Lectures 19, 20, and 21: RNA and Protein Folding

Steven Skiena

Department of Computer Science
State University of New York
Stony Brook, NY 11794–4400

http://www.cs.sunysb.edu/∼skiena

Shape and Structure of Molecules

The primary molecules of biological interest (DNA, RNA, and proteins) all have fundamentally linear structures as a consequence of how they are replicated.

However, the biological *function* of these molecules depends upon how they interact with other molecules.

These interactions depend heavily upon the shape of the molecules involved.

Experimental methods for determining structures, such as x-ray crystalography, are slow, expensive, tricky, and work only in certain environmental conditions.

All this combines to make computational prediction of structures an important and messy problem.

Why Shape?

Molecular shape/structure is determined by many factors:

- Molecular bonds
- Electrostatic forces (i.e. positive/negative charges)
- The size/shape of molecular subunits (amino acids and nucleotide bases)
- Hydrophobicity of the associated bases
- The current environmental conditions (temperature, salinity, acidity).

Levels of Structure

The difficulties of specifying structure leads to a hierarchy of increasingly demanding notions of structure:

- *Primary structure* usually refers to the raw sequence itself.
- *Secondary structure* usually refers to identifying certain self-interacting features of the structure, such as which bases bond with which other bases.
- *Ternary structure* is the complete 'geometric' description of molecule; i.e. the positions of all the bases.
- *Quadrary structure* concerns identifying how certain parts of structures interact with other structures.

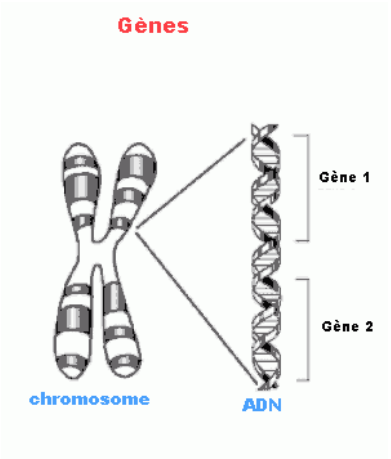
The interest in these different levels of abstraction is (1) that sometimes it is much easier to get accurate predictions at lower levels, and (2) accurate lower-level knowledge may be more useful than less-precise higher-level knowledge.

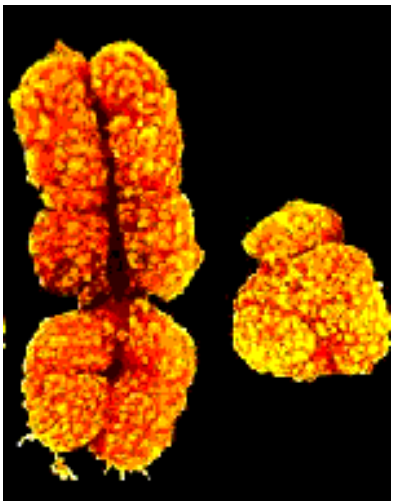
DNA folding

DNA molecules usually come in the form of double-stranded molecules, whose *secondary* structure is the famous ‘double helix’ discovered by Watson and Crick.

The *ternary* structure of DNA is the shape of the chromosomes it folds into.

This is a non-trivial problem – the DNA in a typical human cell would unfold to sequences 2 meters long!





When DNA is in single-stranded mode, other regulatory proteins bind at certain sites to start transcription, defining the *quadrary* structure.

As a single-stranded molecule, DNA folds and behaves much like RNA.

RNA Folding

RNA molecules, like proteins, are usually single strands which *fold* back onto themselves into predefined 3D shapes or structures.

The folding problem seeks the structure or shape of a given sequence.

The shape of certain RNAs plays a major role in determining its interaction with other molecules, for example tRNAs.

Folding occurs in both proteins and RNA, although the issues are different.

Since RNA is single-stranded, its component bases tend to bond with other bases analogously to the bonds formed in double-stranded DNA (A-U, C-G).

What Does RNA Do?

Some RNA molecules have *functions* in the organism other than coding or information functions.

These functions are determined by their interaction with other molecules, which in turn is determined largely by its 3D structure.

For example, tRNA molecules transport amino acids during the process of transcription.

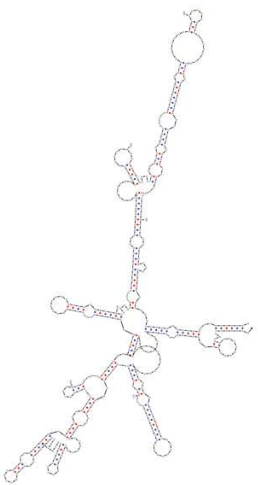
It is widely believed that RNA molecules are the closest thing to the molecules from which life originally evolved.

RNA molecules can perform the function of coding for proteins (information storage) usually associated with DNA.

RNA molecules can have enzymatic functions usually associated with proteins.
Thus it is easier to develop a scenario for the polymerization of nucleic acids than proteins, under the geological conditions of the young Earth.

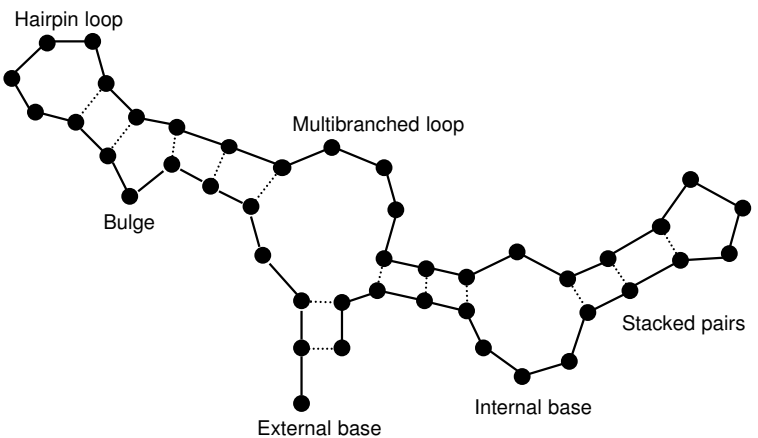
Predicted RNA Secondary Structure

The set of all these pairs constitutes the *secondary structure* of an RNA molecule.



The Zuker-Turner RNA Folding Model

In this widely used model, binding pairs partition the RNA strand into nested loops.



A complicated energy function (derived from laboratory experiments) is used to measure the binding strength of short patterns.

Using dynamic programming Zuker, et.al has developed successful RNA secondary structure prediction algorithms – on average, 73% of known base pairs on domains of fewer than 700 nucleotides.

Maximizing Binding Pairs

The simplest RNA folding model would seek the nested decomposition into bonds such that we maximize the number of bonds of complementary pairs.

We assume that the total binding energy is the sum of the energy of each of the bonds.

We can optimize by dynamic programming.

Recurrence

Let $E(i, j)$ be the maximum number of properly-nested bonds which can be formed by the substring from the ith through jth bases in the sequence S.
Either (i, j) is a bond, or they might bond with bases between i and j, so:

$$E(i,j) = \max \left(\begin{array}{l} E(i+1, j-1) + \alpha(i,j), \\ \max_{k=i}^{j-1}(E(i,k) + E(k+1,j))) \end{array} \right.$$

where $\alpha(i, j)$ is the score you get for matching S_i and S_j.
As there are n^2 cells and each can be filled in linear time, this algorithm takes $O(n^3)$ time.

General RNA Folding Recurrences

More general recurrences are needed to properly account for more general structures, such as (1) rewarding long runs of matched/stacked pairs, and (2) penalizing different types of loops appropriately based on size.

Because of the internal loops term, these recurrences run in $O(n^4)$. Faster recurrences are possible, especially when there are simplifying assumptions about the form of the penalty terms.

Pseudoknots

This model ignores *pseudoknots* formed when bonds are formed which do not respect nesting constraints, i.e. (i, j) and (k, l) both form bonds even though $i < k < j < l$.
Such pseudoknots definitely occur in nature, but are usually ignored in secondary structure prediction because (1) they make the problem too hard computationally, and (2) they might be better handled during ternary structure prediction.

Homology-Based Approaches

De novo RNA structure predictions are reasonably good but not perfect.

Indeed, one problem with the dynamic programming approach (as stated) is that it returns only the single *best* solution, when there might be widely varying structures with almost as good energy scores.

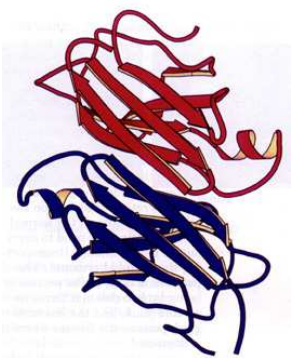
A more accurate approach for determining the structure of functional RNAs would be to use *homology* information across species, since the most important structural components should be conserved in evolution.

Thus potential bonds which appear in multiple sequences are more likely to be real bonds.

Protein Folding

Linear protein molecules rapidly *fold* into predefined 3D shapes or structures.

The properties of any protein is largely determined by its structure.



Proteins can be *denatured* by heat or chemical agents, but then fold back to their original shape.

Protein structures can be experimentally determined by *crystallizing* the protein and then using x-ray crystallography or NMR to find the position of the atoms, but this is a very difficult procedure.

Why is it Difficult?

The folded structure of a sequence is determined by the sequence of successive solid bend angles, where each solid angle can be represented by two planar angles.

Such a problem can be made discrete (at some loss of accuracy) by limiting the number of ways to bend each joint to, say, 7 solid angles.

Even so, a 100 residue protein then has a search space of 7^{100} configurations.

Determining the shape of proteins from sequence is one of today's great computational challenges.

Primary, Secondary, and Tertiary Structures

The *primary* structure of a protein is simply its amino acid sequence.

The *secondary* structure of a protein is the labeling of each residue with whether it is part of an (1) α-*helix*, (2) β-*sheet*, or (3) a connecting *loop*.

Secondary structure prediction is important because the helices and sheets determine the protein *core* which is typically conserved.

Different amino acids have different probabilities of appearing in each of these structures. But beware, since there is a sequence of 5 residues which appears in both α-*helix* and β-*sheet*.

Although the notion of secondary structure seems somewhat ill-defined, there are reasonably successful prediction programs (say correctly labeling 75% of all bases) based on ideas like hidden Markov models.

The 3D or *tertiary* structure of a protein describes the coordinates in space of each amino acid. This geometric information helps determine whether two proteins interact or *dock* with each other.

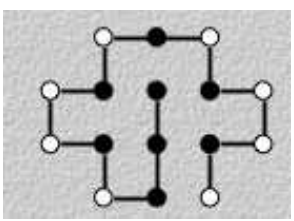
Protein *folding* programs seek to determine the tertiary structure of any protein from its sequence.

Lattice Models

Discrete, folding models provide insight into the statistics and optimization of protein folding.

Grid models limit the position of the residues to discrete points on a lattice.

The HP-model restricts attention to interactions between hydrophobic and polar (hydrophilic) residues.



We seek the self-avoiding lattice walk which maximizes the number of neighboring hydrophobic pairs.

The Hardness of Protein Folding

The computational difficulty of protein folding has led to proofs that the problem of finding the minimum energy configuration is NP-complete under a variety of models, e.g. maximizing the number of adjacent hydrophobic pairs in a 3D lattice model.

Leventhal's paradox is that proteins correctly fold into their pre-ordained shape less than a minute after being synthesized.

How does nature solve this NP-complete problem?

Around the Paradox

Possible reasons around this problem are (1) that the theoretical models used to prove hardness are not what nature is trying to optimize, (2) evolution may have selected for proteins which fold easily, (3) proteins may well fold in locally, not globally optimal ways.

Prions, infectious agents which work by “tricking” proteins to fold in non-functional ways, are presumed responsible for mad-cow disease.

De Novo Structure Prediction

De novo (or ab initio) prediction programs work by defining a global *energy function* and does a search of possible bond-angle configurations to find one which minimizes total energy.

The process is similar watching a restless sleeper folds into the most comfortable (minimum energy) configuration.

The most important issues are (1) the energy function selected, and (2) the optimization procedure employed to search the space.

Optimization by Search

Reasonable energy minimization functions include hydrophobic/hydrophilic interactions, size and flexibility properties of different amino acids, and electrostatic / Van der Waals interactions of nearby atoms.

Standard optimization methods to employ are *gradient descent*, *simulated annealing*, *genetic algorithms*, and *parallel computation*.

Simulated Annealing

The inspiration for simulated annealing comes from cooling molten materials down to solids. To end up with the globally lowest energy state you must cool slowly so things cool evenly.

In thermodynamic theory, the likelihood of a particular particle jumping to a *higher* energy state is given by:

$$e^{(E_i - E_j)/(k_B T)}$$

where E_i, E_j denote the before/after energy states, k_B is the Boltzman constant, and T is the temperature.

Since minimizing energy is a combinatorial optimization problem, we can mimic the physics for computing.

Simulated Annealing Algorithm

Simulated-Annealing()
- Create initial solution S
- Initialize temperature t
- repeat
  - for $i = 1$ to *iteration-length* do
    - Generate a random transition from S to S_i
    - If $(C(S) \leq C(S_i))$ then $S = S_i$
    - else if $(e^{(C(S)-C(S_i))/(k \cdot t)} > random[0, 1))$
      - then $S = S_i$
  - Reduce temperature t
- until (no change in $C(S)$)
- Return S

Components of Simulated Annealing

- *Concise problem representation* – Both a representation of the solution space and an appropriate and easily computable cost function $C(s)$.
- *Transition mechanism between solutions* – Typical transition mechanisms include swapping the position of a pair of items or inserting/deleting a single item.
- *Cooling schedule* – These parameters govern how likely we are to accept a bad transition. At the beginning of the search, we are eager to use randomness to explore the search space widely, so the probability of accepting a negative transition is high.

The Rosetta Method

The quite successful *Rosetta stone* method works by computing possible folds for each subsequence of length 3-9 based on known structures and stitching them together using simulated annealing.

It assumes that the distribution of conformations sampled for a given short segment of the sequence is reasonably well approximated by the distribution of structures adopted by that sequence and closely related sequences in known protein structures.

Fragment libraries for short segments of the chain are extracted from the protein structure database.

At no point is knowledge of the overall native structure used to select fragments or fix segments of the structure.

The conformational space defined by these fragments is then searched using a Monte Carlo procedure with an energy function that favors compact structures with paired strands and buried hydrophobic residues.

A total of 1,000 independent simulations are carried out for each query sequence, and the resulting structures are clustered. One selection method was simply to choose the centers of the largest clusters as the highest-confidence models.

Keeping Score

How can we judge how well a protein prediction program works?

One measure is to align the correct and predicted 3d structures and compute the average (RMS) deviation per residue.

Finding this alignment is not trivial, and misses the fact that the core structure is what is most important.

The CASP project/competition regularly invites structure predictions of proteins about to be experimentally determined, and determines the winner on a more ad hoc basis.

Threading Approaches

Since de novo structure prediction is hard, many programs use known 3D structures as a crutch to help folding new sequences.

This makes sense since all proteins likely descend from a small number of original structures.

Two amino acid sequences with $> 20-30\%$ identical residues likely have similar three dimensional structures.

Thus there may only be a small number of different folds/substructures common to all proteins, and we will likely see them all after determining a given number of structures.

In general, *threading* or *inverse folding* programs are more accurate than de novo prediction programs.

The Threading Problem

The input is (1) a protein sequence, (2) a core model describing the position of the core residues and allowable lengths of loops, and (3) a scoring function to evaluate the given threading.

Reasonable factors in the cost model include (1) the similarity of the base at each position to the original, (2) the length and similarities of the loops,and (3) pairwise interactions between bases at core positions.

Without modeling pairwise interactions, this becomes a simple dynamic programming-type problem.

However, incorporating pairwise interactions turns the problem NP-complete.

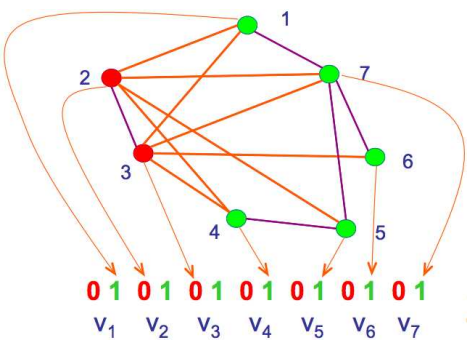
Why is Threading Hard?

A pairwise interacting optimization function requires tabulating the possible substructures for every base assignment, so dynamic programming becomes less feasible.

Thus exhaustive search/heuristics are used in threading programs, but the options are much more constrained than for de novo folding algorithms.

NP-Completeness Proof

An NP-completeness proof is based on max cut in graphs, where a protein of length $2n$, namely $(01)^n$ is threaded along an n-residue core (graph) with maximum loop lengths of 1. Thus every vertex gets assigned a 0 or 1. If the cost function scores one for every graph edge with different residues, the maximum cut maximizes the score.



Protein Shapes

The tertiary structure of a protein specifies the location of each carbon atom along its backbone.

The secondary structure (helices and sheets) captures some notion of shape, but it does not suffice to accurately predict whether two proteins bind or *dock* together.

Predicting protein interactions arises critically in searching databases for potential drugs (*rational* drug design).

In *protein docking*, we seek to (1) predict the binding between two different proteins, or a protein and a flexible *ligand*, and (2) identify the orientation maximizing the interaction.

Protein Representations

A variety of different representations can be used for geometric protein structures:

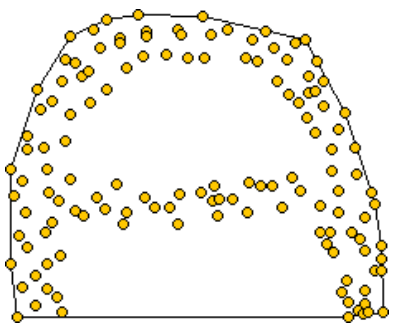
- 3D points in space.
- An arrangement of spheres in space.
- A chain of bond angle pairs.
- A cloud of unit bond vectors.

All are somewhat of a fiction since molecules vibrate, move, and bend.
This flexibility limits our ability to use standard geometric algorithms and concepts.

Geometric Notions of Shape

The idea of the *shape* defined by a set of points is inherently difficult to define.

The *convex hull* of a set of points defines the smallest convex polygon which contain all of them.



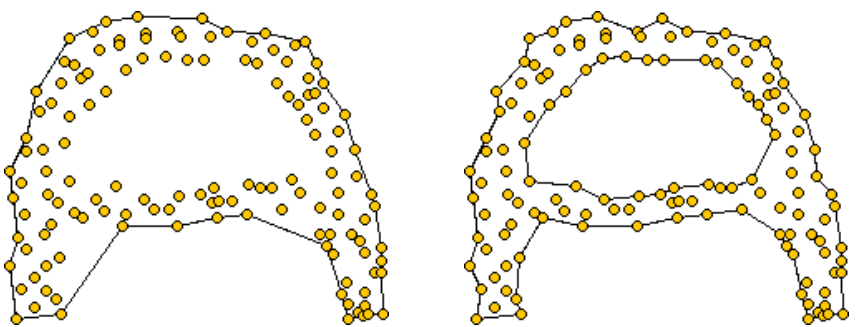
The convex hull fails to pick up the cavities and protrusions which inherently make shapes interesting.

Structures based on connecting points to their nearest neighbors can recover if the points have been sampled densely enough.

Alpha Hulls

The *alpha*-hull is a generalization of the convex hull, where the shape is defined by spheres of radius alpha, for some given value of α.

An edge (face) between two (three) points is *alpha exposed* if there is a sphere of radius alpha which contact these points and contain no internal points.

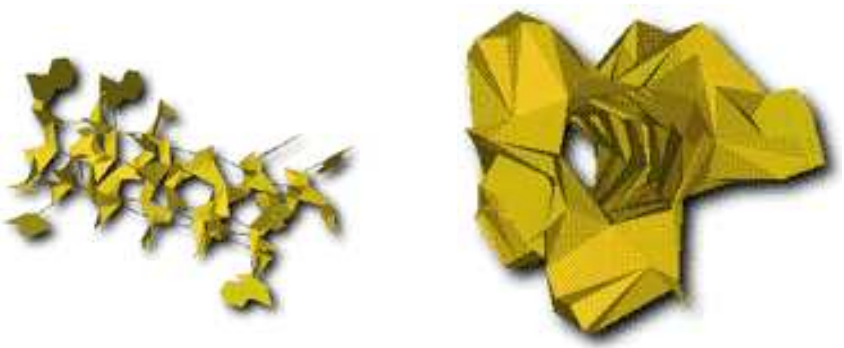


As alpha decreases, concavities get cut out from the convex hull.

The theory gives you little insight into which value of α defines your shape, except by trial and error.

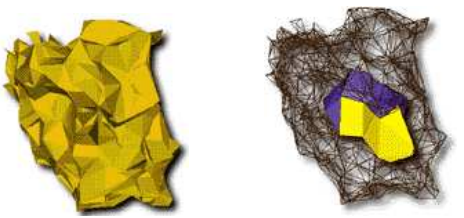
Alpha Shape Examples

Two different alpha-shapes Gramacidin A, the latter highlighting the tunnel through the molecule:

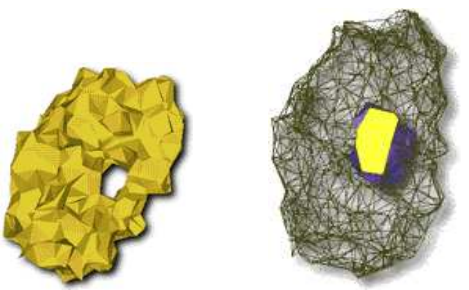


The entire spectrum of alpha-hulls can be constructed in $O(n \log n)$ time in the plane, the same as for convex hulls.

Myoglobin molecule with heme binding pocket:



HIV protease with inhibitor binding site:



Protein-Protein Docking

Typically, both proteins are modeled as rigid bodies, with the geometry used to constrain the possible sites of interaction.
Energy computations are performed at geometrically possible binding sites.
The “right” way to solve such problems is to construct the six dimensional *configuration space* of allowable positions of the second protein, and perform energy calculations at vertices/edges of the allowable region.

Protein-Ligand Docking

Modeling the interactions between a rigid protein and a small but flexible ligand is more complicated, since every hinge in the ligand increases the dimensionality of the problem.

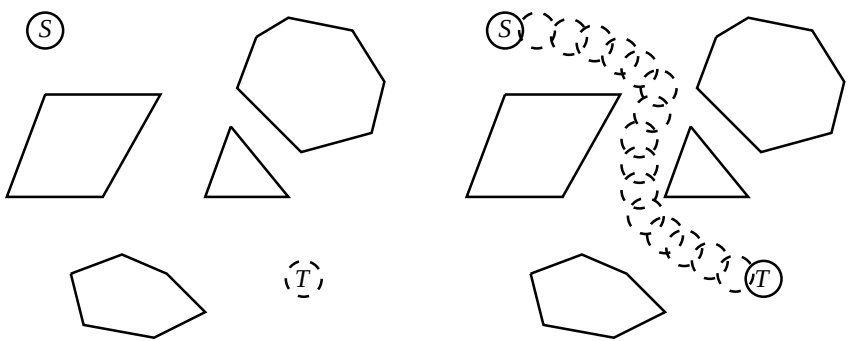
Rough geometric interactions with parts of a ligand can be used to predict possible binding sites, but detailed energy calculations are needed to make precise predictions.

Docking Criteria

Preliminary screenings of possible docking sites can be based on maximizing the number of *contact pairs* or RMS distance.
The docking problem is not purely geometric, since attractive/repulsive forces have strong effects.
The best docking seeks to maximize the surface area and attractive forces while minimizing the energy loss due to solvent interaction.
Small ligands tend to bind in big pockets.

Motion Planning

Finding the best docking is a difficult algorithmic problem because it involves six *degrees of freedom*, the three possible translations (x, y, and z) and three possible rotations.
Binding flexible ligands is analogous to motion planning for articulated robot arms.



Motion Planning Algorithms

Motion planning with many degrees of freedom becomes difficult as the complexity of the surfaces defining the conformation space grows.

A good general approach is to *randomly sample* points in the configuration space, and add edges between nearby collision-free points with collision-free straight line paths.

Heuristic Approaches

One approach to simplifying continuous geometric problems is to insist that all sites lie on a 3D grid.
The finer the grid, the more accurate the predictions, though at greater computational cost.
Another approach to discretization is to analyze the possible positions of isolated spheres which contact the surface, with pockets identified where there are many intersecting spheres.
Geometric hashing stores all possible point triplets (triangles) in both ligand and receptor. The sets of triangles which match defines molecular orientations of interest.
Note that conventional hashing techniques do not really apply, since we are looking for approximate matches.