COMP564: HP lattice models

Jérôme Waldispühl
School of Computer Science, McGill

(slides include graphics from S. Will)
trade-off: choose between

- models, that closely resembles proteins structure
  **BUT** no hope of (algorithmically) finding the native structure

- models, that crudely resembles proteins structure
  **BUT** we can find the native structure
Definition

- only one force: hydrophobic force
- native structure: maximal number of HH-contacts
- used lattices:
  - cubic lattice
  - FCC

Alphabet:
- H = Hydrophob
- P = Polar (hydrophil)

(Lau & Dill, 1989)
Lattice models

FIGURE 1.2: Examples of crystal lattices: (a) cubic, (b) diamond, (c) cubic with planar diagonals, (d) hexagonal, (e) triangular and (f) face-centered-cubic.
FIGURE 1.3: Illustrations of conformations for: (a) the standard HP model on the square lattice, (b) the HP model with side chains on the square lattice, and (c) the HP tangent spheres model with side chains. Black denotes a hydrophobic amino acid, white denotes a hydrophilic amino acid, and gray denotes a backbone element.
Structure in HP model

Sequence HPPHPHPH
Complexity

- NP-Hard on 3D cubic lattice (Berger & Leighton, 1998)
- NP-Hard on 2D square lattice (Crescenzi et al., 1998)

Self-avoiding path on the Square lattice (without symmetry):
Optimization problem (Definition)

**Input**: A protein sequence represented as a binary chain of hydrophilic/hydrophobic residues \((H=1, P=0)\).

**Objective**: Find a folding of the string that forms a self-avoiding walk on the lattice and maximizes the number of long range contacts between hydrophobic residues.

**FIGURE 1.4**: An optimal conformation for the string 0010100001011010 on the 2D square lattice. This conformation has four contacts.
Approximation on square lattice

Notation:
- $E[s]$: number of hydrophobic (1's) in even positions,
- $O[s]$: number of hydrophobic (1's) in odd positions,
- $X(s) = \min\{E[s], O[s]\}$

Upper bound:
odd-1's can only have contacts with even-1's (and vice-versa).
Each (non-terminal) 1 in $s$ can have at most 2 contacts.

Upper bound: $2 \cdot X(s) + 2$

Algorithm:
1. find $p$ s.t. at least one half of odd-1's are on one side and one half of even-1's are on the other side.
2. Embed the odd substring s.t. odd-1's have the same y-coordinate, do the same operation for the even string and fold them together.

(Hart & Istrael, 1995)
Approximation on square lattice

FIGURE 1.5: Illustration of a conformation generated by a simple 1/4-approximation algorithm for the HP model on the square lattice.
Approximation on square lattice

**Upper bound:** $2 \cdot X(s) + 2$

**Minimal number of contact with Algorithm:** $X(s)/2$

$\Rightarrow$ $1/4$ approximation.
Approximation on square lattice

- Approach based on context-free grammars
- Applies classical parsing algorithms
- Same approximation ratio but performs better in practice

(Mauri et al., 1999)
Approximation on square lattice

- $G = \{T, N, S, P\}$, where:
- $T = \{H, P, U\}$ is the set of terminal symbols.
- $N = \{S, L, R\}$ is the set of the non-terminal symbols.
- $R$ is the source symbol, i.e. the root of every parse tree.
- $P$ is the set of the productions, composed by the following production schemes:
  1. $S \rightarrow T_1 \ S \ T_2$
  2. $S \rightarrow T_1 \ L \ T_2 \ S \ T_3 \ L \ T_4$
  3. $S \rightarrow T_1 \ L \ T_2 \ S \ T_3 \ T_4$
  4. $S \rightarrow T_1 T_2 \ S \ T_3 \ L \ T_4$
  5. $S \rightarrow T_1 T_2$
  6. $S \rightarrow T_1 \ L \ T_2 T_3 \ L \ T_4$
  7. $S \rightarrow T_1 T_2 T_3 \ L \ T_4$
  8. $S \rightarrow T_1 \ L \ T_2 T_3 T_4$
  9. $L \rightarrow T_1 \ L \ T_2$
  10. $L \rightarrow T_1 T_2$
      with $T_i \in \{H, P\}$.
  11. $S \rightarrow SUU$
  12. $S \rightarrow UU$
  13. $R \rightarrow SS$

(Mauri et al., 1999)
Approximation on square lattice

(1) $S \rightarrow T_1 S T_2$

(2) $S \rightarrow T_1 L T_2 S T_3 T_4$

(3) $S \rightarrow T_1 L T_2 S T_3 L T_4$

(4) $S \rightarrow T_1 T_2 S T_3 L T_4$

(5) $S \rightarrow T_1 T_2$

(Mauri et al., 1999)
Approximation on square lattice

(6) \[ S \to T_1 \ L \ T_2T_3 \ L \ T_4 \]

\[ \begin{array}{c}
\text{if } T_1 = T_2 = H \\
\text{if } T_1 = T_4 = H \\
\text{if } T_3 = T_4 = H \\
\end{array} \]

(7) \[ S \to T_1T_2T_3 \ L \ T_4 \]

\[ \begin{array}{c}
\text{if } T_3 = T_4 = H \\
\text{if } T_1 = T_4 = H \\
\end{array} \]

(8) \[ S \to T_1 \ L \ T_2T_3T_4 \]

\[ \begin{array}{c}
\text{if } T_1 = T_2 = H \\
\text{if } T_1 = T_4 = H \\
\end{array} \]

(9) \[ L \to T_1 \ L \ T_2 \]

\[ \begin{array}{c}
\text{if } T_1 = T_2 = H \\
\end{array} \]

(Mauri et al., 1999)
Approximation on square lattice

Better approximation ratio (1/3).

For simplification: even length sequences s.t. $E[s] = O[s]$.

Algorithm:
1. find $p$ s.t. moving clockwise in the loop starting at $p$, we find at least as many odd-1’s as even-1’s. Moving counter-clockwise we find at least as many even-1’s as odd-1’s.
2. Let $B_{O}$ be, moving clock-wise, the distance to the first consecutive odd-1’s. Moving counter clock-wise, $B_{E}$ is the distance to the first consecutive even-1’s. Fold accordingly to the following cases:

(Newman, 2002)
Approximation on square lattice

Step 1:

Step 2:

\[ B_\sigma = 1, B_\varepsilon = 1 \quad B_\sigma \geq 3, B_\varepsilon \geq 3 \quad B_\sigma \geq 3, B_\varepsilon = 1 \quad B_\sigma = 1, B_\varepsilon \geq 3 \]

Step 3: Repeat Step 2 until even and odd sides meet.

FIGURE 1.6: The steps used in the 1/3-approximation algorithm for the folding problem in the HP model on the square lattice.

(Newman, 2002)
Approximation on cubic lattice

FIGURE 1.7: An illustration of a conformation generated by folding substrings along diagonals of the cubic lattice.
Approximation on cubic lattice

Upper bound: \(4 \cdot X(s) + 2\)

The Hart & Istrail (1995) approximation algorithm can be generalized (Hart & Istrail, 1996) with an approximation ratio of \(3/8 - \Omega(1/\sqrt{X[s]})\).

(Newman & Ruhl, 2004) improved the ratio to 3/8:

1. Separate \(s\) s.t. one substring contains at least half of odd-1’s and teh other sustring at least half of even-1’s.

2. Fold each substrings along two different diagonals.

Same ideas, can be used to reach a ratio of 0.37501...
Constraints Programming Approach

Steps
1. Bounds
2. Core Construction
3. Mapping
CPSP Tools

[Webpage content]

Choose a tool from the left for ad hoc usage
(CPSP-tools version 2.4.2) (LatPack version 1.7.2)

or

Download the full CPSP-tools or LatPack package for local usage!

If you use the CPSP-tools please cite the following publications:
- Martin Mann, Sebastian Will, and Rolf Backofen.
  CPSP-tools - Exact and Complete Algorithms for High-throughput 3D Lattice Protein Studies.

http://cpsp.informatik.uni-freiburg.de