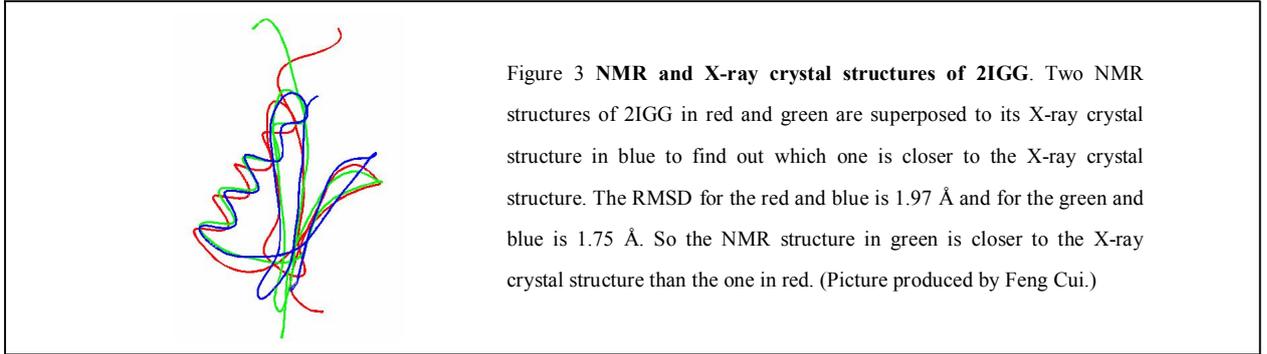
where $e_n = (1, \dots, 1)^T$ in \mathbb{R}^n .

3. Compute $C = Y^T X$ and its singular-value decomposition $C = U \Sigma V^T$. Then,

$$Q = UV^T, \quad \text{and}$$

$$\text{RMSD}(X, Y) = \|X - YQ\|_F / \sqrt{n}.$$

Figure 3 gives an example of using RMSD to compare NMR and X-ray crystal structures. Three structures of the second domain of the immunoglobulin-binding protein (2IGG) [16] are displayed in the figure. The red structure was determined by NMR and was further refined later to become the green one. The blue structure was determined by X-ray crystallography. The RMSD value for the red and blue structures is 1.97 Å, while for the green and blue structures is 1.75 Å, suggesting that the green structure is indeed an improved structure [17].



4. The Karle-Hauptman Matrix in X-ray Crystallographic Computing X-ray crystallography has been a major experimental tool for protein structure determination and is responsible for about 80% of 30,000 protein structures so far determined and deposited in the Protein Data Bank [18]. The structure determination process involves crystallizing the protein, applying X-ray to the protein crystal to obtain X-ray diffractions, and using the diffraction data to deduce the electron density distribution of the crystal (Figure 4). Once the electron density distribution of the crystal is known, a 3D structure for the protein can be assigned [19].

Let $\rho: \mathbb{R}^3 \rightarrow \mathbb{R}$ be the electron density distribution function for the protein and F_H a complex number called a structure factor representing one of the diffraction spots specified by an integral triplet H . The electron density distribution function ρ can be expanded as a Fourier series with the structure factors F_H as the coefficients. In other words, F_H is a Fourier transform of ρ .

$$\rho(r) = \sum_{H \in \mathbb{Z}^3} F_H \exp(-2\pi i H^T r),$$

$$F_H = \int_{\mathbb{R}^3} \rho(r) \exp(2\pi i H^T r) dr.$$

So, if we know F_H for H in a large subset S of \mathbb{Z}^3 , an approximation for ρ can be obtained by setting

$$\rho(r) = \sum_{H \in S \subset \mathbb{Z}^3} F_H \exp(-2\pi i H^T r).$$

Unfortunately, the diffraction data provides only partial information for F_H , i.e., the magnitudes of the structure factors, but not the phases. How to recover the complete information for F_H given only the

magnitudes has thus been a great mathematical challenge known as the phase problem in X-ray crystallography [20].

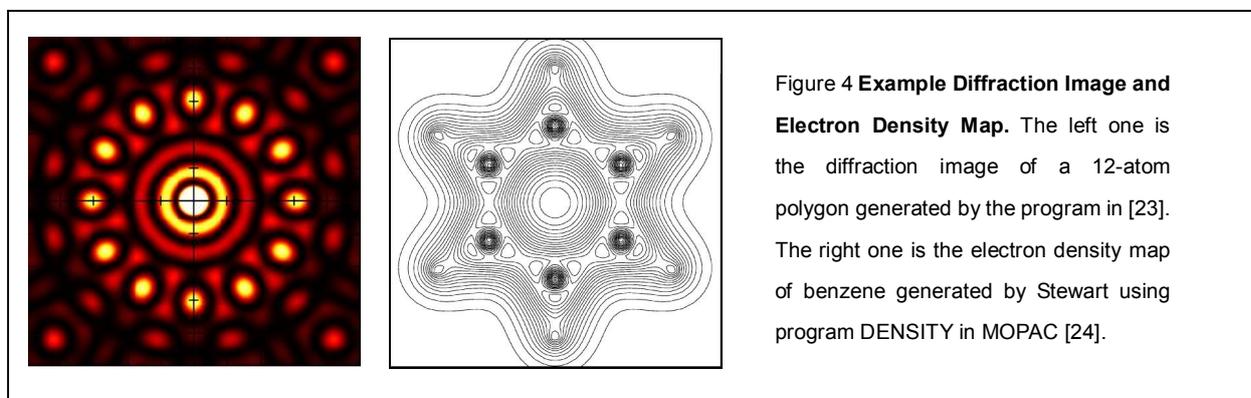
Let H in S be ordered as H_0, H_1, \dots, H_n , where n is usually in the order of 10,000 in practice. Then, a Karle-Hauptman matrix for the structure factors $\{F_H : H = H_0, \dots, H_{n-1}\}$ can be defined as

$$K = \begin{bmatrix} F_{H_0} & F_{H_{n-1}} & \cdots & F_{H_1} \\ F_{H_1} & F_{H_0} & \cdots & F_{H_2} \\ \vdots & \vdots & \ddots & \vdots \\ F_{H_{n-1}} & F_{H_{n-2}} & \cdots & F_{H_0} \end{bmatrix},$$

where

$$F_{H_j} = \int_{R^3} \rho(r) \exp(2\pi i H_j^T r) dr, \quad j = 0, 1, \dots, n-1.$$

This is an important matrix in X-ray crystallography computing, named after two Nobel Laureates, chemist Jerold Karle and mathematician Herbert Hauptman, who received the Nobel Prize in chemistry in 1985 for their work on solving the phase problem in X-ray crystallography. The Karle-Hauptman matrix is frequently used for computing the covariance of the structure factors [21] or the electron density distribution that maximizes the entropy of a crystal system [22]. Here we consider a linear system formed by a Karle-Hauptman matrix and demonstrate how the structure of the matrix can be exploited to achieve



improved performance in related computing.

Let the system be given in the form $Kx = h$, where K is an $n \times n$ Karle-Hauptman matrix and h an n -dimensional complex vector. If a conventional method such as Gaussian Elimination is used, the solution of the system usually takes $O(n^3)$ floating-point operations, which is expensive if n is larger than 1,000 and if the solution is also required multiple times. However, because of the special structure of the matrix K , its inverse can actually be computed easily in an unconventional way:

Theorem 4.1 If K is a Karle-Hauptman matrix, then the inverse of K is also a Karle-Hauptman matrix and can be formed directly as

$$K^{-1} = \begin{bmatrix} E_{H_0} & E_{H_{n-1}} & \cdots & E_{H_1} \\ E_{H_1} & E_{H_0} & \cdots & E_{H_2} \\ \vdots & \vdots & \ddots & \vdots \\ E_{H_{n-1}} & E_{H_{n-2}} & \cdots & E_{H_0} \end{bmatrix},$$

where

$$E_{H_j} = \int_{R^3} \rho^{-1}(r) \exp(2\pi i H_j^T r) dr, \quad j = 0, 1, \dots, n-1. \quad [25]$$

Note that since each element in the inverse matrix can be obtained by doing a Fourier transform for the inverse of ρ and only n distinct elements in the first column are required to form the whole matrix, the calculations can be done in $O(n \log n)$ floating-point operations by using the Fast Fourier Transform [26].

Once the inverse of K is obtained, the solution of the system can be formed with $x = K^{-1}h$. However, the matrix vector product still requires $O(n^2)$ floating-point operations. Note that K^{-1} as well as K has only n distinct elements listed repeatedly in the columns of the matrix with each column having the elements in the previous column circulated by one element from top to the bottom and then bottom to the top. This type of matrix is called the circulant matrix. Let A be an n -dimensional vector and $C(A)$ an $n \times n$ circulant matrix formed with the elements in A . Let B be another n -dimensional vector. Then $C(A)B$ is called the discrete convolution of A and B [27]. With this definition, $K^{-1}h$ is actually a discrete convolution.

Theorem 4.2 Let A, B, a, b be n -dimensional vectors. Suppose that A is the Fourier transform of a , and B is the Fourier transform of b . Then, the discrete convolution $C(A)B$ is equal to the Fourier transform of $a \cdot b = (a_1 b_1, \dots, a_n b_n)^T$. [27]

Based on the above theorem, if h is the Fourier transform of t , then $K^{-1}h$ can be computed by doing a Fourier transform for $\rho^{-1} \cdot t$, where t can be obtained through an inverse Fourier transform for h . The whole computation again takes at most $O(n \log n)$ floating-point operations.

In summary, because of the special structure of the Karle-Hauptman matrix, many related calculations can be carried out much more efficiently than conventional methods, as demonstrated above. Some direct applications of the above calculations can also be found in recent work by Wu, et al. [28] on the development of a fast Newton method for entropy maximization in phase determination.

5. Calculation of Fast and Slow Modes of Protein Motions In a reduced model for protein, a residue is represented by a point, in many cases, the position of C_α or C_β in the residue, and a protein is considered as a sequence of such points connected with strings [29]. If the reduced model of a protein is known, a so-called contact map can be constructed to show how the residues in the protein interact with each other. The map is represented by a matrix with its i, j -entry equal to -1 if residues i and j are within, say 7\AA distance, and 0 otherwise. The contact matrix can be used to compare different proteins. Similar contact patterns often imply structural or functional similarities between proteins [30].

The interaction among the residues can be described by an energy function. When a protein reaches its equilibrium state, the residues in contact can be considered as a set of masses connected with springs. A simple energy function can then be defined using the contact matrix of the protein. Suppose that a protein has n residues with n coordinate vectors x_1, \dots, x_n . Let Γ be the contact matrix for the protein in its equilibrium state.

$$\Gamma_{i,j} = \begin{cases} -1, & \|x_i - x_j\| \leq 7 \text{\AA} \\ 0, & \text{otherwise} \end{cases} \quad i \neq j = 1, \dots, n \quad [30]$$

$$\Gamma_{i,i} = -\sum_{j=1}^n \Gamma_{i,j} \quad i = 1, \dots, n$$

Then, a potential energy function E for the system can be defined such that for any vector $\Delta x = (\Delta x_1, \dots, \Delta x_n)^T$ of the displacements of the residues from their equilibrium positions,

$$E(\Delta x) = \frac{1}{2} k \Delta x^T \Gamma \Delta x,$$

where k is a spring constant. According to statistical physics, the probability of the system having a displacement Δx at temperature T should then be subject to the Boltzmann distribution,

$$p_T(\Delta x) = \frac{1}{Z} \exp(-E(\Delta x)/k_B T) = \frac{1}{Z} \exp(-k \Delta x^T \Gamma \Delta x / 2k_B T),$$

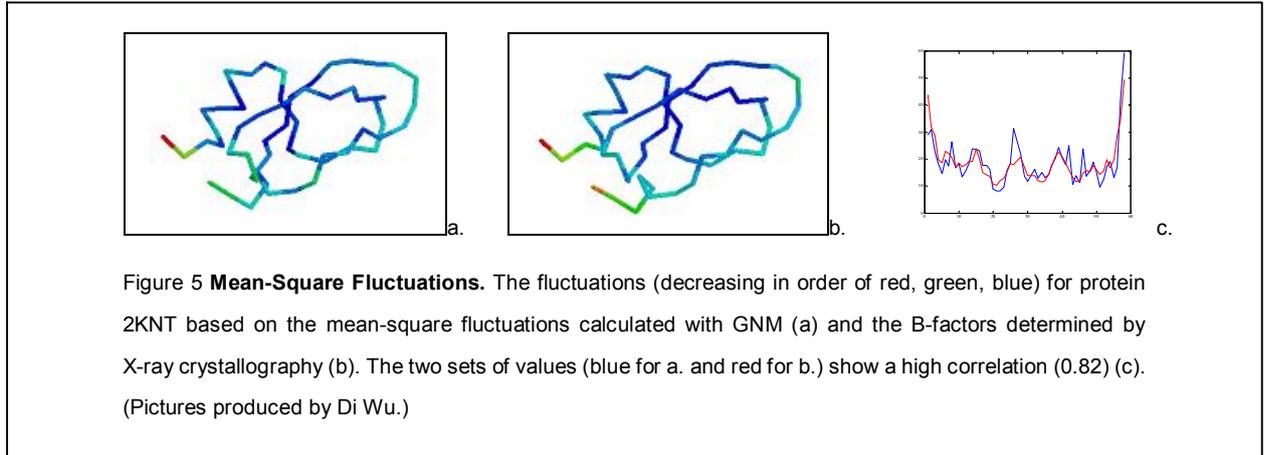
where Z is the normalization factor and k_B the Boltzmann constant.

The above model, called the Gaussian Network Model [31], can be used to find how the residues in the protein move around their equilibrium positions dynamically and in particular, to estimate the so-called mean-square fluctuation for each of the residues $\langle \Delta x_i, \Delta x_i \rangle$, $i = 1, \dots, n$. If the mean-square fluctuation is large, the residue is called hot, and otherwise, is cold, which often correlates with the experimentally detected average atomic fluctuation such as the B-factor in X-ray crystallography [19] and the order parameter in NMR [32]. By using the simple Gaussian Network Model, the mean-square residue fluctuations can actually be estimated with only some simple linear algebra calculations (instead of numerical simulation), due to the following theorem.

Theorem 5.1 Let x_1, \dots, x_n be the equilibrium positions of the residues in a protein, Γ the corresponding contact matrix, and $\Delta x_1, \dots, \Delta x_n$ the corresponding residue fluctuations. Let the singular-value decomposition of Γ be given as $\Gamma = U \Lambda U^T$. Then,

$$\langle \Delta x_i, \Delta x_i \rangle = \frac{1}{Z} \int_{R^{3n}} \Delta x_i^T \Delta x_i \exp(-E(\Delta x)/k_B T) d\Delta x = \sum_{j=1}^n k_B T U_{i,j} \Lambda_{j,j}^{-1} U_{i,j} / k. \quad [31]$$

Figure 5 shows the mean-square fluctuations calculated using the Gaussian Network Model for the structure of protein 2KNT and the comparison with the B-factors of the structure determined by X-ray crystallography. The two sets of values appear to be highly correlated. Based on Theorem 5.1, the calculation of the mean-squares fluctuations requires only a singular-value decomposition of the contact matrix for the protein.



The Gaussian Network Model provides in some sense a coarse-level model for the conventional normal mode analysis (NMA) [33], which is a method to find the fast and slow modes of the protein motion at an equilibrium state by using the singular values of the Hessian matrix of the energy function. Usually, the dynamic behavior of a protein of n atoms can be described by a system of equations as given in the following form (with the masses of the atoms scaled to the unit values),

$$\ddot{x} = -\nabla E(x),$$

where $x = \{x_i = (x_{i,1}, x_{i,2}, x_{i,3})^T : i = 1, \dots, n\}$ is the set of coordinate vectors of the atoms and E the potential energy function of the system. At the equilibrium state x_0 , the energy function can be approximated by a second order Taylor expansion:

$$E(x) \approx E(x_0) + \Delta x^T H \Delta x / 2, \quad H = \nabla^2 E(x_0),$$

where $\Delta x = x - x_0$. It follows that

$$\Delta \ddot{x} = -H \Delta x.$$

Let the singular-value decomposition of the Hessian $H = UAU^T$. Then the solution of the above system can immediately be obtained:

$$\Delta x_i(t) = \sum_{j=1}^n U_{i,j} \alpha_j \cos(\omega_j t + \beta_j), \quad \omega_j = A_{j,j},$$

where α_j and β_j can be determined by the system conditions and in particular

$$\alpha_j^2 = 2k_B T A_{j,j}^{-1},$$

and ω_j is called the j th mode of the protein motion [33]. The larger value the ω_j has, the faster the corresponding mode is. They are all determined by the singular values of H .

From the above solution, the mean-squares fluctuation for each atom i can also be calculated,

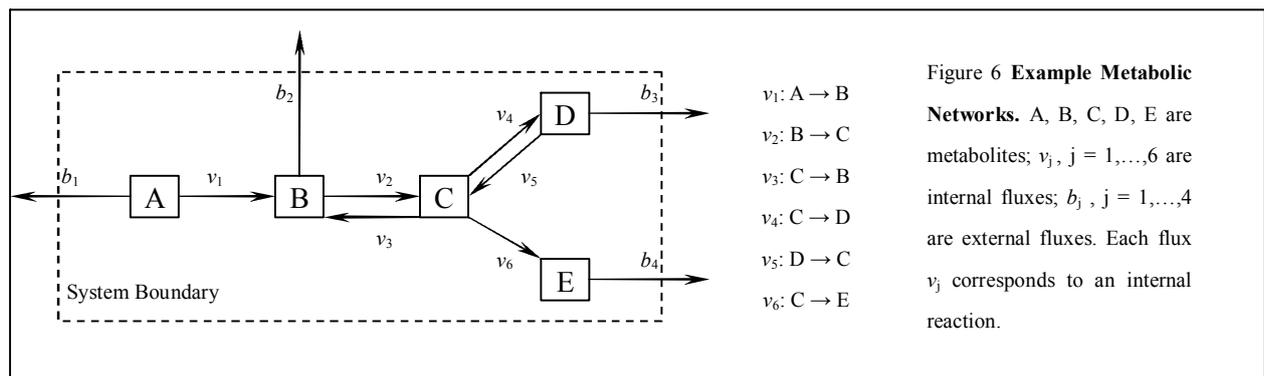
$$\langle \Delta x_i, \Delta x_i \rangle = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau \Delta x_i^2(t) dt = \frac{1}{2} \sum_{j=1}^n U_{i,j} \alpha_j^2 U_{i,j} = \sum_{j=1}^n k_B T U_{i,j} A_{j,j}^{-1} U_{i,j},$$

with $i = 1, \dots, n$ [33]. If a reduced model is used, the formula still apply but yield the fluctuations for the residues, and they are actually equivalent to those derived from the Gaussian Network Model, if the same energy function is used:

Theorem 5.2 The Gaussian Network Model is equivalent to the Normal Mode Analysis for predicting the mean-squares residue fluctuations of a protein, with the energy function defined for the residues instead of the atoms and

$$E(x) = E(x_0) + \frac{1}{2} k \Delta x^T \Gamma \Delta x.$$

6. Flux Balancing Equation in Metabolic Network Simulation A metabolic system is maintained through constant reactions or interactions among a large number of biological and chemical compounds called metabolites [34]. The reaction network describes the structure of a metabolic system and is key to the study of the metabolic function of the system. Figure 6 shows the reaction network for an example metabolic system of five metabolites given in [35].



Each metabolite has a concentration, which changes constantly. The rate of the change is proportional to the amount of the metabolite consumed or produced in all the reactions. Let C_i be the concentration of metabolite i . Let v_j be the chemical flux in reaction j , i.e., the amount of metabolites produced in reaction j per mole. Then,

$$\frac{dC_i}{dt} = \sum_{j=1}^n s_{i,j} v_j,$$

where $s_{i,j}$ is the stoichiometric coefficient of metabolite i in reaction j , and $s_{i,j} = \pm k$, if $\pm k$ moles of metabolite i are produced (or consumed) in reaction j [36]. Let $C = (C_1, \dots, C_m)^T$ be a vector of concentrations of m metabolites, and $v = (v_1, \dots, v_n)^T$ a vector of fluxes of n reactions. Then,

$$\frac{dC}{dt} = Sv,$$

where $S = \{s_{ij} : i = 1, \dots, m, j = 1, \dots, n\}$ is called the stoichiometry matrix [36]. For example, for the system in Figure 6, if both internal and external fluxes are considered, S is a 5×10 matrix and

$$S = \begin{array}{cccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & b_1 & b_2 & b_3 & b_4 \\ \left[\begin{array}{cccccccccc} -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array} \right] \begin{array}{l} \leftarrow A \\ \leftarrow B \\ \leftarrow C \\ \leftarrow D \\ \leftarrow E \end{array} \end{array}$$

Note that the fluxes are functions of the concentrations and some other system parameters. Therefore, the above reaction equations are nonlinear equations of C . However, when the system reaches its equilibrium, $dC/dt = Sv = 0$, when the vector v becomes constant and is called a solution of the steady-state flux equation $Sv = 0$ [36]. The steady-state fluxes are important quantities for characterizing metabolic networks. They can be obtained by solving the steady-state flux equation $Sv = 0$. However, since the number of reactions is usually larger than the number of metabolites, the solution to the equation is not unique. Also, because the internal fluxes are nonnegative, the solution set forms a convex cone, called the steady-state flux cone. Usually, a convex cone can be defined in terms of a set of extreme rays such that any vector in the cone can be expressed as a nonnegative linear combination of the extreme rays,

$$\text{cone}(S) = \{v = \sum_{i=1}^l w_i p_i, \quad w_i \geq 0\},$$

where $p = (p_1, \dots, p_l)^T$ is a set of extreme rays [37]. An extreme ray is a vector that cannot be expressed as a nonnegative linear combination of any other vectors in the cone. A set of vectors is said to be systematically independent if none of them can be expressed as a nonnegative linear combination of others [35]. Since the extreme rays can be used to express all the vectors in a convex cone, they are also called the generating vectors of the cone.

Theorem 6.1 A convex cone has a set of systematically independent generating vectors. They are also unique up to positive scalar multiplications. [35]

Based on this theorem, if the extreme rays of the convex flux cone of a metabolic network are found, all the solutions for the steady-state flux equation can be generated by using the extreme rays. In other words, the extreme rays provide a unique description for the solution space of the steady-state flux equation, and can be used to characterize the whole steady-state capacity of the system.

It is not trivial to find all the extreme rays of a convex cone in general. Several algorithms have been developed specifically for finding the extreme rays of the convex flux cones of metabolic networks [38]. A vector in such a cone corresponds to a metabolic pathway (formed by the nonzero fluxes). An extreme ray therefore corresponds to a so-called extreme pathway of the system. A regular pathway can be obtained by making a nonnegative linear combination of the extreme pathways.

Once the extreme pathways are found, many system properties can be analyzed and some can even be optimized using certain optimization methods [35]. Here we show how specific system properties such as pathway lengths and reaction participations can be obtained from the extreme pathways with some very simple calculations.

Theorem 6.2 Let P be a matrix with each column corresponding to an extreme pathway of a given metabolic network. Let Q be the binary form of P such that $Q_{i,j} = 1$ if $P_{i,j} \neq 0$ and $Q_{i,j} = 0$ otherwise. Then, the diagonal elements of $Q^T Q$ are equal to the lengths of the extreme pathways, while the diagonal elements of $Q Q^T$ show the numbers of extreme pathways the reactions participate in. [39]

Consider the example network in Figure 6 and let S be the stoichiometry matrix including the columns for the internal (v_1, \dots, v_6) as well as external (b_1, \dots, b_4) fluxes. Then, by using an appropriate algorithm (such as the one given in [37]), a matrix of 7 extreme pathways of the system can be found as follows,

$$P^T = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & b_1 & b_2 & b_3 & b_4 \\ \left[\begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 1 \end{array} \right] & \leftarrow p_1 \\ & & & & & & & & & & \leftarrow p_2 \\ & & & & & & & & & & \leftarrow p_3 \\ & & & & & & & & & & \leftarrow p_4 \\ & & & & & & & & & & \leftarrow p_5 \\ & & & & & & & & & & \leftarrow p_6 \\ & & & & & & & & & & \leftarrow p_7 \end{matrix},$$

where row i corresponds to extreme pathway p_i , $i = 1, \dots, 7$. By forming the binary form Q for P and computing

$$Q^T Q = \begin{bmatrix} 3 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 4 & 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 4 & 1 & 0 & 2 \\ 1 & 1 & 2 & 1 & 4 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 & 2 & 1 & 4 \end{bmatrix},$$

and

$$QQ^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 3 & 1 & 1 & 0 & 1 & 0 & 2 & 1 & 1 \\ 0 & 1 & 2 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 3 & 1 & 0 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 & 0 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 1 & 1 & 1 & 1 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 2 & 1 & 0 & 2 & 3 & 1 \\ 0 & 1 & 0 & 0 & 1 & 2 & 0 & 1 & 1 & 2 \end{bmatrix},$$

we obtain, from the diagonal elements of the matrices, the lengths of the pathways, $p_1: 3, p_2: 2, p_3: 4, p_4: 4, p_5: 4, p_6: 2, p_7: 4$, and the participations of the reactions in the extreme pathways, $v_1: 1, v_2: 3, v_3: 2, v_4: 2, v_5: 3, v_6: 2, b_1: 1, b_2: 4, b_3: 3, b_4: 2$. The off-diagonal elements of Q^TQ show the numbers of common reactions in different extreme pathways. For example, $(Q^TQ)_{3,4} = 2$ means that p_3 and p_4 share two common reactions, and $(Q^TQ)_{3,5} = 1$ means that p_3 and p_5 has one common reaction. The off-diagonal elements of QQ^T show the numbers of extreme pathways in which different reactions participate. For example, $(QQ^T)_{2,3} = 1$ means that v_2 and v_3 participate in one extreme pathway together, and $(QQ^T)_{2,8} = 2$ means that v_2 and b_2 participate in two extreme pathways together.

7. Conclusion In this paper, we have reviewed several subjects in biomolecular modeling, where linear algebra has played a central role in related computations. The review has focused on simple showcases and demonstrated important applications of linear algebra in biomolecular modeling. The subjects discussed are of great research interest in computational biology and are related directly to the solutions of many critical but challenging computational problems in biology yet to be solved, including the general distance geometry problem in NMR, the phase problem in X-ray crystallography, the structural comparison problem in comparative modeling, molecular dynamics simulation, and biosystems modeling and optimization, which we have not elaborated in detail in the paper, but the interested readers can further explore.

Linear algebra has also been used to support many basic algebraic calculations required for solving other types of mathematical problems in biomolecular modeling, such as the optimization problems in potential energy minimization [40], the initial value problem in molecular dynamics simulation [41], and the boundary value problem in simulation of protein conformational transformation [42]. They are usually straightforward or routine linear algebra calculations, so we have not covered them in the paper, but they should be considered as equally important as the applications we have discussed.

Acknowledgments

I would like to thank my students, Feng Cui, Ajith Gunaratne, Kriti Mukhopadhyay, Rahul Ravindrudu, Andrew Severin, Matthew Studham, Peter Vedell, Di Wu, Eun-Mee YoonAnn, Rich Wen Zhou, and my colleagues, Dr. Qunfeng Dong, Dr. Wonbin Young, who have read the paper carefully, helped producing some of the pictures, and provided valuable comments and suggestions. The subjects discussed in the paper have actually been the frequent topics of our research meetings. I would also like to thank the Mathematical Biosciences Institute, Ohio State University, for providing support for me to visit the Institute in spring 2005 during which I completed the writing of the paper. This material is based upon work supported by the National Science Foundation under Agreement No. 0112050.

References

1. Schlick, T., *Molecular Modeling and Simulation: An Interdisciplinary Guide*, Springer, 2003.
2. Boyer, R., *Concepts in Biochemistry*, Brooks/Cole Publishing Company, 1999.
3. Creighton, T.E., *Proteins: Structures and Molecular Properties*, Freeman and Company, 1993.
4. Kitano, K., Systems biology: towards system-level understanding of biological systems, in *Proc. 4th Hamamatsu International Symposium on Biology: Computational Biology*, 1999.
5. Blumenthal, L.M., *Theory and Applications of Distance Geometry*, Oxford Clarendon Press, 1953.
6. Saxe, J.B., Embeddability of weighted graphs in K-space is strongly NP-hard, in *Proc. 17th Allerton Conference in Communications, Control and Computing*, 1979, 480-489.
7. Crippen, G.M. and Havel, T.F., *Distance Geometry and Molecular Conformation*, John Wiley & Sons, 1988.
8. Golub, G. and Van Loan, C., *Matrix Computation*, Johns Hopkins University Press, 1989.
9. Dong, Q. and Wu, Z., A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances, *J. Global Optim.* **22**, 365-375, 2002.
10. Dong, Q. and Wu, Z., A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data, *J. Global Optim.* **26**, 321-333, 2003.
11. Havel, T.F., Distance geometry: theory, algorithms, and chemical applications, in *Encyclopedia of Computational Chemistry*, John Wiley & Sons, 1998, 1-20.
12. Rodgers, D.W., Gamblin, S.J., Harris, B.A., Ray, S., Culp, J.S., Hellmig, B., Woolf, D.J., Debouck, C., and Harrison, S.C., The structure of unliganded reverse transcriptase from the human immunodeficiency virus type 1, *Proc. Natl. Acad. Sci. USA* **92**, 1222-1226, 1995.
13. Rhodes, G., *Judging the Quality of Macromolecular Models*, Department of Chemistry, University of Southern Maine, 2000. (<http://www.usm.maine.edu/~rhodes/ModQual/>)
14. Spronk, C.A.E.M., Natuurs, S.B., Bonvin, A.M.J.J., Krieger, E., Vuister, G.W., and Vriend, G., The precision of NMR structure ensembles revisited, *J. Biomolecular NMR* **25**, 225-234, 2003.
15. Eidhammer, I., Jonassen, I., and Taylor, W.R., Structure comparison and structure patterns, *J. Comp. Biol.* **7**, 685-716, 2000.
16. Lian, L.Y., Derrick, J.P., Sutcliffe, M.J., Yang, J.C., and Roberts, G.C.K., Determination of the solution structures of domain II and III of protein G from Streptococcus by ¹H nuclear magnetic resonance, *J. Mol. Biol.* **228**, 1219-1234, 1992.
17. Cui, F., Jernigan, R., and Wu, Z., Refinement of NMR-determined protein structures with database-derived distance constraints, *submitted*, 2004.
18. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, L.N., and Bourne, P.E., The Protein Data Bank, *Nuc. Acid. Res.* **28**, 235-242, 2000.
19. Drenth, J., *Principles of Protein X-ray Crystallography*, Springer-Verlag, 1994.
20. Hauptman, H. and Karle, J., *The Solution of the Phase Problem I: The Centrosymmetric Crystal*, Polycrystal Book Service, 1953.
21. Karle, J. and Hauptman, H., The phases and magnitudes of the structure factors, *Acta Cryst.* **3**, 181-187, 1952.
22. Bricogne, G., Maximum entropy and the foundations of direct methods, *Acta Cryst.* **A40**, 410-445, 1984.
23. Proffen, T.H., Neder, R.B., and Billinge, S.J.L., Teaching diffraction using computer simulations over the Internet, *J. Appl. Cryst.* **34**, 767-770, 2001. (<http://www.med.wayne.edu/biochem/~xray/education.html>)

24. Stewart, J.J.P., *MOPAC Manual*, The Cache Group, Fujitsu America Inc., 2002.
(<http://www.cachesoftware.com/mopac/Mopac2002manual/node383.html>)
25. Bricogne, G. and Gilmore, C.J., A multisolution method of phase determination by combined maximization of entropy and likelihood I. theory, algorithms and strategy, *Acta Cryst.* **A46**, 284-297, 1990.
26. Van Loan, C., *Computational Frameworks for the Fast Fourier Transform*, SIAM, 1992.
27. Tolimieri, R., Myoung, A., and Lu, C., *Algorithms for Discrete Fourier Transform and Convolution*, Springer, 1997.
28. Wu, Z., Phillips, G., Tapia, R., and Zhang, Y., A fast Newton algorithm for entropy maximization in phase determination, *SIAM Review* **43**, 623-642, 2001.
29. Hinds, D.A. and Levitt, M., A lattice model for protein structure prediction at low resolution, *Proc. Natl. Acad. Sci. USA* **89**, 2536-2540, 1992.
30. Miyazawa, S. and Jernigan, R.L., Estimation of effective inter-residue contact energies from protein crystal structures: quasi-chemical approximation, *Macromolecules* **18**, 534-552, 1985.
31. Haliloglu, T., Bahar, I., and Erman, B., Gaussian dynamics of folded proteins, *Phys. Rev. Lett.* **79**, 3090-3093, 1997.
32. Gunther, H., *NMR Spectroscopy: Basic Principles, Concepts, and Applications in Chemistry*, John Wiley & Sons 1995.
33. Levitt, M., Sander, C., and Stern, P.S., Protein normal-mode dynamics: trypsin inhibitor, crambin, ribonuclease, and lysozyme, *J. Mol. Biol.* **181**, 423-447, 1985.
34. Fell, D.A., Systems properties of metabolic networks, in *Proc. International Conference on Complex Systems*, 1997, 21-26.
35. Schilling, C.H., Letscher, D., and Palsson, B., Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective, *J. Theor. Biol.* **203**, 229-248, 2000.
36. Heinrich, R. and Schuster, S., *The Regulation of Cellular Systems*, Chapman and Hall, 1996.
37. Rockafellar, R.T., *Convex Analysis*, Princeton University Press, 1970.
38. Papin, J.A., Stelling, J., Price, N.D., Klamt, S., Schuster, S., and Palsson, B., Comparison of network-based pathway analysis methods, *Trends in Biotechnology* **22**, 400-405, 2004.
39. Papin, J.A., Price, N.D., and Palsson B., Extreme pathway lengths and reaction participation in genome-scale metabolic networks, *Genomic Research* **12**, 1889-1900, 2002.
40. Wales, D.J. and Scheraga, H.A. Global optimization of clusters, crystals, and biomolecules, *Science* **285**, 1368-1372, 1999.
41. Brooks III, C.L., Karplus, M., and Pettitt, B.M., *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics*, John Wiley & Sons, 1988.
42. Elber, R., Meller, J., and Olender, R., Stochastic path approach to compute atomically detailed trajectories: application to the folding of C peptide, *J. Phys. Chem.* **103**, 899-911, 1999.