ELSEVIER

Polymer 43 (2002) 495–501

polymer

www.elsevier.com/locate/polymer

Parameter optimization for the Gaussian model of protein folding ☆

Albert Erkip[a], Burak Erman[a], Chaok Seok[b,*], Ken Dill[b]

[a]*Laboratory of Computational Biology, Faculty of Engineering and Natural Sciences, Sabanci University, Tuzla 81474, Istanbul, Turkey*
[b]*University of California, San Francisco, 3333 California Street, Suite 415, San Fransisco, CA 94118, USA*

This paper is dedicated to Wayne Mattice. His insightful theories and simulations have been a tremendous inspiration to us.

Abstract

Computational models of protein folding and ligand docking are large and complex. Few systematic methods have yet been developed to optimize the parameters in such models. We describe here an iterative parameter optimization strategy that is based on minimizing a structural error measure by descent in parameter space. At the start, we know the 'correct' native structure that we want the model to produce, and an initial set of parameters representing the relative strengths of interactions between the amino acids. The parameters are changed systematically until the model native structure converges as closely as possible to the correct native structure. As a test, we apply this parameter optimization method to the recently developed Gaussian model of protein folding: each amino acid is represented as a bead and all bonds, covalent and noncovalent, are represented by Hooke's law springs. We show that even though the Gaussian model has continuous degrees of freedom, parameters can be chosen to cause its ground state to be identical to that of Go-type lattice models, for which the global ground states are known. Parameters for a more realistic protein model can also be obtained to produce structures close to the real native structures in the protein database.

Keywords: Gaussian model; Protein folding; Parameter optimization

1. Introduction

Computer algorithms that aim to predict the native structure of a protein from its amino acid sequence do not yet have adequate speed or accuracy. And they are usually subject to a type of *irreproducibility*, whereby local kinetic trapping leads to different predicted structures even for the same amino acid sequence, under the same conditions, with the same parameters.

If a folding algorithm had sufficient reproducibility and speed, however, it would be possible to systematically improve its accuracy. Rosen et al. have recently shown that a global minimization method for finding the lowest energy conformations in folding models can serve as a basis for optimizing the parameters of the model [1]. Here, we develop a related approach to finding optimal parameters for folding and docking models, also based on having a reproducible minimizer, but here taken from a beads-and-springs model of protein folding.

One folding model that is fast and has a reproducible minimization strategy is the Gaussian model of protein folding [2]. In the Gaussian model, each amino acid is represented as a single bead and all bonds-covalent or non-covalent-are represented by Hooke's law springs. The parameters in the model are the spring constants. Because all the interactions are simple spring laws, the minimum energy conformation can be computed quickly and reproducibly for any set of spring constants. In this paper, we develop a method for optimizing parameters and we apply it to the Gaussian model of folding. First, we describe the Gaussian model, then the optimization method, and finally we show that it succeeds in finding parameters that give globally optimal conformations for an HP-type lattice model and then for a more realistic continuous protein model with 210 parameters.

2. The Gaussian model of protein folding

A protein molecule is modeled as a linear chain of n beads. The position of the ith bead relative to a fixed laboratory frame is $\mathbf{R}_i$. The instantaneous configuration of the chain is given by the matrix $\mathbf{R}$ of bead coordinates as $\mathbf{R} = \mathrm{col}[R_1, R_2, \ldots, R_n]$ with $R_i = (x_i, y_i, z_i)$ ($R_i = (x_i, y_i)$ in two-dimensions). The beads are subject to covalent bond potentials between neighboring beads along the chain and

☆ This paper was originally submitted to *Computational and Theoretical Polymer Science* and received on 30 November 2000; accepted on 10 February 2001. Following the incorporation of *Computational and Theoretical Polymer Science* into *Polymer*, this paper was consequently accepted for publication in *Polymer*.

* Corresponding author.
E-mail address: chaok@maxwell.ucsf.edu (C. Seok).

0032-3861/02/$ - see front matter
PII: S0032-3861(01)00421-9

to non-bonded interactions with other beads. Both covalent and nonbonded potentials are modeled using linear springs. The covalent bonds are represented by attractive springs that tend to shrink the bond length to zero. Since springs have zero equilibrium bond lengths, such interactions would tend to shrink the protein to a point. In order to prevent this, a mean-field repulsive potential is superimposed on the system that fixes the moment of inertia of the protein at a fixed value, $\mathbf{I}_0$. The energy of the chain is represented by

$$E(\mathbf{R}) = \sum_{1\le i<j\le n} a_{ij}|R_i - R_j|^2. \tag{1}$$

Let $\mathbf{\Gamma}$ be the matrix with entries

$$\gamma_{ij} = \begin{cases} a_{ij} & \text{for } i \neq j \\ -\sum_k a_{ik} & \text{for } i = j \end{cases}, \tag{2}$$

where a_{ij}s are the spring constants for the interaction between ith and jth beads. It is convenient to introduce the vectors, $\mathbf{X} = \mathrm{col}[x_1, x_2, \ldots, x_n]$, $\mathbf{Y} = \mathrm{col}[y_1, y_2, \ldots, y_n]$, $\mathbf{Z} = \mathrm{col}[z_1, z_2, \ldots, z_n]$ so that $\mathbf{R} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$. The energy can then be written as $E(\mathbf{R}) = E(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \mathbf{X}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{X} + \mathbf{Y}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{Y} + \mathbf{Z}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{Z}$. To find the minimum energy state, we look for the configuration $\mathbf{R} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ that minimizes the energy, under the constraint

$$\mathbf{R}^{\mathrm{T}}\mathbf{R} = \begin{bmatrix} \mathbf{X}\cdot\mathbf{X} & \mathbf{X}\cdot\mathbf{Y} & \mathbf{X}\cdot\mathbf{Z} \\ \mathbf{X}\cdot\mathbf{Y} & \mathbf{Y}\cdot\mathbf{Y} & \mathbf{Y}\cdot\mathbf{Z} \\ \mathbf{X}\cdot\mathbf{Z} & \mathbf{Y}\cdot\mathbf{Z} & \mathbf{Z}\cdot\mathbf{Z} \end{bmatrix} = \mathbf{I}_0.$$

Under an orthonormal change of coordinates, corresponding to a unitary transformation U of the space the energy is invariant but the moment of inertia I_0 changes to $U\mathbf{I}_0U^{\mathrm{T}}$. We choose the transformation U that diagonalizes the constraint

$$\begin{bmatrix} \mathbf{X}\cdot\mathbf{X} & \mathbf{X}\cdot\mathbf{Y} & \mathbf{X}\cdot\mathbf{Z} \\ \mathbf{X}\cdot\mathbf{Y} & \mathbf{Y}\cdot\mathbf{Y} & \mathbf{Y}\cdot\mathbf{Z} \\ \mathbf{X}\cdot\mathbf{Z} & \mathbf{Y}\cdot\mathbf{Z} & \mathbf{Z}\cdot\mathbf{Z} \end{bmatrix} = \begin{bmatrix} \alpha_1^2 & 0 & 0 \\ 0 & \alpha_2^2 & 0 \\ 0 & \alpha_3^2 & \alpha_3^2 \end{bmatrix}$$

with $\alpha_1^2 \ge \alpha_2^2 \ge \alpha_3^2 > 0$. Placing the origin at the center of mass of $\mathbf{R}$, we aim to minimization $E(\mathbf{R}) = \mathbf{X}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{X} + \mathbf{Y}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{Y} + \mathbf{Z}^{\mathrm{T}}\mathbf{\Gamma}\mathbf{Z}$, subject to the constraints

$$\sum_{k=1}^n x_k = \sum_{k=1}^n y_k = \sum_{k=1}^n z_k = 0, \tag{3}$$

$$\sum_{k=1}^n x_k^2 = \alpha_1^2, \qquad \sum_{k=1}^n y_k^2 = \alpha_2^2, \qquad \sum_{k=1}^n z_k^2 = \alpha_3^2,$$
$$\sum_{k=1}^n x_k y_k = \sum_{k=1}^n x_k z_k = \sum_{k=1}^n y_k z_k = 0. \tag{4}$$

The solution is related to the eigenvalue problem

$$(\mathbf{\Gamma} - \lambda\mathbf{I})\mathbf{R} = 0. \tag{5}$$

Since $\mathbf{\Gamma}$ is positive Eq. (5) has n real eigenvalues, $0 = \lambda_0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-1}$. We denote by $\{R_i\}$, the corresponding orthonormal set of eigenvectors. The eigenvector $R_0 = n^{-1/2}[1, 1, \ldots, 1]^{\mathrm{T}}$ corresponding to $\lambda_0 = 0$ is redundant due to the first constraint in Eq. (3). From Bessel's inequality and a convexity argument, it follows that the minimum energy $E_0 = \alpha_1\lambda_1 + \alpha_2\lambda_2 + \alpha_3\lambda_3$ is reached when

$$\mathbf{R}_0 = [\alpha_1 R_1, \alpha_2 R_2, \alpha_3 R_3]. \tag{6}$$

3. The parameter optimization scheme

In this section, we describe a method for optimizing parameters. Our initial set of parameters in the Gaussian model is a set of spring constants a_{ij}. We are given the correct native structure of the protein and our aim is to change the spring constants so that the minimum energy structure of the model is as close as possible to the correct native structure.

Let us first assume that the elements of the $\mathbf{\Gamma}$ matrix are differentiable functions of a scalar parameter ϵ. Later in this section we will replace ϵ by the spring constants a_{ij}. $\mathbf{\Gamma} = \mathbf{\Gamma}(\epsilon)$ is an $n \times n$ self-adjoint matrix, differentiable with respect to ϵ. We denote by $\lambda_j(\epsilon)$, and $R_j(\epsilon)$ $j = 1, 2, \ldots, n$, the eigenvalues and normalized eigenfunctions. By the implicit function theorem, these will also be differentiable except for singular values. For a fixed i, differentiating Eq. (5) $(\mathbf{\Gamma}(\epsilon) - \lambda_i(\epsilon)I)R_i(\epsilon) = 0$ we get

$$(\mathbf{\Gamma}(\epsilon) - \lambda_i(\epsilon)\mathbf{I})\dot{R}_i(\epsilon) + (\dot{\Gamma}(\epsilon) - \dot{\lambda}_i(\epsilon)\mathbf{I})R_i(\epsilon) = 0. \tag{7}$$

The coefficient $(\mathbf{\Gamma}(\epsilon) - \lambda_i\epsilon)\mathbf{I})$ of $\dot{R}_i(\epsilon)$ above is singular. Nevertheless, since $|R_i(\epsilon)|^2 = 1$, $\dot{R}_i(\epsilon)\cdot R_i(\epsilon) = 0$; so $\dot{R}_i(\epsilon)$ has no component in the direction of $R_i(\epsilon)$. When $\lambda_i(\epsilon)$ is not a multiple eigenvalue, we can invert the matrix $(\mathbf{\Gamma}(\epsilon) - \lambda_i(\epsilon)\mathbf{I})$ on the subspace orthogonal to $R_i(\epsilon)$. This gives

$$\dot{R}_i(\epsilon) = -(\mathbf{\Gamma}(\epsilon) - \lambda_i(\epsilon)\mathbf{I})^{-1}(\dot{\mathbf{\Gamma}}(\epsilon) - \dot{\lambda}_i(\epsilon)\mathbf{I})R_i(\epsilon),$$

where the 'inverse' should be understood in the above sense. In this case $\dot{R}_i(\epsilon)$ can be explicitly written in terms of the remaining eigenvectors

$$\dot{R}_i(\epsilon) = \sum_{k\neq i} \frac{R_i(\epsilon)^{\mathrm{T}}\dot{\mathbf{\Gamma}}(\epsilon)R_k(\epsilon)}{\lambda_i(\epsilon) - \lambda_k(\epsilon)}. \tag{8}$$

Note that the term $\dot{\lambda}_i(\epsilon)$ $R_i(\epsilon)$ drops out due to orthogonality. This last equation can be derived from the orthonormal eigenvector expansions

$$\dot{R}_i(\epsilon) = \sum_{k=1}^n c_k R_k(\epsilon),$$

$$(\dot{\mathbf{\Gamma}}(\epsilon) - \dot{\lambda}_i(\epsilon)\mathbf{I})R_i(\epsilon) = \sum_{k=1}^n ((\dot{\mathbf{\Gamma}}(\epsilon) - \dot{\lambda}_i(\epsilon)\mathbf{I})R_i(\epsilon)\cdot R_k(\epsilon))R_k(\epsilon). \tag{9}$$

Substituting Eq. (9) into Eq. (1) and solving for the coefficients c_k gives Eq. (8).

In the following sections $\mathbf{\Gamma}$ will be taken to be of the form

$$\mathbf{\Gamma} = \sum_{\nu} \epsilon_{\nu} \Gamma_{\nu}$$

with constant matrices $\mathbf{\Gamma}_{\nu}$. The parameters ϵ_{ν} will represent either a single or a group of spring constants. For the two-dimensional lattice models of Section 4.1 we will take $\epsilon_{\nu} = a_{ij}$, the spring constant between the i-th and the j-th beads, hence ν will vary over all pairs $1 \le i < j \le n$. Then $\mathbf{\Gamma} = \mathbf{\Gamma}_{ij}$ where

$$\mathbf{\Gamma}_{ij} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{10}$$

with 1s at the ii, jj, positions -1's at the ij and ji positions, and all of the remaining entries equal zero. In the three-dimensional protein models ϵ_{ν} will depend on the pair of amino acids interacting; $\epsilon_{\nu} = \epsilon_{AB}$ for the amino acid pair (A, B). In this case the spring constant $a_{ij} = \epsilon_{AB}$ if the i-th bead contains the amino acid A and j-th contains B. Then $\mathbf{\Gamma}_{\nu} = \mathbf{\Gamma}_{AB}$ is the sum of the corresponding $\mathbf{\Gamma}_{ij}$s.

With this choice of $\mathbf{\Gamma}$, the derivative $\partial \Gamma / \partial \epsilon_{\nu}$ is

$$\frac{\partial \mathbf{\Gamma}}{\partial \epsilon_{\nu}} = \mathbf{\Gamma}_{\nu}. \tag{11}$$

Through the optimization process, the parameters ϵ_{ν} change, hence the native structure of the chain also changes. In the present analysis, this change is calculated by using the Gaussian Model. At each new step, the minimum energy configuration of the chain is calculated using the new parameters, and the new parameters are calculated by minimizing the distance between the instantaneous configuration ($\mathbf{R}_0 = [\alpha_1 R_1, \alpha_2 R_2, \alpha_3 R_3]$ of Eq. (6)) and the known minimum energy configuration $\mathbf{N}$, i.e. the target structure. The distance between the native configuration and any other configuration may be expressed in terms of various distance measures. In the present work we choose the affine transformation invariant distance $d(\mathbf{L},\mathbf{N})$ between two configurations $\mathbf{L}$ and $\mathbf{N}$, defined as follows:

$$d(\mathbf{L}, \mathbf{N}) = \sqrt{1 - \frac{|\det \mathbf{L}^{\mathrm{T}} \mathbf{N}|}{\sqrt{\det \mathbf{L}^{\mathrm{T}} \mathbf{L}} \sqrt{\det \mathbf{N}^{\mathrm{T}} \mathbf{N}}}}. \tag{12}$$

This distance is obtained from the inner product $\langle \mathbf{L}, \mathbf{N} \rangle = \det \mathbf{L}^{\mathrm{T}} \mathbf{N}$, which in turn arises from representing $\mathbf{L} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ by the alternating product $\mathbf{X} \wedge \mathbf{Y} \wedge \mathbf{Z}$. In the inner product defined above, linear transformations of the plane act like scalars; to be exact $\langle \mathbf{L}T, \mathbf{N} \rangle = \det T \langle \mathbf{L}, \mathbf{N} \rangle$. The distance is actually semidefinite; $d(\mathbf{L}, \mathbf{N}) = 0$ implies that $\mathbf{N} = \mathbf{L}T$ for some nonsingular linear transformation T. Moreover for any nonsingular linear transformation T, $d(\mathbf{L}T, \mathbf{N}) = d(\mathbf{L}, \mathbf{N})$. This last property of the affine invariant distance allows us to take a shortcut in our computations. Instead of the instantaneous configuration $\mathbf{R}_0$ we simply take $\mathbf{L} = [R_1, R_2]$. For a simpler notation we put $\mathbf{X} = R_1$, $\mathbf{Y} = R_2$, $\mathbf{Z} = R_3$ and $\mathbf{N} = (X_N, Y_N, Z_N)$. We can now differentiate $d^2(\mathbf{L},\mathbf{N})$ with respect to $\mathbf{L}^{\mathrm{T}}\mathbf{L} = 1$

$$\frac{\partial d^2(\mathbf{L}, \mathbf{N})}{\partial \epsilon_{\nu}} = - \frac{\operatorname{sign}(\det \mathbf{L}^{\mathrm{T}} \mathbf{N}) [\det[\dot{\mathbf{X}}, \mathbf{Y}, \mathbf{Z}]^{\mathrm{T}} X_N + \det[\mathbf{X}, \dot{\mathbf{Y}}, \mathbf{Z}]^{\mathrm{T}} Y_N + \det[\mathbf{X}, \mathbf{Y}, \dot{\mathbf{Z}}] Z_N]}{\sqrt{\det \mathbf{N}^T \mathbf{N}}}, \tag{13}$$

where the superposed dot denotes differentiation with respect to ϵ_{ν}, i.e. $\dot{\mathbf{X}} = \partial \mathbf{X} / \partial \epsilon_{\nu}$, $\dot{\mathbf{Y}} = \partial \mathbf{Y} / \partial \epsilon_{\nu}$, $\mathbf{Z} = \partial \mathbf{Z} / \partial \epsilon_{\nu}$. These partial derivatives are computed from Eq. (8), using Eq. (10) for $\mathbf{\Gamma}$.

Calculations start with the given initial values of each ϵ_{ν}^0 and the eigenvectors corresponding to the solution of the Gaussian model, Eq. (3). Then, the new values of ϵ_{ν} are obtained from the relation

$$\epsilon'_{\nu} = \epsilon_{\nu}^0 - h \frac{\partial d^2(\mathbf{L}, \mathbf{N})}{\partial \epsilon_{\nu}}. \tag{14}$$

Here, h is a scaling factor. The new values of ϵ_{ν}s from Eq. (14) are then substituted in Eq. (2) to obtain the new matrix $\mathbf{\Gamma}$, which is then used to solve Eq. (3). Inasmuch as the distance defined by Eq. (12) is invariant to nonsingular linear transformations, the structure obtained at the end of the iterative scheme may be the affinely transformed version of the native structure. This may suitably be remedied by a back transform, by postmultiplying $\mathbf{L}$ with T where the latter is given by

$$T = \begin{bmatrix} X_N \cdot \mathbf{X} & X_N \cdot \mathbf{Y} & X_N \cdot \mathbf{Z} \\ X_N \cdot \mathbf{Y} & Y_N \cdot \mathbf{Y} & Y_N \cdot \mathbf{Z} \\ X_N \cdot \mathbf{Z} & Y_N \cdot \mathbf{Z} & Z_N \cdot \mathbf{Z} \end{bmatrix}. \tag{15}$$

4. Results and discussion

4.1. Two-dimensional lattice model

For the proof of principle, we first take a simple model in which beads are either hydrophobic (H) or polar (P). We consider an HP model: HH and PP pair contacts in the native structure are connected with springs. Calculations of the optimized $\mathbf{\Gamma}$ matrix are performed as follows: The coordinates of the native configuration and initial values of $a_{ij} = \epsilon_{\nu}$ are given as input. The initial values are chosen as follows: (1) spring constants for all of the covalent bonds, are chosen as unity. (2) The spring constants representing the HH and PP interaction energies are chosen as $-\epsilon_1$, same for all HH and PP pairs. (3) Interaction between H and P units are initially chosen as zero. (4) In order to keep the HH pairs in the inner part of the protein and the PP pairs in the outside, a constant energy $-\epsilon_0$ is added to each HH pair and subtracted from each PP pair. These energy parameters constitute the starting values that enter the initial $\mathbf{\Gamma}$ matrix. The vector of positions, $\mathbf{R}$, are normalized and expressed

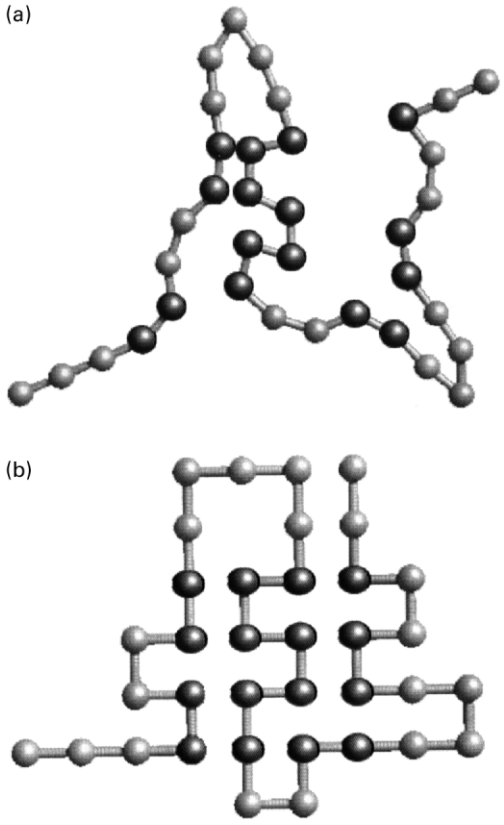

Fig. 1. The minimum energy configuration of the 36-mer (see Ref. [2] for the HP sequence) predicted by the Gaussian model: (a) before, and (b) after optimization.

with respect to the centroidal coordinate frame. Minimization is performed according to Eq. (14). The $\mathbf{\Gamma}$ matrix is upgraded with the new values of a_{ij} and the new configuration is obtained by solving Eq. (1). This procedure is repeated until the distance measure given by Eq. (12) converges to zero.

We have worked out the computations for the 20-mer, the 25-mer and the 36-mer (see Ref. [2]). The configurations obtained are practically the same as the result obtained by exact enumeration. Fig. 1a and b show results for the 36-mer before and after optimization. The entries of optimized $\mathbf{\Gamma}$ matrices are too detailed for reporting here, and are given for only the 20-mer in Table 1. The starting ϵ_1 values are chosen as 0.07, 0.05, and 0.031 for the 20, 25 and 36-mers, respectively. The starting ϵ_0 values are chosen as 0.011, 0.02 and 0.00295 for the 20, 25 and 36-mers, respectively. These initial values are observed to yield fast convergence to the given structures. The time for computations increased with the square of the number of beads along the chain. The final energies obtained at the end of the optimization procedure are 43.3, 46.2 and 101.

A brief description of some observed average features of the entries are as follows: The optimized energies of interaction depend on the chain length, the energies being higher for the shorter chains. Also, the energies depend weakly on the separation between pairs. This is shown in Figs. 2a–c. In

Table 1

	2-P	3-H	4-P	5-P	6-H	7-H	8-P	9-H	10-P	11-P	12-H	13-P	14-H	15-H	16-P	17-P	18-H	19-P	20-H
1-H	− 0.988	− 0.066	0.032	0.041	− 0.128	− 0.100	− 0.61	− 0.121	− .046	− 0.055	− 0.147	− 0.30	− 0.171	− 0.158	− 0.030	− 0.030	− 0.121	− 0.040	− 0.128
2-P		− 0.978	− 0.005	0.047	− 0.023	0.075	− 0.057	0.077	0.033	0.022	0.003	− 0.005	− 0.062	− 0.061	− 0.034	− 0.067	− 0.041	− 0.136	− 0.061
3-H			− 0.998	0.038	− 0.124	− 0.032	− 0.079	− 0.083	0.000	0.041	− 0.149	0.058	− 0.164	− 0.109	0.032	0.039	− 0.070	− 0.023	− 0.117
4-P				− 0.976	− 0.015	0.072	− 0.141	0.007	− 0.053	0.021	− 0.083	0.041	− 0.082	0.014	− 0.015	0.019	0.042	− 0.057	− 0.023
5-P					− 0.989	0.023	− 0.158	− 0.074	− 0.165	− 0.092	− 0.136	− 0.029	− 0.088	0.020	− 0.070	− 0.006	0.056	0.004	0.013
6-H						− 0.959	0.010	− 0.017	0.090	0.097	− 0.103	0.050	− 0.157	− 0.106	− 0.017	0.006	− 0.114	− 0.060	− 0.165
7-H							− 1.005	− 0.111	− 0.065	− 0.061	− 0.184	− 0.041	− 0.169	− 0.114	− 0.110	− 0.075	− 0.127	0.002	− 0.113
8-P								− 0.971	− 0.001	0.045	− 0.022	0.021	− 0.031	0.076	− 0.050	0.031	0.056	− 0.008	− 0.028
9-H									− 1.005	0.028	− 0.126	0.034	− 0.101	0.001	− 0.065	0.017	− 0.058	0.070	− 0.090
10-P										− 0.975	− 0.016	− 0.027	0.061	0.122	− 0.135	− 0.034	0.032	0.042	− 0.002
11-P											− 0.996	− 0.069	0.007	0.056	− 0.189	− 0.139	− 0.060	− 0.038	− 0.041
12-H												− 0.971	− 0.064	0.011	0.007	0.081	− 0.017	0.093	− 0.079
13 − P													− 0.982	0.024	− 0.122	− 0.117	− 0.048	− 0.050	− 0.014
14-H														− 0.973	0.028	0.048	− 0.063	0.022	− 0.108
15-H															− 0.978	− 0.005	− 0.118	− 0.034	− 0.122
16-P																− 0.986	0.031	0.031	0.053
17-P																	− 0.0.997	− 0.013	0.035
18-H																		− 0.980	− 0.065
19-P																			− 0.995

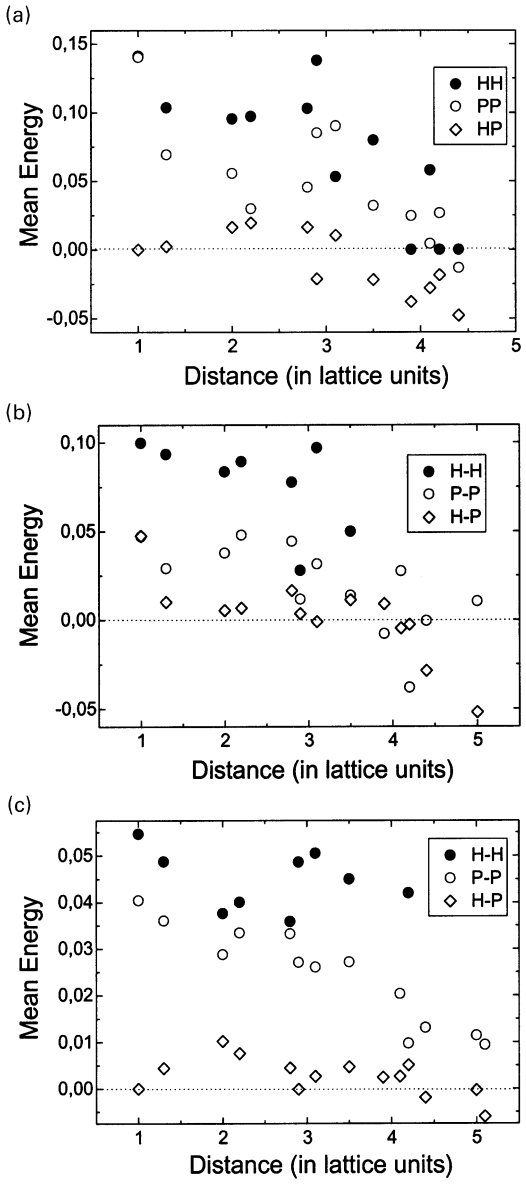

Fig. 2. The dependence of optimized spring constants on distance between beads: (a) for 20-mer, (b) 25-mer, and (c) 36-mer.

order to obtain these figures, the abscissa representing the distance between interacting pairs was divided into small intervals and the interacting pairs in these intervals were identified. The mean energy shown in the ordinate represents the interaction energy of pairs that fall into a given interval of separation. The optimized values of the energies for the HH, PP and the HP pairs are shown by the filled and empty circles and the empty diamonds as indicated in the legends. The spring constant representing the covalent bond between consecutive beads on a chain, which are chosen as unity in the Gaussian model, are not affected much by the optimization. General distance dependent potentials may also be used, by properly transforming them to Gaussian potentials at each step of the iteration.

Fig. 3. Structure of BPTI (dark) calculated from the optimized parameter set compared with that of the real native structure (white). Rms error is 1.7 Å.

4.2. Three-dimensional protein model

The above formulations for a two-dimensional problem can be easily extended to 3 dimensions and more realistic models. We represent all the atomic interactions, both covalent and noncovalent, as spring-like forces, and seek the optimal spring constants so that the lowest-energy structure for a protein calculated with the spring constants is as close as possible to the known native structure. We can find the lowest energy structure L exactly, as described above, and minimize the distance measure defined in Eq. (11) with the native structure N taken from the protein database.

We consider a simple model, where each amino acid in a protein is represented as single bead, and all the covalent bonds connecting neighboring beads are assumed to have the same strength. The parameters we aim to optimize are the spring constants of the nonbonded interactions relative to the single covalent bond spring constant. We assume that the strengths of the nonbonded interactions depend on the types of amino acids that interact, so the total number of parameters is 210, counting all the possible types of amino acid pairs. Initial values of the parameters were chosen to be on the order of 1/100 of the covalent spring constant at the beginning of the iteration, and the final optimized values remained about the same order of magnitude.

We first apply this method to the structure of Bovine Pancreatic Trypsin Inhibitor (BPTI). The structure calculated from the optimized parameters is compared with the real native structure in Fig. 3. The parameter optimization method gives a root mean square error between the two structures of 1.7 Å. This error is, in part, due to the fact that the number of parameters is about the same as the degrees of freedom for this protein.

We then optimize parameters for five different proteins simultaneously by minimizing the sum of squares of the distance errors of the five proteins. The results are shown in Fig. 4. In this case, the errors are larger, 3.5 Å, over the set

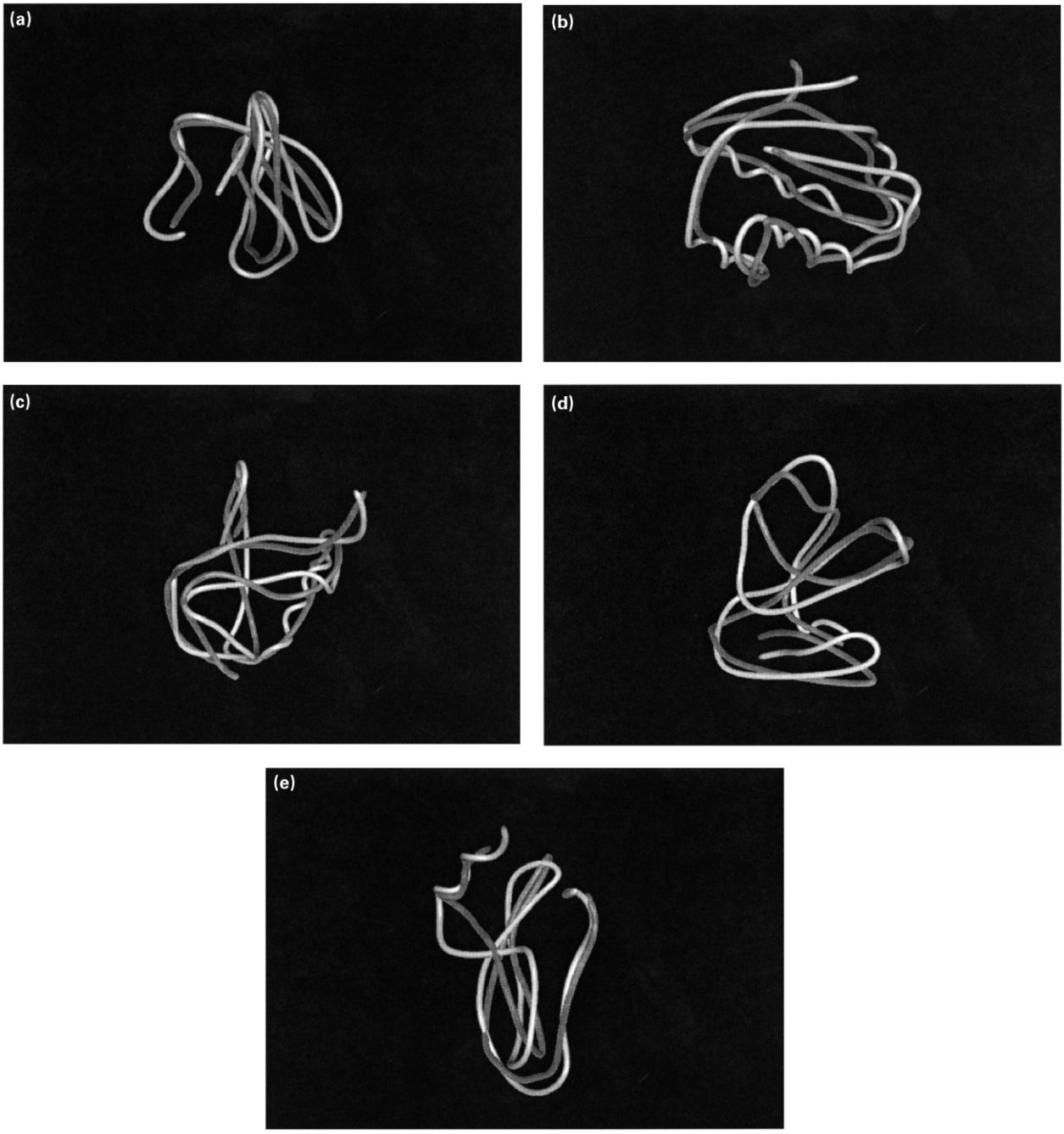

Fig. 4. Structures of five different proteins (dark) calculated from the optimized parameters in the multiple protein optimization compared with their native structures (white). The PDB identifiers and the rms errors are: (a) 1B4O.pdb (3.8 Å), (b) 3EBX.pdb (3.8 Å), (c) 1BTA.pdb (3.2 Å), (d) 1UBI.pdb (3.3 Å), (e) 5PTI.pdb (3.5 Å).

of all five proteins. We find that the structures have correct overall topologies, although there are errors in the detailed local structures. In this case, the number of degrees of freedom is about five times as large as the number of parameters, so it is not trivial to get this small error without the present optimization strategy. The parameter set obtained by this minimization is shown in Table 2.

The errors grow larger when we optimize 210 parameters for increasingly larger numbers of proteins at the same time. This appears to be due to the following limitations of the present method. Currently, we find only local minima of the distance measure in the parameter space, so the iteration method we use does not give the globally optimal parameter set. Finding the global minimum or a minimum close to the global minimum is necessary because the dimensionality of the parameter space is large, so there are very many local minima. Second, we have fixed the covalent bond spring strength, so the bond lengths fluctuate in the

Table 2

	C	M	F	I	L	V	W	Y	A	G	T	S	Q	N	E	D	H	R	K	P
C	4.89	− 2.98	− 4.50	1.63	− 0.78	4.91	3.53	7.27	3.23	0.66	− 0.43	2.59	− 0.97	− 0.29	0.61	4.03	2.99	− 1.81	0.32	− 1.23
M		− 3.24	5.55	− 1.05	− 4.32	0.28	2.46	− 4.89	4.00	− 1.94	2.64	7.37	2.26	4.71	1.79	0.48	1.35	4.17	− 0.28	7.89
F			− 7.70	8.82	3.75	− 2.79	0.25	− 6.62	12.07	2.95	3.89	7.40	1.28	0.51	− 5.70	0.33	2.38	− 4.98	0.55	3.55
I				− 8.48	5.36	3.77	2.57	− 3.04	− 6.74	2.55	0.61	4.41	1.43	2.69	2.43	0.19	− 6.32	− 2.83	4.96	− 2.56
L					− 0.19	1.18	4.52	− 2.34	2.41	0.01	0.77	− 3.06	0.51	4.83	0.48	0.00	0.18	0.03	1.18	0.10
V						− 9.98	− 3.10	2.45	2.33	1.90	0.69	2.91	− 0.81	− 7.80	5.01	− 0.46	− 5.97	4.68	0.88	4.08
W							− 11.1	1.04	9.06	6.59	− 6.69	− 0.22	1.15	1.05	1.44	− 5.48	4.15	− 3.42	1.94	− 3.34
Y								− 3.61	4.69	1.33	7.68	5.48	0.42	4.86	1.30	− 0.85	8.64	− 2.12	0.22	4.84
A									− 8.21	− 1.92	− 2.16	1.63	0.88	4.43	4.12	− 0.83	1.53	2.55	2.88	− 3.22
G										− 1.27	1.54	0.75	2.04	1.73	− 2.17	− 0.71	5.63	0.41	0.24	4.23
T											− 3.55	− 0.28	0.76	− 3.29	2.60	2.68	11.54	3.78	2.57	− 4.22
S												− 6.54	2.47	4.61	3.86	− 3.85	3.56	1.71	− 1.78	− 0.95
Q													− 3.94	6.58	0.96	0.89	1.45	− 0.84	1.68	− 0.94
N														− 11.0	0.89	0.70	2.28	2.61	− 5.78	4.93
E															− 3.92	3.81	1.59	− 0.30	0.18	0.76
D																3.06	− 2.42	4.74	1.09	3.22
H																	2.75	− 0.76	1.42	0.20
R																		5.17	− 1.02	1.21
K																			− 0.92	4.53
P																				− 2.56

predicted structures, making it difficult to achieve accurate predictions of ordered structures such as alpha-helices. Fixing the bond lengths (and/or angles) by introducing quadratic constraints might improve the predictions. This would require iterative solution for the Lagrange multipliers. Finally, we have used only the very simplest possible model here with no atomic detail; better models should also improve the predictions.

5. Conclusions

We have described a method that optimizes many parameters simultaneously in a protein folding model. In particular, the model we use is the Gaussian model, in which monomers are beads, the different types of covalent and noncovalent interactions are all represented as spring forces, and the overall size and shape of the molecule is fixed by a Lagrange multiplier constraint on the radius of gyration. The advantage of the Gaussian Model is that forces are linear in displacements so finding the global minima (native conformation) can easily be handled with matrix algebra, and thus the parameters can be optimized in a systematic way. The method begins with an initial configuration, and with a target conformation, namely the native structure to which the sequence is supposed to fold. The method iteratively computes a series of changes in parameters until the lowest energy state of the molecule comes as close as possible to the native conformation. We illustrate the method on a few small proteins. For example, for BPTI, the method finds 210 contact parameters that cause the predicted native structure to be within about 1.7 Å from the true native state.

References

[1] Rosen JB, Phillips AT, Oh SY, Dill KA. Biophys J 2001;79:2814–24.
[2] Erman B, Dill KA. J Chem Phys 2000;112:1050–6.