Bioinformatics:
Secondary Structure Prediction

Prof. David Jones
d.t.jones@ucl.ac.uk
Possibly the greatest unsolved problem in molecular biology:
The Protein Folding Problem

MWMPPRPEEVARK
LRRLGFVERMAKG
GHRLYTHPDGRIV
VVPFHSGELPKGT
FKRILRDAGLTEE
EFHNL
biological multimeric state

INTERACTIONS

MULTIMERS

MUTANTS & SNPs

SURFACE SHAPE

FOLD

3D STRUCTURE

ELECTROSTATICS

CLUSTERS

LIGANDS

ligand & functional sites
catalytic clusters, mechanisms & motifs
enzyme active sites

Original image by J.M. Thornton
Why predict structure?

Growth of sequence and structure data banks
Growth of sequence and structure data banks

Number of Proteins vs Year

- Structures
- Sequences

Number of Proteins vs Year

- Structures
- Sequences
Protein Secondary Structure

- **ALPHA HELIX**
- **BETA SHEET** (STRAND)
- **COIL**
Secondary Structure Prediction

INPUT: 20-letter Amino Acid Alphabet

OUTPUT: 3-letter Secondary Structure Alphabet

H = Helix
E = Extended (strand)
C = Coil
Goal

• Take primary structure (sequence) and, using rules derived from known structures, predict the secondary structure (helix/strand/coil) that is most likely to be adopted by each residue
1st Generation Methods

- Based on statistical analysis of single amino acid residues
- Examples:
  - Chou & Fasman (1974)
  - Lim (1974)
  - Garnier, Osguthorpe & Robson (GOR) (1978)

- Although these methods were important early contributions to the field and are a good starting point for teaching purposes, in practice these methods are now considered ENTIRELY OBSOLETE

- DON’T USE THEM!
Chou-Fasman method

- Uses table of conformational parameters (propensities) determined primarily from measurements of secondary structure by CD spectroscopy
- Table consists of one “likelihood” for each structure for each amino acid
- For amino acid type A (e.g. leucine/serine/proline etc.) and structure type S (e.g. α helix), a propensity score can be calculated as follows:
  
  $$P_S = \frac{p(A|S)}{p(A)} = \frac{n_{A,S}/n_S}{n_A/n}$$
## Chou-Fasman propensities

(partial table)

<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>$P_\alpha$</th>
<th>$P_\beta$</th>
<th>$P_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glu</td>
<td>1.51</td>
<td>0.37</td>
<td>0.74</td>
</tr>
<tr>
<td>Met</td>
<td>1.45</td>
<td>1.05</td>
<td>0.60</td>
</tr>
<tr>
<td>Ala</td>
<td>1.42</td>
<td>0.83</td>
<td>0.66</td>
</tr>
<tr>
<td>Val</td>
<td>1.06</td>
<td>1.70</td>
<td>0.50</td>
</tr>
<tr>
<td>Ile</td>
<td>1.08</td>
<td>1.60</td>
<td>0.50</td>
</tr>
<tr>
<td>Tyr</td>
<td>0.69</td>
<td>1.47</td>
<td>1.14</td>
</tr>
<tr>
<td>Pro</td>
<td>0.57</td>
<td>0.55</td>
<td>1.52</td>
</tr>
<tr>
<td>Gly</td>
<td>0.57</td>
<td>0.75</td>
<td>1.56</td>
</tr>
</tbody>
</table>
Chou-Fasman method

• Calculation rules are somewhat *ad hoc*
• Example: Method for helix
  – Search for nucleating region where 4 out of 6 a.a. have $P_\alpha > 1.03$
  – Extend until 4 consecutive a.a. have an average $P_\alpha < 1.00$
  – If region is at least 6 a.a. long, has an average $P_\alpha > 1.03$, and average $P_\alpha >$ average $P_\beta$, consider region to be helix
• Unclear in what order the rules should be applied
2nd Generation Methods

• Secondary structure effects strongly depend on neighbouring residues
• 2\textsuperscript{nd} Gen methods therefore based on residue pairs or peptide segments
• Example:
  – GOR III (1987)
• The BIG NEWS, however, was the appearance of the first examples of MACHINE LEARNING in secondary structure prediction
• Originally, neural network methods were developed to model brain function i.e. they were intended to allow simulations of real networks of neurons. Hence the term *Artificial Neural Network*.

• Today these very simple models are obsolete in neuroscience research but instead have become very useful tools for finding patterns in data (data mining).
An artificial neural network (ANN) is a set of linked mathematical functions, which can be thought of as a network of switching units (artificial neurons) “connected” together according to a specific architecture. The objective of an artificial neural network is to learn how to transform inputs into meaningful outputs.
Trivial Example – The Perceptron

Inputs

% Alpha helix
34
.17

% Beta sheet
24
.15

Protein length
260
.01

Connection

Weights

Output

Neuron Output

0.8

“Probability of being a TIM barrel enzyme”

Prediction

Output = \frac{1}{1 + e^{-(\Sigma + C)}}

This is the activation function
Training and Using ANNs

• The procedure for training a neural network is called BACKPROPAGATION

• We start with complete random connection weights

• Then we present an input pattern from our training set to the network and compare the calculated output to what it should be

• We then repeat this for all the examples we have

Pseudocode for ANN training:

initialize network weights (usually small random values)
do forEach training example x
    prediction = neural-net-output(network, x) // forward pass
    actual = expected-output(x) // Calculate what output you expect to see
    compute error (prediction - actual) at the output units
    compute $\Delta w_h$ for all weights from hidden layer to output layer // backward pass
    compute $\Delta w_i$ for all weights from input layer to hidden layer // backward pass continued
    update network weights // input layer is not modified by the error estimate!
until all training examples classified correctly or another stopping criterion satisfied
return the final set of network weights
Training and Testing Sets

- The data we use to adjust the network weights is called the **TRAINING SET**.
- To make sure we have not over-fitted our network to our training set, we should test the network on a completely separate **TESTING SET**.
- This splitting of training and testing data is called **CROSS-VALIDATION** and is an important concept in statistics and machine learning as it allows us to predict how well a method is likely to work on entirely new data.
Trivial example of Neural Network
Training Data

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

The OUTPUT is 1 when both INPUTS are different.

This simple function CANNOT be learned by a perceptron because perceptrons can only learn linearly separable functions.
**XOR Function**

<table>
<thead>
<tr>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
f(a) = \begin{cases} 
1, & \text{for } a > \theta \\
0, & \text{for } a \leq \theta 
\end{cases}
\]

\[
f(w_1, w_2, w_3, w_4, w_5, w_6)
\]

A possible set of values for \(w_s\)

\[
(w_1, w_2, w_3, w_4, w_5, w_6)
\]

\[(0.6, -0.6, -0.7, 0.8, 1, 1)\]

\(\theta = 0.5\) for all units

By using “hidden units” we can extend perceptrons to cover more complex functions e.g. the XOR function.
Some real data....

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Desired Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A R N D C Q E G H I L K M F P S T W Y V</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0</td>
<td><strong>Coil</strong></td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td><strong>Strand</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0</td>
<td><strong>Coil</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0</td>
<td><strong>Strand</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td><strong>Strand</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0</td>
<td>1 <strong>Helix</strong></td>
</tr>
<tr>
<td>1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td><strong>Coil</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0</td>
<td><strong>Strand</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0</td>
<td><strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0</td>
<td><strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0</td>
<td><strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0</td>
<td>0 <strong>Strand</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 <strong>Helix</strong></td>
</tr>
</tbody>
</table>

We randomly select some of the data for use as our **TESTING SET** and keep the rest for our **TRAINING SET**.

And so on...
Predicting Secondary Structure by Machine Learning

Rather than just one residue at a time, we look at neighbouring residues as well!

Window of 15 residues

Classifier (neural network)

The goal is to predict the secondary structure for this central residue
Representation

• Usual representation is to use 21 inputs per amino acid:

Ala : 100000000000000000000
Val : 0000000000000000000010
--- : 0000000000000000000001
3rd Generation Neural Network Methods

PHD

First method to exceed average accuracy of 70% - pioneered the field, but now lags slightly behind best modern methods. Still available as a Web server.


PSIPRED (http://bioinf.cs.ucl.ac.uk/psipred)

Currently the best method for predicting secondary structure (>82% accuracy on average for latest version). Uses PSI-BLAST and a more sophisticated neural network model than PHD. Available both as a Web server and as a downloadable program. If you are only going to use one method, use this one!

3rd Generation Methods

- Exploit evolutionary information
- Based on conservation analysis of multiple sequence alignments (profiles or HMMs)
- Can extract some long-range information via accessibility patterns

- Conserved Hydrophobic Residues -> BURIED
- Variable Polar Residues -> EXPOSED
Residue Variability and Solvent Accessibility

In a family of proteins, positions that tolerate the widest choice of amino acid types across evolution are likely to have high solvent exposure.
Pros & Cons of 3rd Generation Methods

• PROs
  – High residue accuracy
  – Less underprediction of strands
  – Good quality segment end predictions

• CONs
  – Provides prediction for average FAMILY structure NOT THE STRUCTURE OF THE SPECIFIC TARGET SEQUENCE
PSIPRED

- Works directly on PSI-BLAST profiles (PSSMs)
- Uses 2 separate stages of neural networks
  - First network predicts secondary structure
  - Second network cleans outputs from 1st net
- Trained on profile from >3000 different proteins of known structure (taken from PDB)
- Observed secondary structures are defined by DSSP
DSSP (Define Secondary Structure of Proteins) is the *de facto* standard method for assigning secondary structure to the amino acids of a protein, given its atomic coordinates.

It begins by identifying the intra-backbone hydrogen bonds of the protein using a simple electrostatic definition.

Eight types of secondary structure are assigned. The $3_{10}$ helix (G), $\alpha$-helix (H) and $\pi$-helix (I) are recognised by having a particular sequence of hydrogen bonds three, four, or five residues apart respectively. Two types of beta sheet structure are recognised: an isolated beta bridge is denoted with a symbol B, and long runs of bridges are given the symbol E (for extended structure). Finally, T is used for turns, and S is used for regions of high curvature i.e. sharp bends in the polypeptide chain. A blank space indicates coil.

In secondary structure prediction, we typically consider G & H both as Helix states, B and E as Strand states and all other symbols as coil.

PSIPRED

Neural Networks

• 1st level network
  – 15x21 input units
  – 75 hidden units
  – 3 output units (H/E/C)

• 2nd level network
  – 15x4 input units
  – 55 hidden units
  – 3 output units (H/E/C)

These are BIG neural networks!
The PSIPRED Method (slightly simplified)

This is the raw Position Specific Score Matrix (profile) generated by PSI-BLAST

The raw values are normalised so that they range between 0 and 1

15 x 20 scaled inputs to 1st network

1st Network
315 inputs
75 hidden units
3 outputs

Window of 15 x 3 outputs fed to 2nd network

2nd Network
60 inputs
55 hidden units
3 outputs

Note: 60 inputs rather than 45 – why?

1st Network
315 inputs
75 hidden units
3 outputs

1st Network
315 inputs
75 hidden units
3 outputs

This is the raw Position Specific Score Matrix (profile) generated by PSI-BLAST

The raw values are normalised so that they range between 0 and 1

15 x 20 scaled inputs to 1st network

1st Network
315 inputs
75 hidden units
3 outputs

Window of 15 x 3 outputs fed to 2nd network

2nd Network
60 inputs
55 hidden units
3 outputs

Note: 60 inputs rather than 45 – why?

1st Network
315 inputs
75 hidden units
3 outputs

1st Network
315 inputs
75 hidden units
3 outputs

This is the raw Position Specific Score Matrix (profile) generated by PSI-BLAST

The raw values are normalised so that they range between 0 and 1

15 x 20 scaled inputs to 1st network

1st Network
315 inputs
75 hidden units
3 outputs

Window of 15 x 3 outputs fed to 2nd network

2nd Network
60 inputs
55 hidden units
3 outputs

Note: 60 inputs rather than 45 – why?

1st Network
315 inputs
75 hidden units
3 outputs

1st Network
315 inputs
75 hidden units
3 outputs

This is the raw Position Specific Score Matrix (profile) generated by PSI-BLAST

The raw values are normalised so that they range between 0 and 1

15 x 20 scaled inputs to 1st network

1st Network
315 inputs
75 hidden units
3 outputs

Window of 15 x 3 outputs fed to 2nd network

2nd Network
60 inputs
55 hidden units
3 outputs

Note: 60 inputs rather than 45 – why?
### 2<sup>nd</sup> Network: Filtering Raw Predictions from 1<sup>st</sup> Network

<table>
<thead>
<tr>
<th>Stage 1 outputs</th>
<th>Stage 2 outputs</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>49 T E 0.238 0.007 0.856</td>
<td>49 T E 0.204 0.009 0.807</td>
<td>49 T E</td>
</tr>
<tr>
<td>50 L E 0.038 0.004 0.951</td>
<td>50 L E 0.056 0.005 0.947</td>
<td>50 L E</td>
</tr>
<tr>
<td>51 V E 0.023 0.003 0.985</td>
<td>51 V E 0.024 0.006 0.972</td>
<td>51 V E</td>
</tr>
<tr>
<td>52 R E 0.036 0.003 0.966</td>
<td>52 R E 0.069 0.015 0.897</td>
<td>52 R E</td>
</tr>
<tr>
<td>53 V E 0.419 0.001 0.775</td>
<td>53 V E 0.283 0.013 0.616</td>
<td>53 V E</td>
</tr>
<tr>
<td>54 P C 0.808 0.011 0.060</td>
<td>54 P C 0.866 0.024 0.056</td>
<td>54 P C</td>
</tr>
<tr>
<td>55 G C 0.888 0.032 0.065</td>
<td>55 G C 0.901 0.083 0.029</td>
<td>55 G C</td>
</tr>
<tr>
<td>56 S C 0.412 0.328 0.166</td>
<td>56 S H 0.342 0.376 0.138</td>
<td>56 S H</td>
</tr>
<tr>
<td>57 W E 0.151 0.294 0.391</td>
<td>57 W H 0.205 0.543 0.267</td>
<td>57 W H</td>
</tr>
<tr>
<td>58 E H 0.057 0.817 0.141</td>
<td>58 E H 0.081 0.837 0.106</td>
<td>58 E H</td>
</tr>
<tr>
<td>59 I H 0.126 0.962 0.003</td>
<td>59 I H 0.039 0.932 0.009</td>
<td>59 I H</td>
</tr>
<tr>
<td>60 P H 0.018 0.997 0.001</td>
<td>60 P H 0.036 0.950 0.003</td>
<td>60 P H</td>
</tr>
<tr>
<td>61 V H 0.006 0.999 0.001</td>
<td>61 V H 0.028 0.967 0.002</td>
<td>61 V H</td>
</tr>
<tr>
<td>62 A H 0.003 0.999 0.002</td>
<td>62 A H 0.025 0.969 0.002</td>
<td>62 A H</td>
</tr>
</tbody>
</table>

The second network is able to clean up many of the little “blips” produced by the first network! Essentially this is an example of DATA SMOOTHING.
Common Measures of Secondary Structure Prediction Accuracy

• $Q_3$ scores give the percentage of correctly predicted residues across 3 states (H,E,C)
  – This is the most commonly used measure

• Other scores such as Matthew’s Correlation Coefficient try to identify accuracy for individual states (Coil, Strand, Helix) and are more sensitive to over-prediction e.g. if you predict all residues to be random coil you will get a $Q_3$ score of around 50% just because around 50% of residues in proteins are in random coil regions. However, the MCC scores will be close to zero!
Matthews Correlation Coefficient

Probably the more robust metric is the Matthews Correlation Coefficient (MCC). Here we calculate the MCC for predicted helix states:

\[ C_h = \frac{pn - ou}{\sqrt{(p + o)(p + u)(n + o)(n + u)}} \]

- \( p = \) patterns correctly assigned to helix
- \( n = \) patterns correctly assigned to not-helix
- \( o = \) patterns incorrectly assigned to helix
- \( u = \) patterns incorrectly assigned to not-helix

Methods are now so accurate (>80%) that it makes little difference whether you compare methods by \( Q_3 \) or MCC.

\( Q_3 \) tends to win out because it’s so easy to interpret!
Original (1999) PSIPRED Benchmark Results

Mean Q₃ score: 77.8% (>82% today)
Comparison of Generations by Average Q3 Scores

GEN 1    GEN 2    GEN 3

First use of neural networks

Current best methods

Chou & Fasman
GOR I
Qian & Sejnowski
PHD (1994)
PSIPRED

OBSOLETE!!
PSIPRED Example

http://bioinf.cs.ucl.ac.uk/psipred
Bioinformatics:
Threading and Protein Fold Recognition

Prof. David Jones
d.t.jones@ucl.ac.uk
Sequence Comparison

- > 30% Identity between two protein sequences usually implies common structure and possibly common function
- However, there are many exceptions to this “rule of thumb”
Similar Sequence $\implies$ Similar 3-D Structure
(RMSD = 2.1 Å, Seq. ID = 30%)

Ribonuclease MC1  Ribonuclease Rh
Homologous Structures

- **Heat Labile Enterotoxin**
  - Structure similarity: 79%

- **Cholera Toxin**
  - Structure similarity: 97%

- **Pertussis Toxin**
  - Structure similarity: 81%
Protein Structure Prediction Methods

- **Comparative modelling**
  - Requires: Known fold + clear homology

- **Fold recognition**
  - Requires: Known fold

- **Ab initio / new fold methods**
  - Requires: only target sequence

Increasing difficulty

Increasing accuracy/reliability
Tertiary Structure Prediction

■ BASIS: native fold is expected to be the conformation of lowest (free) energy
  ◆ True for small molecules
  ◆ Almost certainly true for small protein domains
  ◆ May well be true for larger protein domains but larger proteins may end up kinetically trapped in a low but not lowest free energy state

■ IMPLICATION: native fold can be found by defining a potential energy function and searching all conformations for the one with lowest energy
The Levinthal Paradox

For a protein sequence of length \( l \), the total number of possible chain conformations \( N \) is given by:

\[
N \approx 10^l
\]

>> Even if a protein was able to rearrange itself at the speed of light it would take \(~10^{75}\) years to locate the global energy minimum for a 100 residue protein.

>> Proteins evolve to have predefined folding pathways which they follow to the native structure
Fold recognition – a short cut to predicting protein tertiary structure

- Although there are vast number of possible protein structures, only a few have been observed in Nature
- The chance of a newly solved structure having a previously unknown fold is only ~10%

>> We might be able to predict protein structure by selecting from already observed folds (a “Multiple Choice Question” version of the protein folding problem!)
Protein Structure Prediction by Threading

Target Sequence: ADAQKAADNKKVNSWTCEDFLAVDESFQPT

Fold Library

Energy Terms
A Scoring Function for Threading

- We want a way of assessing the energy of a model i.e. a model based on an alignment of a sequence with a structure
- Energy functions based on physics do not work
- Let’s use a KNOWLEDGE-BASED STATISTICAL APPROACH

- Look at known structures and see how often particular features (e.g. atomic contacts between amino acids) occur. In other words, we start by estimating probabilities.
Native structure – oppositely charged side chains help stabilise the fold.
First Attempt Threading alignment

Ouch! Two negatively charged side chains can’t fit here because they would tend to destabilize this structure!
Threading alignment 2

Ah, that’s better - opposite charges again!
Converting Probabilities to Energy Estimates

- Probabilities are inconvenient for computer algorithms as we need to multiply them to combine them and we end up with very small numbers.
- Additive quantities (e.g. energies) are much easier to handle.

>> For many applications it is common to transform probabilities into additive energy-like quantities.
The Inverse Boltzmann Principle

The basis of generating statistical potentials from probabilities comes from the *Inverse Boltzmann principle*.

According to the normal Boltzmann principle, the probability of occurrence of a given conformational state of energy $E$ scales with the *Boltzmann factor* $e^{-E/RT}$, where $R$ is the gas constant (1.987 x 10$^{-3}$ kcal.mol$^{-1}$K$^{-1}$) and $T$ is the absolute temperature (e.g. room temperature).
Potentials of Mean Force

Count interactions of given type (e.g. alanine->serine beta-carbon to alpha-carbon) in real protein structures

Count interactions of same type in randomly generated protein structures or randomly selected sites (the reference state)

Ratio of probabilities provides an estimate of the free energy change according to the inverse Boltzmann equation:

\[ \Delta G = -RT \log \left( \frac{p(\text{interaction in real proteins})}{p(\text{interaction in decoy structures})} \right) - RT \log Z \]

These energies are sometimes referred to as POTENTIALS OF MEAN FORCE because we are not considering all of the different component forces (e.g. electrostatic/van der Waals etc.), but instead just an average overall force.
Calculating Potentials of Mean Force

Separation in Sequence = 4
Spatial Distance = 10 Å

How often do we see this configuration in folded proteins?

What about in unfolded protein chains?

e.g. Cα atoms
   (but could be any pair of atom types)
EXAMPLE: Short-range Cβ Pair Potential of Mean Force (Ala-Ala @ Sequence Separation 4)

What creates this apparently favourable interaction?

Here we have done the same calculation shown on the previous slide, but for a range of different spatial distances (4-10 Angstroms)
We can estimate the stability of a given protein model by summing potentials of mean force for all residue pairs

- Loops are sometimes ignored
- Single residue (solvation) terms usually included
- Can scale terms according to protein size

This method can also be used to spot mistakes in X-ray or NMR structures!
Partially correct Immunoglobulin Heavy Chain Fab Fragment Homology Model
Partially correct Immunoglobulin Heavy Chain Fab Fragment Model Coloured by a Potential of Mean Force
Search Problem: Threading

• Searching for the optimum mapping of sequence to structure while optimizing the sum of pair interactions is *NP-complete* (proven by R. Lathrop in 1994) i.e. an exhaustive search is needed to guarantee an optimal solution.

• This search process is called “THREADING” i.e. what is the best way of threading this sequence through this structure.

• In practice due to short range of pair potentials, heuristic solutions work fairly well:

  – Exhaustive search (not practical)
  – Dynamic programming.
  – Double dynamic programming.
  – Branch and bound.
  – Monte Carlo methods: simulated annealing, Gibbs sampling, genetic algorithms.
Threading Methods in Practice

- Compared to comparative modelling, threading methods can produce models where there is no detectable homologue to be found in PDB.
- The simplifying assumption that the backbone of the structure does not change when the sequence changes also results in poor recognition of folds (~30% reliability) and inaccurate models (i.e. inaccurate alignments).
- For distant homologues, however, we can combine information from sequence profiles and secondary structure to improve alignments and make much better models than from threading potentials alone.
- This is what most modern fold recognition or threading methods do.
Examples of Fold Recognition Methods

- THREADER
  - The original (1992) threading method
  - Uses threading potentials and predicted secondary structure
- GenTHREADER
  - Update to the original threading methods
  - Combines threading potentials and predicted secondary structure of THREADER with profile-profile matching
- 3D-PSSM
  - Combines sequence profiles and predicted secondary structure
- HHpred
  - Combines HMMs with predicted secondary structure
- MUSTER
  - Combines multiple sources of information, including secondary structure and sequence-profile information
Comparing Sequence Alignment to Fold Recognition in Homology Searching

Run times to search a set of 77977 bacterial sequences for matches to Adenylate Kinase structure

<table>
<thead>
<tr>
<th>Method</th>
<th>Hits</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST</td>
<td>61</td>
<td>8 sec</td>
</tr>
<tr>
<td>PSIBLAST</td>
<td>171</td>
<td>2 min</td>
</tr>
<tr>
<td>GenTHREADERER</td>
<td>450</td>
<td>18 min</td>
</tr>
</tbody>
</table>

All hits at same error rate (1%)

Although slower, fold recognition can find many more homologues than sequence comparison methods – even PSI-BLAST.
An Early Example of Success in applying Fold Recognition to Genome Annotation
Our Example

- ORF HI0073 from *H. influenzae*
- 114 a.a. long
- Function was UNKNOWN
- BLAST against UNIPROT just turned up the following:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
<th>Identity (%)</th>
<th>Bit Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>E64000</td>
<td>hypothetical protein HI0073 - Haemophilus influenzae (strain HI0073)</td>
<td>180</td>
<td>9e-46</td>
</tr>
<tr>
<td>A71149</td>
<td>hypothetical protein PH0403 - Pyrococcus horikoshii</td>
<td>101</td>
<td>7e-22</td>
</tr>
<tr>
<td>H70345</td>
<td>conserved hypothetical protein aq_507 - Aquifex aeolicus</td>
<td>99</td>
<td>2e-21</td>
</tr>
<tr>
<td>F72600</td>
<td>hypothetical protein APE1270 - Aeropyrum pernix (strain K1)</td>
<td>50</td>
<td>1e-06</td>
</tr>
<tr>
<td>C75046</td>
<td>hypothetical protein PAB0900 - Pyrococcus abyssi (strain ...</td>
<td>43</td>
<td>2e-04</td>
</tr>
<tr>
<td>C64354</td>
<td>hypothetical protein MJ0435 - Methanococcus jannaschii</td>
<td>40</td>
<td>0.002</td>
</tr>
<tr>
<td>D64375</td>
<td>hypothetical protein MJ0604 - Methanococcus jannaschii</td>
<td>38</td>
<td>0.006</td>
</tr>
<tr>
<td>H90364</td>
<td>conserved hypothetical protein [imported] - Sulfolobus so...</td>
<td>37</td>
<td>0.011</td>
</tr>
<tr>
<td>S62544</td>
<td>hypothetical protein SPAC12G12.13c - fission yeast (Schiz...</td>
<td>37</td>
<td>0.011</td>
</tr>
<tr>
<td>C90279</td>
<td>conserved hypothetical protein [imported] - Sulfolobus so...</td>
<td>37</td>
<td>0.015</td>
</tr>
<tr>
<td>H64462</td>
<td>hypothetical protein MJ1305 - Methanococcus jannaschii</td>
<td>36</td>
<td>0.017</td>
</tr>
<tr>
<td>C69282</td>
<td>conserved hypothetical protein AF0259 - Archaeoglobus ful...</td>
<td>36</td>
<td>0.019</td>
</tr>
</tbody>
</table>

.
The PSIPRED Protein Structure Prediction Server

We suggest that you do not bookmark this page as it is liable to move. It is best to access the server via the PSIPRED home page, which has more information about the methods and a full reference list.

Input Sequence

Choose Prediction Method

Output Options

Submit Sequence

E-mail address

Password (only required for commercial e-mail addresses)

Short name for sequence

Predict  Clear form
<table>
<thead>
<tr>
<th>Conf.</th>
<th>Net Score</th>
<th>E-value</th>
<th>PairE</th>
<th>SolvE</th>
<th>Aln Score</th>
<th>Aln Len</th>
<th>Str Len</th>
<th>Seq Len</th>
<th>Alignment</th>
<th>FSSP Code</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>0.570</td>
<td>0.143</td>
<td>-70.9</td>
<td>9.5</td>
<td>216.0</td>
<td>110</td>
<td>511</td>
<td>114</td>
<td>1fa0B0</td>
<td>117.1.1.1.1.1</td>
<td><img src="image1.png" alt="Structure" /></td>
</tr>
<tr>
<td>LOW</td>
<td>0.566</td>
<td>0.156</td>
<td>-70.9</td>
<td>6.6</td>
<td>186.0</td>
<td>100</td>
<td>158</td>
<td>114</td>
<td>1fa0B1</td>
<td>117.1.1.1.1.1</td>
<td><img src="image2.png" alt="Structure" /></td>
</tr>
<tr>
<td>LOW</td>
<td>0.560</td>
<td>0.182</td>
<td>-47.1</td>
<td>10.8</td>
<td>222.0</td>
<td>110</td>
<td>345</td>
<td>114</td>
<td>1f5aA2</td>
<td>-</td>
<td><img src="image3.png" alt="Structure" /></td>
</tr>
<tr>
<td>LOW</td>
<td>0.529</td>
<td>0.402</td>
<td>76.1</td>
<td>9.8</td>
<td>92.0</td>
<td>108</td>
<td>253</td>
<td>114</td>
<td>1knyA0</td>
<td>1.5.6.1.1.1</td>
<td><img src="image4.png" alt="Structure" /></td>
</tr>
<tr>
<td>LOW</td>
<td>0.506</td>
<td>0.726</td>
<td>-13.6</td>
<td>1.4</td>
<td>60.0</td>
<td>92</td>
<td>108</td>
<td>114</td>
<td>1diiA3</td>
<td>-</td>
<td><img src="image5.png" alt="Structure" /></td>
</tr>
<tr>
<td>GUESS</td>
<td>0.488</td>
<td>1.128</td>
<td>-61.3</td>
<td>0.9</td>
<td>53.0</td>
<td>99</td>
<td>326</td>
<td>114</td>
<td>1bpvA0</td>
<td>117.1.1.2.1.1</td>
<td><img src="image6.png" alt="Structure" /></td>
</tr>
<tr>
<td>GUESS</td>
<td>0.483</td>
<td>1.294</td>
<td>-20.3</td>
<td>0.8</td>
<td>52.0</td>
<td>104</td>
<td>402</td>
<td>114</td>
<td>1diiA0</td>
<td>151.1.4.1.2.1</td>
<td><img src="image7.png" alt="Structure" /></td>
</tr>
</tbody>
</table>
Conclusions

- We predicted that HI0073 was a probable nucleotidyl transferase (now confirmed)
- Consistent fold recognition results
  - Match to 1FA0 Chain B
- Secondary structure in reasonable agreement
- Functionally Important Residues were CONSERVED