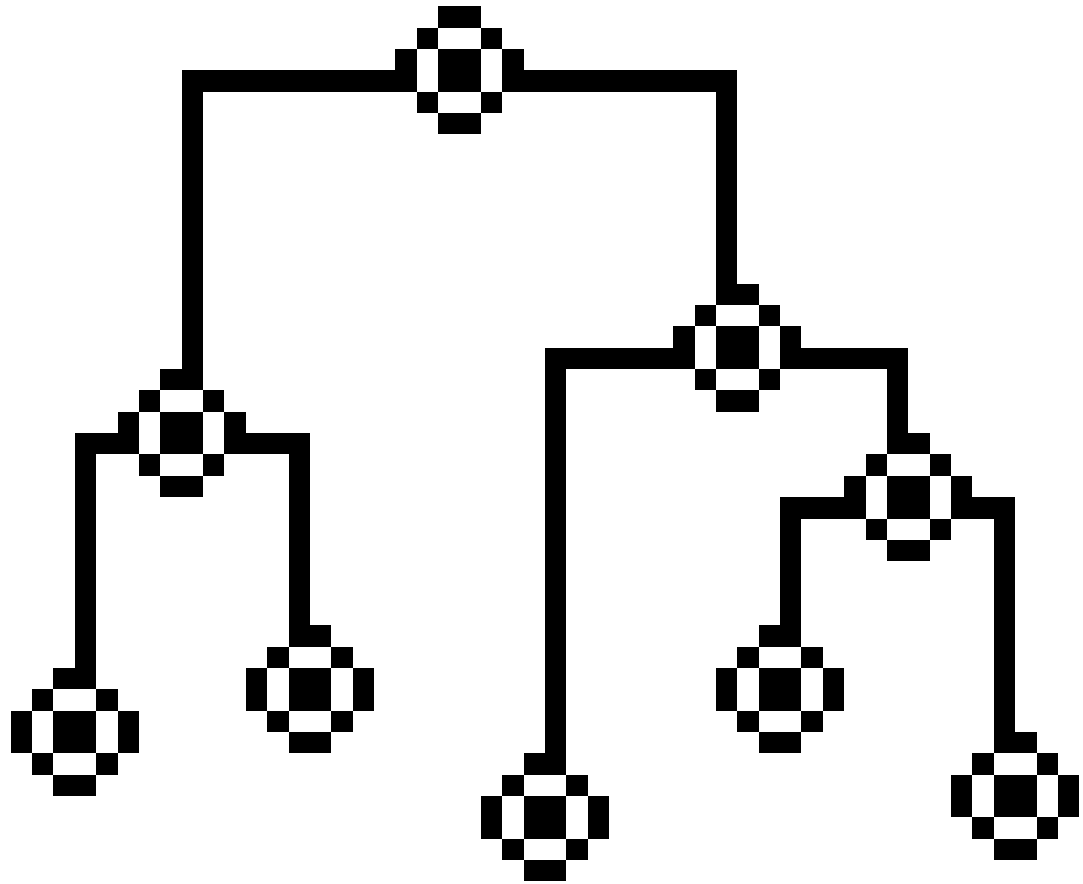
Clustering methods for Bioinformatics



Marc A. Marti-Renom
Adjunct Assistant Professor
http://salilab.org/~marcius

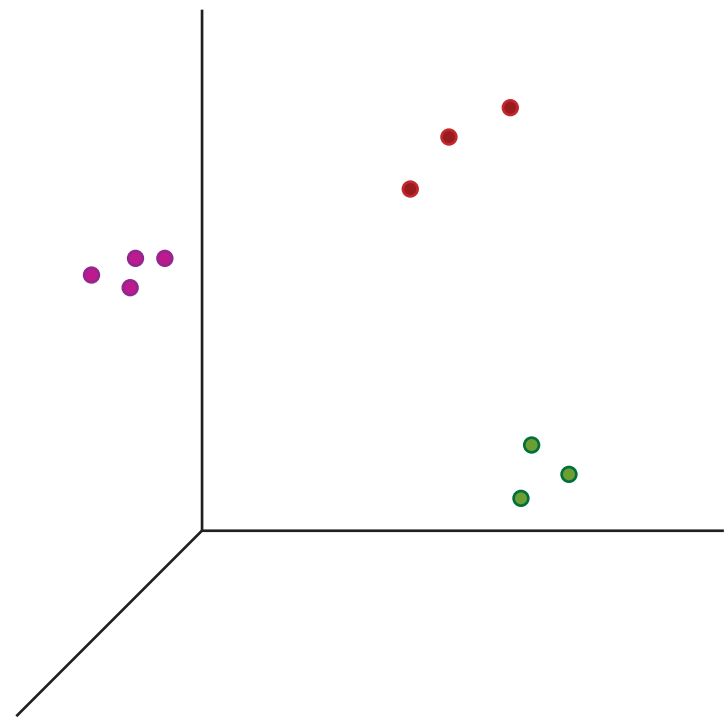
UCSF
University of California
San Francisco

Depts. of Biopharmaceutical Sciences and Pharmaceutical Chemistry
California Institute for Quantitative Biomedical Research
University of California at San Francisco

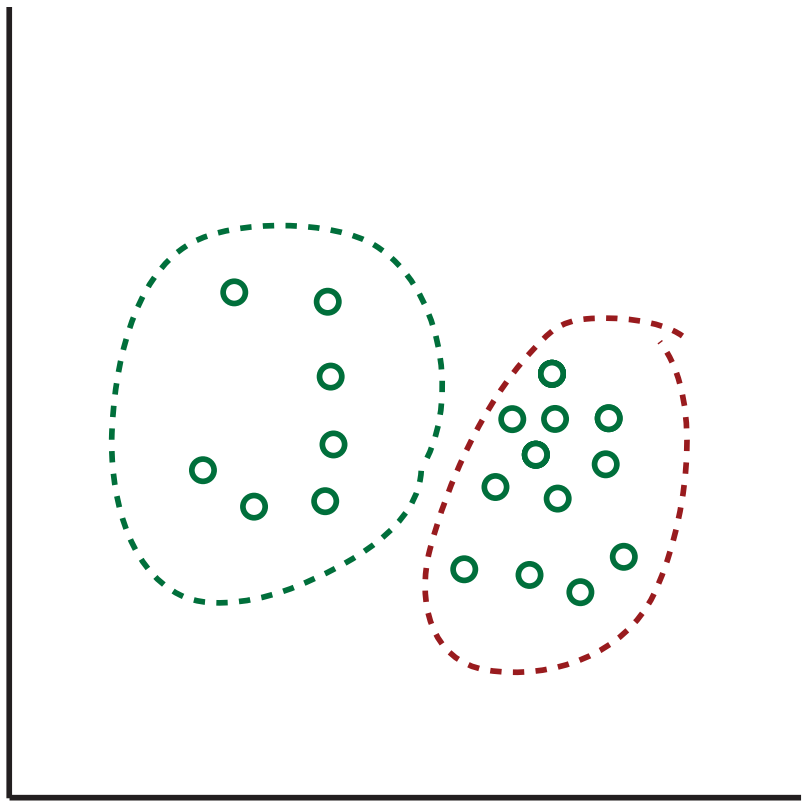
The Problem

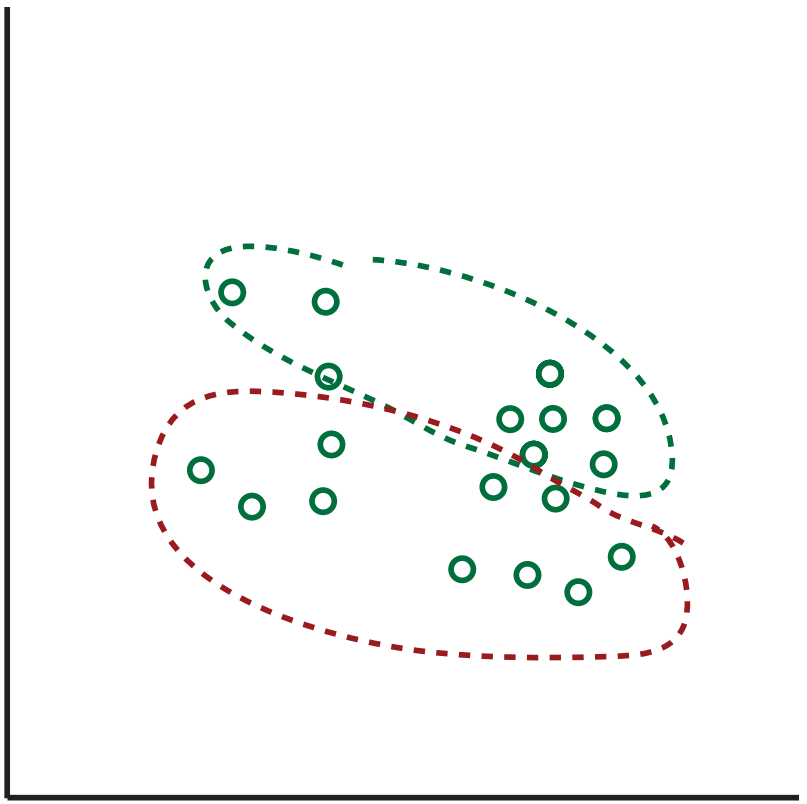
	x	y	z
a	10	8	10
b	10	0	9
c	4	8.6	3
d	7	8	3
e	1	2	3

.
.
.



Good and Bad?



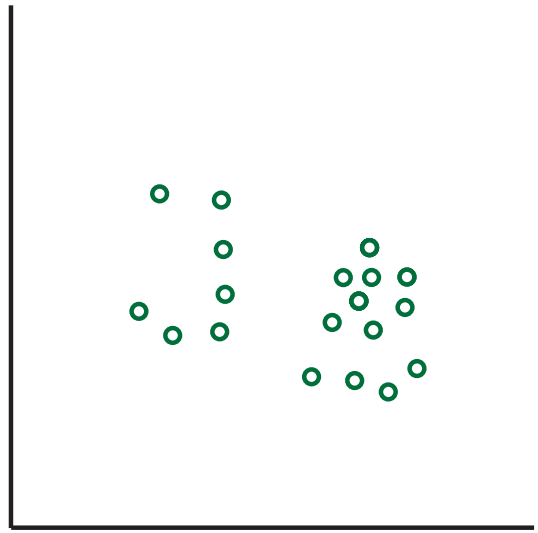


Homogeneity and Separation Principles

- **Homogeneity:** Elements within a cluster are close to each other
- **Separation:** Elements in different clusters are further apart from each other

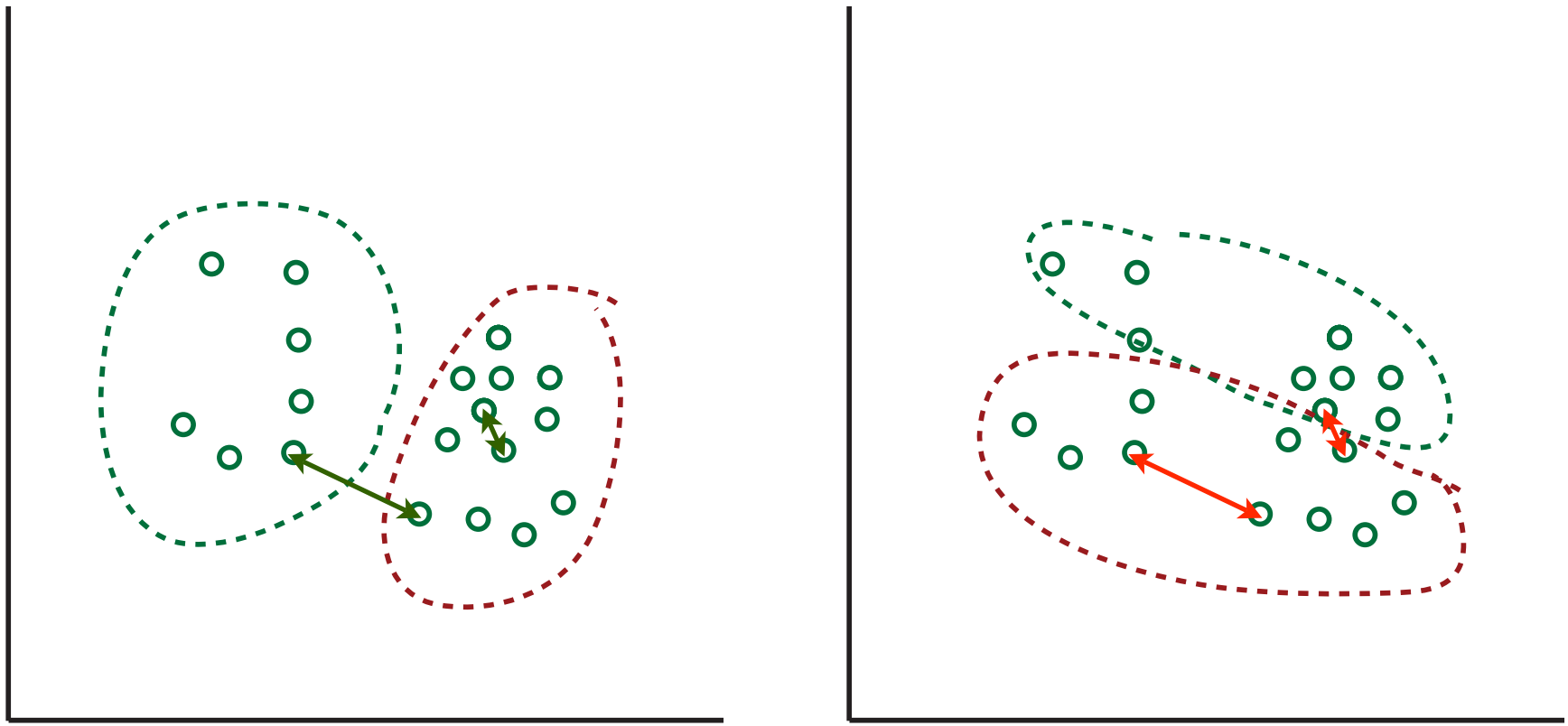
…clustering is not an easy task!

Given these points a clustering algorithm might make two distinct clusters as follows



Good

Bad



Homogeneity & Separation

Example

microarray data

- Microarray data are usually transformed into an intensity matrix (below)
- The intensity matrix allows to make correlations between different genes and to understand how genes functions might be related

Intensity (expression level) of gene at measured time

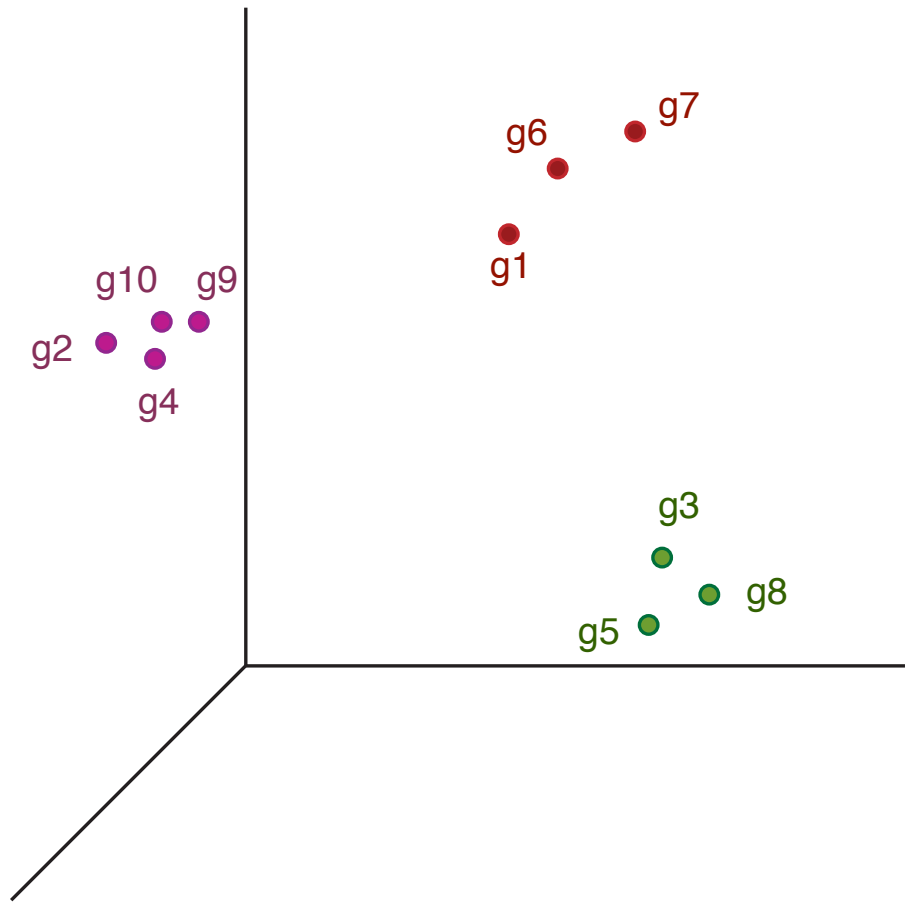
Time:	Time X	Time Y	Time Z
Gene 1	10	8	10
Gene 2	10	0	9
Gene 3	4	8.6	3
Gene 4	7	8	3
Gene 5	1	2	3

Example

microarray data

Time	1 hr	2 hr	3 hr
g_1	10.0	8.0	10.0
g_2	10.0	0.0	9.0
g_3	4.0	8.5	3.0
g_4	9.5	0.5	8.5
g_5	4.5	8.5	2.5
g_6	10.5	9.0	12.0
g_7	5.0	8.5	11.0
g_8	2.7	8.7	2.0
g_9	9.7	2.0	9.0
g_{10}	10.2	1.0	9.2

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0



Clustering techniques

- **Hierarchical**
- **K-Means**
- **Clique Graphs**

- ***Example of graph based clustering***

Hierarchical clustering

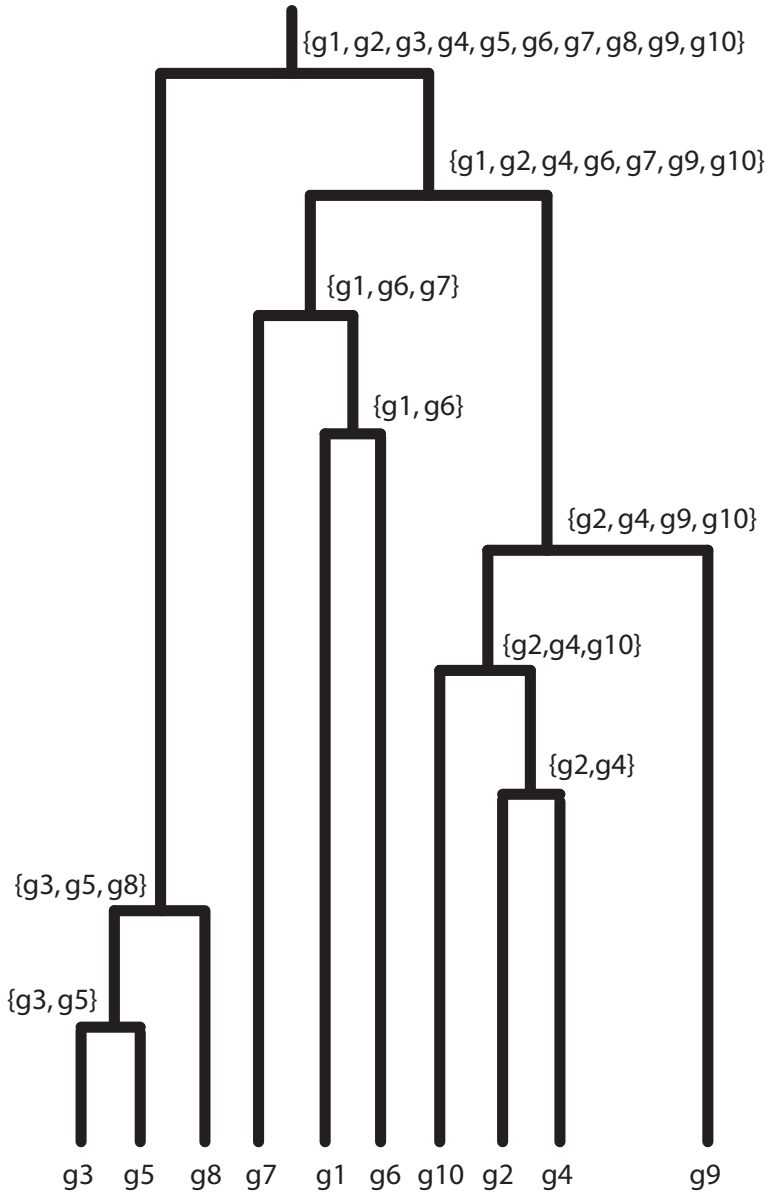
algorithm

1. Form **n** clusters each with one element
2. Construct a graph **T** by assigning one vertex to each cluster
3. while there is more than one cluster
4. Find the two closest clusters C_1 and C_2
5. Merge C_1 and C_2 into new cluster **C** with $|C_1| + |C_2|$ elements
6. Compute distance **d** from **C** to all other clusters
7. Add a new vertex **C** to **T** and connect to vertices C_1 and C_2
8. Remove rows and columns of **d** corresponding to C_1 and C_2
9. Add a row and column to **d** corresponding to the new cluster **C**
10. return **T**

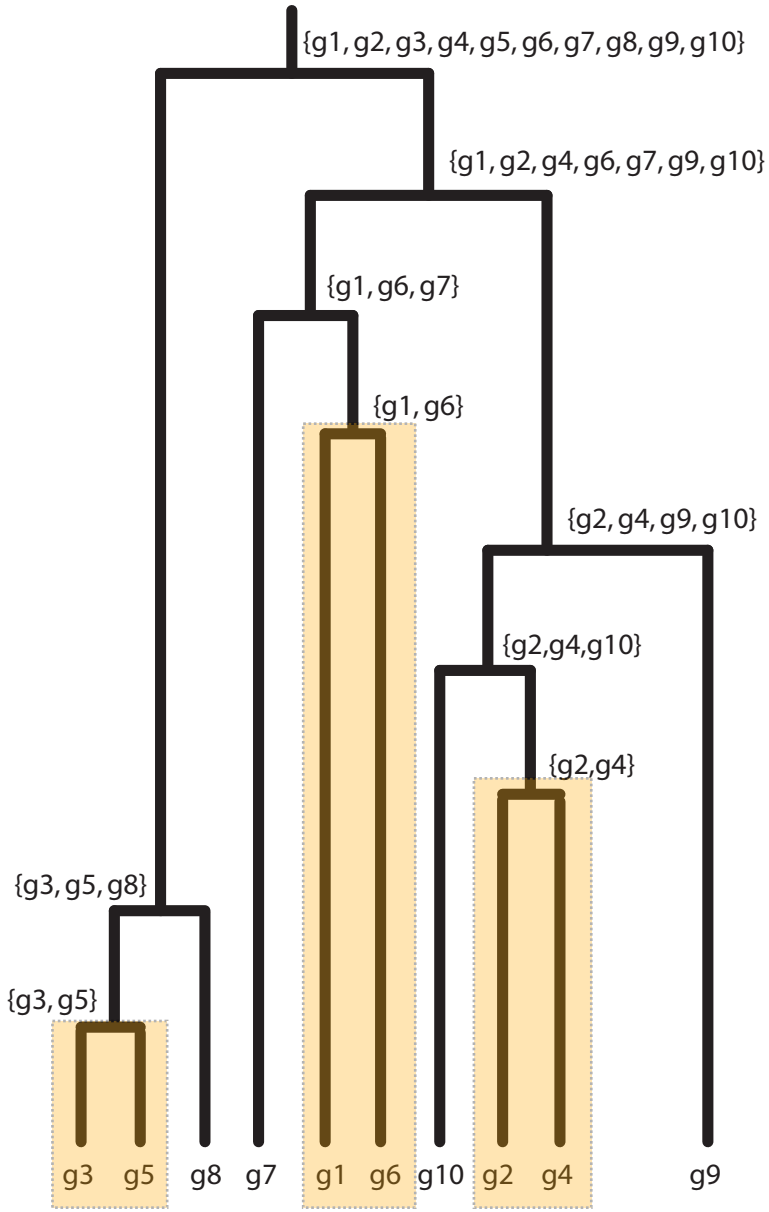
The algorithm takes a *n*x*n* distance matrix ***d*** of pairwise distances between points as an input.

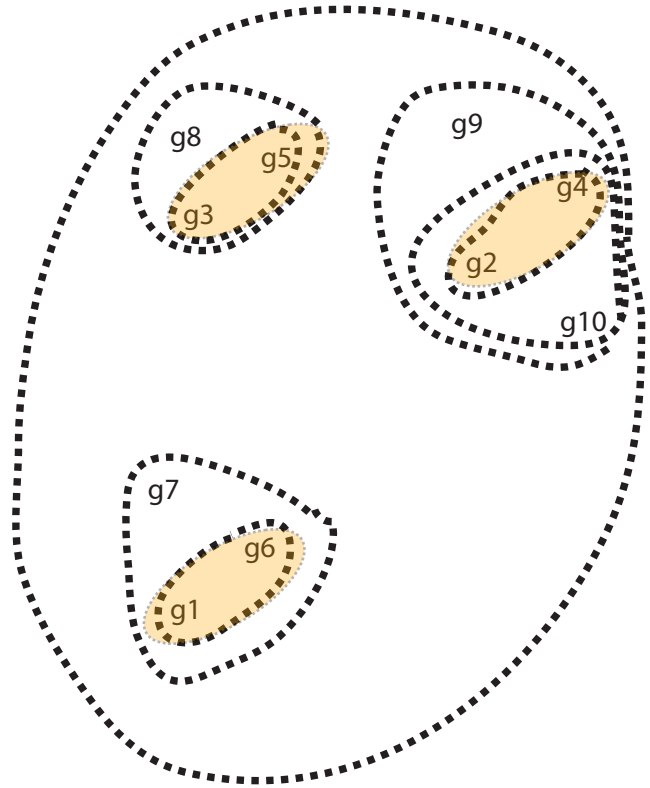
	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

Hierarchical clustering

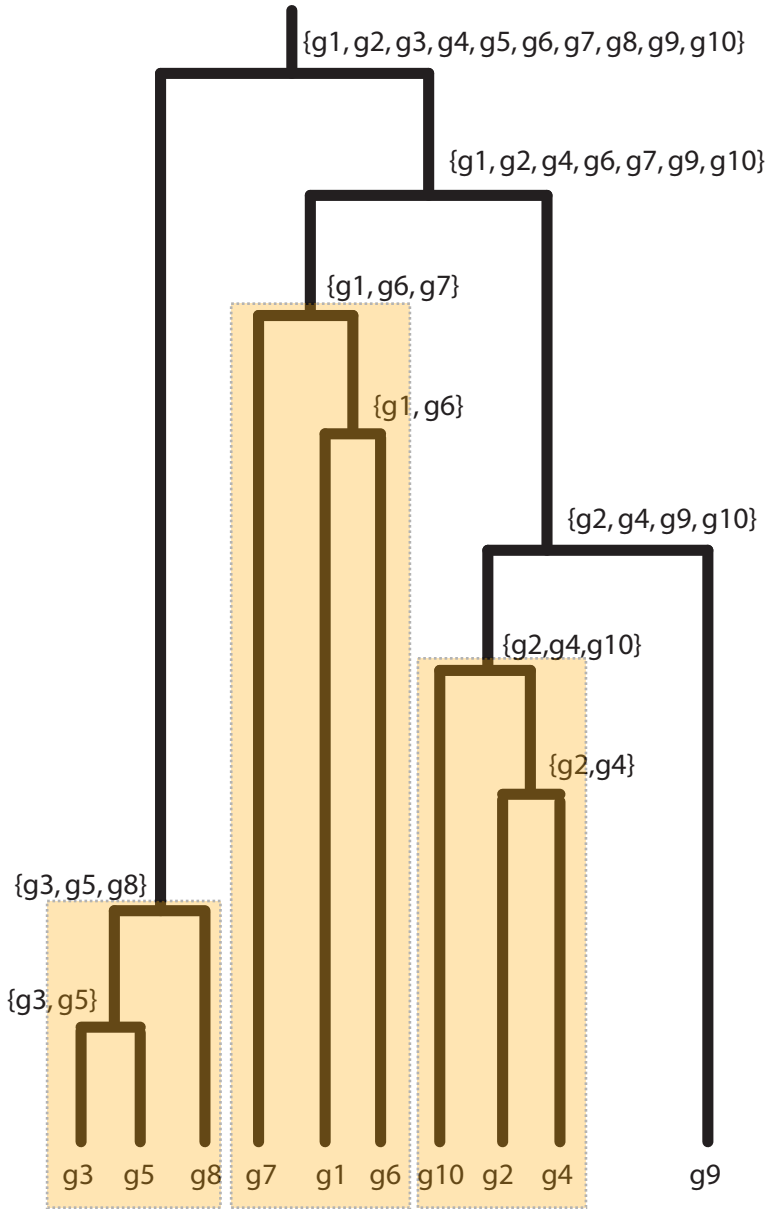


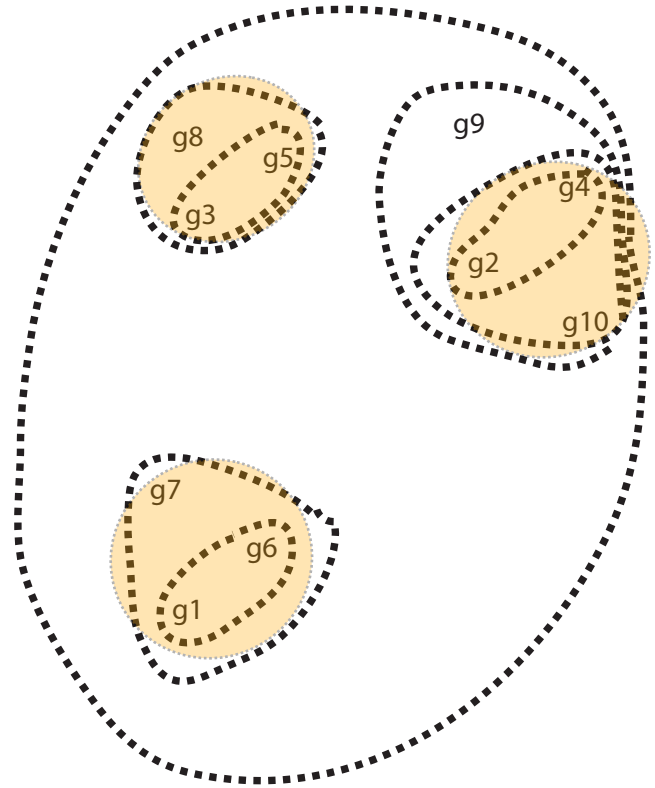
Hierarchical clustering



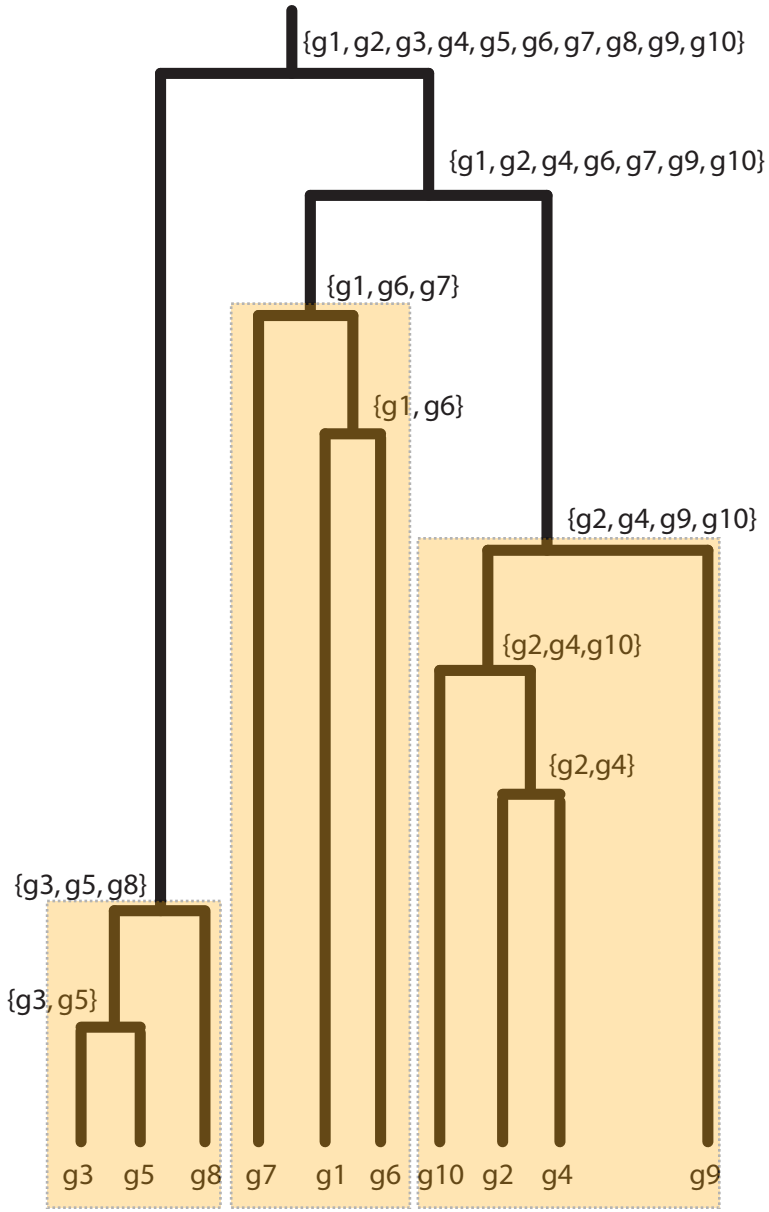


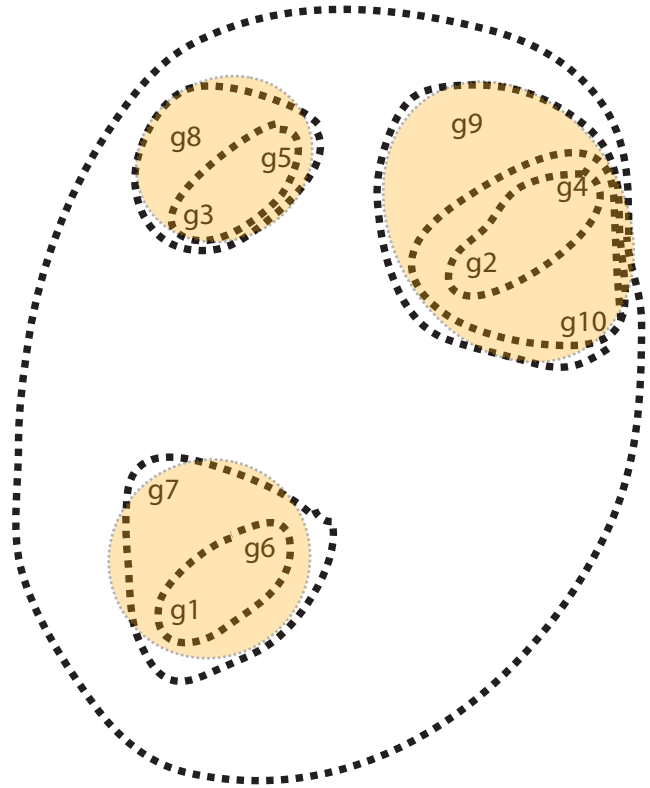
Hierarchical clustering



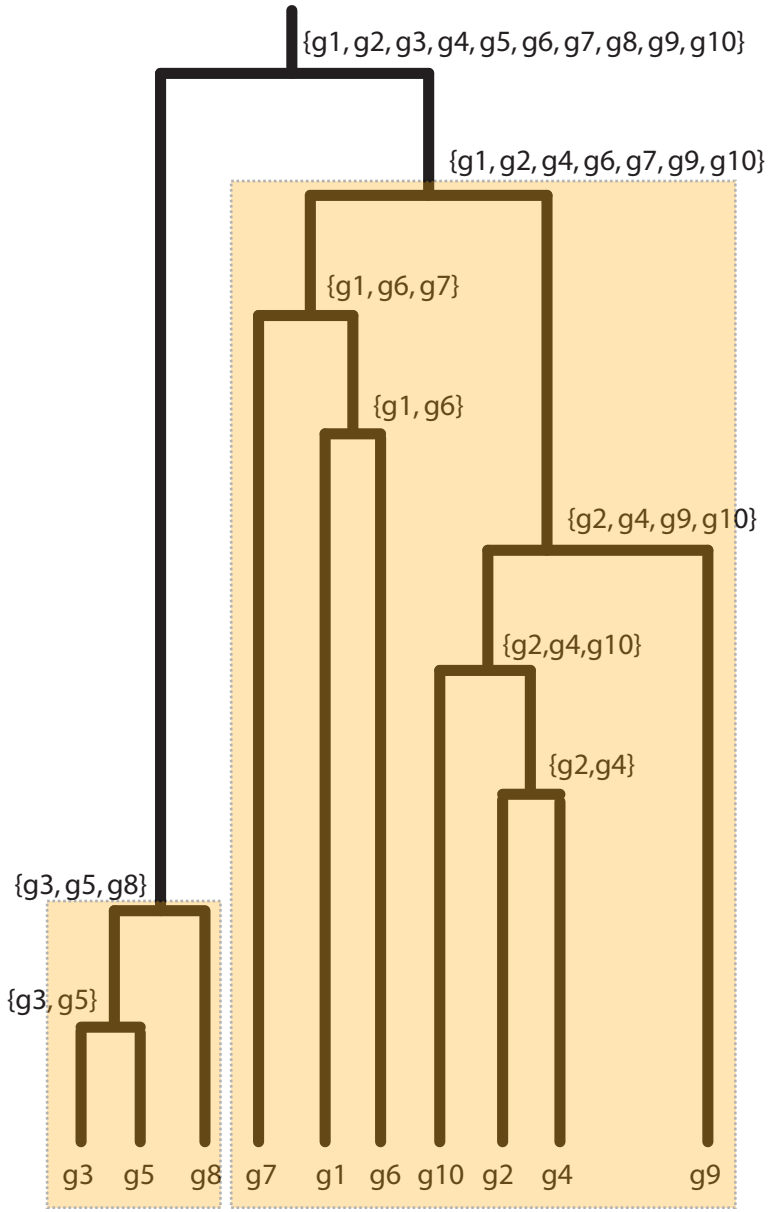


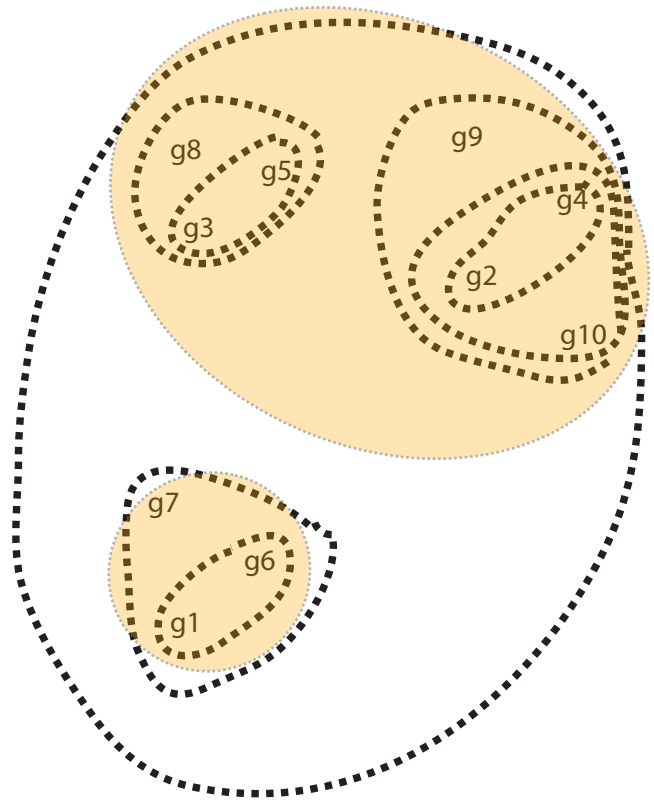
Hierarchical clustering



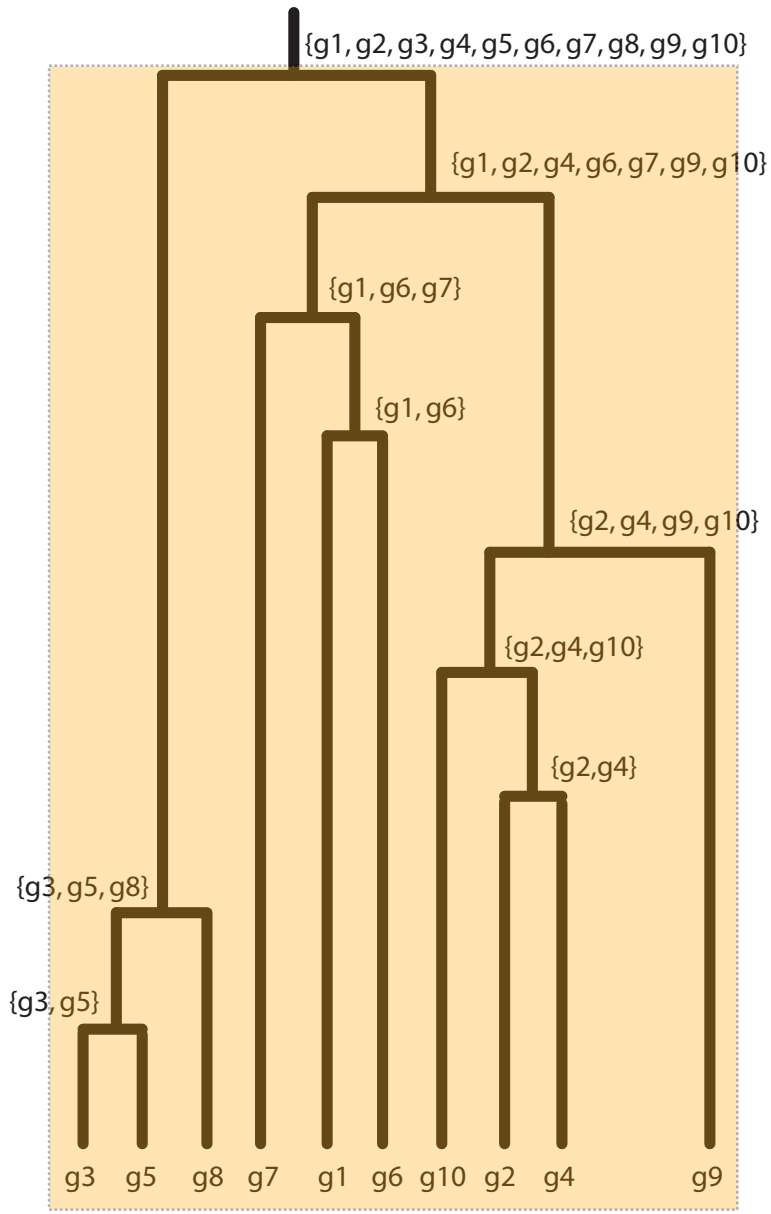


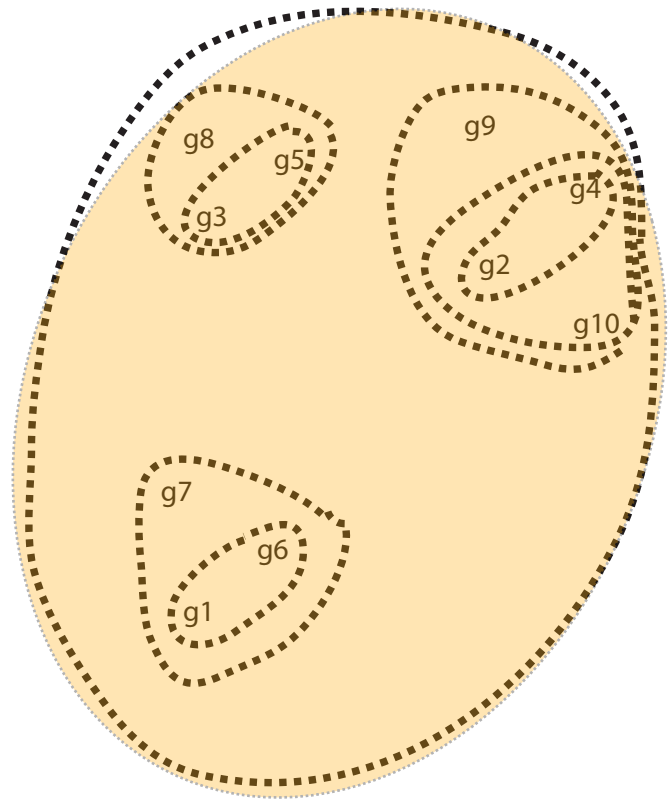
Hierarchical clustering



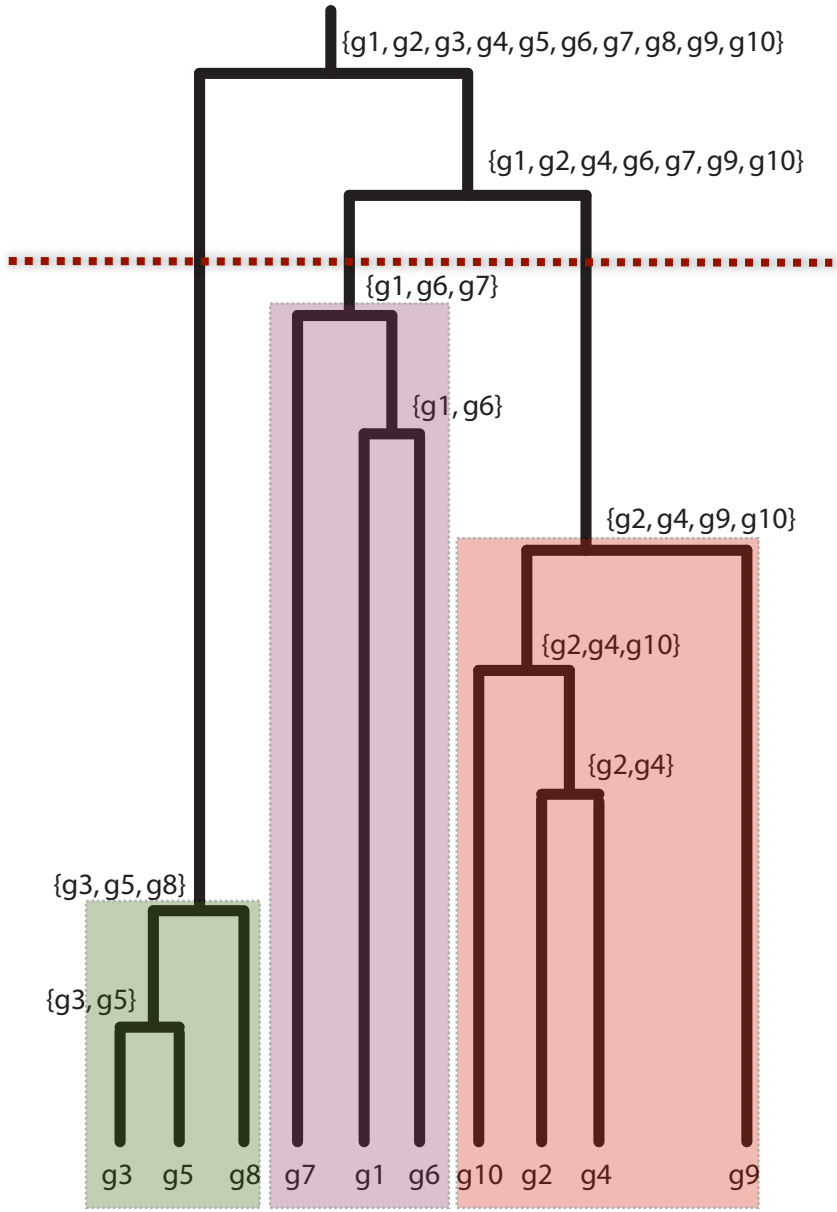


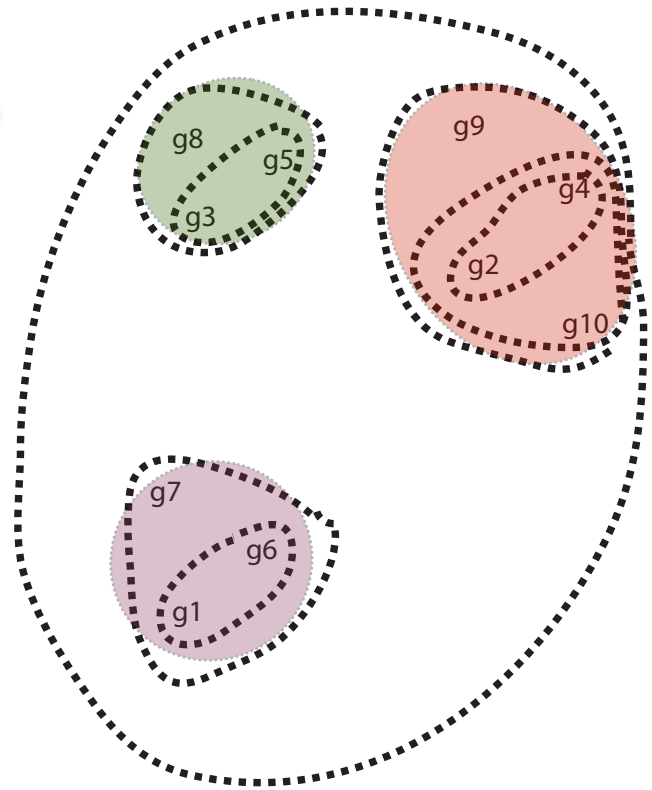
Hierarchical clustering





Hierarchical clustering





Hierarchical clustering

algorithm

1. Form **n** clusters each with one element
2. Construct a graph **T** by assigning one vertex to each cluster
3. while there is more than one cluster
4. Find the two closest clusters C_1 and C_2
5. Merge C_1 and C_2 into new cluster **C** with $|C_1| + |C_2|$ elements
6. Compute distance **d** from **C** to all other clusters
7. Add a new vertex **C** to **T** and connect to vertices C_1 and C_2
8. Remove rows and columns of **d** corresponding to C_1 and C_2
9. Add a row and column to **d** corresponding to the new cluster **C**
10. return **T**

Different ways to define distances between clusters may lead to different clusterings

Hierarchical clustering

algorithm

Distance between two clusters is the **smallest** distance between any pair of their elements

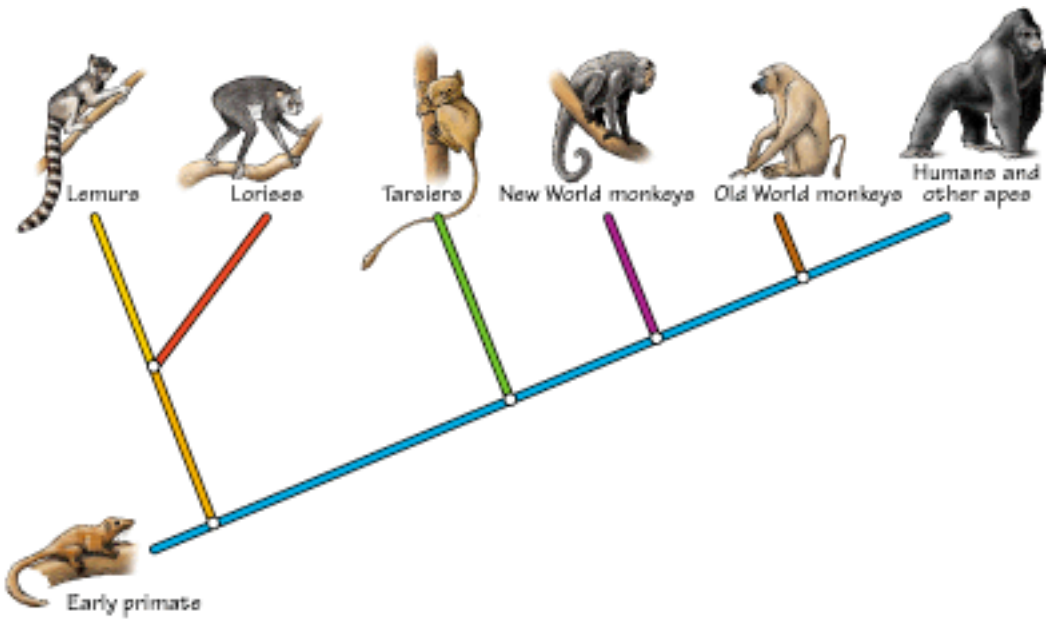
$$d_{\min}(C^*, C) = \min_{x \in C^*, y \in C} d(x, y)$$

Distance between two clusters is the **average** distance between all pairs of their elements

$$d_{avg}(C^*, C) = \frac{1}{|C^*||C|} \sum_{x \in C^*, y \in C} d(x, y)$$

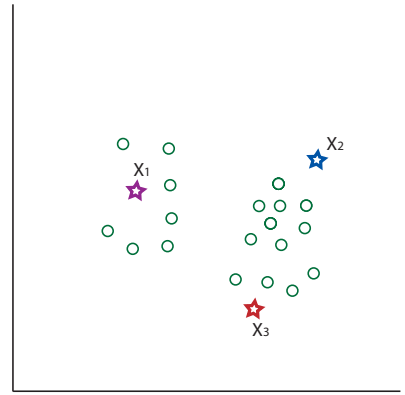
Hierarchical clustering

Hierarchical Clustering is often used to reveal evolutionary history



K-Means clustering

algorithm



- **Input**: A set, ***V***, consisting of *n* points
- **Output**: A **K** points ***x*** (cluster center) that minimizes the squared error distortion *d(**V**,**x**)* over all possible choices of ***x***

1-Means Clustering problem is easy (K=1).

However, it becomes very difficult (NP-complete) for more than one center.

An efficient ***heuristic*** method for K-Means clustering is the Lloyd algorithm

K-Means clustering

squared error distortion

- **cost(P)** = squared error distortion
- **cost(P)** = 0 for a clustering with all **k** points in the center of mass of each cluster.

$$d(V,X) = \frac{\sum_{i=1}^{n} d(v_i, X)^2}{n}$$

K-Means clustering

Lloyd algorithm

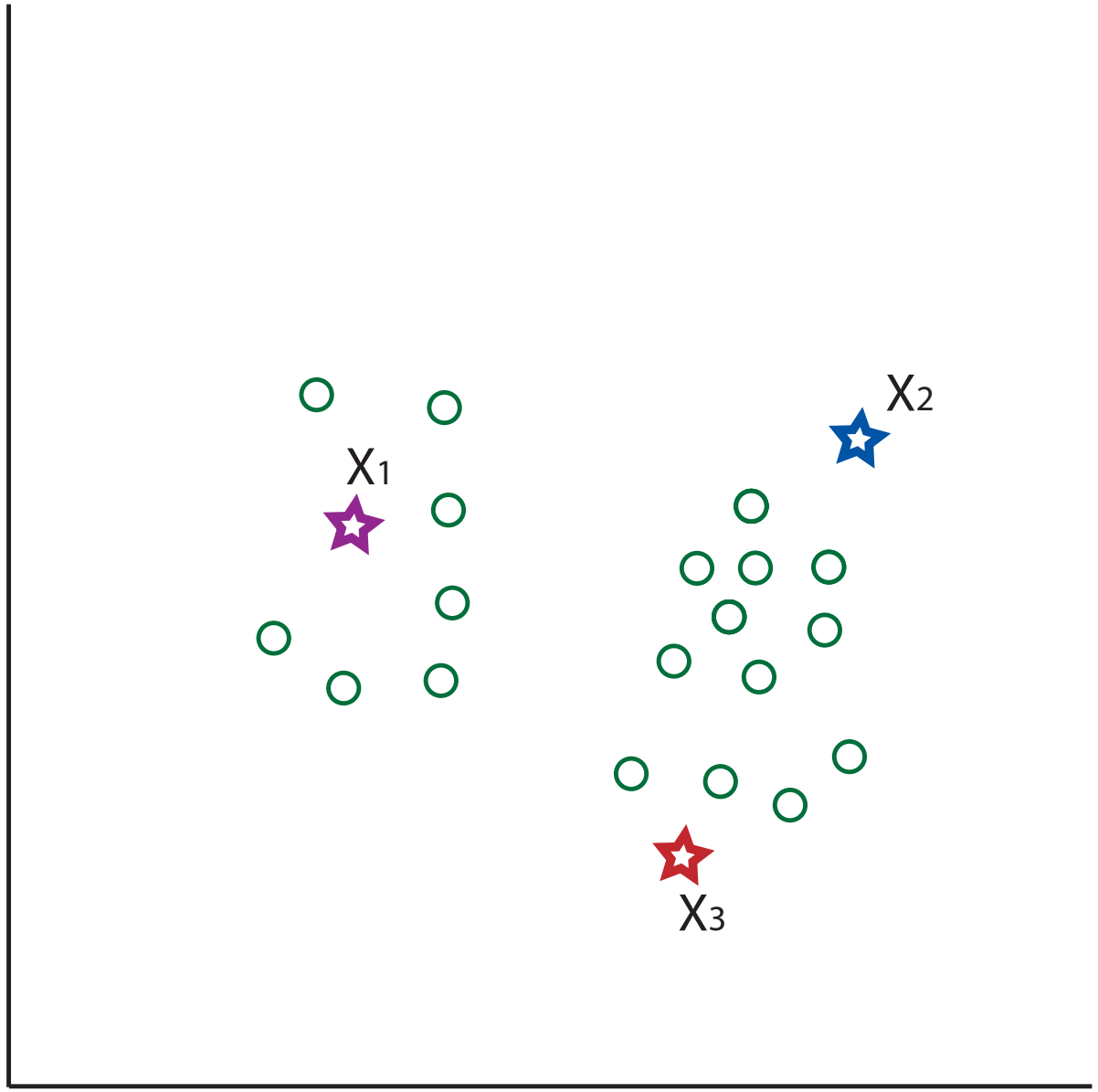
1. Arbitrarily assign the **k** cluster centers
2. While the cluster **centers** keep changing

3. Assign each data point to the cluster $\mathbf{C_i}$ corresponding to the closest cluster representative (center) **(1 ≤ i ≤ k)**

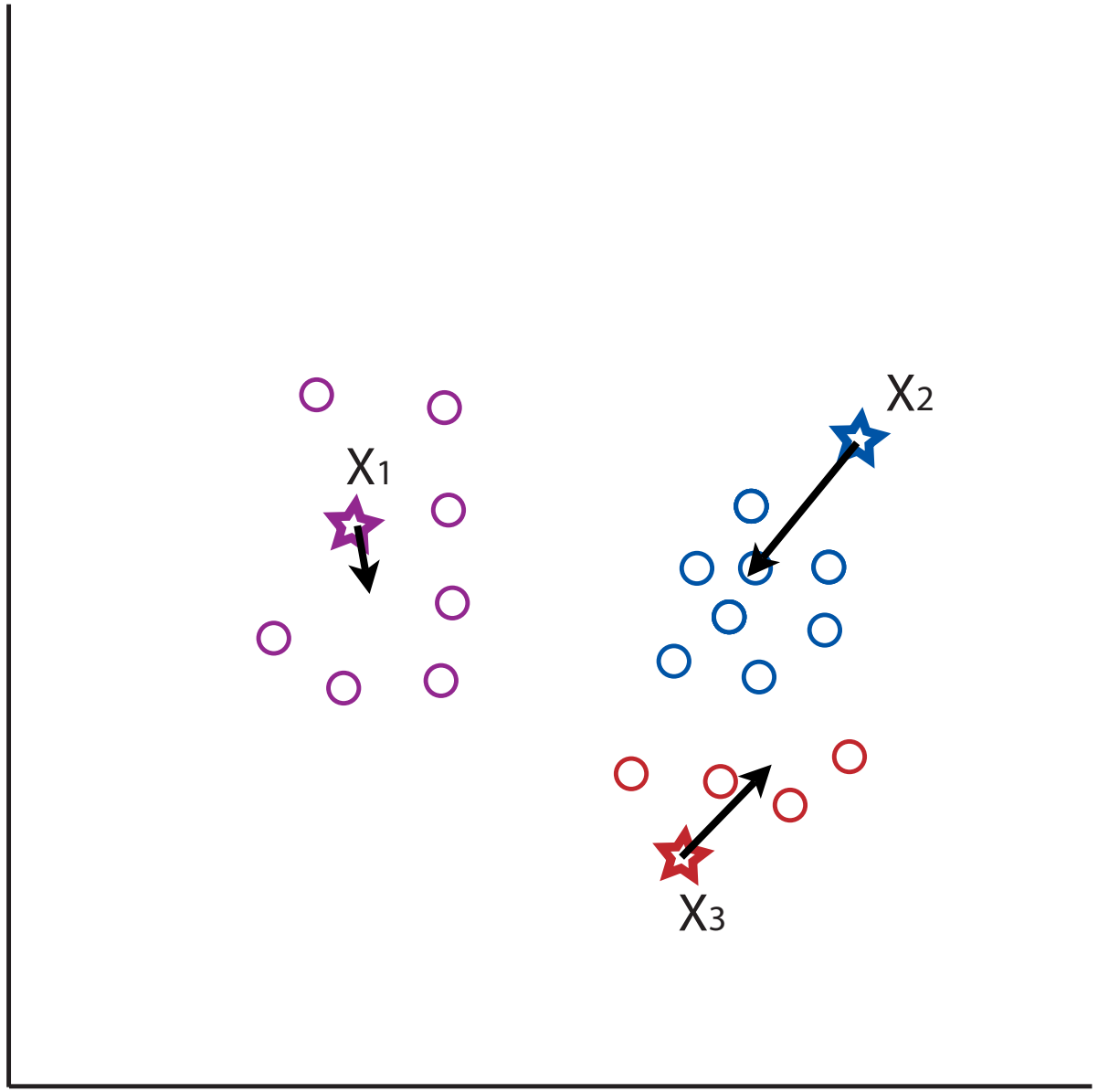
4. After the assignment of **all** data points, compute new cluster **representatives** according to the **center of gravity** of each cluster, that is, the new cluster representative is $\frac{\sum_{v \in C} v}{|C|}$ for all **v** in **C** for every cluster **C**

***This may lead to merely a locally optimal clustering and leads to large rearrangements of nodes**

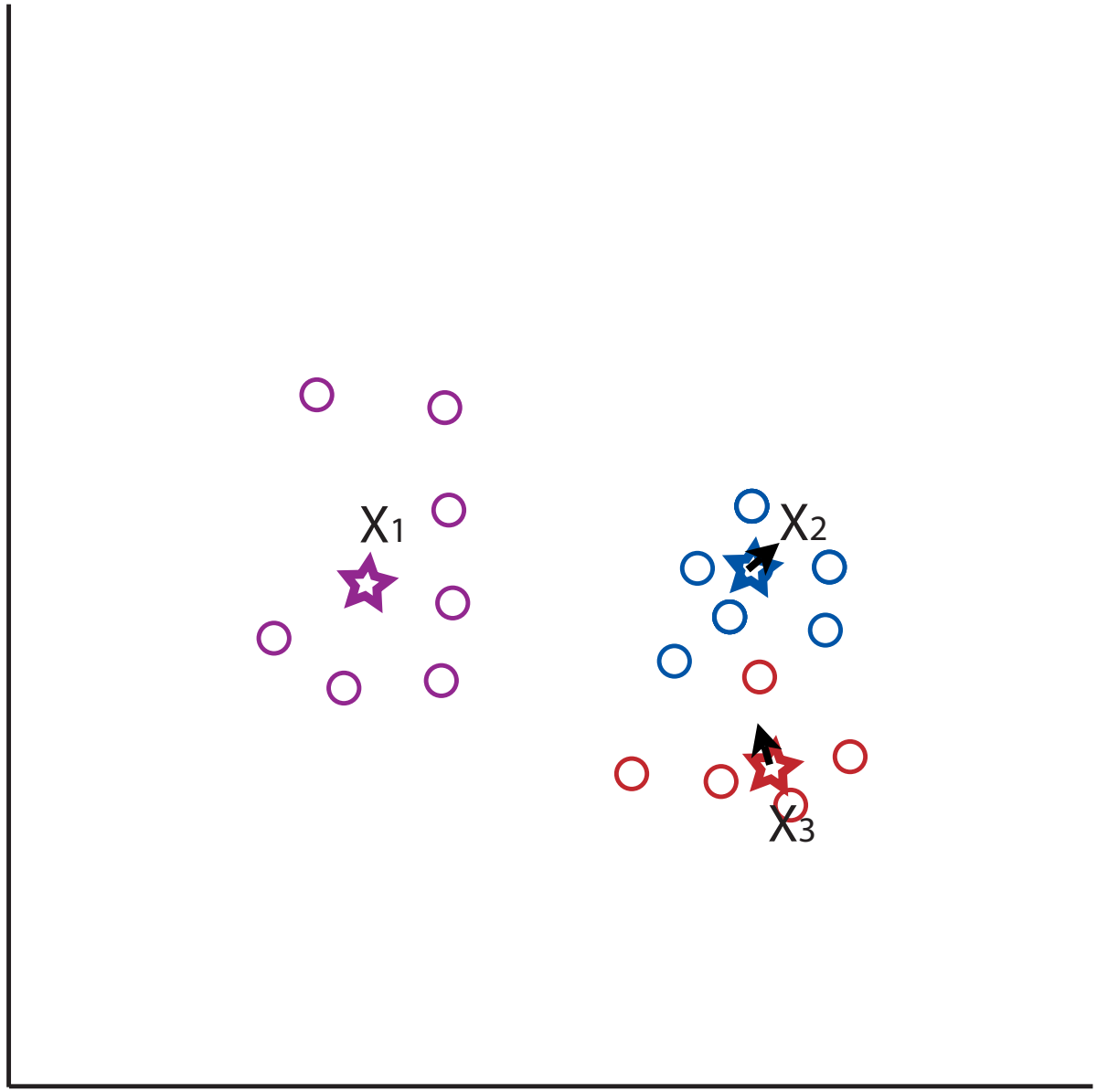
K-Means clustering



K-Means clustering

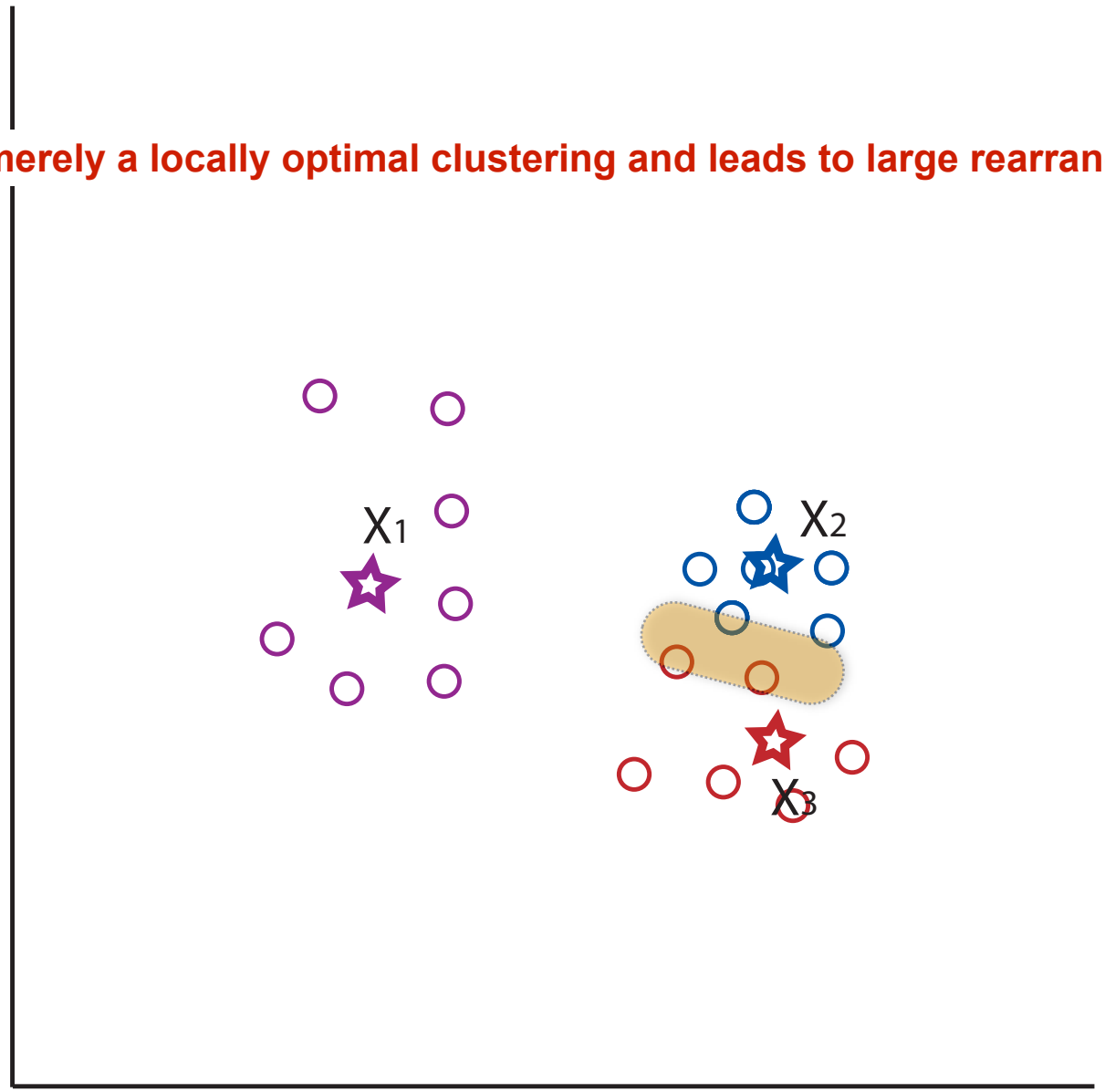


K-Means clustering



K-Means clustering

This may lead to merely a locally optimal clustering and leads to large rearrangements of nodes



K-Means clustering

clustering cost

- The clustering cost measures the quality of a particular partition **P**.
- **cost(P)** = squared error distortion
- **cost(P)** = 0 for a clustering with all **k** points in the center of mass of each cluster.

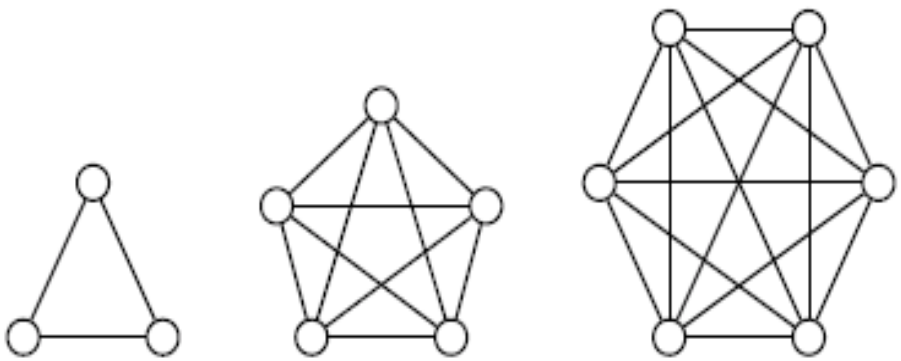
K-Means clustering

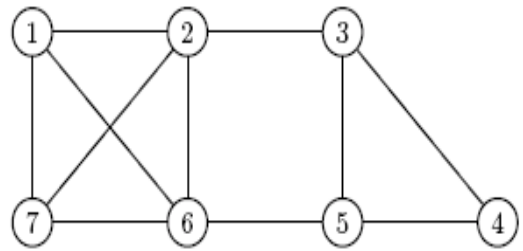
"Greedy" algorithm

1. Select an arbitrary partition **P** into **k** clusters
2. while forever
3. **bestChange** ← 0
4. for every cluster **C**
5. for every element **i** not in **C**
6. if moving **i** to cluster **C** reduces its clustering cost
7. if (cost(**P**) – cost($\mathbf{P}_{i \to C}$) > **bestChange**
8. **bestChange** ← cost(**P**) – cost($\mathbf{P}_{i \to C}$)
9. $\mathbf{i}^*$ ← **i**
10. $\mathbf{C}^*$ ← **C**
11. if **bestChange** > 0
12. Change partition **P** by moving $\mathbf{i}^*$ to $\mathbf{C}^*$
13. else
14. return P

Clique graph clustering

- A **clique** is a graph with every vertex connected to every other vertex
- A **clique graph** is a graph where each connected component is a clique





{1,2,6,7} {2,3} {5,6} {3,4,5}

Clique graph clustering

matrix to graph

- Turn the distance matrix into a distance graph
  - *ie*, genes are represented as vertices in the graph
  - Choose a distance threshold θ
  - If the distance between two vertices is below θ, draw an edge between them
  - The resulting graph may contain cliques
  - These cliques represent clusters of closely located data points!

Clique graph clustering

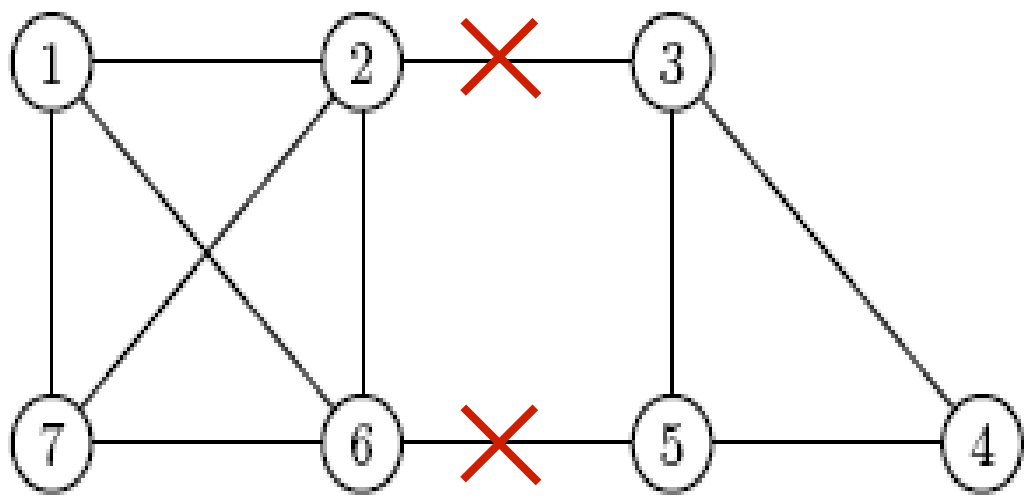
corrupted cliques problem

Input: A graph **G**

Output: The smallest number of additions and removals of edges that will transform *G* into a clique graph

Clique graph clustering

- A graph can be transformed into a clique graph by (minimally) adding or removing edges



Clique graph clustering

matrix to graph

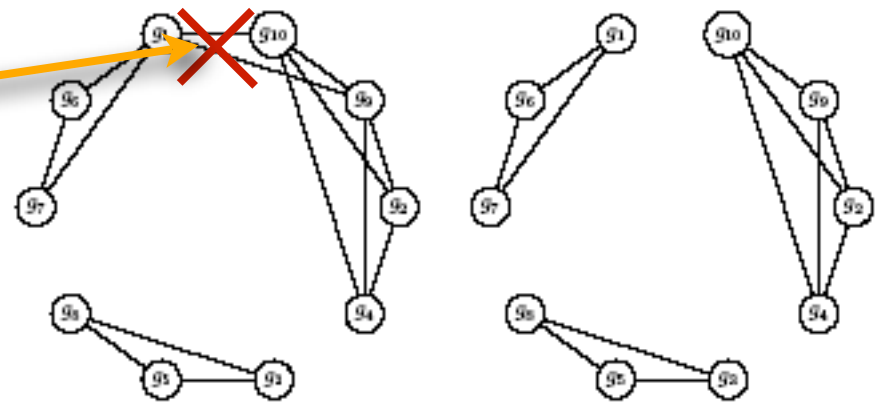
The distance graph (threshold θ=7) is transformed into a clique graph after removing the two highlighted edges

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

(a) Distance matrix, d (distances shorter than 7 are shown in bold).

After transforming the distance graph into the clique graph, the dataset is partitioned into three clusters

Corrupted clique



(b) Distance graph for $\theta = 7$.

(c) Clique graph.

Clique graph clustering

corrupted cliques problem

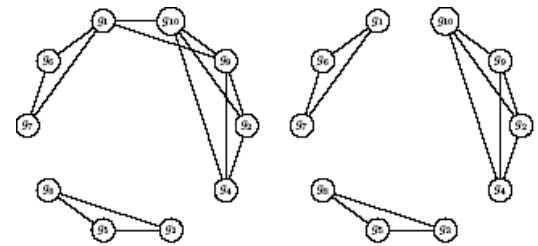
- Corrupted Cliques problem is **NP-Hard**, some heuristics exist to approximately solve it:
- **CAST** (Cluster Affinity Search Technique): a practical and fast algorithm:
  - **CAST** is based on the notion of genes *close* to cluster *C* or *distant* from cluster *C*
  - Distance between gene *i* and cluster *C*:

    d(i,C) = average distance between gene *i* and all genes in *C*

  - Gene i is close to cluster C if $d(i,C) < \theta$ and distant otherwise

Clique graph clustering

CAST algorithm

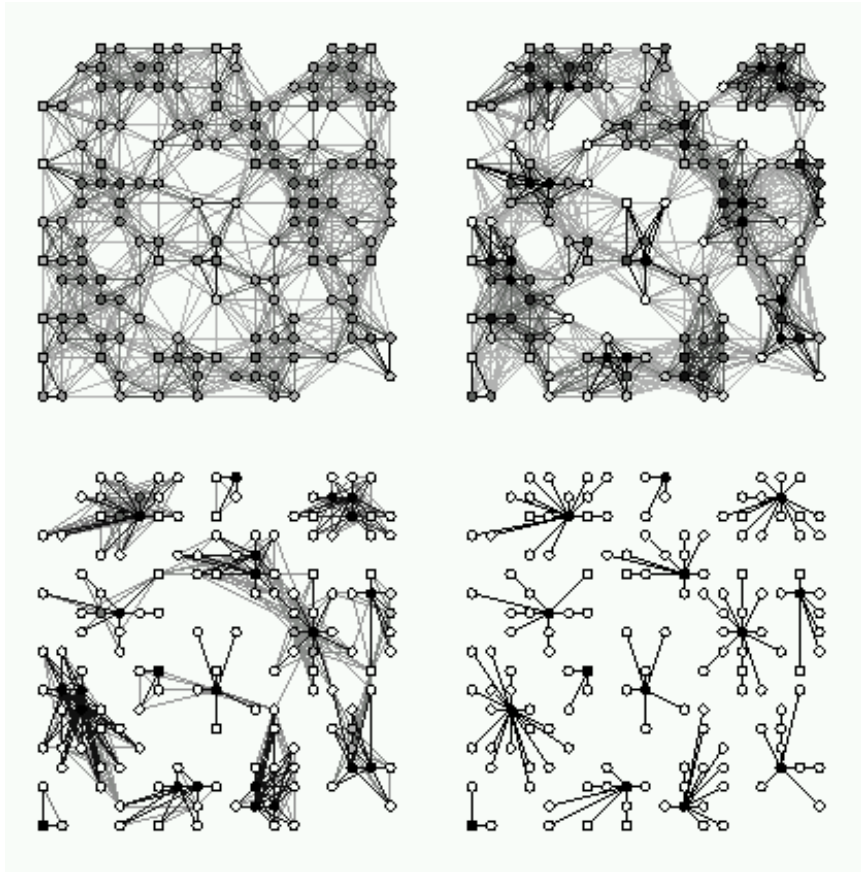


1. **P** ← Ø
2. while **S** ≠ Ø
3. **v** ← vertex of maximal degree in the distance graph **G**
4. **C** ← {**v**}
5. while a close gene **i** not in **C** or distant gene **i** in **C** exists
6. Find the nearest close gene **i** not in **C** and add it to **C**
7. Remove the farthest distant gene **i** in **C**
8. Add cluster **C** to partition **P**
9. **S** ← **S** \ **C**
10. Remove vertices of cluster **C** from the distance graph **G**
11. return **P**

S – set of elements, G – distance graph, θ – distance threshold

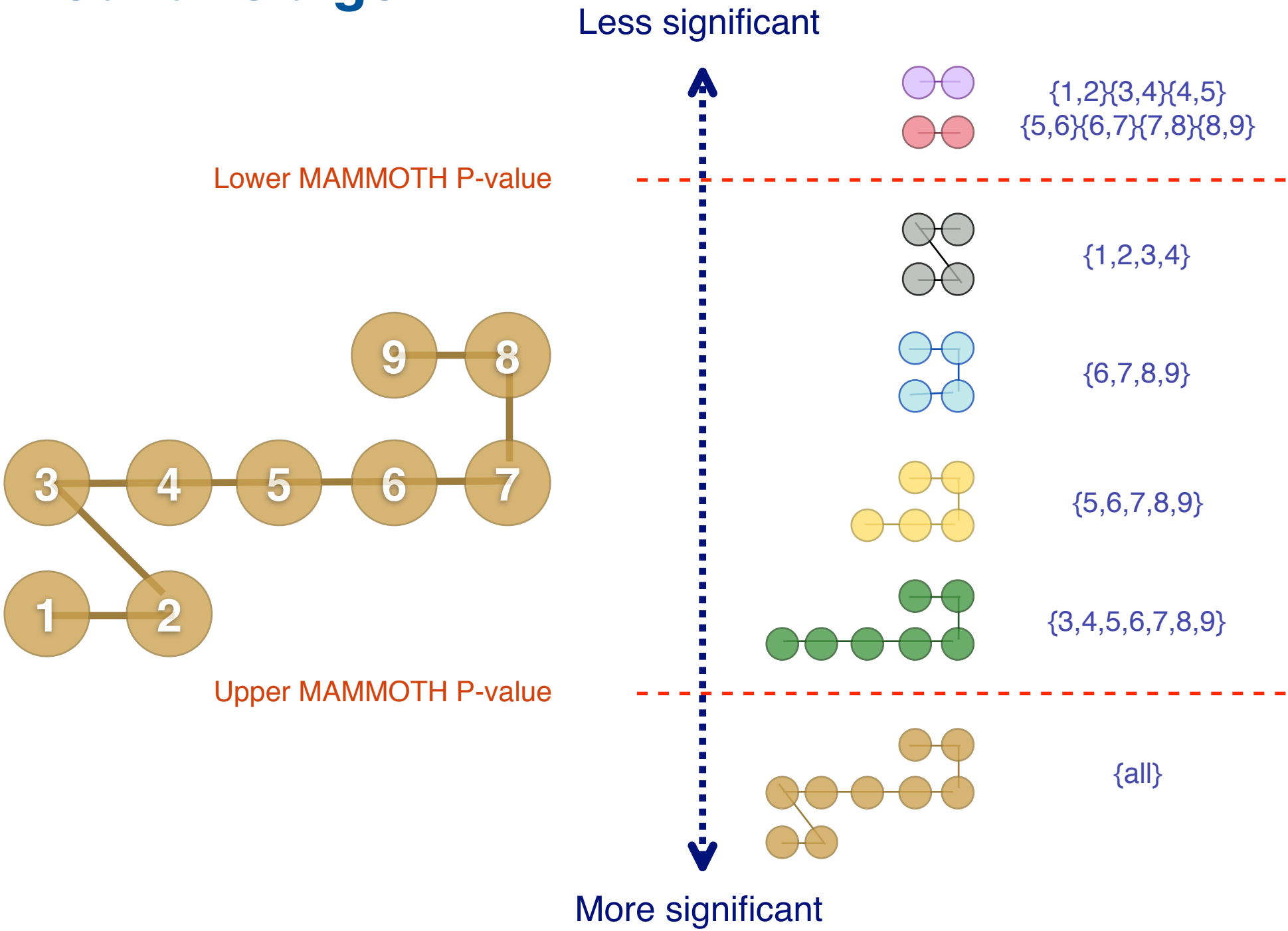
Clustering application

defining domains using a graph based clustering method

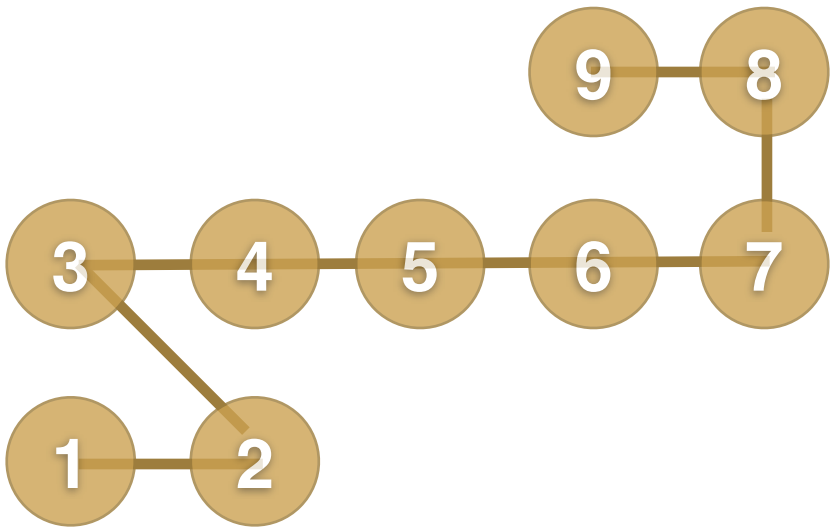


MCL - a cluster algorithm for graphs
http://micans.org/mcl/

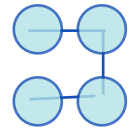
ModParts algorithm



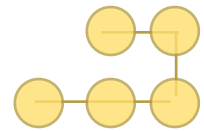
ModParts algorithm



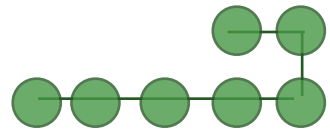
{1,2,3,4}



{6,7,8,9}

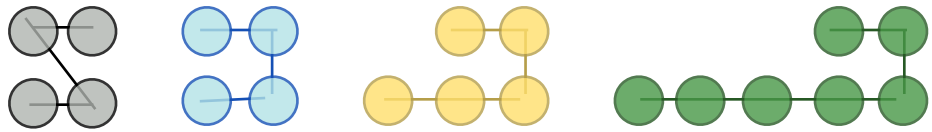


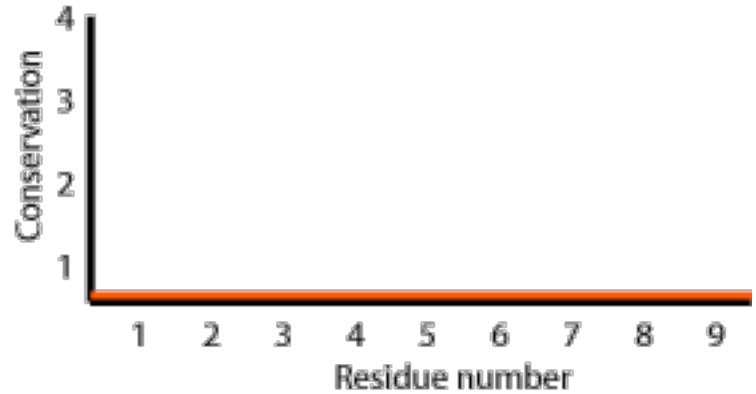
{5,6,7,8,9}

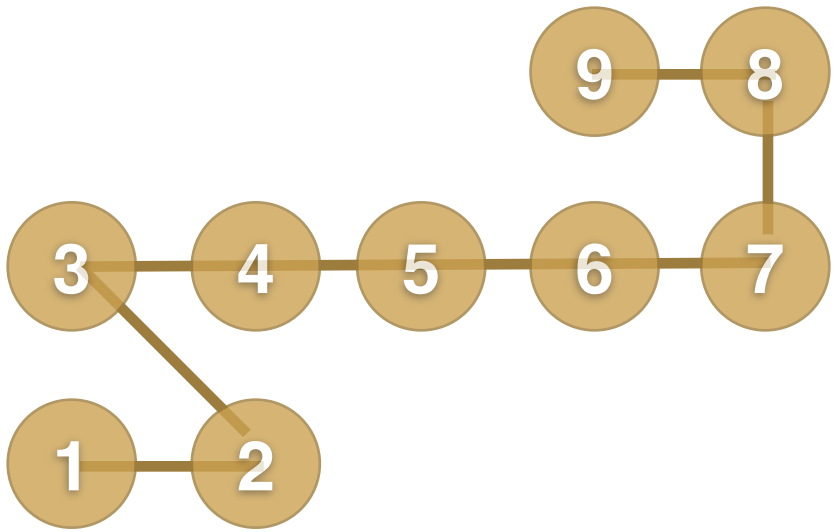


{3,4,5,6,7,8,9}

ModParts algorithm

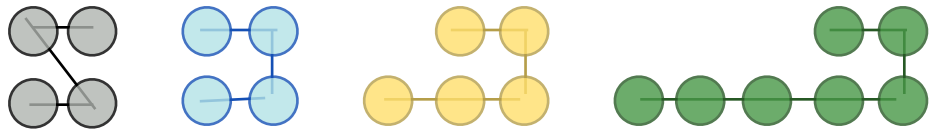


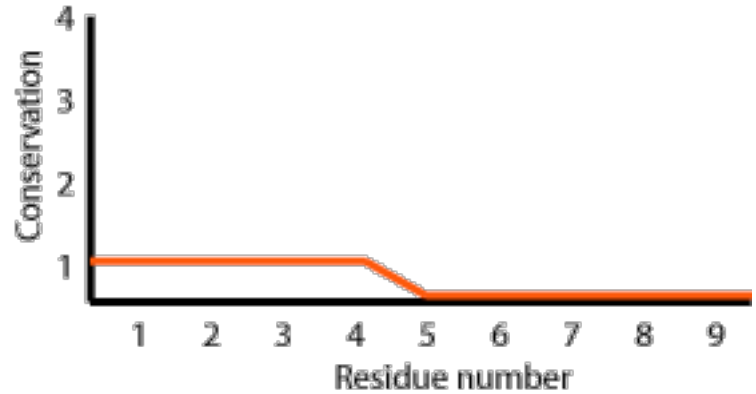


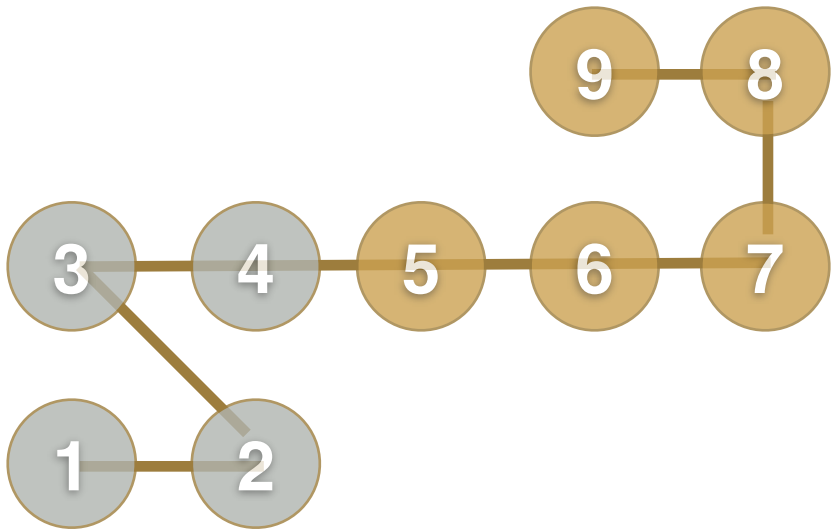


#	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

ModParts algorithm

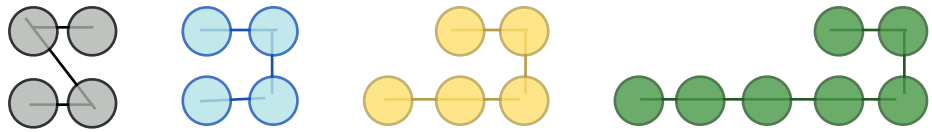


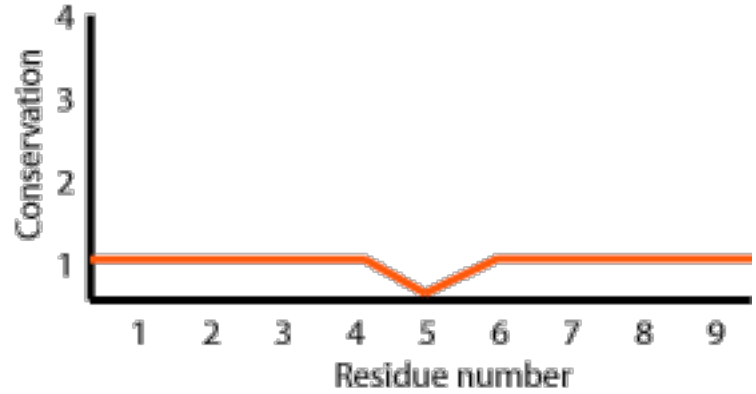


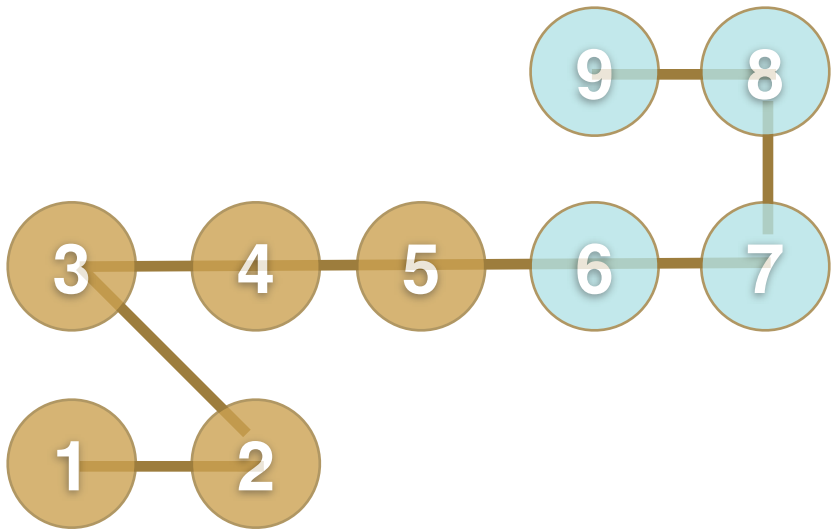


#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

ModParts algorithm

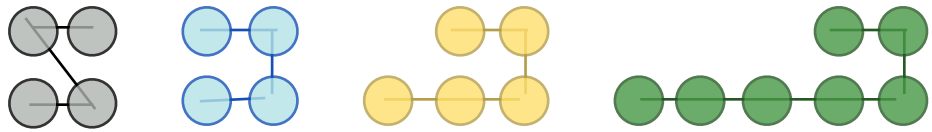


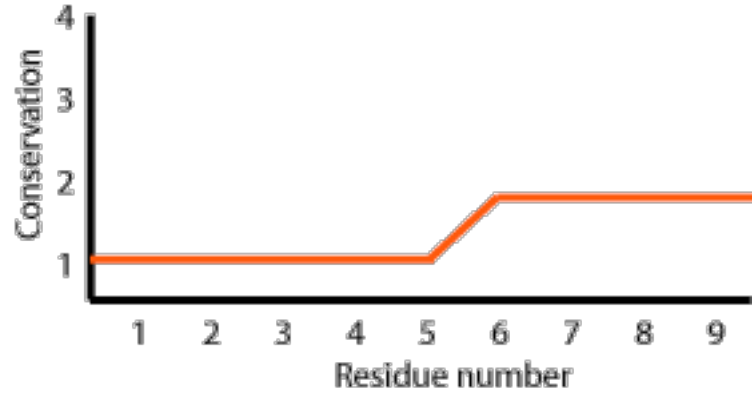


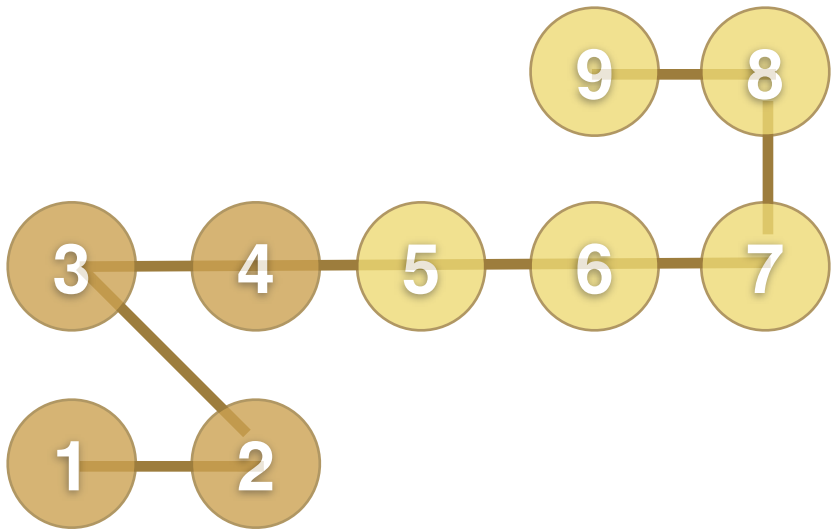


#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	1	1	1
7	0	0	0	0	0	1	1	1	1
8	0	0	0	0	0	1	1	1	1
9	0	0	0	0	0	1	1	1	1

ModParts algorithm

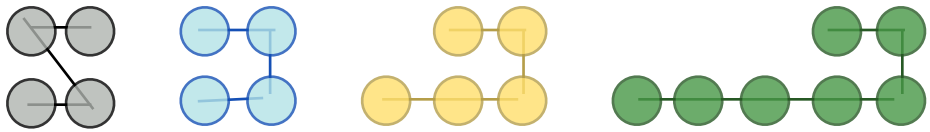


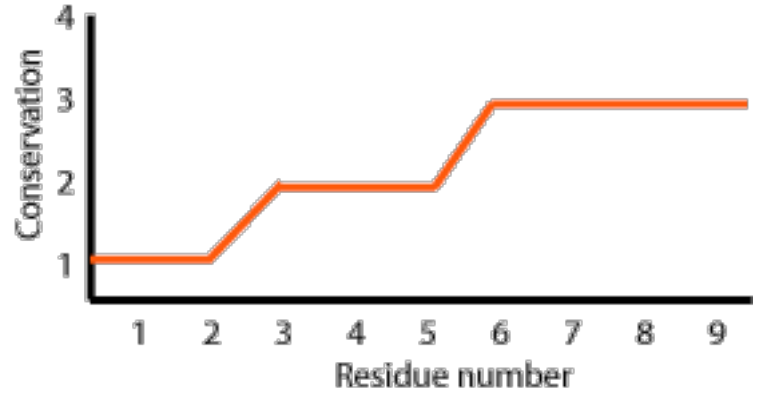


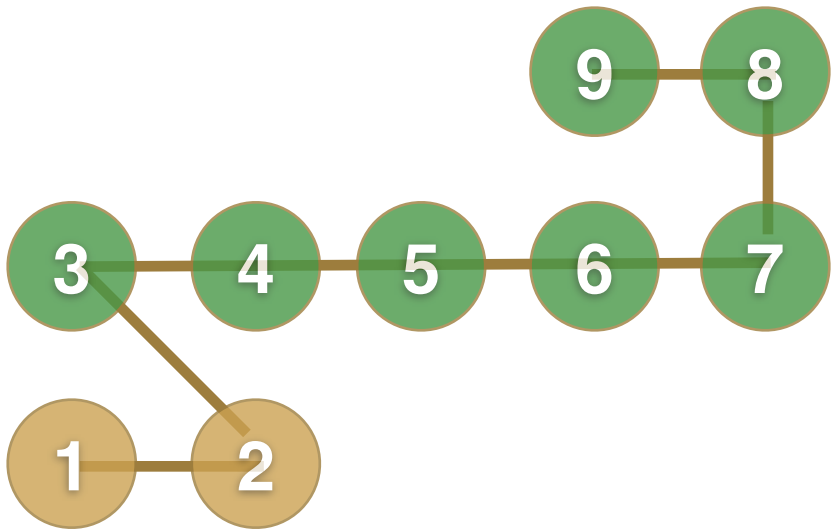


#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0	0
5	0	0	0	0	1	1	1	1	1
6	0	0	0	0	1	2	2	2	2
7	0	0	0	0	1	2	2	2	2
8	0	0	0	0	1	2	2	2	2
9	0	0	0	0	1	2	2	2	2

ModParts algorithm

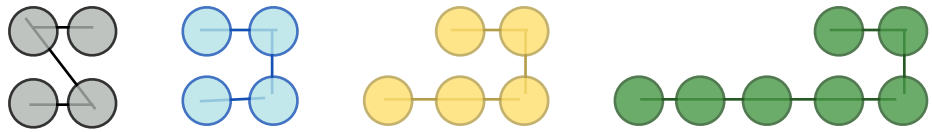


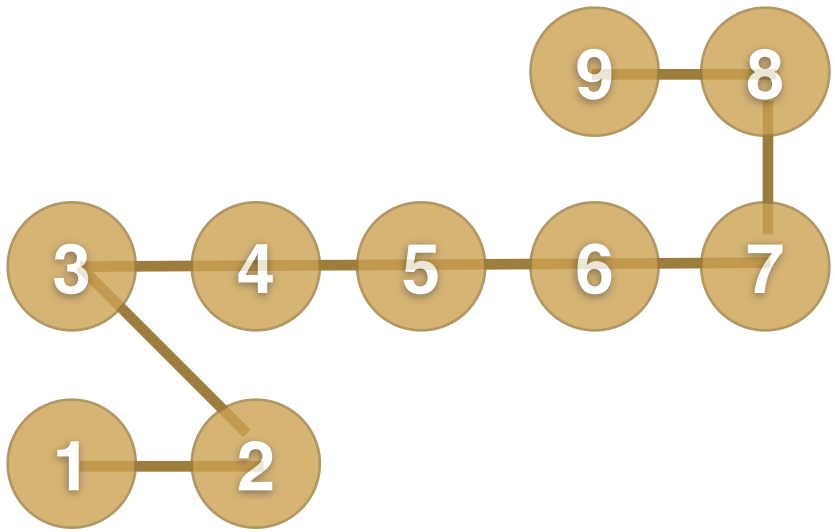


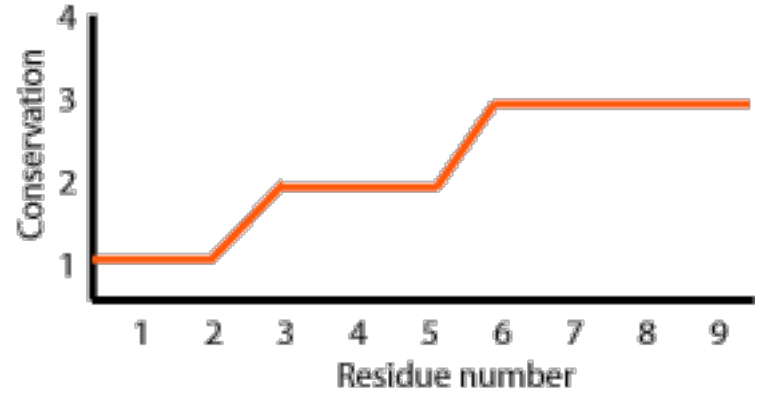


#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	2	2	1	1	1	1	1
4	1	1	2	2	1	1	1	1	1
5	0	0	1	1	2	2	2	2	2
6	0	0	1	1	2	3	3	3	3
7	0	0	1	1	2	3	3	3	3
8	0	0	1	1	2	3	3	3	3
9	0	0	1	1	2	3	3	3	3

ModParts algorithm

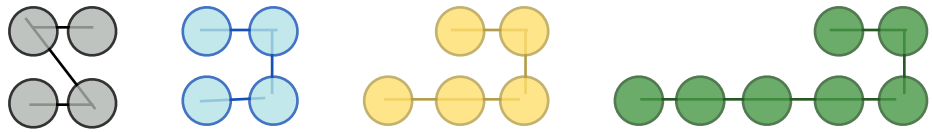


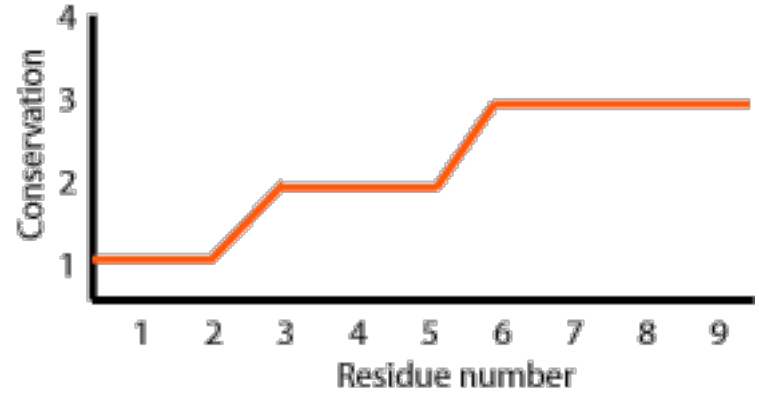


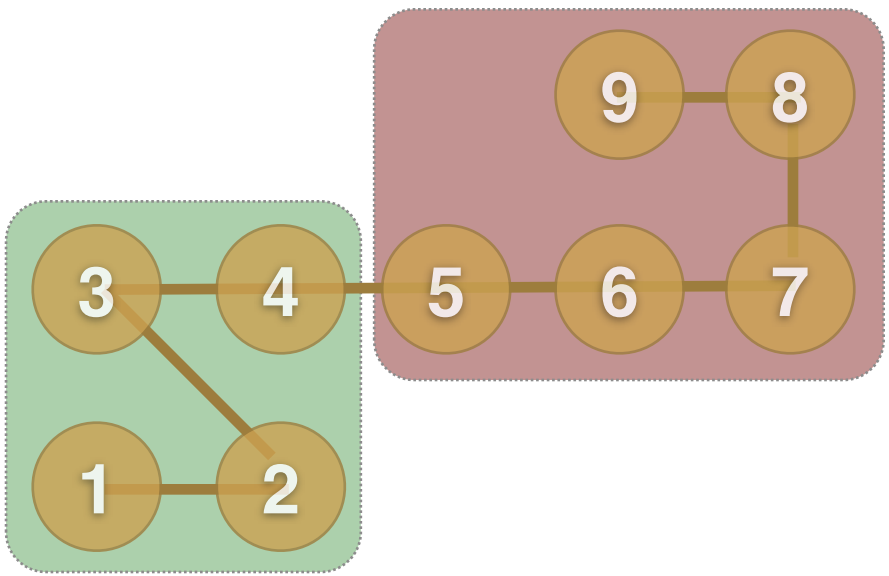


#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	2	2	1	1	1	1	1
4	1	1	2	2	1	1	1	1	1
5	0	0	1	1	2	2	2	2	2
6	0	0	1	1	2	3	3	3	3
7	0	0	1	1	2	3	3	3	3
8	0	0	1	1	2	3	3	3	3
9	0	0	1	1	2	3	3	3	3

ModParts algorithm







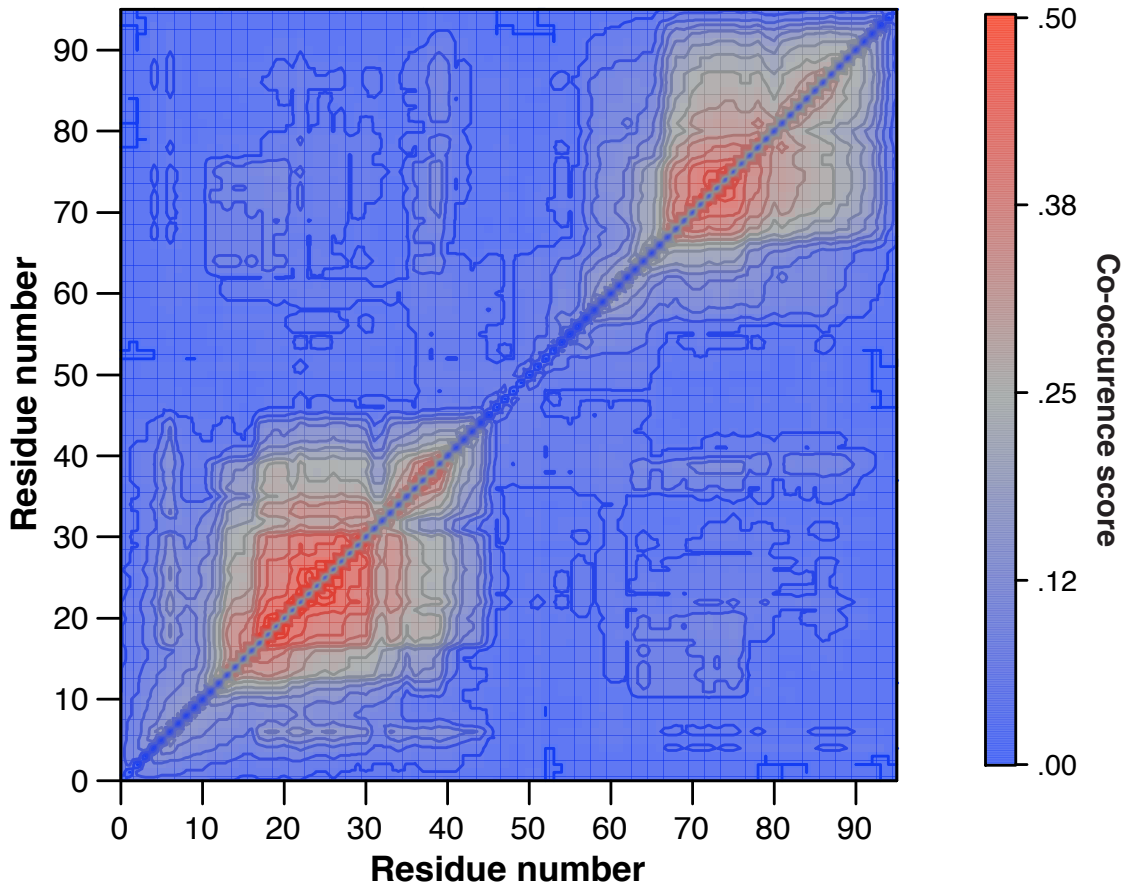
#	1	2	3	4	5	6	7	8	9
1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0
3	1	1	2	2	1	1	1	1	1
4	1	1	2	2	1	1	1	1	1
5	0	0	1	1	2	2	2	2	2
6	0	0	1	1	2	3	3	3	3
7	0	0	1	1	2	3	3	3	3
8	0	0	1	1	2	3	3	3	3
9	0	0	1	1	2	3	3	3	3

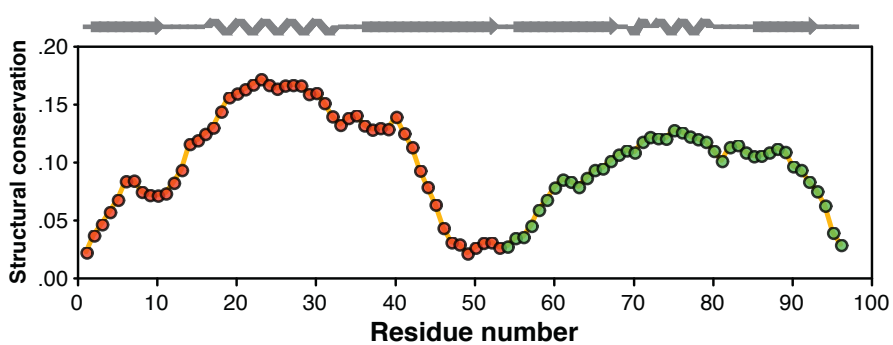
Threshold #3 MCL Cluster level (-I)

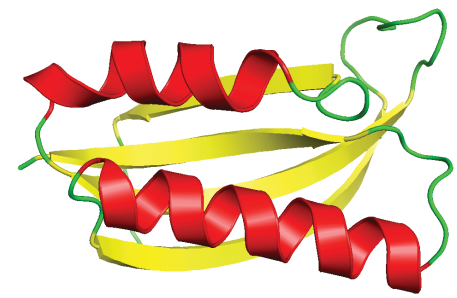
Stijn van Dongen (`http://micans.org/mcl/`)

Repetitions as recurrent fragments

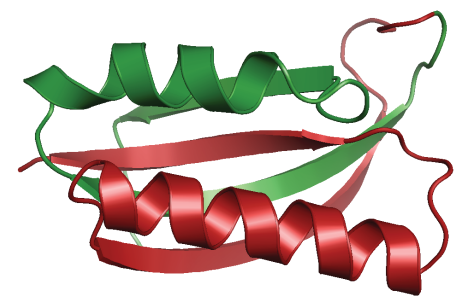
Ribosomal protein S6 (1ris) α+β → *Ferrodoxin Like domain*



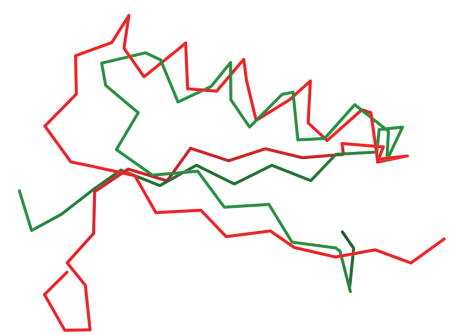




1ris:_



unit 1 1-53
unit 2 54-97



superimposed reversed unit 1 and unit 2
RMSD 3.1Å over 39 Cα carbons

Domains as recurrent fragments

2163 chains from Islam et al. 1995 → 569 Non-redundant

<2Å && <30aa diff.

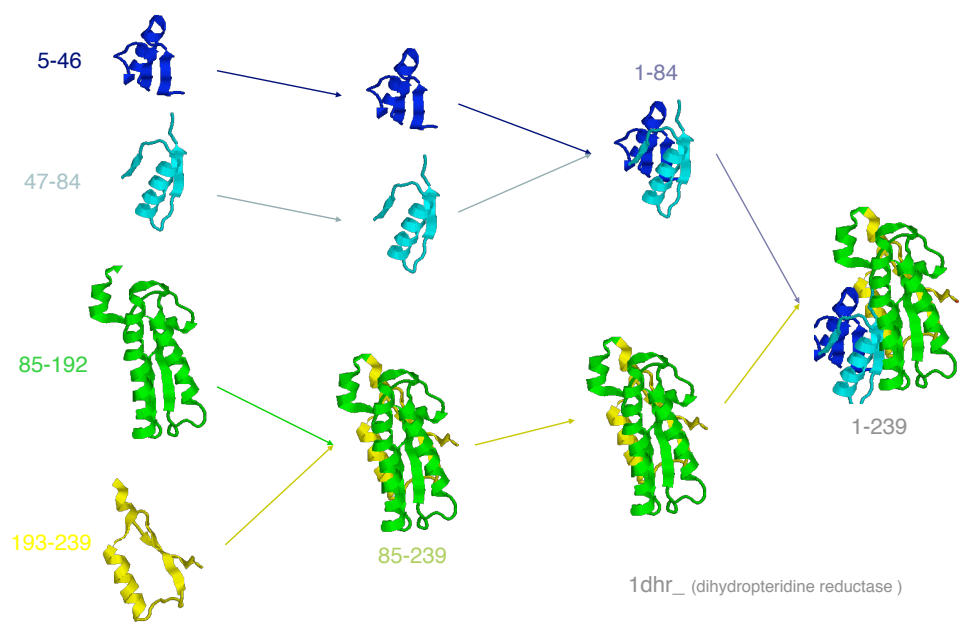
Divide randomly into two sets

Remove of incomplete or obsolete entries.

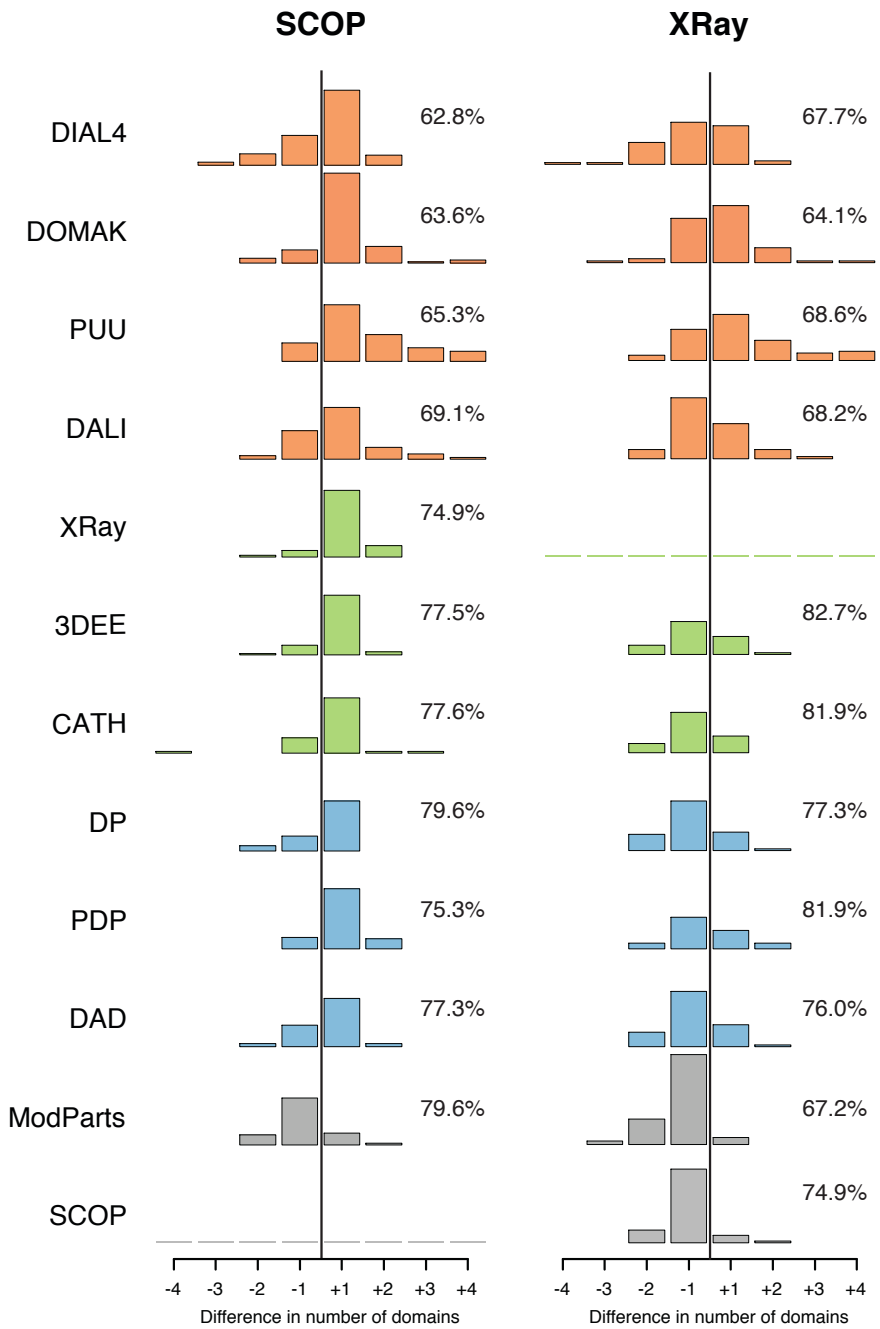
Training set → 242 chains

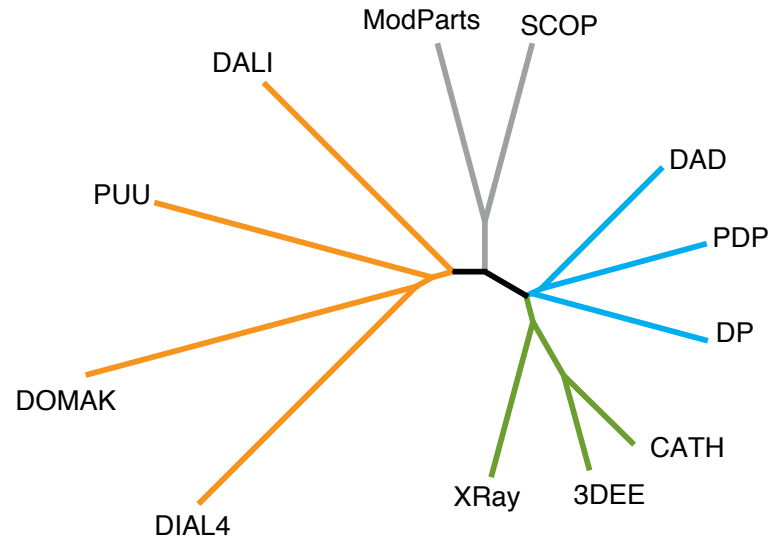
Testing set → 234 chains

R = Volume/ASA

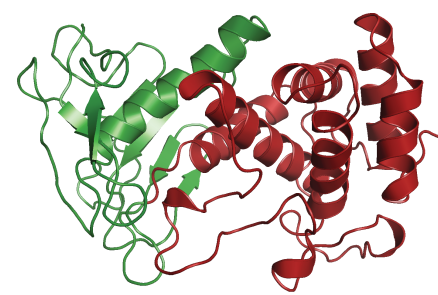


Domains as recurrent fragments



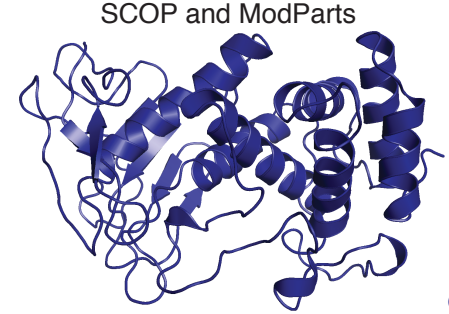


All other methods



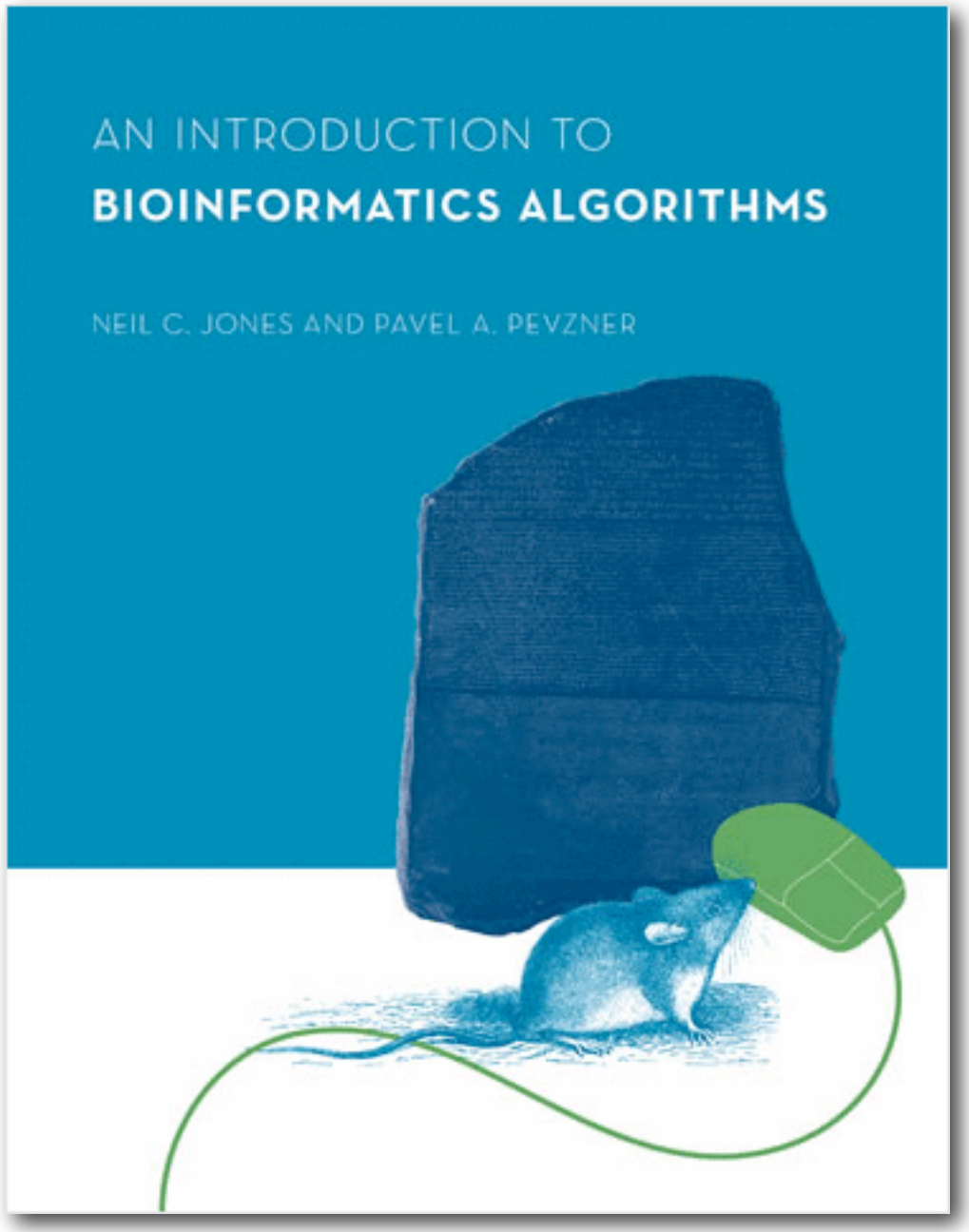
1ezm:_
domain 1 1-134
domain 2 135-298

SCOP and ModParts



1ezm:_
domain 1 1-298

http://www.bioalgorithms.info



BMI-206

Structure-Structure comparisons
Sequence-Structure comparisons

Marc A. Marti-Renom
Assistant Adjunct Professor
Department of Biopharmaceutical Sciences

February 2nd, 2006

How to use this lectures

- Ask!

- Outline
  - Basic introduction
  - Theory (representation-scoring-optimization)
  - Available programs
  - Application

Structure-Structure comparison

- Outline
  - Before we start…
    - Some theory
    - Coverage .vs. Accuracy
  - How can we compare structures…
    - SALIGN (properties comparison)
    - VAST (vector alignment)
    - CE (local heuristic comparison)
    - MAMMOTH (vector alignment)
  - How we classify the structural space…
    - SCOP (manual)
    - CATH (semi-automatic)
    - DBAli (fully automatic and comprehensive)

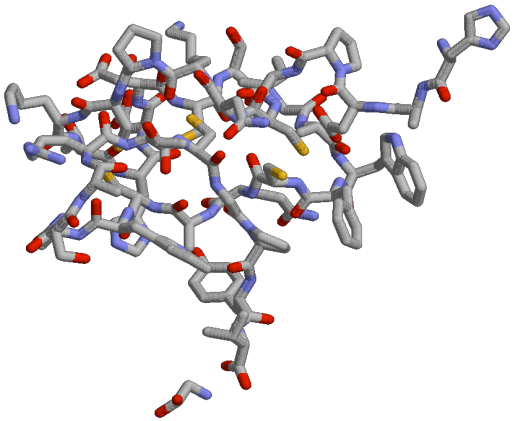
Structure-Structure alignments

As any other bioinformatics problem…

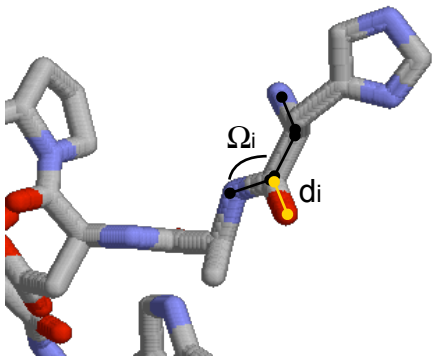
- **Representation**
- **Scoring**
- **Optimizer**

Representation

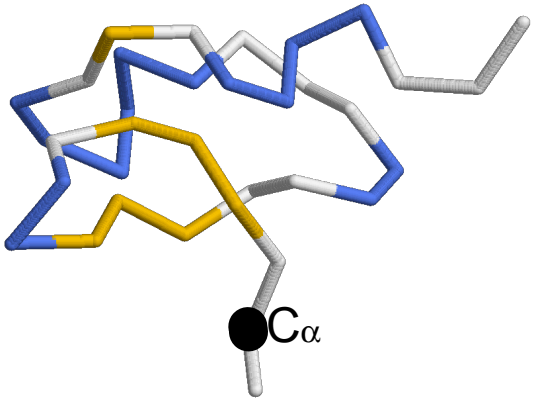
Structures



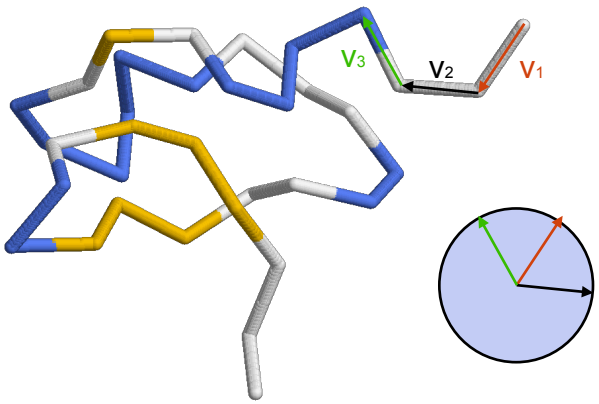
All atoms and coordinates



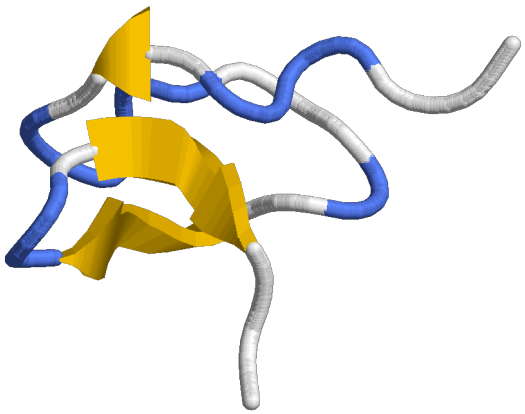
Dihedral space or distance space



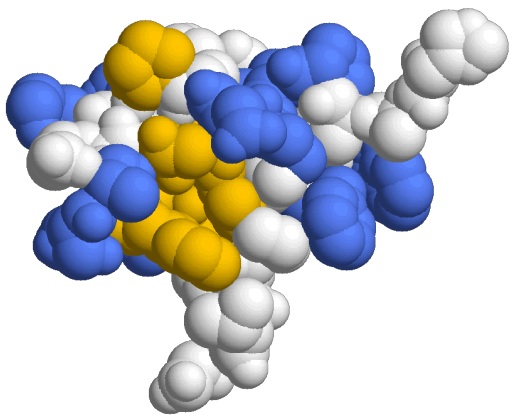
Reduced atom representation



Vector representation



Secondary Structure



Accessible surface (and others)

Scoring

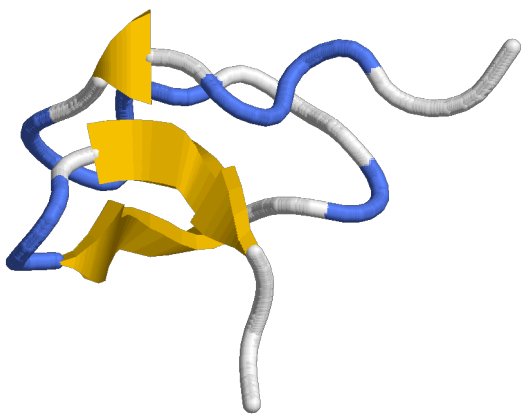
Raw scores

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	**4**	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	**4**	1	-1	1	0	1	0	0	0	-1	0	-1	-2	-2	-2	-2	-2	-3
P	-3	-1	1	**7**	-1	-2	-1	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	-1	-1	**4**	0	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-2	-3
G	-3	0	1	-2	0	**6**	-2	-1	-2	-2	-2	-2	-2	-3	-4	-4	0	-3	-3	-2
N	-3	1	0	-2	-2	0	**6**	1	0	0	-1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	1	-1	-2	-1	1	**6**	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	0	-1	-1	-2	0	2	**5**	2	0	0	1	-2	-3	-3	-3	-3	-2	-3
Q	-3	0	0	-1	-1	-2	0	0	2	**5**	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	0	-2	-2	-2	1	1	0	0	**8**	0	-1	-2	-3	-3	-2	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	0	-1	-1	-2	0	-1	1	1	-1	2	**5**	-1	-3	-2	-3	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**	1	2	-2	0	-1	-1
I	-1	-2	-2	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**	2	1	0	-1	-3
L	-1	-2	-2	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**	3	0	-1	-2
V	-1	-2	-2	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	2
W	-2	-3	-3	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**

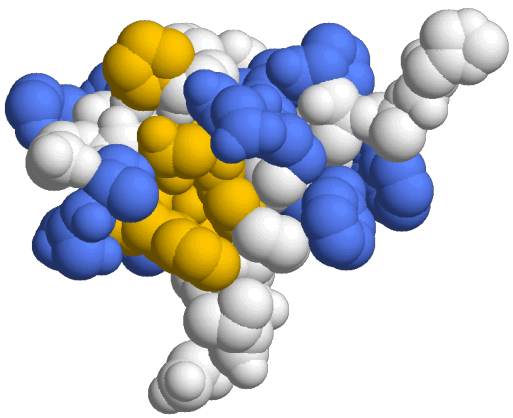
Aminoacid substitutions

$$RMSD = \sqrt{\sum \left(x_i - \bar{x} \right)^2}$$

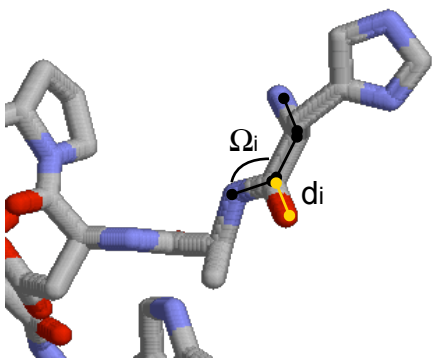
Root Mean Square Deviation



Secondary Structure (H,B,C)



Accessible surface (B,A [%])



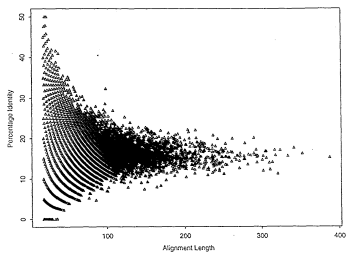
Angles or distances

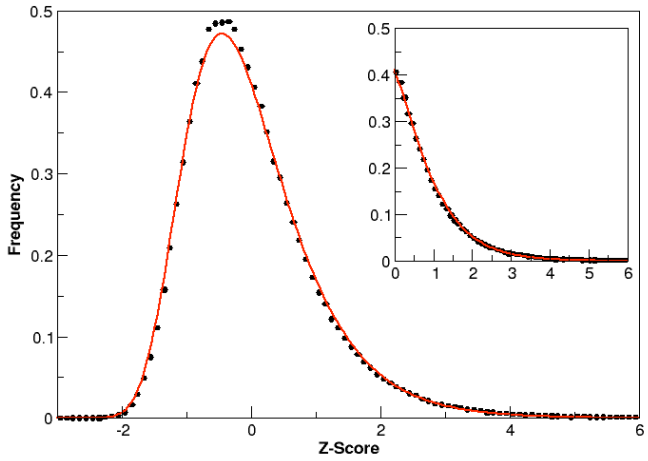
Significance of an alignment (score)

remember Patsy's class

Probability that the optimal alignment of two random sequences/structures of the same length and composition as the aligned sequences/structures have at least as good a score as the evaluated alignment.

Empirical





Sometimes approximated by Z-score (normal distribution).

Analytic

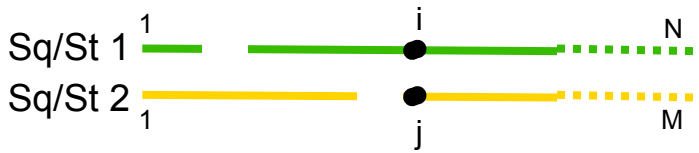
$$P(s) = e^{-\lambda (s-\mu)}$$

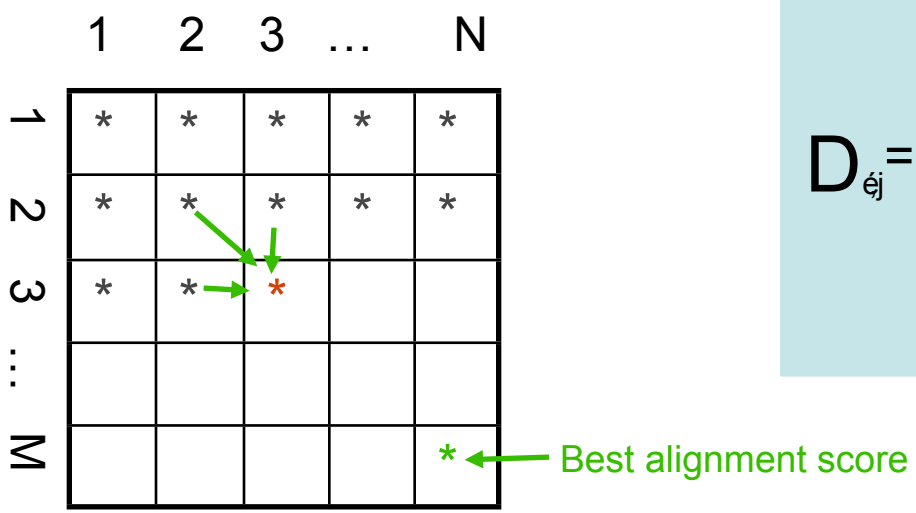
$$P(s \geq x) = 1 - \exp\left(e^{-\lambda (s-\mu)}\right)$$

Karlin and Altschul, 1990 *PNAS* **87**, pp2264

Global dynamic programming alignment

remember Patsy's class





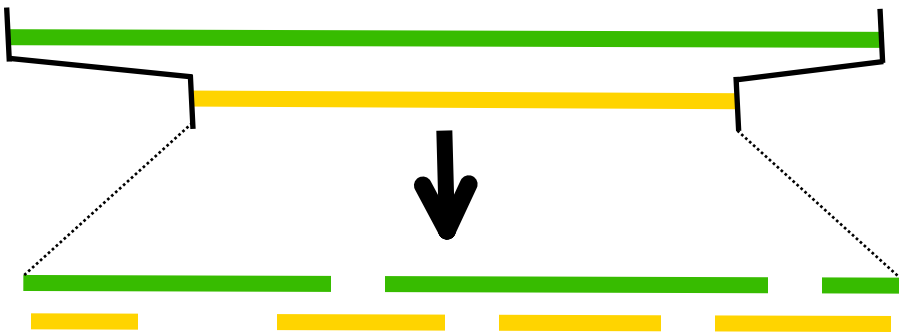
$$D_{ij} = \min \begin{cases} D_{i,j-1} + \text{Score}_{(Ä,rj)} \\ D_{i-1,j-1} + \text{Score}_{(ri,rj)} \\ D_{i-1,j} + \text{Score}_{(ri,Ä)} \\ 0 \end{cases}$$

Backtracking to get the best alignment

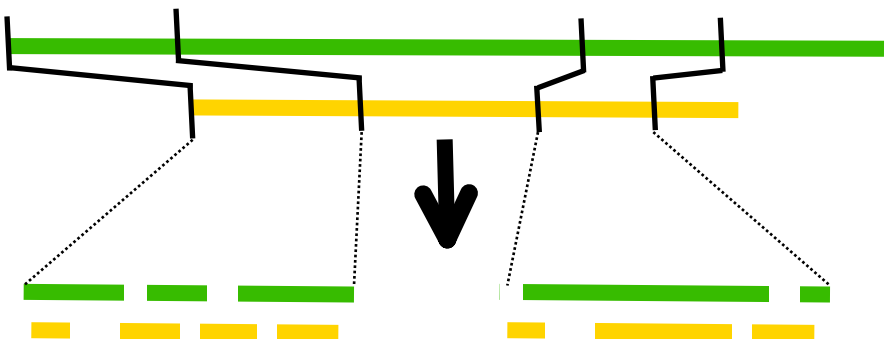
Needleman and Wunsch (1970) *J. Mol Biol*, **3** pp443

Global *.vs.* local alignment

remember Patsy's class



Global alignment



Local alignment

Optimizer

Multiple alignment

remember Patsy's class

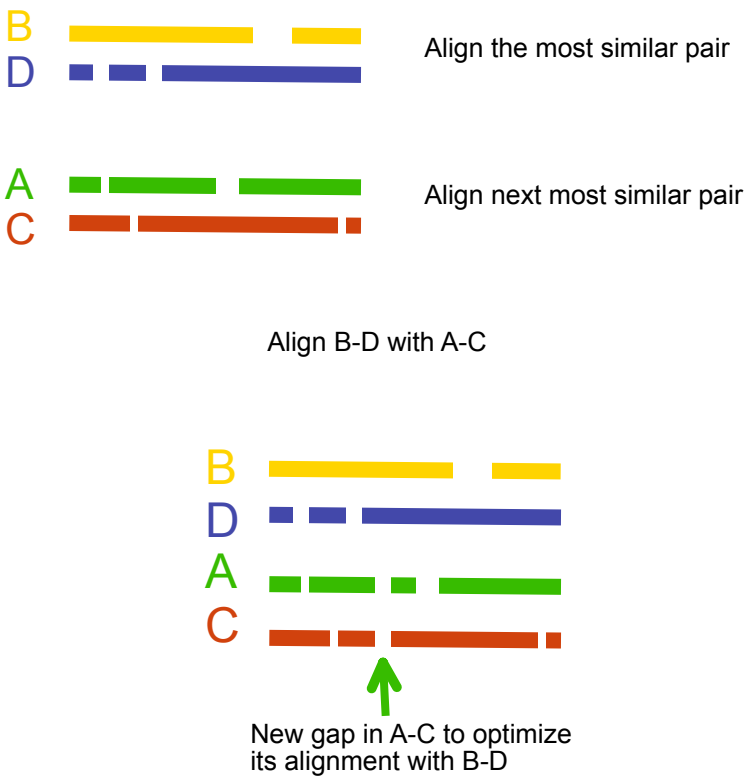
Pairwise alignments

Example – 4 sequences A, B, C, D.

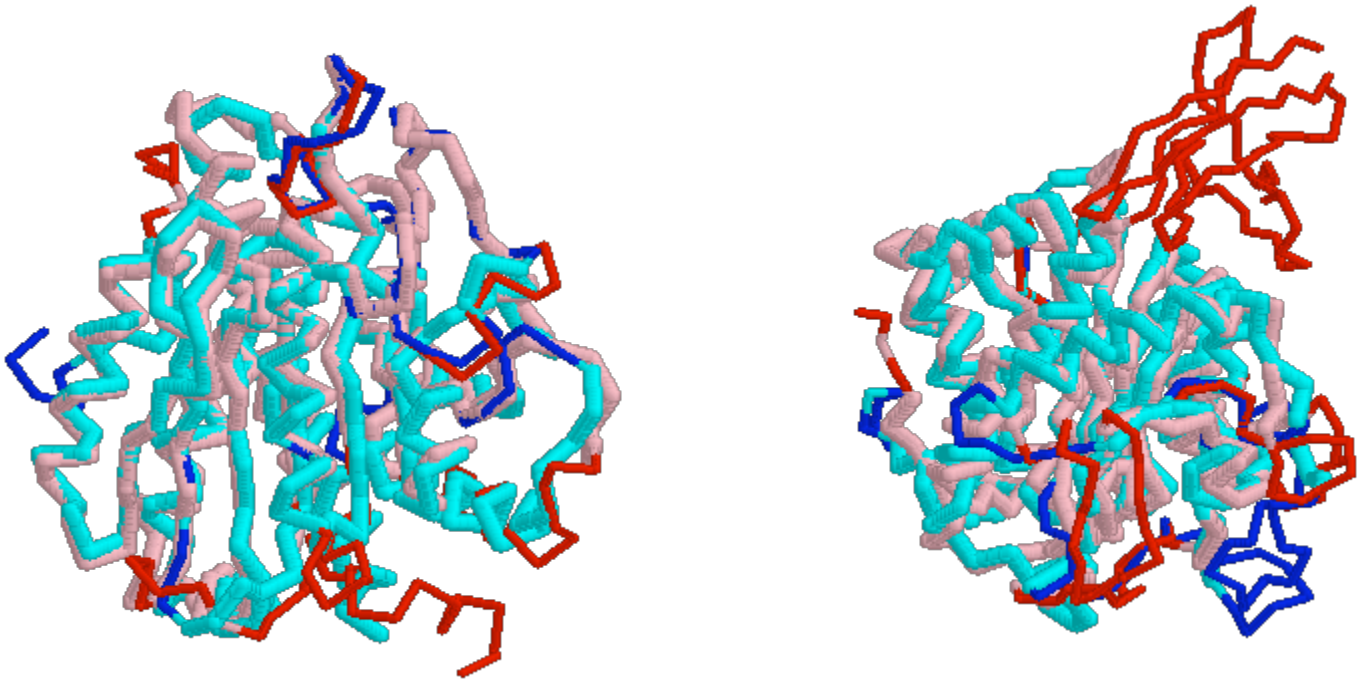
6 pairwise comparisons
then cluster analysis

Multiple alignments

Following the tree from step 1



Coverage .vs. Accuracy

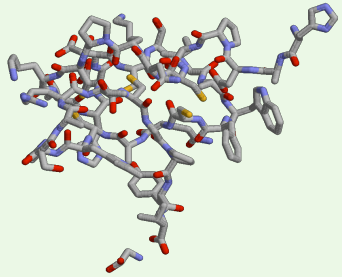


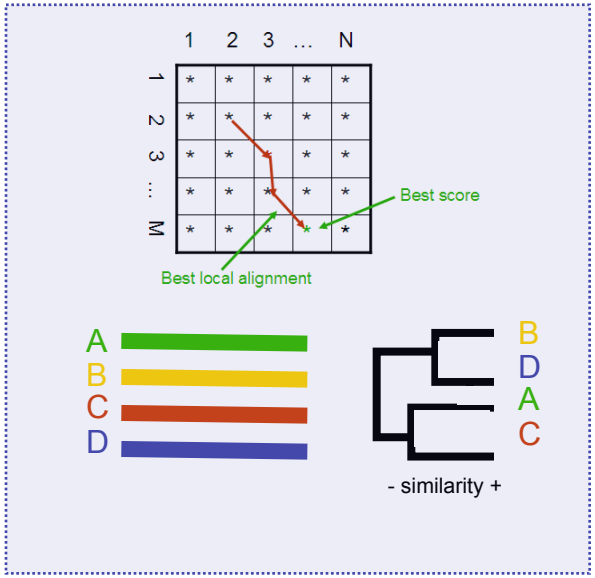
Same RMSD ~ 2.5Å

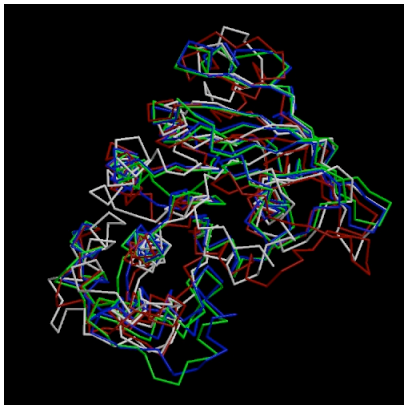
Coverage ~90% Cα

Coverage ~75% Cα

Sequence-Structure alignment by properties conservation (SALIGN-MODELLER)

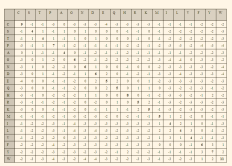




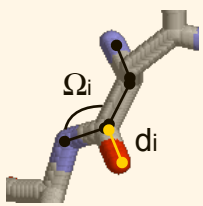


✓ Uses all available structural information
✓ Provides the optimal alignment

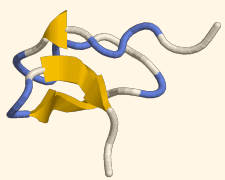
Computationally expensive



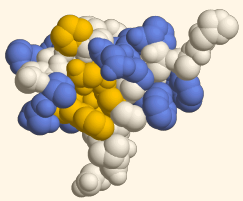
$R_{i,j}$



$D_{i(3),j(3)}$



$S_{i,j}$



$B_{i,j}$

$$RMSD = \sqrt{\sum (x_i - \bar{x})^2}$$

$I_{i,j}$

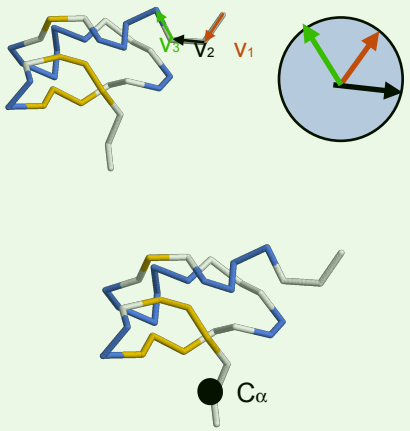
$$Score_{i,j} = w_1 * R_{i,j} + w_2 * D_{i(a),j(a)} + w_3 * S_{i,j} + w_4 * B_{i,j} + w_5 * I_{i,j} + w_6 * X_{i,j}$$

Madhusudhan et al. *in preparation*

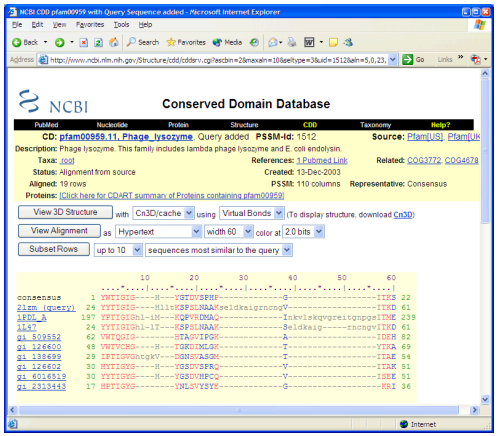
Structural alignment by properties conservation (SALIGN-MODELLER)

http://alto.compbio.ucsf.edu/salign-cgi/index.cgi

Vector Alignment Search Tool (VAST)

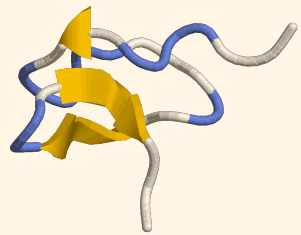


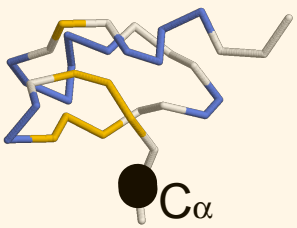
- Graph theory search of similar SSE
- Refining by Monte Carlo at all atom resolution

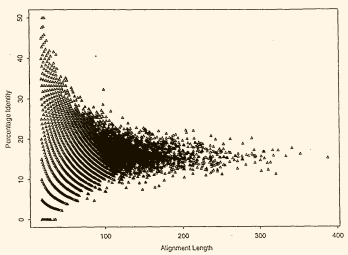


✓ **Good scoring system with significance**

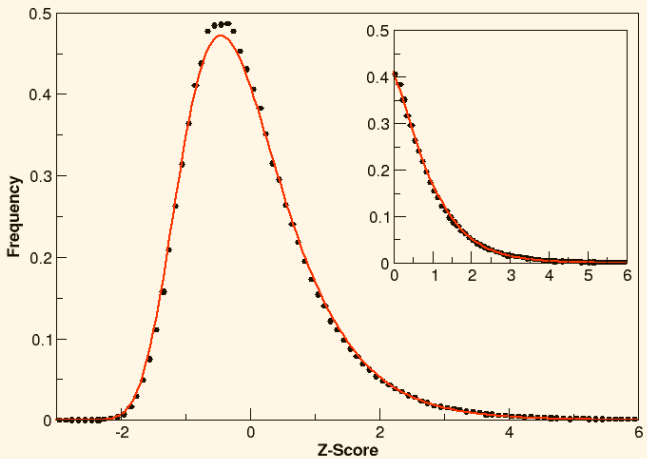
Reduces the protein representation







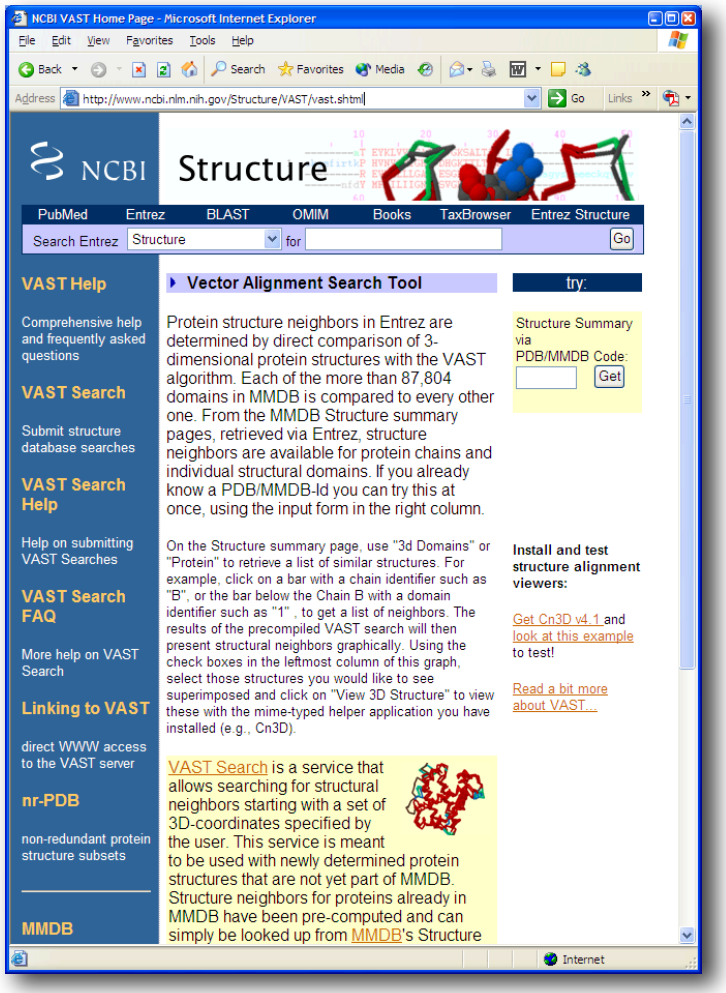
$$RMSD = \sqrt{\sum (x_i - \bar{x})^2}$$



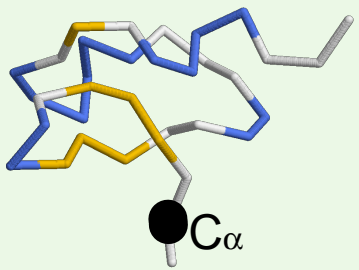
Gibrat JF et al. (1996) *Curr Opin Struct Biol* **3** pp377

Vector Alignment Search Tool (VAST)

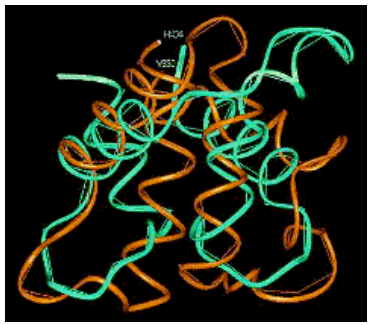
http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml



Incremental combinatorial extension (CE)

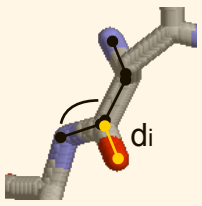


- Exhaustive combination of fragments
- Longest combination of AFPs
- Heuristic similar to PSI-BLAST



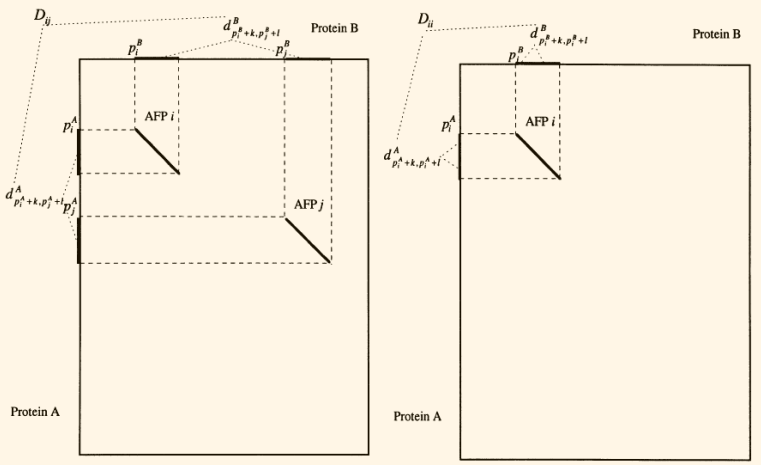
✓ **FAST!**
✓ **Good quality of local alignments**

Complicated scoring and heuristics



8 residues peptides

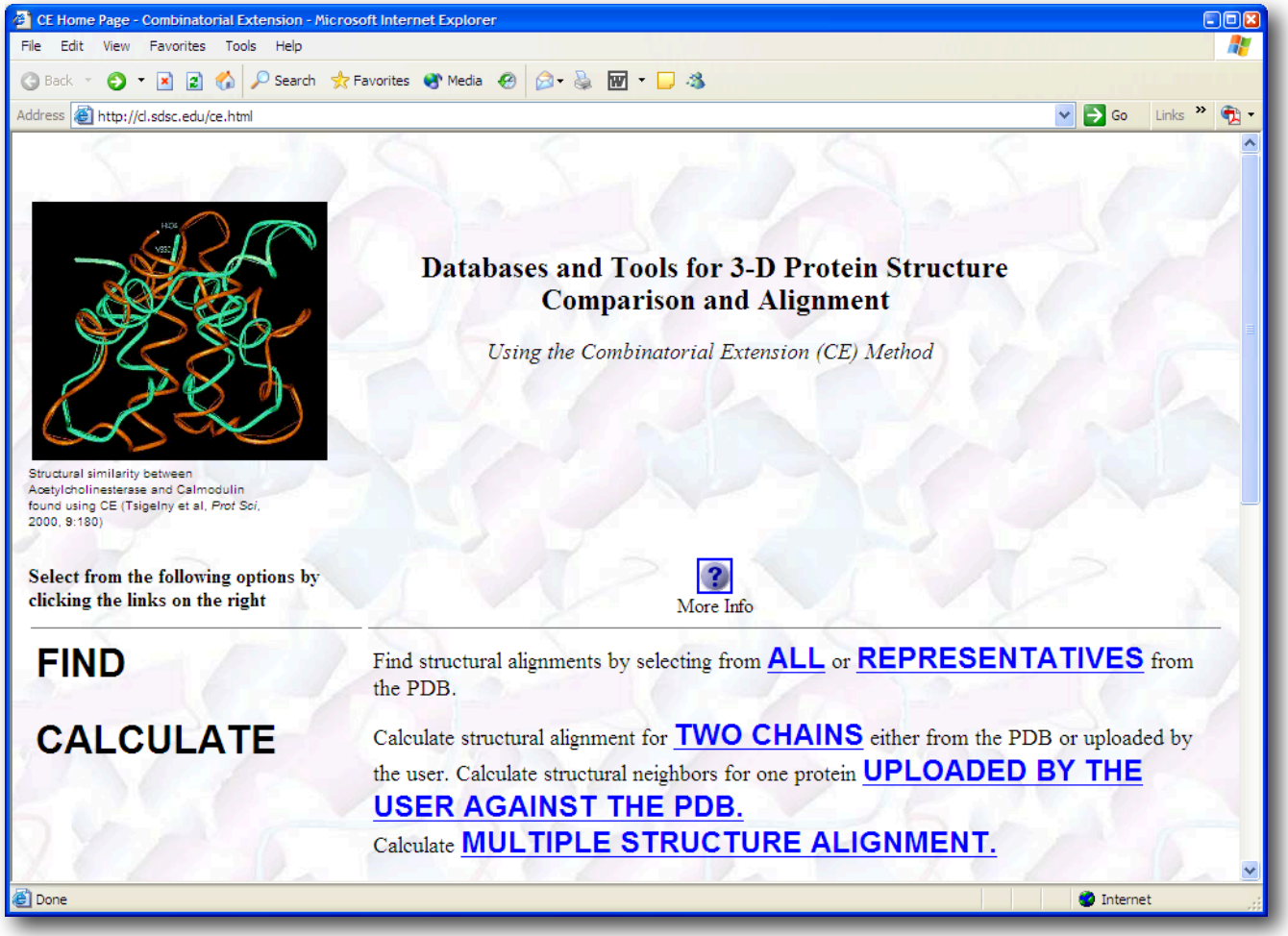
$$RMSD = \sqrt{\sum (x_i - \bar{x})^2}$$



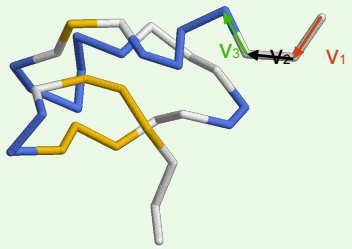
Shindyalov IN, amd Bourne PE. (1998) *Protein Eng.* **9** pp739

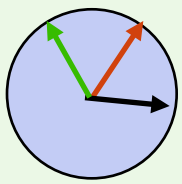
Incremental combinatorial extension (CE)

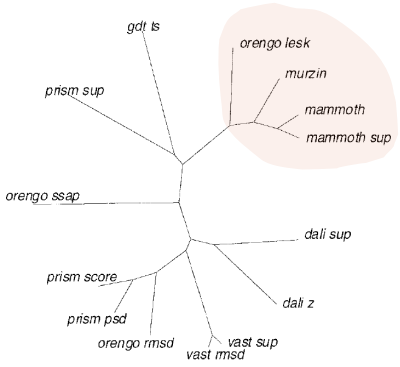
http://cl.sdsc.edu/ce.html



Matching molecular models obtained from theory (MAMMOTH)





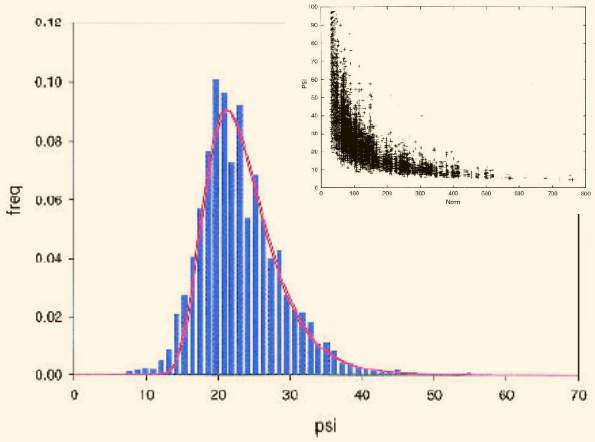


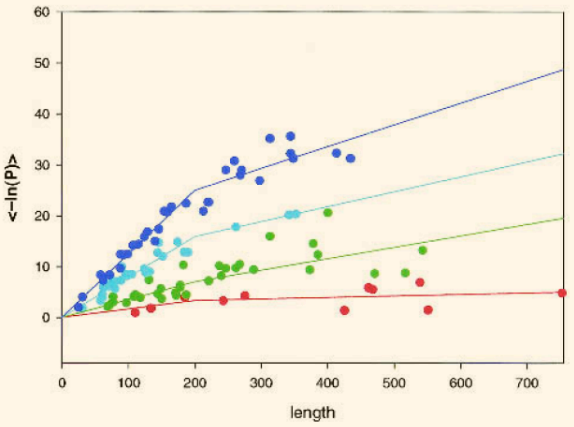
- ✓ **VERY FAST!**
- ✓ **Good scoring system with significance**

Reduces the protein representation

$$URMS^{R} = \sqrt{2.0 - \frac{2.84}{\sqrt{n}}}$$

$$S_{AB} = \frac{(URMS^{R} - URMS^{AB})D}{URMS^{R}}$$

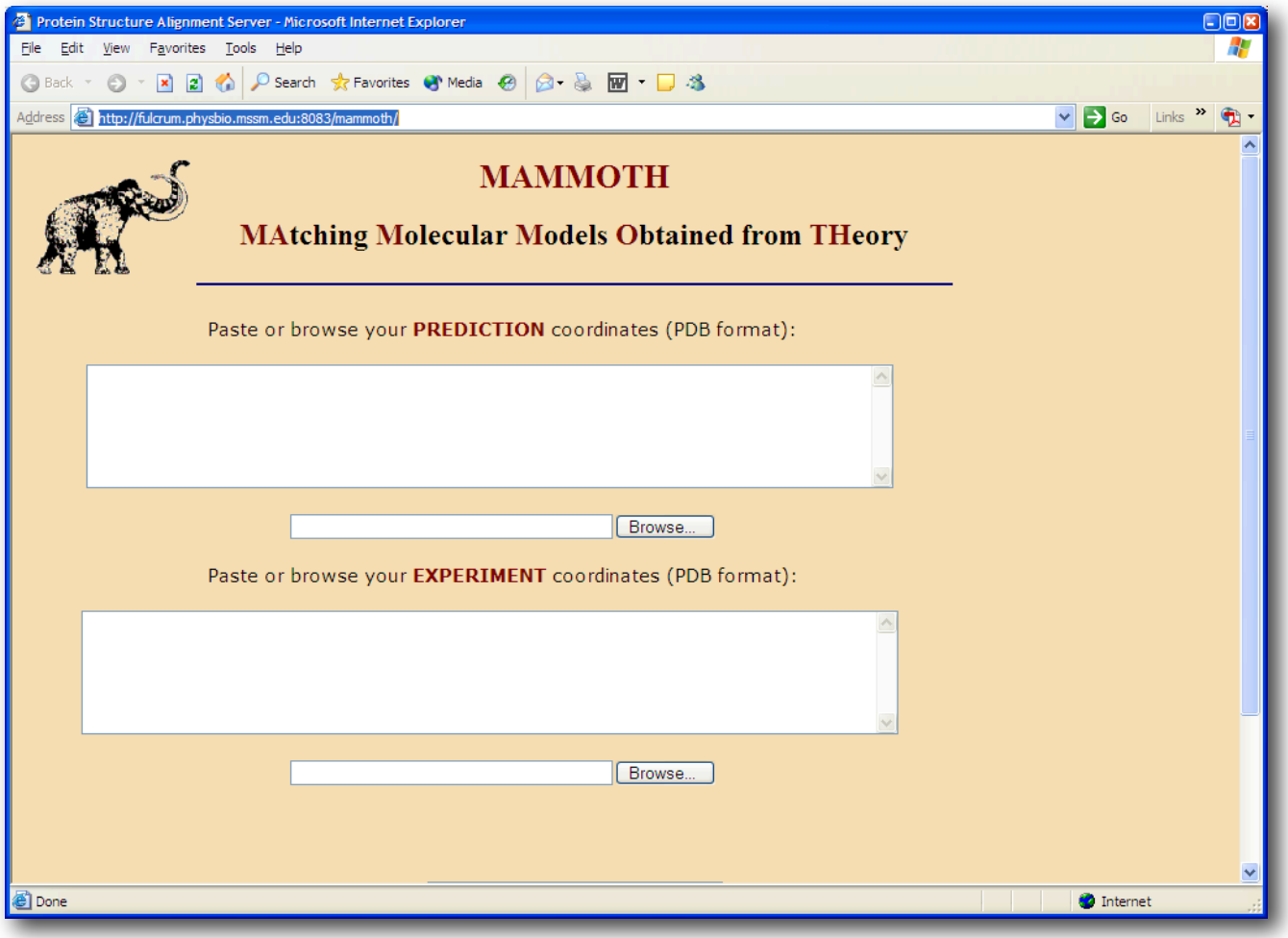




Ortiz AR, (2002) *Protein Sci.* **11** pp2606

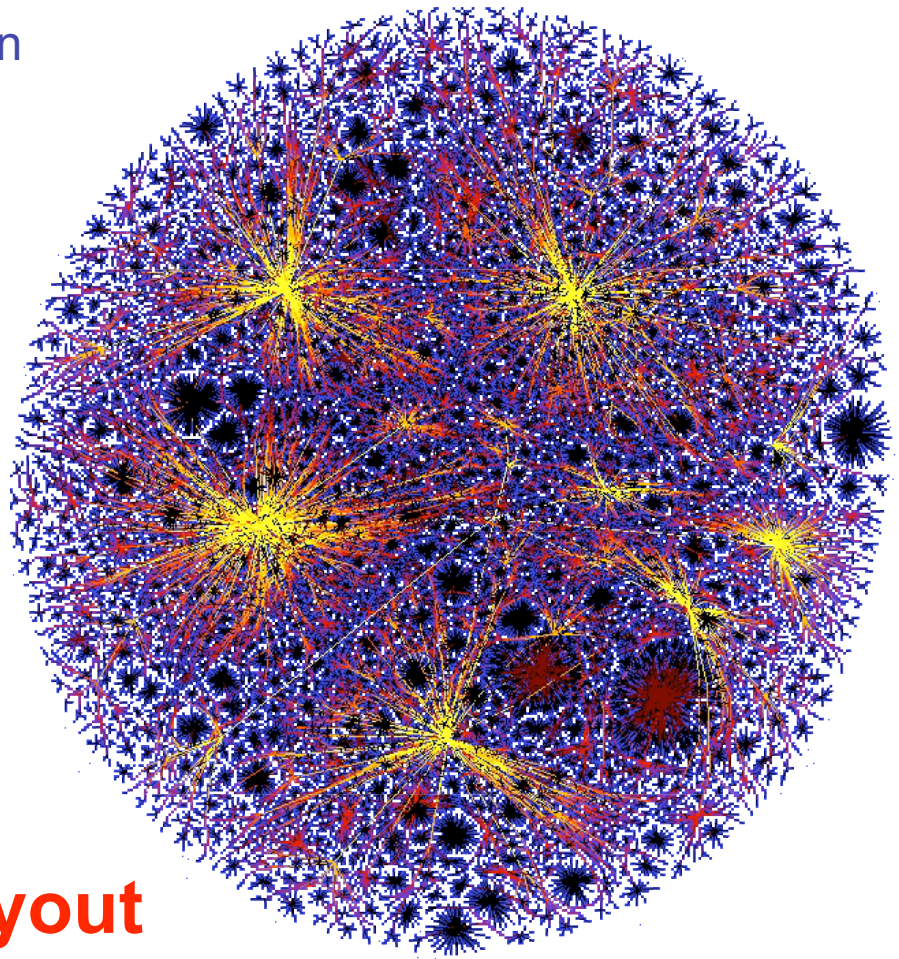
Matching molecular models obtained from theory (MAMMOTH)

http://fulcrum.physbio.mssm.edu:8083/



Classification of the structural space

SCOP classification



Large Graph Layout

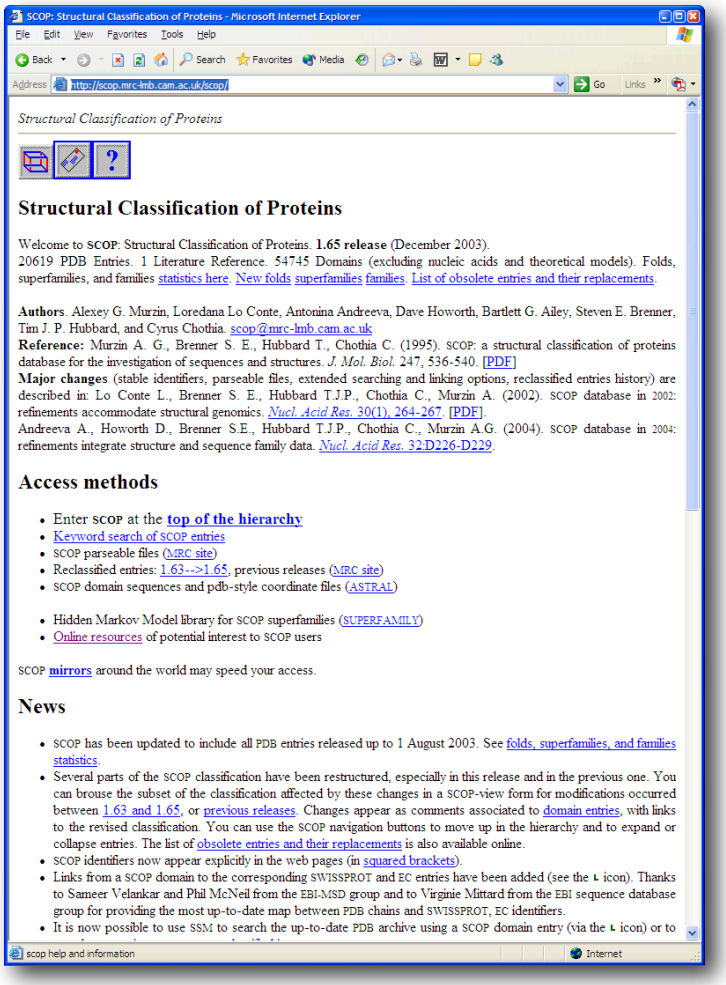
Alex Adai

Adai AT, Date SV, Wieland S, Marcotte EM. J Mol Biol. 2004 Jun 25;340(1):179-90

http://bioinformatics.icmb.utexas.edu/lgl/

SCOP$_{1.65}$ database

http://scop.mrc-lmb.cam.ac.uk/scop/



- ✓ Largely recognized as “standard of gold”
- ✓ Manually classification
- ✓ Clear classification of structures in:
  - CLASS
  - FOLD
  - SUPER-FAMILY
  - FAMILY
- ✓ Some large number of tools already available

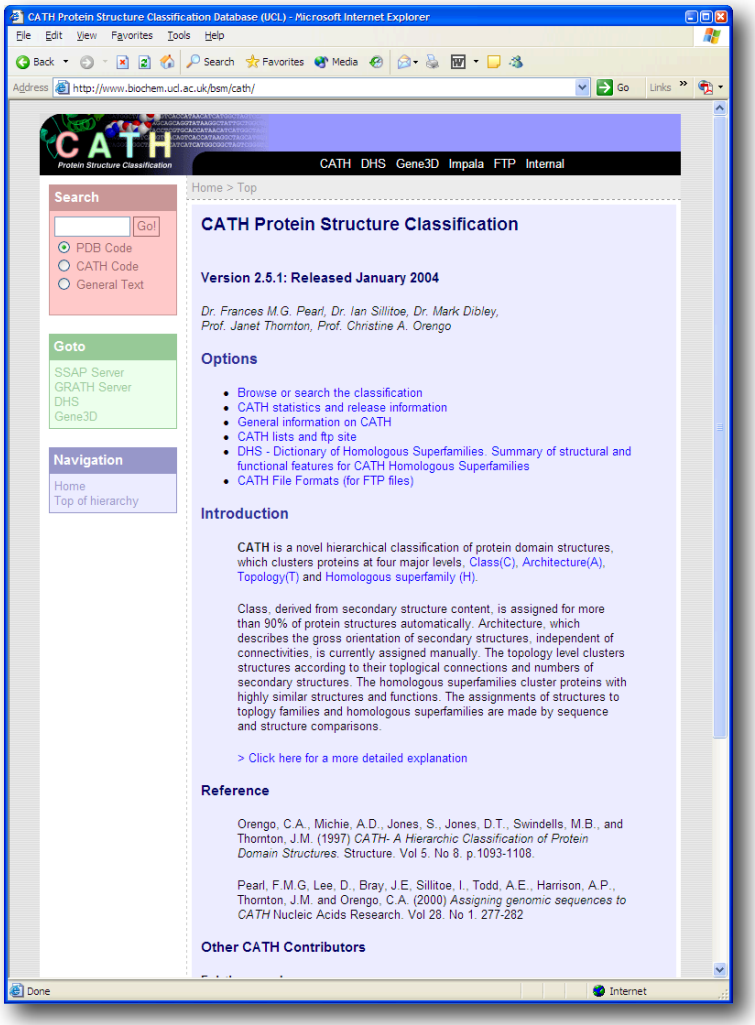
Manually classification
Not 100% up-to-date
Domain boundaries definition

Class	Number of folds	Number of superfamilies	Number of families
All alpha proteins	218	376	608
All beta proteins	144	290	560
Alpha and beta proteins (a/b)	136	222	629
Alpha and beta proteins (a+b)	279	409	717
Multi-domain proteins	46	46	61
Membrane and cell surface	47	88	99
Small proteins	75	108	171
Total	945	1539	2845

Murzin A. G.,el at. (1995). *J. Mol. Biol.* **247**, 536-540.

CATH$_{2.5.1}$ database

http://www.biochem.ucl.ac.uk/bsm/cath/



Uses FSSP for superimposition

- ✓ **Recognized as "standard of gold"**
- ✓ **Semi-automatic classification**
- ✓ **Clear classification of structures in:**
  - **CLASS**
  - **ARCHITECTURE**
  - **TOPOLOGY**
  - **HOMOLOGOUS SUPERFAMILIES**
- ✓ **Some large number of tools already available**
- ✓ **Easy to navigate**

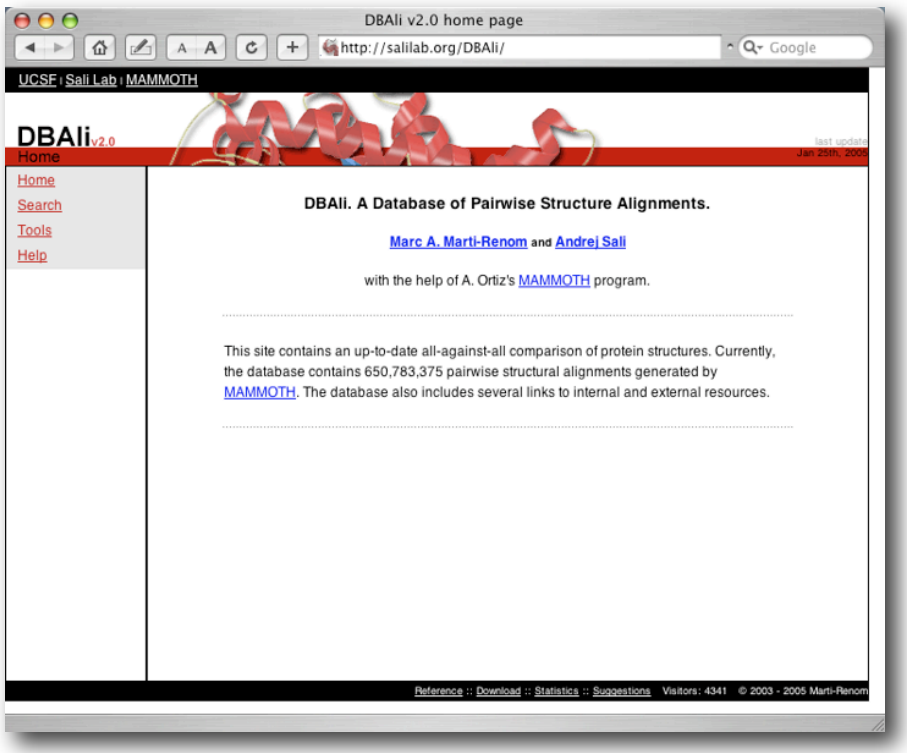
Semi-automatic classification
Domain boundaries definition

C	A	T	H	S	N	I	D
Mainly Alpha	5	251	465	1402	2189	3705	14105
Mainly Beta	19	160	311	1443	2961	4329	18771
Alpha Beta	14	414	706	3014	4781	7660	33080
Few Secondary Structures	1	82	90	144	232	285	1098
Preliminary single domain assigments	10	808	809	906	967	1090	3012
Multi-domain domains	1	12	12	16	25	36	109
CATH-35 Sequence families	1	4707	4707	4719	4768	4862	6168
	1	22	22	27	33	38	198

Orengo, C.A., et al. (1997) *Structure*. **5**. 1093-1108.

DBAli$_{v2.0}$ database

http://salilab.org/DBAli/



Uses MAMMOTH for superimposition

- ✓ **Fully-automatic**
- ✓ **Data is kept up-to-date with PDB releases**
- ✓ **Tools for "on the fly" classification of families.**
- ✓ **Easy to navigate**
- ✓ **Provides some tools for structure comparison**

Does not provide (yet) a stable classification

Pairwise structure alignments

Last update:	January 30th, 2006
Number of chains:	73,817
Number of structure-structure comparisons:*	1,013,210,249

Multiple structure alignments

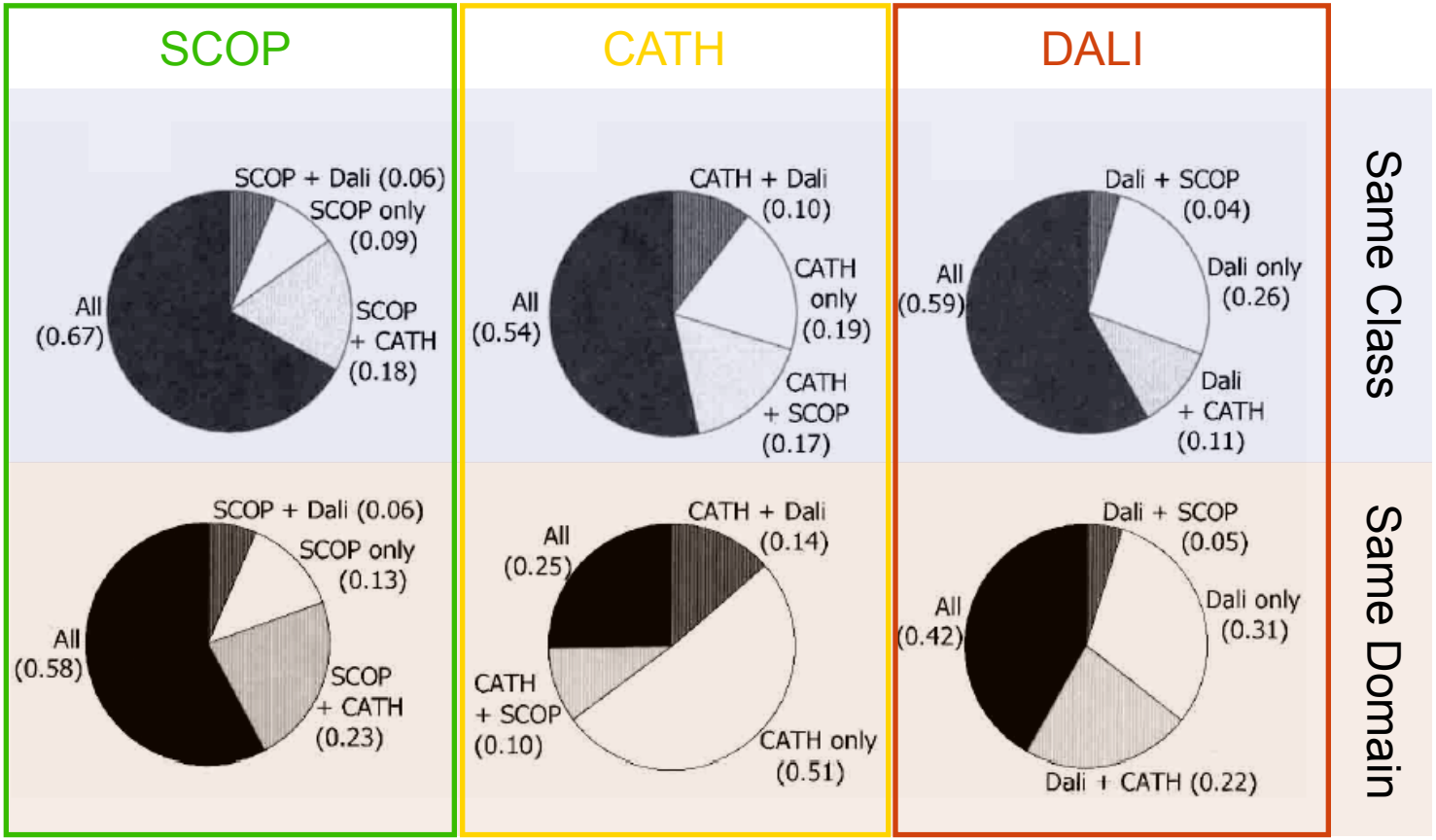
Last update:	November 16th, 2005
Number of representative chains:	24,828
Number of families:	9,588

Marti-Renom et al. 2001. *Bioinformatics*. **17**, 746

Classification of the structural space

Not an easy task!

Domain definition AND domain classification



Day, et al. (2003) *Protein Sciences*, **12** pp2150

Sequence-Structure comparison

- Outline
  - Before we start…
    - Some theory…
    - Domain boundaries
  - Structural predictions from sequence…
    - SALIGN (gap penalties and substitution matrices)
    - mGenThreader (SSE prediction and alignment/potential scores)
    - Fugue (gap penalties and substitution matrices)
    - 3D-Jury (as a meta server example)

General overview (Threading)

- Matches sequences to 3D structures
  - Requires a scoring function to asses the fit of a sequence to a given fold
  - Scoring functions derived from known structures and include atom contact and solvation terms evaluated in a pairwise fashion
  - May include secondary structure terms, multiple alignments…

- Threading servers available using several different approaches
  - Fold recognition server at Imperial College, UK
    - http://www.sbg.bio.ic.ac.uk/~3dpssm/
  - PredictProtein server at EMBL
    - http://www.embl-heidelberg.de/predictprotein/predictprotein.html
  - Protein sequence-structure threading at NCBI
    - http://www.ncbi.nlm.nih.gov/Structure/RESEARCH/threading.shtml

Template comparison methods

- Uses 3D “templates” for searching structural databases
  - active site or binding site templates generated to reflect functionally
  - important structural signatures
- Available software/servers
  - Template Search and Superposition (TESS), Thornton Group
    - http://www.biochem.ucl.ac.uk/bsm/PROCAT/PROCAT.html
    - Wallace AC; Borkakoti N; Thornton JM. (1997) *Protein Science* **6** pp2308
  - “Fuzzy Functional Forms” , Skolnick - commercial availability
    - Fetrow, Js and Skolnick, J (1998) *J. Mo. Biol* **281** pp949
  - Spatial Arrangements of Side-chain and Main-chain (SPASM),
    - http://portray.bmc.uu.se/cgi-bin/dennis/spasm.pl

      Kleywegt GJ (1999). *J. Mol. Biol.* **285** pp1887

Empirical energy functions (PMF)

Idea: **energy leads to structure, thus it should be possible to infer energy from many known structures**

To be used in: **model refinement and assessment**

Properties needed:

- Deep minimum at correct state (native)
- Smooth (energy landscape)
- Simple (CPU calculation)

Types:

- Contact potential
- Distance potentials
- Surface potentials

Approximations/Limitations in PMFs

Database size.

PMF versus Energy (additive/higher order terms).

Reference state.

Physical origin.

Finkelstein et al. (1995) *Proteins* **23**, pp142

Sequence-Structure alignments

As any other bioinformatics problem…

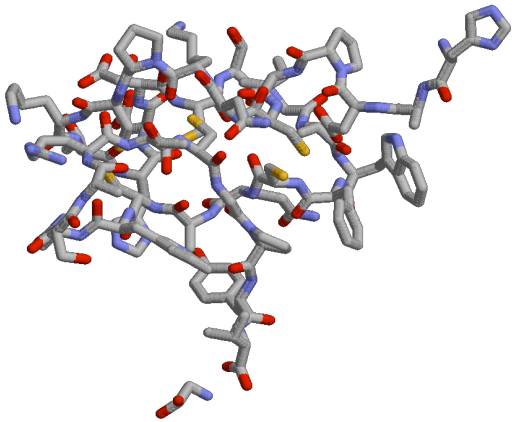
- **Representation**
- **Scoring**
- **Optimizer**

Representation

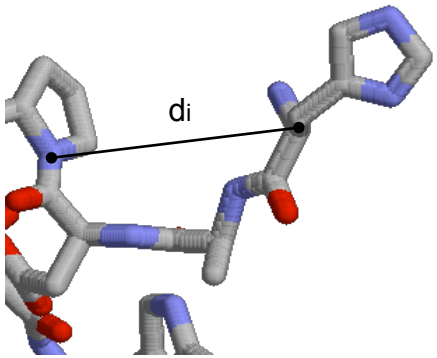
Sequence/Structures

```
>gi42541361
MDIRSVSSLRGLLCLPPSWPRR
```

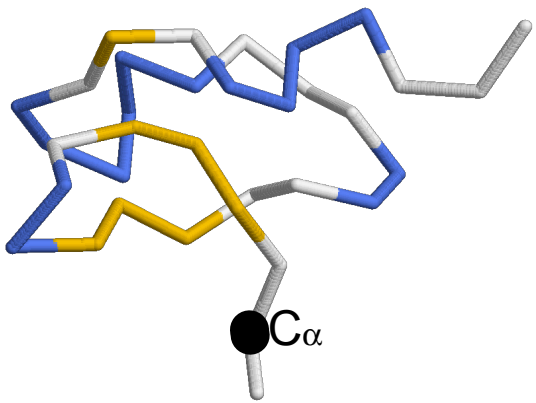
Primary sequence



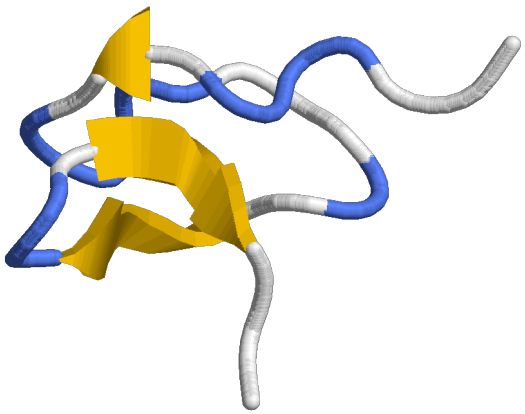
All atoms and coordinates



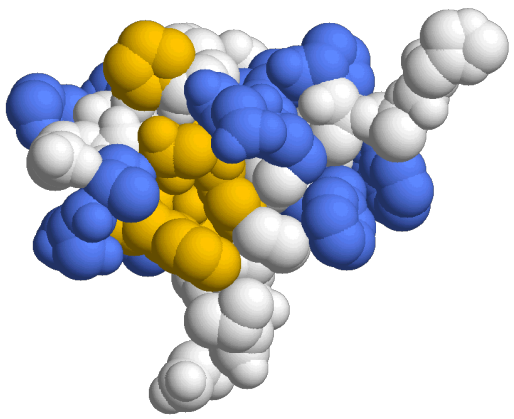
Distance space



Reduced atoms representation



Secondary Structure

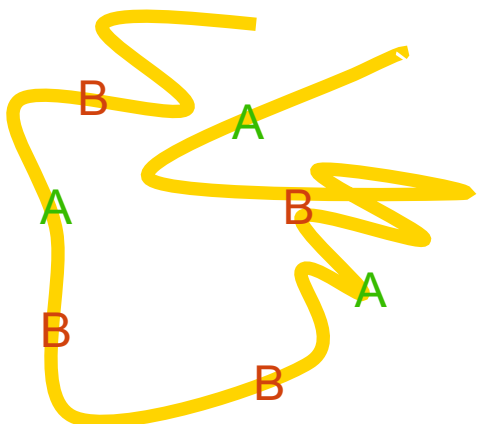


Accessible surface

Statistical Potential (inspiration)

$$K = \frac{[AB]}{[A]\cdot[B]}$$

$$\Delta G = -RT\ln(K) = -RT\ln\frac{[AB]}{[A]\cdot[B]}$$



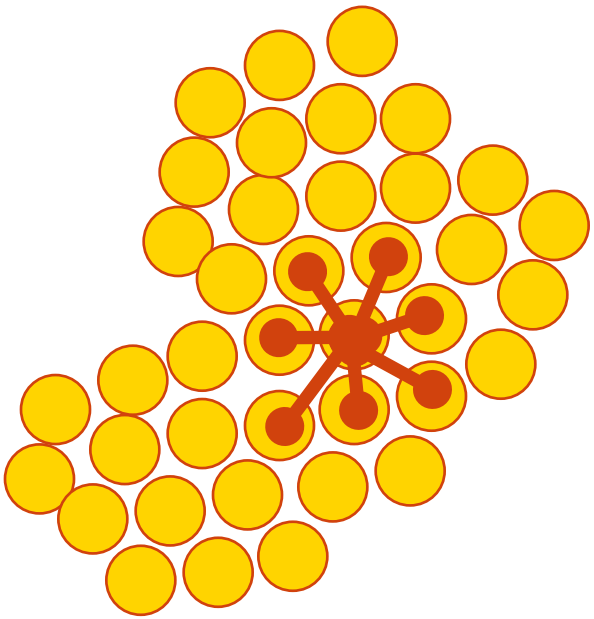
From statistical physics, we know that energy difference between two states (ΔE) and the ratio of their occupancies (N_1:N_2) are related [9]:

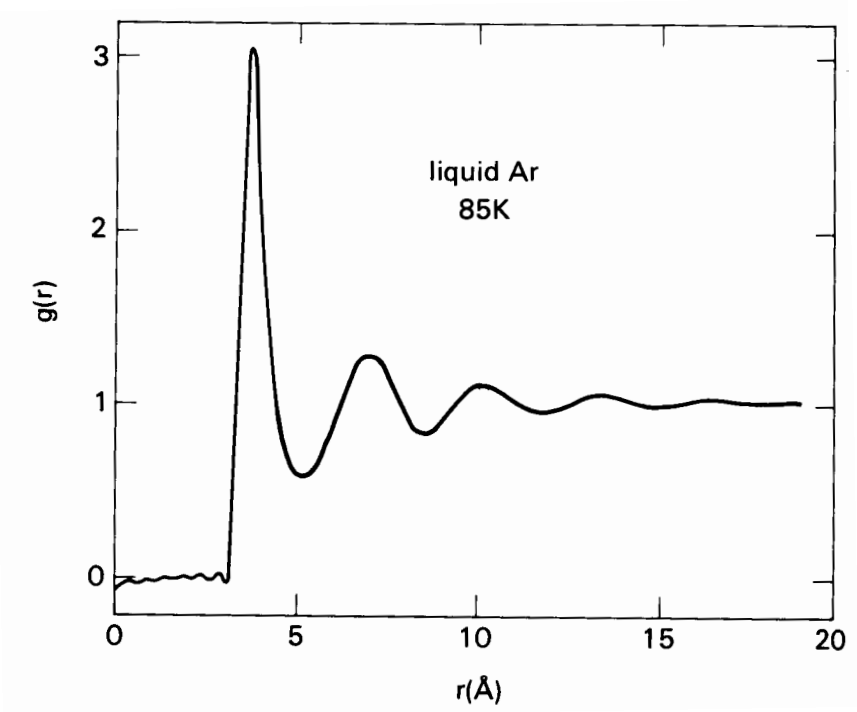
$$\Delta E = -kT \ln\left(\frac{N_1}{N_2}\right) \qquad (1)$$

in which T is the absolute temperature and k is the Boltzmann's constant. As we are interested in an interaction energy between two amino acid side chains, it would seem natural to define N_1 as the number of interactions between these two residues types in a group of real protein structures, a number which is readily available from simple database analysis. But this number must be compared with the number of interactions in some other system, N_2, to obtain the energy difference between them.

Tanaka and Sheraga (1975) *PNAS*, **72** pp3802
Sippl, (1990) J.Mo.Biol. **213** pp859
Godzik, (1996) *Structure* **15** pp363

Statistical Potential (reference state)

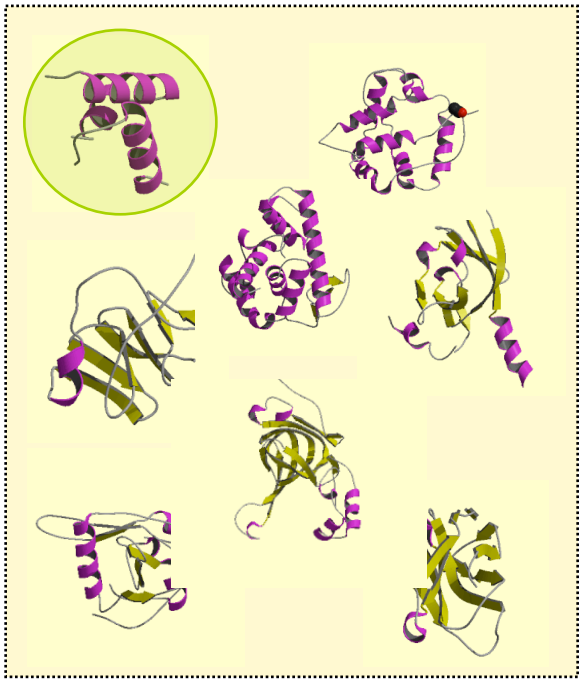




Theory of simple liquids 2nd edition JP Hansen and IR McDonald, Academic Press.

Statistical Potentials (background)

Structural space



Sequence space

MKLLIVLTCISLCSCICTVVQRCASNKPHVLEDPCKVQH

HLSVNQCVLLPQCCPKSCKICTHLISIEVVLTCRAVDKM

MHVNCVEQCSLQDCIKIAPRVLKTCILCVLKPCLTSVSH

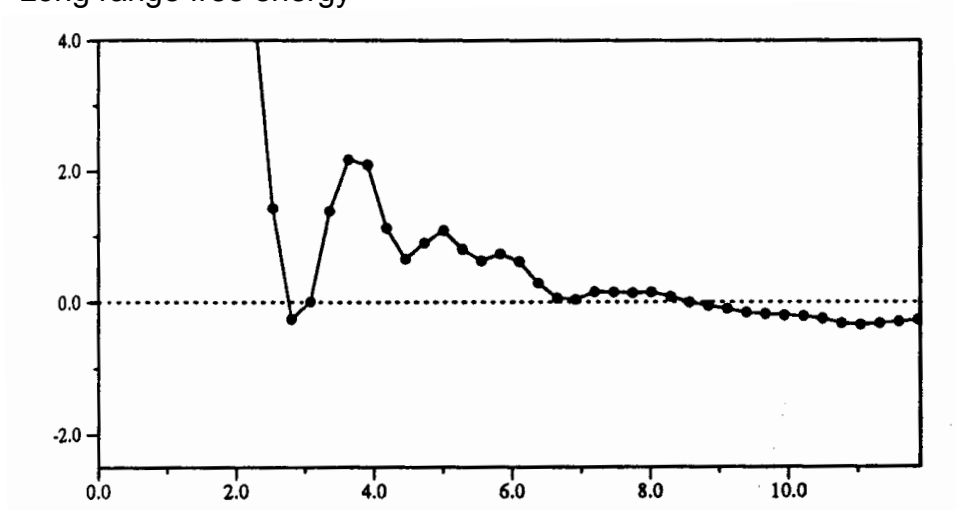
VHLVQPTSCCCKKNCICHVEIRSLDILTKSVQLACLVPM

⋮

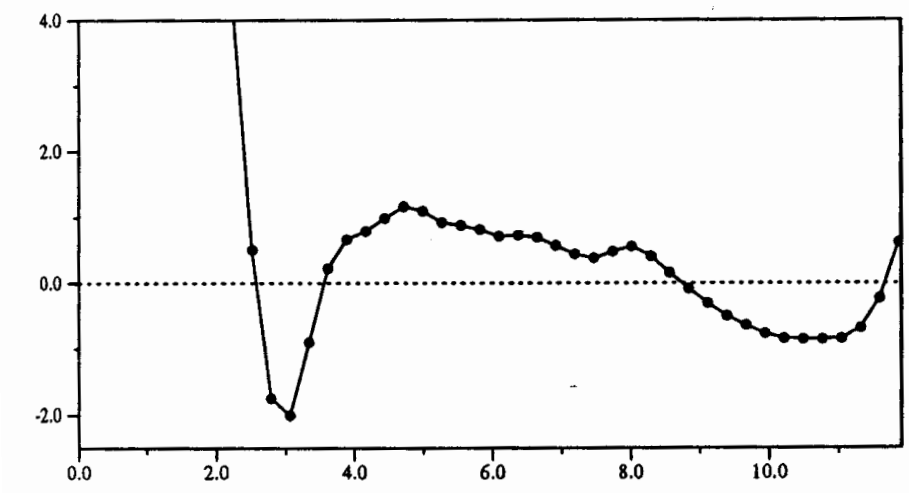
MQCCRVQKICDLLAVELCKLHISTPSCKILCVVTSVPHN

Statistical Potential… Hydrogen Bonds

Long range free energy



Short range free energy



Free energy of the protein backbone hydrogen bond N · · · O compiled from a database of 289 X-ray structures

$$\rho_{NO}(r) = \sum_{ij} \delta(r - r_{ij})$$

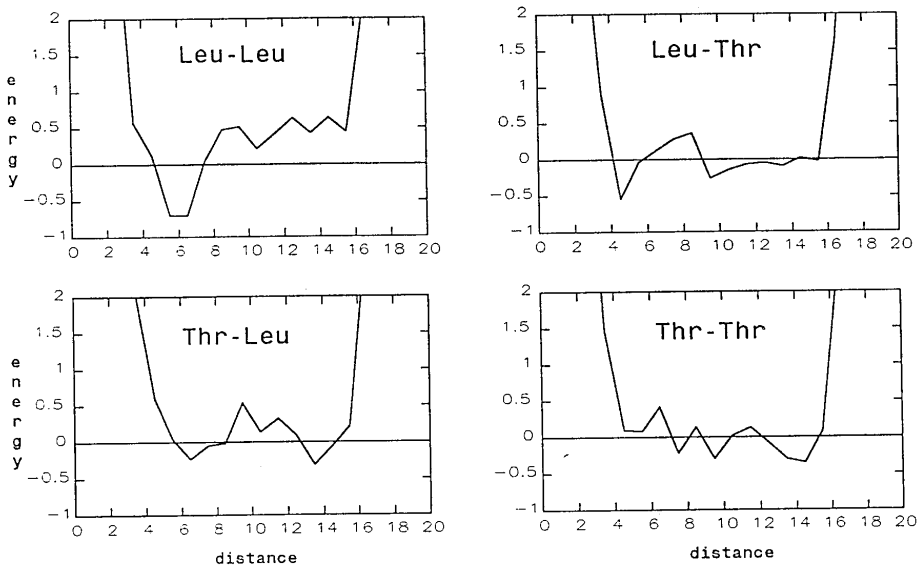
$$g_{NO}(r) = \frac{\rho_{NO}(r)}{\rho^2}$$

$$w_{NO}(r) = -kT \ln\left(g_{NO}(r)\right)$$

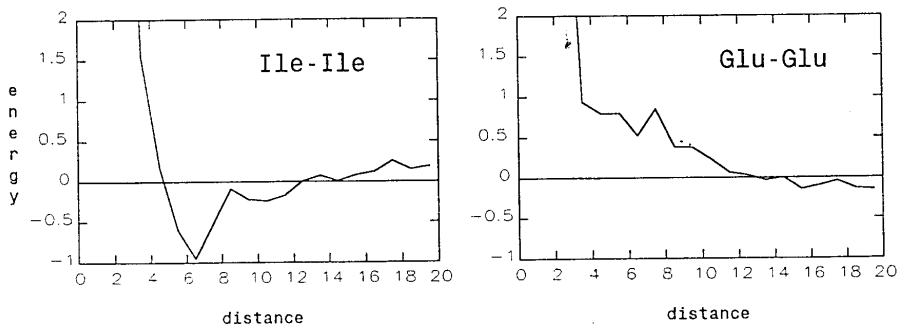
Sippl (1996). *JMB* **260** pp644

Statistical Potential… Distance Potentials

Long range free energy



Short range free energy



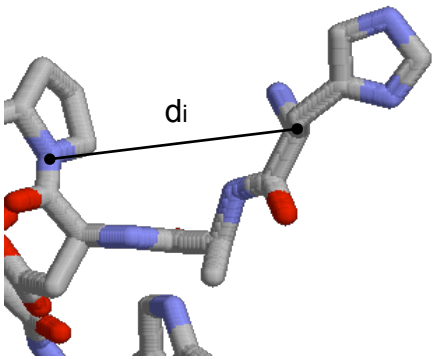
Sippl (1993). *JCAM* **7** pp473

Scoring

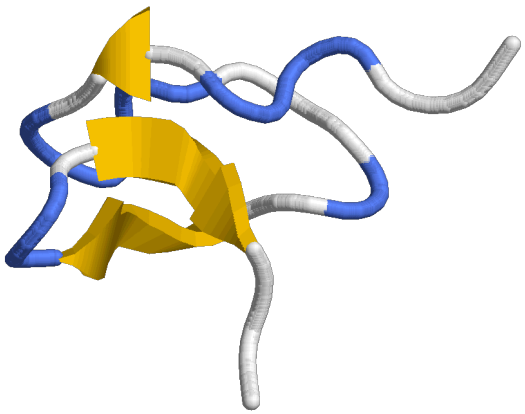
Raw scores of an alignment

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	**4**	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	**4**	1	-1	1	0	1	0	0	0	-1	0	-1	-2	-2	-2	-2	-2	-3
P	-3	-1	1	**7**	-1	-2	-1	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	-1	-1	**4**	0	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-2	-3
G	-3	0	1	-2	0	**6**	-2	-1	-2	-2	-2	-2	-2	-3	-4	-4	0	-3	-3	-2
N	-3	1	0	-2	-2	0	**6**	1	0	0	-1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	1	-1	-2	-1	1	**6**	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	0	-1	-1	-2	0	2	**5**	2	0	0	1	-2	-3	-3	-3	-3	-2	-3
Q	-3	0	0	-1	-1	-2	0	0	2	**5**	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	0	-2	-2	-2	1	1	0	0	**8**	0	-1	-2	-3	-3	-2	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	0	-1	-1	-2	0	-1	1	1	-1	2	**5**	-1	-3	-2	-3	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**	1	2	-2	0	-1	-1
I	-1	-2	-2	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**	2	1	0	-1	-3
L	-1	-2	-2	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**	3	0	-1	-2
V	-1	-2	-2	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	2
W	-2	-3	-3	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**

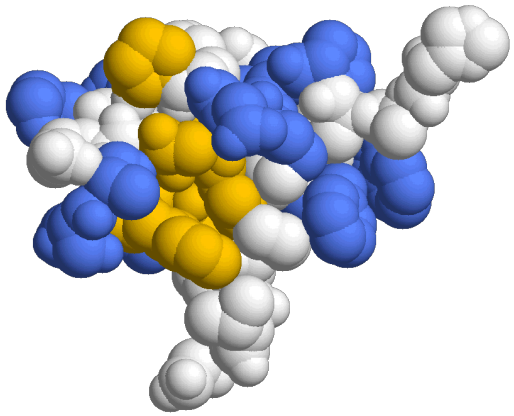
Aminoacid substitutions



Distance space



Secondary Structure (H,B,C)

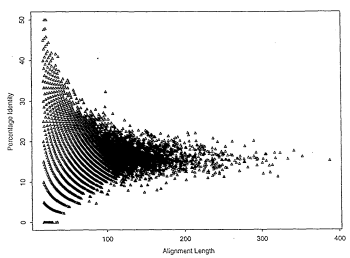


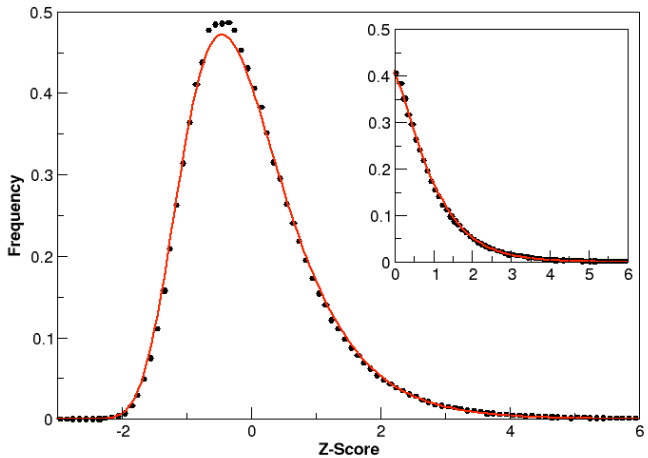
Accessible surface (B,A [%])

Significance of an alignment (score)

Probability that the optimal alignment of two random sequences/structures of the same length and composition as the aligned sequences/structures have at least as good a score as the evaluated alignment.

Empirical





Sometimes approximated by Z-score (normal distribution).

Analytic

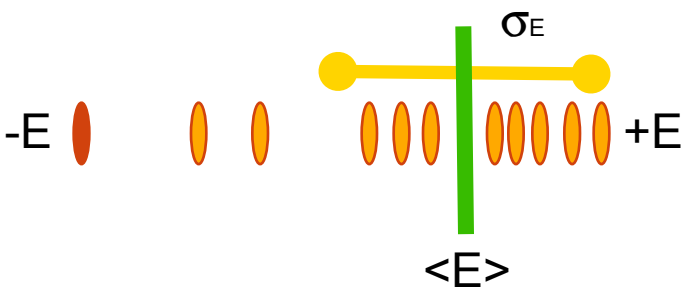
$$P(s) = e^{-\lambda (s-\mu)}$$

$$P(s \geq x) = 1 - \exp\left(e^{-\lambda (s-\mu)}\right)$$

Karlin and Altschul, 1990 *PNAS* **87**, pp2264

Significance of an alignment (score)

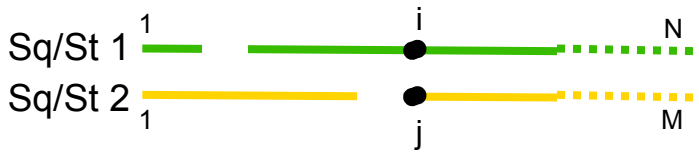
Energy Z-score the model with respect the energy of random models (or rest of decoys).

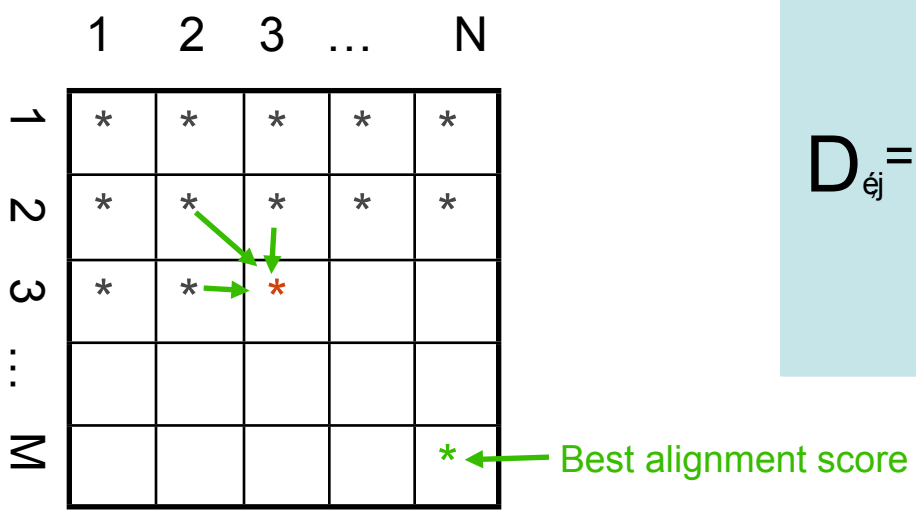


$$Zscore = \frac{(\langle E \rangle - E_m)}{\sigma_E}$$

Global dynamic programming alignment

remember Patsy's class





$$D_{ij} = \min \begin{cases} D_{i,j-1} + \text{Score}_{(Ä,rj)} \\ D_{i-1,j-1} + \text{Score}_{(ri,rj)} \\ D_{i-1,j} + \text{Score}_{(ri,Ä)} \\ 0 \end{cases}$$

Backtracking to get the best alignment

Needleman and Wunsch (1970) *J. Mol Biol*, **3** pp443

Applications of PMFs

Model assessment.

Ab initio folding simulations.

Sequence-structure matching (threading).

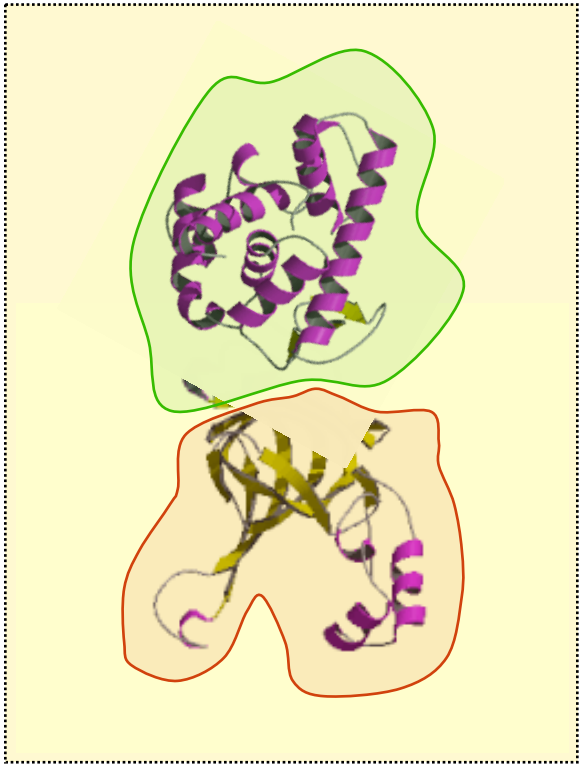
Comparative protein structure modeling (loops, sidechains, …).

Secondary structure prediction, *etc.*

Finkelstein et al. (1995) *Proteins* **23**, pp142

Domain boundaries from sequence

VERY DIFFICULT!!!!



MENFEIWVEKYRPRTLDEVVGQDEVIQRLKGYVERKNIPHLLFSGPPGTGKTATAIALARDLFGENWRDN
FIEMNASDERGIDVVRHKIKEFARTAPIGGAPFKIIFLDEADALTADAQAALRRTMEMYSKSCRFILSCN
YVSRIIEPIQSRCAVFRFKPVPKEAMKKRLLEICEKEGVKITEDGLEALIYISGGDFRKAINALQGAAAI
GEVVDADTIYQITATARPEEMTELIQTALKGNFMEARELLDRLMVEYGMSGEDIVAQLFREIISMPIKDS
LKVQLIDKLGEVDFRLTEGANERIQLDAYLAYLSTLAKK

Domain boundaries from sequence (SnapDragon)

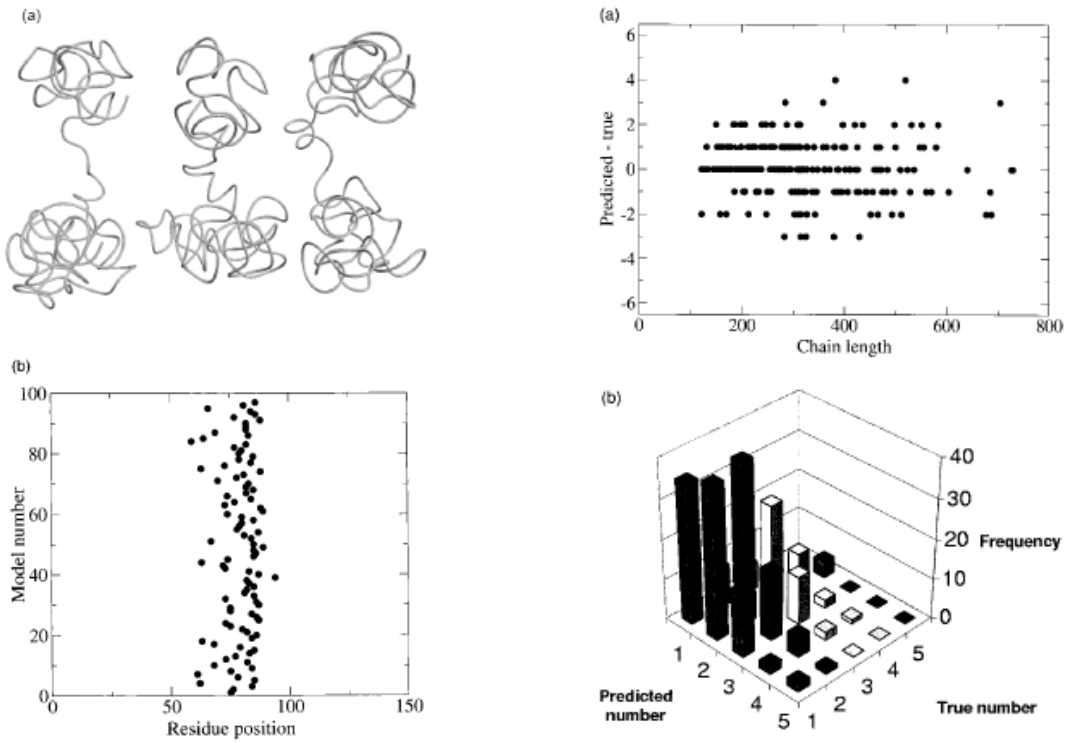


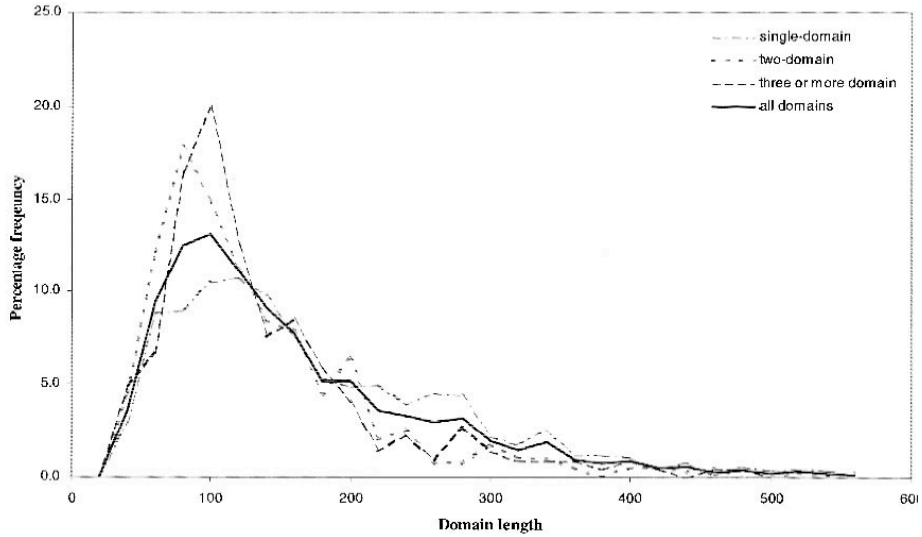
Table 2. Average accuracy percentages of linker prediction over 57 proteins

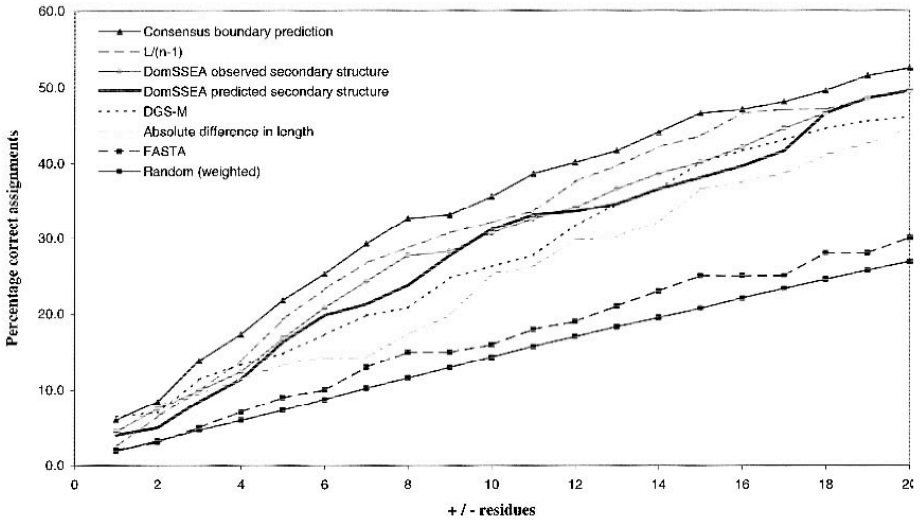
		Continuous set	Discontinuous set	Full set
Randomised background Z-score >2	Coverage	63.3	43.6	54.8
	Success	27.2	31.1	28.9
Self-normalised Z-score >1	Coverage	64.7	39.5	53.5
	Success	26.6	31.7	28.9
Self-normalised Z-score >2	Coverage	48.7	24.3	38.7
	Success	41.3	28.3	29.9

Georgea and Heringa (2002) *J. Mol. Biol.* **316** pp839

Domain boundaries from sequence and predicted SSE (DomSSEA)

Methods	% Correctly assigned	
	All chains	Multidomain chains
DomSSEA observed secondary structure	70.2	24.7
DomSSEA predicted & consensus	68.6	24.0
DomSSEA predicted & L/(N-1)	68.0	24.0
DomSSEA predicted secondary structure	68.7	23.6
Absolute difference in length	62.0	8.4
Average domain length & DGS-M	66.6	6.1
FASTA alignment	57.9	2.3
Random (weighted)	58.3	1.1
DGS-M	76.6	0.0
DGS-W	76.6	0.0



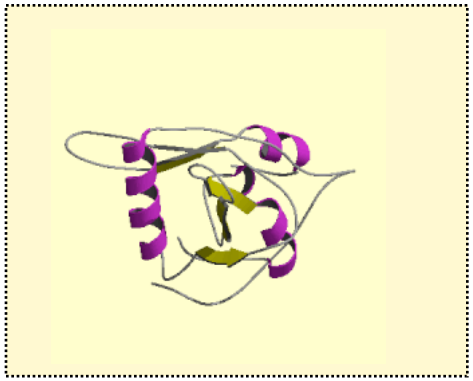


Dersden *et al.* (2003) *Prot. Science* **11** pp2014

Prediction of Secondary Structure (PSI-PRED)

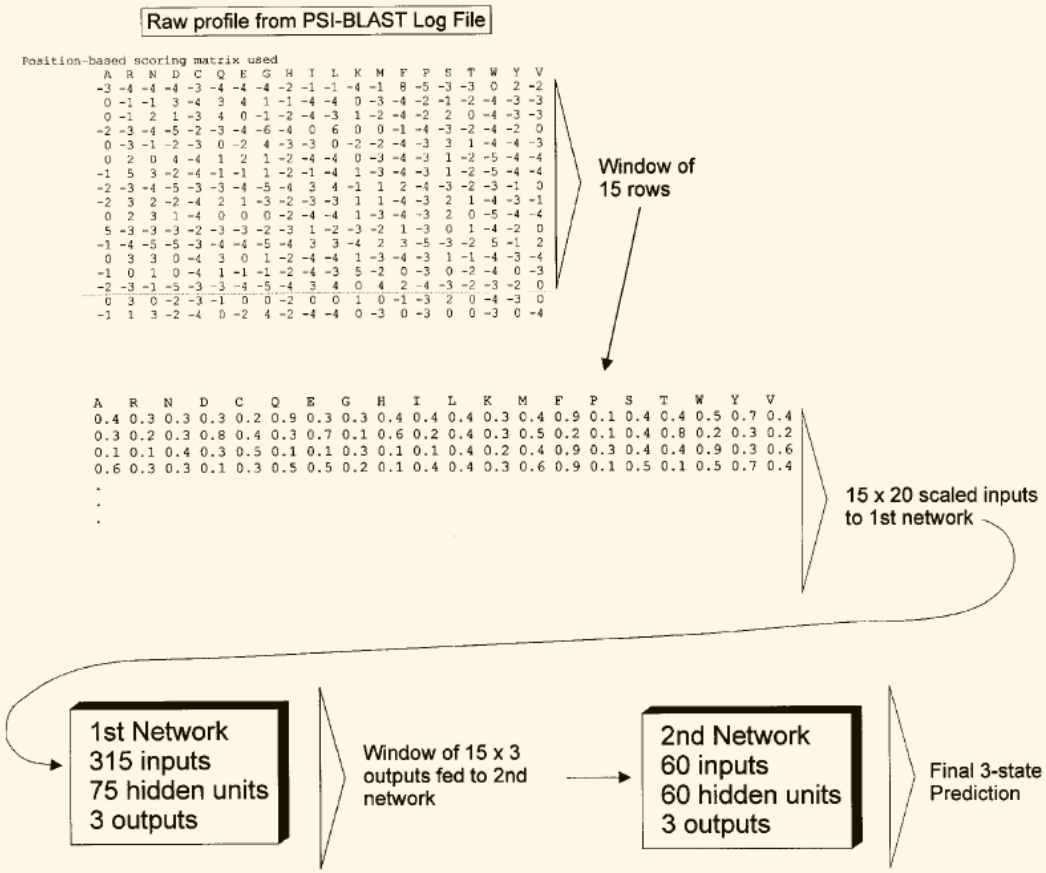
```
>gi42541361
MDIRSVSSLRGLLCLPPSWPRR
```

- Neural Network



✓ **Very simple idea**
✓ **Simple scoring**

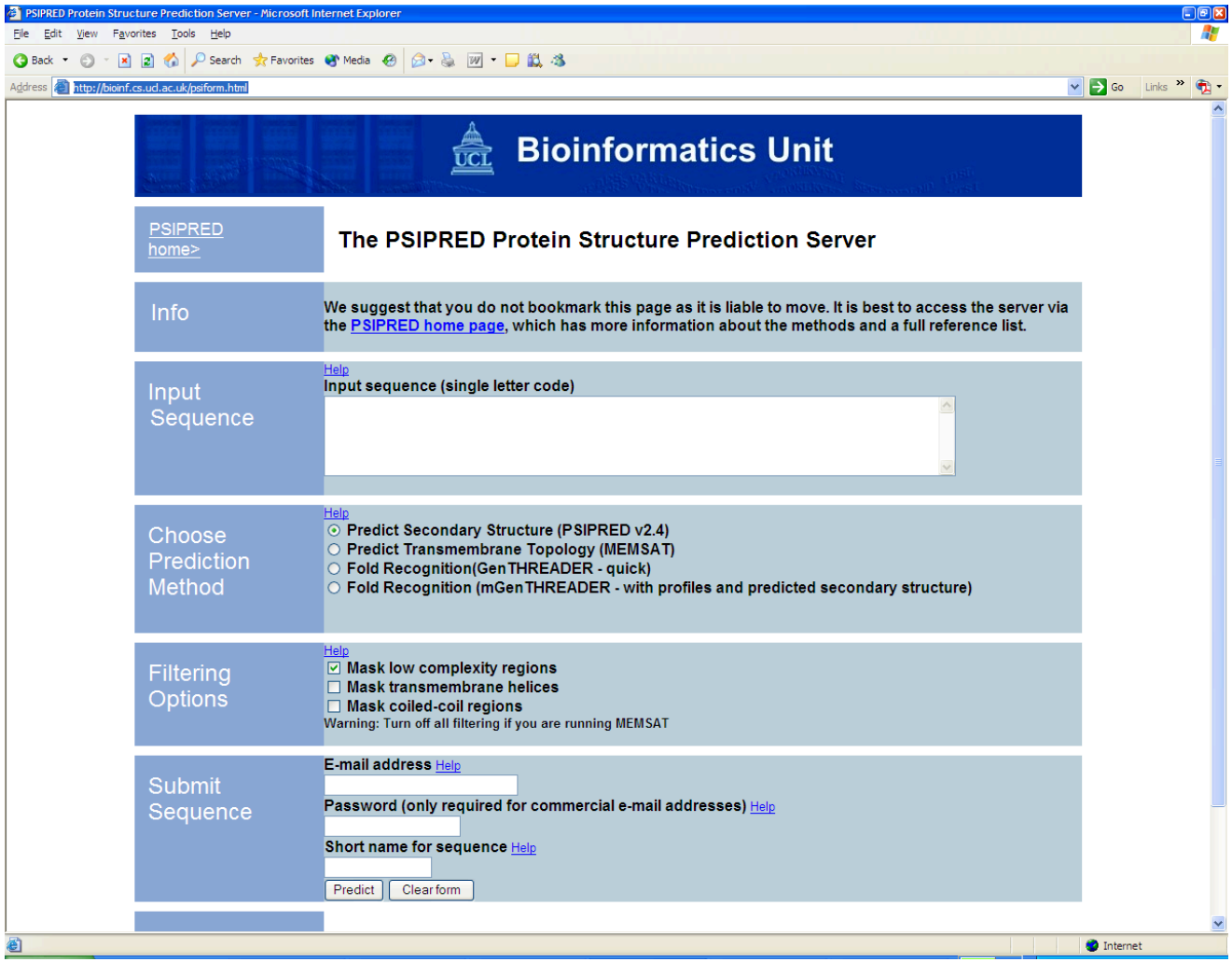
Obscure optimizer



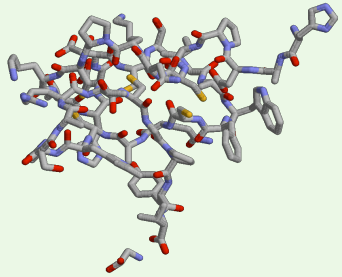
Jones DT. (1999) *J. Mol. Biol.* **292** pp195

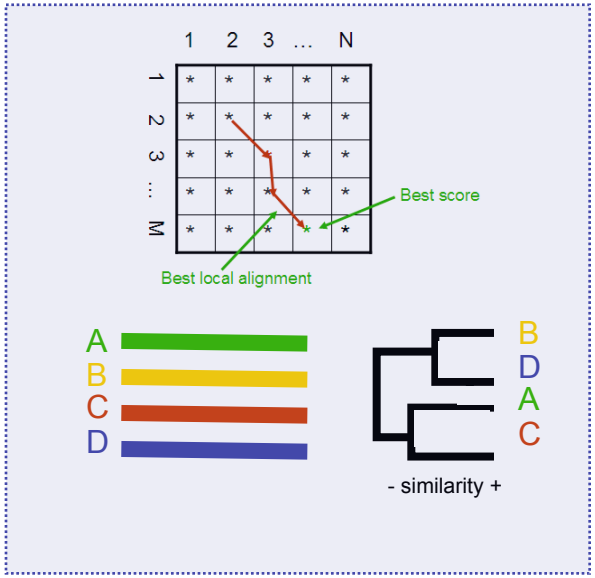
Prediction of Secondary Structure (PSI-PRED)

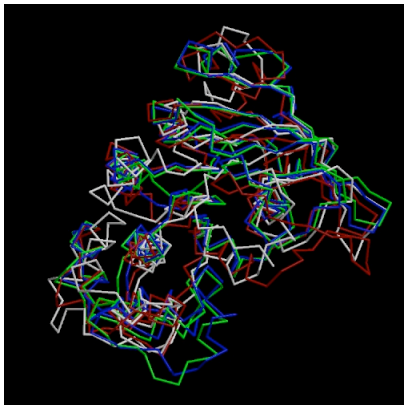
http://bioinf.cs.ucl.ac.uk/psiform.html



Sequence-Structure alignment by properties conservation (SALIGN-MODELLER)

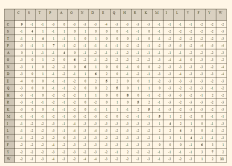




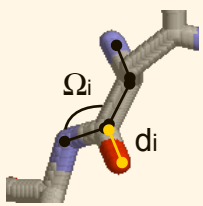


✓ Uses all available structural information
✓ Provides the optimal alignment

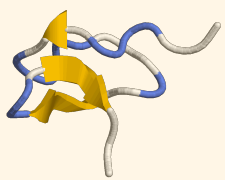
Computationally expensive



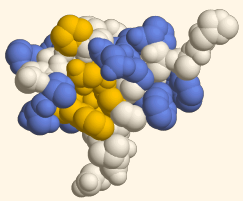
$R_{i,j}$



$D_{i(3),j(3)}$



$S_{i,j}$



$B_{i,j}$

$$RMSD = \sqrt{\sum (x_i - \bar{x})^2}$$

$I_{i,j}$

$$Score_{i,j} = w_1 * R_{i,j} + w_2 * D_{i(a),j(a)} + w_3 * S_{i,j} + w_4 * B_{i,j} + w_5 * I_{i,j} + w_6 * X_{i,j}$$

Madhusudhan et al. *in preparation*

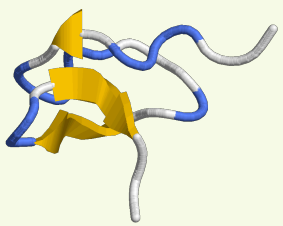
Sequence-Structure alignment by properties conservation (SALIGN-MODELLER)

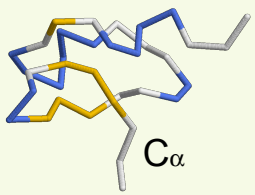
http://alto.compbio.ucsf.edu/salign-cgi/index.cgi

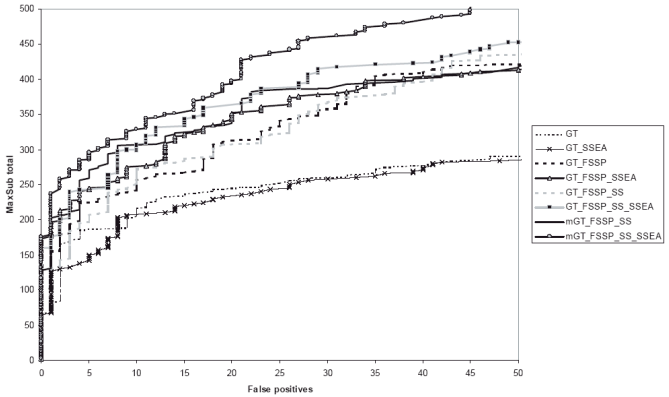
Threading (mGenThreader)

```
>gi42541361
MDIRSVSSLRGLLCLPPSWPRR
```

- Neural Network

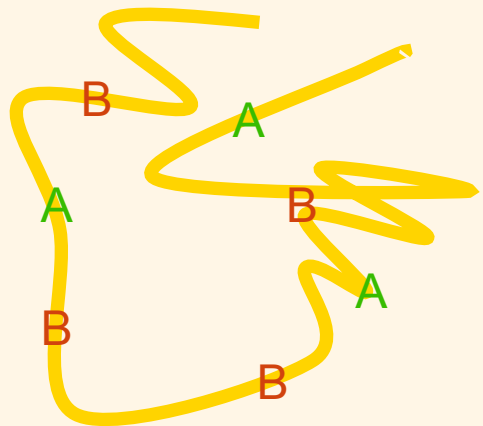


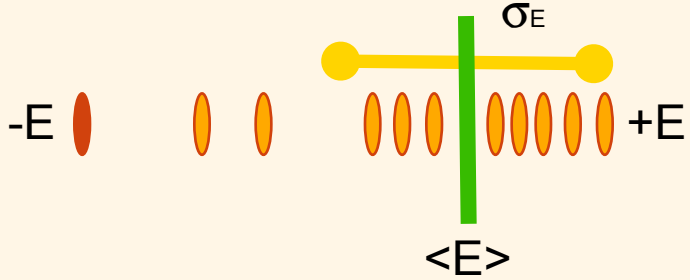




✓ **Good row and significance scoring**

Obscure optimizer



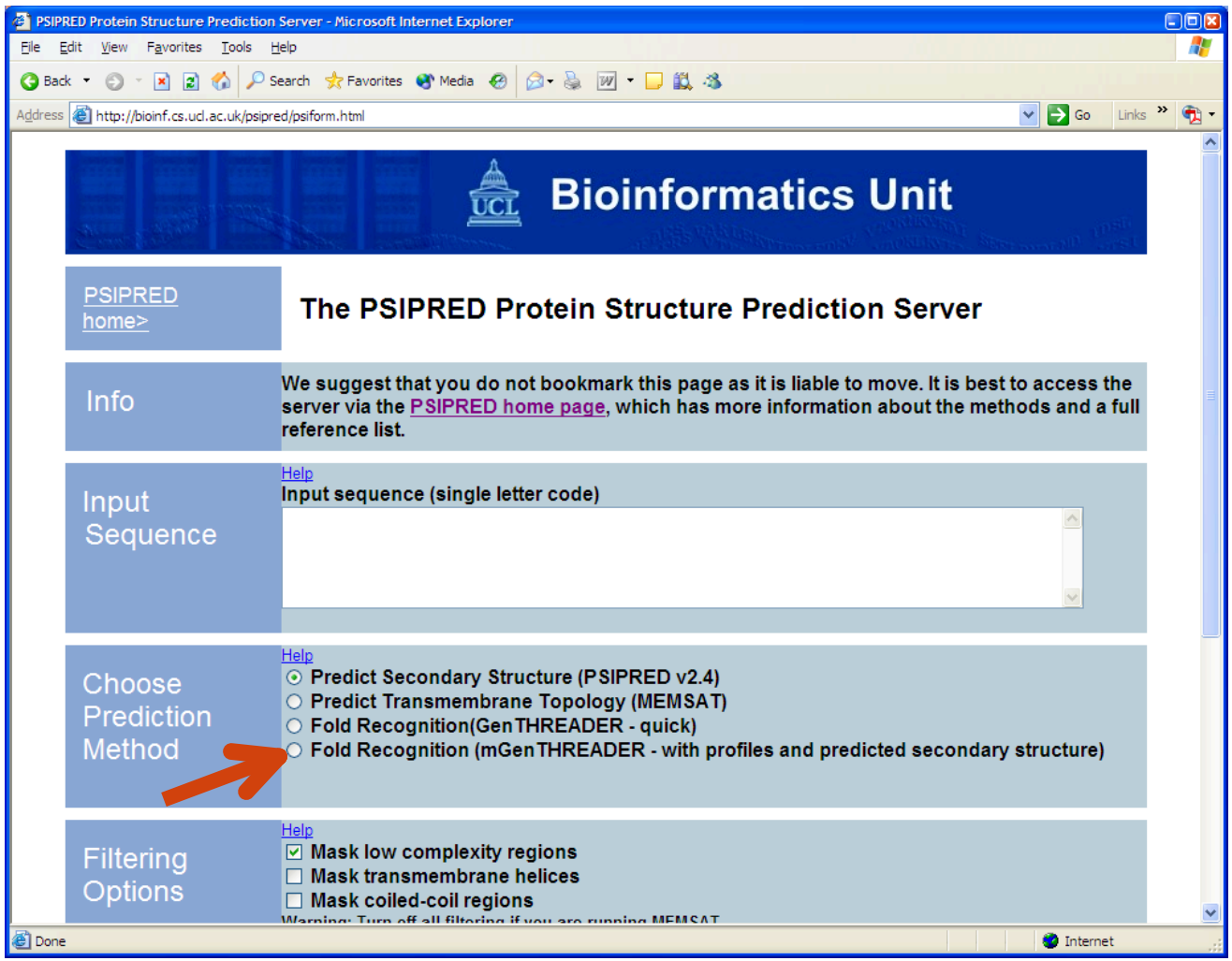


$$Zscore = \frac{(\langle E \rangle - E_m)}{\sigma_E}$$

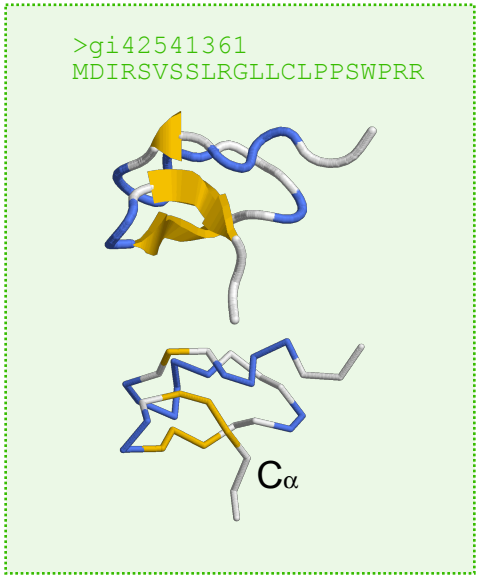
McGuffin LJ, Jones DT. (2003) *Bioinformatics*, **19**, pp874

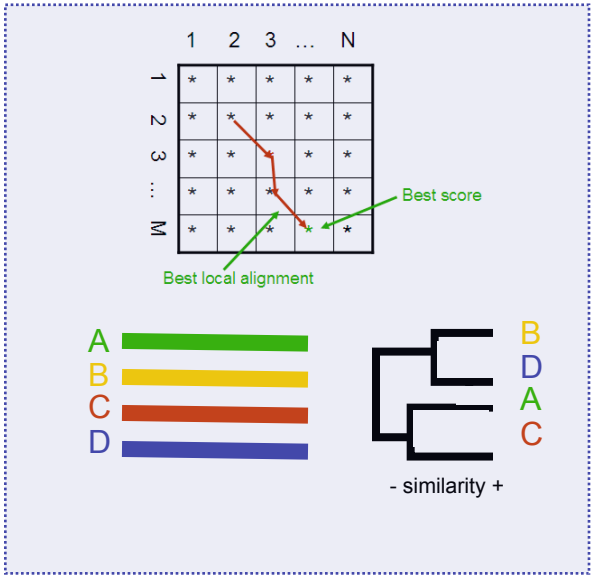
Threading (mGenThreader)

http://bioinf.cs.ucl.ac.uk/psiform.html



Remote homology detection (FUGUE)

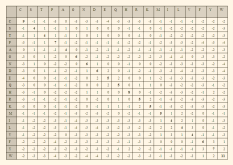




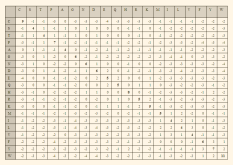
FUGUE

- ✓ Uses most of the structural information
- ✓ Easy to access either locally and on the web
- ✓ Good row and significance scoring

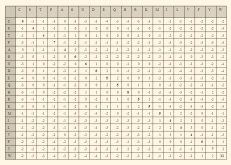
Does not uses multiple sequence information



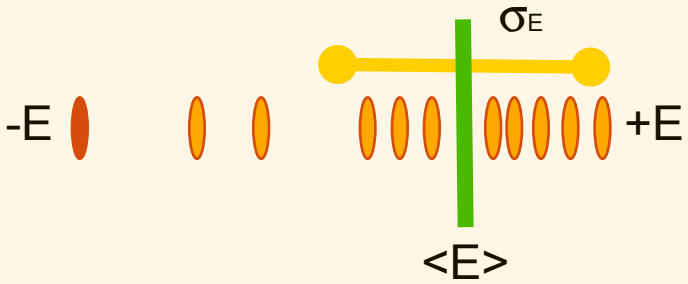
$R^{H}_{i,j}$



$R^{B}_{i,j}$



$R^{C}_{i,j}$

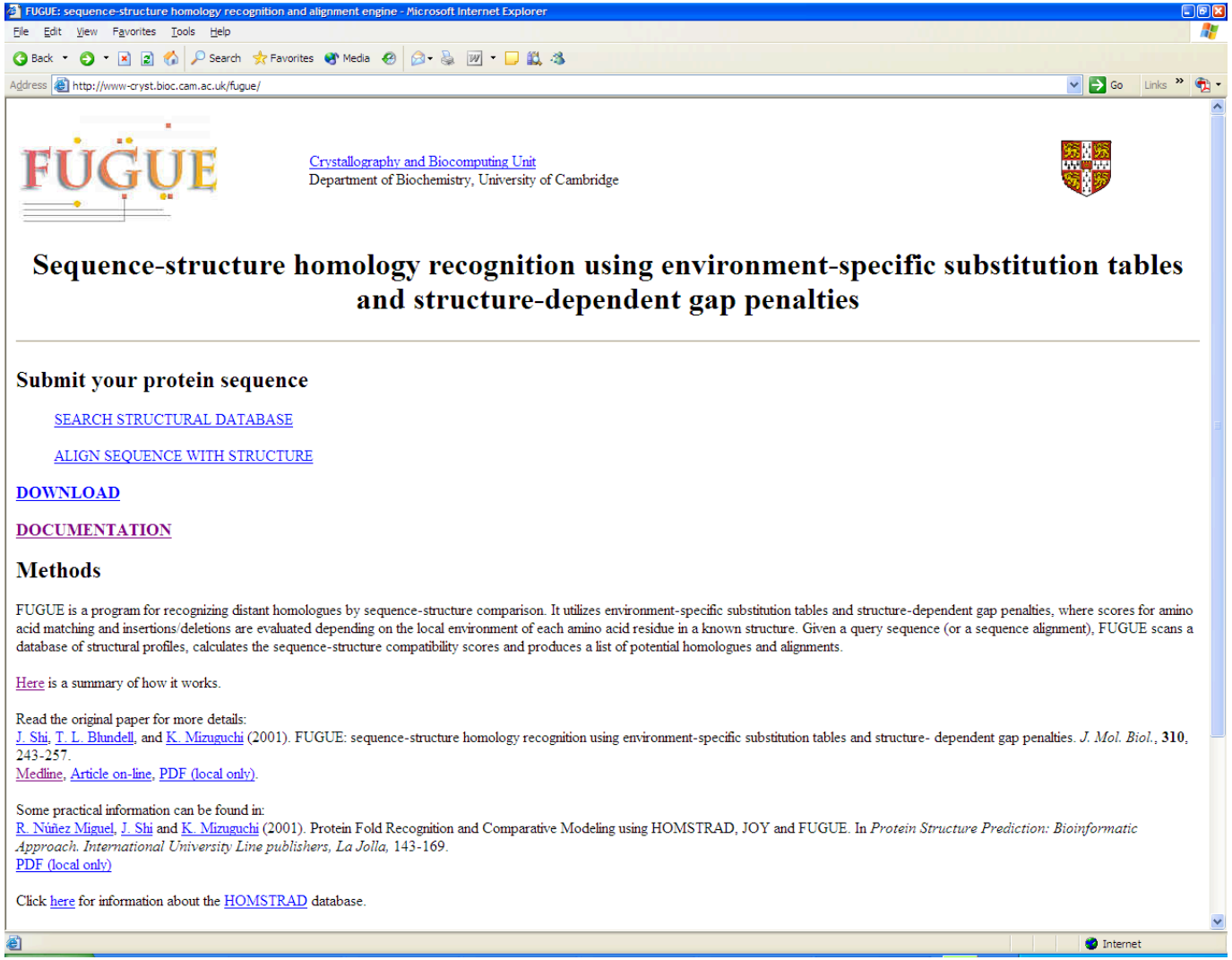


$$Zscore = \frac{(\langle E \rangle - E_m)}{\sigma_E}$$

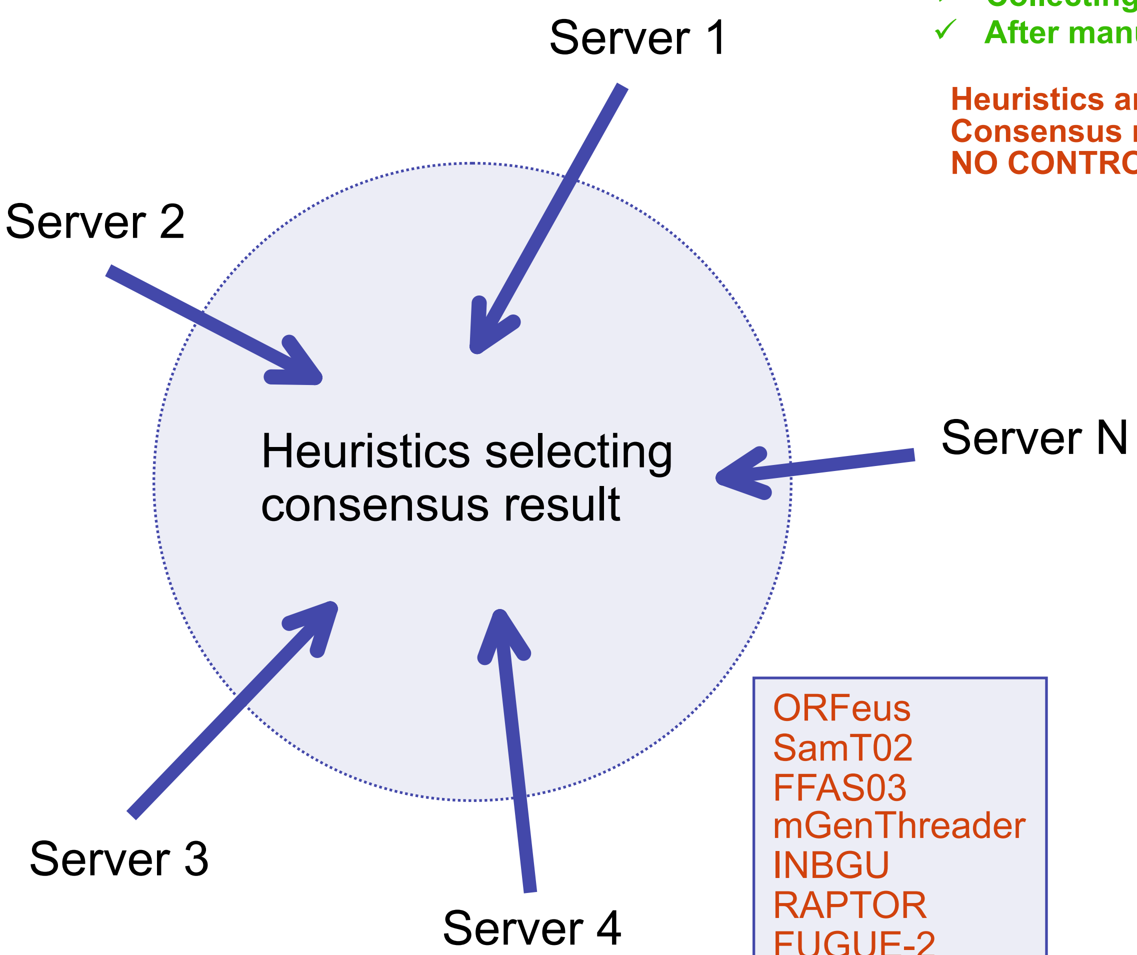
Shi et al. (2001) *J. Mol. Biol* **310** pp241

Remote homology detection (FUGUE)

http://www-cryst.bioc.cam.ac.uk/fugue/



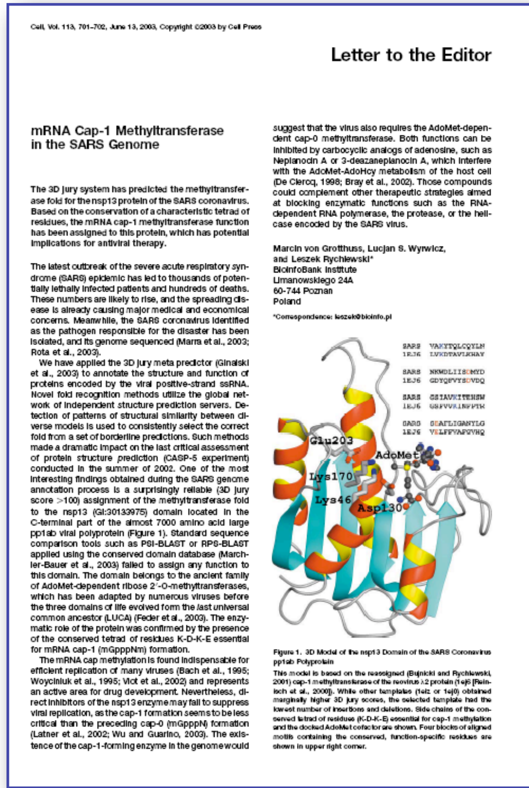
Meta-Servers (3D-Jury)



- ✓ Collecting several results
- ✓ After manual analysis… good results

Heuristics and complicated scoring
Consensus results
NO CONTROL OF DATA GENERATION or SERVERS!

ORFeus
SamT02
FFAS03
mGenThreader
INBGU
RAPTOR
FUGUE-2
3D-PSSM



Cell, Vol. 113, 701–702, June 13, 2003, Copyright ©2003 by Cell Press

Letter to the Editor

mRNA Cap-1 Methyltransferase in the SARS Genome

The 3D jury system has predicted the methyltransferase fold for the nsp13 protein of the SARS coronavirus. Based on the conservation of a characteristic tetrad of residues, the mRNA cap-1 methyltransferase function has been assigned to this protein, which has potential implications for antiviral therapy.

The latest outbreak of the severe acute respiratory syndrome (SARS) epidemic has led to thousands of potentially lethally infected patients and hundreds of deaths. These numbers are likely to rise, and the spreading disease is already causing major medical and economical concerns. Meanwhile, the SARS coronavirus identified as the pathogen responsible for the disaster has been isolated, and its genome sequenced (Marra et al., 2003; Rota et al., 2003).

We have applied the 3D jury meta predictor (Ginalski et al., 2003) to annotate the structure and function of proteins encoded by the viral positive-strand ssRNA. Novel fold recognition methods utilize the global network of independent structure prediction servers. Detection of patterns of structural similarity between diverse models is used to consistently select the correct fold from a set of borderline predictions. Such methods made a dramatic impact on the last critical assessment of protein structure prediction (CASP-5 experiment) conducted in the summer of 2002. One of the most interesting findings obtained during the SARS genome annotation process is a surprisingly reliable (3D jury score >100) assignment of the methyltransferase fold to the nsp13 (GI:30133975) domain located in the C-terminal part of the almost 7000 amino acid large pp1ab viral polyprotein (Figure 1). Standard sequence comparison tools such as PSI-BLAST or RPS-BLAST applied using the conserved domain database (Marchler-Bauer et al., 2003) failed to assign any function to this domain. The domain belongs to the ancient family of AdoMet-dependent ribose 2′-O-methyltransferases, which has been adapted by numerous viruses before the three domains of life evolved form the last universal common ancestor (LUCA) (Feder et al., 2003). The enzymatic role of the protein was confirmed by the presence of the conserved tetrad of residues K-D-K-E essential for mRNA cap-1 (mGpppNm) formation.

The mRNA cap methylation is found indispensable for efficient replication of many viruses (Bach et al., 1995; Woyciniuk et al., 1995; Vlot et al., 2002) and represents an active area for drug development. Nevertheless, direct inhibitors of the nsp13 enzyme may fail to suppress viral replication, as the cap-1 formation seems to be less critical than the preceding cap-0 (mGpppN) formation (Latner et al., 2002; Wu and Guarino, 2003). The existence of the cap-1-forming enzyme in the genome would suggest that the virus also requires the AdoMet-dependent cap-0 methyltransferase. Both functions can be inhibited by carbocyclic analogs of adenosine, such as Neplanocin A or 3-deazaneplanocin A, which interfere with the AdoMet-AdoHcy metabolism of the host cell (De Clercq, 1998; Bray et al., 2002). Those compounds could complement other therapeutic strategies aimed at blocking enzymatic functions such as the RNA-dependent RNA polymerase, the protease, or the helicase encoded by the SARS virus.

Marcin von Grotthuss, Lucjan S. Wyrwicz, and Leszek Rychlewski*
BioinfoBank Institute
Limanowskiego 24A
60-744 Poznan
Poland

*Correspondence: leszek@bioinfo.pl

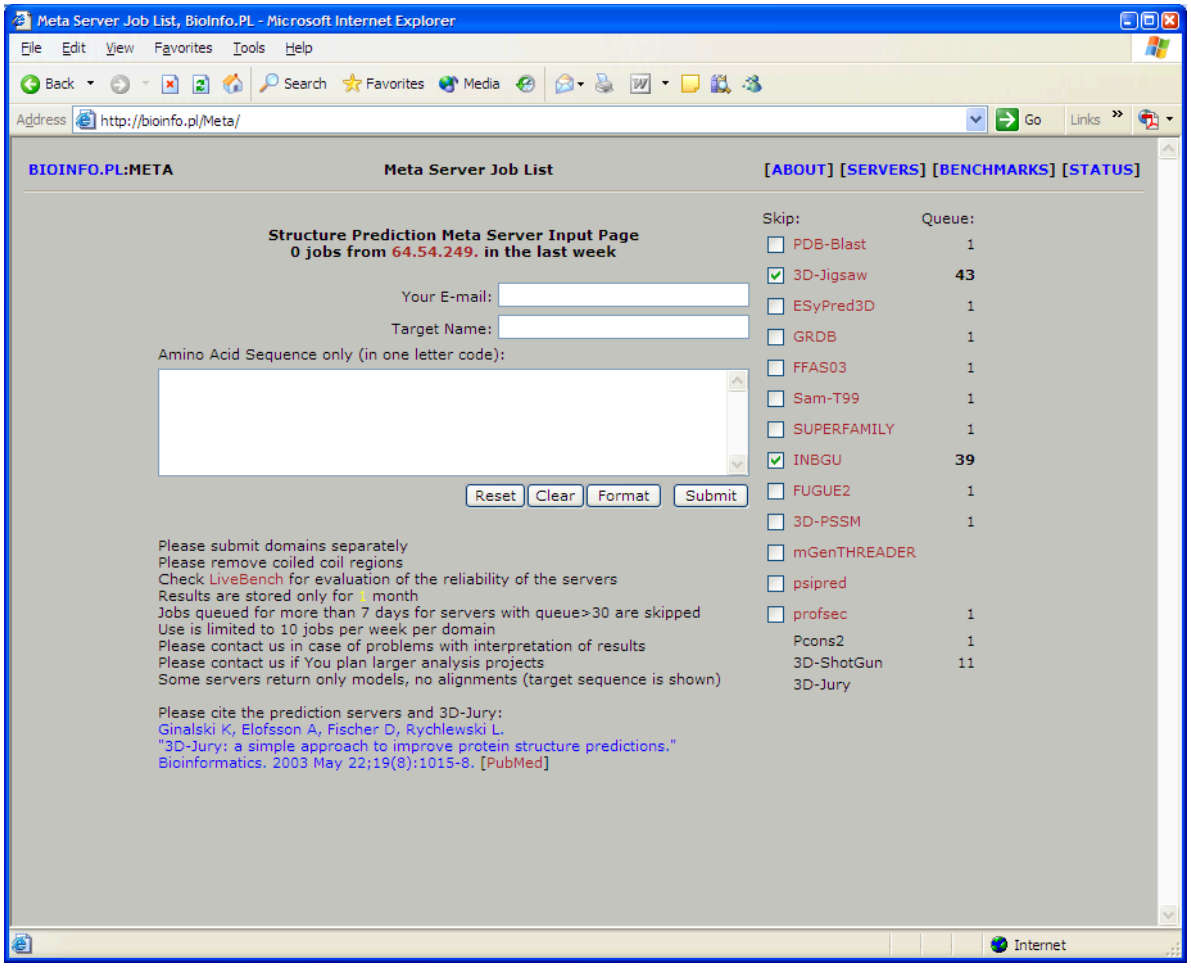
Figure 1. 3D Model of the nsp13 Domain of the SARS Coronavirus pp1ab Polyprotein

This model is based on the reassigned (Bujnicki and Rychlewski, 2001) cap-1 methyltransferase of the reovirus λ2 protein (1ej6 [Reinisch et al., 2000]). While other templates (1eiz or 1ej0) obtained marginally higher 3D jury scores, the selected template had the lowest number of insertions and deletions. Side chains of the conserved tetrad of residues (K-D-K-E) essential for cap-1 methylation and the docked AdoMet cofactor are shown. Four blocks of aligned motifs containing the conserved, function-specific residues are shown in upper right corner.

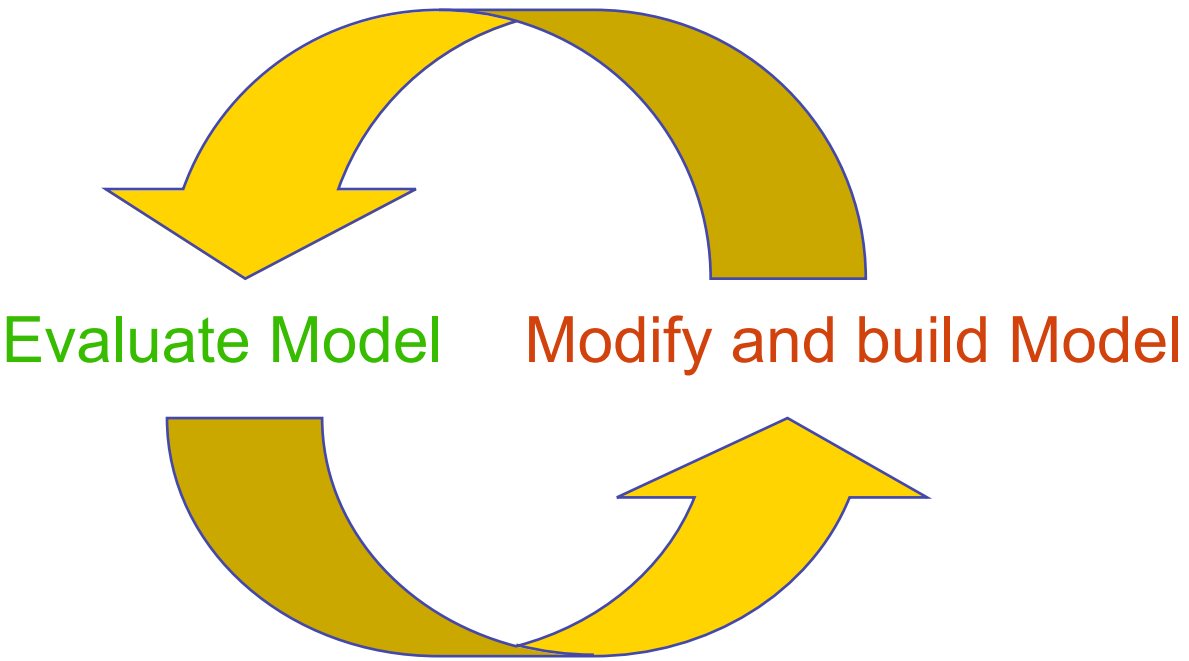
Ginalski K, et al. (2003) *Bioinformatics* **19** pp1015

Meta-Servers (3D-Jury)

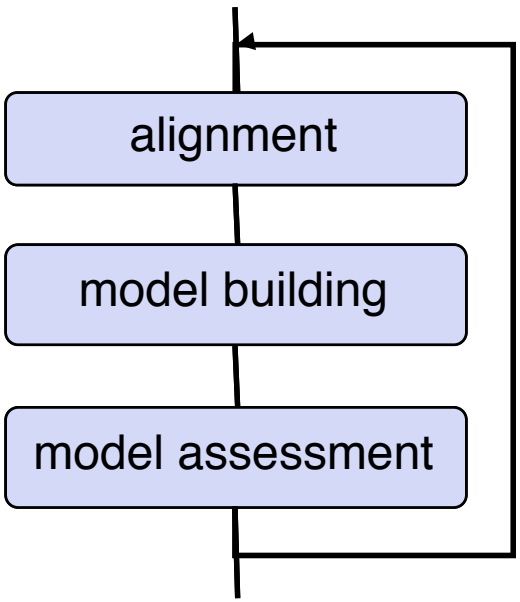
http://bioinfo.pl/Meta/

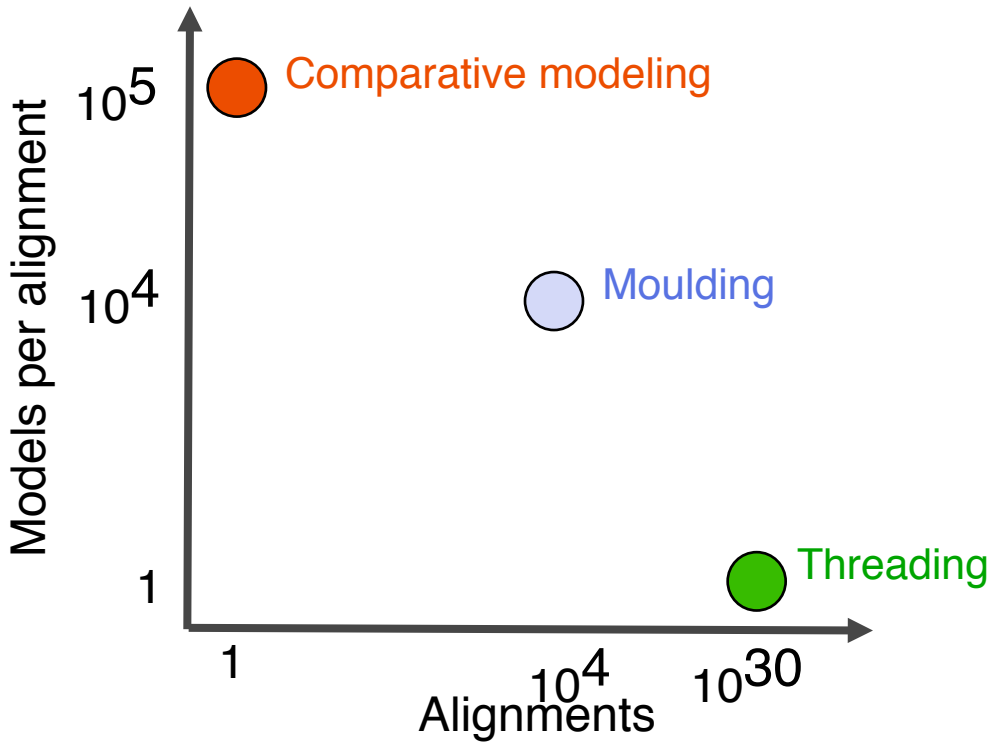


Iterative process… better models(?)

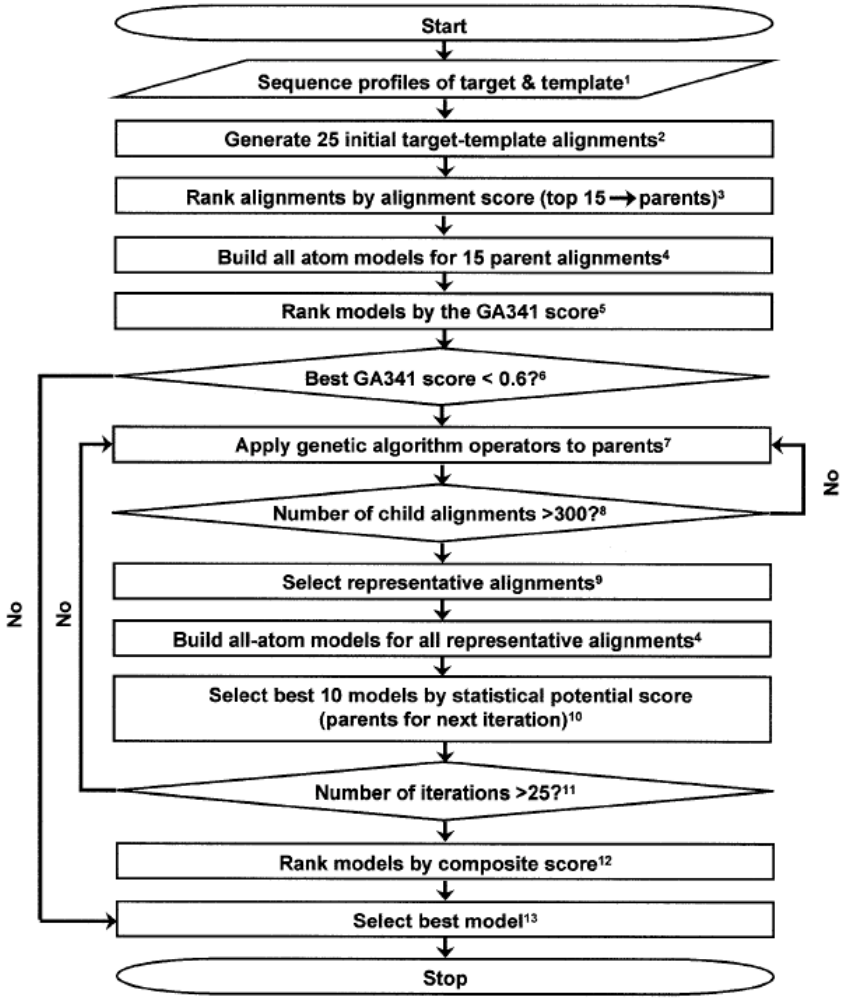


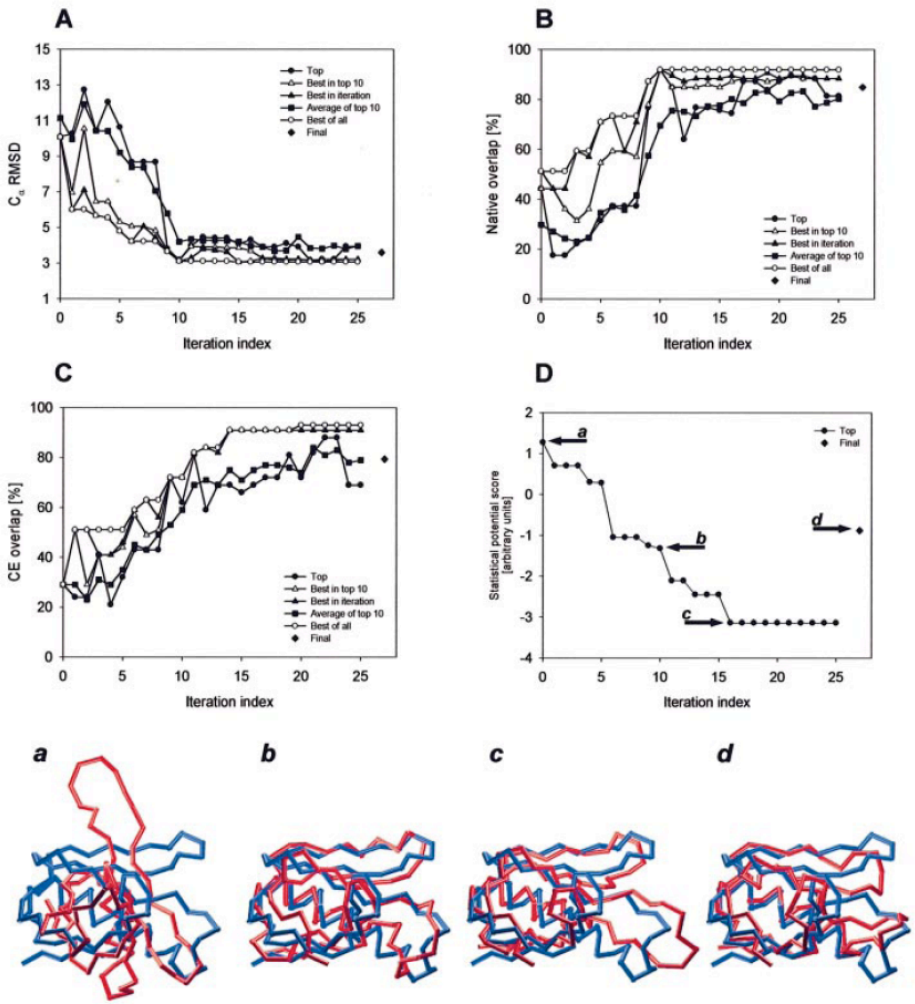
Moulding: iterative alignment, model building, model assessment





Iterative process… MOULDER

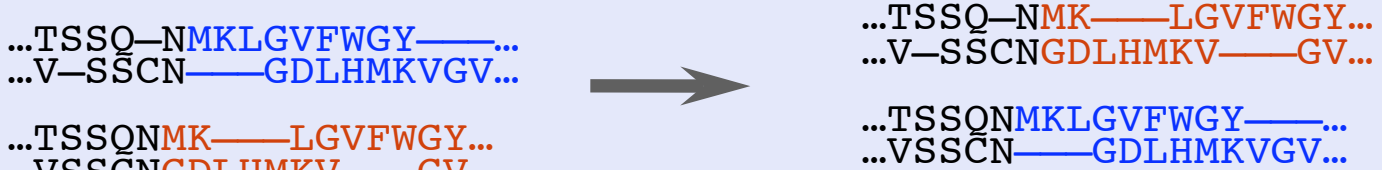




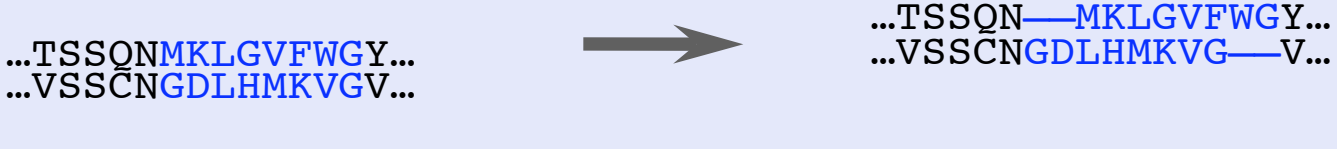
John & Sali (2003). *NAR* **31** pp3982

Genetic algorithm operators

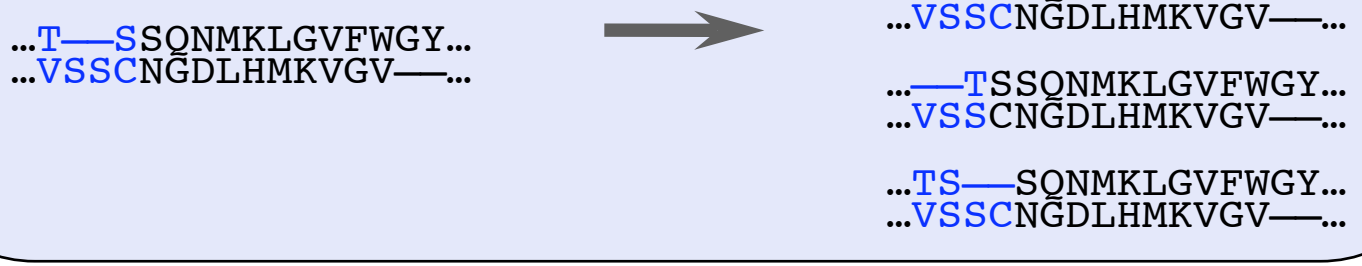
Single point cross-over



Gap insertion



Gap shift



Also, “two point crossover” and “gap deletion”.

Composite model assessment score

Weighted linear combination of several scores:

- Pair (P_p) and surface (P_s) statistical potentials;
- Structural compactness (S_c);
- Harmonic average distance score (H_a);
- Alignment score (A_s).

$$Z = 0.17\ Z(P_P) + 0.02\ Z(P_S) + 0.10\ Z(S_C) + 0.26\ Z(H_a) + 0.45\ (A_S)$$

$Z(\text{score}) = (\text{score} - \mu)/\sigma$

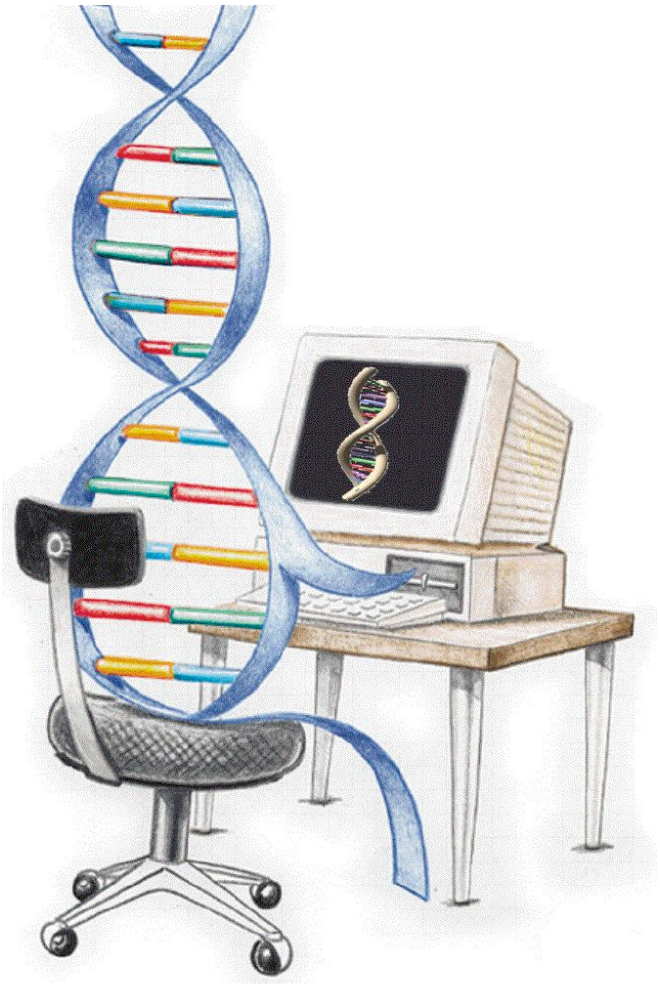
μ … average score of all models

σ … standard deviation of the scores

Benchmark with the “very difficult” test set

D. Fischer threading test set of 68 structural pairs (a subset of 19)

Target -template	Sequence identity [%]	Coverage [% aa]	Initial prediction		Final prediction		Best prediction	
			Cα RMSD [Å]	CE overlap [%]	Cα RMSD [Å]	CE overlap [%]	Cα RMSD [Å]	CE overlap [%]
1ATR-1ATN	13.8	94.3	19.2	20.2	18.8	20.2	17.1	24.6
1BOV-1LTS	4.4	83.5	10.1	29.4	3.6	79.4	3.1	92.6
1CAU-1CAU	18.8	96.7	11.7	15.6	10.0	27.4	7.6	47.4
1COL-1CPC	11.2	81.4	8.6	44.0	5.6	58.6	4.8	59.3
1LFB-1HOM	17.6	75.0	1.2	100.0	1.2	100.0	1.1	100.0
1NSB-2SIM	10.1	89.2	13.2	20.2	13.2	20.1	12.3	26.8
1RNH-1HRH	26.6	91.2	13.0	21.2	4.8	35.4	3.5	57.5
1YCC-2MTA	14.5	55.1	3.4	72.4	5.3	58.4	3.1	75.0
2AYH-1SAC	8.8	78.4	5.8	33.8	5.5	48.0	4.8	64.9
2CCY-1BBH	21.3	97.0	4.1	52.4	3.1	73.0	2.6	77.0
2PLV-1BBT	20.2	91.4	7.3	58.9	7.3	58.9	6.2	60.7
2POR-2OMF	13.2	97.3	18.3	11.3	11.4	14.7	10.5	25.9
2RHE-1CID	21.2	61.6	9.2	33.7	7.5	51.1	4.4	71.1
2RHE-3HLA	2.4	96.0	8.1	16.5	7.6	9.4	6.7	43.5
3ADK-1GKY	19.5	100.0	13.8	26.6	11.5	37.7	7.7	48.1
3HHR-1TEN	18.4	98.9	7.3	60.9	6.0	66.7	4.9	79.3
4FGF-81IB	14.1	98.6	11.3	24.0	9.3	30.6	5.4	41.2
6XIA-3RUB	8.7	44.1	10.5	14.5	10.1	11.0	9.0	34.3
9RNT-2SAR	13.1	88.5	5.8	41.7	5.1	51.2	4.8	69.0
AVERAGE	**14.2**	**85.2**	**9.6**	**36.7**	**7.7**	**44.8**	**6.3**	**57.8**



some biology? please...

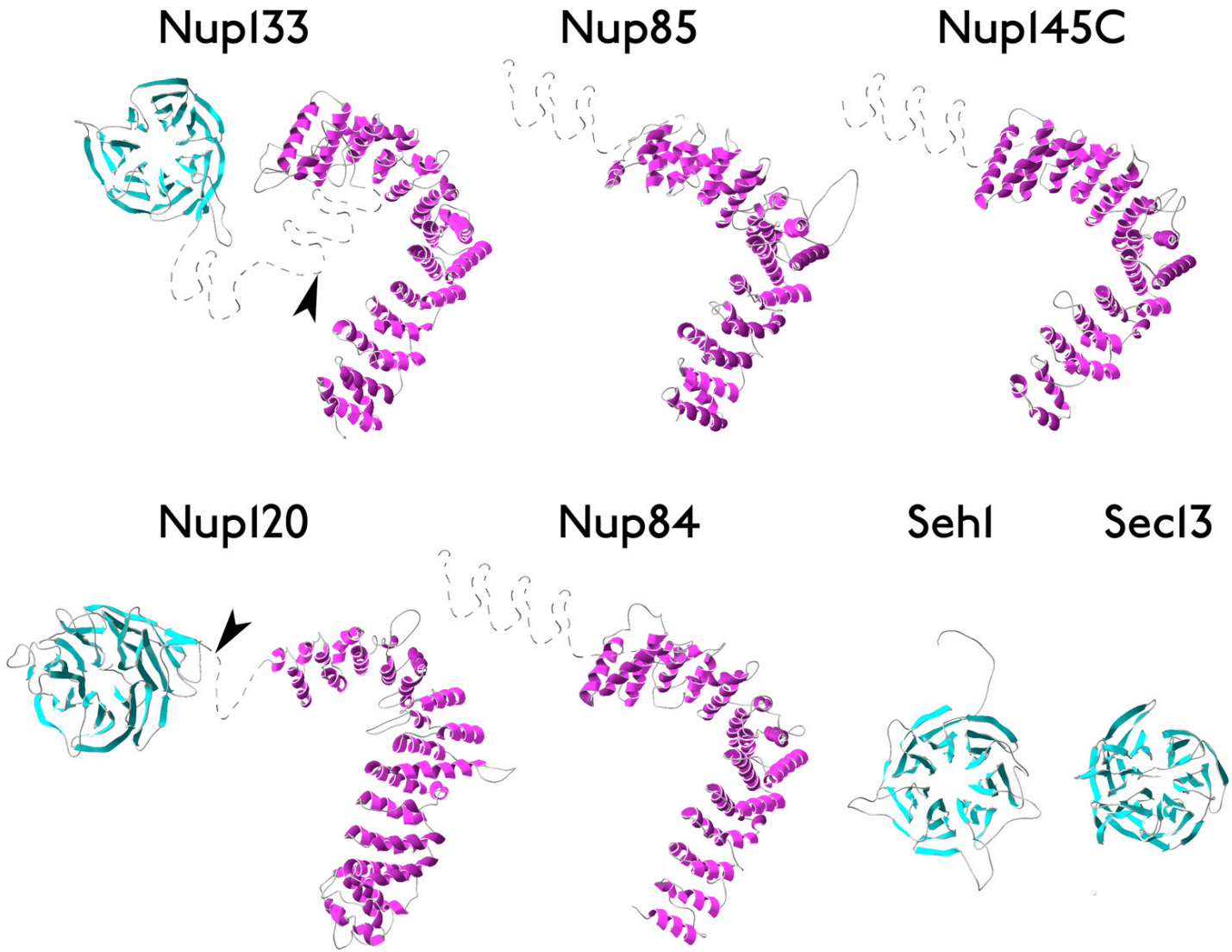
Common Evolutionary Origin of Coated Vesicles and Nuclear Pore Complexes

mGenThreader + SALIGN + MOULDER

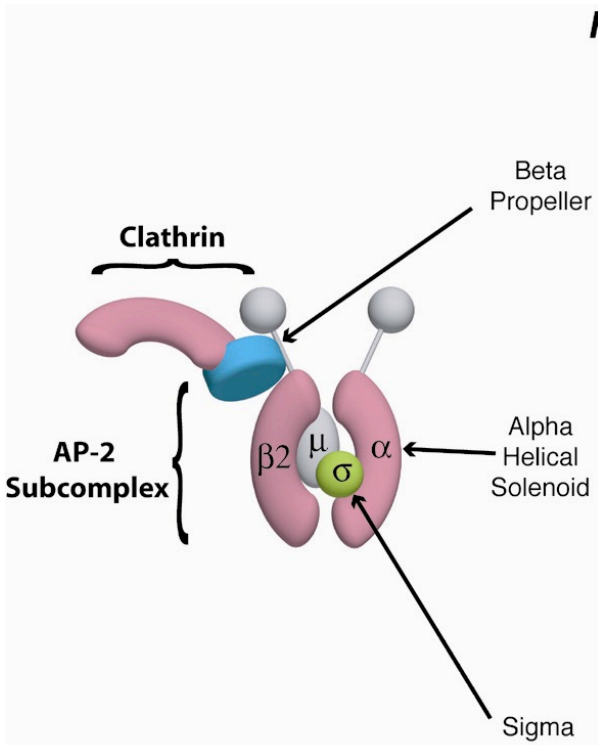
D. Devos, S. Dokudovskaya, F. Alber, R. Williams, B.T. Chait, A. Sali, M.P. Rout.
Components of Coated Vesicles and Nuclear Pore Complexes Share a Common Molecular Architecture.
PLOS Biology **2(12):**e380, 2004

yNup84 complex proteins

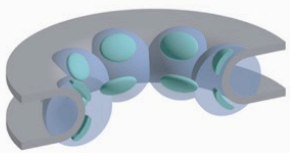
All Nucleoporins in the Nup84 Complex are Predicted to Contain β-Propeller and/or α-Solenoid Folds



NPC and Coated Vesicles Share the β-**Propeller and** α-**Solenoid** Folds and Associate with Membranes



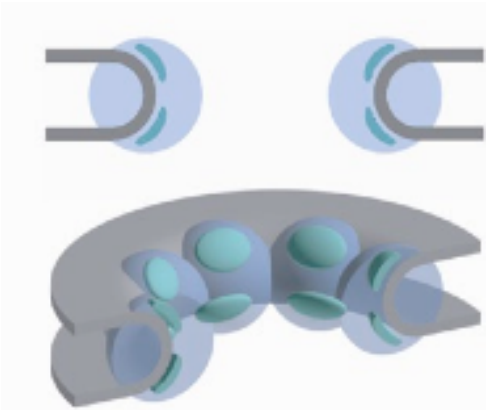
Fold	*CVs*	*COPI*	*COPII*	*NPC*
	Clathrin	α-subunit β'-subunit	Sec13 Sec31	Sec13 Seh1
	Clathrin α/γ/δ/ε-subunit β-subunit	α-subunit β-subunit γ-subunit ε-subunit β'-subunit	Sec31	Nup133 Nup120 Nup85 Nup84 Nup145C



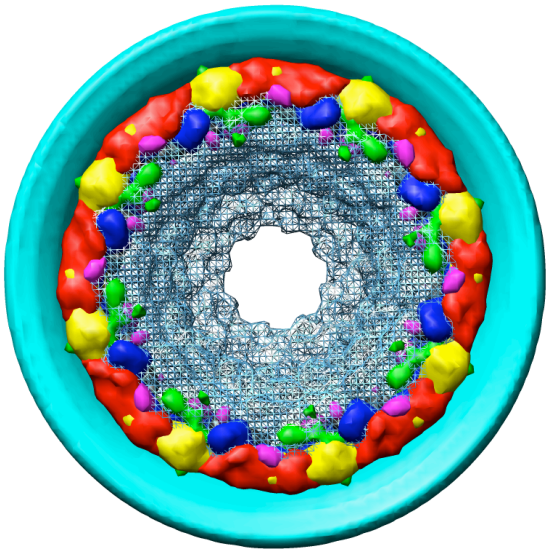
NPC and Coated Vesicles Both Associate with Membranes

Coated Vesicle

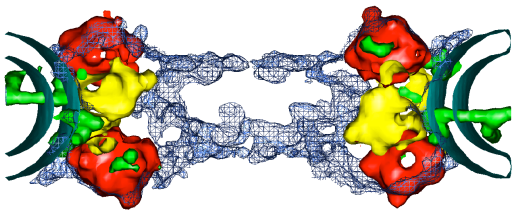
NPC model



top view

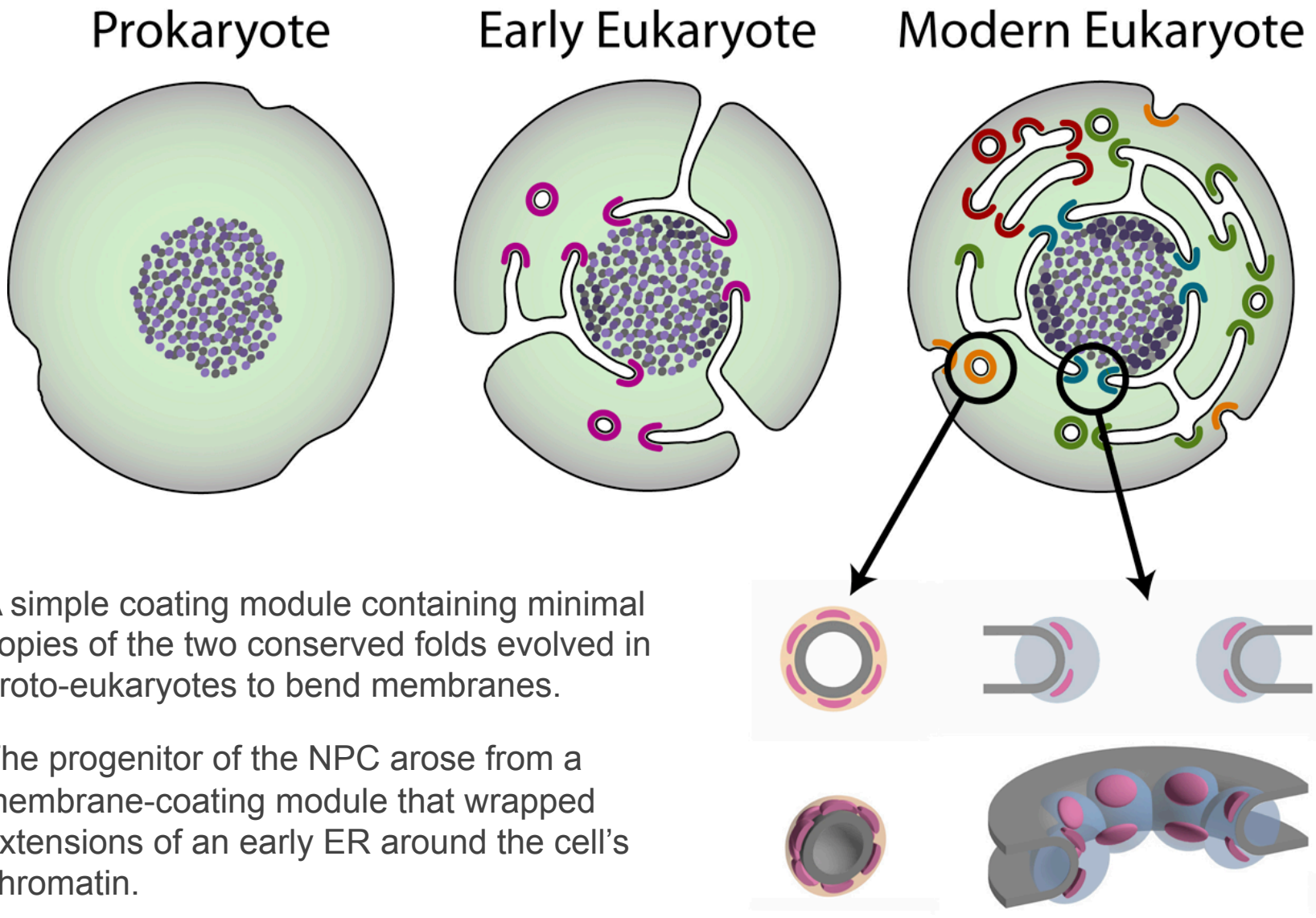


side view



Nup 84 complex

A Common Evolutionary Origin for Nuclear Pore Complexes and Coated Vesicles?
The proto-coatomer hypothesis



A simple coating module containing minimal copies of the two conserved folds evolved in proto-eukaryotes to bend membranes.

The progenitor of the NPC arose from a membrane-coating module that wrapped extensions of an early ER around the cell's chromatin.

BMI 206

MODELLER TUTORIAL

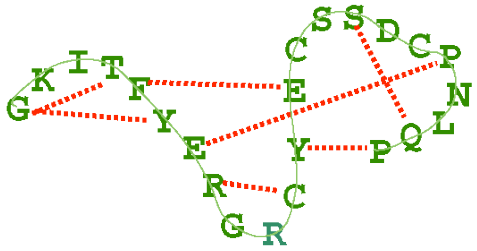
http://www.salilab.org/modeller/tutorial/

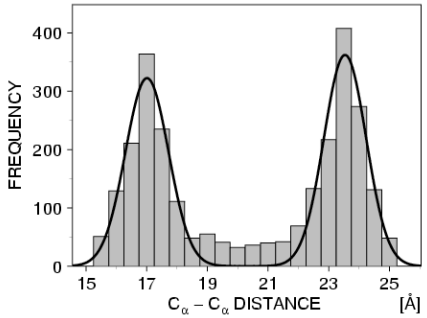
Marc A. Marti-Renom
Assistant Adjunct Professor
Department of Biopharmaceutical Sciences

Comparative Modeling by Satisfaction of Spatial Restraints (MODELLER)

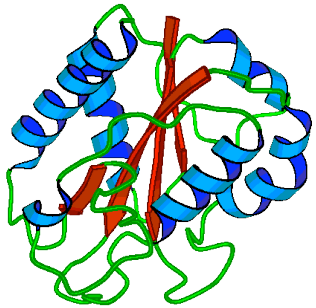
3D GKITFYERGFQGHCYESDC-NLQP…
SE GKITFYERG---RCYESDCPNLQP…

1. Extract spatial restraints





2. Satisfy spatial restraints

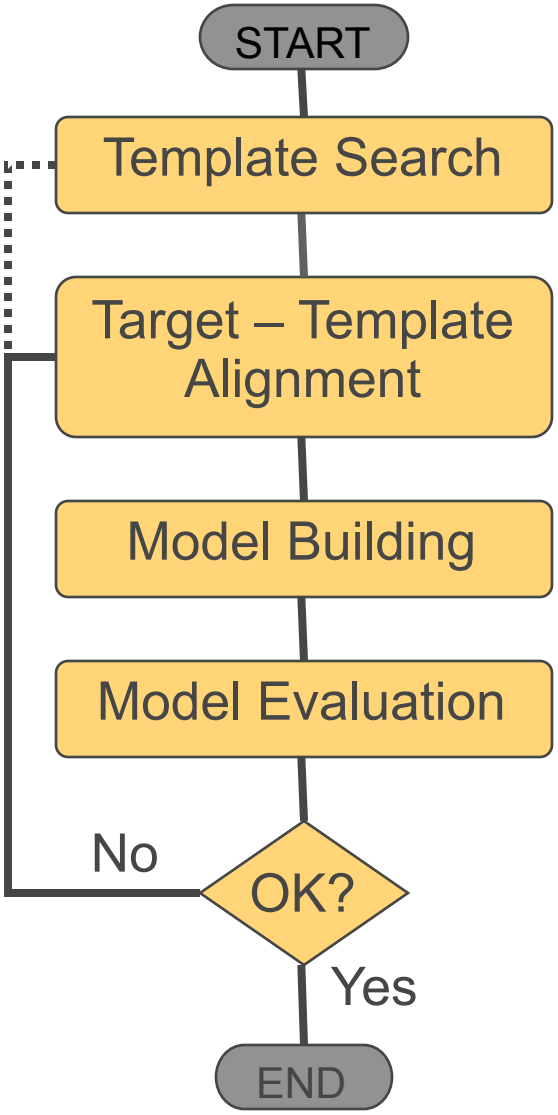


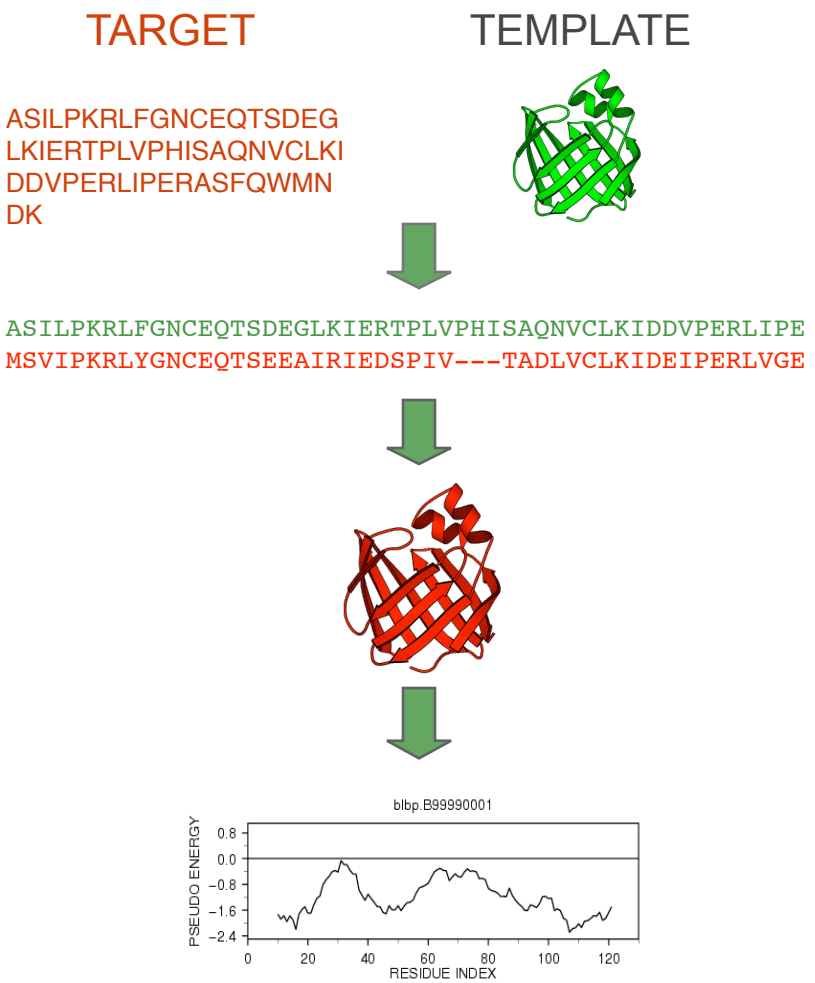
$$F(R) = \prod_i p_i(f_i/I)$$

http://www.salilab.org/modeller

A. Šali & T. Blundell. J. Mol. Biol. 234, 779, 1993.
J.P. Overington & A. Šali. Prot. Sci. 3, 1582, 1994.
A. Fiser, R. Do & A. Šali, Prot. Sci., 9, 1753, 2000.

Steps in Comparative Protein Structure Modeling





A. Šali, Curr. Opin. Biotech. 6, 437, 1995.
R. Sánchez & A. Šali, Curr. Opin. Str. Biol. 7, 206, 1997.
M. Marti et al. Ann. Rev. Biophys. Biomolec. Struct., 29, 291, 2000.

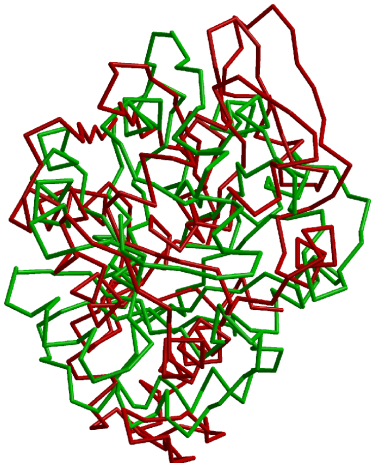
Typical errors in comparative models

MODEL

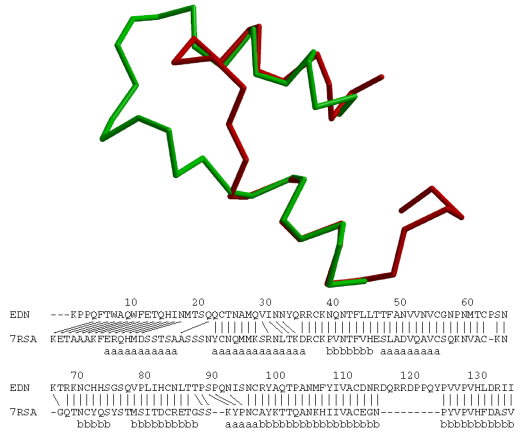
X-RAY

TEMPLATE

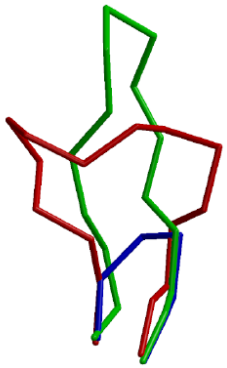
Incorrect template



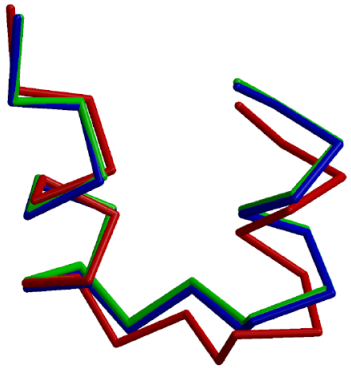
Misalignment



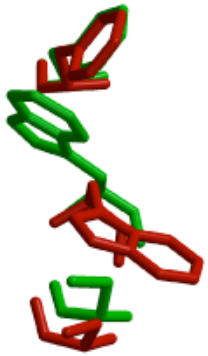
Region without a template



Distortion/shifts in aligned regions

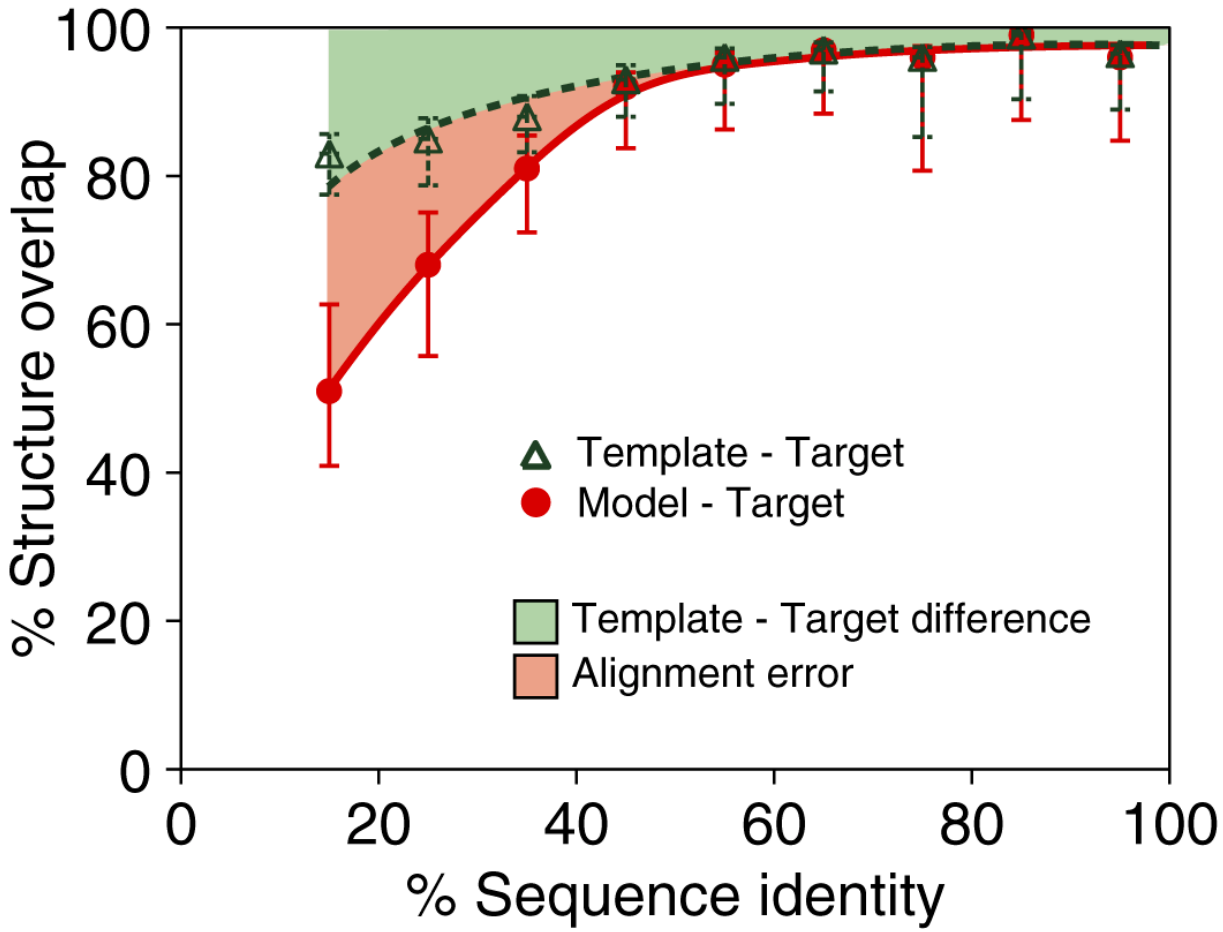


Sidechain packing



Marti-Renom et al. Annu.Rev.Biophys.Biomol.Struct. 29, 291-325, 2000.

Model Accuracy as a Function of Target-Template Sequence Identity

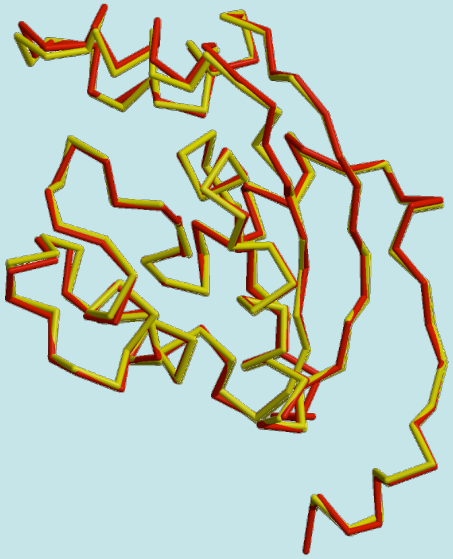


Sánchez, R., Šali, A. Proc Natl Acad Sci U S A. 95 pp13597-602. (1998).

Model Accuracy

HIGH ACCURACY

NM23
Seq id 77%
Cα equiv 147/148
RMSD 0.41Å

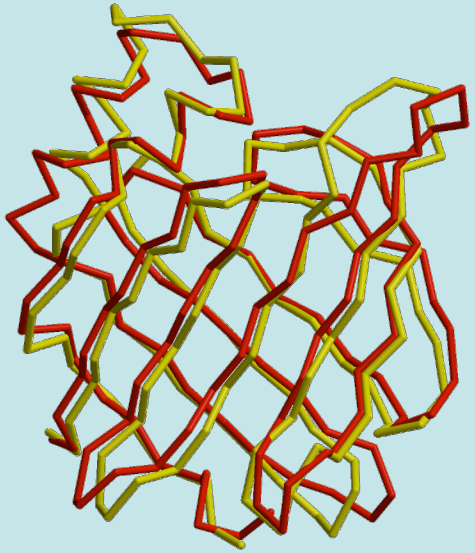


Sidechains
Core backbone
Loops

X-RAY / MODEL

MEDIUM ACCURACY

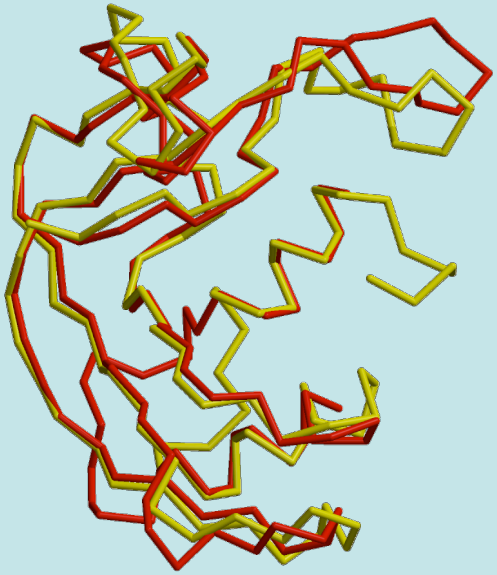
CRABP
Seq id 41%
Cα equiv 122/137
RMSD 1.34Å



Sidechains
Core backbone
Loops
Alignment

LOW ACCURACY

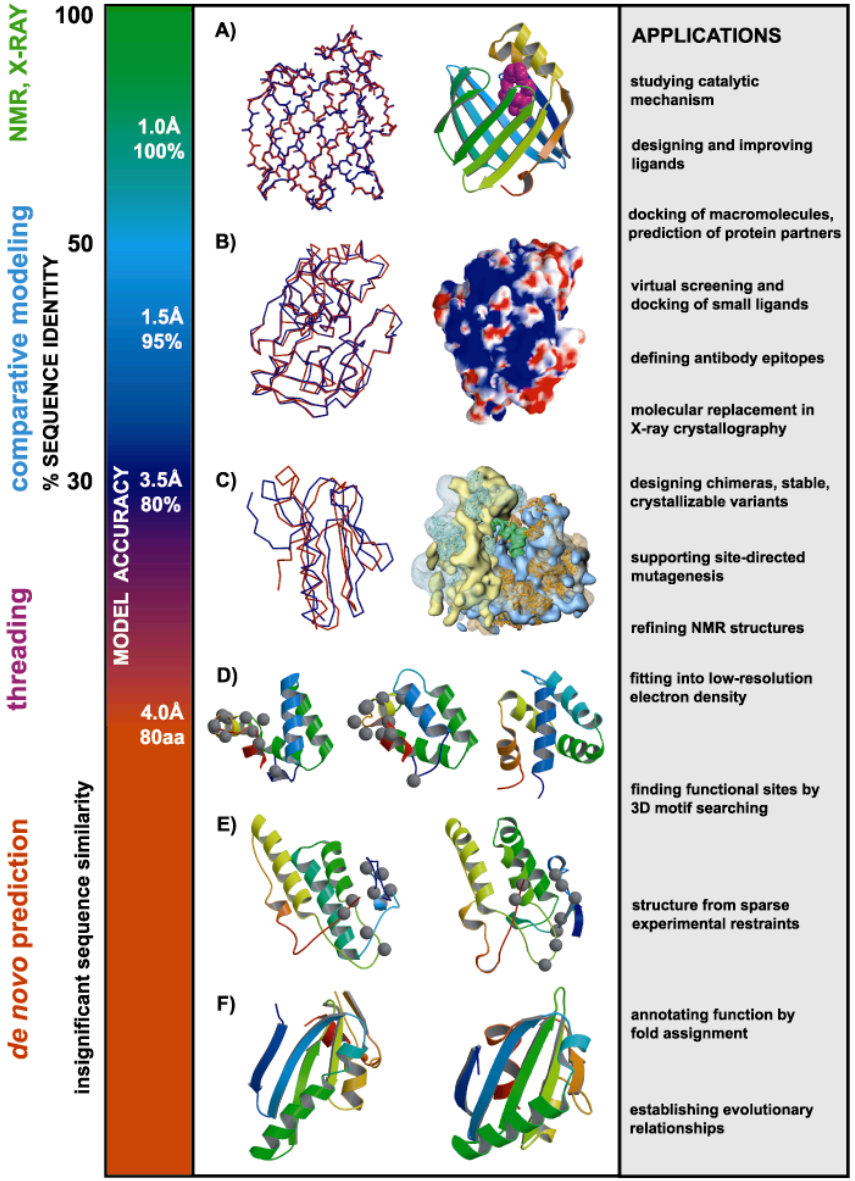
EDN
Seq id 33%
Cα equiv 90/134
RMSD 1.17Å



Sidechains
Core backbone
Loops
Alignment
Fold assignment

Marti-Renom et al. Annu.Rev.Biophys.Biomol.Struct. 29, 291-325, 2000.

Applications of Protein Structure Models



D. Baker & A. Sali. Science 294, 93, 2001.

Obtaining MODELLER and related information

- MODELLER (8v0) web page
- `http://www.salilab.org/modeller/`
  - Download Software (Linux/Windows/Mac/Solaris)
  - HTML Manual
  - **Join Mailing List**



Using MODELLER

- No GUI! ☹
- Controlled by command file ☹☹
- Script is written in PYTHON language ☺
- You may know Python language, is simple ☺☺

Using MODELLER

- INPUT:
  - Target Sequence (FASTA/PIR format)
  - Template Structure (PDB format)
  - Python command file
- OUTPUT:
  - Target-Template Alignment
  - Model in PDB format
  - Other data

Modeling of BLBP
Input

- Target: Brain lipid-binding protein (BLBP)
- BLBP sequence in PIR (MODELLER) format:

```
>P1;blbp
sequence:blbp:::::::
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSID
DRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod8v0 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod8v0 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod8v0 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Python script for target-template alignment

```
# Example for: alignment.align()

# This will read two sequences, align them, and write the alignment
# to a file:

log.verbose()
env = environ()

aln = alignment(env)
mdl = model(env, file='1hms')
aln.append_model(mdl, align_codes='1hms')
aln.append(file='blbp.seq', align_codes=('blbp'))

# The as1.sim.mat similarity matrix is used by default:
aln.align(gap_penalties_1d=(-600, -400))
aln.write(file='blbp-1hms.ali', alignment_format='PIR')
aln.write(file='blbp-1hms.pap', alignment_format='PAP')
```

Run by typing `mod8v0 align.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Output

```
>P1;1hms
structureX:1hms:    1 : : 131 : :undefined:undefined:-1.00:-1.00
VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNTEISFKLGVEFDETTA
DDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTYEKE*
>P1;blbp
sequence:blbp:      : :      : : : : 0.00: 0.00
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSI
DDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Output

```
>P1;1hms
structureX:1hms:    1 : : 131 : :undefined:undefined:-1.00:-1.00
VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNTEISFKLGVEFDETTA
DDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTYEKE*
>P1;blbp
sequence:blbp:      : :      : : : : 0.00: 0.00
VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGEEFEETSI
DDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA*
```

Modeling of BLBP

STEP 1: Align blbp and 1hms sequences

Output

```
 _aln.pos          10         20         30         40         50         60
1hms       VDAFLGTWKLVDSKNFDDYMKSLGVGFATRQVASMTKPTTIIEKNGDILTLKTHSTFKNTEISFKLGV
blbp       VDAFCATWKLTDSQNFDEYMKALGVGFATRQVGNVTKPTVIISQEGGKVVIRTQCTFKNTEINFQLGE
 _consrvd  ****   **** ** *** *** **********    **** **    *       *   ******* * **


 _aln.p    70         80         90         100        110        120        130
1hms       EFDETTADDRKVKSIVTLDGGKLVHLQKWDGQETTLVRELIDGKLILTLTHGTAVCTRTYEKE
blbp       EFEETSIDDRNCKSVVRLDGDKLIHVQKWDGKETNCTREIKDGKMVVTLTFGDIVAVRCYEKA
 _consrvd  ** **   ***   ** * *** ** * ***** **    **   ***    *** *   *   * ***
```

Modeling of BLBP

STEP 2: Model the blbp structure using the alignment from step 1.

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *    # Load the automodel class
log.verbose()                       # request verbose output
env = environ()                     # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = './:../atom_files'

a = automodel(env,
              alnfile  = 'blbp-1hms.ali',     # alignment filename
              knowns   = '1hms',              # codes of the templates
              sequence = 'blbp')              # code of the target
a.starting_model= 1                 # index of the first model
a.ending_model  = 1                 # index of the last model
                                    # (determines how many models to calculate)
a.make()                            # do the actual homology modelling
```

Run by typing `mod8v0 model.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 2: Model the blbp structure using the alignment from step 1.

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *    # Load the automodel class
log.verbose()                       # request verbose output
env = environ()                     # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = './:../atom_files'

a = automodel(env,
              alnfile  = 'blbp-1hms.ali',     # alignment filename
              knowns   = '1hms',              # codes of the templates
              sequence = 'blbp')              # code of the target
a.starting_model= 1                 # index of the first model
a.ending_model  = 1                 # index of the last model
                                    # (determines how many models to calculate)
a.make()                            # do the actual homology modelling
```

Run by typing `mod8v0 model.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 2: Model the blbp structure using the alignment from step 1.

Python script for model building

```
# Homology modelling by the automodel class
from modeller.automodel import *    # Load the automodel class
log.verbose()                       # request verbose output
env = environ()                     # create a new MODELLER environment

# directories for input atom files
env.io.atom_files_directory = './:../atom_files'

a = automodel(env,
              alnfile  = 'blbp-1hms.ali',     # alignment filename
              knowns   = '1hms',              # codes of the templates
              sequence = 'blbp')              # code of the target
a.starting_model= 1                 # index of the first model
a.ending_model  = 1                 # index of the last model
                                    # (determines how many models to calculate)
a.make()                            # do the actual homology modelling
```

Run by typing `mod8v0 model.py` in the directory where you have the python file.
MODELLER will produce a `align.log` file

Modeling of BLBP

STEP 2: Model the blbp structure using the alignment from step 1.

Python script for model building

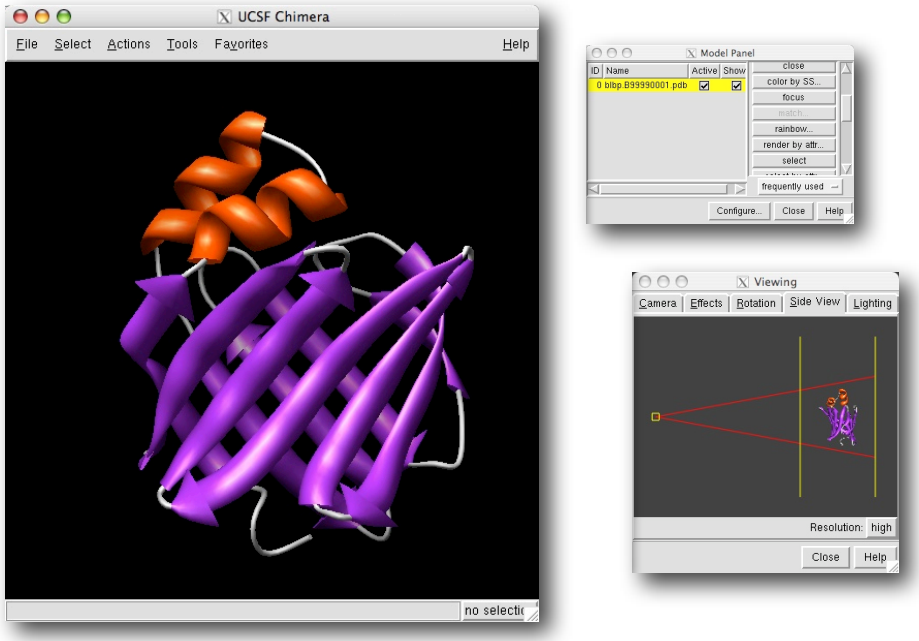
PDB file

Can be viewed with Chimera

http://www.cgl.ucsf.edu/chimera/

Rasmol

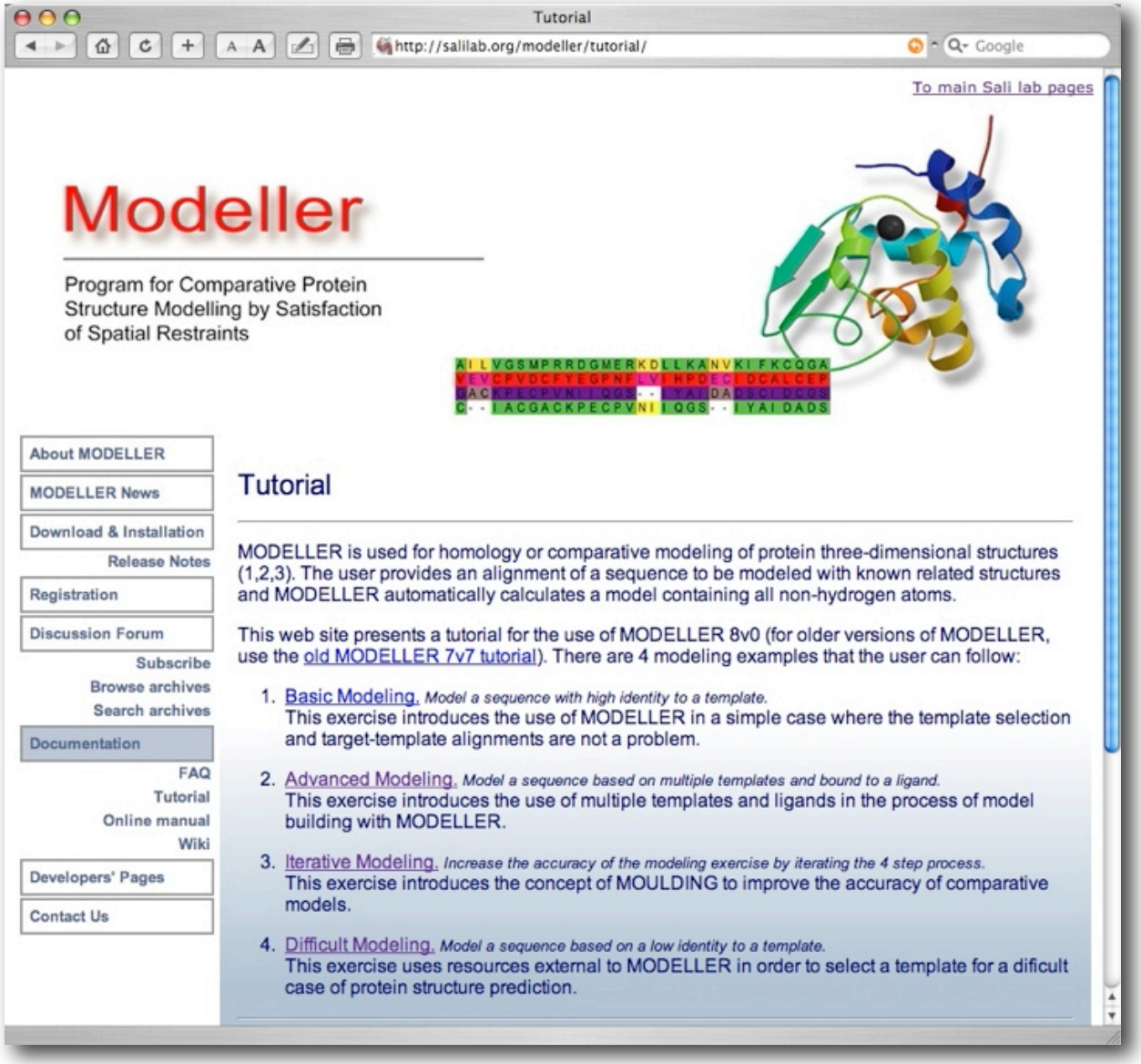
http://www.openrasmol.org



Model file → `blbp.B99990001`

http://www.salilab.org/bioinformatics_resources.shtml

http://www.salilab.org/modeller/tutorial/



References

Protein Structure Prediction:

Marti-Renom el al. Annu. Rev. Biophys. Biomol. Struct. 29, 291-325, 2000.
Baker & Sali. Science 294, 93-96, 2001.

Comparative Modeling:

Marti-Renom el al. Annu. Rev. Biophys. Biomol. Struct. 29, 291-325, 2000.
Marti-Renom el al. Current Protocols in Protein Science 1, 2.9.1-2.9.22, 2002.

MODELLER:

Sali & Blundell. J. Mol. Biol. 234, 779-815, 1993.

Structural Genomics:

Sali. Nat. Struct. Biol. 5, 1029, 1998.
Burley et al. Nat. Genet. 23, 151, 1999.
Sali & Kuriyan. TIBS 22, M20, 1999.
Sanchez et al. Nat. Str. Biol. 7, 986, 2000.
Baker & Sali. Science 294, 93-96, 2001.
Vitkup et al. Nat. Struct. Biol. 8, 559, 2001.