Bioinformatics III
Structural Bioinformatics and Genome Analysis

Chapter 3 Structural Comparison and Alignment

3.1 Introduction

3.2 Main Methods

1. **Basic algorithms review**
   - Dynamic programming
   - Distance matrix
2. **SARF2, CE, DALI, SSAP**

3.3 Recent Methods

MAMMOTH, RAPIDO, SABERTOOT, TOPOFIT

3. Structural Comparison and Alignment
Introduction

Goal:

Determination of equivalences between amino acid residues by taking into account 3D structures

Relationships between primary protein sequence, 3D structure and biological function

Four steps

- **Structure alignment**: find equivalences of amino acid residues based on known 3D structures of protein folds
- **Structure comparison**: once shared similarities are known the structures are compared
- **Structure superposition**: find the optimal overlap of both proteins (matches information about residues for protein A and B is known)
- **Structure classification**: assign the protein to a certain class

Structural alignments provide information that is unavailable through current sequence alignment methods: distant sequence relationship

3. Structural Comparison and Alignment
Introduction

Motivation:

From Genome sequencing to amino acids/nucleotides primary structure
From amino acids/nucleotides primary structure to 3D Structure Prediction

2008 In PDB data base 49 192 Structures structures
2009 56 066 Structures
2010 65 075 Structures

2008 SWISS PROT 356 194 entries sequence
2010 516 603 entries

Ratio of 1 structure to 7 sequences

3. Structural Comparison and Alignment
Introduction

Protein structures are more highly conserved than sequences:

- Evolutionary changes like insertions and deletions take place mainly in loop regions
- No alterations in the final fold and limiting the number of possible folds

Complexity is reduced through evolution maintaining diversity and adaptability

- All proteins all species 1000-5000 protein folds (Chotia, 1992)

Similar structures may be formed by alternative folding of the amino acids' Cα backbone:

- Matched regions separated by unmatched segments

Partial local similarities do not automatically transfer to similarities in structure

Same nucleus BUT different end

- 30% Sequence identity adopt the same folds: homologous folds
- 5% Similarity can result in the same fold: analogous folds

1 Basic algorithms review

Structures can be compared, assuming they adopt the same fold

Structural comparison and alignment as NP-hard problems:

Non-deterministic polynomial time problems solved by heuristic approaches
Possible solution as the best analytical answer but NO biological mean

Find the most suitable method to solve the optimization of the alignment and to reduce the computing time consuming problem: 5 of 10 structures inferred without special algorithms

New proteins weekly released in PDB previous all-against-all comparison
Known sequence-structure relationships are used
PDB structures are grouped (only a subset is compared)

1 Basic algorithms review

Steps for algorithm optimization

A. **Structure comparison and alignment**

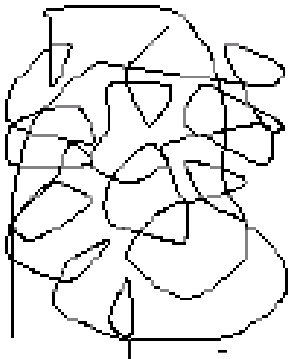
i. Representation of the pair of proteins 1 and 2, domains or fragments to be compared and aligned
ii. Compare 1 and 2
iii. Optimize the alignment between 1 and 2
iv. Statistically significant measurement of the alignment against a random set of structures

B. **Multiple structure alignment**

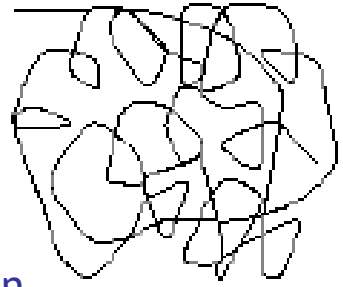
i. Starting from the initial alignment found in Aiii., the next step is running a search within a constraining sequence window to find the optimal alignment against all structures using profiles; HMMs or Monte Carlo approaches.

How to compare structures

Structure 1



Structure 2



Feature extraction

1. Structure description 1

Structure description 2

2. Comparison algorithm

3. Scores

Statistical Significance (PDB)

Similarity-Classification

Dynamic programming in Structural Bioinformatics

Aligning Sequences:

A row of amino acids in one sequence matches a row of identical or substituted positions in the second sequence; insertions or deletions as gaps

Aligning Structures:

A scoring matrix is built to compare the positions of the atoms in both 3D structures

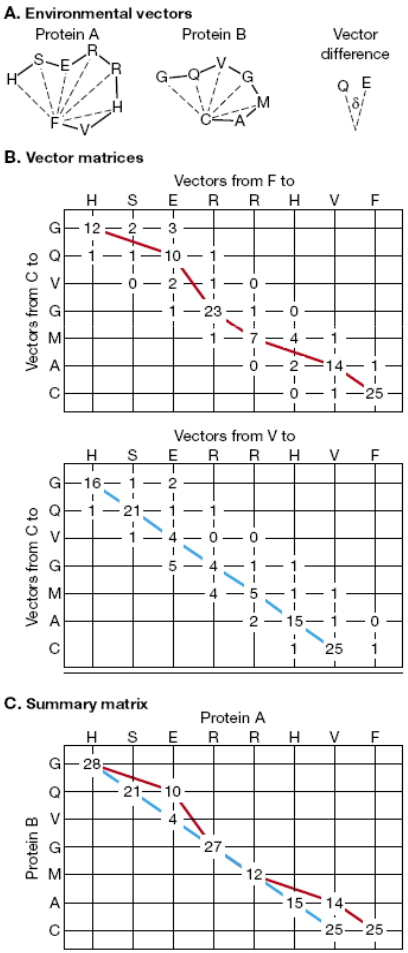
i. Scoring matrix gives scores of how well any of the 20 amino acids fits to a single position in the structure. Calculation of an optimal alignment

ii. Positions of SSEs within a domain: similar types, positions and numbers

iii. Distances between the Cα (NH- Cα- Cβ) and Cβ (Cα-CβO=NH) atoms within these domains, and later within the whole structure

iv. Determination of the degree of superimposition

Dynamic programming in Structural Bioinformatics



Two steps

1. Atoms or molecules as vectors:
A coded value is given describing the local environment of each amino acid

- Interatomic distances
- Bond angles
- R groups

Cartesian coordinates are assigned to each (X, Y, Z)
Corresponding to the beginning and end positions of the SSEs
Direction of the bond angles is included

2. The alignment of 2D structures:
Determine of the interatomic distances between each amino acid in the polypeptide chain

"The better the arrangement, joining and 2D alignments are, the more significant and convincing is the result"

Distance matrix

No alignments help is needed

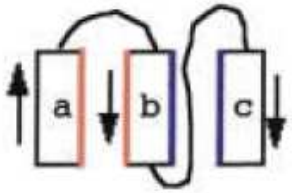
Each position in the 2D matrix represents the distance between corresponding Cα atoms in the 3D structure

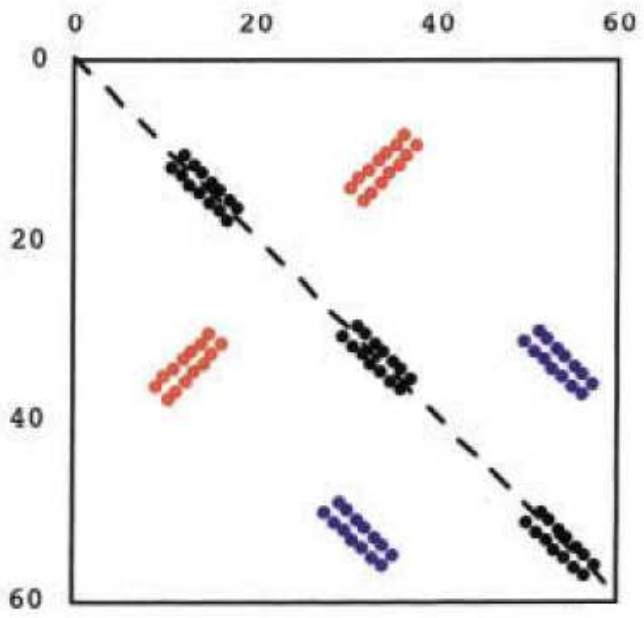
i. Distances between Cα atoms along the polypeptide chain and between Cα atoms within the protein structure are compared

ii. Similar groups of 2D structural elements are superimposed (sum distance minimization in the aligned Cα atoms resulting in a common core)

"The smallest the distance, the most closely packed atoms within SSEs and regions of the 3D structures "

Distance matrix





Bases of Distance method

Degree to which all of the matched elements can be superimposed

Protein A ------- helices a and b interacting
Protein B ------- helices a' and b'

Helices superimposition-set of Cα

i^A and i^B in helix a and a'
j^A and j^B in helix b and b'

For matching pairs

dijA = distance between i^A and j^A

dijB = distance between i^B and j^B

SS = |dijA – dijB| / dij*

dij* = average of dijA and dijB

0.2< SS =0.2 SS 1Å β strands SS 2-3Å α and β