ALGORITHMIC APPROACH TO QUANTIFYING THE HYDROPHOBIC FORCE CONTRIBUTION IN PROTEIN FOLDING

R. BACKOFEN,* S. WILL,* J. BOND AND P. CLOTE*
Institut für Informatik, LMU München, Oettingenstraße 67, D-80538 München
Email: {backofen,clote,wills}@informatik.uni-muenchen.de

Though the electrostatic, ionic, van der Waals, Lennard-Jones, hydrogen bonding, and other forces play an important role in the energy function minimized at a protein's native state, it is widely believed that the *hydrophobic force* is the dominant term in protein folding. In this paper, we attempt to quantify the extent to which the hydrophobic force determines the positions of the backbone α-carbon atoms in PDB data, by applying Monte-Carlo and genetic algorithms to determine the predicted conformation with minimum energy, where only the hydrophobic force is considered (i.e. Dill's HP-model, and refinements using Woese's polar requirement). This is done by computing the root mean square deviation between the normalized distance matrix $D = (d_{i,j})$ ($d_{i,j}$ is normalized Euclidean distance between residues r_i and r_j) for PDB data with that obtained from the output of our algorithms. Our program was run on the database of ancient conserved regions drawn from GenBank 101 generously supplied by W. Gilbert's lab [1,2], as well as medium-sized proteins (E. Coli RecA, `2reb`, Erythrocruorin, `1eca`, and Actinidin `2act`). The root mean square deviation (RMSD) between distance matrices derived from the PDB data and from our program output is quite small, and by comparison with RMSD between PDB data and random coils, allows a quantification of the hydrophobic force contribution
Keywords: lattice, face-centered-cubic, hydrophobic force, automorphism

Introduction

Though not experimentally established, it is commonly believed that a protein's native state can be characterized as that conformation, for which the protein achieves a global free energy minimum. Molecular dynamics modeling, which simulates the conformation changes of a peptide by taking into account the electrostatic, ionic, van der Waals (dipole-dipole), Lennard-Jones, hydrogen-bonding, and other forces considered at the atomic level (for the atoms of the peptide, together with those of the solvent), can currently simulate around 10^{-7} seconds of the folding sequence. This is orders of magnitude less than the time required for a protein to fold (milliseconds to seconds). Moreover, certain studies [3,4] have shown that the energy function used in molecular dynamics is not fully correct, leading to rather different predictions. In summary, molecular dynamics cannot be used singly to determine a protein's native state from its amino acid sequence, because of uncertainties in the energy function and computational intractability due to simulation at the atomic level.

An alternate approach is to simulate the backbone of α-carbons of an n-residue protein as a self-avoiding walk, where n beads on a string occupy adjacent lattice sites. Such lattice models have been investigated by a number of researchers, including [5,6,7,8,9,10,11,12,13,14,15]. In particular, K. Dill [5] proposed the HP-model, where a residue is labeled either as hydrophobic (H) or polar (P), and the global energy

Research of R. Backofen, P. Clote and S. Will partially supported by the Volkswagen Foundation.

is minimized by a self-avoiding walk (excluded volume requirement), which maximizes the number of H-H unit distance contacts; i.e. conformation $(r_1, \ldots, r_n) \in \mathbf{Z}^2$ [resp. $\mathbf{Z}^3$] for 2-dimensional [resp. 3-dimensional] lattice such that $|r_i - r_{i+1}| = 1$ for $1 \leq i < n$, $r_i \neq r_j$ for $1 \leq i < j < n$, and $(r_1, \ldots, r_n)$ minimizes the contact energy

$$E = - \sum_{1 \leq i < j \leq n} B_{i,j} \Delta(r_i, r_j) \tag{1}$$

where $\Delta(r_i, r_j)$ is 1 if if $|r_i - r_j| = 1$, else 0, and $B_{i,j} = 1$ provided the i-th and j-th residue are both hydrophobic, else 0. The HP-model approximates the hydrophobic force, which is not really a force, but rather an aggregate tendency for nonpolar residues to minimize their contact with the solvent. The HP-model is conceptually simple, allows the incorporation of refinements (HPNX-model including electrostatic forces, etc.), and is appears to be computationally less intractable than that of molecular dynamics.[a] In 3 dimensions, the principal disadvantage of the HP-model is its degeneracy; i.e. a given HP-sequence might have distinct conformations having a maximum number of unit-distance H-H contacts.[b]

In this paper we attempt to quantify the contribution of the hydrophobic force in protein folding, using the HP-model (and its extension, using Woese's polar requirement) on a 3-dimensional face-centered-cubic lattice (FCC). The paper is organized as follows. In S1, we describe the genetic algorithms used, how to encode a random walk in the FCC lattice using a sequence of relative directions, and top-level pseudocode of our program. Though we tested Monte-Carlo (MC) with pivot moves, MC with local moves [11,12], a genetic algorithm (GA), a 3-dimensional version of Unger-Moult's [18] hybrid genetic algorithm (UM), and *local-to-global* versions thereof,[c] due to space constraints we report only results obtained with UG. In S2, we describe the general approach using automorphism groups in describing arbitrary lattices, which avoids the use of relative direction sequences. In S3, we describe our output from the data listed in the abstract.

Overall program structure

Relative direction sequences

In the 3-dimensional cubic lattice, each lattice point $p = (x, y, z)$ has 6 nearest neighbors. If p is the position of a monomer in a self-avoiding walk, then the walk can possibly be continued in one of 5 relative directions: *front* (F), *left* (L), *right*

[a] Note that determining the conformation which maximizes unit distance H-H contacts in the HP-model for 2- and 3-dimensional cubic lattices is NP-complete [16,17].

[b] Our algorithm actually uses normalized Woese polar requirement values, hence is unlikely to suffer the degeneracy problem of the HP-model.

[c] i.e. modify equation (1) to obtain $E_k = - \sum_{1 \leq i < j \leq} B_{i,j} \Delta_k(r_i, r_j)$ where $\Delta_k(r_i, r_j) = 1$ if if $|r_i - r_j| = 1$ and $|i - j| \leq k$, else 0. Now define a *neighborhood enlargment schedule* starting with $k = 3$ and gradually increasing k to $n - 1$.

(R), *up* (U), *down* (D), where *back* (B) is not admissible because of the excluded volume condition. The conformation of an n-mer can then be specified by a sequence of $n-1$ relative directions. Such a sequence can be considered as an *individual* or *chromosome* in a genetic algorithm, where a pointwise mutation involving the change of a relative direction causes a bending of the polymer. Following along the self-avoiding walk involves retaining a frame of reference, which is changed using rotation matrices. This is described in the next section.

In the 3-dimensional FCC lattice, each lattice point $p=(x,y,z)$ has 12 nearest neighbors, where 6 are arranged in a hexagon in the same plane, and 3 are arranged in the plane above and 3 in the plane below. These neighbors can be described as those points at unit distance from p, determined by the angles θ, ψ, where θ is the angle of rotation of the x-axis in the xy-plane, and ψ is the angle of rotation of the y-axis in the xz-plane. If p is the position of a monomer in a self-avoiding walk, then the walk can possibly be continued in one of 11 relative directions[d] given in the following table.

Direction	θ	ψ
N	0	0
NE	$\pi/3$	0
SE	$2\pi/3$	0
SW	$4\pi/3$	0
NW	$-\pi/3$	0

Direction	θ	ψ
WU	$\pi/2$	$\pi/3$
NNEU	$-\pi/6$	$\pi/3$
SSEU	$7\pi/6$	$\pi/3$
ED	$-\pi/2$	$-\pi/3$
NNWD	$\pi/6$	$-\pi/3$
SSWD	$5\pi/6$	$-\pi/3$

Pseudocode

INPUT: α-carbon coordinates and their hydrophobicity (H/P) from PDB data for a protein.[e]

OUTPUT: RMSD between the conformation C found to have minimal energy and the original PDB data (denoted as $RSMD_C$), along with percent contribution of the hydrophobic force. The latter is determined as the quotient of the number of random coils, whose RSMD is larger than $RSMD_C$, divided by the number of random coils.

1. Using combinatorial optimization (MC, GA, UM, or their *local-to-global* variant), determine the predicted conformation C as a self-avoiding walk in the FCC lattice.

[d] WU is the direction west up, and ED east down, etc. while S is not available, due to excluded volume. Despite the compass directions, these are relative directions with respect to the current frame of reference.

[e] For more than 30-40 residues, we contract the PDB data from a protein of length L to a representation of size 30, by taking the average position of α-carbons in successive $L/30$-size regions, and rather than using the HP-model, by taking the average Woese polar requirement value [19] in successive $L/30$-size regions, renormalized to values between 0 and 1 (polar requirement values are otherwise between 4.8 and 13.0). The contact energy (1) is then redefined to be $E=-\sum_{1\leq i<j\leq n} p_i \cdot p_j \cdot \Delta(r_i, r_j)$, where the p_i, p_j are normalized polar requirement values for the i-th and j-th region.

2. Compute $D_{\mathrm{HP}} = (d_{i,j})$, where $d_{i,j}$ is Euclidean distance from the i-th to j-th monomer in C.
3. Compute $D_{\mathrm{PDB}} = (e_{i,j})$, where $e_{i,j}$ is Euclidean distance from the i-th residue α-carbon to the j-th residue α-carbon, normalized (i.e. divided) by the *average* distance between successive α-carbons in the linear chain.
4. Compute $RSMD(D_{\mathrm{HP}}, D_{\mathrm{PDB}})$, defined by $\sqrt{\frac{\sum_{1 \leq i < j \leq n} (d_{i,j} - e_{i,j})^2}{\binom{n}{2}}}$.
5. Generate M random coils ($M \approx 200$), analogously compute their D_{RC},[f] and output the number of random coils, whose $D_{\mathrm{RC}} > D_{\mathrm{HP}}$, divided by then number M of random coils. We take this value to be the percent contribution of the hydrophobic force in protein folding.

Hybrid genetic algorithm

In [18], a hybrid genetic algorithm was described for folding on a 2-dimensional cubic lattice. We lift this algorithm to a 3-dimensional FCC lattice, using a general approach with automorphism groups (capable of handling arbitrary lattices), together with oct-trees for space management.

Given an input length n HP-sequence (or sequence of normalized Woese polar requirement values), we maintain a population of P many conformations ($P \approx 200$), as represented by a *chromosome*, or relative direction sequences of length $n - 1$. The population at time t is denoted $P(t)$. The fitness $F(c)$ of conformation c equals $-E(c)$, where the energy is given by equation (1), or its modification for polar requirements.

```
t = 0
initialize population P(t) of random coils
best = argmax  F(x) : x in P(t)
repeat
     t++
     pointwise mutation
     n=0
     while n < P
          select 2 chromosomes m,f
          produce child c by crossover of m,f
          ave = average( F(m), F(f) )
          if F(c) >= ave or random(0,1) < exp -( F(c)-ave )/T
               place c in next generation
               n++
     end while
     update best
```

[f] The graphical display of this distribution is similar to that of an extremal distribution, see appendix for an example

```
until convergence
```

For each chromosome, choose a site $1 \leq i \leq n-1$ and perform a pointwise mutation at site i with with probability p_m. Each chromosome x is selected for crossover according to its fitness (i.e. with probability $F(x)/\sum_{y \in P} F(y)$), using the roulette wheel technique. It should be noted that this algorithm is not a typical genetic algorithm, but rather a hybrid form which incorporates the Metropolis criterion. Our experiments indicate that UM is superior to GA, while Monte-Carlo with the local move set from [11,12] is actually superior to UM. This will be studied in detail in future work.

Methods

A chromosome of $n-1$ relative directions represents a conformation of length n. In extending Unger-Moult's algorithm from a 2-dimensional cubic lattice to 3-dimensional FCC, we show how the concept of relative move sequence can be generalized by pivot moves and lattice automorphisms. This provides a technique for extending the genetic algorithm to arbitrary lattices, where the user defines the lattice without needing to know the automorphisms. For clarity of exposition, we concentrate on the 3-dimensional cubic lattice, though our results are actually for the FCC lattice.

Relative moves in the cubic lattice

In the case of the 2-dimensional cubic lattice used by Unger-Moult, the relative directions or moves right, forward, and left correspond to rotations of $-90°$, $0°$ and $+90°$, respectively. In three dimensions, it is necessary to apply a base transformation with every relative move. In the following, we write $M \circ M'$ for the product of the matrices M and M', which represents the composition of the linear maps represented by M and M'.

A *relative move* m is an element of $\{F, L, R, U, D\}$. The *vector* v_m *assigned to a relative move* m is defined as

$$v_F = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad v_L = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad v_U = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad v_R = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \quad v_D = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

A sequence $w \in \{F, L, R, U, D\}^*$ is called a *relative move sequence.* Given such a sequence, we define $g_{basem}(w)$ to be

$$g_{basem}(w) = \begin{cases} I_3 & \text{if } w = \epsilon \\ g_{basem}(w') \circ B_m & \text{if } w = w'm \end{cases}$$

where I_3 is the 3×3 identity matrix and the matrices B_m for $m \in \{F, L, R, U, D\}$ are defined to be a ± 90 degree rotation turning the vector v_m into $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Hence, we know that B_m is defined as follows:

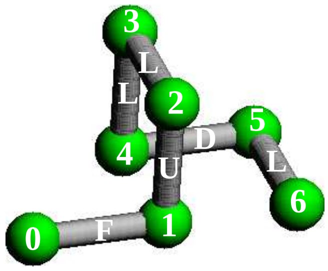
$$B_F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B_L = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B_U = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad B_R = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B_D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Given sequence w, we define $con(w)$ to the conformation c of length $|w|+1$ with $c[0] = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$, and $\forall 1 \leq i \leq |w| : (c[i] = c[i-1] + g_{basem}(w_1 \dots w_{i-1}) \circ v_{w_i})$

Example 1 Let $w = FULLDL$. Then

$$
\begin{array}{llllll}
c[0] = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & g_{basem}(\epsilon) & = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & c[1] = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & g_{basem}(\mathrm{F}) & = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
c[2] = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} & g_{basem}(\mathrm{FU}) & = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} & c[3] = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} & g_{basem}(\mathrm{FUL}) & = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \\
c[4] = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} & g_{basem}(\mathrm{FULL}) & = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix} & c[5] = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} & g_{basem}(\mathrm{FULLD}) & = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \\
c[6] = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} & g_{basem}(\mathrm{FULLDL}) & = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} & & &
\end{array}
$$

The resulting conformation c is



Note that although the absolute move $c[5] - c[4]$ is $(1, 0, 0)$, the following relative move L corresponds (in absolute coordinates) to a right turn. Hence, the relative moves are *not* maps that are applied to the previous absolute move. The next proposition states that a move always corresponds to $(1, 0, 0)$ in the base defined by this move. □

Proposition 2 *Let $\epsilon \neq w = w_1 \dots w_h$, and let $c = con(w)$. Then*

$$c[h] - c[h-1] = g_{basem}(w) \circ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Proof. By Induction. For $|w| = 1$, this is given by the definition of the B_m-matrices. For the induction step, let $w = w'm$ be a sequence of length $h+1$, and let $c = con(w)$. By definition of B_m, we know that $v_m = B_m \circ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Then

$$
\begin{aligned}
c[h+1] - c[h] = g_{basem}(w') \circ v_m &= g_{basem}(w') \circ (B_m \circ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}) \\
&= g_{basem}(w) \circ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}
\end{aligned}
$$

□

We can now specify the effects of changing a single relative move in a sequence of relative moves. This corresponds to the pointwise mutation of a chromosome in the genetic algorithm. We will show that this is nothing else than the application of a rotation.

Theorem 3 *Let $w = mu$ be some relative move sequence, and let $w' = m'u$. Let $c = con(w)$ and $c' = con(w')$. Then there exists a rotation M of $\mathbb{Z}^3$ such that for all $0 \leq i \leq |w|$ we have $c[i] = M \circ c'[i]$.*

Proof. We will show that for all $w = mu$ and $w' = m'u$, there is an M such that

$$g_{basem}(w) = M \circ g_{basem}(w'). \tag{2}$$

From this, the claim follows immediately by induction since $c[0] = M \circ \left(\begin{smallmatrix}0\\0\\0\end{smallmatrix}\right) = c'[0]$, and for every $i \geq 1$ we have

$$\begin{aligned}
c[i] &= c[i-1] + g_{basem}(w_1 \dots w_{i-1}) \circ \left(\begin{smallmatrix}1\\0\\0\end{smallmatrix}\right) \\
&= M \circ c'[i-1] + (M \circ g_{basem}(w'_1 \dots w'_{i-1})) \circ \left(\begin{smallmatrix}1\\0\\0\end{smallmatrix}\right) && \text{(Ind. Hyp)} \\
&= M \circ c'[i-1] + M \circ (c'[i] - c'[i-1]) && \text{(Prop. 2)} \\
&= M \circ \big(c'[i-1] + (c'[i] - c'[i-1])\big) && \text{(Linearity)} \\
&= M \circ c'[i].
\end{aligned}$$

We prove the existence of M as required by claim (2) by induction on the length of w. For $|w| = 1$ let $w = m$ and $w' = m'$, we set $M = B_m \circ B_{m'}^{-1}$. Then

$$\begin{aligned}
g_{basem}(m) = B_m &= B_m \circ (B_{m'}^{-1} \circ B_{m'}) \\
&= (B_m \circ B_{m'}^{-1}) \circ B_{m'} = M \circ B_{m'} = M \circ g_{basem}(m')
\end{aligned}$$

For the induction step, let $|w|$ be a word of length $h+1$, where $h \geq 1$. Then w and w' satisfy $|w| = |w'| = h+1$ and $w_{h+1} = w'_{h+1}$. Hence,

$$\begin{aligned}
g_{basem}(w_1 \dots w_h w_{h+1}) &= g_{basem}(w_1 \dots w_h) \circ B_{w_{h+1}} \\
&= M \circ g_{basem}(w'_1 \dots w'_h) \circ B_{w'_{h+1}} && \text{(Ind. Hyp.)} \\
&= M \circ g_{basem}(w'_1 \dots w'_h w'_{h+1})
\end{aligned}$$

□

Corollary 4 (Mutation) *Let $w = w_1 m w_2$ and $w' = w_1 m' w_2$ be two relative move sequences which differ in one position. Let $c = con(w)$ and $c' = con(w')$ be the corresponding structures. Then there is a rotation M of $\mathbb{Z}^3$ such that*

- *for all $1 \leq i \leq |w_1| : (c[i] = c'[i])$, and*
- *for all $|w_1| + 1 \leq i \leq |w| : (c[i] = M \circ c'[i])$.*

Thus, a mutation in the genetic algorithm corresponds to a rotation of the remaining part of the conformation. This concept is already known in the literature on Monte-Carlo methods for self-avoiding walks, where this kind of mutation is called a *pivot move* [20]. Actually, pivot moves are more than rotations, since also reflections are allowed. Rotations are matrices with determinant 1, whereas reflections have determinant -1. Thus, mutations (and pivot moves) correspond to automorphisms, mapping the lattice to itself. This concept can be defined for arbitrary lattices.

Lattices and lattice automorphisms

Definition 5 (Lattice) *Let $\vec{v_1}, \ldots, \vec{v_m}$ be vectors in $\mathbb{R}^n$. The* lattice *generated by these vectors is the smallest set $L \subset \mathbb{R}^n$ such that*

1. $\{\vec{v_1}, \ldots, \vec{v_m}\} \subset L$,
2. *if $\vec{u} \in L$ and $\vec{v} \in L$, then $\vec{u} + \vec{v}$ and $\vec{u} - \vec{v}$ are also in L.*

The vectors $\vec{v_1}, \ldots, \vec{v_m}$ are called the basis *of L.*

Note that a lattice is *not* a vector space, since one allows only linear combinations with integral coefficients. Thus, a lattice can have a basis of vectors which are *not* linearly independent. Now we want to define the automorphisms corresponding to pivot moves. A linear map b is an *isometry* if it is distance preserving (i.e., if for every $\vec{v}$ we have $\|\vec{v}\| = \|B \circ \vec{v}\|$, where $\|\cdot\|$ is Euclidean distance. Since we are dealing with the Euclidean space $\mathbb{R}^n$, an isometry is given by an orthogonal matrix.

Definition 6 (Lattice Automorphism) *Let L be a lattice. A* lattice automorphism *B is an isometry of $\mathbb{R}^n$ with the property that $L = \{B \circ \vec{v} \mid \vec{v} \in L\}$.*

For many reasons, it is simpler to use the integral representation of a lattice L. Let $\vec{v_1} = \begin{pmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{1n} \end{pmatrix}$ $\vec{v_1} = \begin{pmatrix} v_{21} \\ v_{22} \\ \vdots \\ v_{2n} \end{pmatrix}$ $\ldots$ $\vec{v_m} = \begin{pmatrix} v_{m1} \\ v_{m2} \\ \vdots \\ v_{mn} \end{pmatrix}$ be the basis of the lattice L. Let M be the matrix

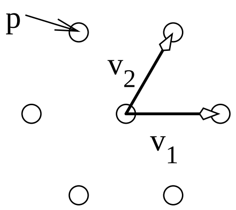
$$\begin{pmatrix} v_{11} & v_{21} & \cdots & v_{m1} \\ v_{12} & v_{22} & \cdots & v_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{1n} & v_{2n} & \cdots & v_{mn} \end{pmatrix}$$

Then M is called the *generator matrix* of L.

Proposition 7 (Integral representation) *Let $\vec{v_1}, \ldots, \vec{v_m}$ be the basis of the lattice L, and let M be the corresponding generator matrix. Then $L = \{M \circ \zeta \mid \zeta \in \mathbb{Z}^m\}$*

Thus, we can represent every lattice by the m-dimensional cubic lattice $\mathbb{Z}^m$ (using the generator matrix to translate from the integral representation into real coordinates). Note that we might have to change the dimension (i.e., $m \neq n$; usually, we have $m \geq n$).

Example 8 Consider the two-dimensional, hexagonal lattice A_2. The center $(0,0)$ and the six nearest points of A_2 are as follows:



Now $v_1 = (1,0)$ and $v_2 = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ is a basis of f, and the generator matrix is $M_{A_2} = \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix}$ The point $p = (\frac{-1}{2}, \frac{\sqrt{3}}{2})$ shown above has the integral representation $(-1,1)$. $\square$

Efficient implementation of pivot moves

The main time consuming operation in the genetic algorithm as described in Section is the application of the pivot moves. If ones uses arbitrary lattices and real coordinates, then the cost for testing self-avoidingness after the application of a pivot move is $O(len)$, and to calculate the energy of the new conformation is $O(len^2)$ (where len is the length of the input sequence).

We have reduced this time complexity by using the integral representation of lattice coordinates. Chromosomes (which represent conformations of the input sequence) are stored in arrays of lattice coordinates $CONF[len]$. Additionally, we maintain for every conformation a corresponding array $COORD[len{+}1][len{+}1][len{+}1]$.[g] $COORDS$ maps lattice coordinates to 0, if the corresponding position is not occupied by some monomer, and i, if the position is occupied by the i^{th} monomer. Formally, we get

$$COORD(x,y,z) = \begin{cases} i \text{ if } CONF[i] = (x', y', z') \text{ and } x = x' \bmod len + 1, \\ \quad y = y' \bmod len + 1 \text{ and } z = z' \bmod len + 1 \\ 0 \text{ else} \end{cases}$$

Using mod $len + 1$ guarantees that all monomers will be stored in $COORDS$, even when they will achieve coordinates (by the application of a pivot move) that are greater than $len + 1$. This technique is the reason that we have to use $len + 1$ in the definition of $COORDS$. If we would have used len, then a monomer with position $(1,0,0)$ would have a contact (in $COORDS$) with a monomer with position $(len,0,0)$ (since $(len,0,0)$ would then have been stored under $(0,0,0)$). With $len{+}1$, this cannot happen, since the maximal extension in any coordinate is len.

Now we can describe the application of a pivot move. We assume that A is an integral matrix representing a pivot move.

```
function pivot(int len,matrix A,int CONF[len], int COORDS[len+1][len+1][len+1])
  site = random(1..len)
  for i=site+1 to len do
      (Xold,Yold,Zold) = CONF[i]
      (X,Y,Z) = A(CONF[i] - CONF[site]) + CONF[site]
      if COORDS[X  mod len+1][Y  mod len+1][Z  mod len+1] = 0
         or COORDS[X  mod len+1][Y  mod len+1][Z  mod len+1] >= i
      then
```

[g] the description here is for the FCC and cubic lattice, where the integral representation uses coordinates of $\mathbb{Z}^3$

```
        C[i] = (X,Y,Z)
        COORDS[Xold  mod len+1][Yold  mod len+1][Zold  mod len+1] = 0
        COORDS[X  mod len+1][Y  mod len+1][Z  mod len+1] = i
     else
        RETURN ``not self-avoiding''
     endif
 endfor
```

Clearly, one has to undo the effects of applying the pivot function in the case it returns "not self-avoiding". One can do this without the need of copying $COORD$ before applying pivot. After the application of the function pivot, the energy of the new conformation has to be calculated. Again, using $COORD$, this can be done with one pass through the conformation (i.e., in $O(n)$).

Since $COORD$ is very space consuming on the one hand, and sparse on the other, we use an oct-tree representation for $COORD$. Thus, energy calculation is $O(n \ln(n))$ instead of $O(n)$.

The only part that is missing is that we need an integral representation A of an automorphism B of a lattice L (as used by the function pivot). As described in [21], this can easily be generated as follows. Let M be the generator matrix of L. Then A is an automorphism of the integral representation of L if 1.) A is integral, and 2.) there is an orthogonal matrix B such that $M \circ A = B \circ M$. Our program can, in a preprocessing step, calculated all possible automorphisms (in integral representation) from the generator matrix.

References

1. W. Gilbert. Personal communication, Jan. 1999.
2. S.J. de Souza, M. Long, R.J. Klein, S. Roy, S. Lin, and W. Gilbert. Toward a resolution of the introns early/late debate: only phase zero introns are correlated with the structure of ancient proteins. *Proc. Natl. Acad. Sci. U.S.A.*, 95(9):5094–9, 1998.
3. M. Teeter. Water-protein interactions: Theory and experiment. *Annu. Rev. Biophys. Biophys. Chem.*, 20:577–600, 1991.
4. M. Teeter. An empirical examination of potential energy minimization using the well-determined structure of the protein crambin. *Journal of the American Chemical Society*, 108:7163–7172, 1986.
5. In B.T. Nall and K.A. Dill, editors, *Conformations and Forces in Protein Folding*. American Association for the Advancement of Science, 1991. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.
6. S. Govindarajan and R.A. Goldstein. Searching for foldable protein structures using optimized energy functions. *Biopolymers*, 36:43–51, 1995.
7. S. Govindarajan and R.A. Goldstein. Why are some protein structures so common? *Proc. Natl. Acad. Sci. USA*, 93:3341–3345, 1996. Biophysics.

8. E. Bornberg-Bauer. How are model protein structures distributed in sequence space? *Biophys. J.*, 73(5):2393–403, 1997.
9. M. Vieth, A. Kolinski, III C.L. Brooks, and J. Skolnick. Prediction of the quaternary structure of coiled coils. Application to mutants of the GCN4 leucine zipper. *Journal of Molecular Biology*, 251:448–467, 1995.
10. H. Li, R. Helling, C. Tang, and N. Wingreen. Emergence of preferred structures in a simple model of protein folding. *Science*, 273:666–669, 1996.
11. A. Šali, E. Shakhnovich, and M. Karplus. Kinetics of protein folding: A lattice model study of the requirements for folding to the native state. *J. Molec. Biol.*, 235:1614–1636, 1994.
12. A. Šali, E. Shakhnovich, and M. Karplus. How does a protein fold? *Nature*, 369:248–251, 19 May 1994. Letters to Nature.
13. K. Yue and K. Dill. *Proc. Natl. Acad. Sci. U.S.A.*, 92:146, 1995.
14. L.M. Gregoret and F.E. Cohen. *J. Mol. Biol.*, 219(1):109–22, 1991.
15. K. Yue, K.M. Fiebig, P.D. Thomas, H.S. Chan, E.I. Shakhnovich, and K.A. Dill. *Proc. Natl. Acad. Sci. U.S.A.*, 92(1):325–9, 1995.
16. P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):523–466, 1998.
17. B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. In S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology (RECOMB98)*, pages 30–39. ACM Press, 1998.
18. R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231:75–81, 1993.
19. C.R. Woese, D.H. Durge, S.A. Dugre, M. Condo, and W.C. Saxinger. On the fundamental nature and evolution of the genetic code. *Cold Spring Harbor Symp. Qant. Biol.*, 31:723–736, 1966.
20. N. Madras and G. Slade. *The Self-Avoiding Walk.* Birkhäuser, Boston, 1993.
21. J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, NY, 1996. 3rd edition.
22. H.S. Chan and K. Dill. Personal communication, 1998.

Acknowledgements

We would like to thank W. Gilbert for generously supplying the database of ancient conserved regions [1,2], to E. Bornberg-Bauer and T. Santner for discussions, and to H.S. Chan and K. Dill for HP-sequence data for non-degenerate 2-dimensional lattice polymers [22], the latter used in calibrating our genetic algorithms, and in comparing Monte-Carlo (with local moves) against GA and UM algorithms (to appear in the full paper). The present work arose from a student project with J. Bond, R. Fiori, M. Ollenschlager, A. Segrich under the direction of P. Clote, involving a quadratic time algorithm for testing the excluded volume

	Run #1			Run #2			Run #3		
Code	Energy	RMSD	%	Energy	RMSD	%	Energy	RMSD	%
aat	-12.013	0.924	100.00	-11.805	1.040	99.95	-12.004	1.028	99.95
acidamy	-10.318	1.003	99.94	-10.828	1.053	99.82	-10.454	0.918	100.00
acyl	-11.396	0.931	100.00	-11.328	1.069	99.94	-11.532	1.043	99.97
adea	-13.671	0.880	100.00	-13.625	1.003	99.85	-13.803	0.984	99.90
adh	-10.942	1.096	99.87	-15.966	1.086	99.23	-15.637	1.199	97.65
aldehy	-12.575	1.059	99.96	-12.728	1.188	99.52	-12.376	1.094	99.90
aldol	-11.987	1.175	99.84	-12.488	1.117	99.94	-12.126	1.089	99.97
alk	-12.501	0.928	100.00	-12.338	0.930	100.00	-12.444	0.996	100.00
asp	-10.592	0.880	99.99	-10.551	1.004	99.93	-10.719	0.957	99.98
csyn	-10.849	1.083	99.86	-10.280	1.026	99.96	-10.402	1.086	99.86
cusod	-16.928	0.990	99.98	-17.237	0.956	99.99	-16.722	0.983	99.98
dhfr	-12.511	1.109	98.64	-13.141	1.109	98.64	-12.696	0.905	99.94
dihydro	-11.248	0.997	99.95	-10.926	1.039	99.89	-11.192	1.028	99.89
eftu	-14.716	1.017	100.00	-14.041	0.962	100.00	-14.516	1.114	99.91
enolase	-13.934	0.995	99.92	-13.918	0.956	99.97	-11.839	0.991	99.92
g6pd	-14.915	1.085	99.95	-14.766	0.913	100.00	-14.940	1.041	100.00
glyphos	-12.084	0.958	99.99	-13.156	0.899	100.00	-11.785	1.009	99.98
gra	-10.734	0.935	99.99	-10.685	1.055	99.81	-10.890	0.969	99.97
hemo	-10.215	1.383	88.32	-9.433	1.520	75.77	-9.963	1.434	84.65
highpi	-12.695	1.052	99.79	-13.382	0.962	99.94	-12.930	1.045	99.79
hsp70	-13.408	0.861	99.99	-13.321	1.012	99.94	-13.489	0.981	99.98
lyso	-16.714	1.443	88.17	-16.695	1.421	89.28	-16.644	1.340	93.38
pgk	-14.586	1.161	99.20	-13.541	1.154	99.37	-13.970	1.086	99.78
pk	-12.914	1.135	99.27	-13.210	1.063	99.66	-12.617	1.088	99.54
thio	-15.698	1.074	99.14	-15.713	1.115	98.26	-15.771	1.106	98.26
xyla	-12.590	1.112	99.67	-13.101	0.965	99.98	-12.497	1.004	99.93

Table 1. Results for the Gilbert data

condition. The linear time test for excluded volume presented here is due to R. Backofen and S. Will and uses automorphism groups for handling general lattices, as well as oct-trees for careful management of space requirements.

Results

The tables 1 and 2 summarizes the output for the the database of ancient conserved regions drawn from GenBank 101 from W. Gilbert's laboratory [1,2]. For every protein, they show three runs of our algorithm. For each run, we have listed the energy of the best conformation found, the RMSD between this conformation and the original conformation, and how many percent of random generated conformations have an RMSD greater than the RMSD of the best conformation found (as described in Step 4 on page).

Code	Run #1 Energy	RMSD	%	Run #2 Energy	RMSD	%	Run #3 Energy	RMSD	%
1eca	-16.554	2.122	48.15	-15.612	2.061	52.25	-16.552	2.119	49.49
2act	-13.846	1.123	99.85	-12.974	1.039	99.98	-13.545	1.126	99.85
2reb	-12.055	1.062	99.25	-12.266	1.167	97.42	-12.524	1.146	97.79

Table 2. Results for real proteins

RMSD Distributions

We show here the distribution of the RMSD between randomly generated conformations and the original protein conformation, for 2reb: