Protein Structure IV: Consensus Structure

Paul Chew
CS 426
Fall 2003

Protein Shape

- Protein function is largely based on the protein's shape
- Protein shape can be represented by the positions of the protein's $\alpha$-carbons
- Adjacent $\alpha$-carbons are always the same distance apart (about 4 Å)

Sequence (as a String) vs. Structure

A protein is a sequence: a string over an alphabet of 20 characters (the 20 amino acids)
- View of a protein as a string leads to powerful tools
  - String matching using Dynamic Programming
  - Approximate string matching
  - Suffix trees
  - ...

A protein is a structure: a 3D geometric shape
- Our goal is to build similar tools to manipulate/analyze shapes as easily as we do strings

Outline

- The profile: a string-based tool for distinguishing protein families
- Goal: a shape-based "profile"
- Related research
- The Consensus Structure for a family of proteins
  - Algorithm
  - Properties
  - Examples

Protein Families

- Evolution theory $\Rightarrow$ a protein ancient ancestor evolved into a family of proteins
- Membership in a protein family is expressed by sequence similarity, but is more strongly expressed by structure similarity
  - 25-30% sequence resemblance (almost always) ensures shape resemblance
- There are databases of protein families: SCOP (Structural Classification of Proteins), also CATH, FSSP
- They are mainly classified by their secondary structures (e.g., all-helix, all-strand, helix-strand)

A String-Based Tool: the Profile

- Members of a protein family can be aligned

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.75</td>
<td>0.75</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>B</td>
<td>0.75</td>
<td>0.75</td>
<td>0.50</td>
<td>0.25</td>
</tr>
<tr>
<td>C</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

- This multiple alignment can be used to build a profile

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.75</td>
<td>0.25</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.75</td>
<td>0.75</td>
<td>0.50</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Using the Profile

Given a new protein string, one can determine its protein family by finding the best-matching profile.

The profile summarizes a protein family’s string-information.

Goal: Summarize a protein family’s shape-information.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.75</td>
<td>0.25</td>
<td>0.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0.25</td>
<td>0.25</td>
<td>0.50</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>

Goal: a Shape-Based “Profile”

- Summarize a protein family by using a concise representation of the family’s shape information.
- Determine how to use this representation to automatically classify proteins.
- Use the representation to improve protein fold prediction (predicting structure from sequence).

Our Result: the Consensus Shape

- A structural analog of the profile.
- The Consensus Shape Algorithm produces:
  - a multiple alignment of structures, and
  - a single (core) structure that summarizes the structural information for the family.

Structure-Alignment / Core-Structure

- Gerstein/Altmann ’95 – Multiple alignment based on sequence to derive a structural core.
- Orengo/Taylor ’96 – Multiple structure alignment based on the structural environment of each residue.
- Gerstein/Levitt ’96,98 – Alignment, average core structure, automatic structure alignment against a manual standard.
- Gelfand et al ’98 – Geometric invariant core based on distance matrices.
- Leibowitz et al ’99 – Multiple structural alignment and core detection based on geometric hashing.
- Many papers on pairwise structural alignment.

Sequence of Unit Vectors

- String or shape, a protein is still a sequence.
  - Sequence of characters (amino acids) vs.
  - Sequence of unit vectors.
- The unit here is the 3.8 Å spacing between α-carbons.

Protein Shape and URMS

- Protein function is largely based on the protein’s geometric shape.
- How do we analyze protein shape?
- Our technique: URMS (Unit-vector Root Mean Square distance).
- Advantages:
  - Insensitive to outliers.
  - Efficient to compute.
  - Equal weight for all portions.
Building a Consensus Shape

- Suppose we have a multiple alignment of proteins in a protein family.
- Idea: Can build a consensus shape by averaging the corresponding unit vectors.
- How do we get the initial multiple alignment?
- How do we make the proteins have consistent orientations?
- What about gaps?
- A profile can be iteratively improved; can we iteratively improve the consensus shape?

Consensus Shape Algorithm

- Input: a set of proteins belonging to a single protein family.
- Output: the consensus shape, a pseudo-protein that summarizes shape information for the family.
- Initialization:
  - Arbitrarily choose one member of the family to be C, the initial consensus shape.
  - Use Dynamic Programming to align the other members of the family to C.
- Optimization: Loop until C quits changing.
  - For each protein P in the family, use Dynamic Programming to align P with C.
  - Improve C.

Alignment via Dynamic Programming

- Alignment RHYPGDFSFA
- ARFPADFTAE
  - Corresponds to
- Alignment RHYPGDFSFA
  - ARFPADFTAE
  - Corresponds to

Initialization: Orientation via DP

- Want pairwise alignment using Dynamic Programming.
- But...
  - We don't know the proper orientation.
  - If the orientation is arbitrary then all unit-vectors look alike.
- Idea: Use a block of, say, 5 unit-vectors as a single "character".
- Instead of a 20-character alphabet of amino acids, we have an infinite alphabet of 5-unit-vector characters.
- A match (using URMS) between 2 such characters gives us an orientation as well as a distance.

Initialization (expanded)

- Initialization:
  - Arbitrarily choose one member of the family to be C, the initial consensus shape.
  - For each protein P in the family:
    - Use Dynamic Programming on substructures of size 5 to match P to C.
    - Use the close matches to "vote" on the proper orientation.
    - Use clustering to do the voting.
    - Use the Frobenius norm for distance between orientations (recall that an orientation is a 3x3 matrix).
    - Use the winning orientation to orient P with respect to C.
Optimization (expanded)

- Optimization: Loop until C quits changing
  - For each protein P in the family:
    - Use Dynamic Programming on vectors to align P with C
    - Cost for two vectors = squared distance between endpoints
    - Gap cost = (see next slide)
    - Result: one DP table (with optimal path) for each protein P
  - Improve the consensus shape, C, by doing one or more of the following (order is unimportant)
    - Recompute C by averaging corresponding unit-vectors
    - For each protein P in the family, orient P to minimize its distance from C
    - Introduce an additional consensus vector
    - Delete an existing (useless) consensus vector

Optimization: DP Cost Rules for Vectors

- Cost for comparing two vectors = squared distance between the vector endpoints
  - Cost(u,v) = |u-v|^2
- Why use the squared distance?
  - Because the sum of squared distances corresponds to the URMS metric
  - We know matches using this metric correspond well to our intuition about matches
- What should we use for a gap cost?
  - A constant indicative of a bad vector match?
  - A zero vector?
  - Idea: Ideally, the gap cost should correspond to some kind of gap vector
  - Use a unit vector in perpendicular direction
  - The gap vector is (0,0,0,1) (i.e., we use the 4th dimension)
  - Cost(g,v) = |g-v|^2

Optimization: Improving Consensus Shape

- Each of these 4 improvement methods can be shown to decrease the total cost (total cost is the sum of the costs of the DP paths, one path for each protein P of the family)
  - Recompute C by averaging corresponding unit-vectors
    - Correspondence is via DP paths
  - For each protein P in the family, orient P to minimize its distance from C
    - Use URMS on corresponding vectors
    - Correspondence is via DP paths
  - Introduce an additional consensus vector
    - Used to replace vertical path segments
  - Delete an existing (useless) consensus vector
    - Used when the consensus vector is a gap vector

Preserved Structure Property

- Theorem:
  If all member of a protein family exhibit a geometric relationship between corresponding α-carbons then that relationship is preserved in the consensus shape
  - In particular, distances and angles are preserved
  - This holds even though vectors between adjacent α-carbons are short in regions where the proteins disagree
  - Requires the use a special gap-direction (a 4th dimension) to distinguish between short vectors due to disagreement vs. short vectors due to gaps

Useful Maps: Consensus vs. Proteins

- For each protein unit-vector, there is a single corresponding consensus vector
- For each consensus residue, each protein of the family has a single corresponding residue

An Alpha Protein Family (Globins)
A Beta Protein Family

Unrelated Proteins

String vs. Structure (4 alpha proteins)

String vs. Structure (10 globins)

Continuing Research

- Exploring use of multiple shape alignment as a way to automatically divide proteins into families
- Exploring use of consensus shape as an aid to threading

Consensus Structure for Threading

- **Pro**
  - Geometric properties that are consistent among family members are preserved in the consensus structure
  - Consensus structure highlights the important features of a family allowing weighted threading

- **Con**
  - Distance between adjacent \(\alpha\)-carbons can be small
  - There are “extra” residues
  - For contact-based threading, not clear what to count as a contact
Discrete/Continuous Optimization

- To find an optimal structure-match between 2 proteins
  - Need to match residues
  - Need to determine orientation
  - If either is known, the other is easy
- Goal is to do both at once

- For matching of 2D pseudo-proteins
  - We have a $2^{\sqrt{\log(n)}}$ algorithm
  - Practical for typical protein lengths??
  - Optimistically: It's not exponential time, so a practical algorithm may exist

- Is there a 3D version?
- Is it practical?
- There are other applications