Protein Structure Prediction

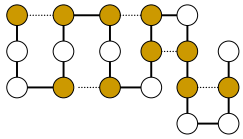
- *Ab-initio* techniques
- Homology modeling
  - Sequence-sequence comparison
- Protein threading
  - Sequence-structure comparison

Lattice models

- Simple lattice models (HP-models)
  - Two types of residues: hydrophobic and polar
  - 2-D or 3-D lattice
  - The only force is hydrophobic collapse
  - Score = number of H–H contacts

Scoring Lattice Models

- H/P model scoring: count hydrophobic interactions.



Score = 5

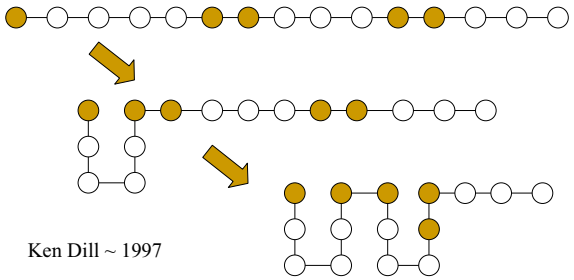
- Sometimes:
  - Penalize for buried polar or surface hydrophobic residues

What can we do with lattice models?

- NP-complete
- For smaller polypeptides, exhaustive search can be used
  - Looking at the "best" fold, even in such a simple model, can teach us interesting things about the protein folding process
- For larger chains, other optimization and search methods must be used
  - Greedy, branch and bound
  - Evolutionary computing, simulated annealing
  - Graph theoretical methods

Learning from Lattice Models

- The "hydrophobic zipper" effect:



Ken Dill ~ 1997

Learning from Lattice Models

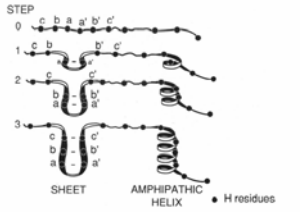
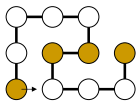
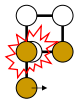


FIG. 1. HZ model of protein-folding pathways. The closest hydrophobic (H) residues (solid dots) in sequence pair together first, e.g., a and a' in step 0. They constrain the chain and bring other H monomers, such as the (b,b') pair, into spatial proximity. Now (b,b') further constrains the chain and brings the (c,c') pair into spatial proximity, etc. As H contacts form and develop a core, helices and sheets zip up if they have appropriate H sequences.

Representing a lattice model

- Absolute directions
  - UURRDLDRRU
- Relative directions
  - LFRFRRLLFL
  - Advantage, we can't have UD or RL in absolute
  - Only three directions: LRF
- *What about bumps?* LFRRR
  - Give bad score to any configuration that has bumps



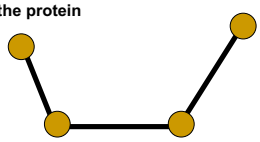


More realistic models

- Higher resolution lattices (45° lattice, etc.)
- Off-lattice models
  - Local moves
  - Optimization/search methods and ϕ/ψ representations
    - Greedy search
    - Branch and bound
    - EC, Monte Carlo, simulated annealing, etc.

Energy functions

- **An energy function to describe the protein**
  - bond energy
  - bond angle energy
  - dihedral angel energy
  - van der Waals energy
  - electrostatic energy
- **Minimize the function and obtain the structure.**
- **Not practical in general**
  - Computationally too expensive
  - Accuracy is poor
- Empirical force fields
  - Start with a database
  - Look at neighboring residues – similar to known protein folds?



Difficulties

Why is structure prediction and especially *ab initio* calculations hard?

- Many degrees of freedom / residue. Computationally too expensive for realistic-sized proteins.
- Remote non-covalent interactions
- Nature does not go through all conformations
- Folding assisted by enzymes & chaperones

Protein Structure Prediction

- *Ab-initio* techniques
- Homology modeling
  - Sequence-sequence comparison
- Protein threading
  - Sequence-structure comparison

Homology modeling steps

1. Identify a set of template proteins (with known structures) related to the target protein. This is based on sequence homology (BLAST, FASTA) with sequence identity of 30% or more.
2. Align the target sequence with the template proteins. This is based on multiple alignment (CLUSTALW). Identify conserved regions.
3. Build a model of the protein backbone, taking the backbone of the template structures (conserved regions) as a model.
4. Model the loops. In regions with gaps, use a loop-modeling procedure to substitute segments of appropriate length.
5. Add sidechains to the model backbone.
6. Evaluate and optimize entire structure.

Homology Modeling

- Servers
  - SWISS-MODEL
  - ESyPred3D

Protein Structure Prediction

- *Ab-initio* techniques
- Homology modeling
- Protein threading
  - Sequence-structure comparison

Protein threading

Structure is better conserved than sequence

Structure can adopt a wide range of mutations.

Physical forces favor certain structures.

Number of folds is limited.
Currently ~700
Total: 1,000 ~10,000

TIM barrel

Protein Threading

- Basic premise

The number of unique structural (domain) folds in nature is fairly small (possibly a few thousand)

- Statistics from Protein Data Bank (~35,000 structures)

90% of new structures submitted to PDB in the past three years have similar structural folds in PDB

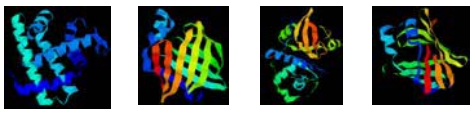
Concept of Threading

- Thread (*align* or *place)* a query protein sequence onto a template structure in "optimal" way
- Good alignment gives approximate backbone structure

Query sequence

MTYKLILNGKTKGETTTEAVDAATAEKVFQYANDNGVDGEWTYTE

Template set



Threading problem

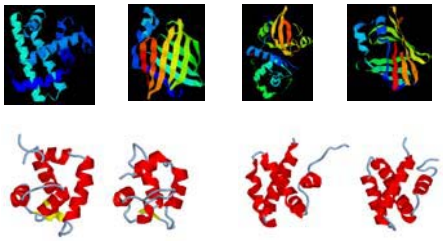
- Threading: Given a sequence, and a fold (template), compute the optimal alignment score between the sequence and the fold.
- If we can solve the above problem, then
  - Given a sequence, we can try each known fold, and find the best fold that fits this sequence.
  - Because there are only a few thousands folds, we can find the correct fold for the given sequence.
- Threading is NP-hard.

Components of Threading

- Template library
  - Use structures from DB classification categories (PDB)
- Scoring function
  - Single and pairwise energy terms
- Alignment
  - Consideration of pairwise terms leads to NP-hardness
  - heuristics
- Confidence assessment
  - Z-score, P-value similar to sequence alignment statistics
- Improvements
  - Local threading, multi-structure threading

Protein Threading – structure database

- Build a template database

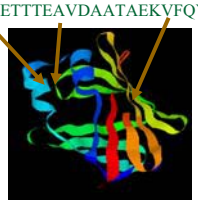


Protein Threading – energy function

MTYKLILNGKTKGETTTEAVDAATAEKVFQYANDNGVDGEWTYTE

how preferable to put two particular residues nearby: E_p

alignment gap penalty: E_g



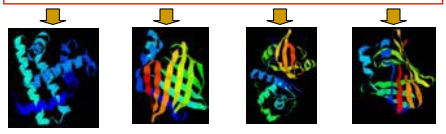
how well a residue fits a structural environment: E_s

total energy: E_p + E_s + E_g

find a sequence-structure alignment to minimize the energy function

Assessing Prediction Reliability

MTYKLILNGKTKGETTTEAVDAATAEKVFQYANDNGVDGEWTYTE



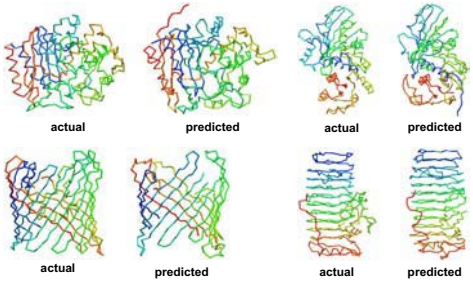
Score = -1500 Score = -720 Score = -1120 Score = -900

Which one is the correct structural fold for the target sequence if any?

The one with the highest score ?

Prediction of Protein Structures

- Examples – a few good examples



actual predicted actual predicted

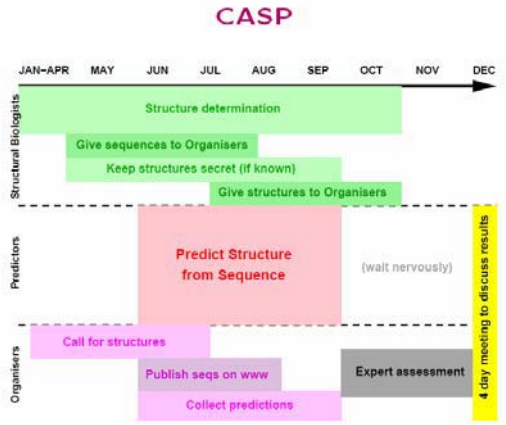
actual predicted actual predicted

Prediction of Protein Structures

- Not so good example

Existing Prediction Programs

- PROSPECT
  - https://csbl.bmb.uga.edu/protein_pipeline
- FUGU
  - http://www-cryst.bioc.cam.ac.uk/~fugue/prfsearch.html
- THREADER
  - http://bioinf.cs.ucl.ac.uk/threader/



CASP/CAFASP

- CASP: Critical Assessment of Structure Prediction

CASP Predictor

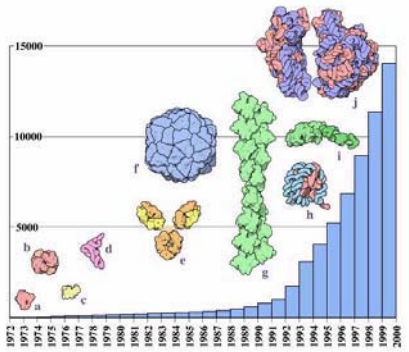
- CAFASP: Critical Assessment of Fully Automated Structure Prediction

CAFASP Predictor

1. Won't get tired
2. High-throughput

CASP6/CAFASP4

- 64 targets
- Resources for predictors
  - No X-ray, NMR machines (of course)
  - CAFASP4 predictors: no manual intervention
  - CASP6 predictors: anything (servers, google,…)
- Evaluation:
  - CASP6 Assessed by experts+computer
  - CAFASP4 evaluated by a computer program.
  - Predicted structures are superimposed on the experimental structures.
- CASP7 will be held this year (November)



(a) myoglobin (b) hemoglobin (c) lysozyme (d) transfer RNA (e) antibodies (f) viruses (g) actin (h) the nucleosome (i) myosin (j) ribosome

Courtesy of David Goodsell, TSRI

Protein structure databases

- PDB
  - 3D structures
- SCOP
  - Murzin, Brenner, Hubbard, Chothia
  - Classification
    - Class (mostly alpha, mostly beta, alpha/beta (interspersed), alpha+beta (segregated), multi-domain, membrane)
    - Fold (similar structure)
    - Superfamily (homology, distant sequence similarity)
    - Family (homology and close sequence similarity)

The SCOP Database

Structural Classification Of Proteins

FAMILY: proteins that are >30% similar, or >15% similar and have similar known structure/function

SUPERFAMILY: proteins whose families have some sequence and function/structure similarity suggesting a common evolutionary origin

COMMON FOLD: superfamilies that have same secondary structures in same arrangement, probably resulting by physics and chemistry

CLASS: alpha, beta, alpha–beta, alpha+beta, multidomain

Protein databases

- CATH
  - Orengo et al
  - Class (alpha, beta, alpha/beta, few SSEs)
  - Architecture (orientation of SSEs but ignoring connectivity)
  - Topology (orientation and connectivity, based on SSAP = fold of SCOP)
  - Homology (sequence similarity = superfamily of SCOP)
    - S level (high sequence similarity = family of SCOP)
  - SSAP alignment tool (dynamic programming)

Protein databases

- FSSP
  - DALI structure alignment tool (distance matrix)
    - Holm and Sander
- MMDB
  - VAST structure comparison (hierarchical)
    - Madej, Bryant et al

Protein structure comparison

- Levels of structure description
  - Atom/atom group
  - Residue
  - Fragment
  - Secondary structure element (SSE)
- Basis of comparison
  - Geometry/architecture of coordinates/relative positions
  - sequential order of residues along backbone, ...
  - physio-chemical properties of residues, …

How to compare?

- **Key problem**: find an optimal correspondence between the arrangements of atoms in two molecular structures (say A and B) in order to align them in 3D
- Optimality of the alignment is determined using a root mean square measure of the distances between corresponding atoms in the two molecules
- **Complication**: It is not known a priori which atom in molecule B corresponds to a given atom in molecule A (the two molecules may not even have the same number of atoms)

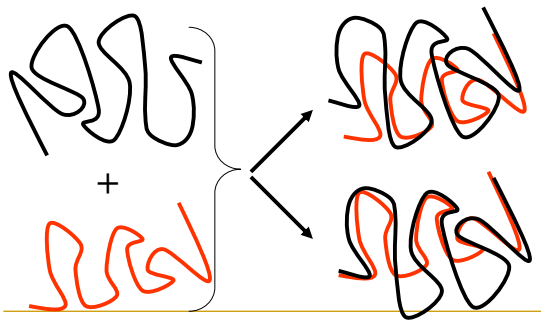
Structure Analysis – Basic Issues

- Coordinates for representing 3D structures
  - Cartesian
  - Other (e.g. dihedral angles)
- Basic operations
  - Translation in 3D space
  - Rotation in 3D space
  - Comparing 3D structures
    - Root mean square distances between points of two molecules are typically used as a measure of how well they are aligned
    - Efficient ways to compute minimal RMSD once correspondences are known (O(n) algorithm)
      - Using eigenvalue analysis of correlation matrix of points
- Due to the high computational complexity, practical algorithms rely on heuristics

Structure Analysis – Basic Issues

- Sequence order dependent approaches
  - Computationally this is easier
  - Interest in motifs preserving sequence order
- Sequence order independent approaches
  - More general
  - Active sites may involve non-local AAs
  - Searching with structural information

Find the optimal alignment



Optimal Alignment

- Find the highest number of atoms aligned with the lowest RMSD (Root Mean Squared Deviation)
- Find a balance between local regions with very good alignments and overall alignment

Structure Comparison

Which atom in structure A corresponds to which atom in structure B ?

```
THESESENTENCESALIGN--NICELY
|||   ||   |||| |||||   ||||||
THE--SEQUENCE-ALIGNEDNICELY
```

Structural Alignment

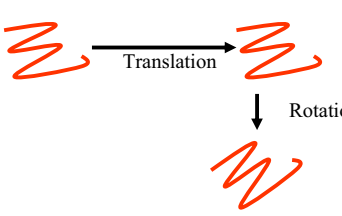
Structural Alignment of Two Globins

An optimal superposition of myoglobin and beta-hemoglobin, which are structural neighbors. However, their sequence homology is only 8.5%

Structure Comparison

Methods to superimpose structures by translation and rotation

$$\begin{pmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ x_3, y_3, z_3 \end{pmatrix}$$



$$\begin{pmatrix} x_1 + d, y_1, z_1 \\ x_2 + d, y_2, z_2 \\ x_3 + d, y_3, z_3 \end{pmatrix}$$

Structure Comparison

Scoring system to find optimal alignment

Answer: Root Mean Square Deviation (*RMSD*)

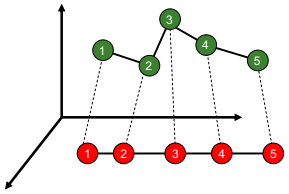
$$RMSD = \sqrt{\frac{\sum_i d_i^2}{n}}$$

n = number of atoms

d_i = distance between 2 corresponding atoms i in 2 structures

Root Mean Square Deviation

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{5} (X_{RED1} - X_{BLUE1})^2}{5}} \sim \frac{d_1 + d_2 + d_3 + d_4 + d_5}{5}$$



RMSD

Unit of RMSD => e.g. Ångstroms

- identical structures => *RMSD* = "0"
- similar structures => *RMSD* is small (1 – 3 Å)
- distant structures => *RMSD* > 3 Å

Pitfalls of RMSD

- all atoms are treated equally
  (*e.g. residues on the surface have a higher degree of freedom than those in the core*)
- best alignment does not always mean minimal RMSD
- significance of RMSD is size dependent

Alternative RMSDs

- aRMSD = best root-mean-square distance calculated over all aligned alpha-carbon atoms
- bRMSD = the RMSD over the highest scoring residue pairs
- wRMSD = weighted RMSD

Source: W. Taylor(1999), *Protein Science*, *8*: 654-665.

Structural Alignment Methods

- **Distance based methods**
  - DALI (Holm and Sander, 1993): Aligning 2-dimensional distance matrices
  - STRUCTAL (Subbiah 1993, Gerstein and Levitt 1996): Dynamic programming to minimize the RMSD between two protein backbones.
  - SSAP (Orengo and Taylor, 1990): Double dynamic programming using intra-molecular distance;
  - CE (Shindyalov and Bourne, 1998): Combinatorial Extension of best matching regions
- **Vector based methods**
  - VAST (Madej et al., 1995): Graph theory based SSE alignment;
  - 3dSearch (Singh and Brutlag, 1997) and 3D Lookup (Holm and Sander, 1995): Fast SSE index lookup by geometric hashing.
  - TOP (Lu, 2000): SSE vector superpositioning.
  - TOPSCAN (Martin, 2000): Symbolic linear representation of SSE vectors.
- **Both vector and distance based**
  - LOCK (Singh and Brutlag, 1997): Hierarchically uses both secondary structures vectors and atomic distances.

Basic DP (STRUCTAL)

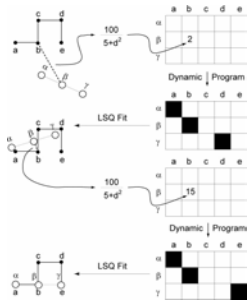
1. Start with arbitrary alignment of the points in two molecules A and B
2. Superimpose in order to minimize RMSD.
3. Compute a *structural alignment (SA) matrix* where entry (i,j) is the score for the structural similarity between the i^{th} point of A and the j^{th} point of B
4. Use DP to compute the next alignment.
   Gap cost = 0
5. Iterate steps 2--4 until the overall score converges
6. Repeat with a number of initial alignments

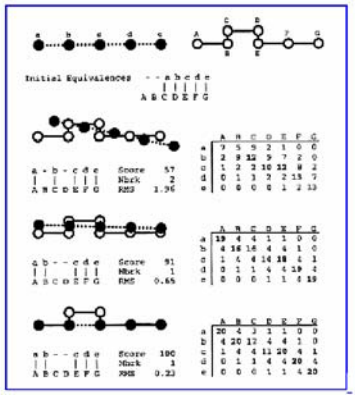
STRUCTAL

- Given
  2 Structures (A & B),
  2 Basic Comparison Operations

1. Given an alignment optimally **SUPERIMPOSE** A onto B
2. **Find an Alignment** between A and B based on their 3D coordinates

$S_{ij} = M/[1+(d_{ij}/d_0)^2]$
M and d_0 are constants





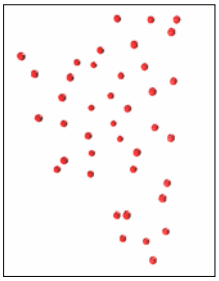
Our own method to compare structures

- CTSS (Curvature, Torsion, Secondary Structure)
- Uses dynamic programming on computed local structural features, not directly on 3D coordinates (like STRUCTAL did)

Protein Structure?

We use C_α coordinates to represent the protein structure.

```
HEADER    PHEROMONE          20-DEC-95   2ERL
.....................................
SEQRES   1  40  ASP ALA CYS GLU GLN ALA
.....................................
ATOM    1  N   ASP  1  -1.115   8.537   7.075
ATOM    2  CA  ASP  1  -1.925   7.470   6.547
ATOM    3  C   ASP  1  -2.009   6.333   7.522
ATOM    4  O   ASP  1  -1.467   6.394   8.624
ATOM    5  CB  ASP  1  -1.526   6.993   5.163
ATOM    6  N   ALA  2  -2.745   5.280   7.165
ATOM    7  CA  ALA  2  -2.945   4.152   7.987
ATOM    8  C   ALA  2  -1.606   3.448   8.305
ATOM    9  O   ALA  2  -1.440   3.010   9.454
ATOM   10  CB  ALA  2  -3.966   3.256   7.436
ATOM   11  N   CYS  3  -0.777   3.267   7.329
ATOM   12  CA  CYS  3   0.570   2.624   7.511
ATOM   13  C   CYS  3   1.328   3.308   8.626
ATOM   14  O   CYS  3   1.802   2.679   9.562
ATOM   15  CB  CYS  3   1.351   2.667   6.209
ATOM   16  SG  CYS  3   2.981   1.901   6.318
.....................................
```

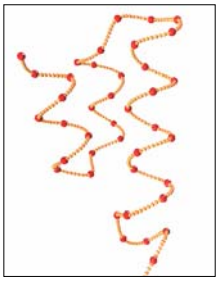


PDB File

Protein Structure

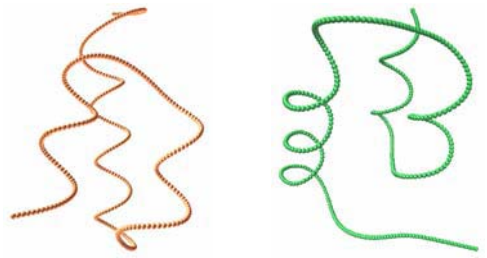
The C_α coordinates of a protein define a curve in 3D space.

```
HEADER    PHEROMONE          20-DEC-95   2ERL
.....................................
SEQRES   1  40  ASP ALA CYS GLU GLN ALA
.....................................
ATOM    1  N   ASP  1  -1.115   8.537   7.075
ATOM    2  CA  ASP  1  -1.925   7.470   6.547
ATOM    3  C   ASP  1  -2.009   6.333   7.522
ATOM    4  O   ASP  1  -1.467   6.394   8.624
ATOM    5  CB  ASP  1  -1.526   6.993   5.163
ATOM    6  N   ALA  2  -2.745   5.280   7.165
ATOM    7  CA  ALA  2  -2.945   4.152   7.987
ATOM    8  C   ALA  2  -1.606   3.448   8.305
ATOM    9  O   ALA  2  -1.440   3.010   9.454
ATOM   10  CB  ALA  2  -3.966   3.256   7.436
ATOM   11  N   CYS  3  -0.777   3.267   7.329
ATOM   12  CA  CYS  3   0.570   2.624   7.511
ATOM   13  C   CYS  3   1.328   3.308   8.626
ATOM   14  O   CYS  3   1.802   2.679   9.562
ATOM   15  CB  CYS  3   1.351   2.667   6.209
ATOM   16  SG  CYS  3   2.981   1.901   6.318
.....................................
```



PDB File

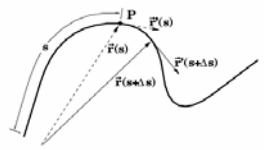
Matching Two Curves

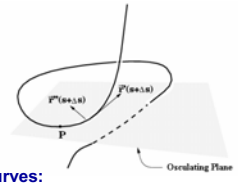


Are they similar?
Which parts?

Curvature and Torsion

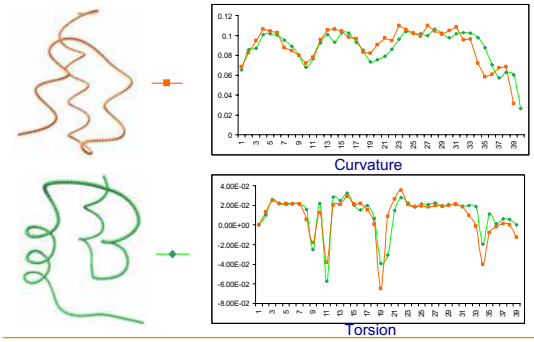
- **Curvature:** Measure of how far the curve deviates from being *linear*
- **Torsion:** Measure of how far the curve deviates from being *planar*





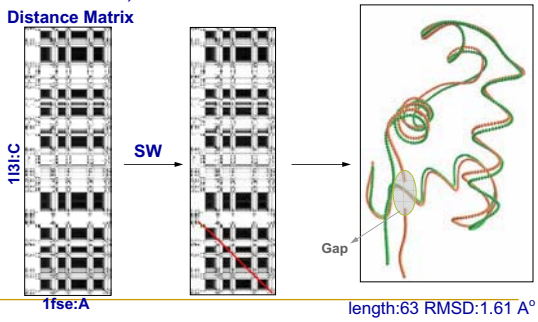
- **Fundamental Theorem of Space Curves:** *If two space curves have same curvature, torsion values from starting to ending positions, then they are the same curves (modulo translation and rotation)*

Curvature and Torsion



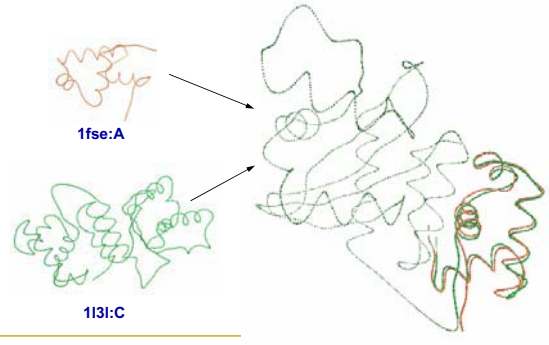
Alignment

- **Pairwise alignment by Smith-Waterman dynamic programming on the curvature, torsion distance matrices:**



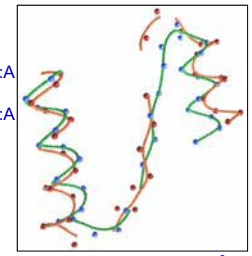
length:63 RMSD:1.61 A°

SW Alignment Result



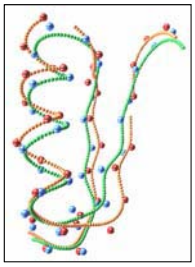
Sample Alignment Results

1b16:A & 1h05:A



length:35 RMSD:3.26 A°

1b16:A & 1qp8:A



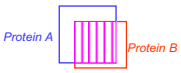
length:35 RMSD:1.58 A°

DALI Method

- Distance mAtrix aLIgnment
- Liisa Holm and Chris Sander, “Protein structure comparison by alignment of distance matrices”, *Journal of Molecular Biology Vol. 233*, 1993.
- Liisa Holm and Chris Sander, “Mapping the protein universe”, *Science Vol. 273*, 1996.
- Liisa Holm and Chris Sander, “Alignment of three-dimensional protein structures: network server for database searching”, *Methods in Enzymology Vol. 266*, 1996.

How DALI Works?

- Based on fact: similar 3D structures have similar intra-molecular distances.
- Background idea
  - Represent each protein as a 2D matrix storing intra-molecular distance.
  - Place one matrix on top of another and slide vertically and horizontally – until a common the sub-matrix with the best match is found.

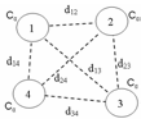


- Actual implementation
  - Break each matrix into small sub-matrices of fixed size.
  - Pair-up similar sub-matrices (one from each protein).
  - Assemble the sub-matrix pairs to get the overall alignment.

Structure Representation of DALI

- 3D shape is described with a *distance matrix* which stores all *intra-molecular distances* between the C_α atoms.
- Distance matrix is independent of coordinate frame.
- Contains enough information to re-construct the 3D coordinates.

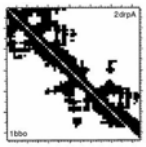
Protein A



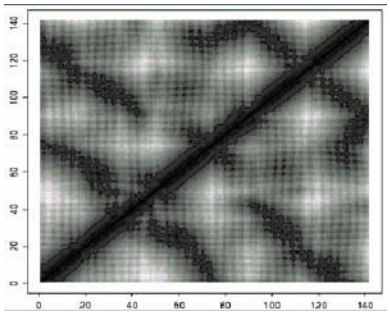
Distance matrix for Protein A

	1	2	3	4
1	0	d_{12}	d_{13}	d_{14}
2	d_{12}	0	d_{23}	d_{24}
3	d_{13}	d_{23}	0	d_{34}
4	d_{14}	d_{24}	d_{34}	0

Distance matrix for 2drpA and 1bbo



Intra-molecular distance for myoglobin



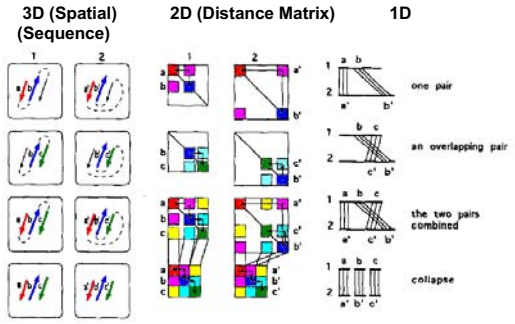
DALI Algorithm

1. Decompose distance matrix into elementary *contact patterns* (sub-matrices of fixed size)
   - Use hexapeptide-hexapeptide contact patterns.
2. Compare contact patterns (pair-wise), and store the matching pairs in *pair list.*
3. Assemble pairs in the correct order to yield the overall alignment.

Assembly of Alignments

- Non-trivial combinatory problem.
- Assembled in the manner (AB) – (A’B’), (BC) – (B’C’), . . . (i.e., having one overlapping segment with the previous alignment)
- Available Alignment Methods:
  - *Monte Carlo* optimization
  - Brach-and-bound
  - Neighbor walk

Schematic View of DALI Algorithm

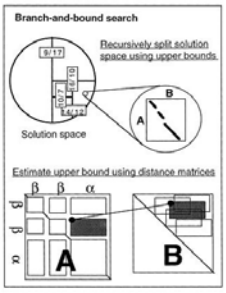


Monte Carlo Optimization

- Used in the earlier versions of DALI.
- Algorithm
  - Compute a similarity score for the current alignment.
  - Make a random trial change to the current alignment (adding a new pair or deleting an existing pair).
  - Compute the change in the score (ΔS).
  - If $\Delta S > 0$, the move is always accepted.
  - If $\Delta S <= 0$, the move may be accepted by the probability $\exp(\beta * \Delta S)$, where β is a parameter.
  - Once a move is accepted, the change in the alignment becomes permanent.
  - This procedure is iterated until there is no further change in the score, i.e., the system is converged.

Branch-and-bound method

- Used in the later versions of DALI.
- Based on *Lathrop and Smith's* (1996) *threading* (sequence-structure alignment) algorithm.
- *Solution space* consists of all possible placements of residues in protein A relative to the segment of residues of protein B.
- The algorithm recursively split the solution space that yields the highest upper bound of the similarity score until there is a single alignment trace left.



LOCK

- Uses a hierarchical approach
- Larger secondary structures such as helixes and strands are represented using vectors and dealt with first
- Atoms are dealt with afterwards
- Assumes large secondary structures provide most stability and function to a protein, and are most likely to be preserved during evolution

LOCK (Contd.)

- Key algorithm steps:
  1. Represent secondary structures as vectors
  2. Obtain initial superposition by computing local alignment of the secondary structure vectors (using dynamic programming)
  3. Compute atomic superposition by performing a greedy search to try to minimize *root mean square deviation* (a RMS distance measure) between pairs of nearest atoms from the two proteins
  4. Identify "core" (well aligned) atoms and try to improve their superposition (possibly at the cost of degrading superposition of non-core atoms)
- Steps 2, 3, and 4 require iteration at each step

Alignment of SSEs

- Define an orientation-dependent score and an orientation-independent score between SSE vectors.
- For every pair of query vectors, find all pairs of vectors in database protein that align with a score above a threshold. Two of these vectors must be adjacent. Use orientation independent scores.
- For each set of four vectors from previous step, find the transformation minimizing rmsd. Apply this transformation to the query.
- Run dynamic programming using both orientation-dependent and orientation-independent scores to find the best local alignment.
- Compute and apply the transformation from the best local alignment.
- Superpose in order to minimize rmsd.

Atomic superposition

- Loop
  - find matching pairs of C_α atoms
  - use only those within 3 A
  - find best alignment
- until rmsd does not change

Core identification

- Loop
  - find the best core (symmetric nns) and align; remove the rest
- until rmsd does not change

VAST

- Begin with a set of nodes (a,x) where SSEs a and x are of the same type
- Add an edge between (a,x) and (b,y) if angle and distance between (a,b) is same as between (x,y)
- Find the maximal clique in this graph; this forms the initial SSE alignment
- Extend the initial alignment to C_α atoms using Gibbs sampling
- Report statistics on this match

Quality of a structure match

- Statistical theory similar to BLAST
- Compare the likelihood of a match as compared to a random match
- Less agreement regarding score matrix
  - z-scores of CE, DALI, and VAST may not be compatible