Outline

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction
Outline

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction
HIERARCHY OF PROTEIN STRUCTURE

1. Primary

2. Secondary

3. Tertiary

4. Quaternary
X-Ray Crystallography

- crystallize and immobilize single, perfect protein
- bombard with X-rays, record scattering diffraction patterns
- determine electron density map from scattering and phase via Fourier transform:

\[ F(\Delta k) = \frac{V}{2\pi} \int_{x=0}^{x=1} \int_{y=0}^{y=1} \int_{z=0}^{z=1} \rho(x,y,z) e^{i\Delta k (xa+yb+zc)} \, dx \, dy \, dz \]

- use electron density and biochemical knowledge of the protein to refine and determine a model

"All crystallographic models are not equal. ... The brightly colored stereo views of a protein model, which are in fact more akin to cartoons than to molecules, endow the model with a concreteness that exceeds the intentions of the thoughtful crystallographer. It is impossible for the crystallographer, with vivid recall of the massive labor that produced the model, to forget its shortcomings. It is all too easy for users of the model to be unaware of them. It is also all too easy for the user to be unaware that, through temperature factors, occupancies, undetected parts of the protein, and unexplained density, crystallography reveals more than a single molecular model shows."

NMR Spectroscopy

protein in aqueous solution, motile and tumbles/vibrates with thermal motion

NMR detects chemical shifts of atomic nuclei with non-zero spin, shifts due to electronic environment nearby

determine distances between specific pairs of atoms based on shifts, “constraints”

• determine distances between specific pairs of atoms based on shifts, “constraints”

• use constraints and biochemical knowledge of the protein to determine an ensemble of models

using constraints to determine secondary structure
Outline

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction
PDB Growth
Only a few folds are found in nature

PROTEINS

One thousand families for the molecular biologist

Cyrus Chothia

How many families of proteins are there? By putting together the information to be found in papers published over the past few months we can make an initial estimate, and my calculation suggests that the large majority of proteins come from no more than one thousand families.

Proteins are clustered into families

<table>
<thead>
<tr>
<th>Genome projects</th>
<th>Total number of genes</th>
<th>Genes related to those previously determined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td></td>
<td>Ref.</td>
</tr>
<tr>
<td>Caenorhabditis elegans chromosome III (part)</td>
<td>32</td>
<td>14 (44%)</td>
</tr>
<tr>
<td>Yeast chromosome III</td>
<td>182</td>
<td>52–66 (29–36%)</td>
</tr>
<tr>
<td>chromosome IX (part)</td>
<td>46</td>
<td>15 (33%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Total number of clones</th>
<th>Clones related to previously determined protein sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human brain</td>
<td>~1,400</td>
<td>406 (~30%)</td>
</tr>
<tr>
<td>Caenorhabditis elegans</td>
<td>512 (34%)</td>
<td>210 (36%)</td>
</tr>
<tr>
<td>St Louis–Cambridge NIH</td>
<td>585</td>
<td>512 (34%)</td>
</tr>
</tbody>
</table>

TABLE 2 Genome projects

<table>
<thead>
<tr>
<th>Species</th>
<th>Approximate number of genes</th>
<th>Tentative date of completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Escherichia coli</td>
<td>4,000</td>
<td>1995–98</td>
</tr>
<tr>
<td>Yeast</td>
<td>7,000</td>
<td>2000</td>
</tr>
<tr>
<td>Caenorhabditis elegans</td>
<td>15,000</td>
<td>2000</td>
</tr>
<tr>
<td>Human</td>
<td>50–100,000</td>
<td>2015</td>
</tr>
</tbody>
</table>
Protein classification

- Number of protein sequences grows exponentially
- Number of solved structures grows exponentially
- Number of new folds identified very small (and close to constant)
- Protein classification can
  - Generate overview of structure types
  - Detect similarities (evolutionary relationships) between protein sequences
  - Help predict 3D structure of new protein sequences
Structure Classification Databases

- **SCOP**
  - Manual classification (A. Murzin)
  - scop.berkeley.edu

- **CATH**
  - Semi manual classification (C. Orengo)
  - www.biochem.ucl.ac.uk/bsm/cath

- **FSSP**
  - Automatic classification (L. Holm)
  - www.ebi.ac.uk/dali/fssp/fssp.html
Protein structure classification: SCOP
Structural Classification of Proteins (SCOP)
http://scop.berkeley.edu/

- **Class**
  - Similar secondary structure content
  - All $\alpha$, all $\beta$, alternating $\alpha/\beta$ etc

- **Fold (Architecture)**
  - Major structural similarity
  - SSE’s in similar arrangement

- **Superfamily (Topology)**
  - Probable common ancestry
  - HMM family membership

- **Family**
  - Clear evolutionary relationship
  - Pairwise sequence similarity $> 25\%$
Major classes in SCOP

• Classes
  – All α proteins
  – All β proteins
  – α and β proteins (α/β)
  – α and β proteins (α+β)
  – Multi-domain proteins
  – Membrane and cell surface proteins
  – Small proteins
  – Coiled coil proteins
**α/β alternating**
- Parallel β sheets, β-α-β units

**α + β**
- Anti-parallel β sheets, segregated α and β regions
- Helices mostly on one side of sheet
Folds

- >~50% secondary structure elements arranged in the same order in sequence and in 3D
- No evolutionary relation
Superfamilies

- Proteins which are (remotely) evolutionarily related
  - Sequence similarity low
  - Share function
  - Share special structural features
- Relationships between members of a superfamily may not be readily recognizable from the sequence alone
Families

- Proteins whose evolutionarily relationship is readily recognizable from the sequence (>~25% sequence identity)
- Families are further subdivided into Proteins
- Proteins are divided into Species
  - The same protein may be found in several species
Protein Classification

- Given a new protein sequence, can we place it in its “correct” position within an existing protein hierarchy?
Outline

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction
Protein Structure Comparison

• Global versus local alignment

• Measuring protein shape similarity

• Protein structure superposition

• Protein structure alignment
Global versus Local

Global alignment
Global versus Local (2)

motif

Local alignment
Measuring protein structure similarity

Given two “shapes” or structures A and B, we are interested in defining a distance, or similarity measure between A and B.

- Visual comparison
- Dihedral angle comparison
- Distance matrix
- RMSD (root mean square distance)
Comparing dihedral angles

*Torsion angles* ($\phi, \psi$) are:
- local by nature
- invariant upon rotation and translation of the molecule
- compact ($O(n)$ angles for a protein of $n$ residues)

*But...*

Add 1 degree to all $\phi$, $\psi$
**Distance matrix**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3.8</td>
<td>6.0</td>
<td>8.1</td>
</tr>
<tr>
<td>2</td>
<td>3.8</td>
<td>0</td>
<td>3.8</td>
<td>5.9</td>
</tr>
<tr>
<td>3</td>
<td>6.0</td>
<td>3.8</td>
<td>0</td>
<td>3.8</td>
</tr>
<tr>
<td>4</td>
<td>8.1</td>
<td>5.9</td>
<td>3.8</td>
<td>0</td>
</tr>
</tbody>
</table>
Distance matrix (2)

• **Advantages**
  - invariant with respect to rotation and translation
  - can be used to compare proteins

• **Disadvantages**
  - the distance matrix is $O(n^2)$ for a protein with $n$ residues
  - comparing distance matrix is a hard problem
  - insensitive to chirality
Root Mean Square Distance (RMSD)

To compare two sets of points (atoms) $A=\{a_1, a_2, \ldots a_N\}$ and $B=\{b_1, b_2, \ldots, b_N\}$:

- Define a 1-to-1 correspondence between $A$ and $B$

  for example, $a_i$ corresponds to $b_i$, for all $i$ in $[1,N]$

- Compute RMS as:

$$RMS(A, B) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d(a_i, b_i)^2}$$

$d(A_i, B_i)$ is the Euclidian distance between $a_i$ and $b_i$. 
Protein Structure Superposition

• Simplified problem: we know the correspondence between set A and set B
• We wish to compute the rigid transformation $T$ that best align $a_1$ with $b_1$, $a_2$ with $b_2$, ..., $a_N$ with $b_N$
• The error to minimize is defined as:

$$\mathcal{E} = \min_T \sum_{i=1}^{N} \| T(a_i) - b_i \|^2$$
Protein Structure Superposition (2)

• A rigid-body transformation $T$ is a combination of a translation $t$ and a rotation $R$: $T(x) = Rx + t$

• The quantity to be minimized is:

$$
\varepsilon = \min_{t,R} \sum_{i=1}^{N} \|Ra_i - b_i + t\|^2
$$
The translation part

$\varepsilon$ is minimum with respect to $t$ when:

$$\frac{\partial \varepsilon}{\partial t} = 2 \sum_{i=1}^{N} (Ra_i - b_i + t) = 0$$

Then:

$$t = -R \left( \sum_{i=1}^{N} a_i \right) + \sum_{i=1}^{N} b_i$$

If both data sets A and B have been centered on 0, then $t = 0$ !

**Step 1:** Translate point sets A and B such that their centroids coincide at the origin of the framework
The rotation part (1)

Let $\mu_A$ and $\mu_B$ be the centers of A and B, and $A'$ and $B'$ the matrices containing the coordinates of the points of A and B centered on O:

$$
\mu_A = \frac{1}{N} \sum_{i=1}^{N} a_i
$$

$$
\mu_B = \frac{1}{N} \sum_{i=1}^{N} b_i
$$

$$
A = \begin{bmatrix}
    a_1 - \mu_A & a_2 - \mu_A & \ldots & a_N - \mu_A
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
    b_1 - \mu_B & b_2 - \mu_B & \ldots & b_N - \mu_B
\end{bmatrix}
$$

Build covariance matrix:

$$
C = AB^T
$$

$$
\begin{bmatrix}
    \text{3xN} \\
    \text{Nx3}
\end{bmatrix}
\begin{bmatrix}
    \text{x} \\
    \text{3x3}
\end{bmatrix}
$$
The rotation part (2)

Compute SVD (Singular Value Decomposition) of C:

\[
C = UDV^T
\]

\[U \text{ and } V \text{ are orthogonal matrices, and } D \text{ is a diagonal matrix containing the singular values.}
\]
\[U, V \text{ and } D \text{ are } 3 \times 3 \text{ matrices}
\]

Define \( S \) by:

\[
S = \begin{cases} 
  I & \text{if } \det(C) > 0 \\
  \text{diag}\{1,1,-1\} & \text{otherwise}
\end{cases}
\]

\[R = USV^T
\]
The algorithm

1. Center the two point sets A and B

2. Build covariance matrix:
   \[ C = AB^T \]

3. Compute SVD (Singular Value Decomposition) of C:
   \[ C = UDV^T \]

4. Define S:
   \[ S = \begin{cases} I & \text{if } \det(C) > 0 \\ \text{diag}\{1,1,-1\} & \text{otherwise} \end{cases} \]

5. Compute rotation matrix
   \[ R = USV^T \]

6. Compute RMSD:
   \[ RMSD = \sqrt{ \frac{1}{N} \left( \sum_{i=1}^{N} a_i^2 + \sum_{i=1}^{N} b_i^2 - 2 \sum_{i=1}^{3} d_i s_i \right) } \]
Example: NMR structures

Superposition of NMR Models

1AW6
Protein Structure Alignment

The Problem:

Given two sets of points $A=(a_1, a_2, \ldots, a_n)$ and $B=(b_1, b_2, \ldots, b_m)$ in 3D space, find the optimal subsets $A(P)$ and $B(Q)$ with $|A(P)|=|B(Q)|$, and find the optimal rigid body transformation between the two subsets $A(P)$ and $B(Q)$ that minimizes a given distance metric $D$ over all possible rigid body transformation $G$, i.e.

$$\min_{G} \{D(A(P) - G(B(Q)))\}$$

The two subsets $A(P)$ and $B(Q)$ define a “correspondence”, and $p = |A(P)|=|B(Q)|$ is called the correspondence length.
Protein Structure Alignment

Iterate $N$ times:

1. Set **Correspondence** $C$ to a **seed** correspondence set (small set sufficient to generate an alignment transform)
2. Compute the alignment transform $G$ for $C$ and apply $G$ to the second protein $B$
3. Update $C$ to include all pairs of features that are close apart
4. If $C$ has changed, then return to Step 2
Protein Structure Alignment: Example
Existing Software

• DALI (Holm and Sander, 1993)
• SSAP (Orengo and Taylor, 1989)
• STRUCTAL (Levitt et al, 1993)
• VAST [Gibrat et al., 1996]
• LOCK [Singh and Brutlag, 1996]
• CE [Shindyalov and Bourne, 1998]
• SSM [Krissinel and Henrik, 2004]
• …
Outline

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction
Why do we need computational approaches?

- The goal of research in the area of structural genomics is to provide the means to characterize and identify the large number of protein sequences that are being discovered.

- Knowledge of the three-dimensional structure helps in the rational design of site-directed mutations, can be of great importance for the design of drugs, greatly enhances our understanding of how proteins function and how they interact with each other, for example, explain antigenic behaviour, DNA binding specificity, etc.

- Structural information from x-ray crystallographic or NMR results obtained much more slowly, techniques involve elaborate technical procedures, many proteins fail to crystallize at all and/or cannot be obtained or dissolved in large enough quantities for NMR measurements, The size of the protein is also a limiting factor for NMR.

- With a better computational method this can be done extremely fast.
Computational methods for Protein Structure Prediction

- Homology or Comparative Modeling
- Fold Recognition or threading Methods
- Ab initio methods that utilize knowledge-based information
- Ab initio methods without the aid of knowledge-based information
Homology Modeling Process

- Template recognition
- Alignment
- Determining structurally conserved regions
- Backbone generation
- Building loops or variable regions
- Conformational search for side chains
- Refinement of structure
- Validating structures
Template Recognition

• First we search the related proteins sequence(templates) to the target sequence in any structural database of proteins

• The accuracy of model depends on the selection of proper template

• FASTA and BLAST from EMBL-EBI and NCBI can be used

• This gives a probable set of templates but the final one is not yet decided

• After initial alignments and finding structurally conserved regions among templates, we choose the final template
Sequence alignment is central technique in homology modeling

- Used in determining which areas of the reference proteins are conserved in sequence

- Hence suggesting where the reference proteins may also be structurally conserved

- After SCRs are found, it is used to establish one to one correspondence between the amino acids of reference proteins and the target in SCRs

- Thus providing basis of the transforming of coordinates from the reference to the model
The First Developed Algorithm

- Needleman and Wunch algorithm for pairwise sequence alignment
- It is based on Dynamic Programming Algorithm
- It's a Global Alignment approach
Dynamic Programming Algorithm

- A dynamic programming algorithm solves a problem by combining solutions to subproblems that are computed once and saved in a table or matrix.
- The basic idea behind dynamic programming is the organization of work in order to avoid repetition of work already done.
- DPAs are typically used when a problem has many possible solutions and an optimal one has to be found.
Dynamic Programming Algorithm

Mathematical formulation

$$F(i,j) = \text{MAX}\{\begin{array}{l}
F(i-1, j-1) + s(x_i, y_j), \\
F(i-1, j) - d, \\
F(i, j-1) - d
\end{array}\}$$

Where $F(i,j)$ is the value in cell $(i,j)$; $s$ is the score for that match in the table; $d$ is the gap penalty
Dynamic Programming Algorithm

\[ F(i-1,j-1) + s(S_i, T_j) \]

\[ F(i,j-1) - d \]
Dynamic Programming Algorithm

\[ F(i, j) = \max \{ F(i-1, j-1) + s(S_i, T_j), F(i-1, j) - d, F(i, j-1) - d \} \]
Dynamic Programming Algorithm

Example alignment:
ELVIS LIVES?

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>L</th>
<th>V</th>
<th>I</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Programming Algorithm

**BLOSUM62 substitution matrix**

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>P</th>
<th>A</th>
<th>G</th>
<th>N</th>
<th>D</th>
<th>E</th>
<th>Q</th>
<th>H</th>
<th>R</th>
<th>K</th>
<th>N</th>
<th>L</th>
<th>V</th>
<th>F</th>
<th>Y</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>9</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>S</td>
<td>-1</td>
<td>4</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>-1</td>
<td>1</td>
<td>7</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>G</td>
<td>-3</td>
<td>0</td>
<td>1</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
<td>-4</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>N</td>
<td>-3</td>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>D</td>
<td>-3</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>-4</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>Q</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>H</td>
<td>-3</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>R</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>K</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>M</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>I</td>
<td>-1</td>
<td>2</td>
<td>-2</td>
<td>-3</td>
<td>1</td>
<td>-4</td>
<td>3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
</tr>
<tr>
<td>L</td>
<td>-1</td>
<td>2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>4</td>
<td>3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>V</td>
<td>-1</td>
<td>2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>4</td>
<td>3</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>F</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-4</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Y</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
</tr>
<tr>
<td>W</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-4</td>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
</tbody>
</table>

\[ s(S_i, T_j) \]
Dynamic Programming Algorithm

### Simple global alignment

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>L</th>
<th>V</th>
<th>I</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>-2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>-2</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Dynamic Programming Algorithm

Traceback step

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>L</th>
<th>V</th>
<th>I</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>-2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>-2</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>V</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Albert Einstein College of Medicine
OF Yeshiva University
Dynamic Programming Algorithm

Resulting alignment

EL-VIS
-ELVIS

ELVI-S
-ELVIS

Score:
-2+4+3+3-2+4
=10
The Modified Version Of Needleman Wunch

- Smith Waterman algorithm is modified Needleman Wunch
- It is for local alignment
- Locate the best local alignment between two sequences
- What is Global and Local Alignment
- In global, we try to find similarity in whole sequence
- In local, we try to find small similar segments within sequences
Local Alignment

- Comparing sequences of different length
- Proteins are from different protein families

Tools based on local alignment

- BLAST & FASTA – alignment against databases
- LALIGN & EMBOSS align – alignment of two sequences
- Infact there are more tools, these are the widely used
Dynamic Programming Algorithm

Local alignment

- Differences from global alignments
  - Minimum value allowed = 0
    - Corresponds to starting a new alignment
  - The alignment can end anywhere in the matrix, not just the bottom right corner
    - To find the best alignment, find the best score and trace back from there
  - Expected score for a random match MUST be negative
Dynamic Programming Algorithm

Local alignment formula

\[ F(i,j) = \text{MAX} \{ \]
\[ 0, \]
\[ F(i-1, j-1) + s(x_i, y_j), \]
\[ F(i-1, j) - d, \]
\[ F(i, j-1) - d \]
\[ \} \]
Dynamic Programming Algorithm

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>M</th>
<th>A</th>
<th>L</th>
<th>L</th>
<th>I</th>
<th>S</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Dynamic Programming Algorithm

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>M</th>
<th>A</th>
<th>L</th>
<th>L</th>
<th>I</th>
<th>S</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>M</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>
Dynamic Programming Algorithm

## Resulting alignments

<table>
<thead>
<tr>
<th>ALLISH</th>
<th>ALLISH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIGN</td>
<td>A-LIGN</td>
</tr>
</tbody>
</table>

Score:
4+4-2+4+0+1
=11

Score:
4-2+4+4+0+1
=11
Dynamic Programming Algorithm

Alignment with affine gap scores

- Affine gap scores
  - “affine” means that the penalty for a gap is computed as a linear function of its length
  - Have a gap opening penalty
  - Also have a less prohibitive gap extension penalty
- Have to keep track of multiple values for each pair of residue coefficients i and j in place of the single value F(i,j)
  - We keep two matrices, M and I
Dynamic Programming Algorithm

Affine gap score formula

\[ M(i, j) = \max \left\{ \begin{array}{l}
M(i-1, j-1) + s(x_i, y_j), \\
I(i-1, j-1) + s(x_i, y_j)
\end{array} \right\} \]

\[ I(i, j) = \max \left\{ \begin{array}{l}
M(i, j-1) - d, \\
I(i, j-1) - e, \\
M(i-1, j) - d, \\
I(i-1, j) - e
\end{array} \right\} \]
Comparison Of Different Algorithms

- **Traditional algorithms**
  - Find optimal alignment under a specific scoring criterion that includes the scoring matrix and gap penalties
  - Optimal alignment is quite often not the true biological alignment (Argos et al, 1991, Agarwal and States, 1996)

- **Heuristic algorithms**
  - Heuristic search tools find the optimal alignment with high probability and are less computationally expensive
  - HMM based search methods improve both the sensitivity and selectivity of sequence database searches, using position dependent scores to characterize and build a model for an entire family of sequences
  - Probabilistic Smith-waterman is based on HMM for a single sequence, more accurate from others for complete length protein query sequences in large protein family but poor when used with partial length query sequence
Multiple Sequence Alignment

- This is all about pairwise alignment

- In general homology modeling, we would like to include more than two protein references for the template protein

- It helps in finding conserved domains among similar reference proteins

- Therefore providing more information about structurally conserved domains in sequences

**Multiple Sequence Alignment - Methods**

- Multiple alignment is more difficult than pairwise alignment because the number of possible alignments increases exponentially with the number of sequences to be aligned

- No ideal method exists, several heuristic algorithms are being used

- Simple way is to use Needleman and Wunch algorithm for pairwise alignment in multidimensional space

- Disadvantage of this is exponential increase of memory usage and time consumption
Determining Structurally Conserved Regions (SCRs)

- When two or more reference protein structures are available
- Establish structural guidelines for the family of proteins under consideration
- First step in building a model protein by homology is determining what regions are structurally conserved or constant among all the reference proteins
- Target protein is supposed to assume the same conformation in conserved regions
Structurally Conserved Regions

- SCRs are regions in all proteins of a particular family that are nearly identical in structures.
- Tend to be at inner cores of the proteins
- Usually contain alpha-helices and beta sheets

There are generally two main approaches

- Constructing c-alpha distance matrix
- Aligning vectors of secondary structure units
Alignment of Target Protein with SCR

- After doing alignments and finding SCRs
- We align the unknown sequence with the aligned reference proteins with the knowledge of SCRs
- Assignment of coordinates within conserved regions is done
- SCR cannot contain insertions and deletions
Assignment of coordinates within conserved region

- Once the correspondence between amino acids in the reference and model sequences has been made, the coordinates for an SCR can be assigned.
- The reference proteins' coordinates are used as a basis for this assignment.

- Where the side chains of the reference and model proteins are the same at corresponding locations along the sequence, all the coordinates for the amino acid are transferred.

- Where they differ, the backbone coordinates are transferred, but the side chain atoms are automatically replaced to preserve the model protein's residue types.
Assignment of coordinates in loop or variable region

Two main methods

- Finding similar peptide segments in other proteins
- Generating a segment de-novo
Assignment of coordinates in loop or variable region

Finding similar peptide segment in other proteins

- Advantage: all loops found are guaranteed to have reasonable internal geometries and conformations

- Disadvantage: may not fit properly into the given model protein’s framework

In this case, de-novo method is advisable
Selection Of Loops

- Check the loops on the basis of steric overlaps
- A specified degree of overlap can be tolerated
- Check the atoms within the loop against each other
- Then check loop atoms against rest of the protein’s atoms
Side Chain Conformation Search

- With bond lengths, bond angles and two rotable backbone bonds per residue $\phi$ and $\psi$, it's very difficult to find the best conformation of a side chain.

- In addition, side chains of many residues have one or more degree of freedom.

- Hence Side chain conformational search in loop regions is must.

- Side chain residues replaced during coordinate transformations should also be checked.
Selection Of Good Rotamer

- Fortunately, statistical studies show side chains adopt only a small number of many possible conformations.

- The correct rotamer of a particular residue is mainly determined by local environment.

- Side chains generally adopt conformations where they are closely packed.
Selection Of Good Rotamer … Contd

It is observed that:

- In homologous proteins, corresponding residues virtually retain the same rotameric state (Ponder and Richards 1987, Benedetti et al. 1983) 

- Within a range of $\chi$ values, 80% of the identical residues and 75% of the mutated residues have the same conformations (Summers et al. 1987).

- Certain rotamers are almost always associated with certain secondary structure (McGregor et al. 1987).
Refinement of model using Molecular Mechanics

Many structural artifacts can be introduced while the model protein is being built

- Substitution of large side chains for small ones
- Strained peptide bonds between segments taken from difference reference proteins
- Non optimum conformation of loops
Energy Minimisation is used to produce a chemically and conformationally reasonable model of protein structure. Two mainly used optimisation algorithms are Steepest Descent and Conjugate Gradients.
Model Validation

- Every homology model contains errors. Two main reasons
  - % sequence identity between reference and model
  - The number of errors in templates
- Hence it is essential to check the correctness of overall fold/structure, errors of localized regions and stereochemical parameters: bond lengths, angles, geometries
Challenges

- To model proteins with lower similarities (e.g., < 30% sequence identity)

- To increase accuracy of models and to make it fully automated

- Improvements may include simultaneous optimization techniques in side chain modeling and loop modeling

- Developing better optimizers and potential function, which can lead the model structure away from template towards the correct structure

- Although comparative modelling needs significant improvement, it is already a mature technique that can be used to address many practical problems
Fold Recognition Methods

- Profile-based methods
- Threading-based methods
Profile-based methods

- Physico-chemical properties of the amino acids of the target protein must “fit” with the environment in which they are placed in the modeled structure.
Profile-based methods
Threading-based methods

• A library of different protein folds is derived from the database of protein structures.
• Each fold is considered as a chain tracing through space; the original sequence being ignored completely.
• The test sequence is then optimally fitted to each library fold, allowing for relative insertions and deletions in loop regions.
• The ‘energy’ of each possible fit (or threading) is calculated by summing the proposed pairwise interactions and the solvation energy.
• The library of folds is then ranked in ascending order of total energy, with the lowest energy fold being taken as the most probable match.
Ab initio methods
Summary

• Introduction to protein structures
• Protein structure classification
• Protein structure comparison
• Protein structure prediction