Approximation Algorithms for Protein Folding Prediction

Giancarlo Mauri* Giulio Pavesi† Antonio Piccolboni‡

Abstract

We present a new polynomial-time algorithm for the protein folding problem in the two-dimensional HP model introduced by Dill [1], which has been recently proved to be NP-hard [2]. Our algorithm guarantees a performance ratio (i.e., the ratio between the energy of the solution found by the algorithm and the optimal one) of 1/4, equalling the two best polynomial-time performance guaranteed algorithms for this problem [3]. However, experimental results on a large set of random instances have shown an average performance ratio for our algorithm of 0.67, versus 0.55 and 0.48 for the other two.

1 Introduction

Proteins are polymer chains of amino acid residues of twenty different kinds. Under specific environmental conditions (i.e. inside living organisms), they fold to form a *unique* geometric pattern, known as *native state*, that determines their macroscopic properties, behavior and function. Usually, possible protein conformations are analyzed in terms of their *free energy*. According to the *Thermodynamical Hypothesis*, the native structure of a protein is the one corresponding to a global minimum of its free energy. The form of the energy function changes according to the *model* adopted.

2 The HP Model

One of the most successful and best-studied abstract models is the *two-dimensional hydrophobic-hydrophilic model*, or HP model, proposed by Dill. Basically, the amino acid residues can be divided in two classes: the hydrophobic, i.e. non-polar, and hydrophilic, i.e. polar. Experiments have shown that during the folding process the hydrophobic residues tend to interact with each other, forming the core of the final structure, shielded from the environment by the hydrophilic ones. Therefore, the protein instance can be reduced to a *binary* sequence of H's (meaning hydrophobic) and P's (meaning polar, or hydrophilic). Furthermore, the conformational space is *discretized* into a square lattice. Thus, since two residues cannot occupy the same space position, the possible conformations for the protein in this model are *self-avoiding walks* (SAWs) on a two-dimensional grid. From now on, we will refer to a pair of residues as *in contact* if they are adjacent on the lattice but not in the sequence. The free energy function for this model is based on the number of hydrophobic residues that are in contact on the lattice. Every H–H contact on the lattice brings a free energy of -1. Every other contact has a free energy of 0. Thus, the problem is to find the conformation that maximizes the number of contacts between H's. This problem has been proved to be *NP-hard* [2].

3 The Algorithm

According to the model, a protein instance can be represented by a string $s = s_0 \dots s_n$, where $s_i \epsilon \{\mathrm{H}, \mathrm{P}\}$. Our algorithm is based on the following steps:

1. Define an ambiguous grammar that generates all the possible instances of the problem.
2. Define a relation between the derivations of the grammar and a subset of all the possible SAWs, where to every production of a derivation recursively corresponds a spatial position of the terminal symbols generated by the production itself.
3. Assign to every production of the grammar an appropriate *score*, representing (a lower bound to) the number of contacts between H's generated by the spatial position of the symbols associated to the production in the SAW corresponding to the parse tree.
4. Apply a parsing algorithm to find the tree with the highest score (computed as the sum of the scores of the productions of the tree), that is, the tree corresponding to the SAW with minimal energy in the subset generated by the grammar.

The first grammar we defined for our algorithm has three terminal symbols (H, P and a dummy symbol U), three non-terminal symbols (the source symbol R, L and S), and 115 productions. An example of the layout of the terminal symbols associated to each production can be seen in Figure 1. In the example, the production

*Dept. of Computer Science, University of Milan, Italy. E-mail: mauri@dsi.unimi.it

†jeevez@ginevra.usr.dsi.unimi.it

‡piccolbo@dsi.unimi.it

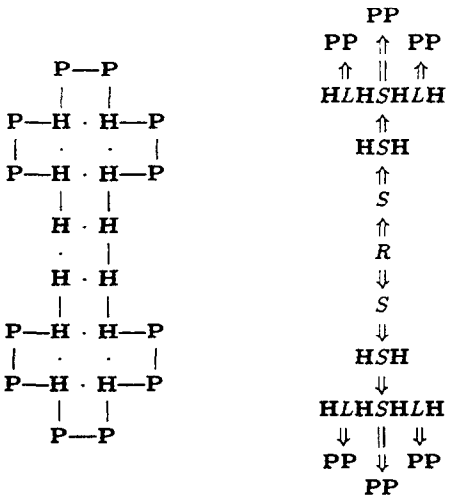

Figure 1: Structure generated by the algorithm for the sequence HHPPHPPHPPHHHHPPHPPHPPHH and corresponding parse tree with score 11. Contacts between H's are shown by dots (·).

$S \rightarrow HSH$ has score one, $S \rightarrow HLHSHLH$ score four, and so on. The parsing algorithm is based on the algorithm that computes the viterbi parse of a string generated by a stochastic grammar proposed by Stolcke [4]. It preserves its worst case time ($O(n^3)$) and space ($O(n^2)$) complexity.

4 Performance Analysis

Let h_e be the number of H's in even position in a given sequence s; h_o the number of H's in odd position; $h^* = \min(h_e, h_o)$. We also define $OPT(s)$ as the free energy of the optimal conformation for a given sequence s. It can be easily proved that $OPT(s) \geq -2h^* - 2$. Now, let $\mathcal{R}_G$ and $\mathcal{R}_G^\infty$ be the absolute and the asymptotic performance ratios of our algorithm.

LEMMA 4.1. *Given a sequence s, there always exists a structure for s, corresponding to a parse tree, that brings* $\lceil \frac{h^*+1}{2} \rceil$ *contacts.*

THEOREM 4.1.

$$\mathcal{R}_G \geq \frac{-\lceil \frac{h^*+1}{2} \rceil}{-2h^* - 2} = \frac{1}{4} \tag{4.1}$$

$$\mathcal{R}_G^\infty \geq \frac{1}{4} \tag{4.2}$$

The lower bounds for the performance ratios of our algorithm equal the performance ratios of the best two algorithms known [3].

Algorithm	$\mathcal{B}$	$\mathcal{C}$	CFG
Time Complexity	$O(n)$	$O(n^2)$	$O(n^3)$
Guaranteed Performance Ratios	1/4	1/4	1/4
Average Performance Ratio	0.48	0.55	0.67
Worst Case Performance Ratio Found	0.25	0.33	0.375

Figure 2: Guaranteed and experimental performance ratios of algorithms $\mathcal{B}$ and $\mathcal{C}$ [3], and our algorithm (CFG), on a large number of random instances, with different values of $P_H = \Pr[s_i = H], \forall i \in [0, n]$.

5 Conclusions

We have proved that our algorithm has the same performance guarantee as the best known algorithms, but experimental results (shown in Fig. 2) suggest that it is even better in an average case sense. Moreover, whereas the 1/4 bound is tight for the best known algorithms, the tightness of the same performance bound for our algorithm is still an open problem. In fact, theorem 4.1, based on lemma 4.1, simply guarantees that for each instance s there always exists, among those that can be generated by the algorithm, a structure whose energy gives performance ratios of 1/4, but not that this structure is the one actually generated, that is, the one with lowest energy. This fact, together with the encouraging experimental results, leads us to the conjecture that a tight bound to the performance of our algorithm (or of an improvement of it, based on larger grammars) could be in fact the experimental one, that is 3/8.

References

[1] K. A. Dill, *Dominant forces in protein folding.* Biochemistry, 24:1501, 1985.

[2] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, M. Yannakakis, *On the Complexity of Protein Folding.* Proc. of RECOMB '98.

[3] W. E. Hart, S. C. Istrail, *Fast Protein Folding in the Hydrophobic-Hydrophilic Model Within Three-eights of Optimal.* Journal of computational biology, spring 1996.

[4] A. Stolcke, *An Efficient Probabilistic Context-Free Parsing Algorithm That Computes Prefix Probabilities.* Computational Linguistics, 21(2), 165–201, 1995.