1 Lecture 17: Protein Structure Prediction

References:

- bcbcf.bsd.uchicago.edu/tclark/classics/ungerandmoul1993.pdf
- www.cs.biu.ac.il/~amir/Postscripts/thread.ps
- http://www.med.unibs.it/~marchesi/forces.html
- http://citeseer.nj.nec.com/storm96prediction.html
- http://www.chem.ualberta.ca/~plambeck/che/p102/p02051.htm
- http://www.cryst.bbk.ac.uk/PPS2/course/section7/os_sol.html
- http://www.cs.brandeis.edu/~cs178/classnotes/PFolding/
2 Protein Folding

- a protein is a sequence of amino acids
- 20 different kinds of amino acids are known to occur in proteins
- the length of most protein sequences range from 50 to 1000 amino acids
- apparently, the amino acid sequence alone is enough to guide the folding
  - if two protein molecules have the same sequence, they fold up into the same shape
3 Protein Folding

- knowing a protein’s precise spatial structure is important since it is the structure that determines the molecule’s function

- experimentally, it is:
  - easy to experimentally determine the primary amino acid sequence of proteins
  - hard to experimentally determine the three-dimensional structures of proteins using techniques such as x-ray crystallography and nuclear magnetic resonance

- so there is an intense search for protein structure prediction methods: methods that can predict the three-dimensional structure of a protein based on its primary amino acid sequence
4 Forces Controlling Protein Structure

- hydrogen bonding
- electrostatic forces
- van der Waals Forces
- hydrophobic forces
5 Detailed Simulations

- the most detailed simulations of protein folding track the motion of every atom
- they try to reproduce all the chemistry and physics going on in the system
- however, such detailed models require hours of computer time just to simulate a few picoseconds of molecular dynamics
6 Lattice Models

- often use 2D or 3D square lattices in simpler models
- number of different conformations of a given protein sequence is limited in a lattice model
- this means we can try all foldings to see which one is “best”
- in a 2D square lattice, one can easily see that for a chain with $n$ beads, the number is $\leq 4 \times 3^{n-2}$ (why?)
- we can also find a lower bound for the number of different legal conformations (i.e., a self-avoiding conformation so that no two beads occupy the same point in the lattice)
7 Staircase

Figure 1: A “staircase” conformation in the 2-dimensional square lattice.
8 Lower Bound

- it is easy to show that there are at least $2^{n-1}$ legal conformations
- a “staircase” conformation is a folding in which a change of direction alternates between a left turn and a right turn
- all staircase conformations are legal
- if the position of bead $i$ is determined, then bead $i + 1$ can only be placed in two different adjacent lattices sites: either go straight or make a turn
- this means there are $2^{n-1}$ different staircase conformations of a sequence of length $n$
9 HP Model

- the hydrophobic effect is of great importance to protein folding

- each amino acid can be either of two types:
  - H (hydrophobic / non-polar): they move away from water
  - P (hydrophilic / polar): they move toward water

- H type amino acids tend to aggregate in the presence of water forming a protein core; that is, they are moving away from the water, protected by hydrophilic amino acids that are attracted to water

- idea: try to fold protein so that H’s are adjacent in lattice; however, there is no H-H bond really

- while resultant shape may not be indicative of true shape, it is hoped that the shapes would be functionally similar
10 Free Energy

- prediction of protein structure typically based on fact that native conformation is one for which the free energy achieves the global minimum

- free energy depends on both potential energy and entropy

- if you could straighten out a protein, the result would be a state of high potential energy and low entropy

- the potential energy is high because amino acids that “want” to be close together are held at a distance

- the entropy is low because the straight chain is a highly ordered configuration

- if let go, the chain springs back into a shape with lower potential energy and higher entropy, which translates into a lower value of free energy
11 Free Energy in HP Model

- the free energy function in the HP model is simply taken to be the number of H-H contacts in the lattice negated (not counting explicit bonds between consecutive H’s in the chain)

- thus the conformation of lowest free energy is the one that maximizes the number of such H-H contacts
12 Conformation in 2D HP Model

Figure 2: A conformation in the 2D HP model with free energy 9\epsilon.
13 Exercise

- Come up with the best conformation of the following sequence in the HP model:

- HPPHPHHHPHPH
14 Exercise

- Come up with the best conformation of the following sequence in the HP model:
- HPHPHHHHPHPH
- (note mistake: all dots on bottom boundary should be open circles)

![Diagram](image)

Figure 5: An optimal conformation of the string 100101110101 with score 5.
15 HP Model

- finding the fold yielding maximum score is NP-complete in both 2D and 3D lattices

- however, approximation algorithms exist for this model (approximation algorithms do not necessarily exist for all NP-complete/NP-hard problems)

- Hart and Istrail have developed an algorithm that is within $3/8$ of optimal for the 3D lattice (e.g., if the best score is 80, then the algorithm guarantees a score of at least 30)

- they also developed an algorithm that is within $1/4$ of optimal for the 2D lattice (e.g., if the best score is 80, then the algorithm guarantees a score of at least 20)
16 Approximation Algorithms

- why should we care about approximation algorithms?

- just because a problem is NP-complete/NP-hard, doesn’t mean we should give up; a reasonable solution is still required

- while it may not be possible to find the best solution in a reasonable amount of time, we should at least try to find a good solution

- we can use various heuristics like genetic algorithms, hill climbing, etc.

- unlike simple heuristics, approximation algorithms (more sophisticated heuristics) provide a worst-case guarantee
17 Another Approach: Monte Carlo Method

1. start from a random or straight-line conformation (population size=1 here)

2. from a conformation \( S_1 \) with energy \( E_1 \), make a single random change of the conformation to conformation \( S_2 \) and evaluate its energy \( E_2 \)

3. if \( E_2 \leq E_1 \), then accept the change to conformation yielding \( S_2 \);
   otherwise decide randomly whether to accept the change based on the energy increase of the change:

\[
\text{Rnd} < e^{\left( \frac{E_1 - E_2}{c_k} \right)}
\]

where \( \text{Rnd} \) is a random number in \([0, 1]\) and \( c_k \) is gradually decreased (cooled) during the simulation to achieve convergence; observe that if \( E_2 \) is much larger than \( E_1 \), then acceptance is unlikely; also observe that if \( c_k \) is close to 0, then acceptance is also unlikely.
Monte Carlo Method

- random change can be done by randomly selecting an amino acid and rotating the C-terminal portion of the chain around the amino acid
- (if result is not a self-avoiding chain, try another random change)
Monte Carlo Method
20 Genetic Algorithm

- genetic programming is a kind of genetic algorithm
- genetic algorithms need not evolve programs; they may evolve other non-program artifacts as well such as protein structures
- to implement a genetic algorithm, one needs to choose
  - method for encoding the data
  - method of applying genetic operators
  - method for selecting individuals for the next generation
21 Genetic Algorithm: Encoding

- idea is to simply use conformations as individuals; no special encoding is used

- (alternatives may be to encode conformations in a binary string for example)
Genetic Algorithm: Genetic Operators

- in each generation, *each structure* is subject to a number of mutation steps
  - each mutation is the same as a single Monte Carlo random change; it is also subject to similar acceptance criteria as in the Monte Carlo process

- at the end of this Monte Carlo stage, the crossover operation is performed
  - lower energy conformations have a higher chance of being selected
23 Genetic Algorithm: Crossover

- N-terminal portion of structure (A) is connected to C-terminal portion of structure (B)
24 Genetic Algorithm: Crossover

- there are three ways to join parts together (connecting chains with angles of $0^\circ$, $90^\circ$, $270^\circ$)

- these possibilities are tested in random order to find one that is valid; if none lead to a self-avoiding structure, then another pair of structures is selected
### 25 Genetic Algorithm: Crossover

- in our example, a $270^\circ$ angle yields

![Diagram of genetic algorithm crossover]

- observe child has a lower energy than either parent
26 Genetic Algorithm: Crossover

- once a valid structure $S_k$ is created from crossover, its energy $E_k$ is evaluated and compared to the average energy $ar{E}_{ij} = (E_i + E_j)/2$ of its parents
- the structure $S_k$ is accepted iff $E_k \leq \bar{E}_{ij}$ or if
  \[ Rnd < e\left(\frac{E_{ij} - E_k}{c_k}\right) \]
- crossover operation is repeated until $N - 1$ new accepted child structures have been constructed
- in addition, the lowest energy conformation in each generation is directly replicated to the next generation
27  Sample Results

- (A) shows five structures from 5th generation;
- (B) shows five structures from 10th generation, including one with the lowest possible energy for this sequence of -9
28 Comparison

• genetic algorithm approach found to be superior to Monte Carlo

• note that Monte Carlo only has a population of 1, so genetic algorithm approach takes a lot more memory

• however, if we consider the number of energy evaluations (the most expensive part), we find that genetic algorithm approach is much better at finding lower energy conformations than Monte Carlo given the same number of energy evaluations
29 Another Approach: Threading

- even the simplest models such as the 2D and 3D HP models yield NP-complete problems
- there was a hope that by “inverting” protein folding we would obtain a simpler problem to solve:
  - thread a protein chain into known protein structures (determined experimentally) to see which one is most likely to be representative of the fold for the given protein
- this approach makes sense since there are many more different sequences than different structures, so many sequences must fold to similar conformations
30 Protein Threading

- let \( k \) be the number of amino acids in the given protein
- let \( n \) be the number of amino acids in a known protein structure
- the number of ways in which we can thread a given protein to a known structure is equal to the number of ways in which we can distribute \( k \) identical objects into \( n \) different cells, where a cell may contain more than one object
- thus, the number of possible threaded is:

\[
\frac{(n - 1 + k)!}{(n - 1)!k!}
\]

(why?)
31 Protein Threading

- consider distributing \( k \) identical objects into \( n \) different cells
- use \( n - 1 \) barriers to separate the \( n \) different cells; denote a barrier by "*"
- in addition to the barriers, we also have \( k \) identical objects; denote these by "a"
- example: for \( n = 3 \) and \( k = 5 \), we might have: aaa**aa; the first cell contains 3 a’s, the second cell contains 0 a’s, and the third cell contains 2 a’s
- now we find the total number ways in which we can arrange the barriers and identical objects into a string:

\[
\frac{(n - 1 + k)!}{(n - 1)!k!}
\]

(why?)
32 Protein Threading

- as we have seen, the size of the search space of possible threadings is

\[
\frac{(n - 1 + k)!}{(n - 1)!k!}
\]

- if \( k = 50 \) and \( n = 50 \) (for small proteins), we have 50,445,672,272,782,096,667,406,248,628 possible threadings

- moreover, it has been shown that search in this space belongs to the NP-complete class of difficult problems

- so threading is no easier than the direct approach!

- however, threading typically yields better results, so we shall discuss it further
33 Genetic Threading

- since threading is NP-complete, we need to use a heuristic
- for each known structure, we can find the best threading using a genetic algorithm
- given the best threading for each known structure, we can then identify the known structure most likely to represent the folding of the given protein
34 Genetic Threading: Encoding

- use a sequence of numbers for the encoding

APSWFIGNALGATS
35 Genetic Threading: Encoding

- in example on previous slide, we thread protein sequence APSWFIGNALGATS through a known structure
- particular threading shown is denoted by 1111100111311
- first 6 1’s in the sequence indicate that APSWFI will go in the first 6 slots in the known structure
- next two 0’s indicate that next two known structure slots are not used at all
- next three 1’s indicate that GNA will go into the next three known structure slots
- next 3 indicates that L will go into the next structure slot but G and A will not be assigned to any known structure slot
- next two 1’s indicate that TS go into the last two known structure slots
36 Genetic Threading: Encoding

• when does the encoding string represent a valid threading?
  – the total sum of the numbers that appear in the string must be equal to the length of the given protein
  – the length of the string must be equal to the length of the known protein structure
37 Threading Fitness Function

- a fitness function can take into account a variety of factors
  - reward common amino acid pair interactions (e.g., if Arg-Asp pairs are found in a distance of about 5 Angstroms in a much higher frequency than expected, then this interaction should be rewarded in a threading)
  - would reward a fold where hydrophobic amino acids are buried inside the structure while hydrophilic amino acids lie at the surface
  - could penalize cases where given protein amino acids are not put in a known structure slot at all (e.g., when you have > 1 in the number sequence)
  - could penalize cases where known structure slots are skipped (e.g., when you have 0 in the number sequence)
38 Genetic Threading: Mutations

- mutations are performed by increasing or decreasing randomly the value of a number in the string
- this is offset by an opposite change in the same amount in other positions
- example:
  - 11111100111311 mutated to
  - 11111100211211 by changing the 3 to 2 at position 12 and offsetting this by increasing 1 to 2 at position 9
39 Genetic Threading: Mutations

- there are many variations possible that involve
  1. selecting number of mutations to be performed in each string
  2. selecting number of amino acids that can be changed in each mutation
  3. whether an increase in one position must be compensated by a change in a neighboring position or can the correction be spread all over the string, etc.

- another possibility is to reverse a small substring; this clearly maintains the validity of a threading
40 Genetic Threading: Crossovers

- crossovers are performed by randomly choosing a position, and
  - taking the prefix up to that position in one parent combined with the suffix from another parent to produce one child, and then
  - combining the suffix from the first parent with the prefix of the second parent to produce another child

- clearly, the resultant string length is correct but the sum of the numbers may no longer add up to the length of the given protein

- in that case, a random position is chosen near the crossover point and its value is changed accordingly
41 Genetic Threading: Crossovers

- example:
  - parents are 11201120111111 and 111111001111311 (these sum to 14)
  - say the crossover point is position 8, underlined above
  - one child would get 11201120111311
  - however, this sequence sums to 16 not 14
  - correction mechanism might change string around position 8 to yield 11201120001311
  - again, variations on this technique are possible
42 Results

- experimentation shows that this genetic algorithm is a feasible and efficient approach to threading
  - it is quite robust and not overly depending on the particular selection of parameters or genetic operators
43 Levinthal’s Paradox

• we have seen that even the simplest model for protein folding is NP-complete
• we have also seen that threading is also NP-complete
• Levinthal’s question: if protein folding is so hard, how do proteins do it?
Levinthal’s Paradox (answer #1)

- perhaps protein molecules are capable of mathematical wizardry beyond the reach of conventional computers
- if this is so, you could encode an instance of any NP-complete problem in a synthetic sequence of amino acids, then let the protein fold itself up
- from the folded configuration, you could read out the solution to the NP-complete problem instance
Levinthal’s Paradox
(answer #2)

- perhaps proteins do not always fold efficiently and spontaneously
- for example, some may need help from “chaperon” molecules
- some may fold erroneously and are recycled by enzymes
- perhaps the native state of some proteins is not in the state of lowest free energy but perhaps something close to it (remember approximation algorithms do exist for the HP-model)
Levinthal’s Paradox

(Answer #3)

• perhaps proteins do quickly find the best among all possible foldings, but only because they have evolved to exhibit precisely this property

• in other words, the only amino acid sequences that survive under evolution are those that happen to fold rapidly

• sequences that fold hierarchically could fit this description: if small sections of the chain fold independently into secondary structures such as helices, which then aggregate without further internal rearrangement, then it is possible to avoid a combinatorial explosion in possible foldings