The Elastic Net Algorithm and Protein Structure Prediction

KEITH D. BALL,[1] BURAK ERMAN,[2] KEN A. DILL[1]
[1]*Department of Pharmaceutical Chemistry, University of California at San Francisco,*
3333 California Street, Suite 415, San Francisco, California 94118
[2]*Faculty of Engineering and Natural Sciences, Sabanci University, 81474 Tuzla, Istanbul, Turkey*

Received 16 February 2001; Accepted 1 June 2001

Abstract: Predicting protein structures from their amino acid sequences is a problem of global optimization. Global optima (native structures) are often sought using stochastic sampling methods such as Monte Carlo or molecular dynamics, but these methods are slow. In contrast, there are fast deterministic methods that find near-optimal solutions of well-known global optimization problems such as the traveling salesman problem (TSP). But fast TSP strategies have yet to be applied to protein folding, because of fundamental differences in the two types of problems. Here, we show how protein folding can be framed in terms of the TSP, to which we apply a variation of the Durbin–Willshaw elastic net optimization strategy.[1] We illustrate using a simple model of proteins with database-derived statistical potentials and predicted secondary structure restraints. This optimization strategy can be applied to many different models and potential functions, and can readily incorporate experimental restraint information. It is also fast; with the simple model used here, the method finds structures that are within 5–6 Å all-C_α-atom RMSD of the known native structures for 40-mers in about 8 s on a PC; 100-mers take about 20 s. The computer time τ scales as $\tau \sim n$, where n is the number of amino acids. This method may prove to be useful for structure refinement and prediction.

© 2002 John Wiley & Sons, Inc.   J Comput Chem 23: 77–83, 2002

Key words: protein structure prediction; elastic net method; Traveling Salesman Problem; combinatorial optimization; deterministic annealing

Introduction

In recent CASP experiments,[2,3] several *ab initio* protein structure prediction methods have shown remarkable success in blind prediction tests, in which only the amino acid sequence of an unpublished protein structure is given. It is now possible to predict substantial parts of the structures of various types of proteins up to 150–200 amino acids long to within 3–15 Å RMSD of their true native structures, within days to weeks on a single desktop computer.[4–13] However, the results of the most recent CASP experiment indicate that *ab initio* methods have a long way to go until they can provide the desired accuracy and reliability. This observation suggests that new approaches to structure prediction are worth exploring.

At the same time, there is a need to improve the speeds of these predictions, particularly if they are to be used for studying whole genomes. Our focus here is on increasing the computational speed of structure prediction. Current *ab initio* methods (recently categorized as *new fold* methods in CASP4) search rugged high-dimensional protein energy landscapes to find their global minima. Conformational search methods usually explore a large number of different protein conformations by branch-and-bound or stochastic search methods (typically Monte Carlo simulated annealing) or molecular dynamics. Branch-and-bound methods are slow because the computer time $\tau \sim a^n$ scales exponentially with the chain length n, where a is a branching constant.[14–16] For stochastic methods, the computer time depends not only on n but also on the amino acid sequence. Because the landscapes are rugged and the rate-limiting steps involve escaping from local minima, the time required to reach the global minimum is usually not known or predictable.

In contrast, for the traveling salesman problem (TSP), a well-known problem in combinatorial optimization, deterministic methods have been developed that reach near-optimal solutions very quickly.[17–22] In addition, these methods have been successful in solving a host of other optimization problems.[23–25] These strategies have not heretofore been applied to protein structure prediction, because protein folding differs in fundamental ways from the TSP and other matching problems. Nevertheless, both TSP and protein folding can be described in terms of fitness landscapes. This opens the possibility that many of the efficient combinatorial optimization techniques used for matching problems may be

Correspondence to: K. D. Ball; e-mail: kdb@maxwell.ucsf.edu

Contract/grant sponsors: Burroughs Wellcome Fund Interfaces Program and the Florida State University (Fellowship from the Program in Mathematics and Molecular Biology to K.D.B.)

Contract/grant sponsor: NIH; contract/grant number: GM34993 (to K.A.D.)

applicable to protein structure prediction and other structural optimization problems in the physical and biological sciences. Here we show how to frame protein structure prediction in terms of one such matching problem—the TSP—which we solve using a modification of the Durbin–Willshaw elastic net method.[1]

In the following section, we describe the elastic net method as introduced by Durbin and Willshaw and applied to the TSP. We then show how the algorithm may be derived in the language of statistical mechanics and deterministic annealing. In the third section we demonstrate how the method can be generalized and applied to protein structure prediction. In the fourth section we present preliminary results we have obtained for a variety of small to intermediate-sized proteins. Finally, we discuss possible improvements and future applications of our method.

Elastic Net Method

Introduction and Application to the TSP

The Durbin–Willshaw elastic net (EN) method[1] belongs to the class of geometric neural networks, which also includes self-organizing maps.[22] Although it is closely related to the Hopfield–Tank algorithm[18] and other neural network optimization methods, it performs as well as or better than these methods, and tends to be faster.[1, 22, 23] The EN method has found many applications: as a clustering algorithm,[26] in pattern recognition and machine vision, and other matching problems.[27] Although the EN method yields consistently longer TSP tours than either simulated annealing, k-opt exchange heuristics[22] or the Lin-Kernighan[17] algorithm, and also becomes slower than these methods for more than 100 cities,[22] the method has the advantage of being easily parallelizable, and is more versatile than the exchange heuristics, which are applicable only to the TSP and similar matching problems.[28] This flexibility makes the EN algorithm a suitable candidate for structural optimization. The method has further advantages over stochastic methods such as simulated annealing, which are discussed later. Here we illustrate the essential ingredients of the EN method by reviewing its original formulation by Durbin and Willshaw and their application of the method to the TSP on the two-dimensional Euclidean plane.

In the TSP, we are given the coordinates of a fixed set of m cities $\mathbf{S} = \mathbf{S}_1, \ldots, \mathbf{S}_m$ in the xy-plane, where vector $\mathbf{S}_i$ is the location of city i. A salesman must visit each city once and only once, and does so by following a path, or *tour*, passing though the cities in a particular order. The tour consists of *n tour points* $\mathbf{R} = \mathbf{R}_1, \ldots, \mathbf{R}_n$, with $\mathbf{R}_j$ being the location of the salesman's jth stop.

Solving the TSP is a matter of matching the tour stops to the cities such that (1) each city is visited exactly once (the *matching constraint*), (2) the tour is closed ($\mathbf{R}_{n+1} \equiv \mathbf{R}_1$), and (3) the tour is the shortest possible path through the cities obeying (1) and (2); that is, the total length of the tour

$$l = \sum_{j=1}^{n} |\mathbf{R}_{j+1} - \mathbf{R}_j| \tag{1}$$

is at its global minimum. To meet these conditions, the elastic net (EN) strategy uses two different sets of zero-equilibrium-length harmonic springs. The first set of n springs connects the tour points in a closed loop. We refer to these as the *tour springs*, because they describe how the tour points are connected in succession. In this representation, the tour is analogous to a linear polymer chain—beads connected by springs—which can explore different conformations in the plane to create a route through the cities. Each tour spring has an energy $E_j = \alpha_t(\mathbf{R}_{j+1} - \mathbf{R}_j)^2$, where α_t is a force constant. These attractive springs serve to keep the tour as short as possible. The second set of mn springs connects each tour point $\mathbf{R}_j$ to each city $\mathbf{S}_i$ with an energy proportional to $(\mathbf{S}_i - \mathbf{R}_j)^2$. These *city springs* tend to force the salesman's tour to pass through the cities.

In the EN algorithm, there are three possible choices for the number of cities m and the number of tour stops n: (1) $m < n$, (2) $m = n$, and (3) $m > n$. So that the matching constraint in the TSP be satisfied, we must select either case (1) or (2). Because in case (1) it is possible to have more tour points than cities, only m of these tour points will ultimately become tour stops (i.e., tour points matched uniquely to cities). Workers in the field have typically set $n = 2.5m$.[1, 20, 29, 30] At finite temperatures, an elastic tour will tend to "cut corners" near cities to minimize the length penalty. Using more tour stops than cities in the TSP reduces this tendency by reducing the average interval between tour stops, which reduces the elastic tension in the tour. This slackening allows the city springs in case (1) to pull the tour stops closer to them at a finite-temperature equilibrium than in case (2).

The EN method achieves the matching constraint via an iterative procedure that computes on each cycle a Boltzmann-like function ϕ for the city springs,

$$\phi\big(|\mathbf{S}_i - \mathbf{R}_j|, K\big) = \exp\big(-(\mathbf{S}_i - \mathbf{R}_j)^2/2K^2\big). \tag{2}$$

Normalizing over all n city springs connected to city i gives the following statistical weights for each of these springs:

$$w_{ij} = \frac{\phi(|\mathbf{S}_i - \mathbf{R}_j|, K)}{\sum_k \phi(|\mathbf{S}_i - \mathbf{R}_k|, K)}, \tag{3}$$

where K is a parameter. These weights act as variable spring constants, which for fixed K become weaker as the distance $|\mathbf{S}_i - \mathbf{R}_j|$ increases. To find the optimal tour, the EN method incrementally changes the coordinates of each $\mathbf{R}_j$ on the tour according to the rule

$$\Delta\mathbf{R}_j = \alpha_c \sum_i w_{ij}(\mathbf{S}_i - \mathbf{R}_j) + K\alpha_t(\mathbf{R}_{j+1} - 2\mathbf{R}_j + \mathbf{R}_{j-1}). \tag{4}$$

Here α_c and α_t are coefficients that control the relative strengths of the city and tour springs, respectively.

The progressive evolution of the tour as a function of K has been rigorously analyzed by Durbin et al.,[29] and is outlined below. The EN method starts at a sufficiently high K such that all city springs have essentially the same strength, and provide a uniform force on each point along the salesman's tour. Because the points feel no net pull toward any of the cities, the tour shrinks to its shortest possible length, namely $l = 0$. As K is lowered, the spring weight determined by eq. (3) approaches unity for the shortest city spring attached to a given city, while the strengths of the other $n - 1$ springs associated with this city diminish exponentially in a winner-take-all fashion. Hence, below some critical value of K the closest tour point is pulled preferentially toward this city, while the others are pulled away from it and move towards other cities. The choice of w_{ij} in eq. (3) ensures that in the limit $K \to 0$ exactly one tour point will be pulled to each city, thus satisfying the matching constraint. Because the position updates in eq. (4) can be performed simultaneously for each j, the EN algorithm is parallelizable, although in this article we run the algorithm in single-processor mode. In addition, the algorithm generalizes to any number of dimensions.

At $K = 0$, the total energy of the tour springs will have reached a local extremum, which is hoped to be at or close to the global minimum. It is important to note that the above protocol actually minimizes the sum of squared distances (the L2 norm), not the desired L1 norm given by eq. (1), because the algorithm has proven to be unstable beyond a limited number of cities when using the L1 norm.[23] Workers in the field have typically assumed that the city topology that minimizes path length in the L2 norm will, for the typical city distribution, also be the shortest path topology under the L1 norm. Tests we have run on small city distributions support this assumption. In any case, this issue is limited to objective functions that represent length penalties, and will not arise in our predictions of protein structure.

Statistical Mechanics Description

The EN strategy has a close analogy to statistical mechanics.[23, 28, 29, 31] In this analogy, the parameter K is identified with a temperature T via the relation $T = K^2$ and the Boltzmann constant set to unity.[23] Statistical mechanics has provided a metaphor for many optimization strategies,[32] with simulated annealing being the best-known example.[33] Unlike simulated annealing, which is stochastic, the EN method is deterministic, and is therefore much more efficient.

Following Yuille,[23] we now show that the eq. (4) can be derived in terms of minimization on a free energy landscape. We start with a Hamiltonian for the extended system of tour points and cities, with both city spring and tour spring interactions:

$$H = \sum_{i,j} \frac{V_{ij}}{2} (\mathbf{S}_i - \mathbf{R}_j)^2 + E_{\text{tour}}, \tag{5}$$

where $E_{\text{tour}} = \alpha_t \sum_j (\mathbf{R}_{j+1} - \mathbf{R}_j)^2$ is the total energy of the tour springs. For the conformation of a physical system (e.g., a protein or polymer chain) E_{tour} is equivalent to the internal energy, which in general may include all types of noncovalent interactions, in addition to the covalent bonds holding the chain together. Here the V_{ij} are the coupling constants of the cities to the tour points. In the extended system, the V_{ij} are fixed in any single instance, but are not known in advance. Possible values of the matrix V must satisfy the matching constraint, which requires $V_{ij} = 1$ if the jth tour stop is matched to the ith city, and $V_{ij} = 0$ otherwise. That is, the matrix V_{ij} must contain only a single "1" in each of its columns and rows. The partition function Z for the chain is given by $Z = \int_{\mathbf{R}} \sum_{V_{ij}} e^{-H(\mathbf{R})/T}$ where the sum is taken over all possible conformations $\mathbf{R}$ of the chain and over all possible matrices $\mathbf{V} = [V_{ij}]$. From the Hamiltonian in eq. (5) we obtain

$$Z = \int d^{3n}\,\mathbf{R} \sum_V \left\{ \prod_i e^{-\sum_j V_{ij}(\mathbf{S}_i - \mathbf{R}_j)^2/2T} \right\} e^{-E_{\text{tour}}/T}. \tag{6}$$

In general, finding the matrix V, which maximizes the partition function by exhaustive search, is a computationally prohibitive combinatorial problem, even with the matching constraints on the V_{ij}. Instead, Yuille considered an ensemble of possible matchings, and assumed that the coupling of a tour point to each city is governed in a statistical sense by a Boltzmann probability distribution, with the coupling probabilities being independent for each city. This formalism implies a fuzzy matching of tour points to cities that evolves in a continuous fashion toward a discrete match. In the limit $T \to 0$, violations of the matching constraint become highly improbable, so the assumption of independent probabilities is valid. With these assumptions, the partition function may be written in a much simpler form:

$$Z = \int d^{3n}\,\mathbf{R} \prod_i \left\{ \sum_j e^{-(\mathbf{S}_i - \mathbf{R}_j)^2/2T} \right\} e^{-E_{\text{tour}}/T}. \tag{7}$$

The macroscopic partition function Z is the integral over contributions at each conformation $\mathbf{R}$. The contributions to Z at each $\mathbf{R}$ may be written in terms of a microscopic free energy $F(\mathbf{R}, T)$, so eq. (7) becomes

$$Z = \int d^{3n}\,\mathbf{R} \exp\bigl(-F(\mathbf{R}, T)/T\bigr), \tag{8}$$

where $F(\mathbf{R}, T)$ is the internal free energy landscape

$$\begin{aligned} F(\mathbf{R}, T) &= U(\mathbf{R}) - TS(\mathbf{R}) \\ &= E_{\text{tour}} - T \sum_i \log \left\{ \sum_j e^{-(\mathbf{S}_i - \mathbf{R}_j)^2/2T} \right\}. \end{aligned} \tag{9}$$

The internal free energy is a result of both the internal conformational energy $U \equiv E_{\text{tour}}$ and the entropic effects of the city springs acting through S, which tend to disperse the tour points evenly in space at high temperatures. As T is lowered, conformations that are closer to satisfying the matching constraint have higher entropy, and thus have lower free energy than other conformations. As $T \to 0$ only conformations that satisfy the constraint exactly will have a significant occupation probability.

In searching over the energy landscape for the native state, the protein attempts to find the conformation $\mathbf{R}$ that minimizes $F(\mathbf{R})$, or simultaneously minimizes $U(\mathbf{R})$ and maximizes $S(\mathbf{R})$. In the zero-temperature limit, the stable equilibrium conformation is at the global free energy minimum; because $TS \to 0$ in this limit, the global optimum of E_{tour} is also obtained.

The problem is, therefore, reduced to finding the global minimum of $F(\mathbf{R}, T)$ in the zero-temperature limit. As Stolorz[24, 28] has shown, the EN algorithm attempts this through deterministic annealing on the larger landscape $F(\mathbf{R}, T)$. In this manner it is similar to optimization techniques that perform "simulated annealing in temperature,"[32] such as the Scheraga diffusion equation method[34] and Gaussian density annealing.[35] Furthermore, Stolorz has pointed out that eq. (9) has the form of a constrained optimization problem, where U is the function to be optimized, subject to the constraint S. Taking T to be a Lagrange multiplier, both $\mathbf{R}$ and T may be simultaneously optimized by satisfying the constrained optimization criteria

$$\nabla_{\mathbf{R}} F(\mathbf{R}, T) = \nabla_{\mathbf{R}} U(\mathbf{R}) - T\nabla_{\mathbf{R}} S(\mathbf{R}, T) = 0, \tag{10}$$

$$\partial_T F(\mathbf{R}, T) = -S(\mathbf{R}, T) - T\partial_T S(\mathbf{R}, T) = 0. \tag{11}$$

These criteria can be met by respectively iterating the steepest-descent and ascent differential steps $\Delta\mathbf{R}_j = -\nabla_{\mathbf{R}} F(\mathbf{R}, T)\Delta\tau$ and $\Delta T = +\partial_T F(\mathbf{R}, T)\Delta\tau$. In terms of eqs. (9), (10), and (11), these steps are

$$\Delta\mathbf{R}_j = \alpha_c \sum_i w_{ij}(\mathbf{S}_i - \mathbf{R}_j) - \alpha_t \nabla_{\mathbf{R}_j} E_{\text{tour}}, \tag{12}$$

$$\Delta T = \gamma\left[\sum_i \log \sum_j e^{-(\mathbf{S}_i - \mathbf{R}_j)^2/2T} - \frac{1}{2T}\sum_j \frac{\sum_i (\mathbf{S}_i - \mathbf{R}_j)^2 e^{-(\mathbf{S}_i - \mathbf{R}_j)^2/2T}}{\sum_k e^{-(\mathbf{S}_i - \mathbf{R}_k)^2/2T}}\right]$$

where α_c, α_t are the constants in eq. (4) that determine the relative step contributions to the conformation update of the city and tour-spring forces, respectively, and γ controls the rate of annealing. The time step $\Delta\tau$ is implicit in the scaling constants α_c, α_t, and γ. A steepest-ascent step, rather than the usual descent step, is used to update T, because $F(\mathbf{R}, T)$ is monotonically increasing in the direction $T \to 0^+$. Simultaneous minimization of T is not necessary; one of the commonly used annealing schedules could be used in place of eq. (13), as long as γ is chosen to be sufficiently small. The advantage of the constrained optimization approach is that it provides an adaptive step size not only in optimizing $\mathbf{R}$, but also T. Solution times τ for the EN algorithm scale as $\tau \sim mn$, which becomes n^2 when $m = n$. If the number of cities m is held constant for proteins of varying length, the computational time grows only as $\tau \sim n$.

The Elastic Net Method Applied to Protein Structure Prediction

To apply the elastic net strategy for the TSP to protein structure prediction, we regard the protein chain as the polymer-like tour of the salesman. Each tour point j at position $\mathbf{R}_j$ represents the C_α atom of an amino acid. Although the canonical TSP is two-dimensional, the protein chain is embedded in a three-dimensional space that also contains a quasi-uniform distribution of m cities. The protein chain has unconnected ends, as opposed to the closed "necklace" used in the TSP.

The problem in applying TSP strategies to protein folding is that the most obvious analogy to the cities in the TSP—that they should correspond to the desired position of amino acids—fails. In the TSP, the city coordinates are known in advance, and any viable tour must thread through them. But for protein folding, there is no advance knowledge of the native conformation. The key to our strategy is in recognizing that at high temperatures, the city-dependent entropic term in eq. (9) smoothes the free energy landscape. At sufficiently high T the smoothing makes the landscape convex, which prevents the system from becoming trapped in a metastable local minimum. Furthermore, in the low-temperature limit of the EN, or in any viable solution of the TSP, a tour passing through distinct cities (or clusters of cities, as discussed below) is approximately a self-avoiding flight. Hence the effect of S in eq. (9) is similar to that of the entropy of excluded volume derived from polymer theory.[36] In conformational searches performed by molecular dynamics or stochastic annealing, all-atom potential functions are extremely rugged due to the inclusion of very stiff repulsive-core terms that enforce excluded volume. These methods spend a vast majority of their time trying to circumvent high-energy barriers. In contrast, by enforcing the matching constraint the city springs of the EN algorithm enforce excluded volume in an approximate way without a sharp increase in landscape ruggedness and computer time.

This role for the cities suggests that the ideal city distribution would be one that is both compatible with native protein structures and covers the space of compact protein folds in an unbiased way, consistent with the principle of maximum entropy.[37] We therefore distribute the cities on a cuboctahedral lattice, which has been shown to effectively capture the virtual bond angles and symmetries of amino acid distributions in real proteins.[38] This choice of lattice thus helps to minimize potential distortions of protein structure models caused by a bias arising from the lattice symmetry. In our method, $\mathbf{S}_i$ is now the lattice coordinate of the ith city relative to the lattice center, given by the relation $\sum_i \mathbf{S}_i = \mathbf{0}$. A low-bias lattice distribution will have more cities than there are tour points ($m > n$). In this case, a more generalized matching constraint will be enforced: namely, that on average a given bead will be pulled approximately towards the centroid of the m/n closest cities.[26] In short, the idea is that the protein is a polymer chain, akin to the salesman's tour, that is subject to three forces: (1) the tour springs representing the covalent bonds between monomers j and $j + 1$; (2) the city springs that enforce a relatively smooth funnel-shaped landscape, partly due to excluded volume; and (3) intramolecular, noncovalent springs, taken from a statistical potential (see below). These forces drive the chain towards its native conformation.

Perhaps the simplest off-lattice model of sequence–structure relationships in proteins is the Gaussian model,[39–41] in which each amino acid is a single bead, each covalent bond is a strong harmonic spring, and noncovalent interactions between pairs of amino acids are represented by weaker harmonic springs. This spring model resembles the length function E_{tour} used in the TSP. Using a generalized form of the Gaussian energy model, we replace E_{tour} in eq. (9) with E_{prot}, the internal energy of the protein chain. Burak and Dill[42] have tested our prediction algorithm on a two-dimensional off-lattice HP Gaussian protein model.[43, 44] Here, we use a more detailed energy function in three dimensions:

$$E_{\text{prot}} = E_{\text{stat}} + E_{\text{cov}} + E_{\text{HP}} + E_{\text{restr}}, \tag{13}$$

$$E_{\text{stat}} = \sum_{i<j} a_{ij}^{\text{stat}} r_{ij}^2,$$

$$E_{\text{covalent}} = \sum_{i<n} a^{\text{cov}} (r_{i,i+1} - \overline{r_{i,i+1}})^2,$$

Table 1. Sample Parameters for the Model Protein Energy Function.

Parameter	Relative Strength
a^{stat}	0.05
a^{cov}	6.0
a^{H}	0.1
a^{P}	−0.1
a^{restr}	2.0–5.0

Here $a^{\text{stat}} = \max |a_{ij}^{\text{stat}}|$. All coefficients are given in scaled units in which the force constant of each city spring is unity.

$$E_{\text{HP}} = \sum_{i} a_i^{\text{HP}} r_i^2,$$

$$E_{\text{restr}} = \sum_{(i,j)\in\mathcal{R}} a_{ij}^{\text{restr}} (r_{ij} - \overline{r_{ij}})^2.$$

The set of parameter values we used for the models we present below are given in Table 1. As in the Gaussian model, pairwise noncovalent interresidue interactions are represented by the statistical potential E_{stat}. The coefficients a_{ij}^{stat} are taken from the Thomas–Dill knowledge-based potential.[45] Covalent peprtide bonds are represented as stiff harmonic potentials with equilibrium bond lengths of $\overline{r_{ij}} = 3.8$ Å. The burial of hydrophobic residues and segregation of polar residues to the protein surface is enhanced by the centrosymmetric square-law potential E_{HP}.

In cases where a set $\mathcal{R}$ of experimental or knowledge-based distance restraints is available, they may be added to the energy function as additional springs, which comprise the term E_{restr}. For the results presented in the next section, this term contains restraints for disulfide bonds known from the RCSB Protein Data Bank (PDB), with a spring constant of 5.0. In addition, we include restraints based on secondary structure predictions obtained using the PHD server[46] with a spring constant of 2.0. These restraints are added as harmonic springs for residue pairs $(j, j+k)$, where $k = [1, 4]$ for α-helices, and $k = [1, 3]$ for β-strands. For helices, the equilibrium distances for these restraints are determined from an idealized helix geometry, and for β-strands from approximate averages of these distances as observed in the PDB.[47]

For our calculations, we include cities on the cuboctahedral lattice within a simulation sphere of a given radius. The lattice site separation is set to 3.8 Å to mimic relative residue positions in real proteins. To reduce the number of cities used while maintaining an ordered sample over a sphere of radius 20–25 Å, we place cities on only a fraction of the lattice sites, determined by a Gaussian distribution $p_{\text{site}} \propto \exp[-(r/\rho)^2]$, where $\rho = 10$ Å. For simplicity and predictability, we use a stepped linear annealing schedule instead of the dynamic schedule in eq. (13). Having tested the Stolorz annealing schedule more completely, we will use it for future structure predictions. For the initial protein conformation, we use a random spherical distribution of residues within the bounds of the lattice. We started our predictions at a temperature high enough that equilibrated protein models take the shape of collapsed balls of diameter ~4–5 Å. This finite diameter is due chiefly to the persistence of peptide bond lengths, which are maintained by stiff spring interactions.

We have applied the EN algorithm to a set of 14 proteins from the PDB in the range 36–136 amino acids. In addition, we tested our method on three protein domains that were targets in CASP3. We have tested the full annealing process of systematically lowering the system from high T ($\approx$60) down to $T \sim 1$. We have also found, however, that to a nearly equally good approximation, we can eliminate the iterations altogether by equilibrating at at moderately high $T \approx 40$ to obtain a viable protein chain conformation, then equilibrating at a single lower temperature T^* chosen in advance. We determined T^* by running predictions on a separate test set of proteins and found the value that tended to yield the lowest overall RMSD for these test proteins.

Results and Discussion

The purpose of this article is to explore the computational speed of a new optimization strategy, not to evaluate our simple square-law model of proteins. Nevertheless, some of the predictions are not unreasonable for a low-resolution model. Our results are given in Table 2, with selected predictions shown in Figure 1. Not surprisingly, the errors increase for larger proteins, approximately linearly with the radius of gyration R_γ of the protein, or as $n^{1/3}$. Also unsurprising was that our method works better for proteins that are globular than with those that are extended. This tendency is borne out in Table 2 by the disproportionally large RMS errors for the extended protein chains 1c94A (a single long α-helix) and 1a1hA, which are monomers in larger oligomeric proteins.

In our tests so far, the quality of the structures predicted by our algorithm appears to be largely independent of the type of fold. At present, the geometrical distance restraints we employ for enforcing both α-helices and β-strands improve our results by somewhat

Table 2. All-C$_\alpha$ RMSD Values for Elastic Net Predictions Using Restraints for Secondary Structure Predicted by PFD.[46]

Protein	n_{res}	RMSD (Å)	t (s)
1vii	36	5.17	7.37
1ajj	37	6.16	10.18
1c94A*	37	12.52	9.58
2erl	40	5.46	9.13
1res	43	5.92	8.73
1crn	46	5.65	9.31
1uxd	59	7.75	12.13
1era	62	7.99	13.08
2sn3	65	7.82	12.63
1ctf	68	7.99	15.29
1d3bA*	72	9.72	15.02
1hoe	74	8.70	15.41
1ubi	76	9.79	15.22
1d3bB*	81	10.79	18.79
1a1hA	85	14.56	18.32
1plc	99	10.65	19.91
1eca	136	11.91	32.89

Here n_{res} is the number of residues in each protein chain. Proteins are labeled by their PDB codes. Times shown are for an Athlon 700 MHz CPU. CASP3 targets are indicated with asterisks.

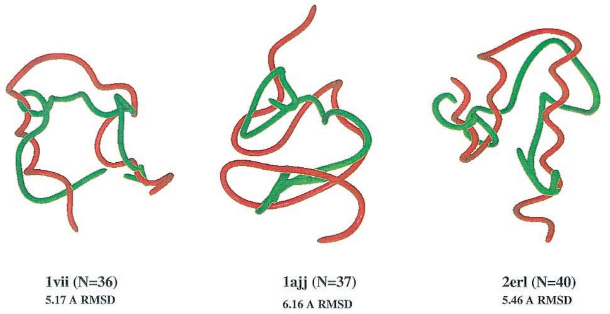

Figure 1. Selected protein models from Table 2 (green) compared to the true native states (red).

less than 1 Å on average. Here, the improvements obtained by using even strong restraints is limited, because the city springs exert an increased reactive force on the residues, due to the strain imposed by the restraint interactions. In terms of RMS errors, the present predictions are roughly typical of the average quality of *ab initio* predictions in CASP3 or better. For the CASP3 target T0059 (1d3bA in Table 2), considered to be one of the "hard" structures, the average RMS error of CASP3 models was 11.21 ± 2.14 Å RMSD, with only the five best models having less than 8 Å RMSD. Our error is 9.72 Å RMSD, which puts it in the top 20% of the 100 model predictions for this target. Although not as accurate as the best of the most recent *ab initio* CASP predictions, the present optimization strategy is about 100- to 1000-fold faster. The present method may, therefore, be useful as a low-resolution front end to higher-resolution refinement methods. Our method also has the advantage that it can be fully automated; we used it as the core of our fully automated ELAN-PROT prediction server,[48] in the recent CAFASP2 experiment,[49] which ran in tandem with CASP4.

There are several ways in which we are currently improving our method. We believe it can be improved by using a better energy function, by including hydrogen bonds, and by adding better secondary structure predictions. We also plan to tailor our algorithm more specifically to the task of structure prediction by determining the optimal positioning of the cities, as well as modifying the role of the city springs and the evolution of the city spring constants. Our approach could also be used with a more detailed chain model.

Conclusions

There are fast methods for finding near-optimal solutions of classical global optimization problems such as the Traveling Salesman Problem. In contrast, protein folding algorithms have tended to employ less efficient stochastic sampling strategies, such as Monte Carlo or molecular dynamics. We have described here a way to use combinatorial strategies in protein structure prediction. We illustrate the elastic net method of Durbin and Willshaw in conjunction with a square-law–based Gaussian model of proteins, in which the noncovalent interactions are taken from a statistical potential. We find that the method converges to a solution in about 8 CPU-seconds on a PC for typical 40-mers, about 20 s for 100-mers, and with a computer time that scales linearly with chain length. This method is not limited to square-law models; it can be used with a wide range of different potential functions. It can also incorporate external restraints taken from experiment, and could potentially be used in conjunction with data from NMR experiments, the mass spectrometry method of Young et al.,[50] or from homology models[9] to hasten the experimental structure determination. Moreover, it is fully automatable. Our approach may serve as an efficient way to speed up *ab initio* protein structure prediction and restraint-based structure determination.

References

1. Durbin, R.; Willshaw, D. Nature 1987, 326, 689.
2. CASP3: Proteins (Suppl. 3) 1999, 37, 1.
3. Fourth Meeting on the Critical Assessment of Techniques for Protein Structure Prediction (http://predictioncenter.llnl.gov/casp4/).
4. Baker, D. Nature 2000, 405, 39.
5. Eyrich, V. A.; Standley, D. M.; Friesner, R. A. J Mol Biol 1999, 288, 725.
6. Ishikawa, K.; Yue, K.; Dill, K. A. Prot Sci 1999, 8, 716.
7. Lee, J.; Liwo, A.; Ripoli, D. R.; Pillardy, J.; Scheraga, A. H. Proteins (Suppl. 3) 1999, 37, 204.
8. Ortiz, A. R.; Kolinski, A.; Skolnick, J. Proc Natl Acad Sci USA 1998, 95, 1020.
9. Ortiz, A. R.; Kolinski, A.; Rotkiewicz, P.; Ilkowski, B.; Skolnick, J. Proteins (Suppl. 3) 1999, 37, 177.
10. Samudrala, R.; Xia, Y.; Huang, E.; Levitt, M. Proteins (Suppl. 3) 1999, 37, 194.
11. Simons, K. T.; Bonneau, R.; Ruczinski, I.; Baker, D. Proteins (Suppl. 3) 1999, 37, 171.
12. Skolnick, J.; Kolinski, A.; Ortiz, A. R. J Mol Biol 1997, 265, 217.
13. Srinivasan, R.; Rose, G. D. Proteins 1995, 22, 81.
14. Eyrich, V. A.; Standley, D. M.; Felts, A. K.; Friesner, R. A. Proteins 1999, 35, 41.
15. Standley, D. M.; Eyrich, V.; Felts, A.; Friesner, R. A.; McDermott, A. E. J Mol Biol 1999, 285, 1691.
16. Yue, K.; Dill, K. A. Protein Sci 1996, 5, 254.
17. Lin, S.; Kernighan, B. Operat Res 1973, 21, 498.
18. Hopfield, J. J.; Tank, D. W. Biol Cybern 1985, 52, 141.
19. Miller, D. L.; Pekny, J. F. Science 1991, 251, 754.
20. Potvin, J. Y. ORSA J Comput 1993, 5, 328.
21. Budinich, M. Neural Comput 1996, 8, 416.
22. Johnson, D. S.; McGeoch, L. A. In Local Search in Combinatorial Optimization; Aarts, E.; Lenstra, J. K., Eds.; John Wiley & Sons: New York, 1997.
23. Yuille, A. Neural Comput 1990, 2, 1.
24. Stolorz, P. In Advances in Neural Information Processing Systems; Moody, J. E.; Hanson, S. J.; Lippmann, R. P., Eds.; Morgan Kaufmann: San Mateo, CA, 1992; p. 1026, Vol. 4.
25. Yuille, A. L.; Kosowsky, J. J. Neural Comput 1994, 6, 341.
26. Rose, K.; Gurewitz, E.; Fox, G. C. Phys Rev Lett 1990, 65, 945.
27. Simić, P. D. Neural Comput 1991, 3, 268.
28. Stolorz, P. Preprint, January 21, 1996.
29. Durbin, R.; Szeliski, R.; Yuille, A. Neural Comput 1989, 1, 348.
30. Vakhutinsky, A. I.; Golden, B. L. J. Heuristics 1995, 1, 67.
31. Simić, P. D. Network 1990, 1, 89.
32. Straub, J. E. In Recent Developments in Theoretical Studies of Proteins; Elber, R., Ed.; Adv Series Phys Chem; World Scientific: Singapore, 1996; p. 137, Vol. 7.
33. Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, M. P. Science 1983, 220, 671.
34. Piela, L.; Kostrowicki, J.; Scheraga, H. A. J Phys Chem 1989, 93, 3339.
35. Ma, J.; Straub, J. E. J Chem Phys 1994, 101, 533.

36. Dill, K. A.; Stigter, D. Adv Protein Chem 1995, 46, 59.
37. Jaynes, E. T. Phys Rev 1957, 106, 620.
38. Raghunathan, G.; Jernigan, R. L. Protein Sci 1997, 6, 2072.
39. Bahar, I.; Atilgan, A. R.; Erman, B. Fold Des 1997, 2, 173.
40. Bahar, I.; Wallqvist, A.; Covell, D. G.; Jernigan, R. L. Biochemistry 1998, 37, 1067.
41. Erman, B.; Dill, K. J Chem Phys 2000, 112, 1050.
42. Erman, B.; Dill, K. A. Phys Rev Lett, submitted.
43. Lau, K. F.; Dill, K. A. Macromolecules 1989, 22, 3986.
44. Dill, K. A.; Bromberg, S.; Yue, K.; Fiebig, K. M.; Yee, D. P.; Thomas, P. D.; Chan, H. S. Protein Sci 1995, 4, 561.
45. Thomas, P. D.; Dill, K. A. Proc Natl Acad Sci USA 1996, 93, 11628.
46. Rost, B.; Sander, C. J Mol Biol 1993, 232, 584.
47. Yue, K.; personal communication.
48. ELAN-PROT: ELAstic Net algorithm for PROTein structure prediction (http://www.dillgroup.ucsf.edu/elan/).
49. Second Critical Assessment of Fully Automated Structure Prediction (http://www.cs.bgu.ac.il/~dfischer/CAFASP2/).
50. Young, M. M.; Tang, N.; Hempel, J. C.; Oshiro, C. M.; Taylor, E. W.; Kuntz, I. D.; Gibson, B. W.; Dollinger, G. Proc Natl Acad Sci USA 2000, 97, 5802.