DALI: Distance-matrix ALIgnment

Holm, L. and Sander, C. (1996)
Mapping the Protein Universe
Science vol. 273, 595-602.

The objective of shape comparison in DALI is to assign a one-to-one equivalence between the residues, where non-matching residues can be skipped in either chain.

This is done by finding similar patterns in distance matrices.

Constructing distance matrices (or “contact maps”) is easy;
finding maximal matching sub-matrices is hard.

Two algorithms in DALI

Scan for obvious similarities using a fast (but, in general, less accurate) algorithm, then rescan for more subtle similarities using more sophisticated (but slower) algorithms.

A) Fast heuristic 3D lookup (“hashing”)

Catches easy-to-find structural similarities.

Represent secondary structure elements by 3D line segments; match vector relationships from the query protein with a stored list; when enough matches are found with a database protein, sample a limited set of superpositions.

B) Branch-and-bound algorithm

Guaranteed to find the global optimum, but slower
(worst case: exponential number of steps).

Find the best matching sub-matrices for proteins A and B;
then recursively split the solution sub-space.

Shape comparison in DALI

(i) a suitable representation:

list of Cα atoms described by their x, y and z coordinates.

(ii) an objective function to be optimised:

accommodate the largest possible number of equivalent points within small deviations in position (typically less than 2 to 3 *angstrom*).

(iii) a comparison algorithm:

find matching sub-matrices and merge these into larger consistent blocks of agreement by removing intervening rows and columns.

(iv) appropriate decision rules:

statistical significance of comparison score (Z-score);
equivalent sets of residues (structural alignment);
3D view of the matched parts superimposed.

Problems when searching a protein structure database

(Want to perform all-against-all comparison)

Unequal representation of protein families.

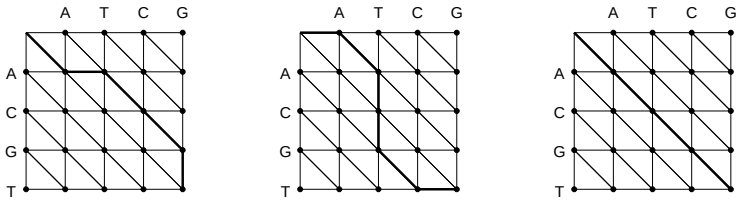
Some redundancy can be eliminated by removing proteins with mutual sequence identity greater than 25%.
But many structurally similar proteins remain.

The problem of domains.

Similar sub-structures recur between several proteins.

Today we can identify sets of domains with distinct folds from resources like CATH and SCOP.

Each path represents an alignment




```
A-CGT
| ||
ATCG-
```



```
-ACGT-

AT--CG
```



```
ACGT
|
ATCG
```

- Vertical steps add a gap to the horizontal sequence
- Horizontal steps add a gap to the vertical sequence

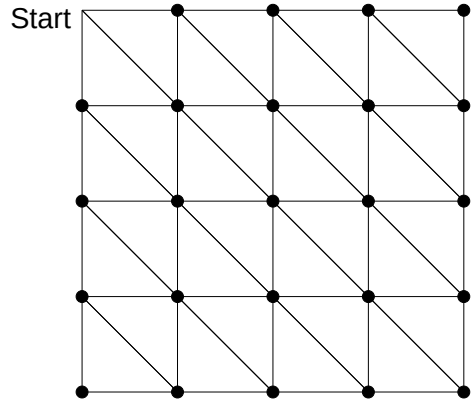
Pairwise global alignment (Needleman-Wunsch algorithm)

Rigorous algorithms use dynamic programming to find an optimal alignment.

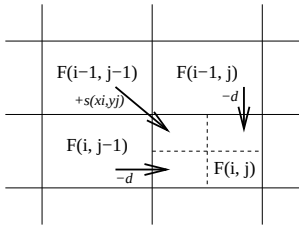
- match score
- mismatch score
- gap penalty

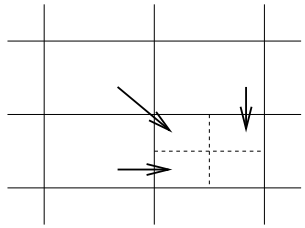
$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

How many paths?

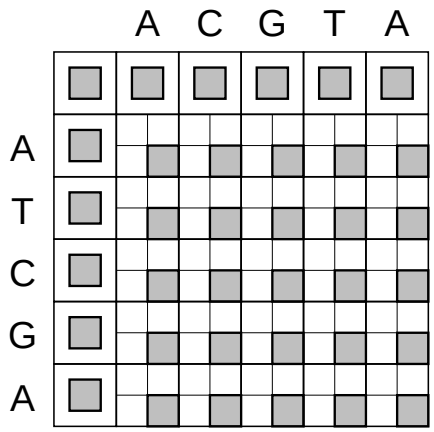


Dynamic programming





Score matrix



Axes of secondary structure elements

[Singh A.P. and Brutlag, D.L. (1997) “Hierarchical protein structure superposition using both secondary structure and atomic representations”, Proc. Int Conf. Intell. Syst. Mol. Biol., 5, 284-293]

Strand:

$$X_{start} = (X_i + X_{i+1})/2$$

$$X_{end} = (X_j + X_{j-1})/2$$

Helix:

$$X_{start} = (0.74 * X_i + X_{i+1} + X_{i+2} + 0.74 * X_{i+3})/3.48$$

$$X_{end} = (0.74 * X_j + X_{j-1} + X_{j-2} + 0.74 * X_{j-3})/3.48$$

BLOSUM62

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