Università degli Studi di Padova
1222·2022 800 anni

Dipartimento Matematica

Data Science
University of Padova

STRUCTURAL ALIGNMENTS

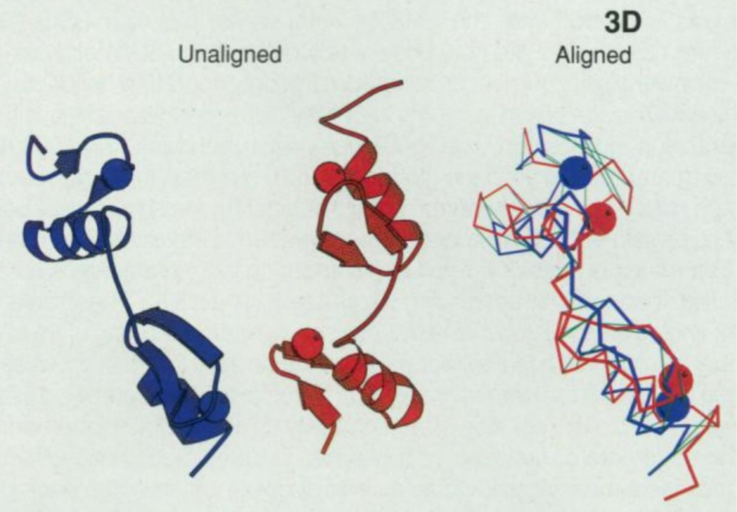
Master of Science in Data Science

Damiano Piovesan

Compare structures

Which points in A are equivalent to points in B?

- Suitable representation of the object to study
- Function to be optimized
- Comparison algorithm
- Rules to evaluate the significance of the result



Superposition Vs alignment

Superposition

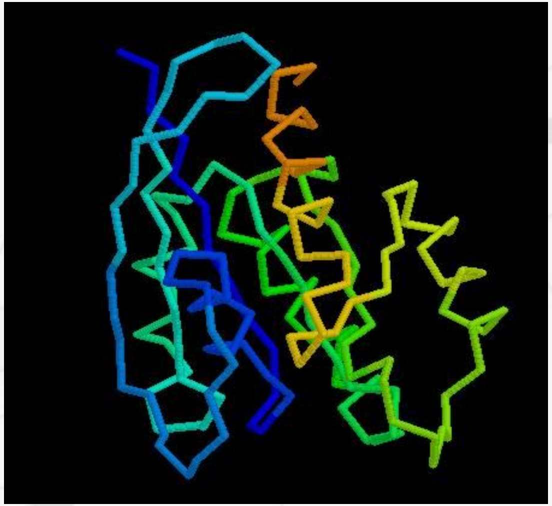
- What are the “aligned atoms” is pre-defined
- Based on translation and rotation transformations
- Used to compare different conformation of the same structure

Structural alignment

- No *a priori* knowledge of equivalent positions
- NP-hard problem. N^M possible alignments (to align N residues onto a structure of M segments)
- Used to compare different / related proteins
- Database search. Structural (evolutionary) relationships

Representation

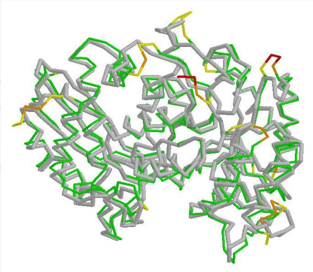
Simplify the problem, consider only one atom (point) for each residue (example Cα)

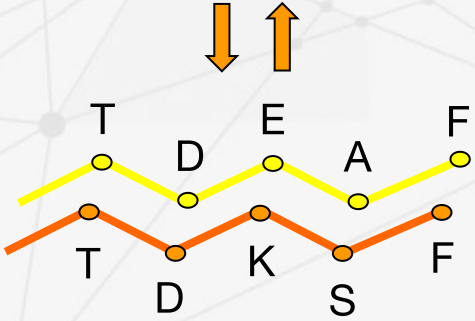


Target function

- Minimize the **Root-mean-square deviation** (RMSD)
- $\boldsymbol{r}_{ai}$ and $\boldsymbol{r}_{bi}$ are the coordinates of the *i* **equivalent atoms** in structure *a* and *b*
- ***n*** is the number of paired atoms in the structure

$$RMSD = \sqrt{\frac{\sum (r_{ai} - r_{bi})^2}{n}}$$





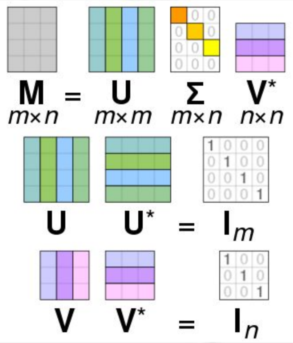
Superposition

- Place the barycenter of the two proteins at the origin of the coordinate system (translation)
- Compute the optimal rotational matrix (least RMSD)

Kabsch algorithm

- Build the **3xN** matrices **X** and **Y** containing, for the sets x and y respectively, the coordinates for each of the **N atoms** after centering the atoms by subtracting the centroids
- Compute the **cross-covariance** matrix **C = XY^T**
- Compute the **SVD** (Singular Value Decomposition) of **C = VSW^T**
- Compute **d** = sign(det(C)), to see if it is left/right handed
- Compute the **optimal rotation U** as

$$U = W \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} V^T$$



References

SVD

https://en.wikipedia.org/wiki/Singular_value_decomposition

Linear algebra (3Blue1Brown, Grant Sanderson)

https://youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab

Kabsch algorithm maths

https://cnx.org/contents/HV-RsdwL@23/Molecular-Distance-Measures

Kabsch in Python

https://github.com/charnley/rmsd

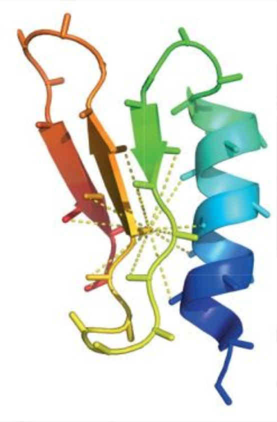
Structural alignments

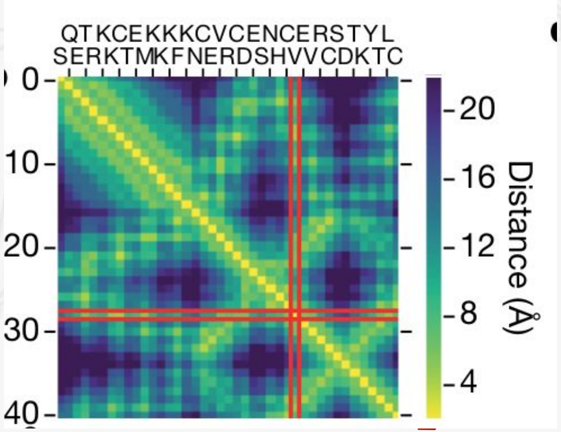
- SSAP - Sequential Structure Alignment Program
  - Compares residue vectors (used in the CATH database)
- DALI
  - Compare distance matrices
- CE - Combinatorial Extension
  - Compare aligned fragment pairs (AFP)
- TM-align - Template Modelling align
  - Heuristic dynamic programming iterations (state-of-the art)

SSAP - Sequential Structure Alignment Program

First tentative

- **Distance** of a given residue to all other residues in the same structure
- No need of superposition
- Not dependent on the coordinate reference frames
- Constant between equivalent positions in different structures
- Invariant under rotation
- **Limitation** → Similar distances between pairs of atoms that might be in completely different relative directions

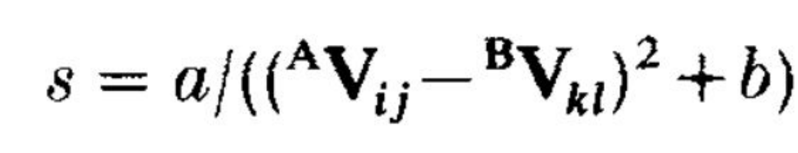




SSAP - Sequential Structure Alignment Program

Second tentative

- Comparison of **interatomic vectors** rather than simple distance
- Local frame of reference for every residue
- X-axis → N - C
- Y-axis by the C_β- H
- Z-axis perpendicular to Y-axis and X-axis
- Parameters a = 50, b = 2

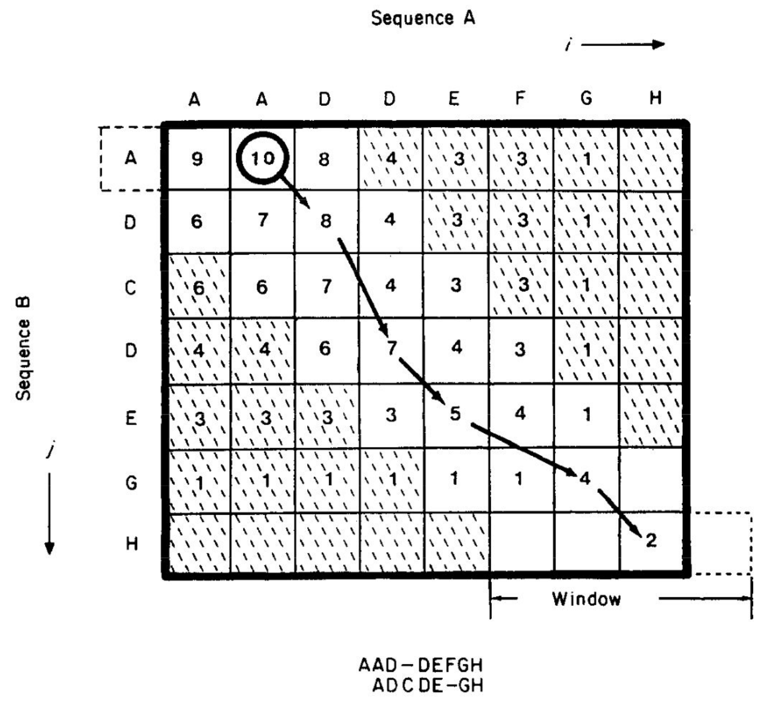


$$s = a/((^{\mathrm{A}}\mathbf{V}_{ij} - {}^{\mathrm{B}}\mathbf{V}_{kl})^2 + b)$$

SSAP - Sequential Structure Alignment Program

- Dynamic programming
- Start from lower-right corner and go up to upper-left
- Match 2, gap cost -1
- Trace-back

Equivalent to the Needleman & Wunsch algorithm for sequence alignments



	A	A	D	D	E	F	G	H
A	9	10	8	4	3	3	1	
D	6	7	8	4	3	3	1	
C	6	6	7	4	3	3	1	
D	4	4	6	7	4	3	1	
E	3	3	3	3	5	4	1	
G	1	1	1	1	1	1	4	
H								2

AAD–DEFGH
ADCDE–GH

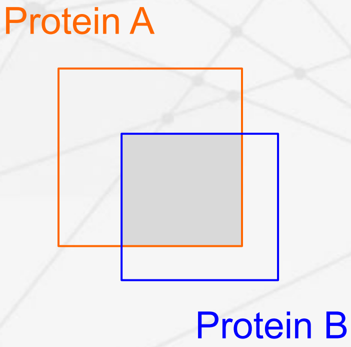
DALI

Fast search algorithm (database search)

- Compare secondary structure elements (SSEs)

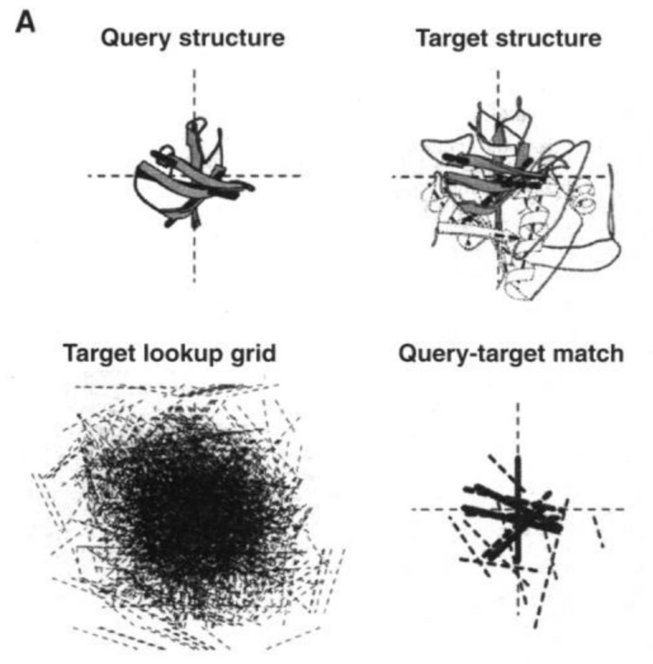
Accurate algorithm (pairwise alignment)

- Compare distance matrices (one for each structure)



DALI - Fast algorithm

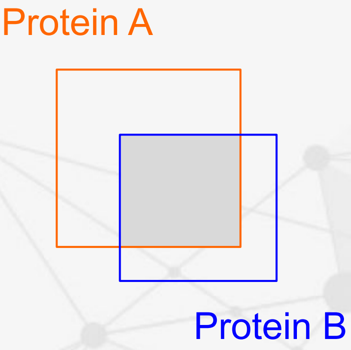
- Each pair of SSEs (within 12 A) define a different coordinate frame (target structure / lookup grid)
- One SSEs is centered on the origin and aligned to the Y axis and rotated so that the second SSE is in the positive x-y plane
- The lookup grid is probed with the query structure (query-target match)

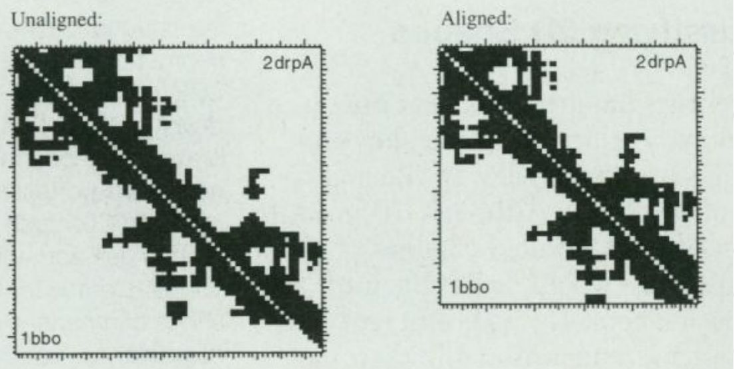


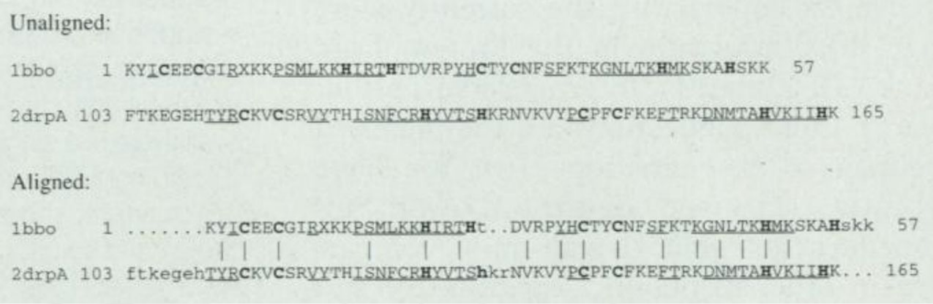
DALI

DALI - Distance matrix ALIgnment

- Similar structures have similar distance matrices
- Place one matrix on top of another and slide vertically and horizontally until the sub-matrix with the best match is found







Unaligned:

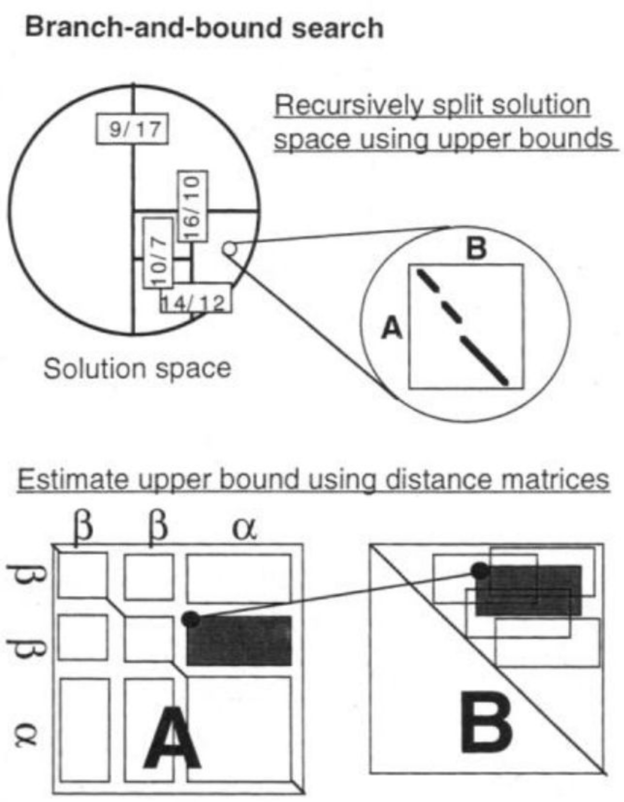
```
1bbo    1 KYICEECGIRXKKPSMLKKHIRTHTDVRPYHCTYCNFSFKTKGNLTKHMKSKAHSKK   57

2drpA 103 FTKEGEHTYRCKVCSRVYTHISNFCRHYVTSHKRNVKVYPCPFCFKEFTRKDNMTAHVKIIHK 165
```

Aligned:

```
1bbo    1 .......KYICEECGIRXKKPSMLKKHIRTHt..DVRPYHCTYCNFSFKTKGNLTKHMKSKAHskk   57
                  | |  |       |    |  |     |  |  |  |   |  | | | | |
2drpA 103 ftkegehTYRCKVCSRVYTHISNFCRHYVTShkrNVKVYPCPFCFKEFTRKDNMTAHVKIIHK... 165
```

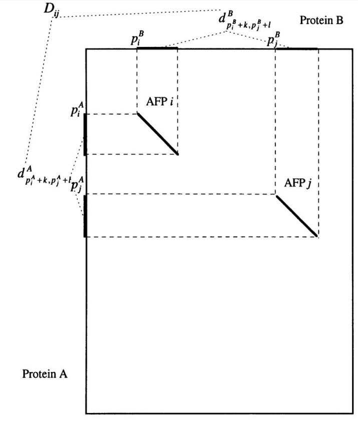
DALI - Accurate algorithm

- Proteins structures are represented as distance matrices
- Internal squares correspond to secondary structure segments
- Test all possible placements of residue in B relative to segments in A
- Recursively split the solution space until there is a single alignment trace
- The best match maximize pair score (sum of similarity of distances)

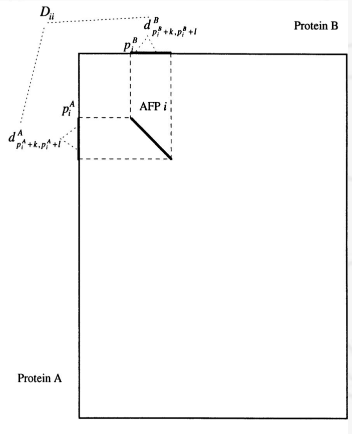

Branch-and-bound search
9/17
16/10
10/7
14/12
Solution space
Recursively split solution space using upper bounds
B
A
Estimate upper bound using distance matrices
β
β
α
β
β
α
A
B

Combinatorial Extension (CE)

- Find the longest continuous path P of **aligned fragment pairs (AFPs) of size m** in a similarity matrix
- AFP length → **8 residues**, max gap length → 30
- Heuristic for path extension, consider only the best AFP
- Final evaluation, RMSD, Z-score



Distance between the combination of two AFPs, one already in the path and one to be added



Used to evaluate a single AFP

TM-align & TM-score

- A small number of **local deviations** could result in a high RMSD, even when the global topologies of the compared structures are similar
- **TM-score**, weights the residue pairs at smaller distances relatively stronger than those at larger distances
- For random structures is the average distance between an aligned pair of residues
- Not dependent on the protein size

$$\text{TM-score} = \text{Max}\left[\frac{1}{L_{\text{Target}}}\sum_{i}^{L_{\text{ali}}}\frac{1}{1+\left(\frac{d_i}{d_0\left(L_{\text{Target}}\right)}\right)^2}\right]$$

$$d_0\left(L_{\text{Target}}\right) = 1.24\sqrt[3]{L_{\text{Target}}-15}-1.8$$

TM-align

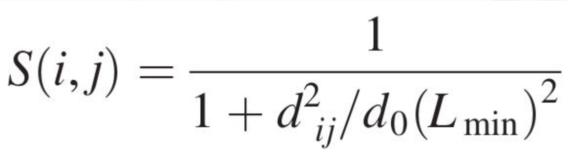
Initial structural alignment

- Align the secondary structures with dynamic programming (1 match, -1 gap opening).
- Alpha, beta, coil states are assigned based on coordinates of neighbouring residues
- Gapless threading against the largest structure using TM-score as comparison metric
- DP with gap-opening penalty of -1

TM-align

Heuristic iteration

- Rotate the structure by the TM-score rotation matrix
- Apply DP with gap opening matrix and with a score similarity matrix equal to



$$S(i,j) = \frac{1}{1 + d^{2}_{ij}/d_0(L_{\min})^2}$$

- Repeat until the alignment becomes stable