

## The protein folding problem

What determines the structure of a protein?

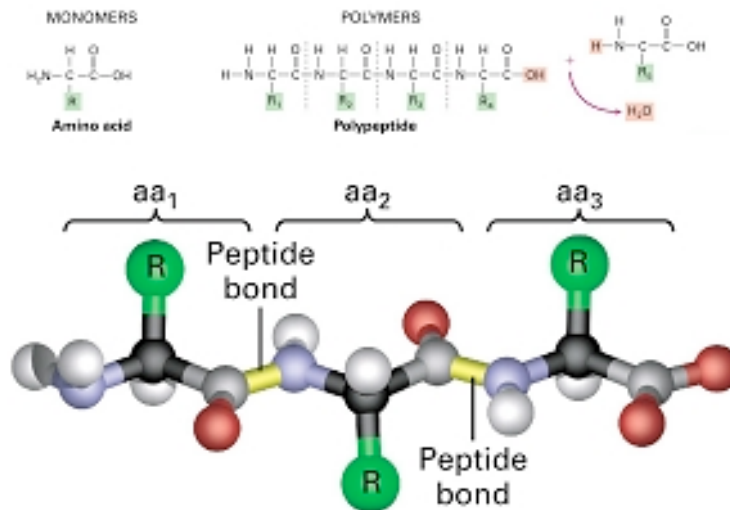
- i. Energy
- ii. Kinematics

Protein structure is hierarchical. The primary structure implies the higher orders of structure. Combinations of smaller motifs are followed by combinations of domains on a higher level (tertiary structure). Domains are the building blocks of proteins, and many domains might constitute a domain. We can then have a supermolecular structure containing several proteins (like ribosome).

## Primary Structure: Sequence



- The primary structure of a protein is the amino acid sequence



Geometrically, we see very complicated structures of strings woven into a knot. In general, we do not know what force drives protein folding, but kinematics plays a crucial role.

E.g. Lets say we have a little creature that invented some useful protein. but it

takes a million years to fold because it must form a complicated structure. Then it will never get the chance to fold and hence evolutionarily, it is prone to mutating out of existence. So proteins that occur in nature must fold in a fairly short amount of time.

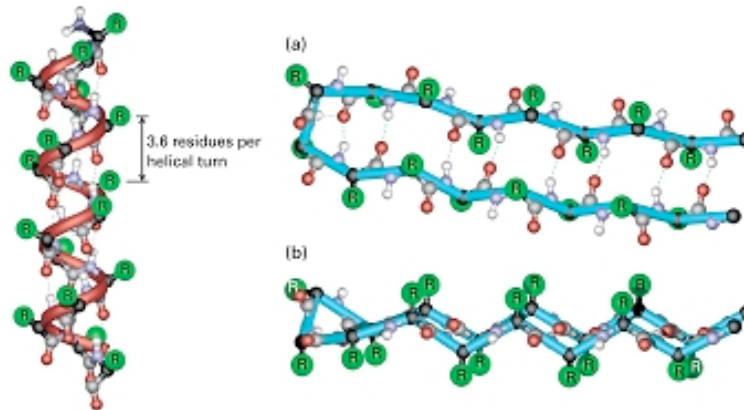
Secondary structures: Alpha helices and Beta strands

Alpha helices: 3.6 amino acids per turn with hydrogen and oxygen molecules  
Beta strands form sheets instead of helices, they can be parallel or anti-parallel, stabilized also by hydrogen bonds.

## Secondary Structure: $\alpha$ , $\beta$ , & loops



- $\alpha$  helices and  $\beta$  sheets are stabilized by hydrogen bonds between backbone oxygen and hydrogen atoms



Motifs that appear again and again in domain: e.g. Helix-Loop-Helix, Coiled Coil (two alpha helices wound together and shared by two proteins), Beta Barrel (a beta sheet wound in a helical structure). These motifs comprise domains, the functional and evolutionary units of a protein.

A crude classification of protein folds can be devised based on how alpha helices and beta strands are combined.

Protein folding is the holy grail of computational biology. Ultimately, this will allow us to predict protein structures in an arbitrary genome, and consequently protein function.

Energy: The different forces involved in protein folding, in order of frequency:

1. Hydrogen bonds (3-7 kcal/mole)
2. Ionic interactions between charged amino acids (10 kcal/mole)
3. Hydrophobic interactions, most common, experienced by every amino acid (1-2 kcal/mole)
4. Van der Waals interactions that prevent multiple molecules from coming too close (1 kcal/mole)
5. Disulfide Bridge (51 kcal/mole)

### How do we predict protein structure?

Two methods: Experimental and Computational. Computational methods are still in their infancy.

Experimental methods;

X-Ray crystallography(standard): Form a crystallization of protein structures, which is complicated. But the crystal can be x-rayed to locate atoms within the crystal.

Computational methods:

**Ab Initio** - use only energy, geometry, kinematics

**Homology** - best match to a database of sequences.

Combinations of these like **Threading** and Meta-servers.

An Initio:

a. Sample the global conformation space

Lattice models/Discrete-state models

Molecular dynamics

These are useful for making predictions about how kinematics and energy functions operate in general. For example, with lattice models, you can judge the effects of misfolding.

See papers by Vijay Pande's group for reference.

b. Picking naive conformations with an energy function

Solvation model: how protein interacts with water.  
Pair interactions between amino acids.

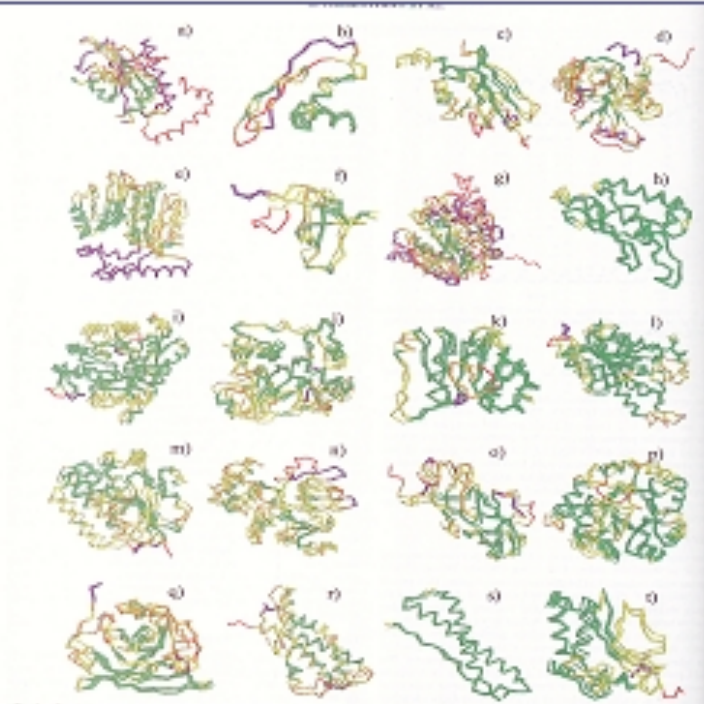
c. Predicting secondary structure

Local homology and Fragment libraries.

Use a database of structures and make matches. Accurate to about 70%.



## Homology-based Prediction



*Green areas show parts that are matched in the target protein and a given protein from the database. Other areas are those that do not match.*

Some tools that do this:

*ROSETTA*

i. *PSI-BLAST* - homology search  
Discard sequences with >25% homology

ii. *PHD*

For each 3-long and 9-long sequence fragment, get 25 fragments that match 'well'

### iii. Markov-Chain Monte Carlo method

Insert and remove iteratively one short structure fragment at a time.

Use a random configuration of fragments and iteratively take out a fragment, replace it with another and see if it improves (converge to a good energy) and employ a selection method to choose the one with high, convergent energy.

There are only 1000 different families of proteins (in 1992, by *Cyrus Choothiya*). Some of the reason must be that proteins do not have many ancestors, that is, evolutionary proteins are closely related.

Similarly, we can locate 429 domains that are very similar to 80% of domains found in animals, and 90% found in fungi and plants.

Domains evolve primarily by duplication when domains are close together or less commonly, by combinations of different proteins.

It is not hard to predict protein structure for a protein that is similar, to a degree of about 30% or more, to other proteins with known structure.

Often, it is easier to predict structure of alpha helices and beta sheets than loops. So a homology match might not provide much information about small differences between proteins given by loops, that might be much more important for function. In general, predicting function from structure is difficult and hard to generalize.

### Threading

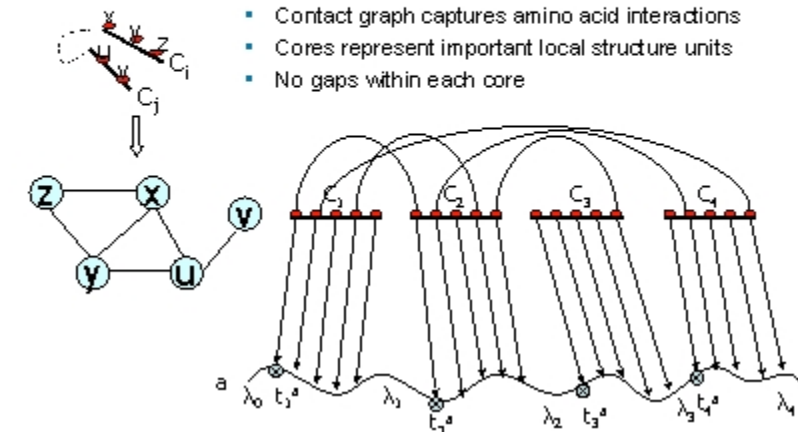
It uses structure to compute energy function during alignment. This might be useful when a protein has similar folds to known proteins but also deviations?

- Build a structural template database
- Define a sequence-structure energy function
- Apply a threading algorithm to query sequence
- Perform local refinement of secondary structure
- Report best results for structural model

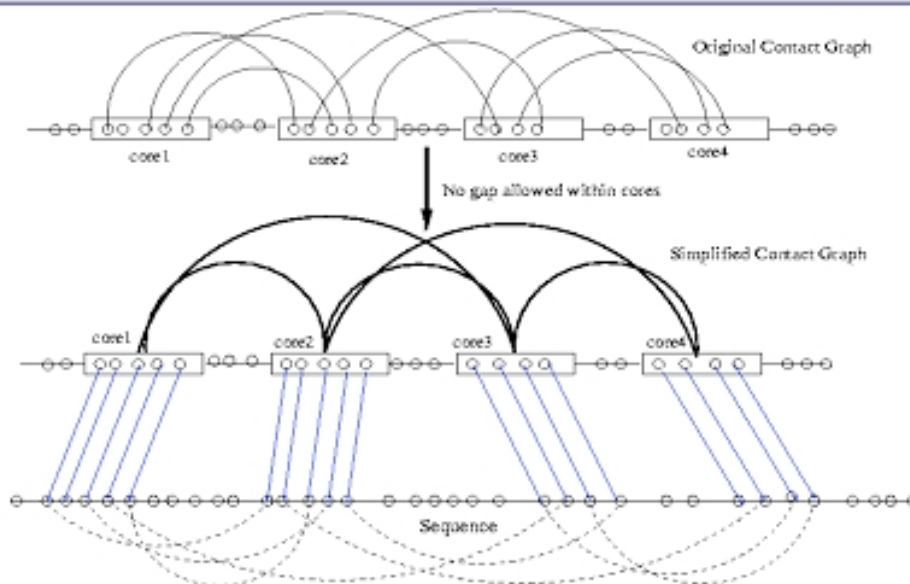
Main difference between homology and threading: Threading uses the structure to compute energy function during alignment.

Formulation:

## Threading – Formulation



## Threading – Formulation



How hard is Threading?

Unfortunately, the presence of non local interactions means that dynamic

programming is not an option.

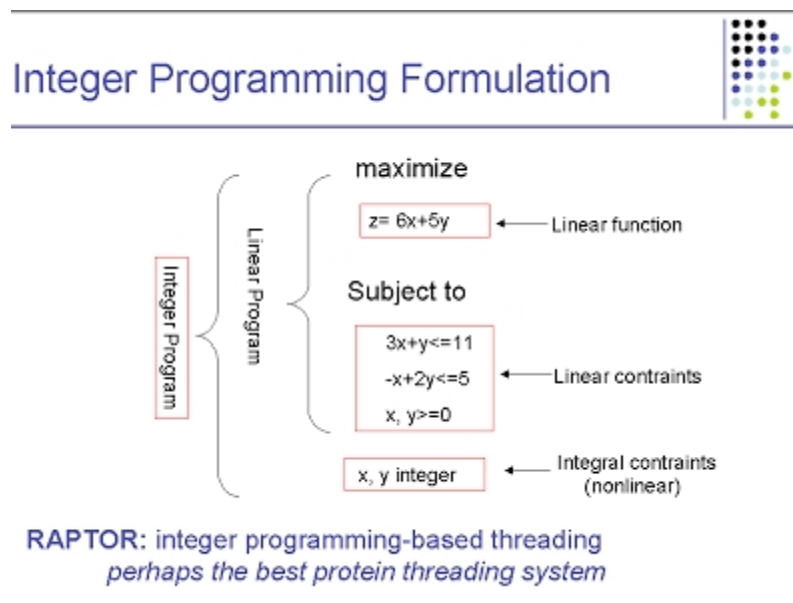
It is at least as hard as MAX-CUT: Given a graph  $G=(V,E)$ , find a cut  $(S,T)$  of  $V$  with maximum number of edges between  $S$  and  $T$ .

The Bad News: APX-complete even when each node has at most  $B$  edges (where  $B>2$ )

Batzoglou showed this but tragically, another paper with the same result reached the publication line first.

The best threading program:

*RAPTOR*: Integer Programming Formulation



Based on the contact map graph of the protein 3D structure template, the protein threading problem is formulated as a large scale integer programming (IP) problem. The IP formulation is then relaxed to a linear programming (LP) problem, and then solved by the canonical branch-and-bound method. The final solution is globally optimal with respect to energy functions. In particular, our energy function includes pairwise interaction preferences and allowing variable gaps which are two key factors in making the protein threading problem NP-hard. A surprising result is that, most of the time, the relaxed linear programs generate integral solutions directly.

See the paper *RAPTOR: optimal protein threading by linear programming* for reference.

So all these programs compete in CASP or CAFASP (full automated version of CASP), and RAPTOR does well for several targets. But if the target protein is similar (say 30%) to a known protein, Homology is generally best.

## Summary of current state of the art



Method	Sequence Similarity	Template Coverage	Accuracy	Difficulty	Comput. Expense
Homology	>30%	>90%	1-3 Å	Trivial	Seconds
FR/Homology	20-30%	>75%	2-5 Å	Easy	Minutes
Fold Recognition	<20%	>50%	3-10 Å	Moderate	Hours
Ab initio	<10%	0	5-20 Å	Hard	Days