CMPS 6630: Introduction to Computational Biology and Bioinformatics

Structure Comparison

Protein Structure Comparison

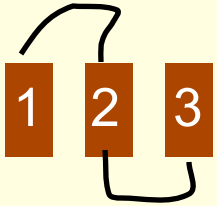
Motivation

- Understand sequence and structure variability
- Understand Domain architecture of proteins
- Understand evolution of protein function
- Infer structural relationships
- Infer evolutionary relationships
- Determine coverage of fold space
- Use in predictive modeling

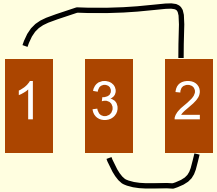
Structure Comparison

Points to Consider

- Feature Extraction
  What features are to be extracted and compared?
- **Fine level** (residue or atom) vs. **Coarse Level** (SSE)
  Fine level can be used to make functional hypotheses
  Coarse level used for global fold comparison / classification



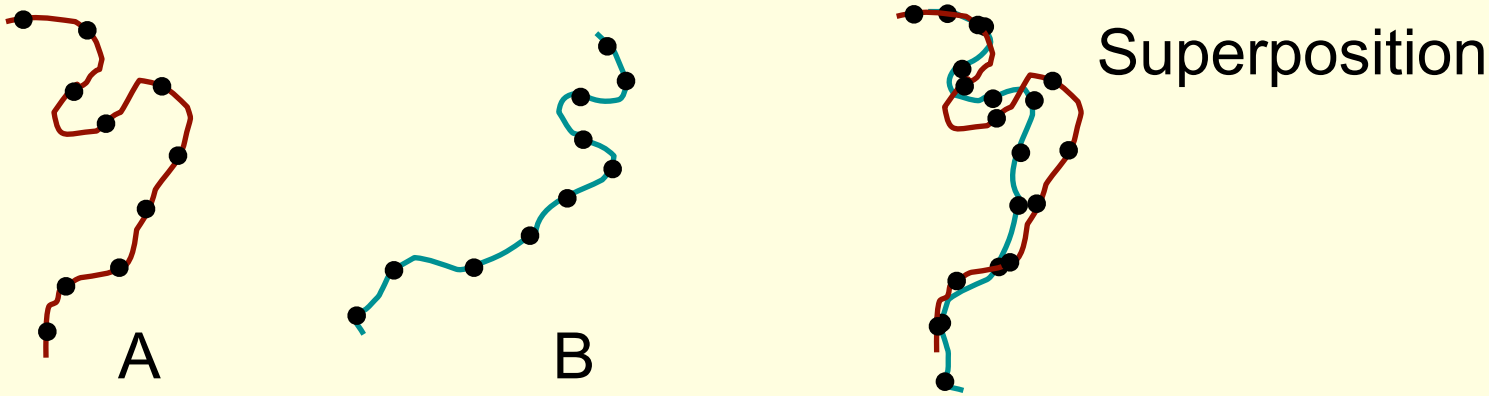
- Maintenance of **Topology**?
  Does pattern need to have similar sequential ordering?
- Method of **Comparison**?
  How similar do structural elements need to be to match?
  Should be:
  - Invariant to trivial changes (ie. rotation / translation)
  - Robust, description should not change drastically due to minor changes in structure



Structure Comparison

Protein Similarity

Given a ***correspondence*** and an ***optimal*** positioning of two structures, how ***close*** are corresponding residues/ elements?



$$S^*(A,B) = \min_{T,C} \boxed{D}(A, \boxed{T}(B), \boxed{C})$$

T: Transformation C: Correspondence

RMSD Root Mean Squared Distance

- Most common method to score similarity of two structures
- Most useful to compare relatively similar structures
- Often computed from C_α only
- Requires residue correspondence between two proteins
- Distance measured in Angstroms

  Smaller RMSD implies more similar structures

Coordinate RMSD

$$\mathrm{RMSD}_C(E) = \min_T \sqrt{\frac{1}{\sum_{i=1}^{r} w_i} \sum_{i=1}^{r} w_i (T\alpha_i - C(\alpha_i))^2}$$

T : Transformation

w_i : weights (often 1)

α_i : set of equivalenced atoms

RMSD Root Mean Squared Distance

Distance RMSD

Rotation and Translation Invariant

$$\mathrm{RMSD}(E) = \frac{1}{r}\sqrt{\sum_{1 \le i,j \le n} (\delta_{ij}^{A} - \delta_{ij}^{B})^2}$$

Note that distance-RMSD increases with the size of the point sets being compared. We can normalize by the square root of the length:

$$\mathrm{RMSD}(E) = \frac{1}{r}\sqrt{\sum_{1 \le i,j \le n} \frac{(\delta_{ij}^{A} - \delta_{ij}^{B})^2}{n}}$$

RMSD

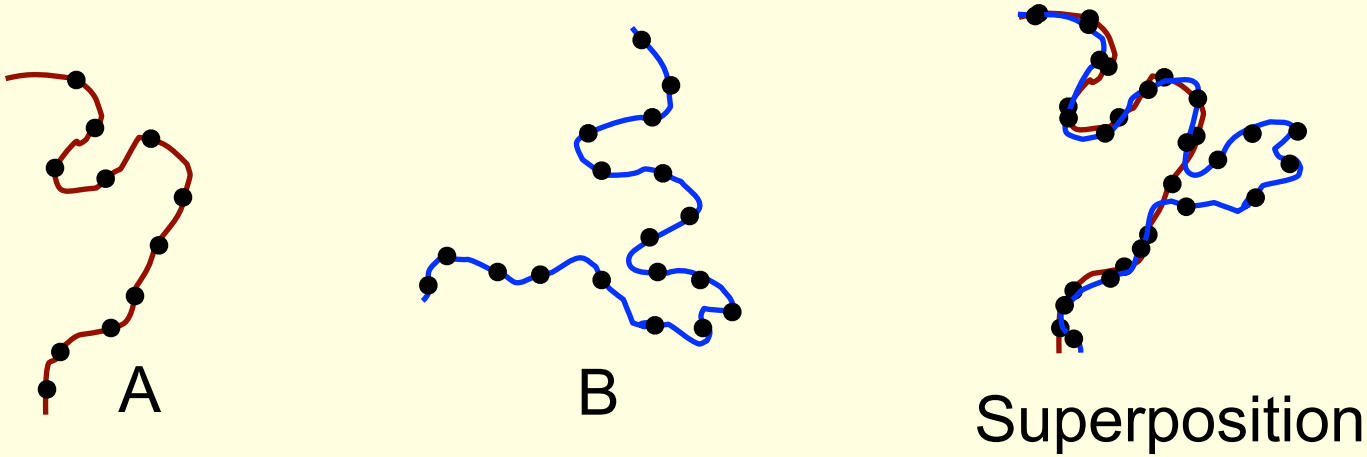
$$S^*(A, B) = \min_{T,C} D(A, T(B), C)$$

If distance measure (D) is RMSD and correspondence (C) is given, then T can be computed easily using SVD.

A **B** **Aligned Centroids** **Optimal Superposition**

RMSD Limitations

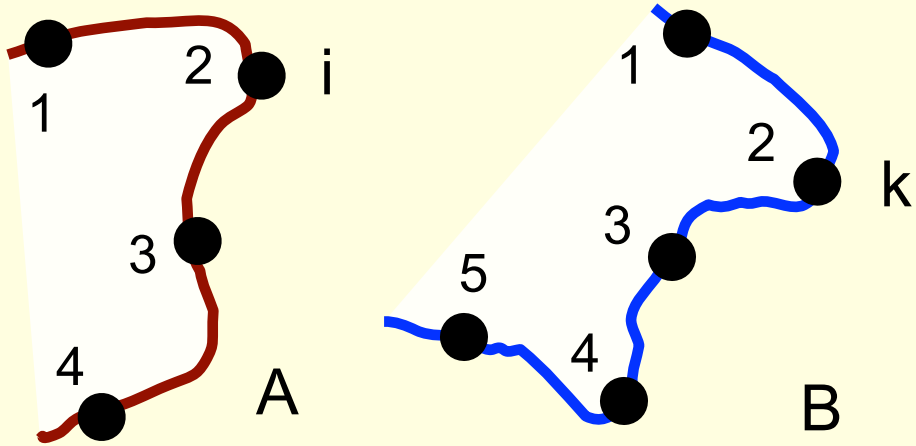
- Equivalence of positions (correspondence) must be known
- Relative displacement of one subdomain within one structure can result in poor overall fit
- Insertions and Deletions? Gaps?



What can we learn from sequence alignment?

Structural Alignment

Lets try the same thing to compute alignment and similarity.



	1	2	3	4
1				
2				
3				
4				
5				

Similarity Matrix

D

How to compute the similarity $D_{i,k}$?

Structural Alignment

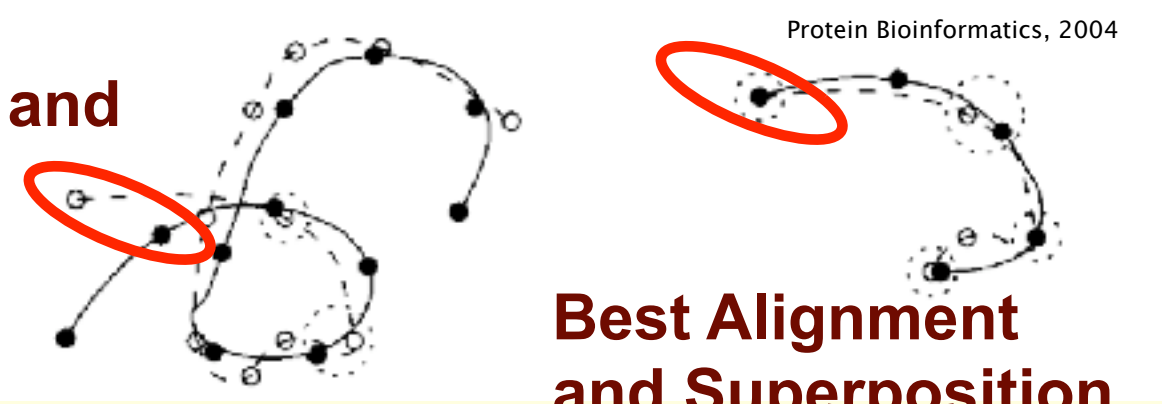
Will Dynamic Programming Work for Structural Alignment?

?

Optimal Substructure: an optimal solution to the problem contains within it optimal solutions to subproblems

Overlapping Subproblems: the space of subproblems must be 'small', solving the same subproblems over and over

Best Alignment and Superposition of Entire Chain



Best Alignment and Superposition of first 5 Residues

Structural Alignment

Will Dynamic Programming Work for Structural Alignment?

?

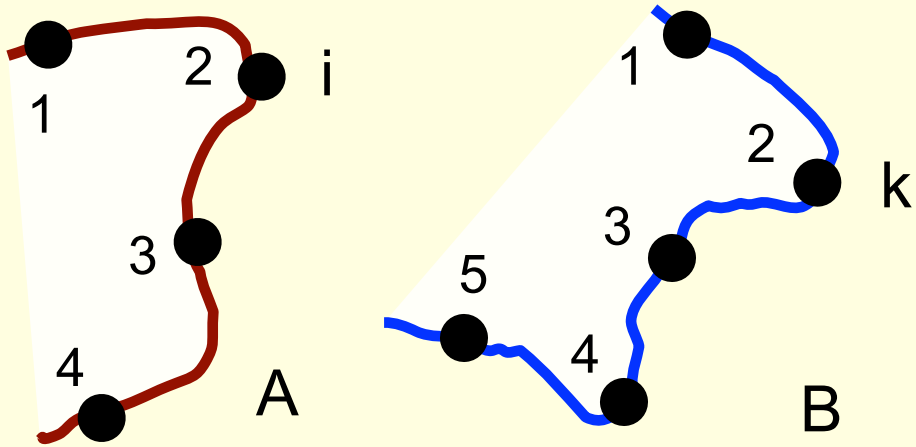
~~**Optimal Substructure:** an optimal solution to the problem contains within it optimal solutions to subproblems~~

~~**Overlapping Subproblems:** the space of subproblems must be 'small', solving the same subproblems over and over~~

Any choice to align two substructures (local alignment) will affect the scoring of the global alignment between the complete structures.

The independence requirement is violated and DP can no longer guarantee an optimal solution.

Structural Alignment



Similarity Matrix

	1	2	3	4
1				
2				
3				
4				
5				

D

Each entry should indicate likelihood of match by incorporating global information

Incorporate some global information locally.

SSAP

Structure and Sequence Alignment Program

Double Dynamic Programming - flashy name for utilizing two levels of dynamic programming (two levels of scoring matrices)

	1	2	3	4
1				
2				
3				
4				
5				

High-Level Scoring Matrix
***D_H*:** locally incorporates some global information

Position *i*,*j* score is likelihood that *i*,*j* will appear in the final alignment*. This is determined by computing the best alignment forced to contain each *i*,*j* and using the transformation that best superimposes *i* and *j*.

This is done with the low-level scoring matrix.

** this is somewhat different than before*

SSAP

	1	2	3	4
1				
2				
3				
4				
5				

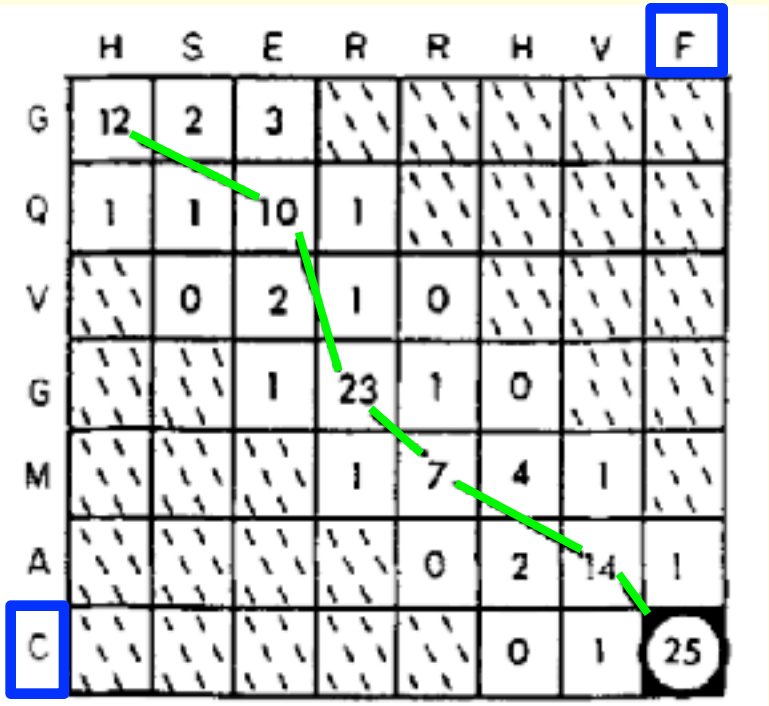
For each element $D_H(i, j)$, define a separate low-level scoring matrix, ${}^{ij}D_L$. Element ${}^{ij}D_L(k,l)$ gets a score specifying how well a_k fits to b_l given that a_i is 'perfectly' aligned with b_j.

High-Level Scoring Matrix

D_H: locally incorporates some global information

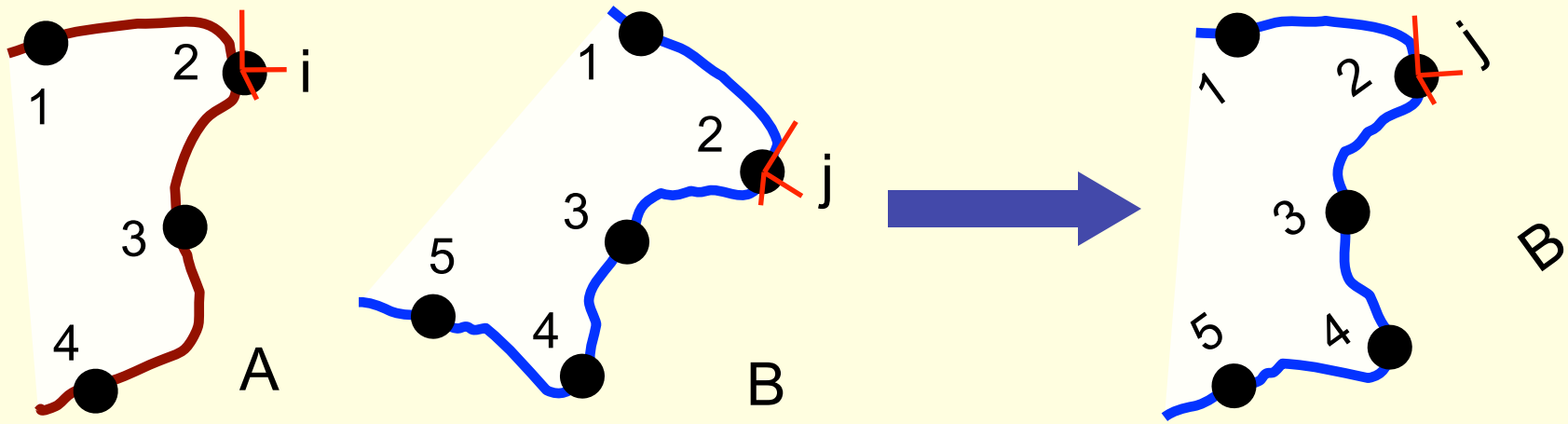
Results from DP on low-level matrix are added to the high-level scoring matrix.

~Voting



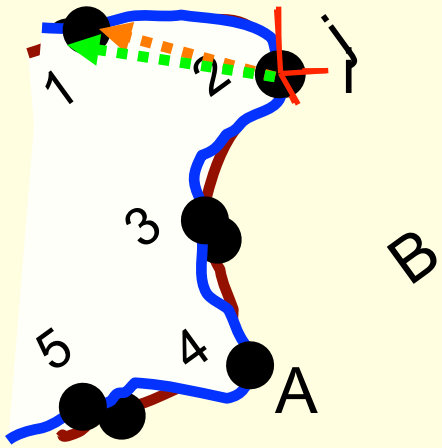
${}^{ij}D_L$ Anchor F and C

SSAP



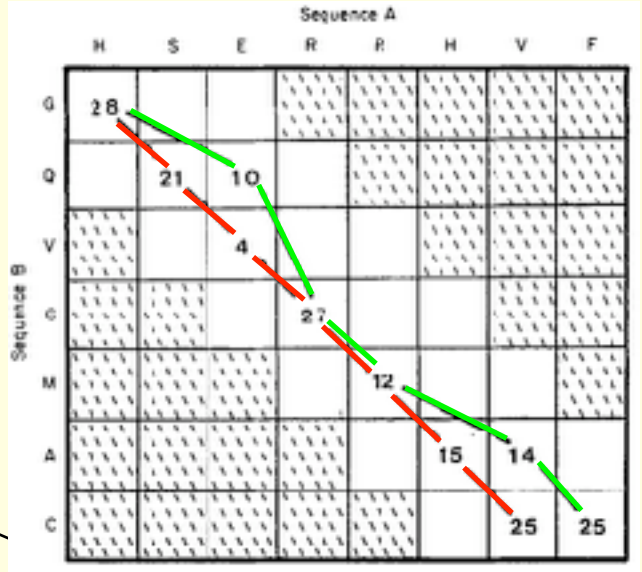
Rotated to align local coordinate frame of i and j

For all pairs of points, compute similarity score based on vector differences.

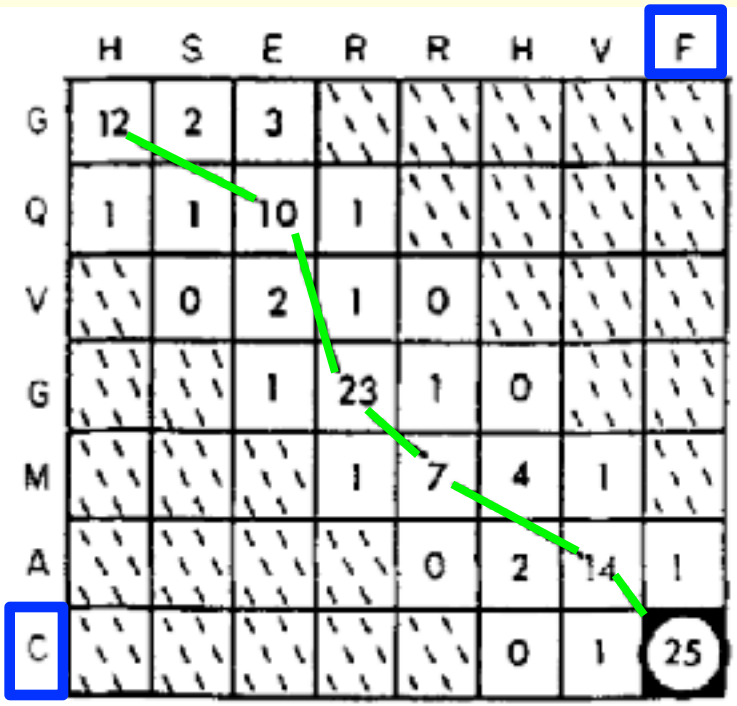


SSAP

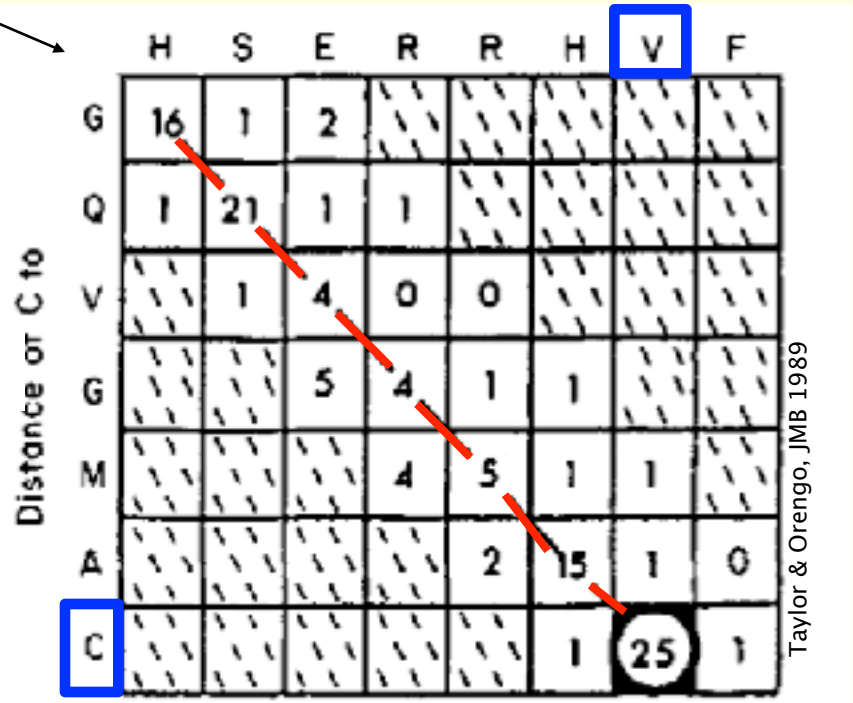
Low-level Scoring Matrices D_L with overlaid best alignment from DP



High-Level Scoring Matrix



Anchor F and C



Taylor & Orengo, JMB 1989

Anchor V and C

SSAP

For each potential correspondence of residues i, j:

1. Pin down the correspondence i, j
2. Use local coordinate frames to orient the two proteins
3. Compute a low-level scoring matrix where the score between residues x and y is based on similarity of their positions relative to i and j.
4. Use this low-level scoring matrix to find best alignment given correspondence between i and j.
5. Use the result of this best alignment to 'vote' for correspondences in the high-level scoring matrix

The correct transformation should bring multiple consistent pairs of residues into proximity and so it should get voted for many times.

Time Complexity: $O(n^4)$

SSAP

- Returns a correspondence between two structures and a similarity score
- Can be used to compare structures for classification
- No guarantee on optimality - what cases can't we handle?
- Works well but is slow $O(n^4)$
- As database size grows this becomes a problem
- Used by the **CATH** protein structure classification database

Many SSAP Extensions:

Incorporate sequence information

Multiple ways of computing low-level scoring matrix

Iterative version - keep track of a set of candidate anchor points, the best alignment is computed and the anchor point list is updated until convergence.

GRATH

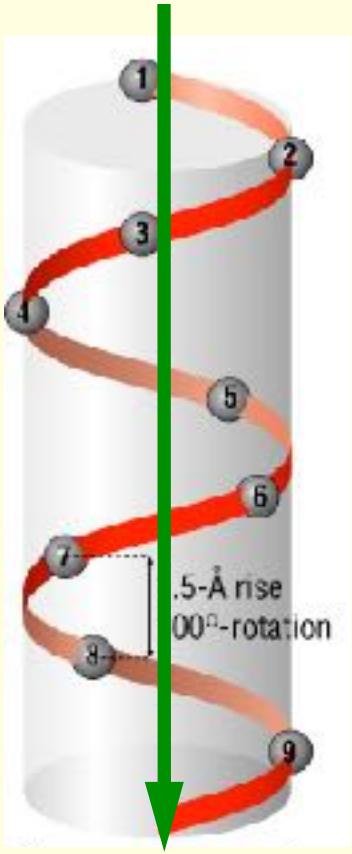
Need for faster structure comparison to replace SSAP

Goal: Produce a front-end filter for the more reliable SSAP - only those structures are are reasonably close need to be compared with SSAP

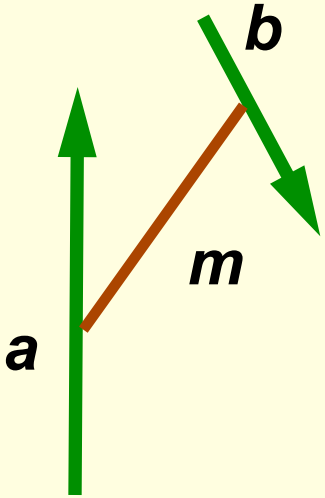
Graph based method
Secondary Structure Matching

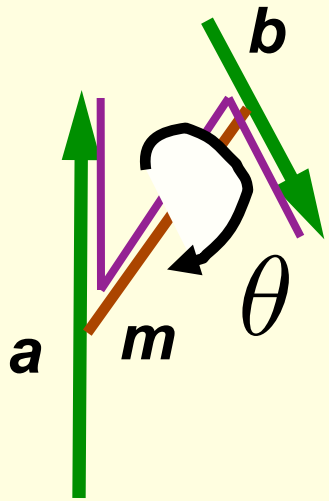
Nodes are secondary structures
Edges contain relationship information

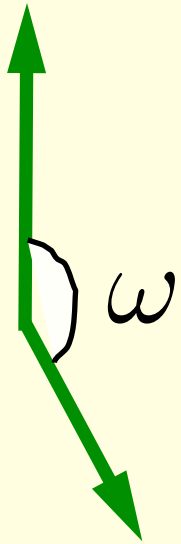
Axial vectors computed for each SS element via least squares fitting of C_α

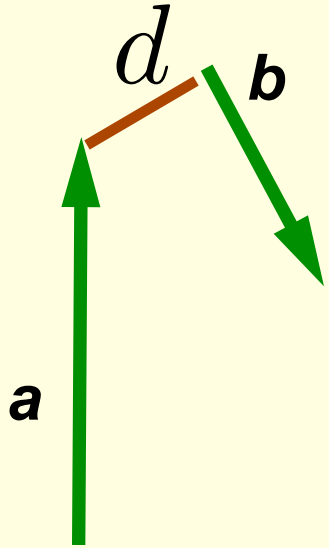


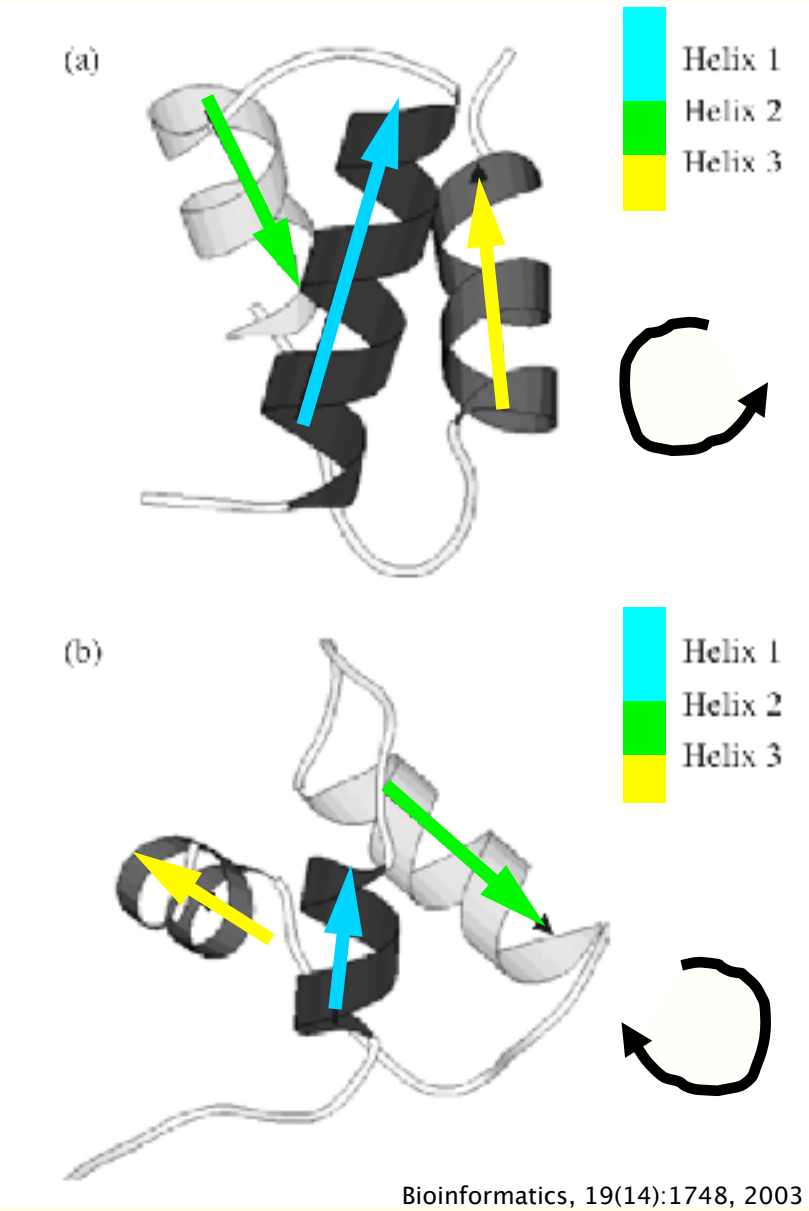
GRATH









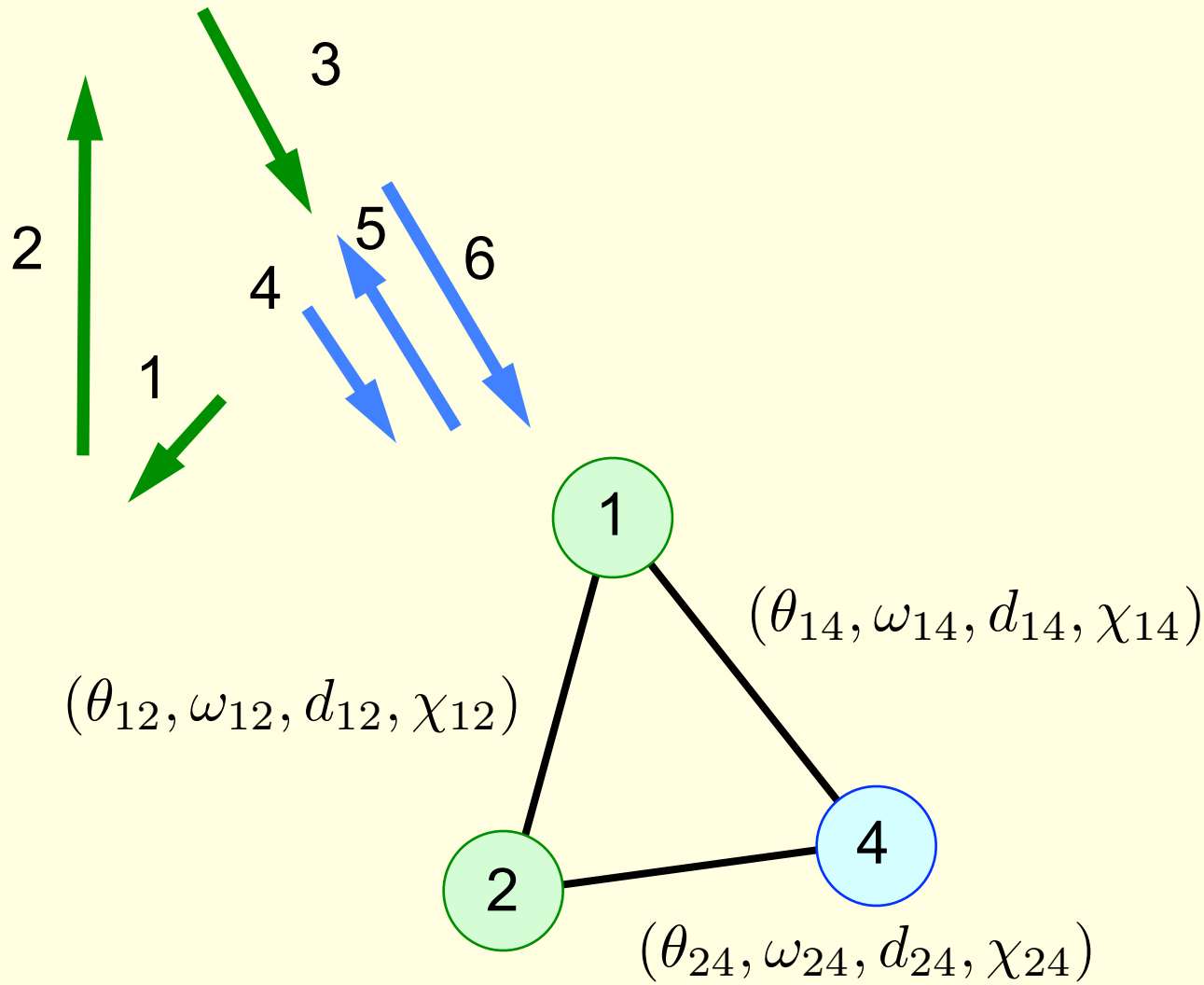


Bioinformatics, 19(14):1748, 2003

χ : left, right, other

GRATH


Tertiary


3
2
5
6
4
1
1
$(\theta_{14}, \omega_{14}, d_{14}, \chi_{14})$
$(\theta_{12}, \omega_{12}, d_{12}, \chi_{12})$
4
2
$(\theta_{24}, \omega_{24}, d_{24}, \chi_{24})$

GRATH

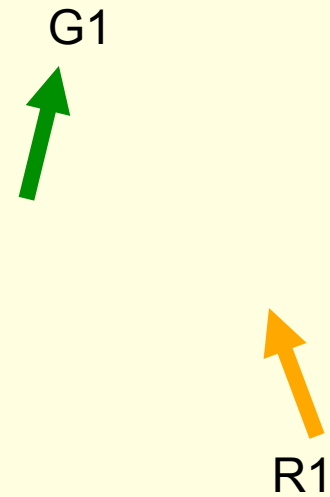
- Given a graph for each protein, compute similarities
- Use of two matrices
  - **Secondary Structure Similarity Matrix**
    Identical secondary structural elements marked
  - **Correspondence Matrix**
    Consistent pairs of secondary structure marked

		R1 α	R2 β	R3 α	R4 β	R5 β
G1	α	1		2		
G2	α	3		4		
G3	β		5		6	7
G4	β		8		9	10

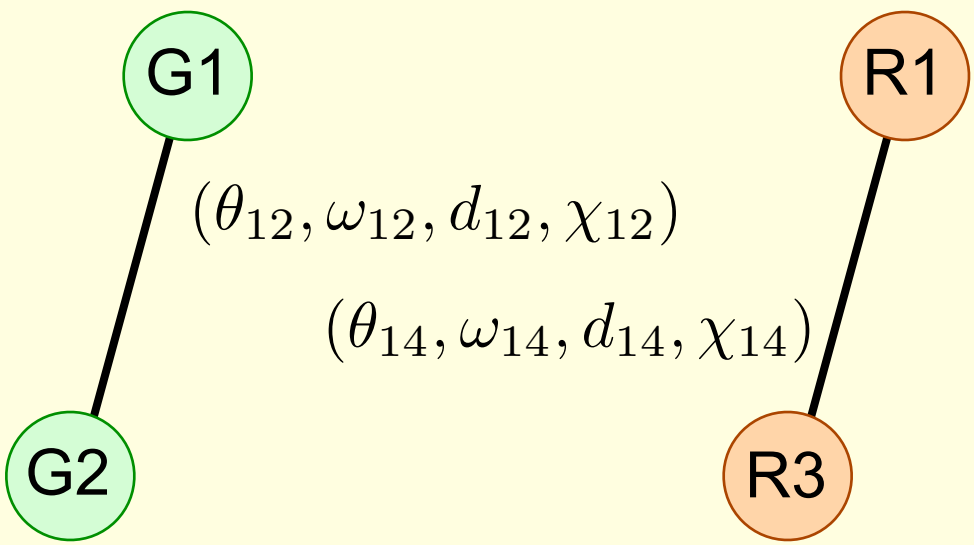
Bioinformatics, 19(14):1748, 2003

		R1	R2	R3	R4	R5
		α	β	α	β	β
G1	α	1		2		
G2	α	3		4		
G3	β		5		6	7
G4	β		8		9	10

k matches in SS Similarity Matrix



Consider all k^2 pairs of matches.
The consistency of each pair of matches is recorded in the *k* x *k* **Correspondence Matrix**.



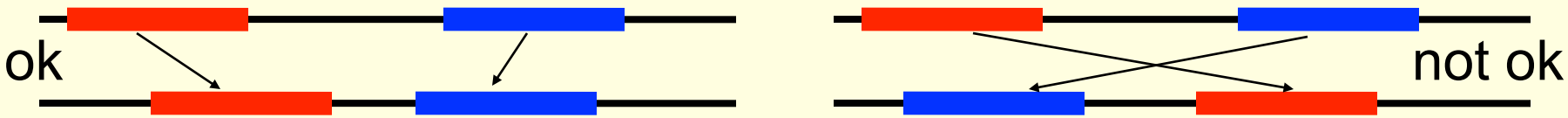
Matches are consistent if:

distance, angle, torsion, and chirality are within error tolerance

The *k* x *k* **Correspondence Matrix** indicates consistent pairs of secondary structure matches

Enforce topology

- Maintain ordering
- Maintain self-consistency (a SS can not match itself, ie. the pair 1 and 3 is not allowed)



	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	0	1	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	1
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Bioinformatics, 19(14):1748, 2003

Can be considered a graph with *k* nodes and edges between nodes *i* and *j* if $M_{i,j} = 1$

scoring.....

GRATH Scoring

$$S = \frac{W_1 \frac{\mathrm{CS}}{\mathrm{Max(SS1,SS2)}} + W_2 \frac{\mathrm{Min(R1,R2)}}{\mathrm{Max(R1,R2)}} + W_3 \frac{(\mathrm{CR1/R1} + \mathrm{CR2/R2})}{2}}{W_4} \quad (1)$$

SS1, *SS2*: number of secondary structure
R1, *R2*: number of amino acids
CS: clique size
CR1, *CR2*: residues in secondary structures of clique
W_1, W_2, W_3, W_4: weights ($W_4 = W_1+W_2+W_3$)

GRATH Scoring

Atomic Coordinates

Extract Secondary Structure

Compute Axial Vectors

Compute pairwise vector similarity measures

Generate Secondary Structure Similarity Matrix

Examine all pairs of matches for consistency
generate Consistency Matrix

Clique Detection

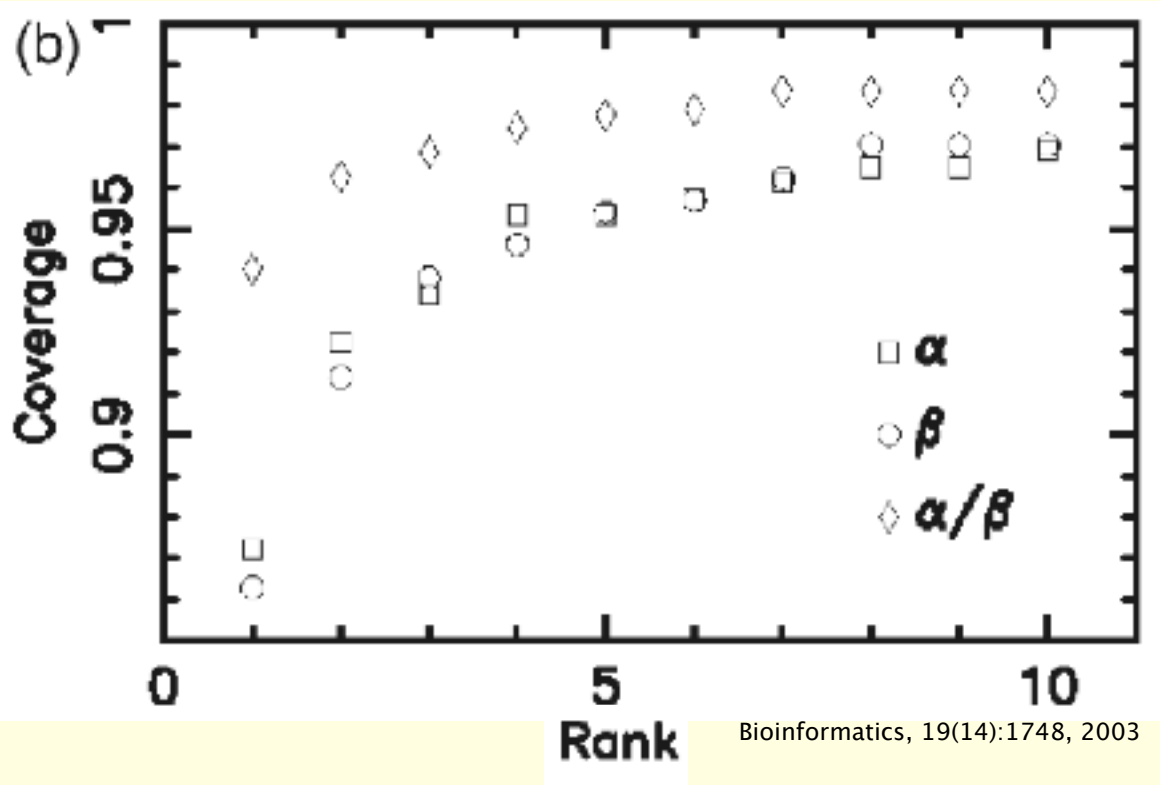
Scoring

GRATH Results

- GRATH almost always find the correct fold at the top or close to the top of the ranked list

Correct fold is within top-10 results 98% of time

ENRICHMENT

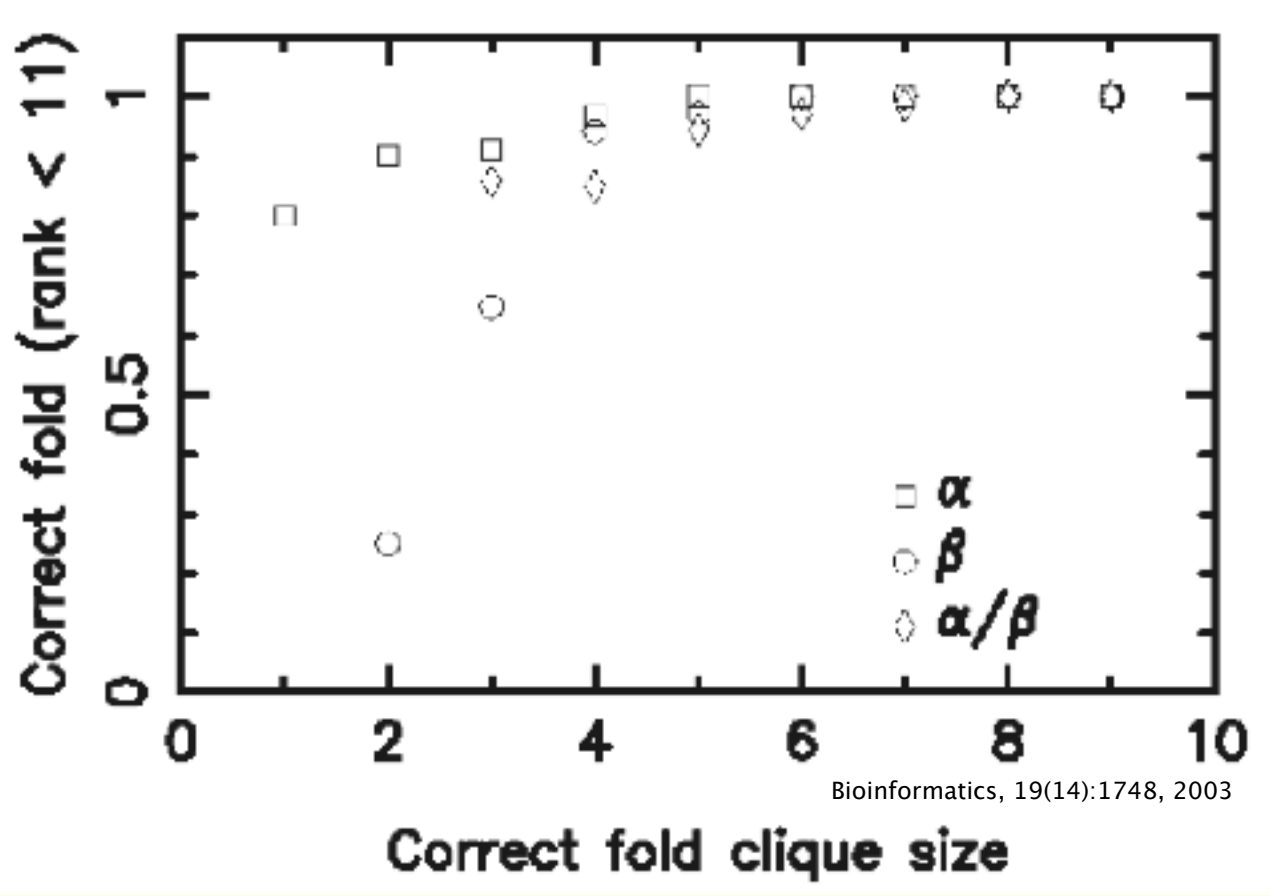


Bioinformatics, 19(14):1748, 2003

Empirically 3-4 Orders of Magnitude faster than SSAP

GRATH Results

- The larger the found clique size, the better GRATHs ability to find the correct fold



Bioinformatics, 19(14):1748, 2003

Enrichment

The increase in frequency of true positives in a dataset

Enrichment Factor - the ratio of the frequency of positive samples in the filtered dataset to the frequency of positive samples in the original dataset.

- **Lossy Enrichment** - some results meeting specified criteria are lost. Some false negatives.
- **Lossless Enrichment** - no results meeting specified criteria are lost. No false negatives.

Common Theme

A fast enriching algorithm is often followed by a slower, but more precise, verification algorithm for identifying true positives. (ie. GRATH and SSAP)

SCOP: Structural Classification of Proteins

http://scop.berkeley.edu/index.html

- Manual classification based on
  Sequence, Structure, and Function
- Unit of classification is the ***domain***
- Hierarchical Classification of Structures (7 levels)
  Class - Fold - Superfamily - Family -
  Protein Domain - Species - PDB Entry

SCOP: Structural Classification of Proteins

http://scop.berkeley.edu/index.html

Class: Based on secondary structure content
all alpha, all beta, alpha/beta, alpha+beta

Fold: Based on number, type, and arrangement of secondary structural elements. ***Same core structure and topology***.

Superfamily: Based on hypothesized evolutionary relationship. Proteins have ***similar structure and function*** (but not necessarily sequence)

Family: Proteins with clear evolutionary relationship. ***Similar structure, function, and >30% sequence identity***

SCOP: Structural Classification of Proteins. 1.71 release

27599 PDB Entries (October 2006). 75930 Domains.

Class	Num folds	Num Superfamilies	Num Families
All alpha proteins	226	392	645
All beta proteins	149	300	594
Alpha and beta proteins (a/b)	134	221	661
Alpha and beta proteins (a+b)	286	424	753
Multi-domain proteins	48	48	64
Membrane and cell surface prot.	49	90	101
Small proteins	79	114	186
Total	971	1589	3004

SCOP has been very useful
Over 700 direct citations!

- Study evolution of enzymatic function
- Study of distantly related proteins with the same fold
- Study sequence and structure variability
- Derive AA similarity matrices
- Composition of multi-domain proteins
- Identification of new targets for structural genomics initiatives

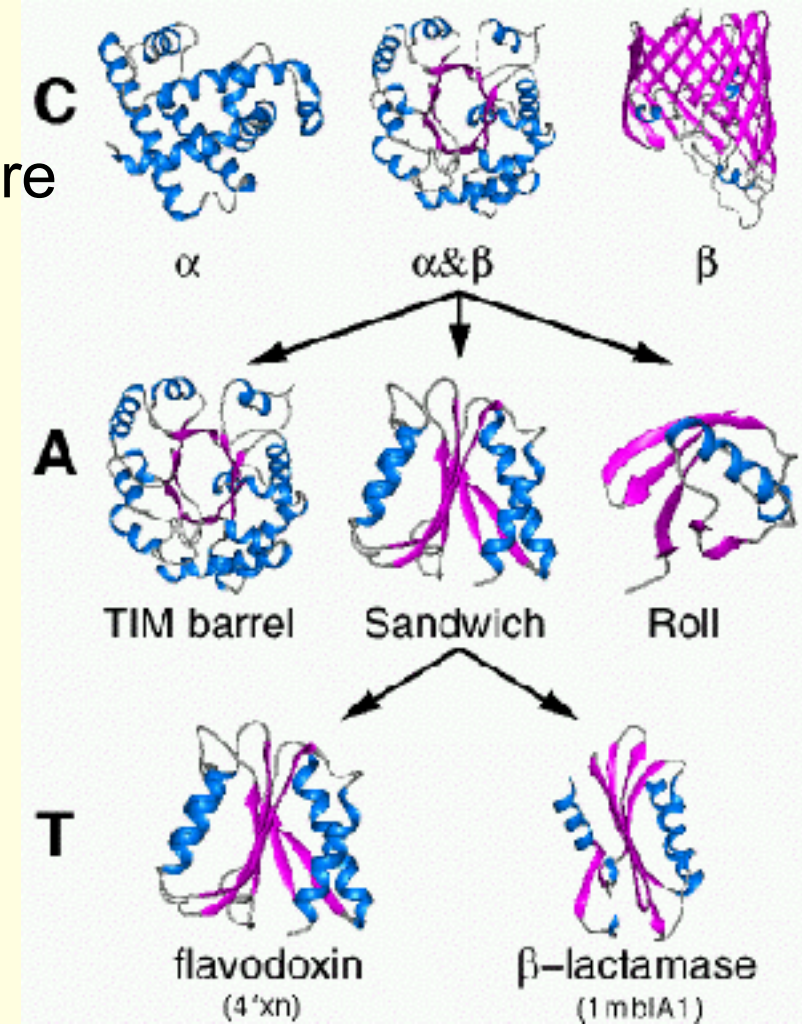
CATH

http://www.cathdb.info/latest/index.html

Class, Architecture, Topology, Homologous Superfamily

+ Sequence Family and PDB Entry

- Hierarchical Grouping of Structure
- Significantly Automated
  (but not completely)
- Initiated in 1993



CATH

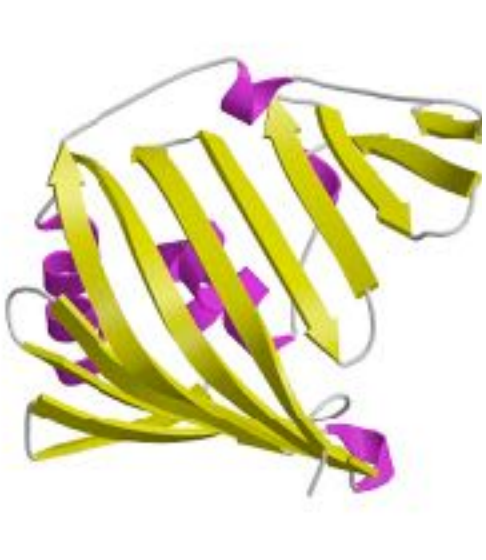
http://www.cathdb.info/latest/index.html

Class, Architecture, Topology, Homologous Superfamily

+ Sequence Family and PDB Entry

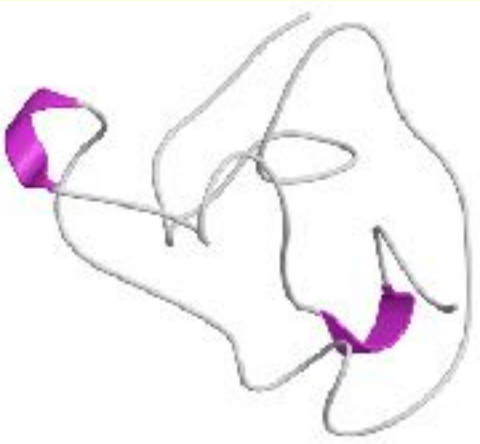
Class

Mainly Alpha



Mainly Beta

Mixed Alpha-Beta



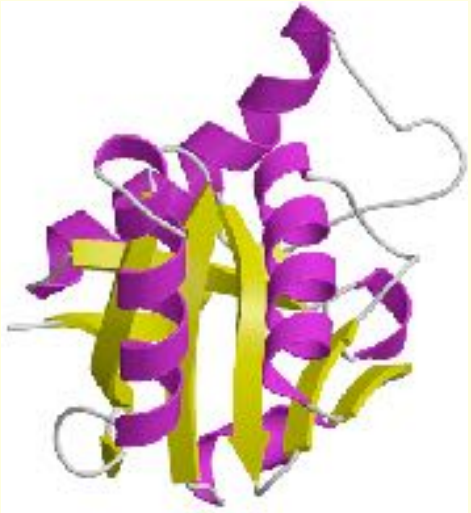
Few Secondary Structures

CATH

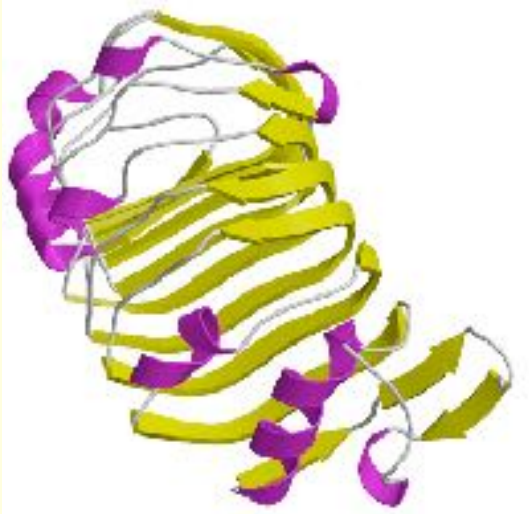
Architecture

Arrangement of secondary structures
Ignores order of secondary structures
    Ignores topology
Manually assigned into ~40 different architectures

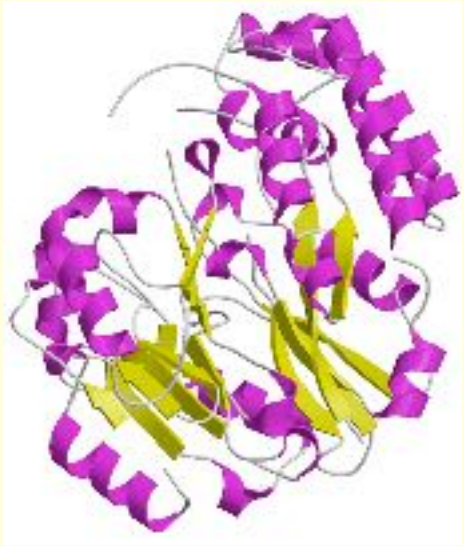
Roll



Alpha-Beta Barrel



Alpha-Beta Horseshoe



5-Stranded Propeller

CATH

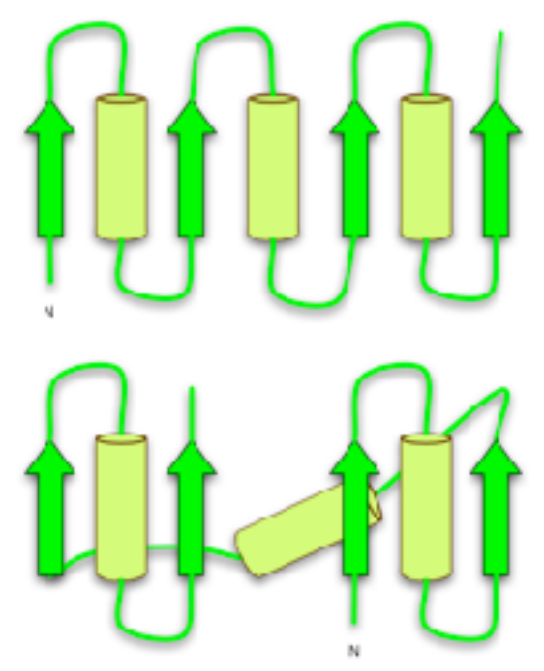
Topology (Fold Family)

For proteins with same Architecture, do they have same topology (ie. ordering of elements)

Performed automatically using SSAP and rules

SSAP score >70 and >60% of larger protein matches smaller

Similar architecture, different topology



CATH

Homologous Superfamily

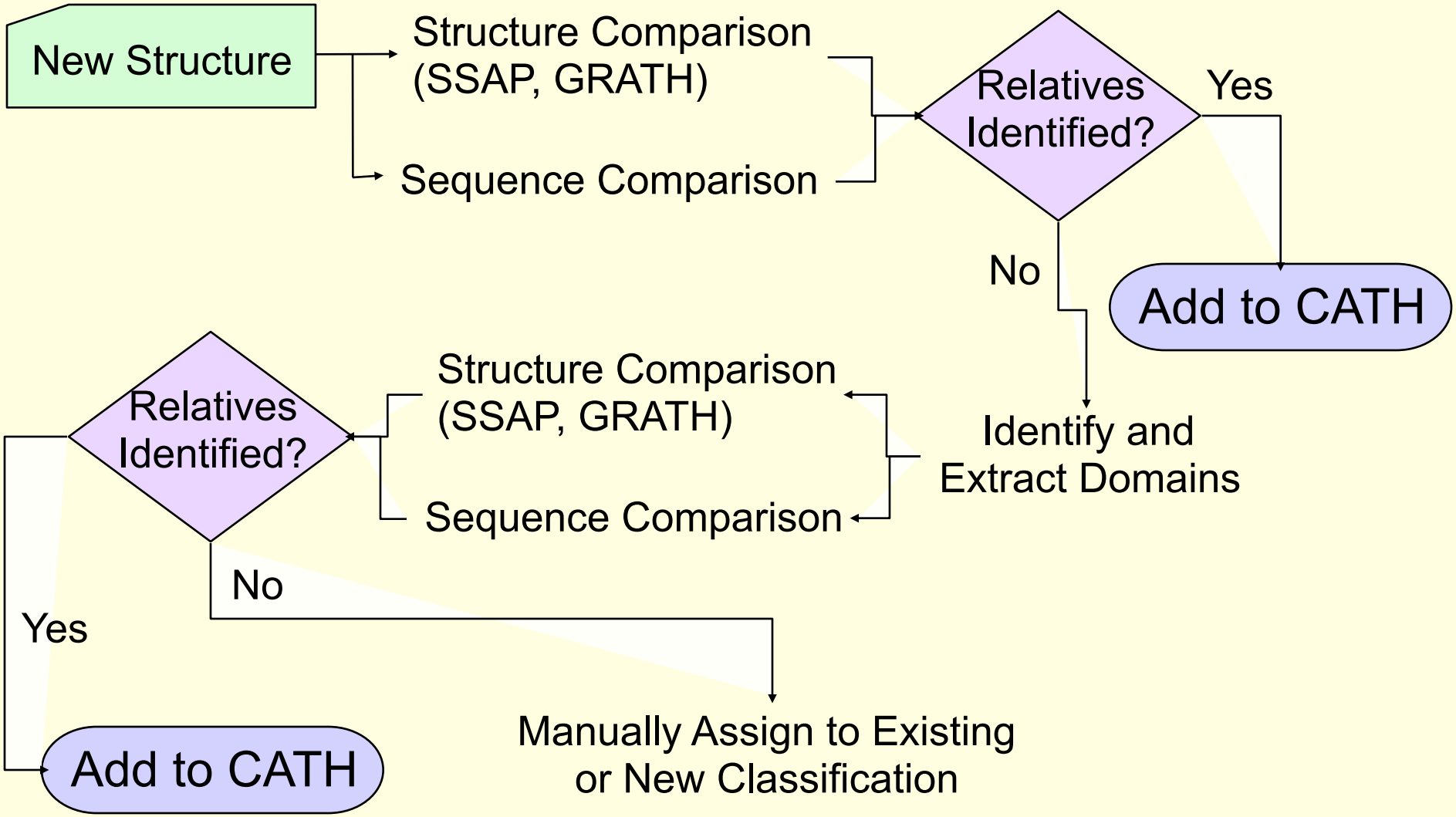
Structures grouped by evolutionary relationships
Includes sequence information
To be in same Homologous Superfamily:
Must have 60% of larger struct equivalent to smaller AND
Sequence identity >35% OR
SSAP score >80 and sequence identity >20% OR
SSAP score >80 and domains with related function

Statistics:

CATH v3.1
Jan, 2007

	C	A	T	H
Mainly Alpha	5	305	652
Mainly Beta	20	192	415
Mixed Alpha/Beta	14	496	922
Few Sec Struct	1	92	102

CATH


New Structure
Structure Comparison (SSAP, GRATH)
Sequence Comparison
Relatives Identified?
Yes
Add to CATH
No
Identify and Extract Domains
Structure Comparison (SSAP, GRATH)
Sequence Comparison
Relatives Identified?
No
Yes
Add to CATH
Manually Assign to Existing or New Classification