CMSC 858P Lecture 26: Protein folding – threading
5/6/08
Glossary

• Residue – any single amino-acid
• Side-chain – chemical group off the backbone
• Peptide – a short chunk of protein
• Polypeptide – protein
Threading: reverse structure prediction

- Main hypothesis: while there are many protein sequences, there are much fewer folds. I.e. nature keeps reinventing useful structures

- Given a database of structures and a query string, find which structure “fits” the string best
Initial idea: 3D-1D scores

- From a 3D structure, determine “environment” for every amino-acid
  - buried (inside the protein)
  - outside
  - inner side of helix
  - outer side of helix
  - etc...
- Annotate each position in protein with the environment information
  ACKCAHGT -> $E_1E_2E_1E_3E_4E_2E_3E_1E_4$
- Why this is reasonable? Amino-acids have “preference” for specific environments
Alignment to an environment string

• Idea: use gapped alignment algorithm to estimate how likely it is for a sequence to conform to a structure (represented as an environment string)

  \[
  E_1E_2 - E_1E_3 - - E_4E_2 - E_3E_1E_4
  \]

  A G H - K T G A L K M N G

• Question: what is the score of aligning an amino-acid to an environment?
Answer: use statistics

- For each environment – calculate likelihood (observed frequency) of all amino-acids based on known structures
- For each environment – empirical estimation of gap opening/extension penalties
- Alignment algorithm – use Gribskov’s profile method: replace each environment character with the amino-acid frequency table for that environment

\[
S(E_1, G) = \sum_{AA} S(AA, G) \times \text{freq}_{E_1}(AA)
\]

<table>
<thead>
<tr>
<th>E_1</th>
<th>A 0.22</th>
<th>S(AA, G) – e.g. from BLOSUM matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K 0.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>W 0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Environments – not good enough

• Each amino-acid may have multiple contacts
A better model

residue interactions (and associated energy parameters)

core “modules” (helix, sheet, etc.)

variable length connections (gaps)
The threading problem

• Model assumptions:
  – loop AA composition and length contributes to energy score (note: can also place restrictions on minimum/maximum size in gaps)
  – interactions are pair-wise: interaction energy depends on at most two AAs
  – individual AAs in core modules also contribute to energy due to local environment

• Thread a protein sequence through a structure model s.t.
  – the place-holders are filled with amino-acids
  – a variable number of amino-acids fall in the gaps
  – overall energy is minimized

• Easy to say, hard to do: Thus defined (variable length gaps AND pair-wise interactions) the problem is NP-hard!
NP-hard => heuristics

• Branch and bound (Lathrop, Smith)
  – Represent all possible folds (search space) s.t. it is easy to compute a lower bound on the score
  – Note: a threading is uniquely defined by the coordinates of the core elements – a set of threadings is a hyper-rectangle in a C-dimensional space where C is the # of core elements
  – Divide search space and compute energy lower-bounds on each sub-division (choose a dimension (core) and a coordinate and split hyper-rectangle at that location)
  – Recurse on sub-division with lowest lower-bound
Hyper-rectangle heuristic

Each “module” corresponds to a dimension - offset of module in the protein

Fixing one module restricts the flexibility in assigning the remaining modules (imagine beads on a string)
NP-completeness

From: *Computers and Intractability*
M. R. Garey and D. S. Johnson
(W. H. Freeman 1979)

“I can’t find an efficient algorithm, I guess I’m just too dumb.”

“I can’t find an efficient algorithm, because no such algorithm is possible!”

“I can’t find an efficient algorithm, but neither can all these famous people.”
Threading is NP-hard - proof

• Reduction from ONE-IN-THREE 3 SAT
  – n boolean variables, k boolean clauses with exactly 3 literals
  – 3 SAT – is there a setting of the variables such that all clauses are simultaneously true?
  – ONE-IN-THREE 3SAT – 3SAT but each clause made true by exactly one literal

• Proof: for any instance of 3SAT, create an instance of the protein threading problem s.t. a solution to the threading problem implies a solution to 3SAT
Proof ...cont

• Protein sequence
  – T, F – state of each boolean value
  – P,Q,R – which literal makes a clause true
  – protein: PQRPQRPQR...TFTFTF....

• Core model
  – one core element (with one AA) for each clause
  – one core element (with one AA) for each boolean
  – interactions from each clause to the booleans present in it. edge also encodes which literal (1,2,3) and whether value is negated
  – edge score = 0 if label consistent with amino-acid assignment and 1 otherwise (e.g. QF is consistent with edge 2,NOT)
  – optimal threading has score 0 and solves 3SAT
Discussion

- Both variable length gaps and pairwise interactions are essential!
- If no variable length gaps – can try all threadings in polynomial time irrespective of interactions
- If no pairwise interactions – dynamic programming can figure out the correct assignment (essentially the alignment problem)
Structure to structure alignment

• Given two proteins with known structure, how do we align them to each other?

• Double Dynamic Programming
  – distance matrix depends on distance between residues
  – pick a pair of residues (i,j) and assume they are paired up
  – use dynamic programming to align the rest of the protein – score will represent score for pairing of i,j
  – use a final dynamic programming step to align the proteins based on scores determined above
Example

Structure A

Coordinate system coincidence at $a_i, b_j$

B

$d_{kl}$