A PAC-Learning Algorithm for Conformation Rules and its Experiments

Osamu Maruyama [1] Erika Tateishi [1] Emiko Furuichi [2]
maruyama@ims.u-tokyo.ac.jp erika@ims.u-tokyo.ac.jp emiko@grt.kyushu-u.ac.jp
Satoru Kuhara [3] Satoru Miyano [1]
kuhara@grt.kyushu-u.ac.jp miyano@ims.u-tokyo.ac.jp

[1] Human Genome Center, Institute of Medical Science, University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108, Japan
[2] Fukuoka Women's Junior College, Gojo, Dazaifu, Fukuoka 818-01, Japan
[3] Graduate School of Genetic Resources Technology, Kyushu University
Hakozaki, Higashi-ku, Fukuoka 812, Japan

1 Introduction

Computational methods for protein conformation have been extensively developed for searching minimal free-energy conformations. A recursive method is developed to identify a large number of low energy conformations and genetic algorithms are also applied to this problem. Another interesting heuristic method is the hydrophobic zipper method in [1, 2]. Based on the fact many hydrophobic contacts are topologically local, the hydrophobic zipper method randomly selects hydrophobic contacts among neighbors in a sequence and zips up other hydrophobic contacts.

Inspired by this hydrophobic zipper method, but apart from the free-energy minimization problem, we define a conformation rule as a rewriting rule of hypergraphs. Then we develop a PAC-learning algorithm for conformation rules and present some experimental results on amino acid sequences of proteins.

2 PAC-Learning of Conformation Rules

A protein P with a tertiary structure $(p_1, A_1), \cdots, (p_n, A_n)$, where $p_i = (x_i, y_i, z_i)$ is the position of the amino acid residue A_i for $1 \leq i \leq n$, is loosely represented by a node-labeled hypergraph $G = (V, F, \varphi)$ in the following way: The node set is $V = \{1, \cdots, n\}$, where the number i corresponds to the position of the ith amino acid residue. The nodes are labeled with an alphabet Δ of "colors" by a mapping φ. It is often used to classify the amino acid residues into several categories (e.g., hydrophobicity). φ and Δ represent such a classification of amino acid residues. A hyperedge e in F describes that the nodes in e are within some distance. We assume that $\{i, i+1\}$ is in F for $1 \leq i \leq n-1$. Thus there are many variations for representing the structure of a protein by a hypergraph.

A *bundle rule* is a pair $\rho = (B, U)$ of a hypergraph $B = (V, F, \psi)$ and a subset U of V such that $|U| \geq 2$, $U \notin F$ and $e \cap U \neq \emptyset$ for any hyperedge e in F. This bundle rule creates a new hyperedge U if the neighborhood of U is in the form of B.

A *conformation unit* is a finite set $\gamma = \{(B_1, U_1), \ldots, (B_t, U_t)\}$ of bundle rules and a *conformation rule* is defined as a sequence $\sigma = (\gamma_1, \ldots, \gamma_m)$ of conformation units. A conformation rule is applied to a sequence from local toward global as shown in Fig. 1, and finally produces a hypergraph.

We have shown that some class of conformation rules is polynomial-time PAC-learnable in the sense of [3] and have developed a PAC-learning algorithm which produces a conformation rule from a collection of sequences.

```
Input: a conformation rule (γ_1, ..., γ_m) and s = x_1 ··· x_n in Δ^+
Output a hyper graph H_s = (V_s, F_s, ψ_s)
procedure Conform((γ_1, ..., γ_m), s)
begin
  V_s := {1, ..., n};
  let ψ_s be a mapping defined by ψ_s(i) = x_i for 1 ≤ i ≤ n;
  F := {{i, i+1} | 1 ≤ i ≤ n−1};
  τ := min{n, m};
  for ℓ ← 1 to τ do
  begin
    w := ℓ + 2; /* w is the window size */
    TEMP := ∅;
    foreach i : 1 ≤ i ≤ n − w + 1 do
    begin
      j := i + w − 1;
      foreach e : e ⊆ {i, ..., j} with |e| ≤ k do
      begin
        F̃ := ⋃_{l∈e} N_H(l), where H = (V_s, F, ψ_s);
        Ṽ := {u | u ∈ e' for some e' ∈ F̃};
        ψ̃ := ψ_s|_Ṽ; /* the restriction of ψ_s to Ṽ */
        if B̃ = (H̃, e) ≈ B for some B in γ_ℓ, where H̃ = (Ṽ, F̃, ψ̃);
            then add a hyperedge e to TEMP;
      end;
    end;
    F := F ∪ TEMP;
  end;
  F_s := F;
end
```

Fig. 1: Conformation algorithm

3 Method of Experiments

We have implemented the PAC-learning algorithm with Common Lisp and chosen 153 proteins from PDB for our experiments. Each protein file is expressed as a distance matrix $\mathcal{M}$ of positions of amino acid residues, where the (i,j)-entry of $\mathcal{M}$ is 1 if the distance between the ith and jth amino acid residues is at most 6Å and 0 otherwise.

The size of a hyperedge is restricted to be two, three and four because of the difficulty arising from time and space complexity. The alphabet Δ is set to be the collection of the three symbols, H (hydrophobic), P (hydrophilic) and N (neutral).

The first step is to learn a conformation rule from 5$\sim$20 proteins. The second step is to apply the conformation rule to a sequence for prediction. The comparison between the prediction and the original structure shows that, for some part of a sequence, the corresponding structure is correctly predicted.

References

[1] Dill, K.A., Fiebig, K.M. and Chan, H.S., Cooperatively protein-folding kinetics, *Proc. National Academy of Science, U.S.A.* **90**, 1942–1946, 1993.

[2] Hart, W.E. and Istrail, S.C., Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal, *J. Computational Biology* **3**, No. 1, 53–96, 1996.

[3] Natarajan, B.K., *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, 1991.