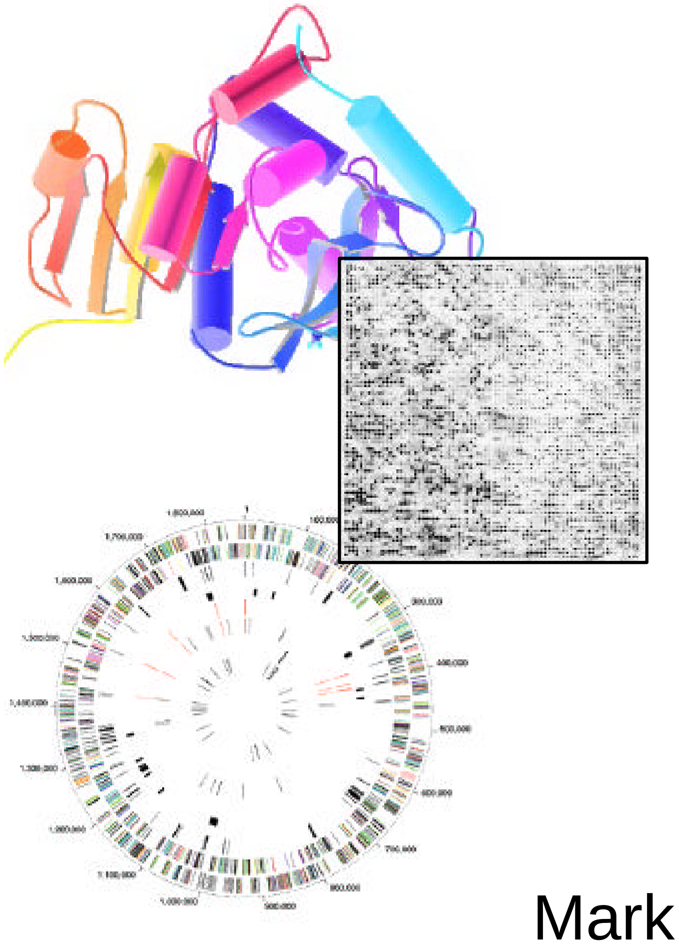
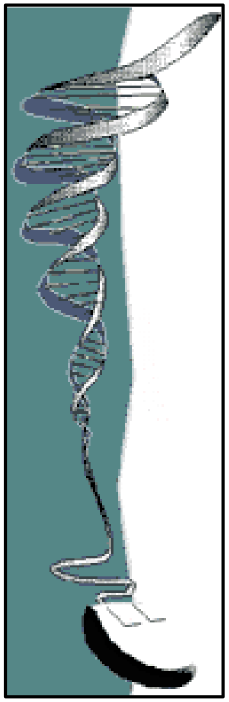
BIOINFORMATICS
Structures





Mark Gerstein, Yale University

bioinfo.mbb.yale.edu/mbb452a

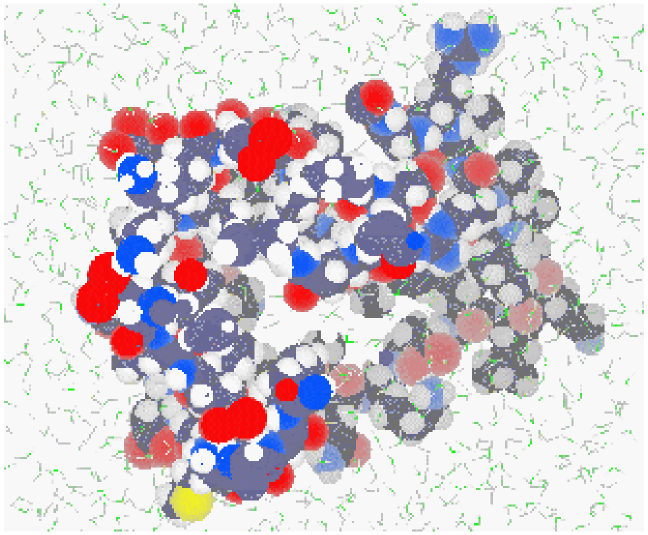
Contents: Structures

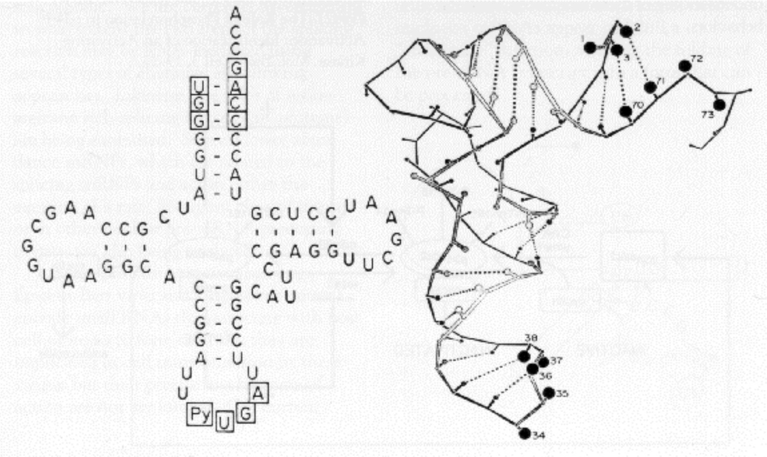
- What Structures Look Like?
- Structural Alignment by Iterated Dynamic Programming
    - ◊ RMS Superposition
- Scoring Structural Similarity
- Other Aspects of Structural Alignment
    - ◊ Distance Matrix based methods

Molecular Biology Information: Macromolecular Structure

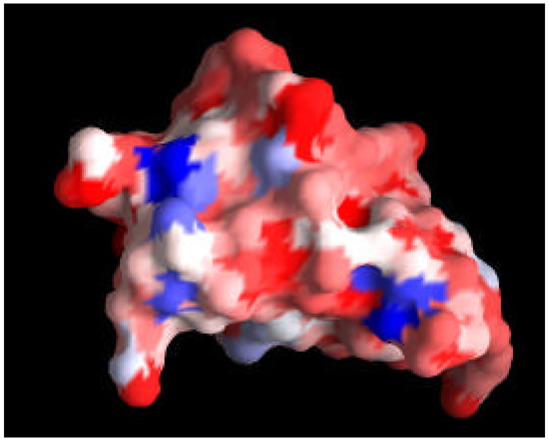
- DNA/RNA/Protein
  - ◊ Almost all protein

(RNA Adapted From D Soll Web Page,
Right Hand Top Protein from M Levitt web page)





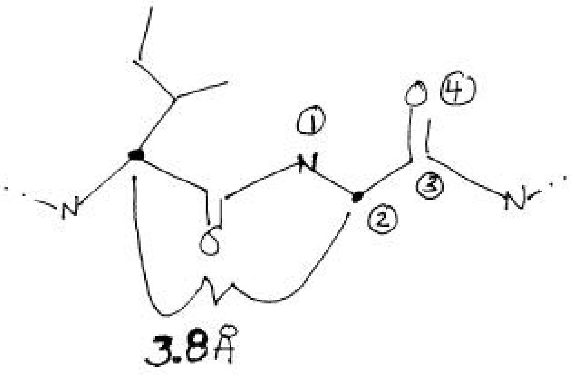
'Identity elements' in *Escherichia coli* glutamine tRNA.



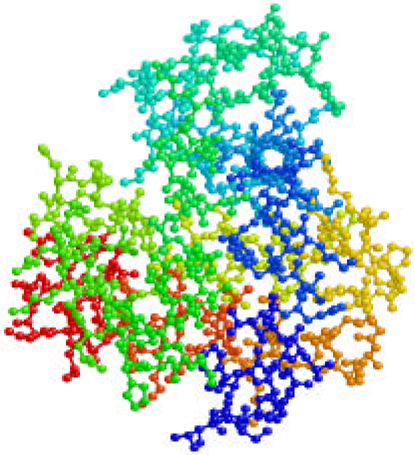
Molecular Biology Information: Protein Structure Details

- Statistics on Number of XYZ triplets
  - ◊ 200 residues/domain -> 200 CA atoms, separated by 3.8 A
  - ◊ Avg. Residue is Leu: 4 backbone atoms + 4 sidechain atoms, 150 cubic A
    - o => ~1500 xyz triplets (=8x200) per protein domain
  - ◊ 10 K known domain, ~300 folds

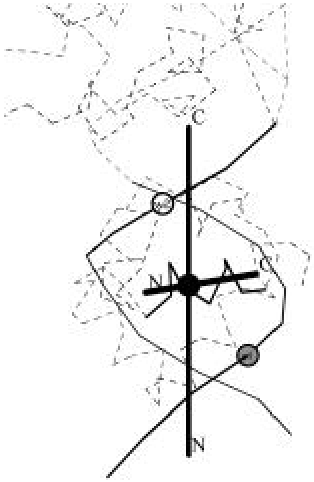
```
ATOM      1  C   ACE     0       9.401  30.166  60.595  1.00 49.88      1GKY  67
ATOM      2  O   ACE     0      10.432  30.832  60.722  1.00 50.35      1GKY  68
ATOM      3  CH3 ACE     0       8.876  29.767  59.226  1.00 50.04      1GKY  69
ATOM      4  N   SER     1       8.753  29.755  61.685  1.00 49.13      1GKY  70
ATOM      5  CA  SER     1       9.242  30.200  62.974  1.00 46.62      1GKY  71
ATOM      6  C   SER     1      10.453  29.500  63.579  1.00 41.99      1GKY  72
ATOM      7  O   SER     1      10.593  29.607  64.814  1.00 43.24      1GKY  73
ATOM      8  CB  SER     1       8.052  30.189  63.974  1.00 53.00      1GKY  74
ATOM      9  OG  SER     1       7.294  31.409  63.930  1.00 57.79      1GKY  75
ATOM     10  N   ARG     2      11.360  28.819  62.827  1.00 36.48      1GKY  76
ATOM     11  CA  ARG     2      12.548  28.316  63.532  1.00 30.20      1GKY  77
ATOM     12  C   ARG     2      13.502  29.501  63.500  1.00 25.54      1GKY  78
. . .
ATOM   1444  CB  LYS   186      13.836  22.263  57.567  1.00 55.06      1GKY1510
ATOM   1445  CG  LYS   186      12.422  22.452  58.180  1.00 53.45      1GKY1511
ATOM   1446  CD  LYS   186      11.531  21.198  58.185  1.00 49.88      1GKY1512
ATOM   1447  CE  LYS   186      11.452  20.402  56.860  1.00 48.15      1GKY1513
ATOM   1448  NZ  LYS   186      10.735  21.104  55.811  1.00 48.41      1GKY1514
ATOM   1449  OXT LYS   186      16.887  23.841  56.647  1.00 62.94      1GKY1515
TER    1450      LYS   186                                          1GKY1516
```



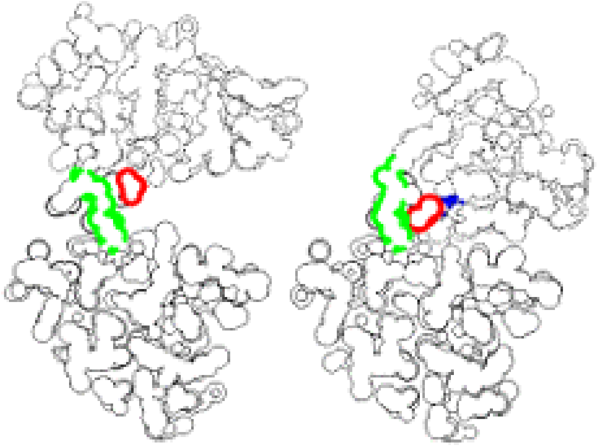
Other Aspects of Structure, Besides just Comparing Atom Positions



Atom
Position,
XYZ triplets



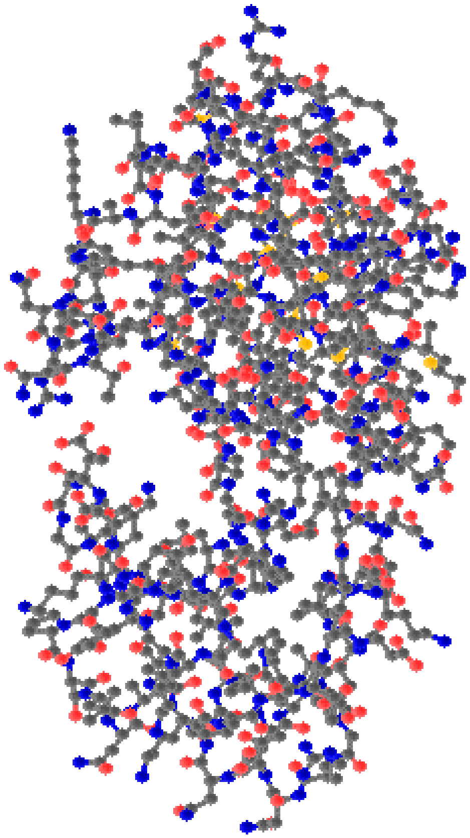
Lines, Axes,
Angles



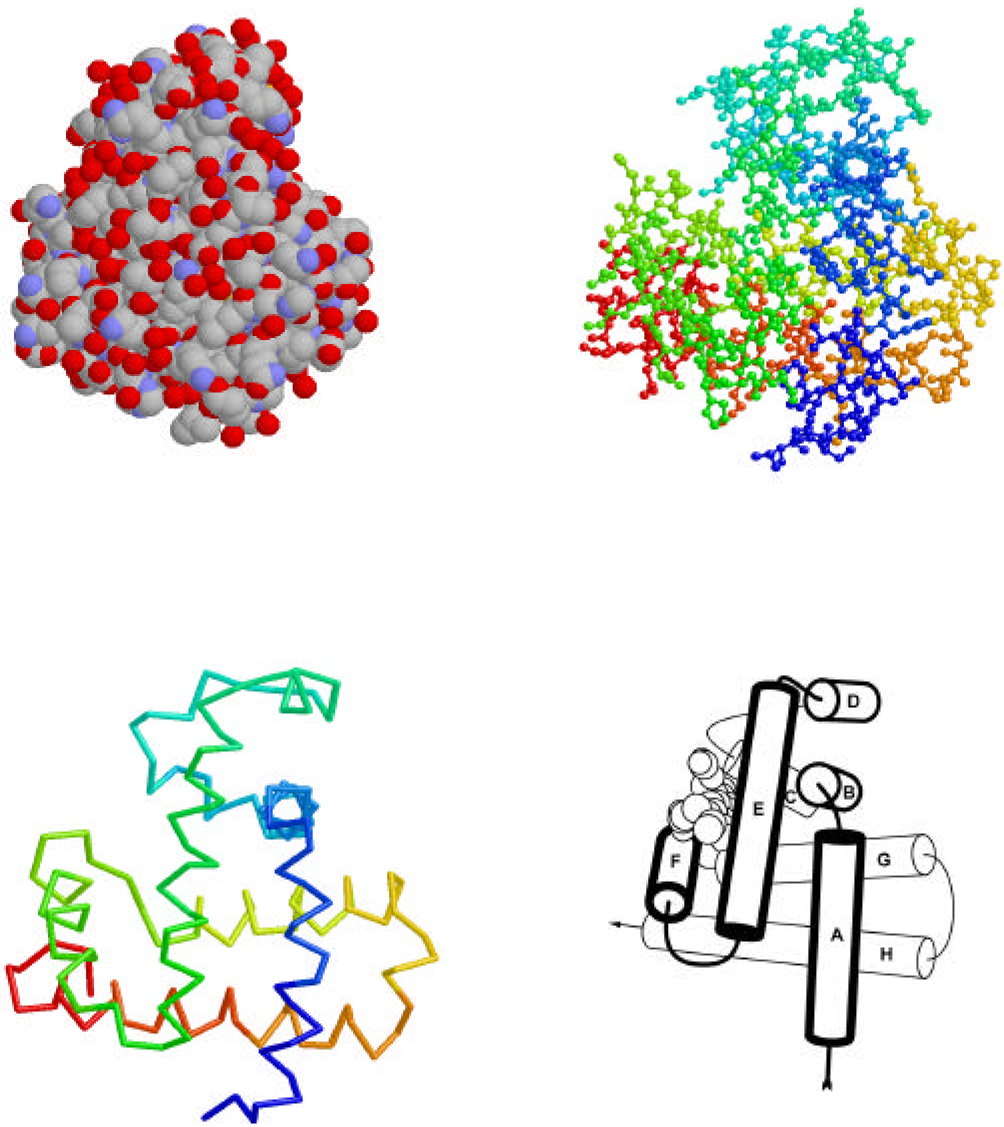
Surfaces, Volumes

What is Protein Geometry?

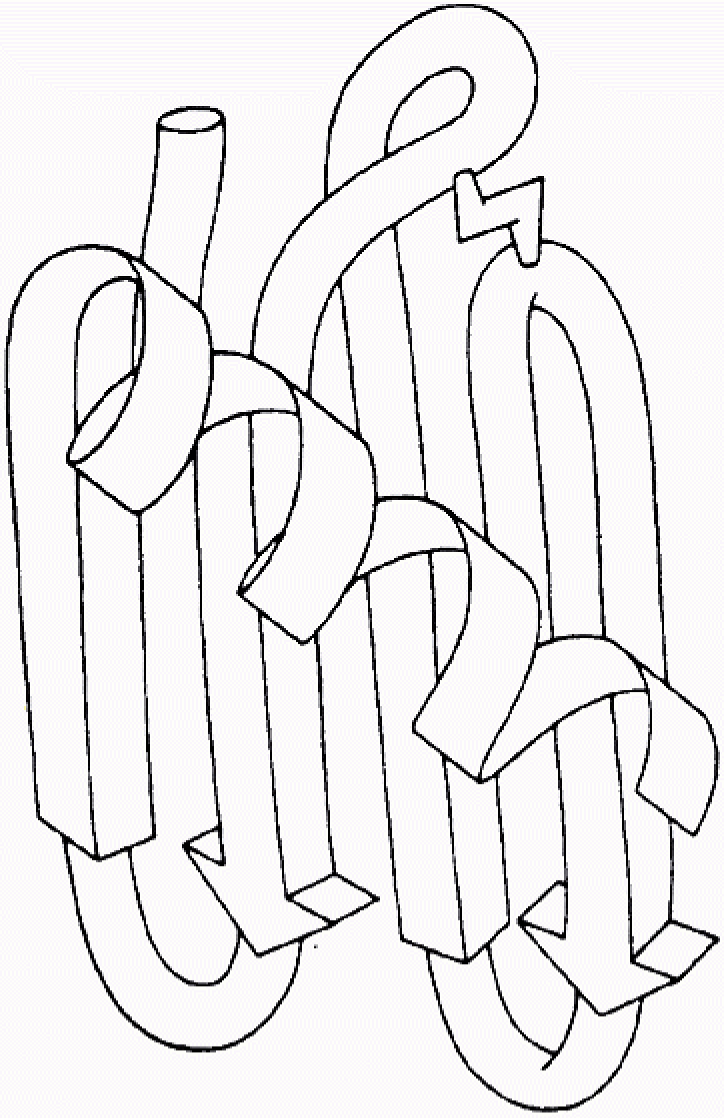
- Coordinates (X, Y, Z's)
- Derivative Concepts
    - Distance, Surface Area, Volume, Cavity, Groove, Axes, Angle, &c
- Relation to
    - Function, Energies (E(x)), Dynamics (dx/dt)



Depicting Protein Structure: Sperm Whale Myoglobin



Incredulase



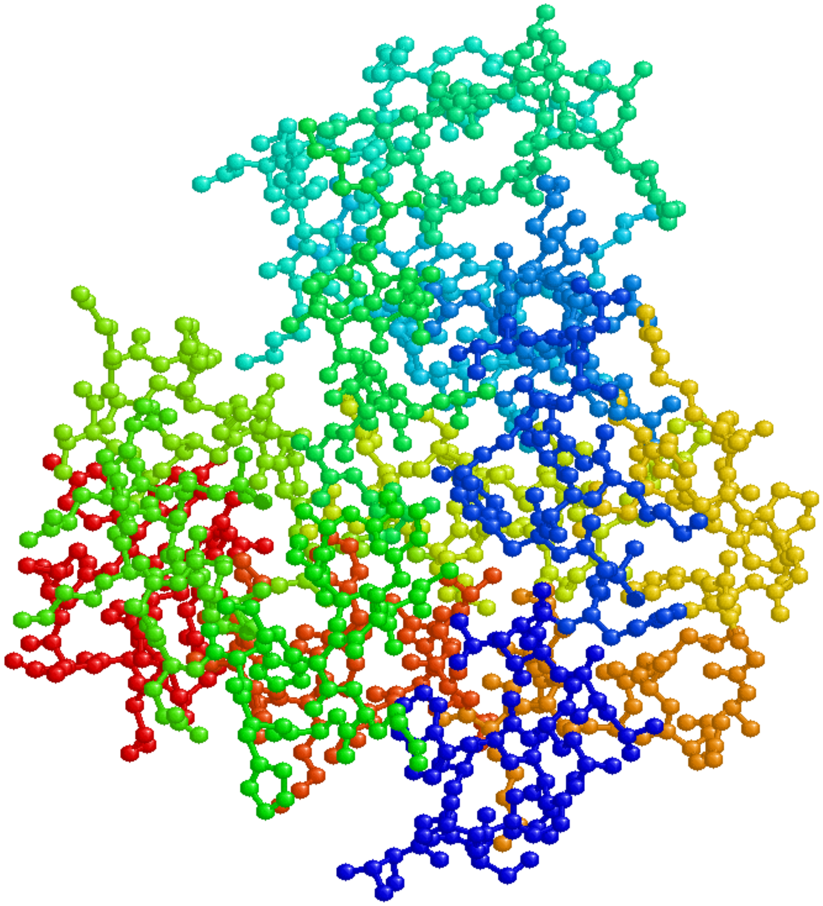
Incredulase

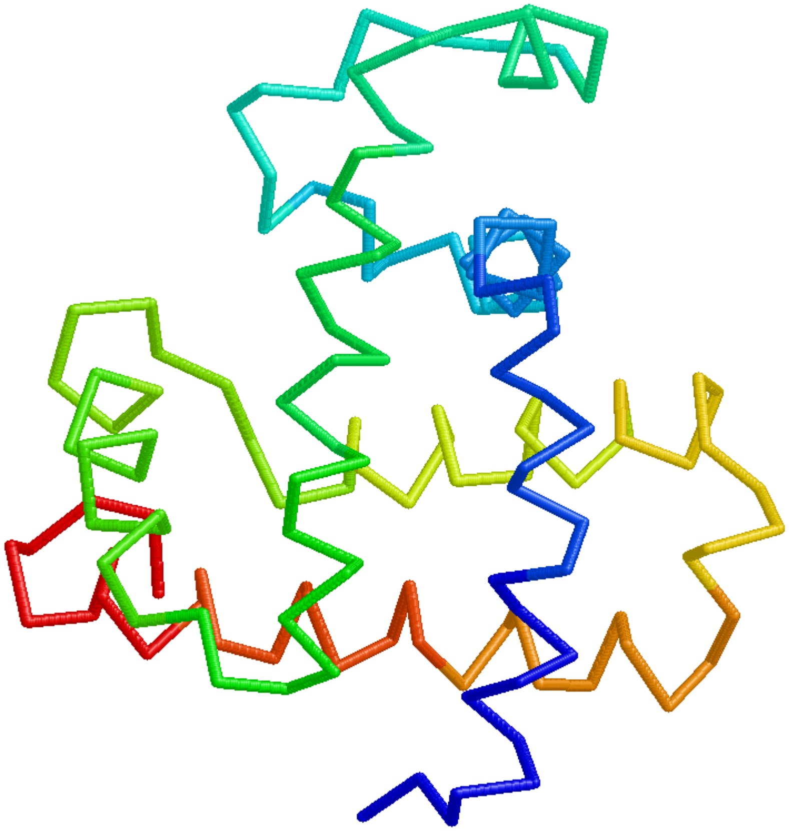
J.S. Richardson and D.C. Richardson, "*Some design principles: Betabellin*", in D.L. Oxender and C.F. Fox (Eds.), "*Protein Engineering*", Alan R. Liss, 1987, p. 149-163

Structure alignment - Method

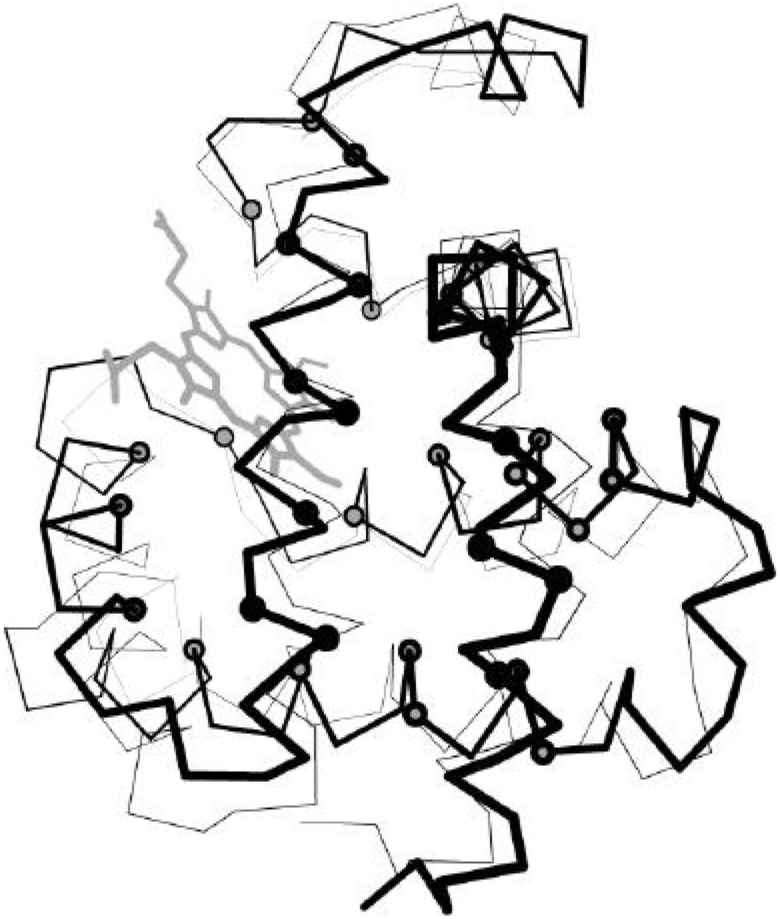
- What Structures Look Like?
- Structural Alignment by Iterated Dynamic Programming
    - ◊ RMS Superposition
- Scoring Structural Similarity
- Other Aspects of Structural Alignment
    - ◊ Distance Matrix based methods
    - ◊ Fold Library
- Relation of Sequence Similarity to Structural and Functional Similarity
- Protein Geometry
- Surface I (Calculation)
- Calculation of Volume
- Voronoi Volumes & Packing
- Standard Volumes & Radii
- Surfaces II (Relationship to Volumes)
- Other Applications of Volumes -- Motions, Docking

Sperm Whale Myoglobin

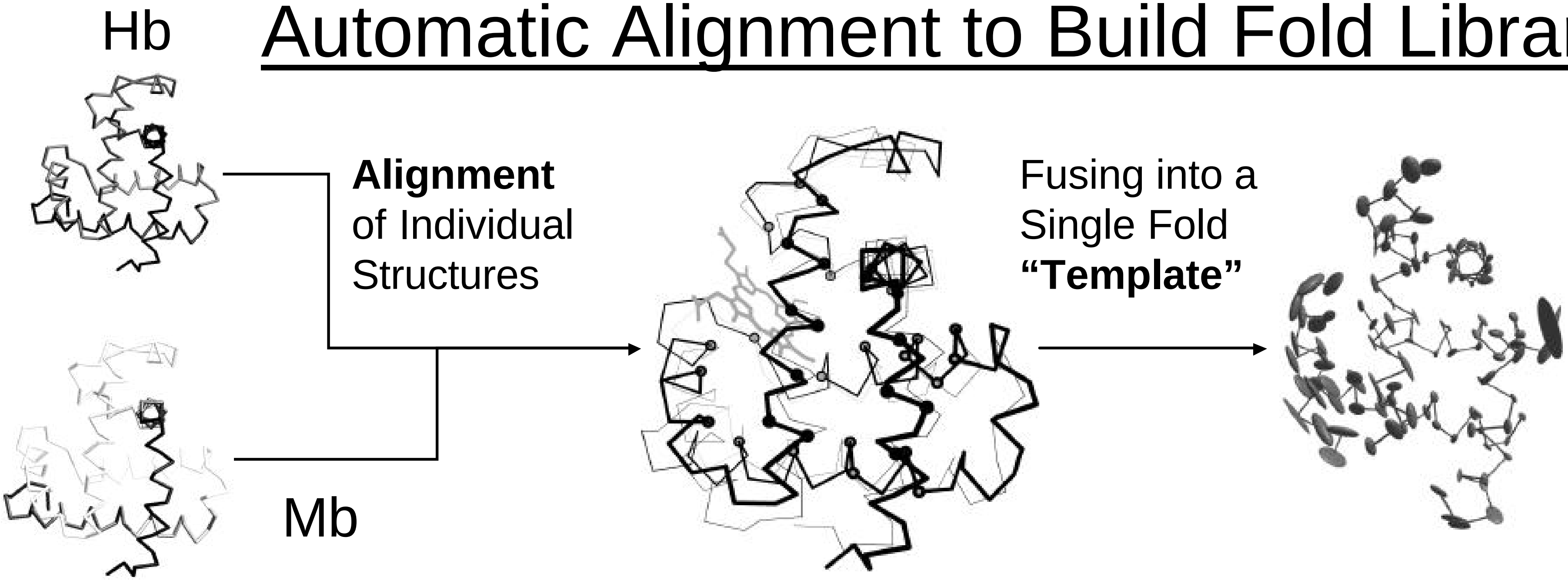




Structural Alignment of Two Globins



Automatic Alignment to Build Fold Library

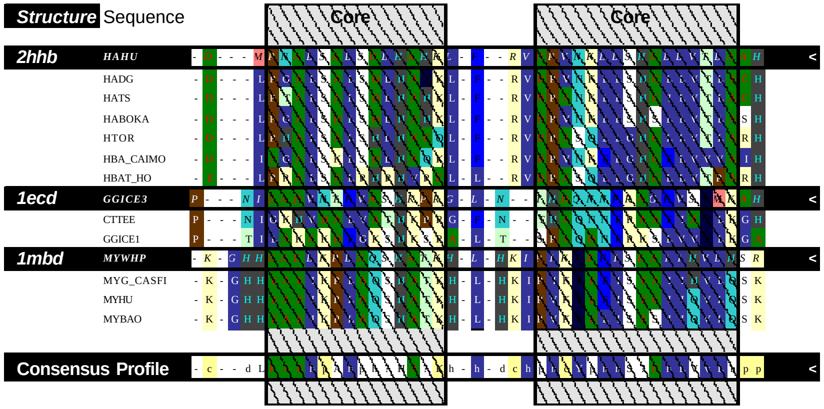


```
Hb VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSAQVKGHGKKVADALTNAV
      ||| ..   |   |.|| |  .  | .  | | | |   |  |  |       .|  .| ||  |  ||   .
Mb VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKHGVTVLTALGAIL

Hb AHVD-DMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------
             |  | . || | .. .     .|  ..  |    |..|     . . | |      .  ||.
Mb KK-KGHHEAELKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRKDIAAKYKELGYQG
```

Elements: Domain definitions; Aligned structures, collecting together Non-homologous Sequences; Core annotation

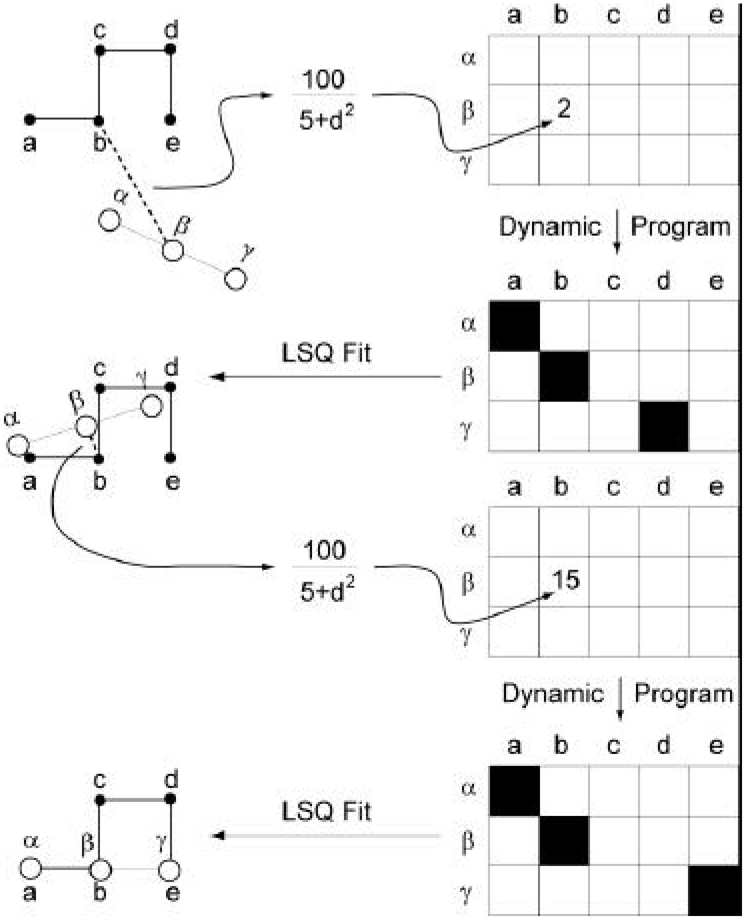
Previous work: Remington, Matthews '80; **Taylor, Orengo '89**, '94; Artymiuk, Rice, Willett '89; Sali, Blundell, '90; Vriend, Sander '91; Russell, Barton '92; **Holm, Sander '93**; Godzik, Skolnick '94; Gibrat, Madej, Bryant '96; Falicov, F Cohen, '96; Feng, Sippl '96; G Cohen '97; Singh & Brutlag, '98



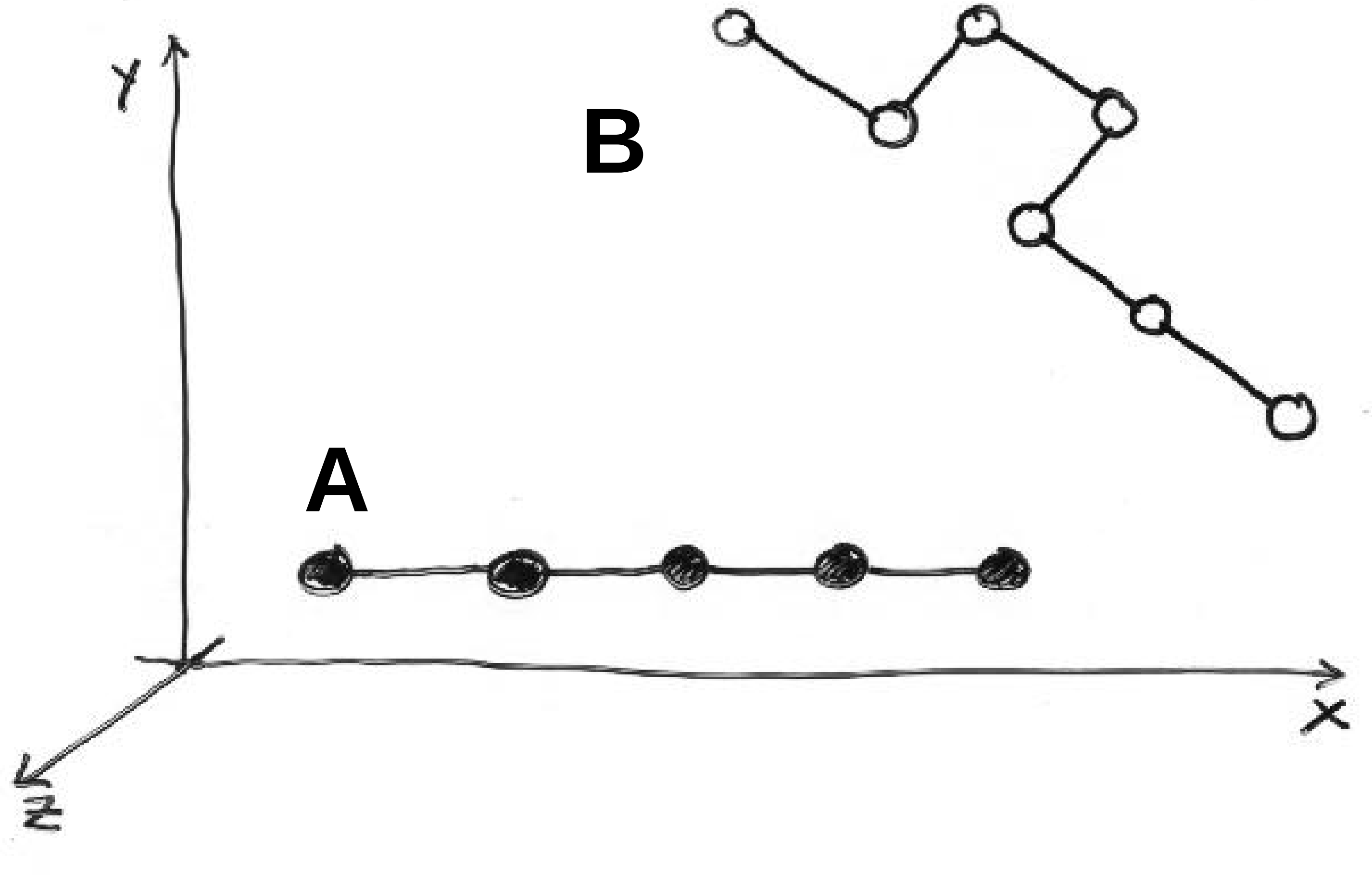
Automatically Comparing Protein Structures

- Given
  2 Structures (A & B),
  2 Basic
  Comparison Operations
  1. Given an alignment optimally **SUPERIMPOSE** A onto B

     Find Best R & T to move A onto B
  2. **Find an Alignment** between A and B based on their 3D coordinates

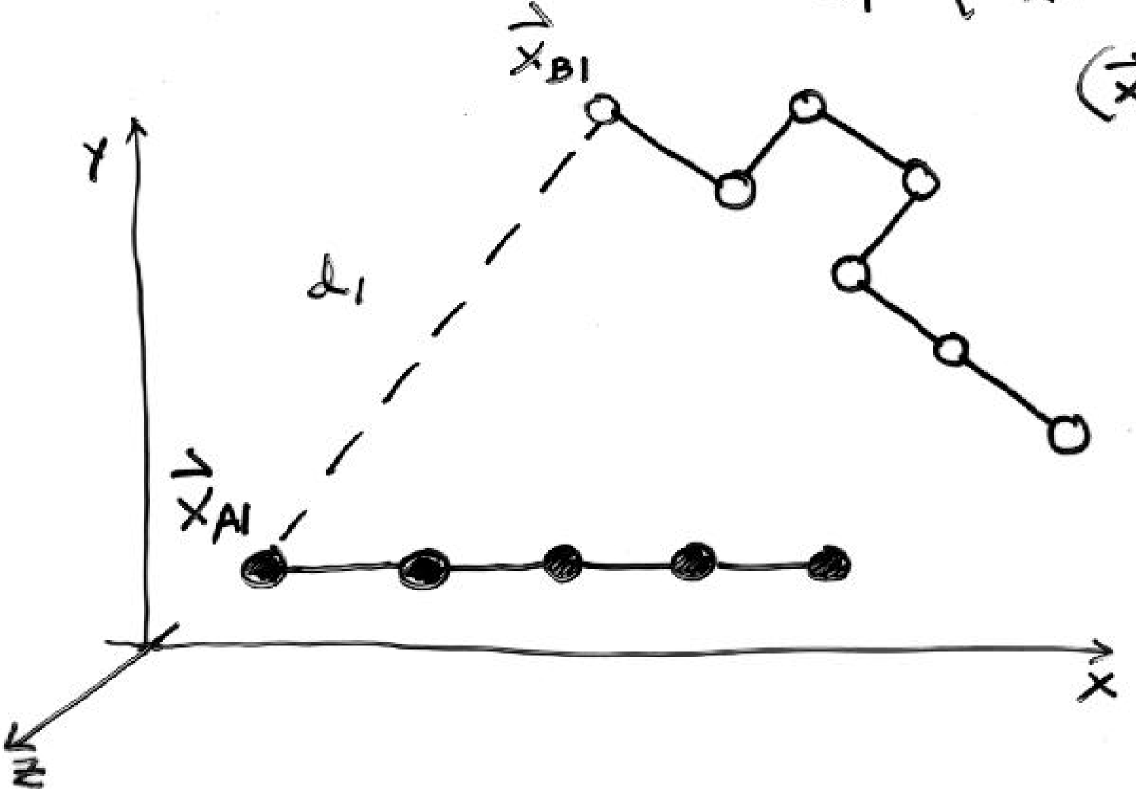


RMS Superposition (1)



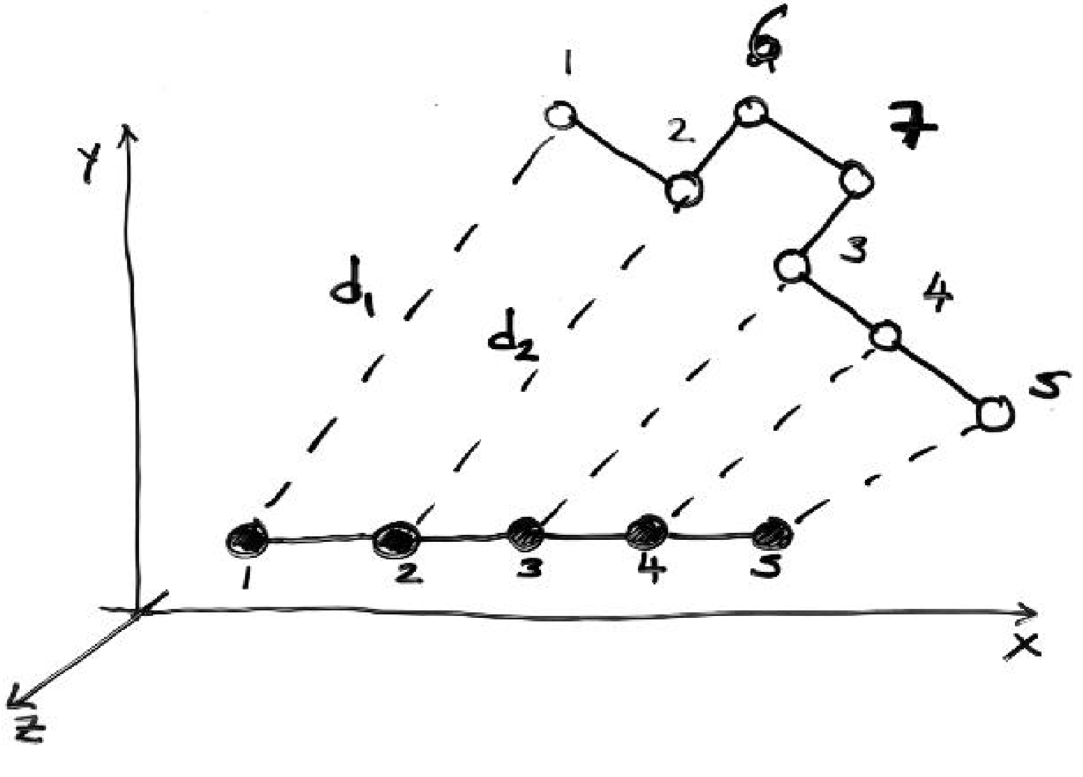
RMS Superposition (2): Distance Between an Atom in 2 Structures

$$d_1^2 = (\vec{x}_{A1} - \vec{x}_{B1}) \cdot (\vec{x}_{A1} - \vec{x}_{B1})$$



RMS Superposition (3): RMS Distance Between Aligned Atoms in 2 Structures

$$RMS = \sqrt{\frac{\sum_{i=1}^{5} (\vec{x}_{Ai} - \vec{x}_{Bi})^2}{5}} \sim \frac{d_1 + d_2 + d_3 + d_4 + d_5}{5}$$



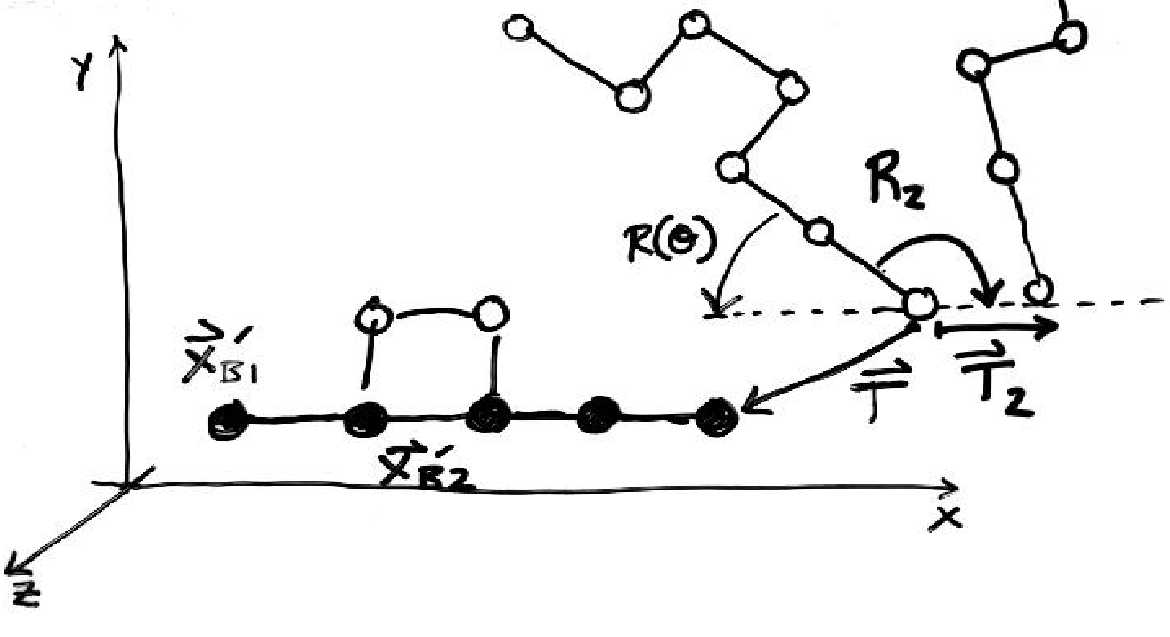
RMS Superposition (4): Rigid-Body Rotation and Translation of One Structure (B)

$$\vec{x}_{Bi}' = R(\Theta)\vec{x}_{Bi} + \vec{T}$$

ROTATE & TRANSLATE

$\boxed{6}$ parameters

$\vec{T} = (T_x\ T_y\ T_z)\quad R(\Theta, \varphi, \psi)$



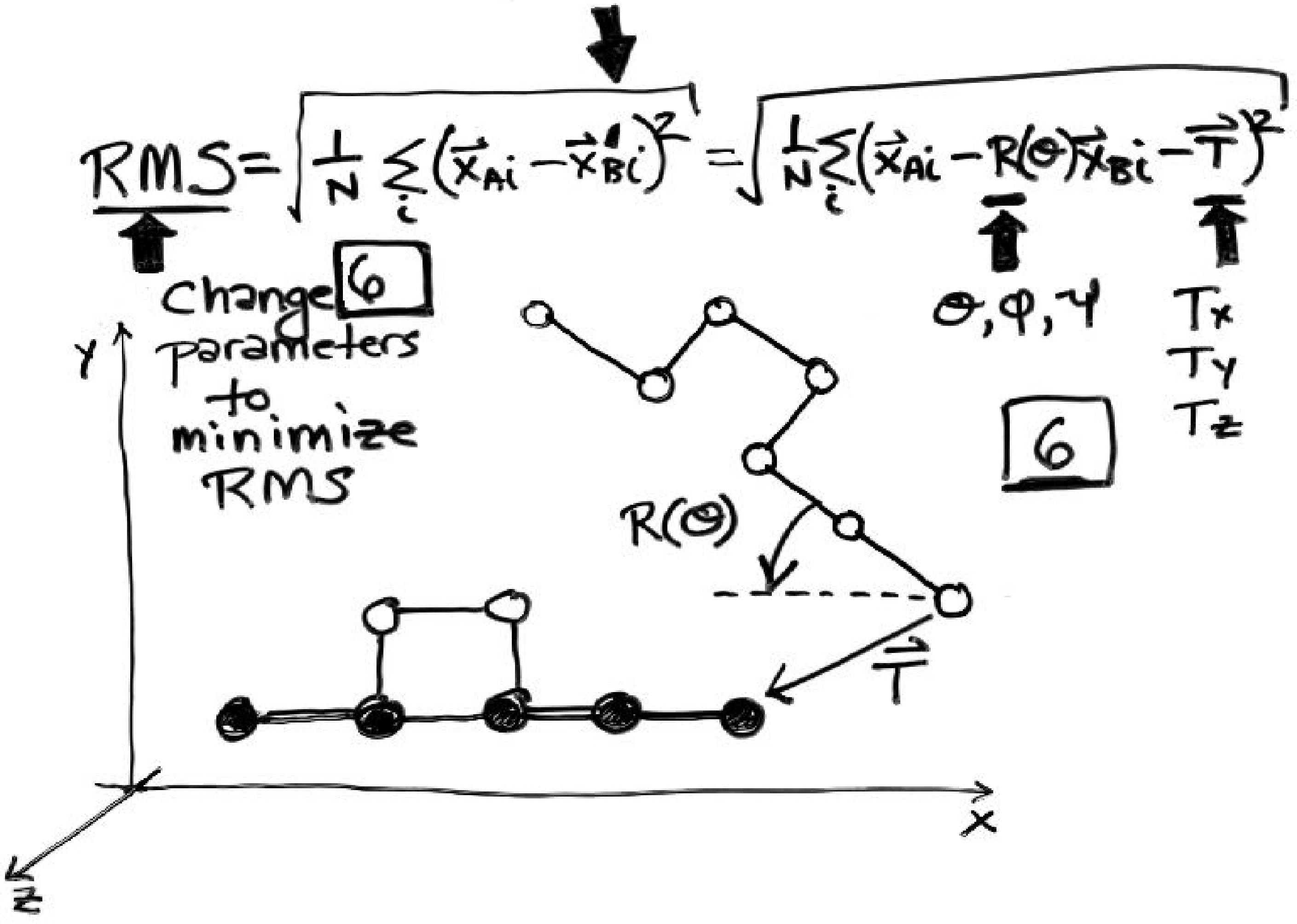
RMS Superposition (5): Optimal Movement of One Structure to Minimize the RMS

Methods of Solution:

springs (F ~ kx)

SVD

Kabsch



$$\underline{RMS} = \sqrt{\frac{1}{N}\sum_i (\vec{x}_{Ai} - \vec{x}_{Bi})^2} = \sqrt{\frac{1}{N}\sum_i (\vec{x}_{Ai} - R(\theta)\vec{x}_{Bi} - \vec{T})^2}$$

Alignment (1)
Make a Similarity Matrix
(Like Dot Plot)

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						1					1		
C			1					1		1			
K													
C			1					1		1			
R						1					1		
B		1											
P												1	

Structural Alignment (1b) Make a Similarity Matrix

(Generalized Similarity Matrix)

- PAM(A,V) = 0.5
  - ◊ Applies at every position
- S(aa @ i, aa @ J)
  - ◊ Specific Matrix for each pair of residues
    i in protein 1 and
    J in protein 2
  - ◊ Example is Y near N-term. matches any C-term. residue (Y at J=2)
- S(i,J)
  - ◊ Doesn't need to depend on a.a. identities at all!
  - ◊ Just need to make up a score for matching residue i in protein 1 with residue J in protein 2

i →

		1	2	3	4	5	6	7	8	9	10	11	12	13
		A	B	C	N	Y	R	Q	C	L	C	R	P	M
1	A	**1**												
2	**Y**					**1**			**5**	**5**	**5**	**5**	**5**	**5**
3	C			**1**					**1**		**1**			
4	Y					**1**								
5	N				**1**									
6	R						**1**					**1**		
7	C			**1**					**1**		**1**			
8	K													
9	C			**1**					**1**		**1**			
10	R						**1**					**1**		
11	B		**1**											
12	P												**1**	

J ↓

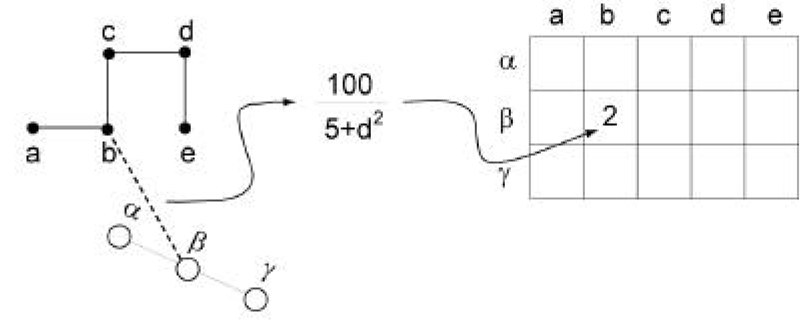
Structural Alignment (1c*) Similarity Matrix for Structural Alignment

- Structural Alignment
  - ◊ Similarity Matrix S(i,J) depends on the 3D coordinates of residues i and J
  - ◊ Distance between CA of i and J

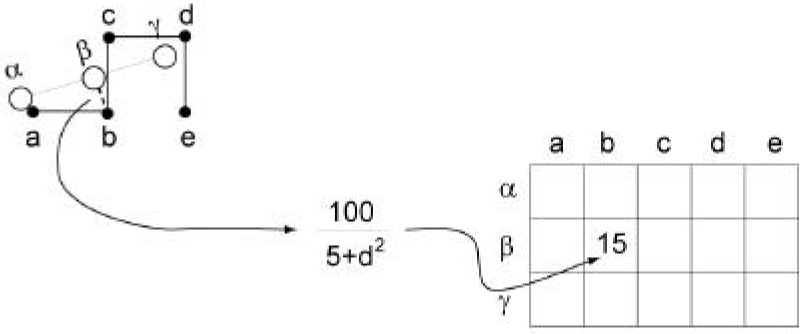
$$d = \sqrt{(x_i - x_J)^2 + (y_i - y_J)^2 + (z_i - z_J)^2}$$

  - ◊ M(i,j) = 100 / (5 + d^2)

- Threading
  - ◊ S(i,J) depends on the how well the amino acid at position i in protein 1 fits into the 3D structural environment at position J of protein 2



	a	b	c	d	e
α					
β		2			
γ					



	a	b	c	d	e
α					
β		15			
γ					

Alignment (2): Dynamic Programming, Start Computing the Sum Matrix

```
new_value_cell(R,C) <=
  cell(R,C)                      { Old value, either 1 or 0    }
  + Max[
        cell (R+1, C+1),         { Diagonally Down, no gaps    }
        cells(R+1, C+2 to C_max),{ Down a row, making col. gap }
        cells(R+2 to R_max, C+2) { Down a col., making row gap }
       ]
```

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	**1**												
Y					**1**								
C			**1**					**1**		**1**			
Y					**1**								
N				**1**									
R						**1**					**1**		
C			**1**					**1**		**1**			
K													
C			**1**					**1**		**1**			
R						**1**					**1**		
B		**1**											
P												**1**	

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						1					**1**		
C			1					1		1			
K													
C			1					1		1			
R						1					***2***	**0**	**0**
B	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
P	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**

Alignment (3):Dynamic Programming, Keep Going

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						1					**1**		
C			1					1		1			
K													
C			1					1		1			
R						1					***2***	**0**	**0**
B	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
P	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						***5***	**4**	**3**	**3**	**2**	**2**	**0**	**0**
C	**3**	**3**	**4**	**3**	**3**	**3**	**3**	**4**	**3**	**3**	**1**	**0**	**0**
K	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**2**	**1**	**0**	**0**
C	**2**	**2**	**3**	**2**	**2**	**2**	**2**	**3**	**2**	**3**	**1**	**0**	**0**
R	**2**	**1**	**1**	**1**	**1**	**2**	**1**	**1**	**1**	**1**	**2**	**0**	**0**
B	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
P	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**

Alignment (4): Dynamic Programming, Sum Matrix All Done

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	1												
Y					1								
C			1					1		1			
Y					1								
N				1									
R						*5*	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	*8*	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
Y	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

Alignment (5): Traceback

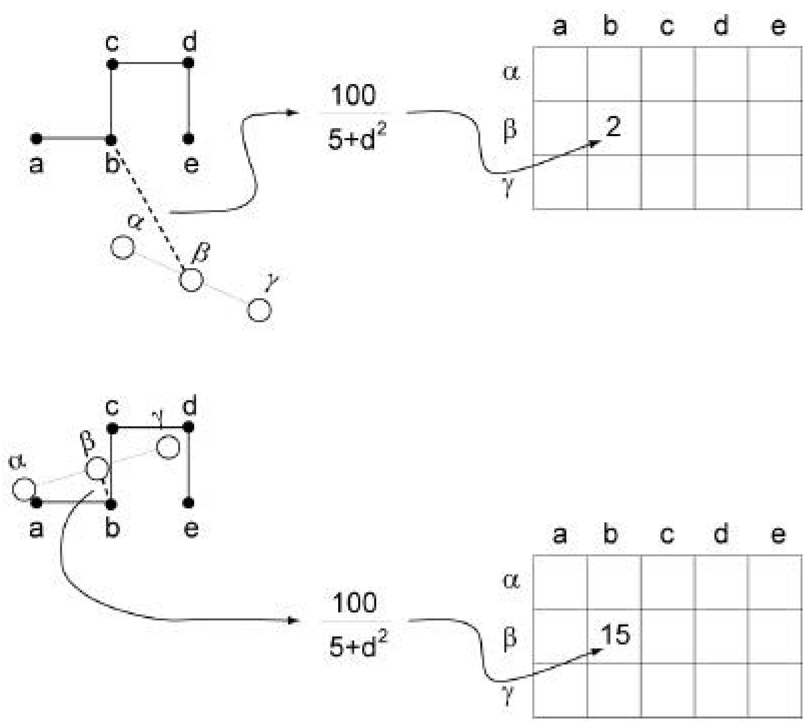
Find Best Score (8) and Trace Back

```
A B C N Y - R Q C L C R - P M
A Y C - Y N R - C K C R B P
```

	A	B	C	N	Y	R	Q	C	L	C	R	P	M
A	8	7	6	6	5	4	4	3	3	2	1	0	0
Y	7	7	6	6	6	4	4	3	3	2	1	0	0
C	6	6	7	6	5	4	4	4	3	3	1	0	0
Y	6	6	6	5	6	4	4	3	3	2	1	0	0
N	5	5	5	6	5	4	4	3	3	2	1	0	0
R	4	4	4	4	4	5	4	3	3	2	2	0	0
C	3	3	4	3	3	3	3	4	3	3	1	0	0
K	3	3	3	3	3	3	3	3	3	2	1	0	0
C	2	2	3	2	2	2	2	3	2	3	1	0	0
R	2	1	1	1	1	2	1	1	1	1	2	0	0
B	1	2	1	1	1	1	1	1	1	1	1	0	0
P	0	0	0	0	0	0	0	0	0	0	0	1	0

In Structural Alignment, Not Yet Done (Step 6*)

- Use Alignment to LSQ Fit Structure B onto Structure A
  - ◊ However, movement of B will now change the Similarity Matrix
- This Violates Fundamental Premise of Dynamic Programming
  - ◊ Way Residue at i is aligned can now affect previously optimal alignment of residues (from 1 to i-1)



```
ACSQRP--LRV-SH -R SENCV
A-SNKPQLVKLMTH VK DFCV-
```

Structural Alignment (7*), Iterate Until Convergence

1 Compute Sim. Matrix

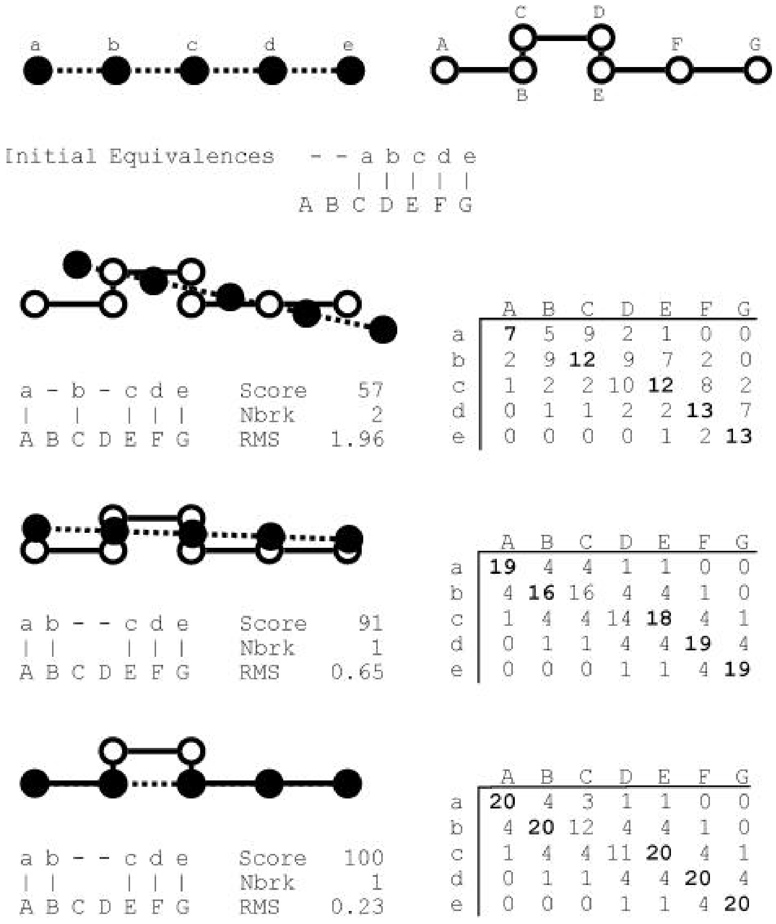
2 Align via Dyn. Prog.

3 RMS Fit Based on Alignment

4 Move Structure B

5 Re-compute Sim. Matrix

6 If changed from #1, GOTO #2



```
Initial Equivalences   - - a b c d e
                           | | | | |
                       A B C D E F G
```

```
a - b - c d e    Score     57
|   |   | | |    Nbrk       2
A B C D E F G    RMS     1.96
```

	A	B	C	D	E	F	G
a	**7**	5	9	2	1	0	0
b	2	9	**12**	9	7	2	0
c	1	2	2	10	**12**	8	2
d	0	1	1	2	2	**13**	7
e	0	0	0	0	1	2	**13**

```
a b - - c d e    Score     91
| |     | | |    Nbrk       1
A B C D E F G    RMS     0.65
```

	A	B	C	D	E	F	G
a	**19**	4	4	1	1	0	0
b	4	**16**	16	4	4	1	0
c	1	4	4	14	**18**	4	1
d	0	1	1	4	4	**19**	4
e	0	0	0	1	1	4	**19**

```
a b - - c d e    Score    100
| |     | | |    Nbrk       1
A B C D E F G    RMS     0.23
```

	A	B	C	D	E	F	G
a	**20**	4	3	1	1	0	0
b	4	**20**	12	4	4	1	0
c	1	4	4	11	**20**	4	1
d	0	1	1	4	4	**20**	4
e	0	0	0	1	1	4	**20**

Structure alignment - Scoring

- What Structures Look Like?
- Structural Alignment by Iterated Dynamic Programming
  - ◊ RMS Superposition
- Scoring Structural Similarity
- Other Aspects of Structural Alignment
  - ◊ Distance Matrix based methods
  - ◊ Fold Library
- Relation of Sequence Similarity to Structural and Functional Similarity

- Protein Geometry
- Surface I (Calculation)
- Calculation of Volume
- Voronoi Volumes & Packing
- Standard Volumes & Radii
- Surfaces II (Relationship to Volumes)
- Other Applications of Volumes -- Motions, Docking

Score S at End Just Like SW Score, but also have final RMS

S = Total Score

S(i,j) = similarity matrix score for aligning i and j

Sum is carried out over all aligned i and j

n = number of gaps (assuming no gap ext. penalty)

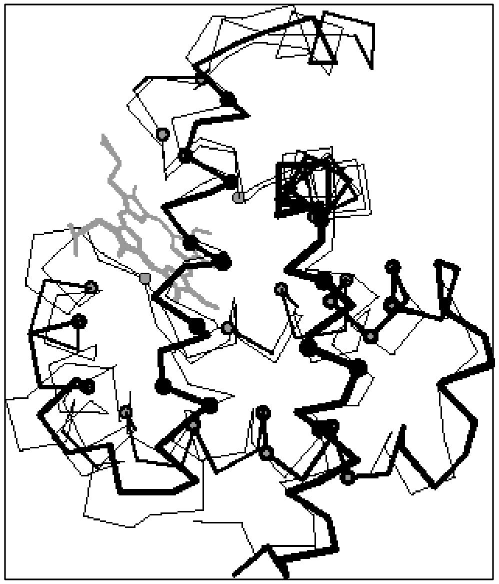
G = gap penalty

$$S = \sum_{i,j} S(i,j) - nG$$

Some Similarities are Readily Apparent others are more Subtle

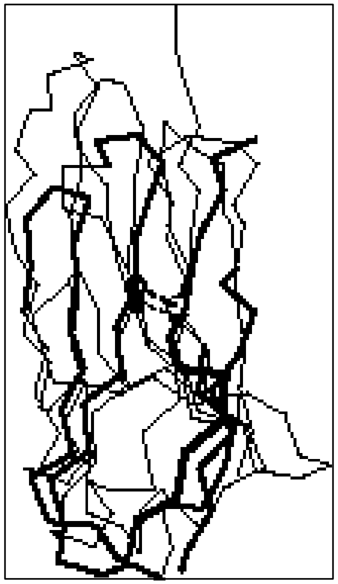
Easy:
Globins

125 res.,
~1.5 Å

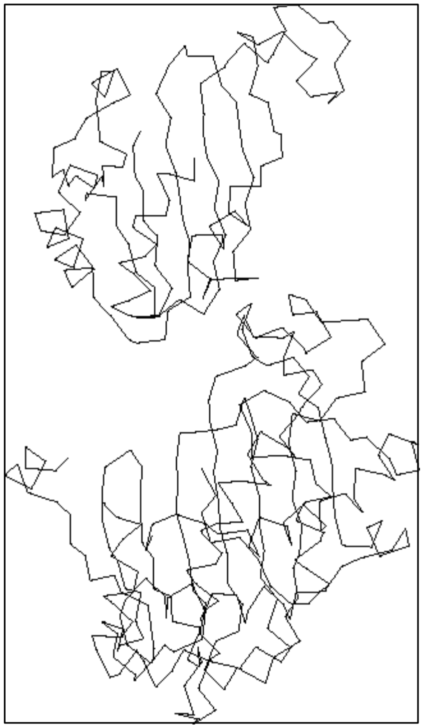


Tricky:
Ig C & V

85 res.,
~3 Å



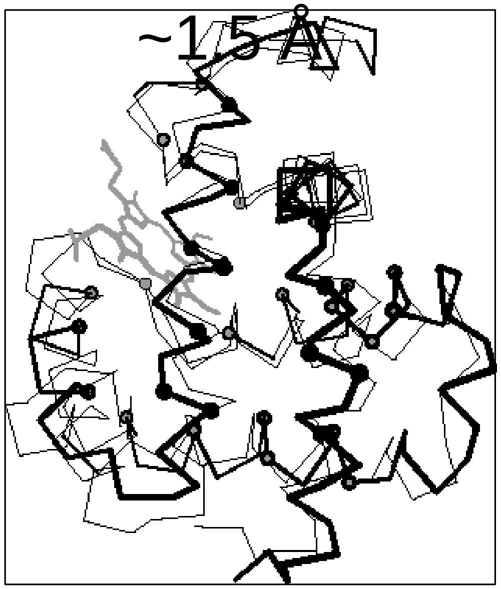
Very Subtle: G3P-dehydrogenase, C-term. Domain
>5 Å



Some Similarities are Readily Apparent others are more Subtle

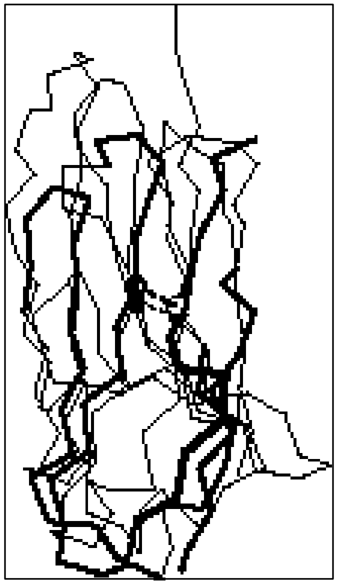
Easy: Globins

125 res., ~1.5 Å



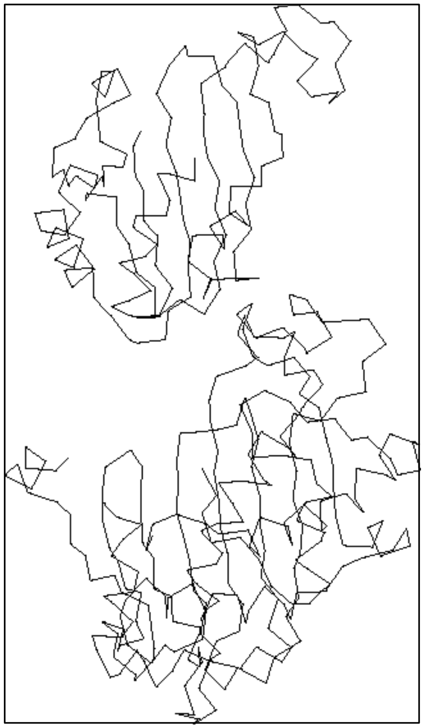
Tricky: Ig C & V

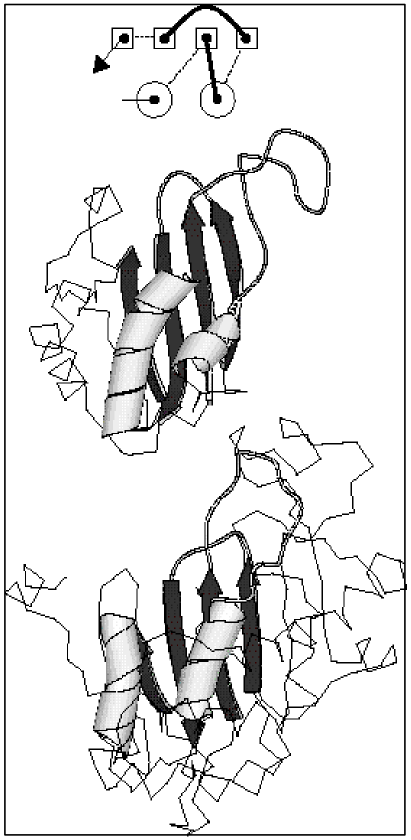
85 res., ~3 Å



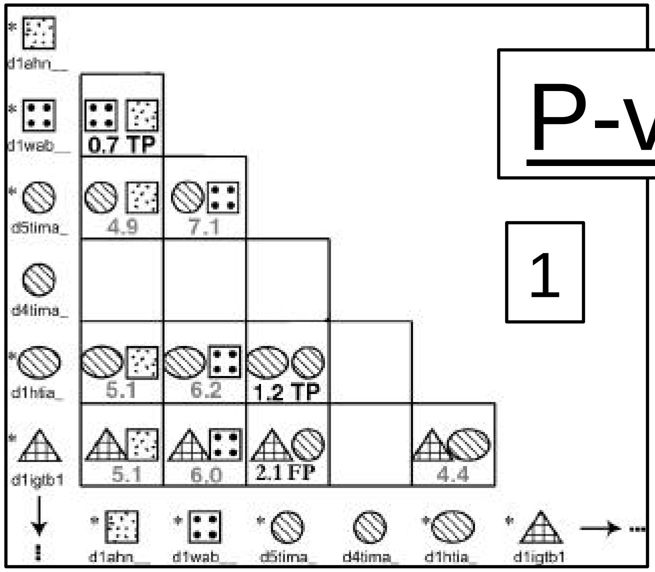
Very Subtle: G3P-dehydrogenase, C-term. Domain

>5 Å

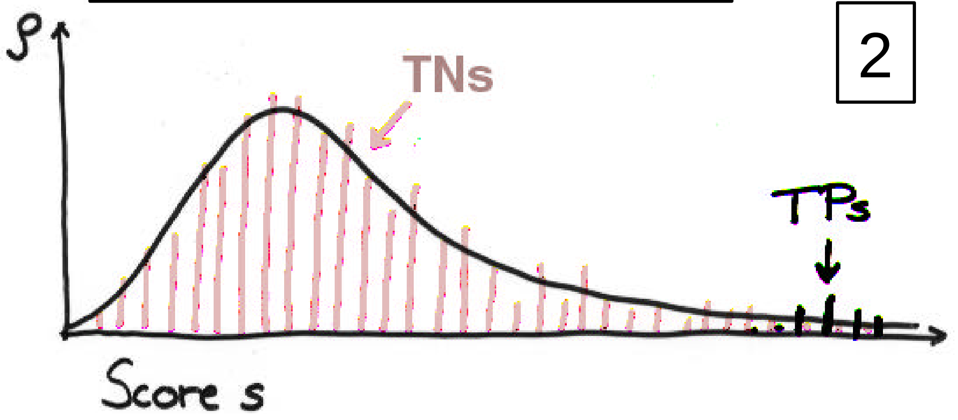




P-values

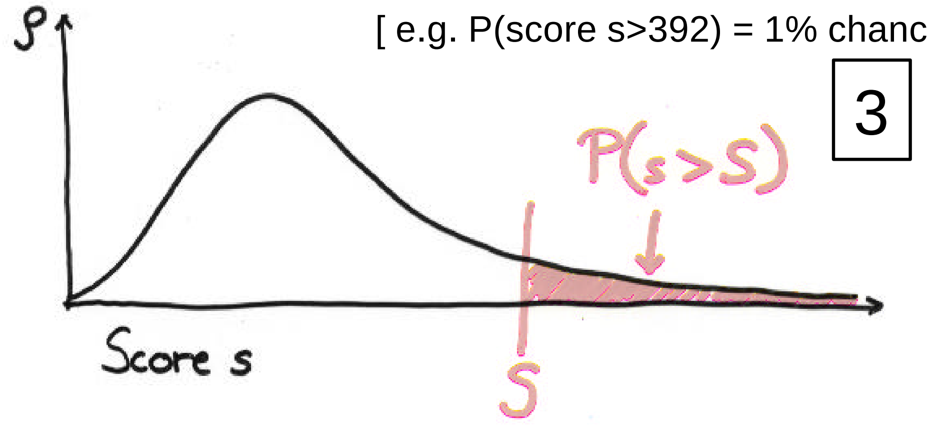


1



2

[e.g. P(score s>392) = 1% chance]



3

- Significance Statistics
  - ◊ For sequences, originally used in Blast (Karlin-Altschul). Then in FASTA, &c.
  - ◊ Extrapolated Percentile Rank: How does a Score Rank Relative to all Other Scores?
- Our Strategy: Fit to Observed Distribution

1) All-vs-All comparison

2) Graph Distribution of Scores in 2D (N dependence); 1K x 1K families -> ~1M scores; ~2K included TPs

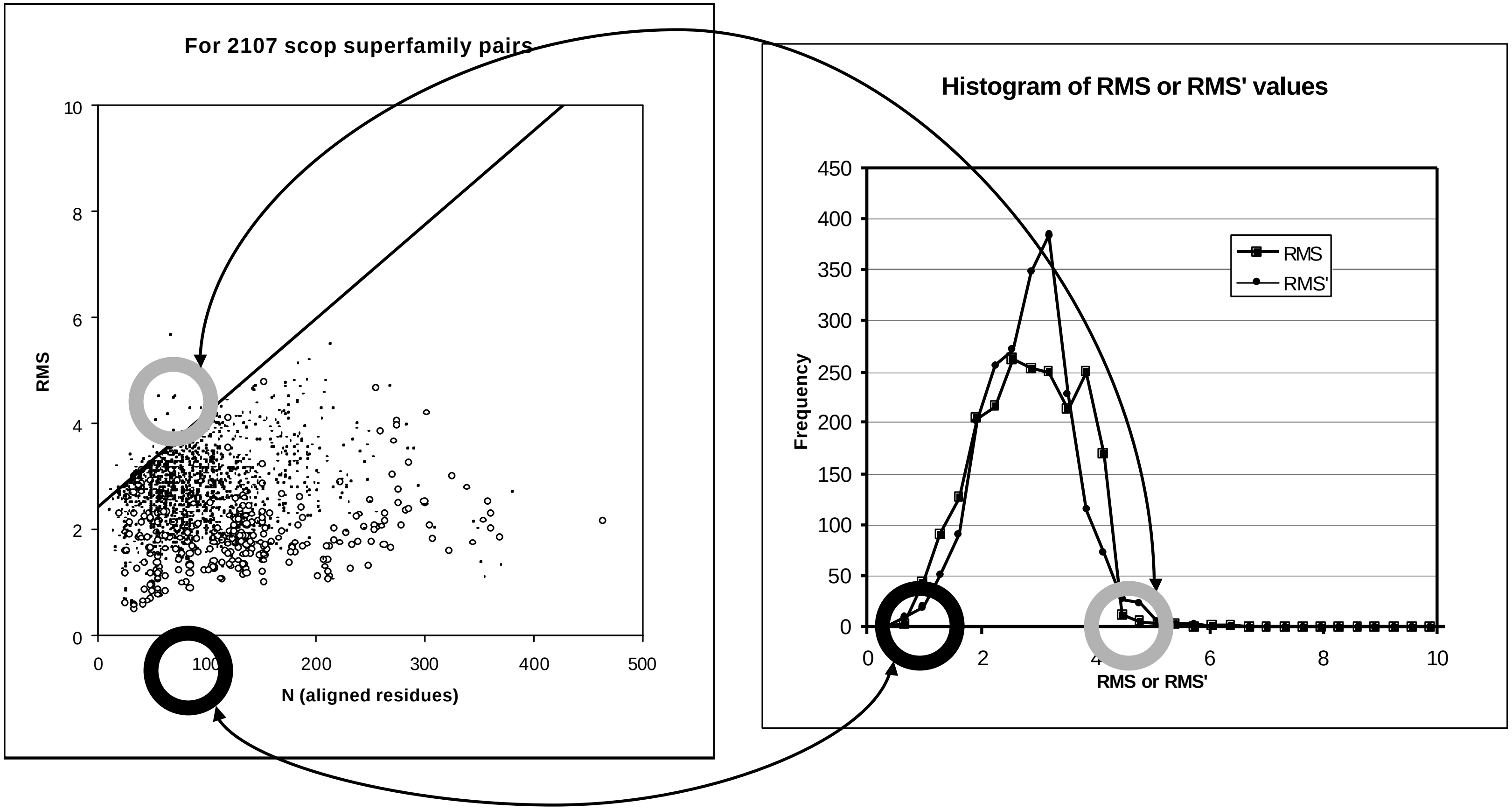
3) Fit a function ρ(S) to TN distribution (TNs from scop); Integrating ρ gives P(s>S), the CDF, chance of getting a score better than threshold S randomly

4) Use same formalism for sequence & structure

Statistics on Range of Similarities

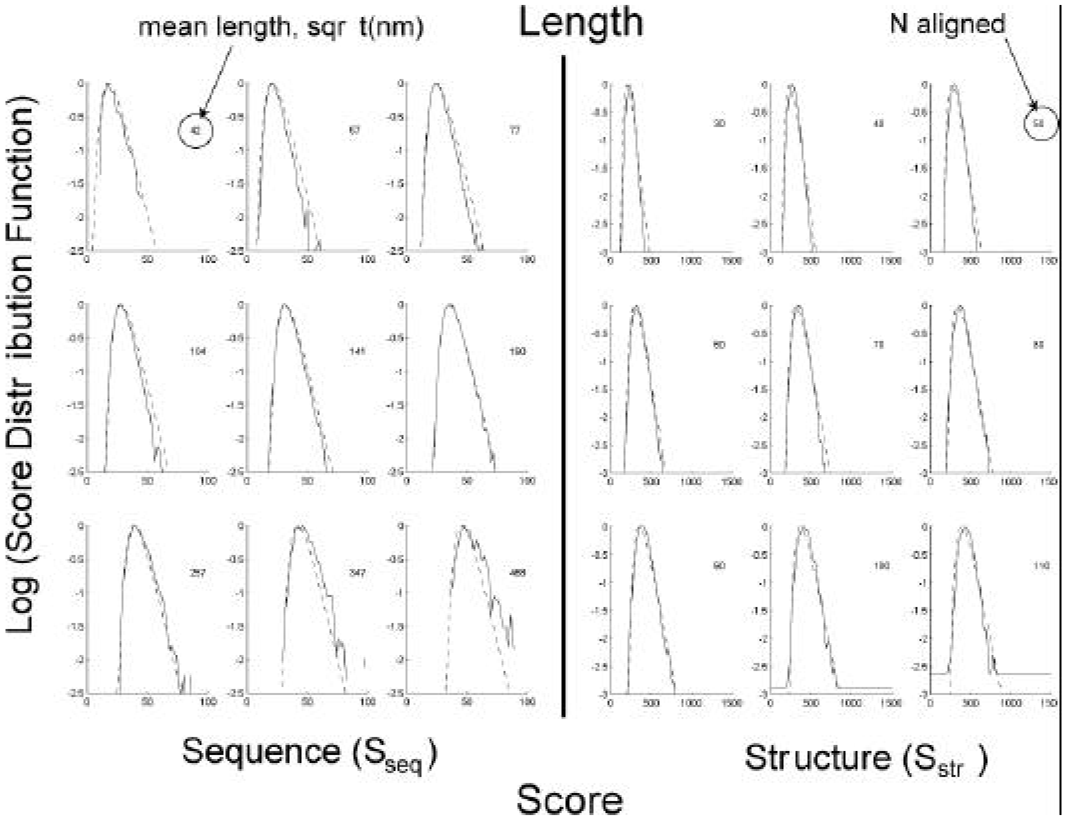
For 2107 pairs, only 2% Outliers (with subtle similarity)

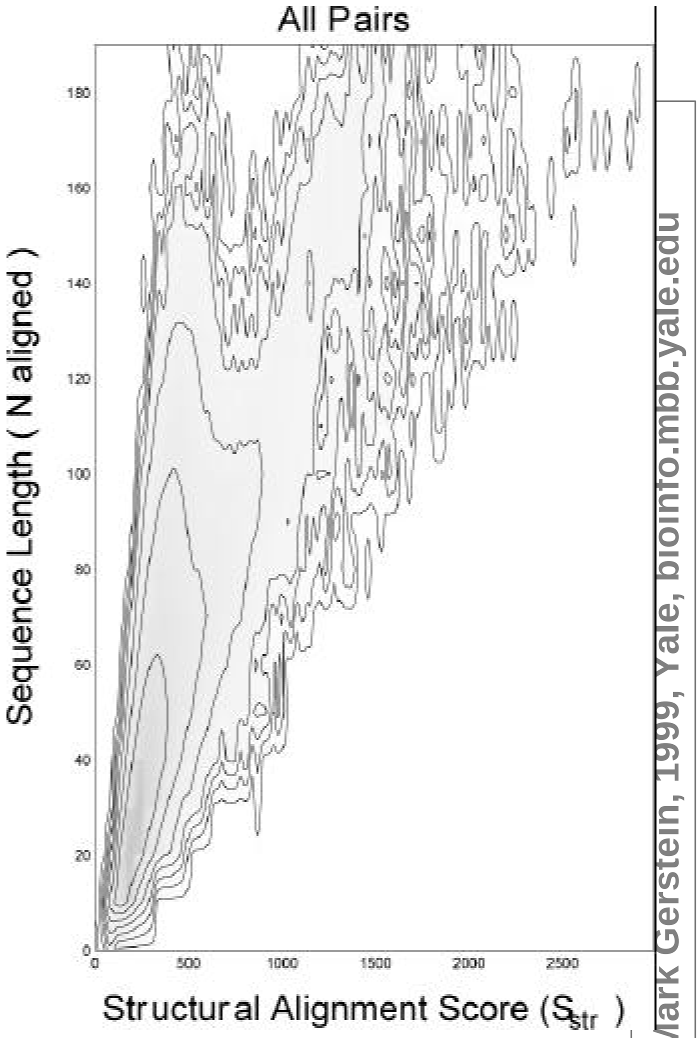
RMS



Num. Aligned

Scores from Structural Alignment Distributed Just Like Ones from Sequence Alignment (E.V.D.)





Same Results for Sequence & Structure

3 Free Parm. fit to EVD involving: ***a, b,*** s . These are the only difference betw. sequence and structure.

$$Z = \frac{S - \left(a \ln N + b\right)}{s}$$

$$S = \sum_{i,j} M(i, j) - G$$

$$\boxed{r(z) = \exp\left(-z - e^{-z}\right)}$$

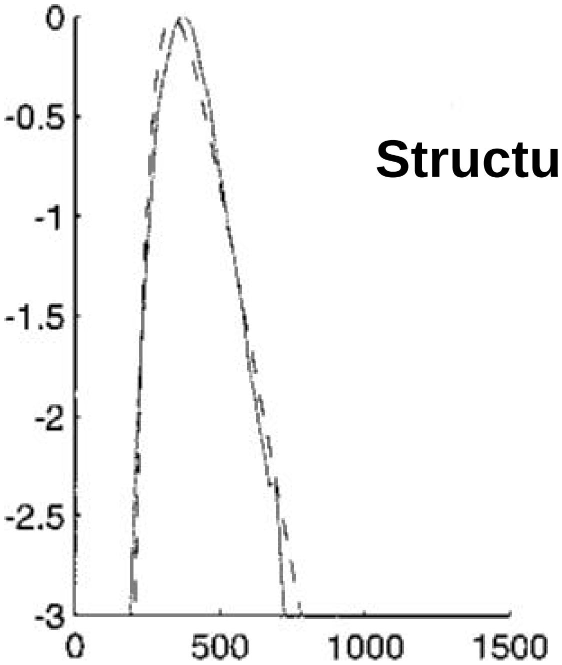
N, G, M also defined differently for sequence and structure.

N = number of residues matched.

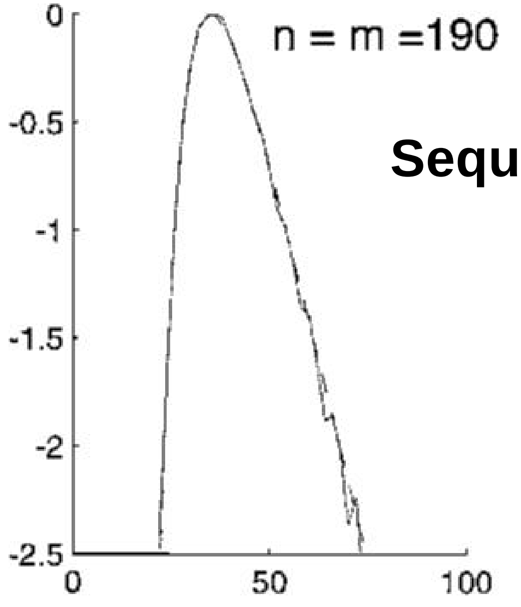
G = total gap penalty.

M(i,j) = similarity matrix

(Blossum for seq. or $\mathrm{M}_{\mathrm{str}}(i,j)_{,}$ struc.)



Structure



Sequence

Score Significance (P-value) derived from Extreme Value Distribution (just like BLAST, FASTA)

F(s) = E.V.D of scores

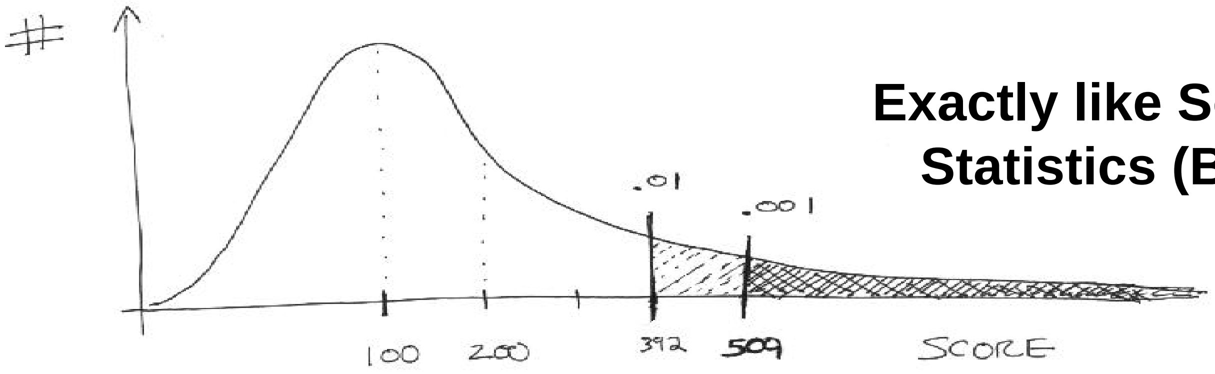
F(s) = exp(-Z(s) - exp(-Z(s)))

Z(s) = As + ln(N) + B

s = Score from random alignment

N length of sequence matched

A & B are fit parameters



P(s>S) = CDF = integral[F(s)]

P(s>S) = 1 - exp(-exp(-Z(s)))

Given Score S (1%), P (s > S) is the chance that a given random score **s** is greater than the threshold

i.e. P-value gives chance score would occur randomly

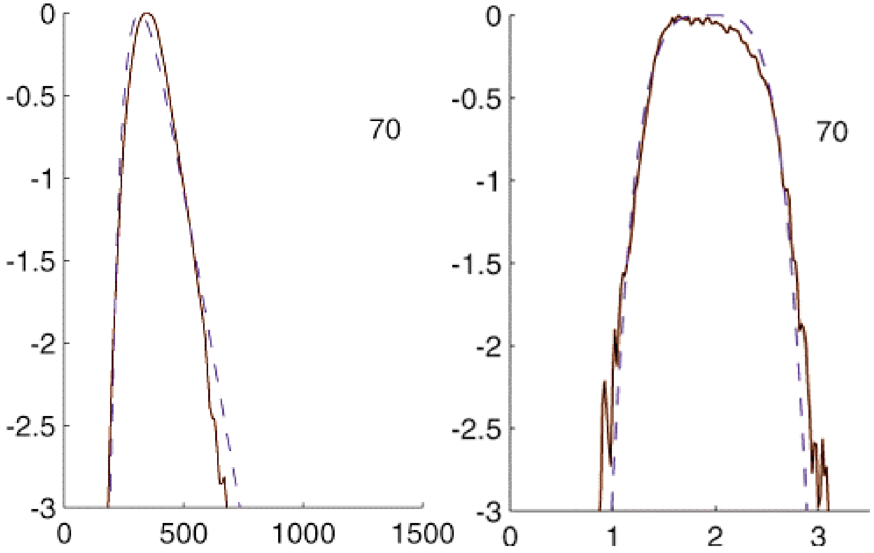
Exactly like Sequence Matching Statistics (BLAST and FASTA)

RMS is a similarity Score

- Also, RMS doesn’t work instead of structural alignment (no EVD fit)
  - ◊ RMS penalizes worst fitting atoms, easily skewed

S_{str} RMS

$$\sum \frac{100}{5+\mathbf{d}_i^2} \; vs \; \sqrt{\sum \mathbf{d}_i^2}$$



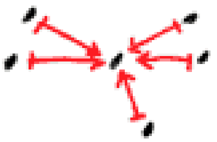
Structure alignment - Other methods

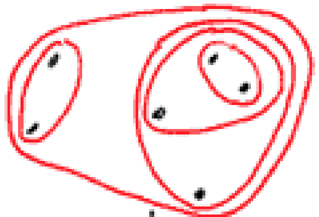
- What Structures Look Like?
- Structural Alignment by Iterated Dynamic Programming
  - ◊ RMS Superposition
- Scoring Structural Similarity
- Other Aspects of Structural Alignment
  - ◊ Distance Matrix based methods
  - ◊ Fold Library
- Relation of Sequence Similarity to Structural and Functional Similarity

- Protein Geometry
- Surface I (Calculation)
- Calculation of Volume
- Voronoi Volumes & Packing
- Standard Volumes & Radii
- Surfaces II (Relationship to Volumes)
- Other Applications of Volumes -- Motions, Docking

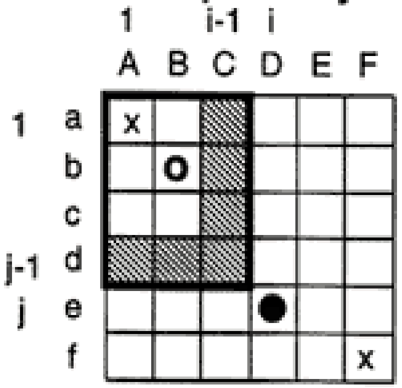
Refine Method

- Multiple Aligment by aligning to central structure





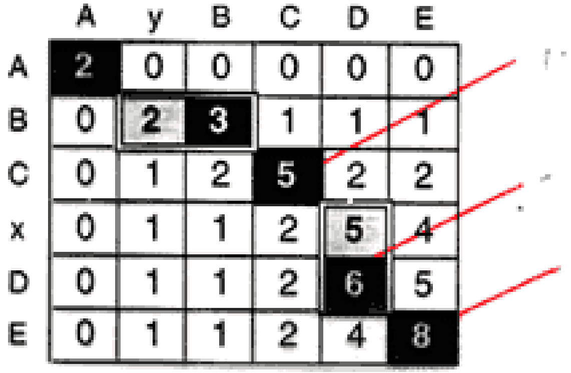
- More Complex Dynamic Programming



AB-C-DEF
abc-de-f

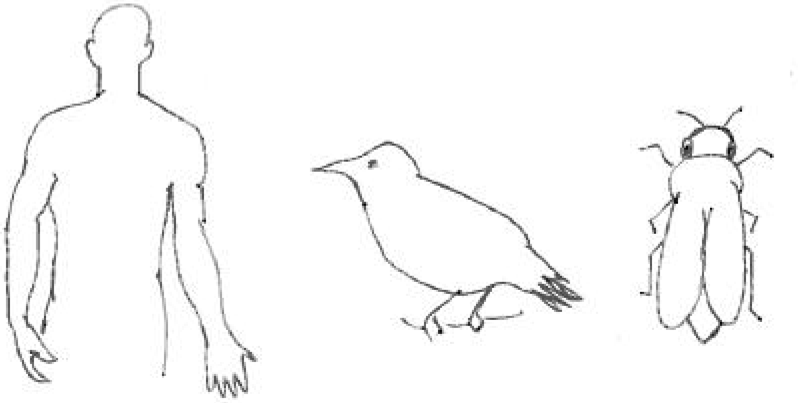
n^2 vs. n^4

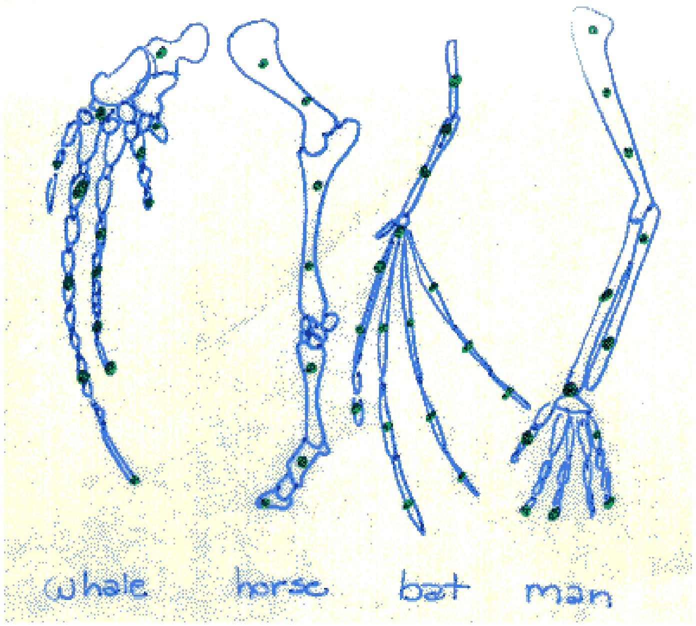
- Find "best" aligned regions
  - "Core-finding" to remove outliers
  - "Noisy" suboptimal paths

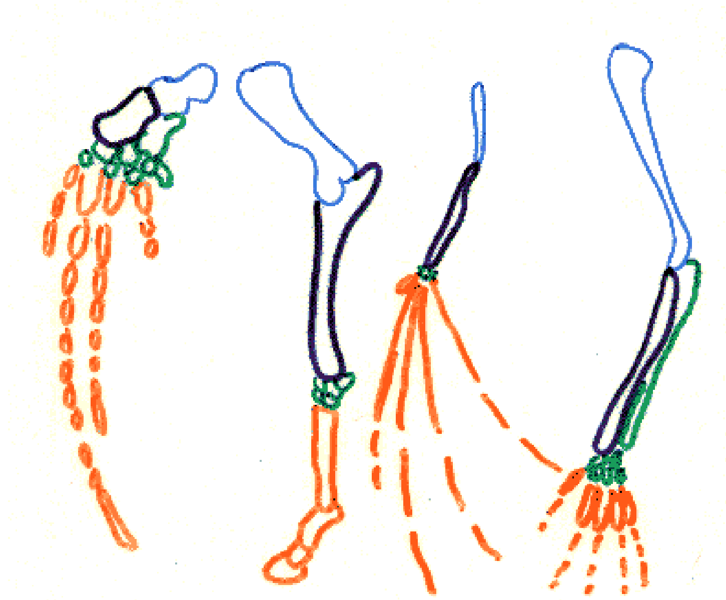


	A	y	B	C	D	E
A	2	0	0	0	0	0
B	0	2	3	1	1	1
C	0	1	2	5	2	2
x	0	1	1	2	5	4
D	0	1	1	2	6	5
E	0	1	1	2	4	8

Significance Ignoring Crucial Features in Structural Similarity

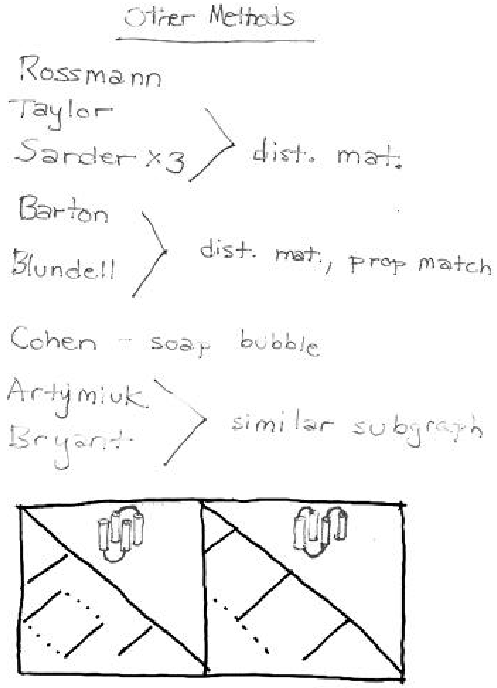






Other Methods of Structural Alignment

- RMS fitting used universally, but other alignment methods
- Comparison of Distance Matrices
  - ◊ Holm & Sander, DALI
  - ◊ Taylor & Orengo



Other Methods

Rossmann
Taylor
Sander x3 } dist. mat.
Barton
Blundell } dist. mat., prop match
Cohen - soap bubble
Artymiuk
Bryant } similar subgraph

Structure Hashing
    Bryant, VAST
    Rice, Artymiuk
Others
    Cohen (Soap)
    Sippl
    Godzik (Lattice)