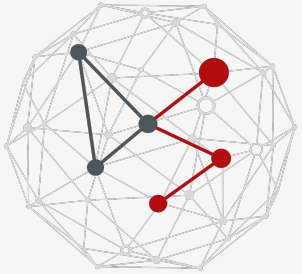


DATA SCIENCE
UNIVERSITY OF PADOVA

STRUCTURAL COMPARISON

Master of Science in Data Science

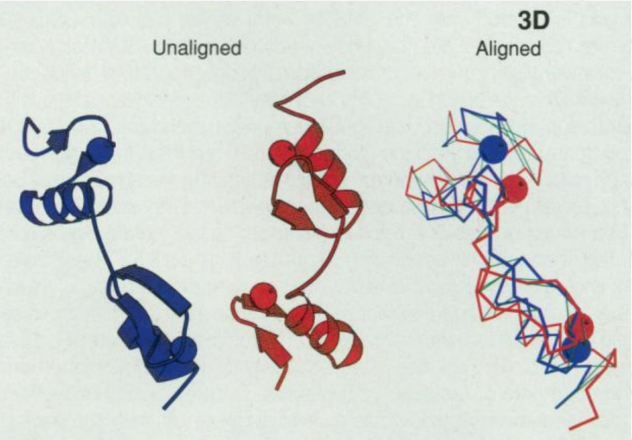
Damiano Piovesan

Compare structures

DIPARTIMENTO MATEMATICA

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

DIPARTIMENTO MATEMATICA

Superposition

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

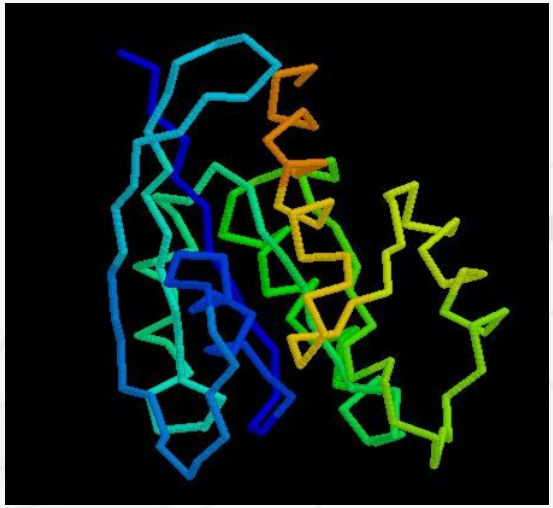
Structural alignment

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

Representation

DIPARTIMENTO MATEMATICA

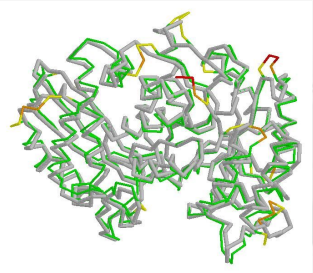
Simplify the problem, consider only one atom (point) for each residue (example Cα)

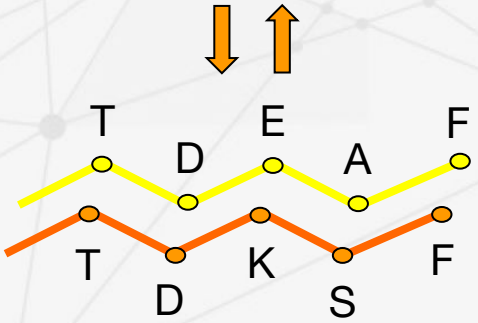


Target function

- Root-mean-square deviation
- r_{ai} and 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

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





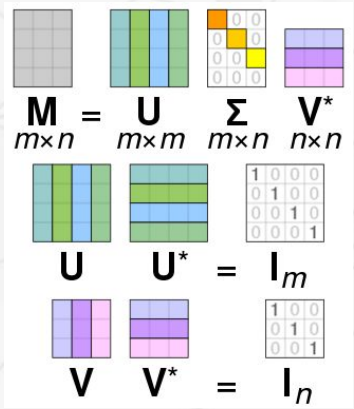
Superposition

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

Kabsch algorithm

DIPARTIMENTO MATEMATICA

- Center the coordinates (**N atoms**) of the of the structures (subtract centroids coordinates)
- Build the **3 x N** matrices (**P** and **Q**) with the atomic coordinates
- Compute the **cross-covariance** matrix $\mathbf{M = P^T Q}$
- Compute the **SVD** (Singular Value Decomposition) of $\mathbf{M = U\Sigma V^T}$
- Compute $\mathbf{d = sign(det(VU^T))}$, to see if the coordinate system is left/right handed
- Compute the **optimal rotation R** as $R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} U^T$
- **R** minimizes $\sum_{k=1}^{N} |Rq_k - p_k|$, where q_k and p_k are rows in Q and P



References

Kabsch algorithm

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

Kabsch in Python

https://github.com/charnley/rmsd

SVD

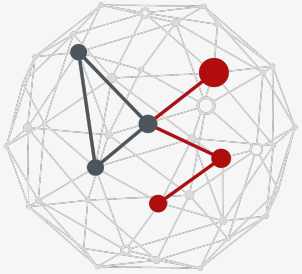
https://en.wikipedia.org/wiki/Singular_value_decomposition

Linear algebra (3Blue1Brown, Grant Sanderson)

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

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
MATEMATICA



DATA SCIENCE
UNIVERSITY OF PADOVA

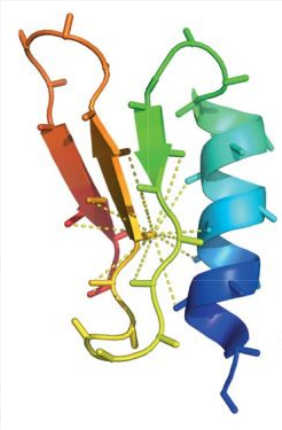
DISTANCE MATRIX & CONTACT MAP

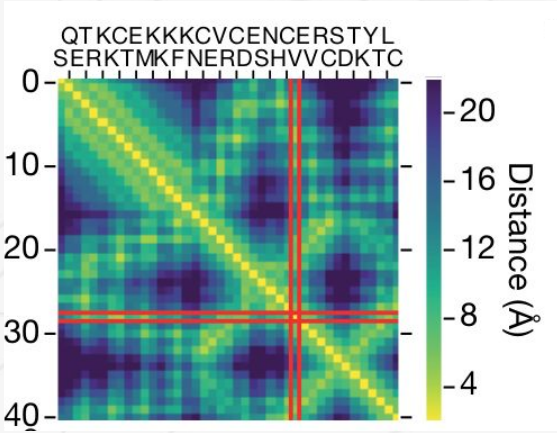
Master of Science in Data Science

Damiano Piovesan

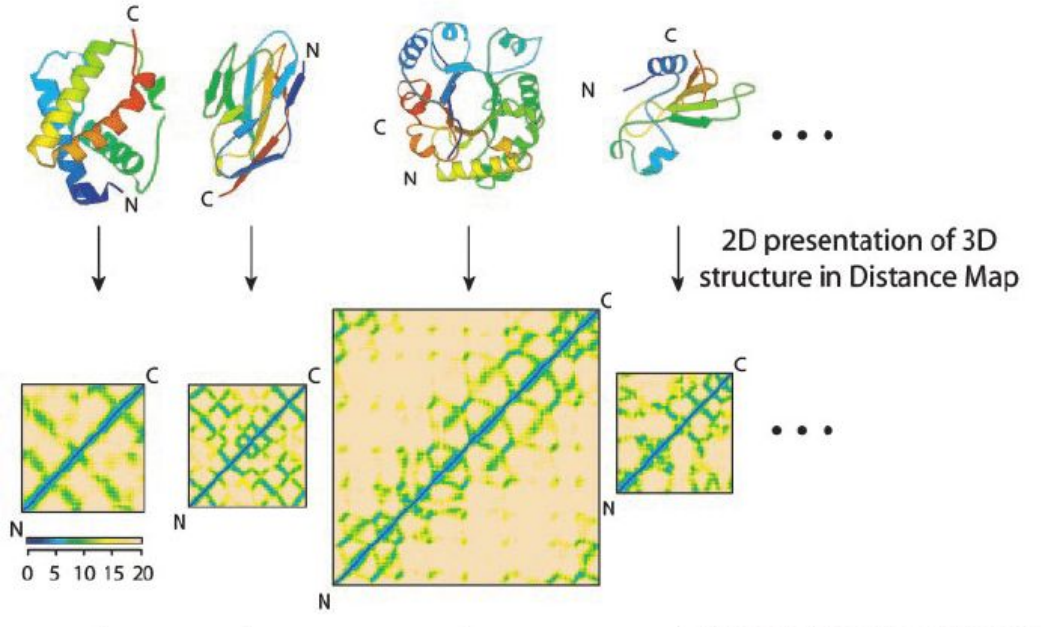
Distance matrix

- L × L matrix, where L is the protein length
- Each element represents the distance between two atoms in the 3D space
- Can be built considering all atoms or just a representative atom for each amino acid (C_α, C_β, center of mass, …)



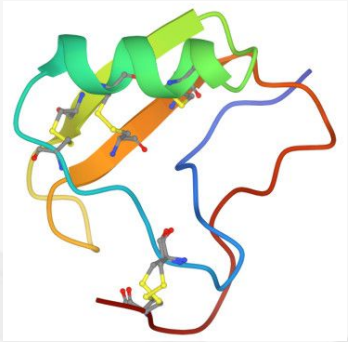


Distance matrix examples

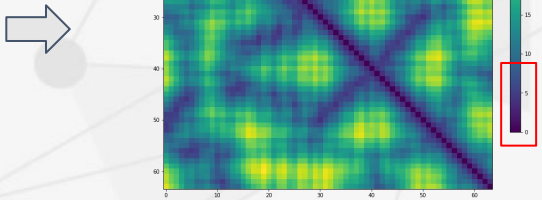


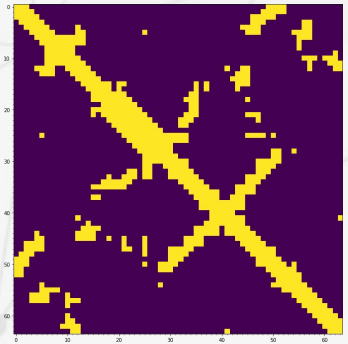
Contact map

- Binary matrix
- 1 → contact, 0 → no contact
- Calculated from the distance matrix by applying a distance cutoff (e.g. 8 Å)



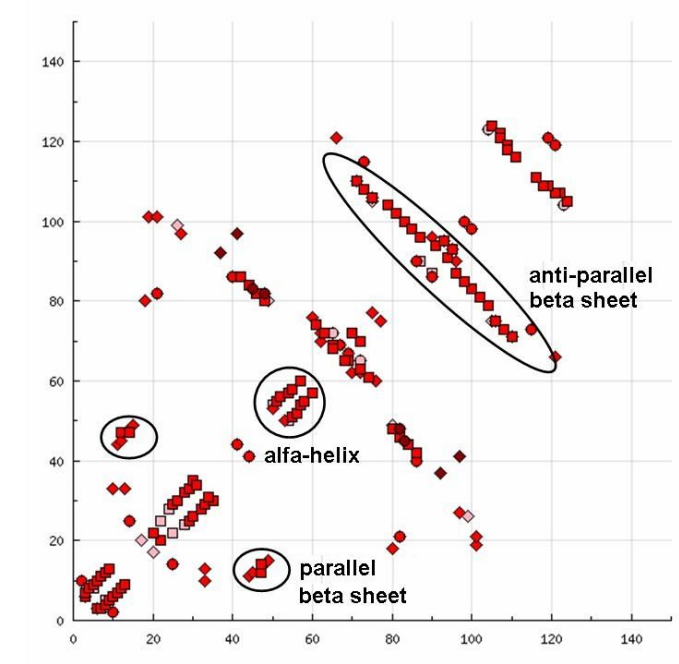
1AHO





Contact map interpretation

- Contiguous contacts close and parallel to the main diagonal → **helices**
- Contiguous contacts parallel but distant from the main diagonal → **parallel beta sheets**
- Contiguous contacts perpendicular to the main diagonal → **anti-parallel beta sheets**

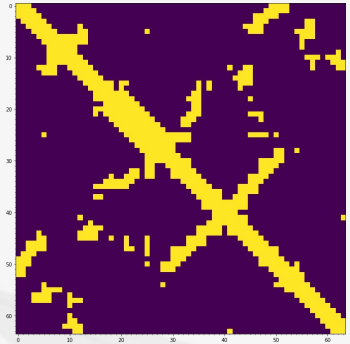


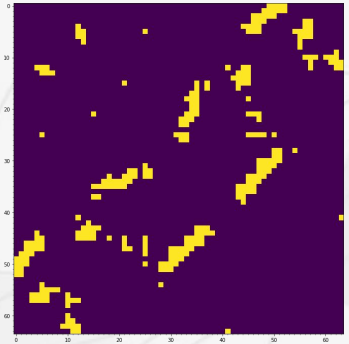
Contact map - sequence separation

DIPARTIMENTO MATEMATICA

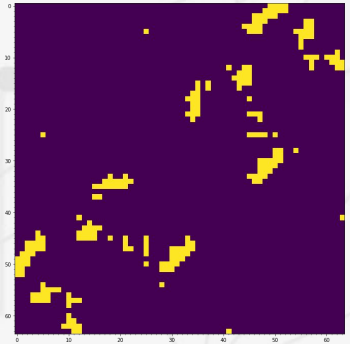
Long
Middle
Short

VKDGYIVDDVNCTYFCGRNAYCNEECTKLKGESGYCQWASPYGNACYCYKLPDHVRTKGPGRCH

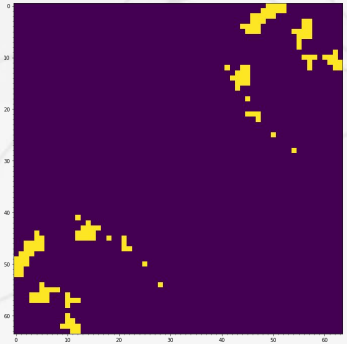




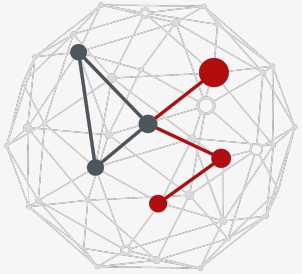
i - j ≥ 6



i - j ≥ 12



i - j ≥ 24



DATA SCIENCE
UNIVERSITY OF PADOVA

STRUCTURAL ALIGNMENTS

Master of Science in Data Science

Damiano Piovesan

Structural alignments

DIPARTIMENTO MATEMATICA

Objective → assignment of **one-to-one equivalence** between (C_α) atoms

- **Nonmatching residues** can be skipped in either chains
- In some applications the **linear order** of equivalent pairs along the sequence is maintained → the **continuity of the polymer chain** is considered a key aspect of shape
- Accommodate the **largest possible number of equivalent points** within small deviations in position, typically less than **2 to 3 Å**

Structural alignments

DIPARTIMENTO MATEMATICA

- **SSAP** - Sequential Structure Alignment Program (C. Orengo, 1988)
  - Dynamic programming (used in the CATH database)
- **DALI** - Distance matrix ALIgnment (C. Sander, 1996)
  - Distance matrices (used in FSSP database)
- **CE** - Combinatorial Extension
  - Aligned fragment pairs (AFP)
- **TM-align** - Template Modelling align
  - Heuristic dynamic programming iterations (state-of-the art)

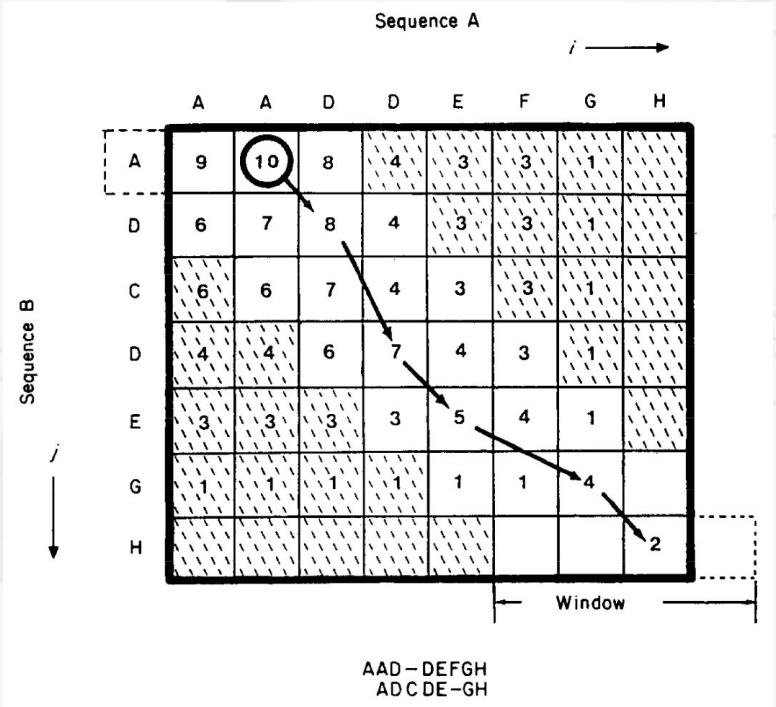
SSAP - Sequential Structure Alignment Program

- Combines sequence-base comparison methods (**dynamic programming**) with structural information
- Sequence-based comparisons are based on Dayhoff matrix (PAM) which provides a measure of **similarity between pairs of amino acids**
- SSAP develops a distance measure that is not dependent on the coordinates reference frame
    - Version 1 → **Cartesian distance** of a given residues to all other residues
    - Version 2 → Distance between a **vector representation** of residues (one vs. all other residues)

SSAP

Equivalent to the **Needleman & Wunsch** algorithm for sequence alignments

- Dynamic programming
- Trace-back



SSAP

Version 1

- **Distance** of a given residue to all other residues in the same structure

$$s = a/(|{}^{A}d_{ij} - {}^{B}d_{kl}| + b) \quad \rightarrow \quad S_{ik} = \sum_{m=-n}^{+n} a/(|{}^{A}d_{i,i+m} - {}^{B}d_{k,k+m}| + b)$$

- 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

Version 2

- Comparison of **interatomic vectors** rather than atomic distances

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

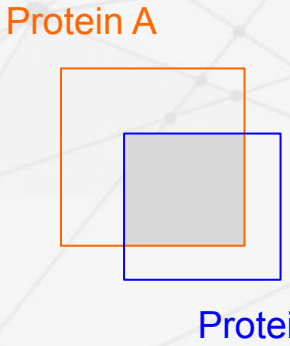
- Local frame of reference for every residue → recognizes different orientations in different coordinate frames
  - X-axis → N - C
  - Y-axis → C_β - H
  - Z-axis perpendicular to Y-axis and X-axis
- Parameters a = 50, b = 2

DIPARTIMENTO MATEMATICA

DALI - Distance matrix ALIgnment

Example → discovery of a **common structural core** in two apparently **unrelated enzymes** from different species with apparently different amino acid sequences

- Compare **distance matrices** by sliding one on top of the other and identify **similar substructures**
- **Combinatorial optimization problem** → merging matching submatrices to larger consistent blocks by the removal of intervening rows and columns

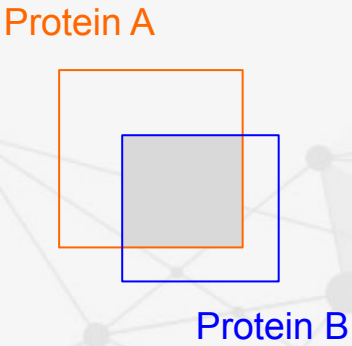


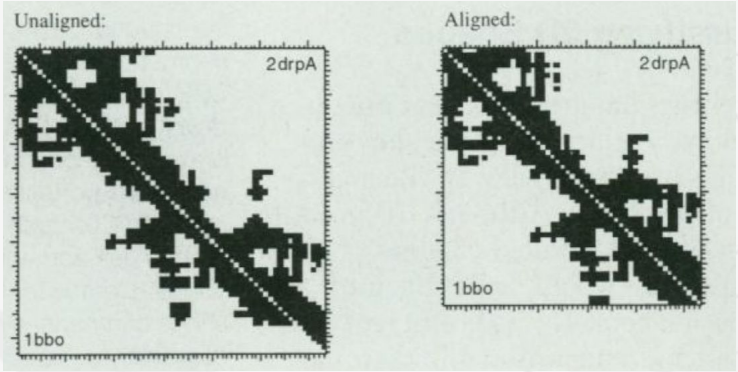
DALI

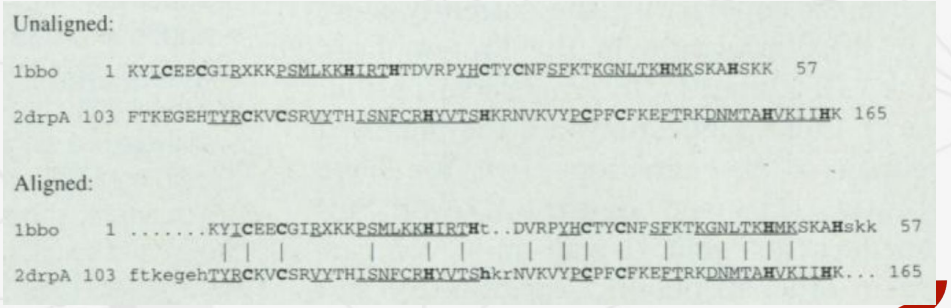
DIPARTIMENTO MATEMATICA

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







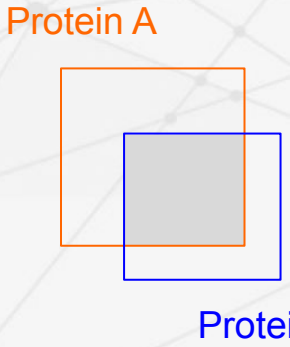
DALI

Fast algorithm (database search)

- Compare **secondary structure elements** (**SSEs**)

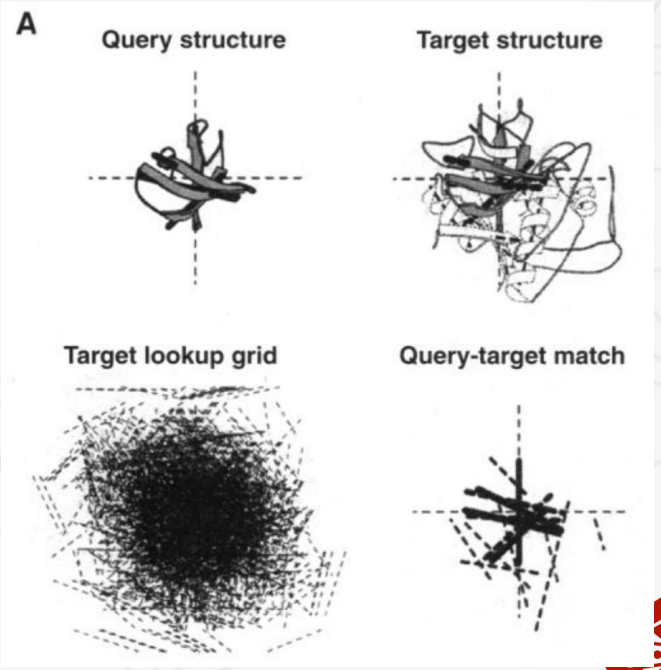
Accurate algorithm (pairwise alignment)

- Compare **distance matrices** (one for each structure)



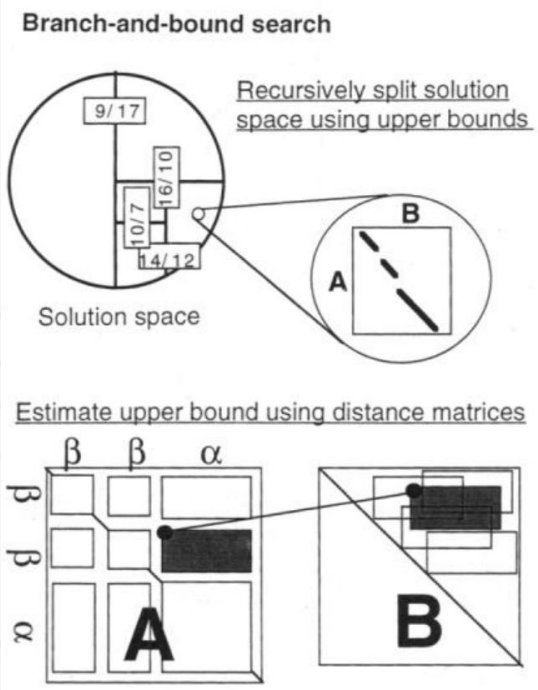
DALI - Fast algorithm

- Each **pair of SSEs** (within 12 A) defines a different coordinate frame
- 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



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)

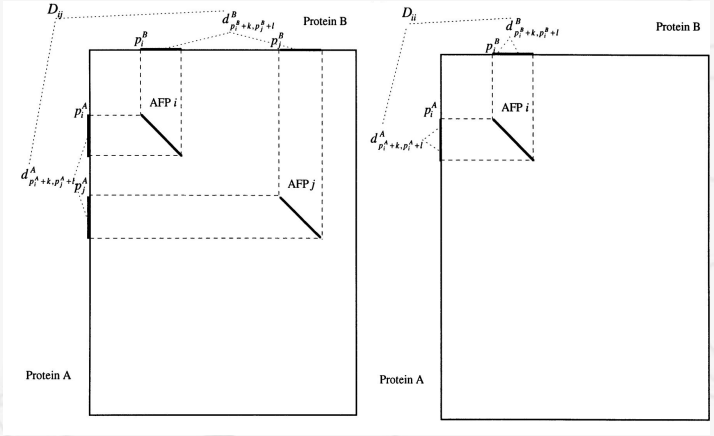


CE - Combinatorial Extension

- **Combinatorial extension (CE)** of an alignment path defined by **alignment fragment pairs (AFP)**
- Based on **local geometry** rather than global features (SS orientation, topology, …)

CE

- Find the longest continuous path P of **aligned fragment pairs (AFPs)** of **size *m*** in a similarity matrix
- Similarity evaluation
  a. Evaluation of individual AFP
  b. Evaluation of independent AFP
  c. RMSD of the optimal superposition of the aligned path
- Parameters
  a. AFP length (*m*) → **8 residues**
  b. Max gap length → **30**

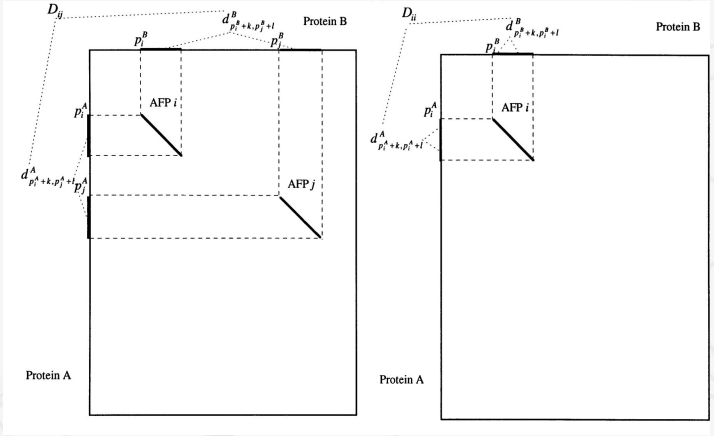


Distance between the combination of two AFPs, one already in the path and one to be added

Used to evaluate a single AFP

CE

- Path extension strategies
  a. Consider **all AFPs** that extend the path and satisfy the similarity criteria
  b. Consider only the **best AFP** which extends the path and satisfies the similarity criteria
  c. Use some intermediate strategy
- **Starting point** → All possible starting points in the similarity matrix
- **Extension** → gap are not evaluated, statistical significance (s.s.) not evaluated
- **Evaluation** of the longest path → (**z-score**) Probability of finding an alignment path of the same length with the same or smaller number of gaps and distance from a random comparison of structures using a non-redundant set

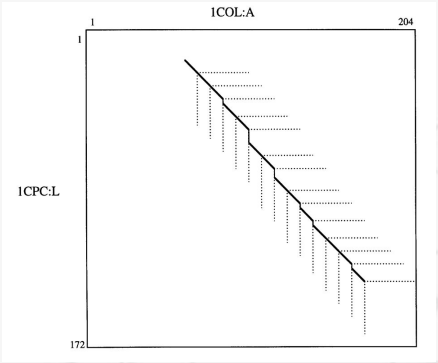


Distance between the combination of two AFPs, one already in the path and one to be added

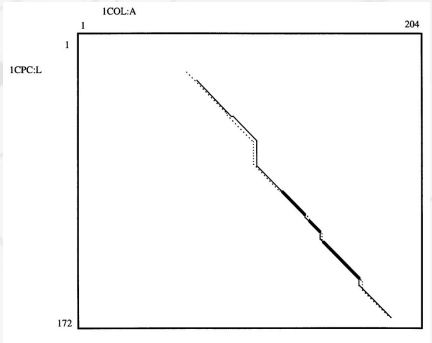
Used to evaluate a single AFP

CE

- **Extension** → gap are not evaluated, statistical significance (s.s.) not evaluated
    a. Only AFPs with RMSD < 3 Å are considered
    b. New AFPs are less than 4 Å distant from the previous AFP
    c. Decision on path extension/termination → The average distance between AFPs in the path is less than 4 Å
- **Final optimization**
    a. The 20 best paths are evaluated (RMSD)
    b. Gaps relocated (± m/2) (RMSD)
    c. Dynamic programming on the distance matrix of the residues (gap penalty 5 initiation, 0.5 extension)



Extension



Optimization

TM-score (TM-align method)

- A small number of **local deviations** could result in a high **RMSD**, even when the global topologies of the compared structures are similar
- The average **RMSD** of randomly related proteins depend on the length of compared structures
- **TM-score**, weights the residue pairs at smaller distances relatively stronger than those at larger distances
- **TM-score** is more sensitive to the **global topology** than to the **local structural variations**
- **TM-score** is normalized in a way that the score magnitude relative to **random structures** is not dependent on the **protein's size**

TM-score (TM-align method)

- For random structures is the average distance between an aligned pair of residues
- Does not depend on the protein size
- The maximum is taken over all possible structure superpositions of the model and template (or some sample thereof)

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

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

$\mathbf{L_{Target}}$ is the length of target protein

$\mathbf{L_{ali}}$ is the number of aligned residues

$\mathbf{d_i}$ is the distance between the ith pair of aligned residues

The **maximum** is taken over all possible structure superpositions of the model and template (or some sample thereof)

TM-align

Initial structural alignment methods

1. Align the **secondary structures** with **dynamic programming (DP)**
    - 1 match, -1 gap opening
    - Alpha, beta, coil states are assigned based on coordinates of neighbouring residues (5 residues window)
2. **Gapless threading** against the larger structure using **TM-score** as comparison metric instead of RMSD
3. Same as method 1, but the score in the DP is half/half the score matrix of method 1 and method 2

TM-align

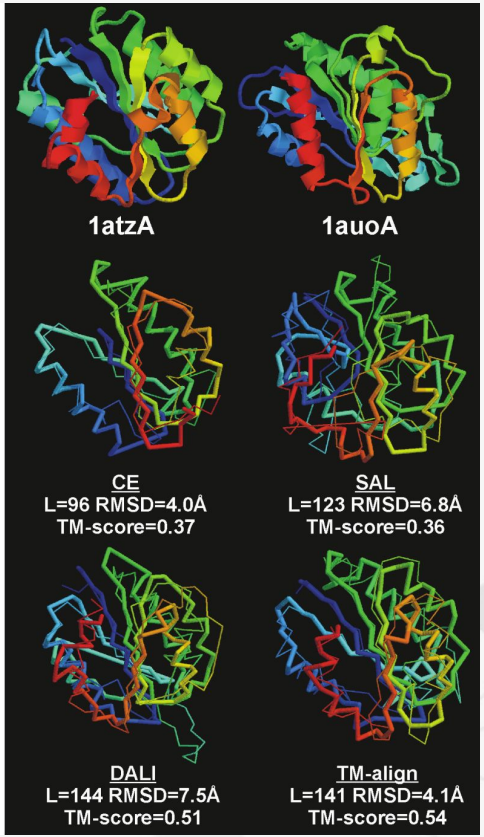
Heuristic iteration

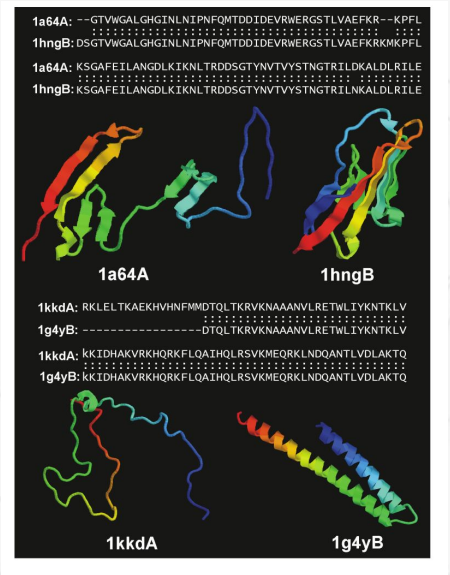
- Rotate the structure by the TM-score rotation matrix obtained from initial alignments
- Apply dynamic programming with gap opening cost -0.6 and with a score similarity matrix equal to (L_{min} being the length of the smaller protein):

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

- Calculate TM-score rotation according to the new alignment
- Repeat until the alignment becomes stable (2-3 iterations to converge)

TM-align





Mutation
(K44 and M45)

Ca^{2+} and calmodulin removed

Services

TM-align

https://aideepmed.com/TM-align/

RCSB-PDB

https://www.rcsb.org/alignment

PyMOL

Examples

DIPARTIMENTO MATEMATICA

- 1atzA 1auoA (TM-align paper, 25% identity)
- Next lecture
  - 1his 3ins (90% identity)
  - 1vid 1chd (10% identity)
- PyMOL
  - fetch 1vid 1chd
  - align 1chd,1vid # sequence alignment followed by a structural superposition
  - super 1chd,1vid # mobile, target (sequence-independent)
  - cealign 1vid,1chd # target, mobile (very low sequence identity)