Protein Structure Comparison: Generic Framework and Applications

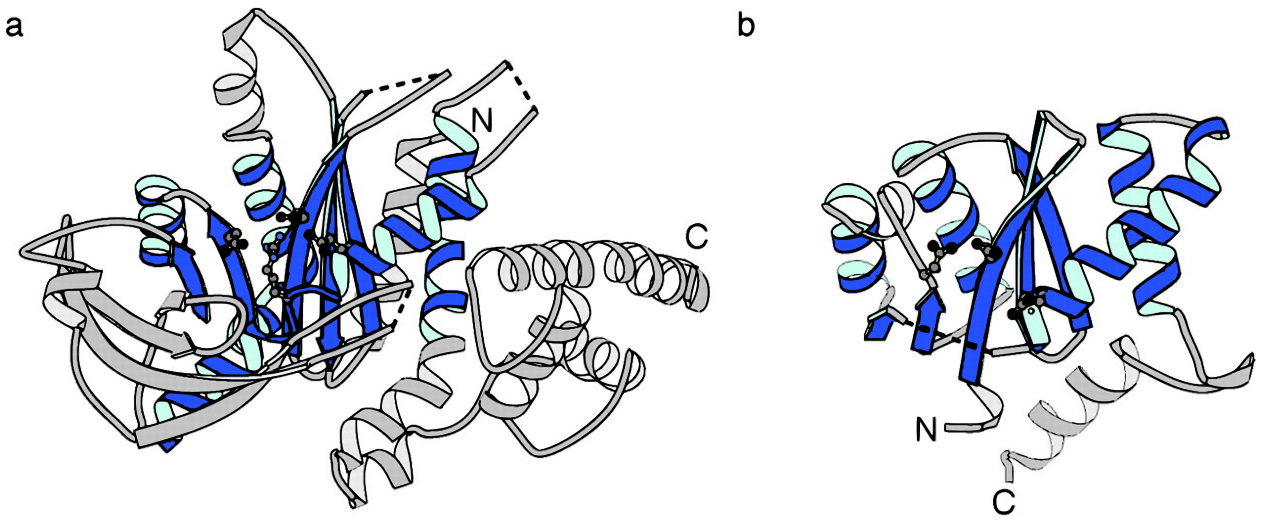
Rumen Andonov

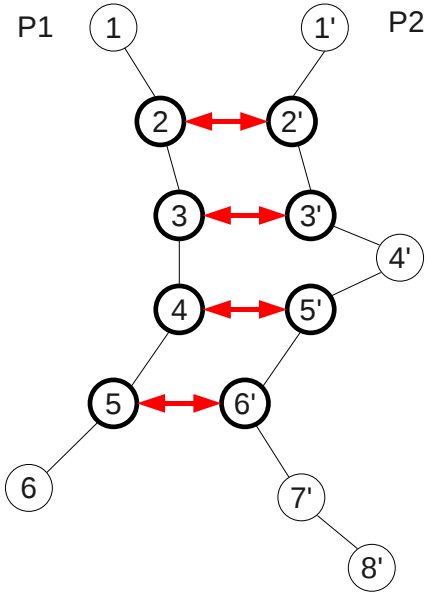
GenScale, IRISA/INRIA and University of Rennes 1

Outline

- Protein structure comparison
- Internal distances similarity measures
- Contact Map Overlap maximisation (CMO)
  - Integer Programming approach for CMO
  - Lagrangian relaxation
  - Computational results
- ACF : local protein structure comparator based on DAST
- Distance matrix ALIgnment (DALI)
  - Integer Programming approach for DALI
  - Lagrangian relaxation
  - Computational results
- Conclusion

Comparing two proteins

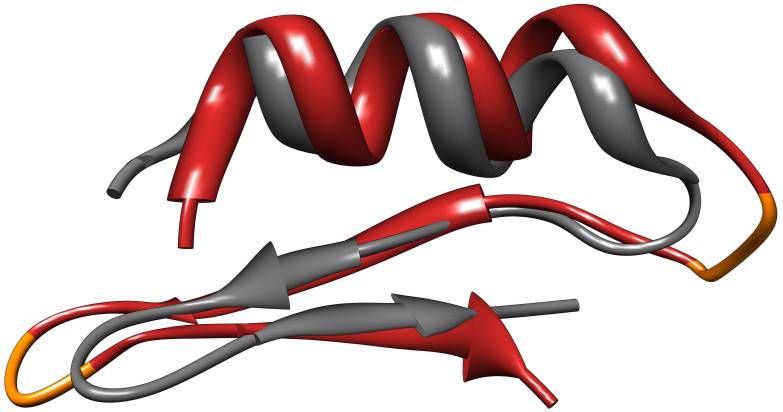




- **An amino-acid alignment** :
  - what is **common** between P_1 and P_2 ?
  - order-preserving one-to-one matching
- **A similarity score** :
  - how similar are P_1 and P_2 ?
  - normalised in $[0, 1]$

$sim(P_1, P_2) \simeq \mathbf{57}\%$

Common sub-structures ?



A part the 1st protein (in red) which is similar (can be well superimposed) to a part from the 2nd protein (in grey).

Root Mean Square Deviation (RMSD)

Given a set of n deviations $S = \{s_1, s_2, \ldots, s_n\}$

$$RMSD(S) = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} s_i^2}$$

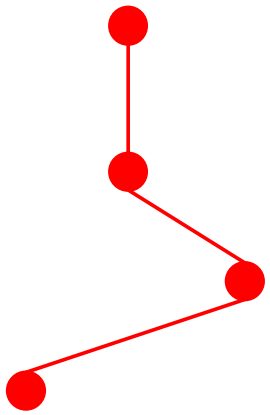
Biologists use two different *RMSD* measures which differ on the measured deviations :

- $RMSD_c$ = deviation between superimposed coordinates
- $RMSD_d$ = deviation between matched internal distances

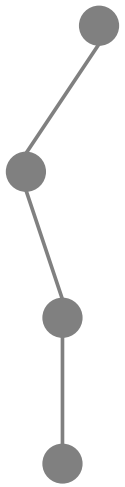
First measure : $RMSD_c$

Root Mean Squared Deviation of superimposed Coordinates

Sub-structure of P1

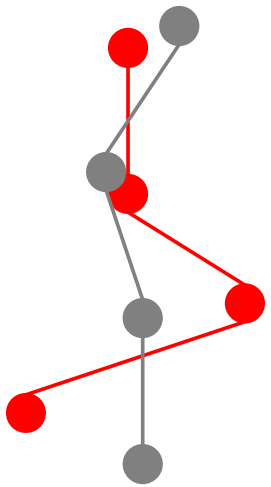


Sub-structure of P2



First measure : $RMSD_c$

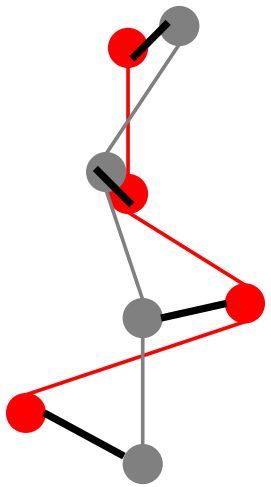
Root Mean Squared Deviation of superimposed Coordinates



- First : superimpose them (3D transformation T)

First measure : $RMSD_c$

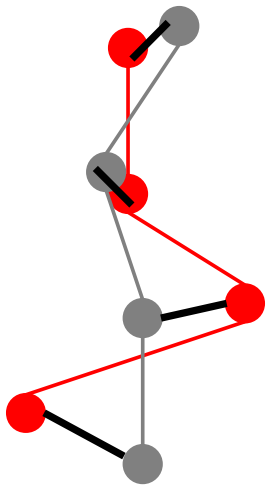
Root Mean Squared Deviation of superimposed Coordinates



- First : superimpose them (3D transformation T)
- Deviations : distances between each superimposed amino-acids

First measure : $RMSD_c$

Root Mean Squared Deviation of superimposed Coordinates

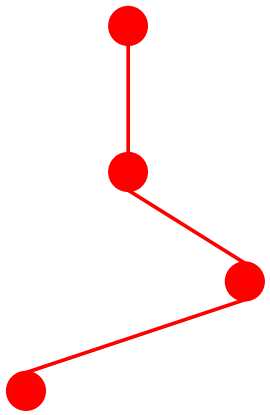


- First : superimpose them (3D transformation T)
- Deviations : distances between each superimposed amino-acids
- Problem : finding transformation T

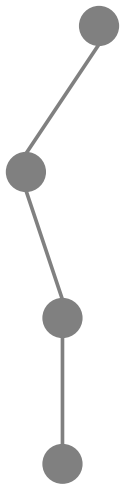
Second measure : $RMSD_d$

Root Mean Squared Deviation of internal distances

Sub-structure of P1

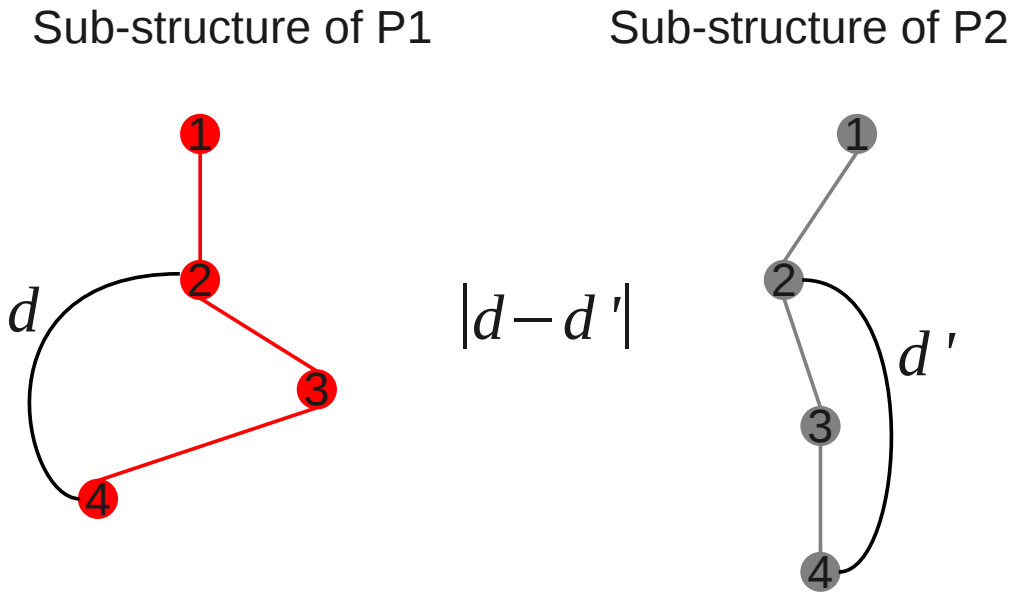


Sub-structure of P2



Second measure : $RMSD_d$

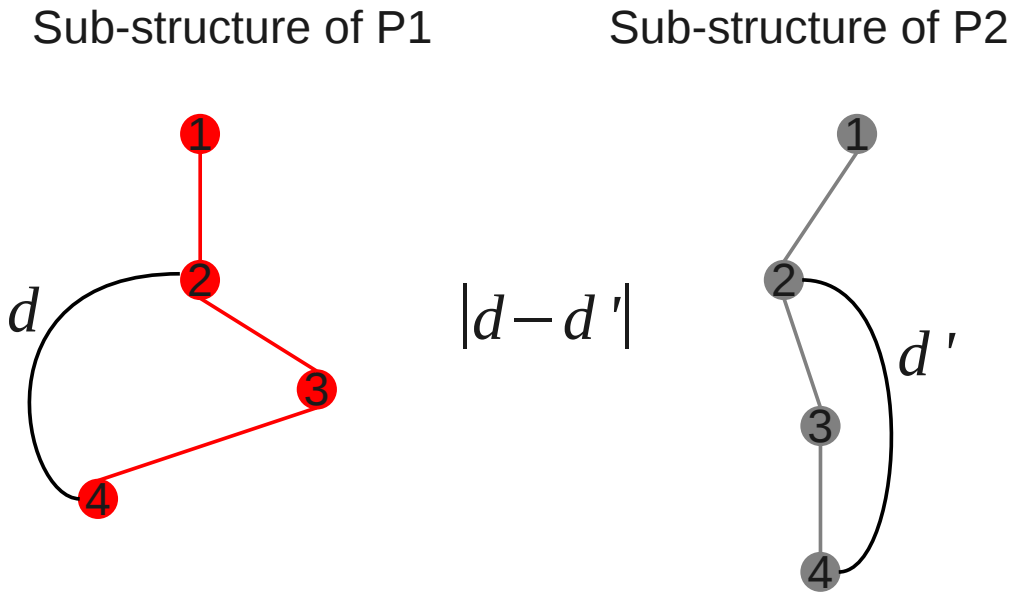
Root Mean Squared Deviation of internal distances



- For all matched internal distances $d \leftrightarrow d'$, the deviation is $|d - d'|$

Second measure : $RMSD_d$

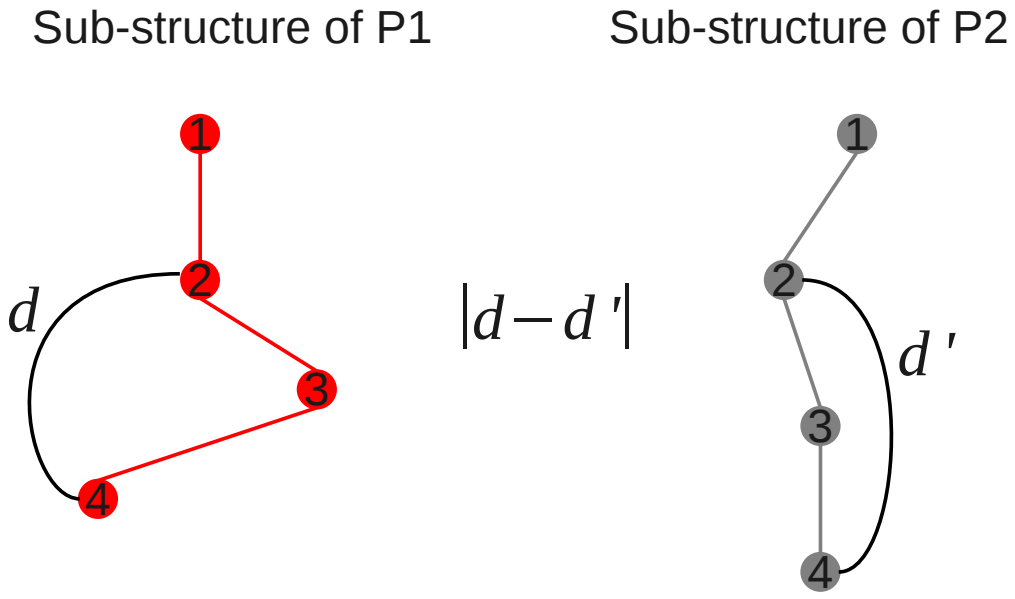
Root Mean Squared Deviation of internal distances



- For all matched internal distances $d \leftrightarrow d'$, the deviation is $|d - d'|$
- No transformation T to compute

Second measure : $RMSD_d$

Root Mean Squared Deviation of internal distances



- For all matched internal distances $d \leftrightarrow d'$, the deviation is $|d - d'|$
- No transformation T to compute
- Problem : not sensible to chirality

Optimization standpont

Given a set of n deviations $S = \{s_1, s_2, \ldots, s_n\}$

$$RMSD(S) = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} s_i^2}$$

Goals :

- minimize *RMSD*
- but maximize the length of the alignment

This is multiobjective optimization.

Many approaches have been proposed...

Based on internal distances :

- Dali (Sander & Holms, 93)
- CMO (Godzik & Skolnick, 94)
- Paul (Wohlers, Petzold, Domingues & Klau, 09)
- DAST (Malod-Dognin, Andonov and Yanev, 10)
- ...

Based on coordinate superimpositions :

- MyFit/GaFit (May & Johnson, 94)
- VAST (Gibrat, Madej & Bryant, 96) Monte Carlo opt.
- CE (Shindyalov & Bourne, 98), approximation of Markov chains
- TM-Align (Zhang & Skolnick 2005)
- SAMO (Chen et al., 06), multi-objective optimization
- ...

Pitfalls

- No consensus which scoring is the best (Godzik, 96 ; Hasegawa and Holm, 09)

  $\Rightarrow$ No easy tool is availlable for comparing different scoring schemes

- Computing optimal alignments is often NP-Hard

  $\Rightarrow$ Heuristics are widely used, without score guaranties

How to get a consensus ?

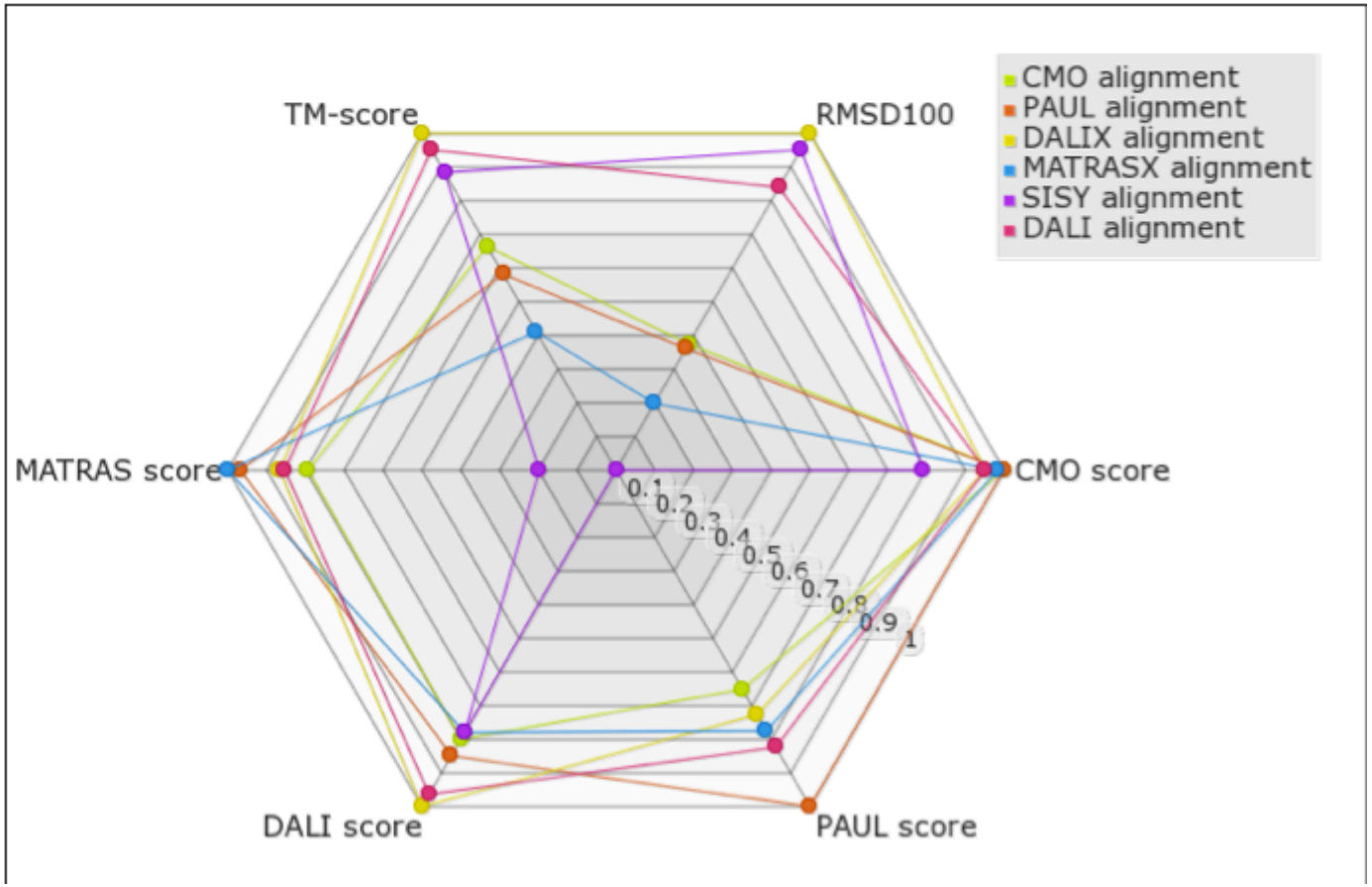


FIGURE: Comparing 1otrA versus 2di0A using various similarity measures

Web server CSA (Comparative Structural Alignment)

OXFORD JOURNALS CONTACT US MY BASKET MY ACCOUNT

Nucleic Acids Research

ABOUT THIS JOURNAL CONTACT THIS JOURNAL SUBSCRIPTIONS CURRENT ISSUE ARCHIVE SEARCH

Oxford Journals › Life Sciences › Nucleic Acids Research › Featured Articles - July 2012

FEATURED ARTICLES - JULY 2012

Featured Articles highlight the best papers published in NAR. These articles are chosen by the Executive Editors on the recommendation of Editorial Board Members and Referees. They represent the top 5% of papers in terms of originality, significance and scientific excellence. The articles are accompanied by a brief synopsis explaining the findings of the paper and where they fit in the broader context of nucleic acids research.

Previous Featured Articles

Click for a list of previous Featured Articles.

Recently Added Featured Articles

CSA: comprehensive comparison of pairwise protein structure alignments
Wohlers I, Malod-Dognin N, Andonov R, Klau GW.

CSA is a web server for the computation, evaluation and comprehensive comparison of pairwise protein structure alignments. CSA's alignment engine computes alignments with a certified quality guarantee that often proves optimality of the returned solutions. It currently supports four inter-residue distance-based scoring schemes: contact map overlap, PAUL, DALI and MATRAS. Computed and further, uploaded alignments are compared using a number of quality measures and intuitive visualizations. CSA is available at http://csa.project.cwi.nl.
Read on

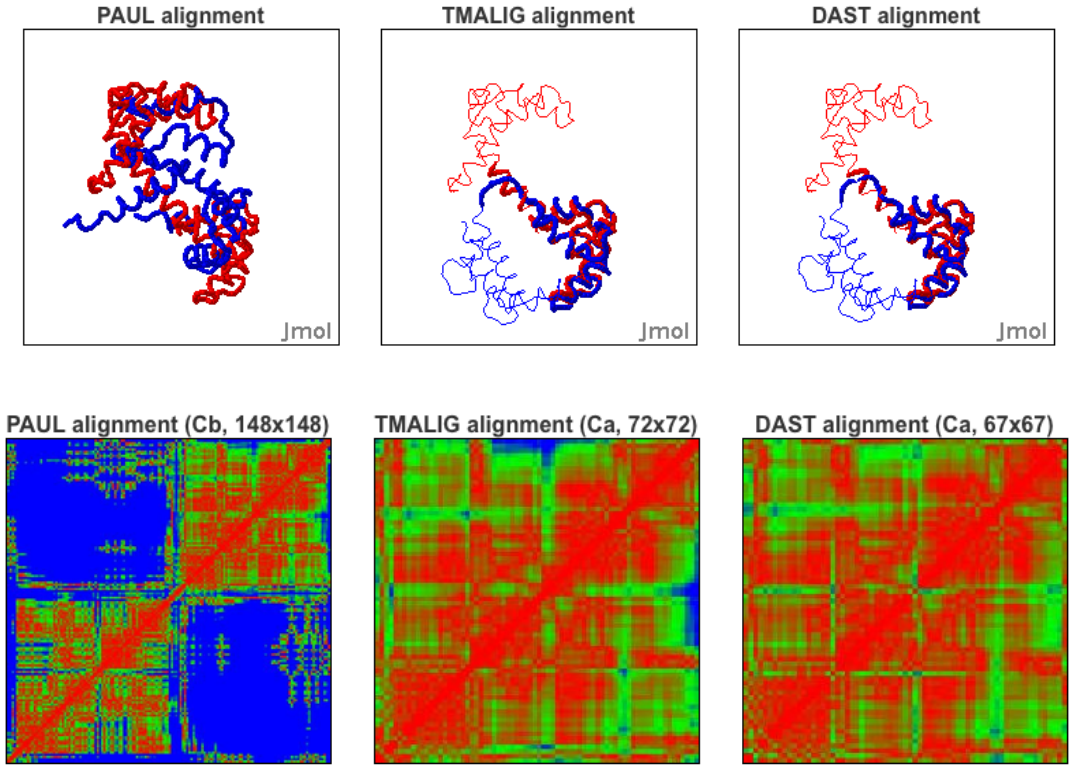
THE JOURNAL

- About this journal
- NAR Methods online
- 2012 Database Issue
- 2012 Web Server Issue
- NAR Special Collections
- Referee Information
- Rights & Permissions
- Dispatch date of the next issue
- This journal is a member of the Committee on Publication Ethics (COPE)
- view Recent Comments on articles
- We are mobile – find out more

Impact factor: 8.026

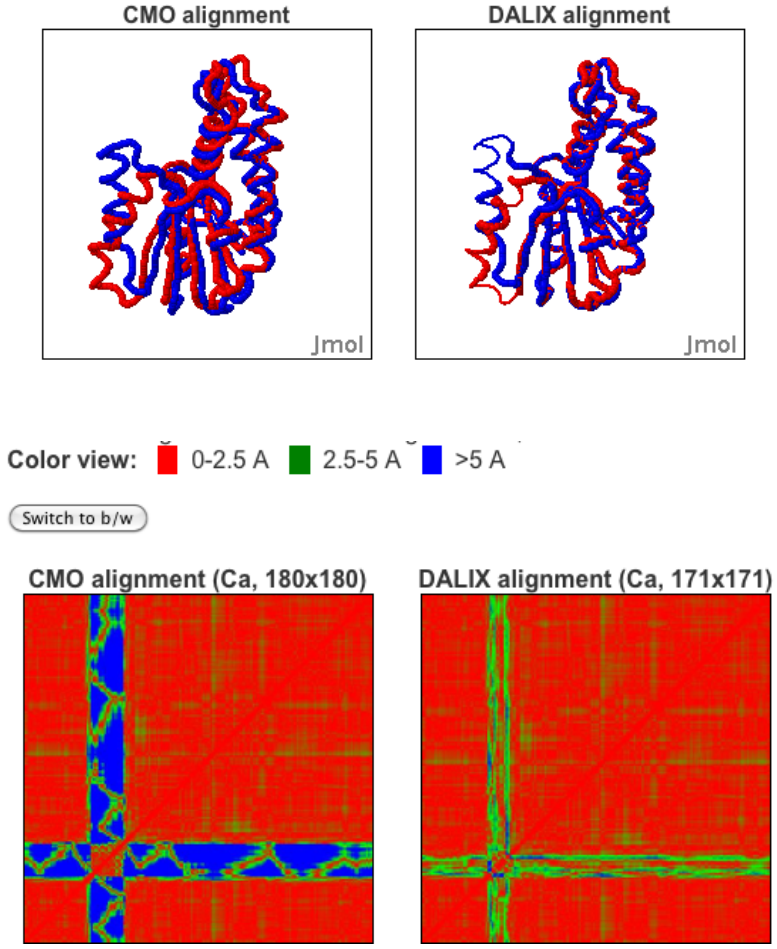
CSA : Case study 3 : Detecting hinges–example with 1 hinge

4clnA versus 2bbmA



CSA : Case study 3 : Detecting hinges–example with 2 hinges

1cbuB versus 1c9kB



Focus and goal of this talk

Fundamental internal distances similarity measures

- **DALI** (Sander & Holms, 93) : one of the first score and heuristic.
- **CMO** (Godzik & Skolnick, 94) : the simplest internal distances similarity measure.
- **Paul** (Wohlers, Petzold, Domingues & Klau, 09) : intermediate score.

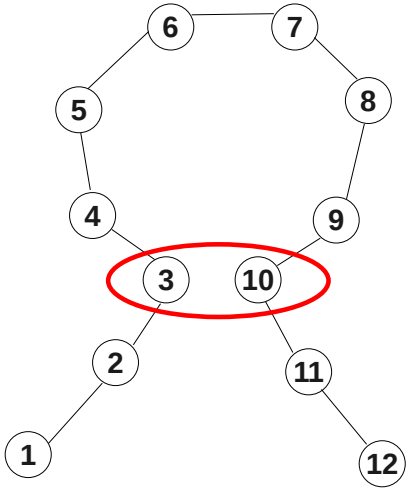
Existing exact solvers

- LAGR (Caprara & Lancia, 2004), CMOS (Xie & Sahinidis, 2007), Paul (Wohlers, Petzold, Domingues & Klau, 09), A_purva (Andonov & al. 2011)
- Based on IP approaches coupled with branch and bound.
- Upper-bounds = upper-estimations based on Lagrangian relaxations.
- Lower-bounds = feasible solution (sub-optimal)
- A_purva shown to be the fastest and providing tight upper and lower bounds.

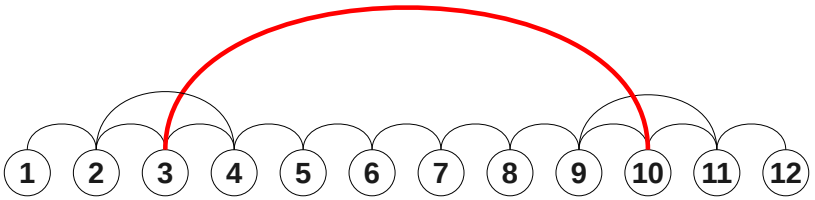
The Contact Map Overlap maximization

CMO : based on small internal distances

A contact = an internal distance smaller than 7.5Å



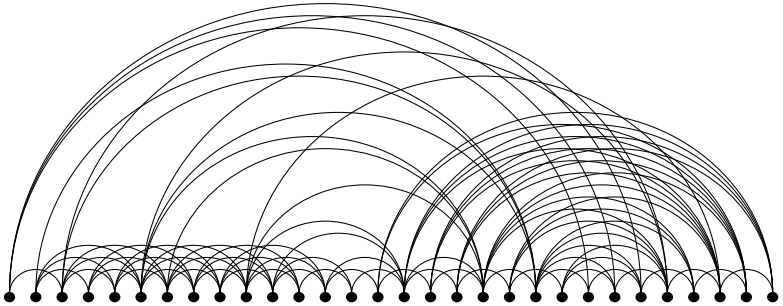
Protein 3D Structure

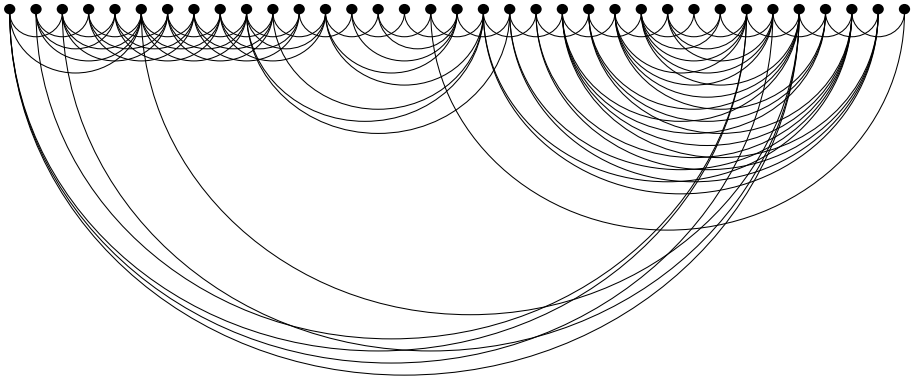


Contact map graph

2 amino-acids in contacts ($C_\alpha - C_\alpha$ distance $\leq$ 7.5Å) $\Leftrightarrow$ an edge in the contact map. a **contacts** in the structure $\Leftrightarrow$ an **edge** in the contact map

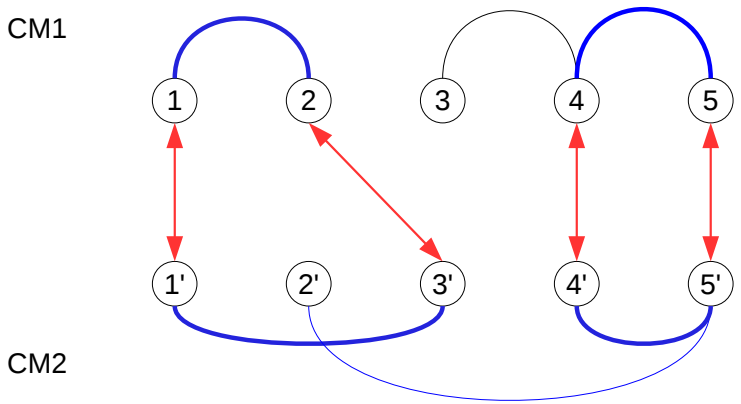
CMO : the approach





Aligning two proteins $\Rightarrow$ aligning two contact map graphs

CMO : Maximizing the number of common contacts

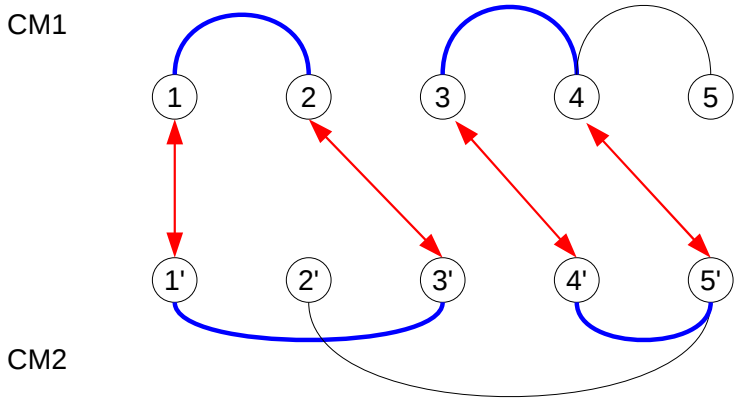


A **common contact** (or contact overlap) occurs when two matchings pairs **i** $\leftrightarrow$ **k** and **j** $\leftrightarrow$ **l** match one contact in $P1$ with one contact in $P2$

- Under this matching, the two proteins share a common small internal distance
- The above alignment has two common contacts
- Optimal alignment $\rightarrow$ the one that maximizes the number of common contacts

CMO : Maximising the number of common contacts

Given two contact maps $CM1, CM2$, an optimal CMO alignment **maximizes** the number of common contact edges **NCC**



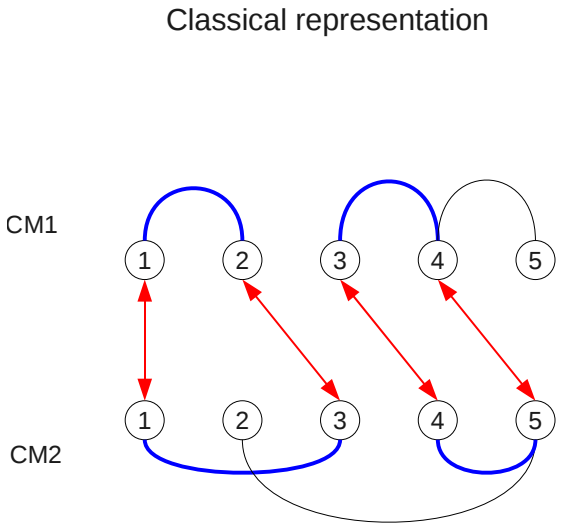
- Similarity score :

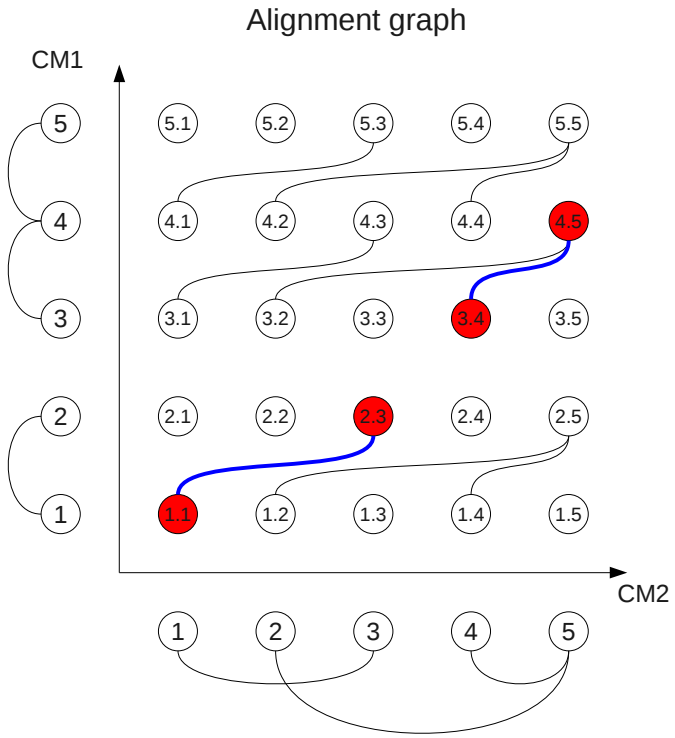
$$\mathrm{SIM}(CM1, CM2) = \frac{2 \times NCC}{|E_1| + |E_2|} \tag{1}$$

  where $|E_1|, |E_2|$ are the edges of $CM1, CM2$.

- A challenging NP-Complete problem (Goldman, Istrail & Papadimitriou in 99)

CMO, the alignment graph approach





- An optimal alignment $\Leftrightarrow$ an **Increasing Subset of Vertices** having a maximum number of edges

Approach : branch and bound using new Integer Programming formulation

- LAGR (Caprara & Lancia, 2002)
- CMOS (Xie & Sahinidis, 2007)
- A_purva (Andonov, Malod-Dognin, Yanev, 2008)

All exact branch and bound approaches providing :

- Upper-bounds = upper-estimations of the number of common contacts
- Lower-bounds = feasible solution (sub-optimal)

Efficiency depends on the quality (tightness & time) of the bounds

A_purva uses a two steps method

1. Reformulate CMO as an Integer Programming problem P
2. Bounds from Lagrangian relaxation of P

Achievements I : Automatic classification

Protocol :

Alignments returned by short runs of A_purva

- No branch and bound
- Only 500 iterations of the minimiser over λ

Score given to CHAVL (Lerman, 93)

- Unsupervised ascendant classification tool

Achievements I : Automatic classification

Protocol :

Alignments returned by short runs of A_purva

- No branch and bound
- Only 500 iterations of the minimiser over λ

Score given to CHAVL (Lerman, 93)

- Unsupervised ascendant classification tool

Results :

- **Exactly the same** classification as SCOP for the Skolnick set
  - Total running times 297 sec.

Achievements II : Automatic classification

Proteus_300 :

- 300 proteins, 10 families, test set based on ASTRAL compendium
- Same protocol as before (only 500 iterations, computed scores given to CHAVL)
- Score function :

$$\mathrm{SIM}(P_1, P_2) = \frac{2 \times LB}{|E_1| + |E_2|} \tag{2}$$

- A_purva needed 13 hours and 38 minutes to complete all 44,850 pairwise comparisons
- have been obtained only minor disagreements with the SCOP classification
- Proteus_300 start to be used by the community as a benchmark
- Results appeared in J. of Computational Biology (2011) and WABI'08

Achievements III : Familly identification

SHREC'10 contest :

- Given 1000 known protein structures classified into 100 CATH superfamilies (10 protein structures per super-families)
- 50 "unknown" protein structures are given later to the participants
- Obj : Participants had three days to classify the 50 unknown proteins into the 100 CATH superfamilies
- A_purva achieved the highest success rate (80% of correctly classified proteins during the competition by using a similarity function different from (2)), as well as the highest sensitivity and specificity. We observed afterwards that (2) gives 92% success rate.
- Result appeared in Eurographics Workshop on 3D Object Retrieval (2010)

Recapitulation

A_purva :

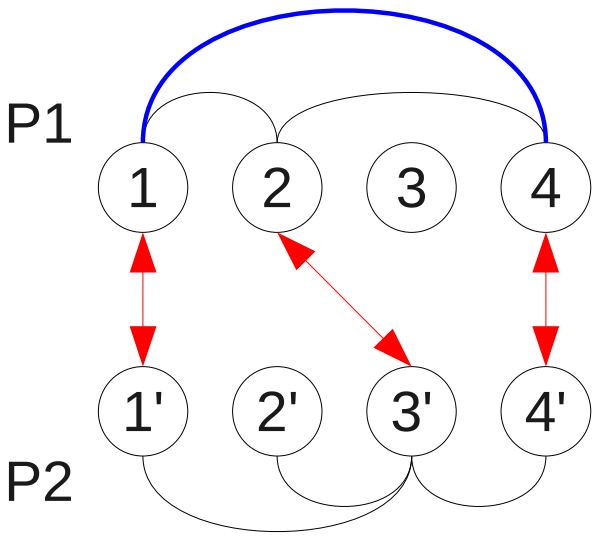
- Is an exact solver for protein structure comparison.
- Its characteristics to provide tight upper and lower bounds of the solution makes it very suitable for large-scale data set processing and classification.
- Availability on our plateforme : http ://apurva.genouest.org
- webserver CSA (Comparative Structural Alignment) : http ://csa.project.cwi.nl

Related publications :

- Maximum contact map overlap revisited. *J. Comput. Biol.*, 18(1) :1–15, 2011.
- An efficient lagrangian relaxation for the contact map overlap problem. In *WABI '08*, pp. 162–173. Springer-Verlag, 2008.
- Shrec-10 track : Protein models. *3DOR : Eurographics Workshop on 3D Object Retrieval*, pp. 117–124 2010

The Distance-Based Alignment Search Tool (DAST)

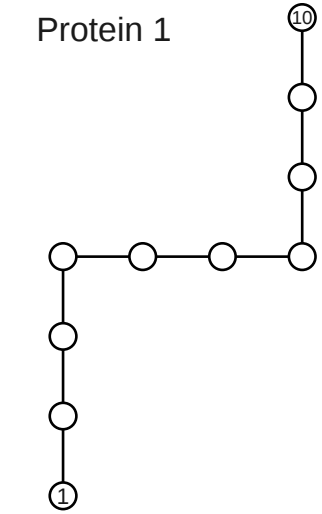
CMO introduces some “errors” :

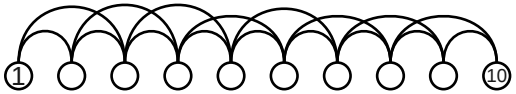


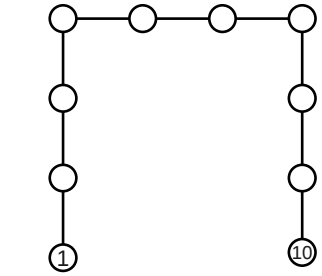
Matching “$1 \leftrightarrow 1', 2 \leftrightarrow 3', 4 \leftrightarrow 4'$”

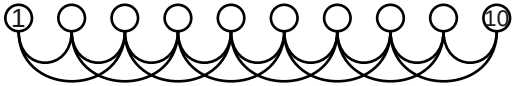
- maximum number of common contacts (2)
- 1 and 4 from P_1 are in contact, while $1'$ and $4'$ in P_2 are remote

CMO : Problem of forgetting long internal distances

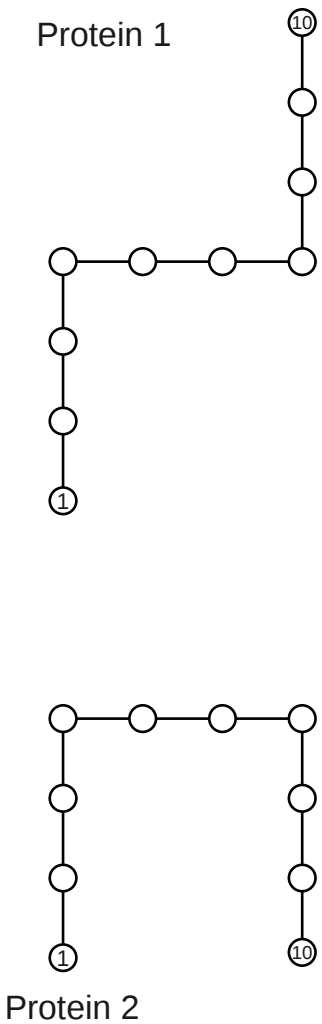


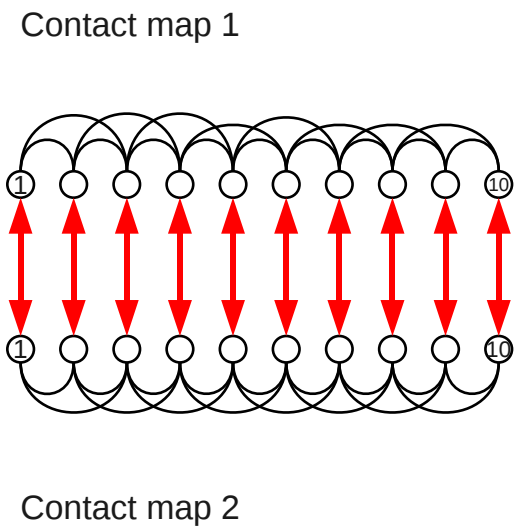




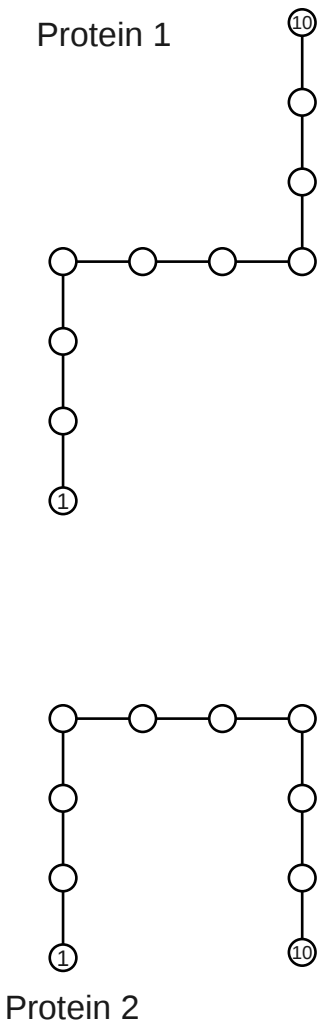


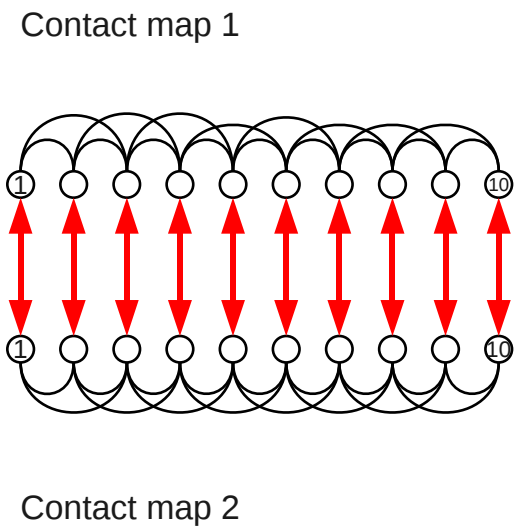
CMO : Problem of forgetting long internal distances





CMO : Problem of forgetting long internal distances





- **CMO gives maximum score $\rightarrow$ perfectly identical proteins ? !**

DAST : Principle

Replace the notion of common contact

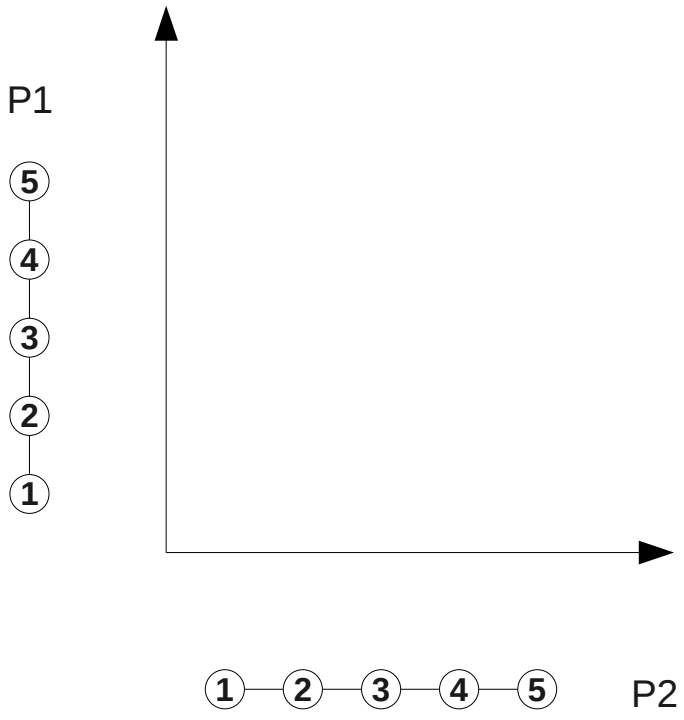
- With the more general notion of similar internal distance

Align only similar internal distances

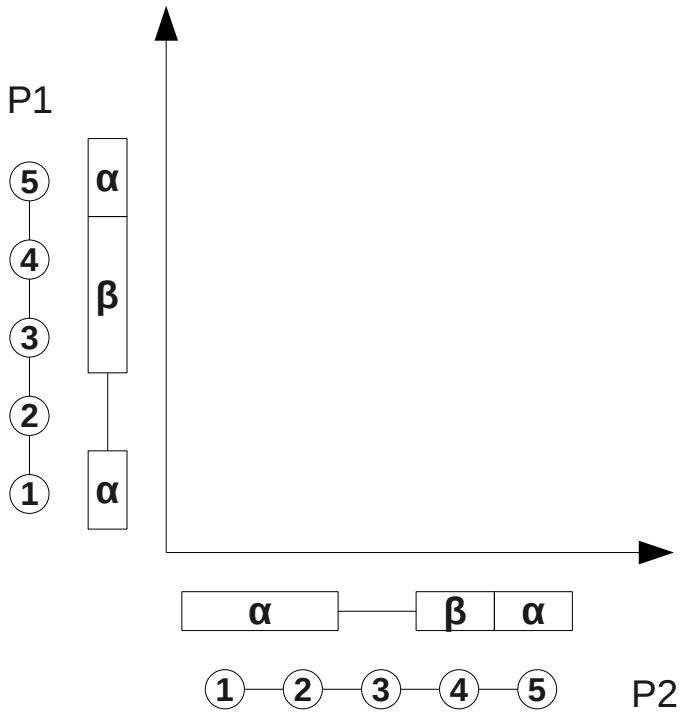
- If all aligned internal distances are equal ($\approx \theta$),

  RMSD of internal distance is $\leq \theta$

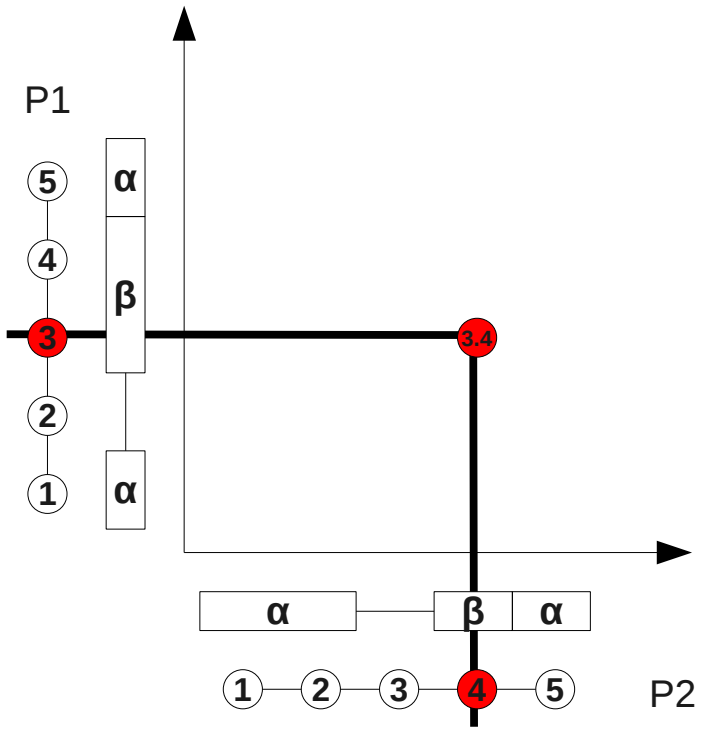
DAST : Matching two amino-acids



DAST : Matching two amino-acids

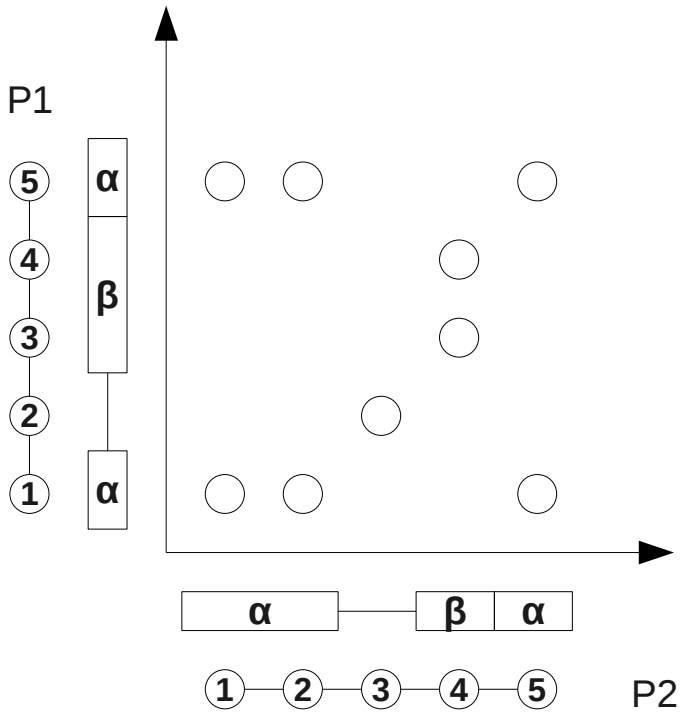


DAST : Matching two amino-acids



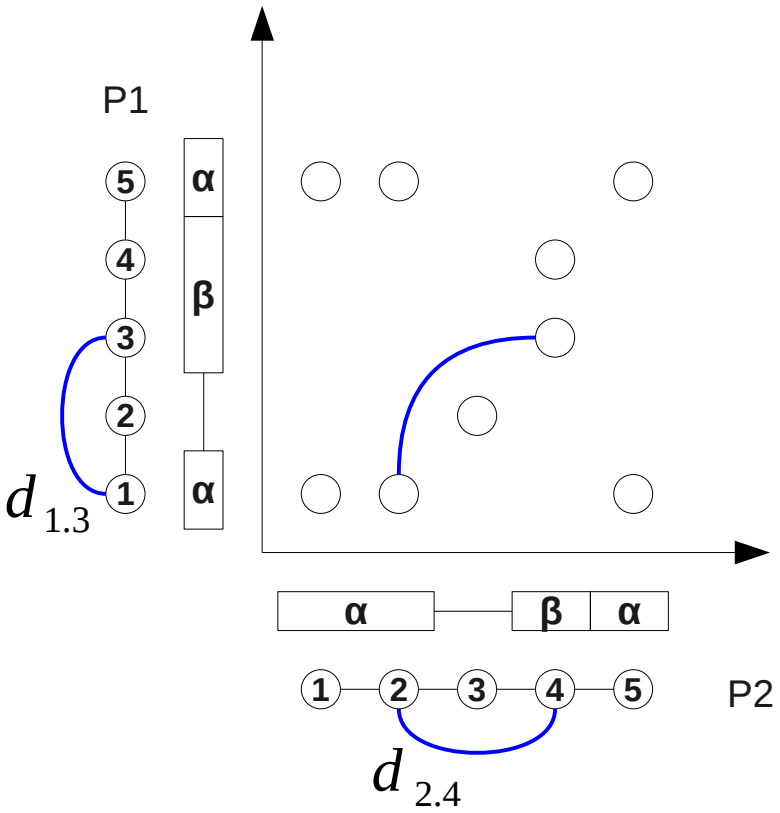
- If $i \in P1$ and $k \in P_2$ come from same kind of SSEs :
  - $\rightarrow$ Vertex $i.k$ is in the alignment graph

DAST : Matching two amino-acids



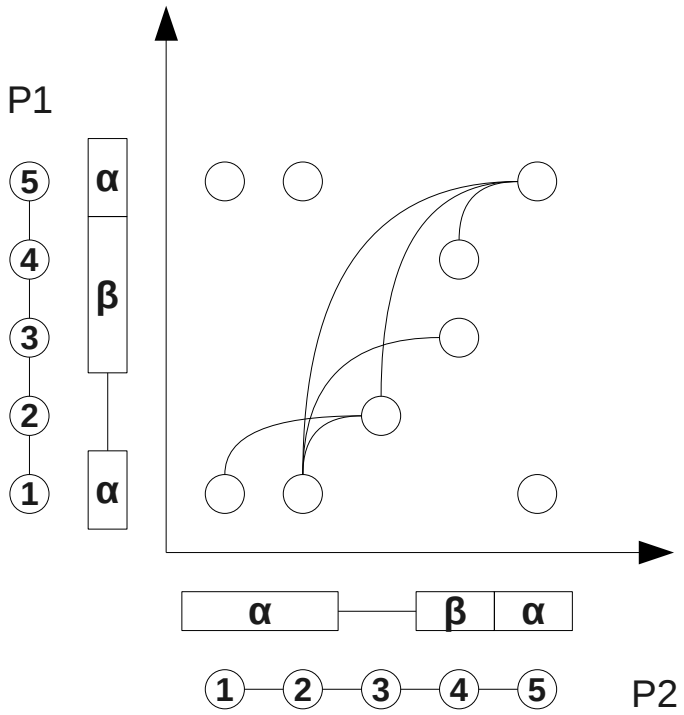
- If $i \in P1$ and $k \in P_2$ come from same kind of SSEs :
  - $\rightarrow$ Vertex $i.k$ is in the alignment graph

DAST : Matching two internal distances



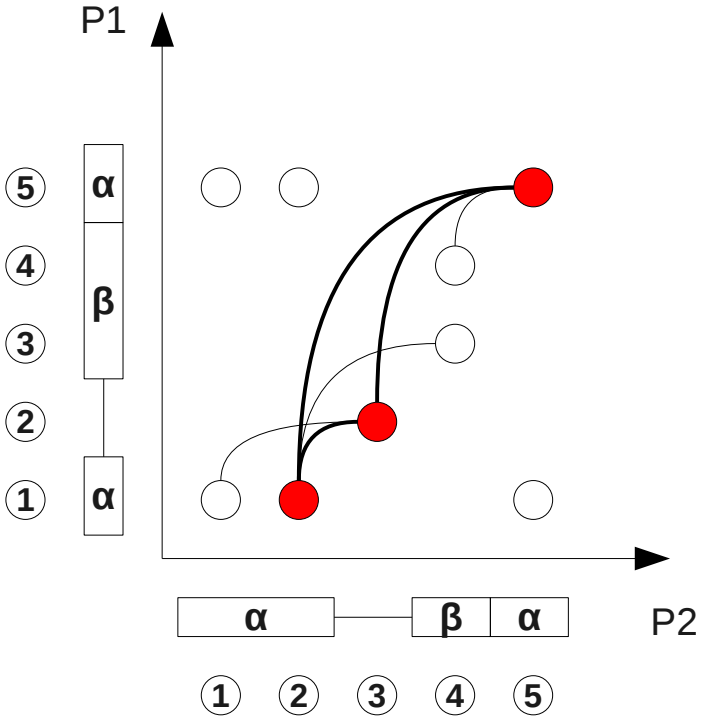
- If $|d_{ij} - d_{kl}| \leq \theta$ then
  - $\rightarrow$ Edge $(i.k, j.l)$ is in the alignment graph

DAST : Matching two internal distances



- If $|d_{ij} - d_{kl}| \leq \theta$ then
  - $\rightarrow$ Edge $(i.k, j.l)$ is in the alignment graph

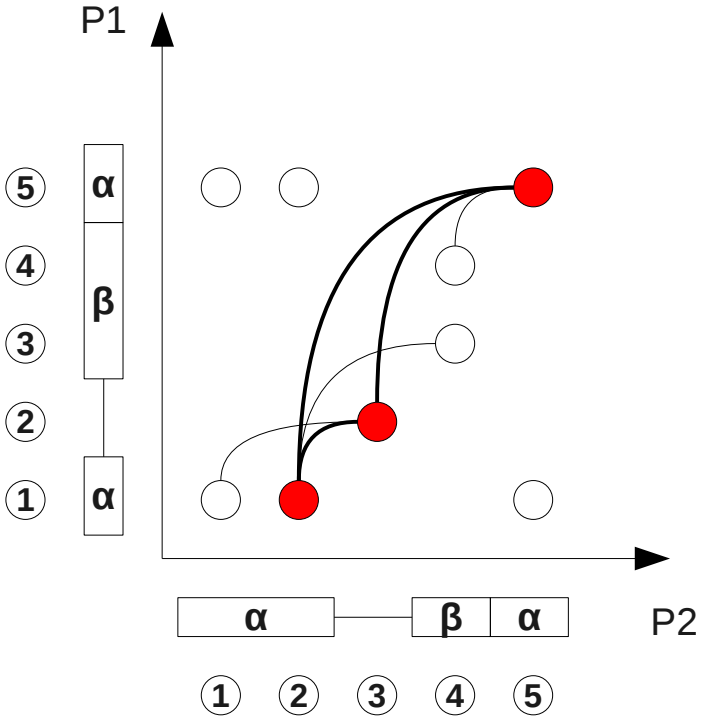
DAST : Feasible and optimal matching



Feasible matching :

- A clique
  - → All matched internal distances are similar ($\approx \theta$)

DAST : Feasible and optimal matching



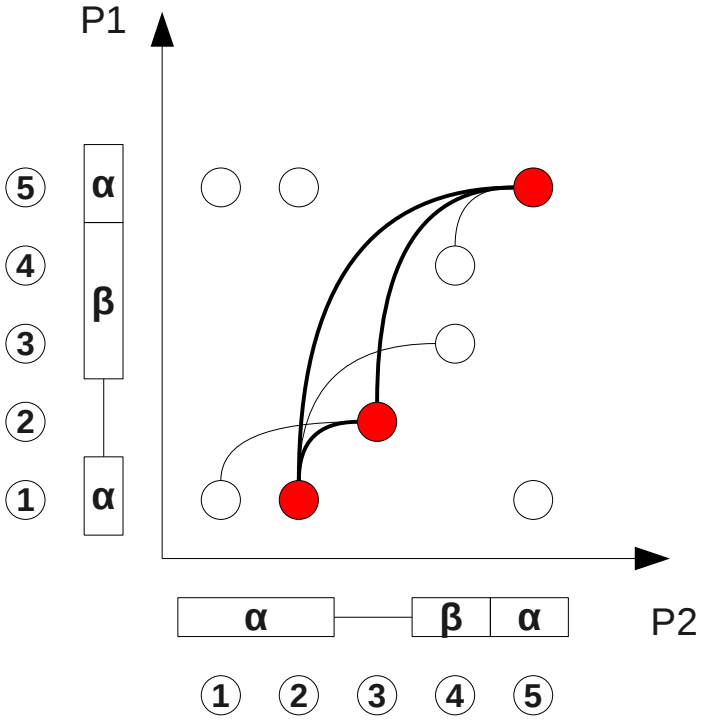
Feasible matching :

- A clique
  - → All matched internal distances are similar ($\approx \theta$)

Optimal matching :

- Maximum clique
  - → Longest (in terms of amino-acids) of such matching

DAST : Feasible and optimal matching



Feasible matching :

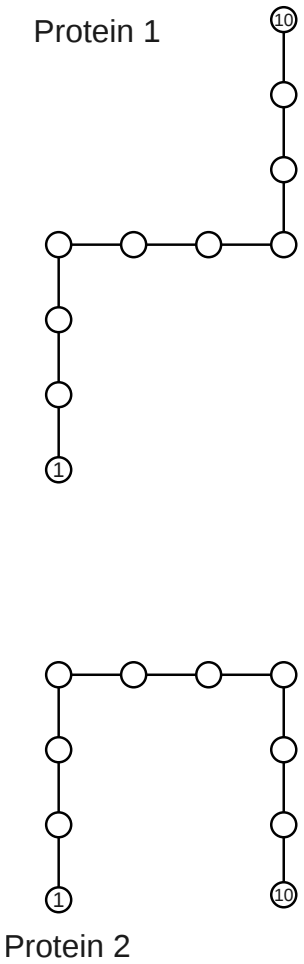
- A clique
  - → All matched internal distances are similar ($\approx \theta$)

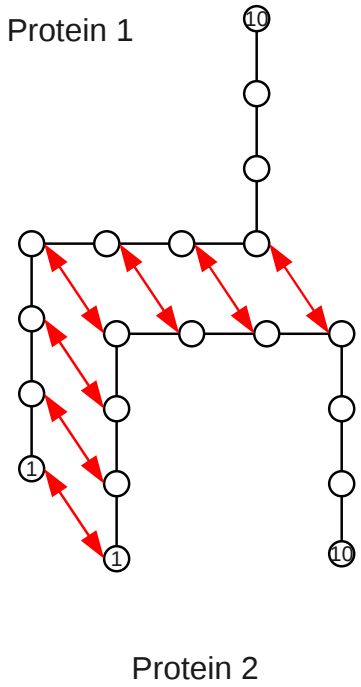
Optimal matching :

- Maximum clique
  - → Longest (in terms of amino-acids) of such matching

RMSD of internal distance is $\leq \theta$

Back to our strange CMO alignment





CMO vs DAST alignments

DAST : $RMSD_d \leq 2$Å, but shorter alignments than CMO

	Instance	Length (AA)		$RMSD_d$ (Å)	
		CMO	DAST	CMO	DAST
similar instances	1amkA–1aw2A	**247**	200	1.39	**0.68**
	1amkA–1htiA	**247**	204	1.24	**0.74**
	1qmpA–1qmpB	**129**	118	**0.22**	**0.22**
	1ninA–1plaA	**97**	58	1.42	**0.96**
	1tmhA–1treA	**254**	233	0.90	**0.44**
dissimilar instances	1amkA–1b00A	**120**	41	5.62	**1.23**
	1amkA–1dpsA	**163**	32	13.01	**1.06**
	1b9bA–1dbwA	**123**	44	6.02	**1.11**
	1qmpA–2pltA	**95**	17	7.36	**1.18**
	1rn1A–1b71A	**104**	26	11.22	**0.82**

CMO : 7.5 Å contact maps, DAST : 3 Å distance threshold

Remark : DAST approach is significantly slower than CMO (A_purva solver).

Optimal DALI protein structure alignment

What is CMO weakness ?

FIGURE: Alignment of 1aawA (gray) and 1gxiE (pink), an instance of the Sisy set. Optimal superposition according to the respective alignment. Residues colored in dark tone are aligned, residues colored in light tone are unaligned. **Left :** The Sisy reference alignment **(29 aligned residues, RMSD of 1.14)**. **Middle :** The optimal CMO alignment ; it correctly aligns 96.55% of the aligned residues of the reference alignment. Alignment length is **56, RMSD 4.25**. Additional gaps are inserted. Overaligning and insertion of additional gaps leads to a low RMSD value. **Right :** The heuristic DALI alignment correctly aligns all residues of the reference alignment, but extends the alignment length to **50 (RMSD of 2.55)**.

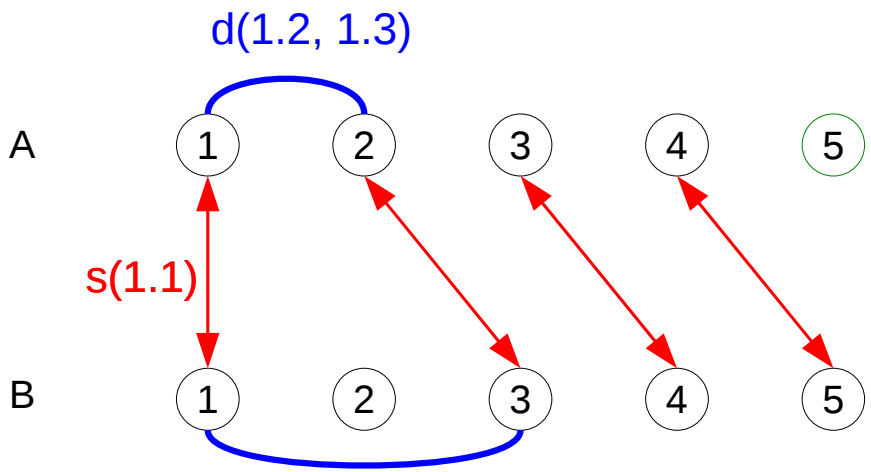
Distance matrix ALIgnment (DALI)

DALI approach

- DALI (distance matrix alignment) (Holm and Sander,1993) – one of the most widely used structural alignment heuristics.
- Available via the European Bioinformatics Institute (EBI) structural analysis tool box, it processes about 1500 pairwise alignment user requests a month
- DALI paper has been cited almost 3000 times (more than 5000 times including closely related and follow-up papers), more often than any other structural alignment program.

DALI generic scoring scheme

Various scores when comparing protein A with B



- Distance score (d) = compatibility between internal distances ((1.2) $\leftrightarrow$ (1.3)) ;
- Sequence score (s) = compatibility between matched residues (1 $\leftrightarrow$ 1) ;

Optimal alignment = residues matching maximizing the sum $S(A, B)$=d+s

DALI approach : more details

Function $d(\cdot,\cdot)$, which is used in the objective function, is the DALI elastic similarity function that scores pairs A_{ij} and B_{kl} of inter-residue distances as

$$d(A_{ij}, B_{kl}) = \left(0.2 - \frac{|A_{ij} - B_{kl}|}{\frac{1}{2}(A_{ij} + B_{kl})}\right) e^{-\left(\frac{\frac{1}{2}(A_{ij}+B_{kl})}{20}\right)^2}$$

Based on the overall DALI score $S(A,B)$, the DALI z-score $Z(A,B)$ is computed as follows :

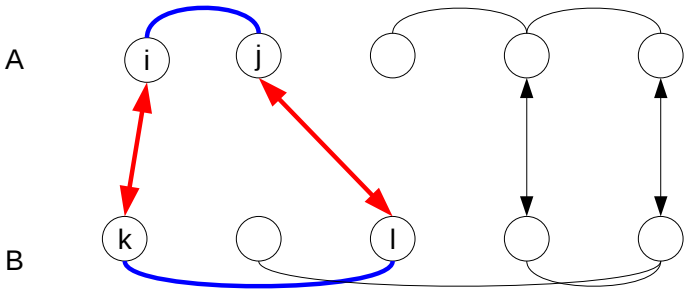
$$Z(A,B) = \frac{S(A,B) - m(L)}{0.5 \cdot m(L)}.$$

The term $m(L)$ for $L = \sqrt{n_A n_B}$ is the approximate mean score and the denominator $0.5 \cdot m(L)$ estimates the average standard deviation. The z-score thus measures the significance of the detected structural similarity based on an experimentally determined background distribution of DALI scores.

Integer Programming approach to DALI

Mathematical model : objective and variables

Protein comparison is order-preserving one-to-one amino-acid matching/alignment



$$\text{objective : max} \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

- $x_{ik} \in \{0, 1\}$: aligning residues i and k
- $y_{ikjl} \in \{0, 1\}$: aligning distance between i and j with distance between k and l
- $\mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}})$: structure score (i.e. DALI elastic similarity function)
- $\mathbf{s}(\mathbf{A_i}, \mathbf{B_k})$: sequence score

Alignment graph formalism

It contains both :

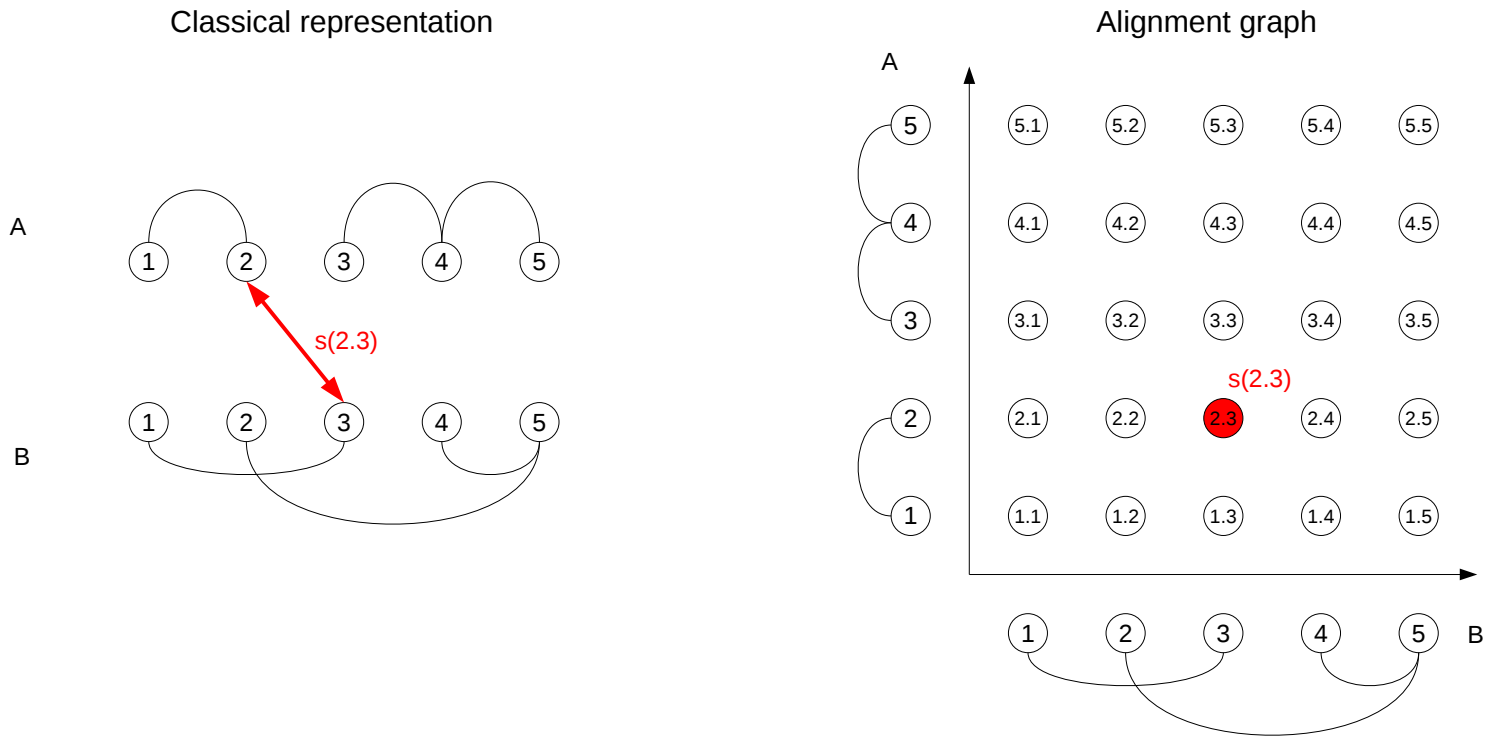
- Rules for creating the alignment graph
- Definition of the sub-graph corresponding to an optimal alignment

Inspired by the Contact Map Overlap for Protein Comparison

- Led to an efficient CMO solver (Andonov et al., 11)

Goal of this study : To adapt it for DALI approach

Alignment graph formalism : Vertices and weights

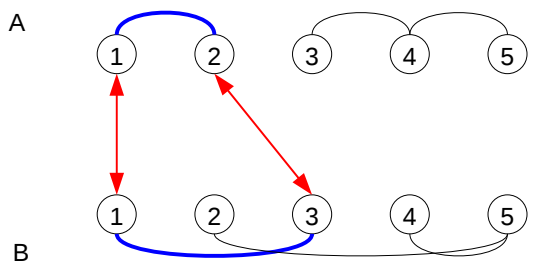


- To each matching pair $\mathbf{i} \leftrightarrow \mathbf{k}$ corresponds a vertex $\mathbf{i.k}$ in the alignment graph
- its weight, $\mathbf{s(A_i, B_k)}$, corresponds to the sequence score when aligning residues *i* and *k*

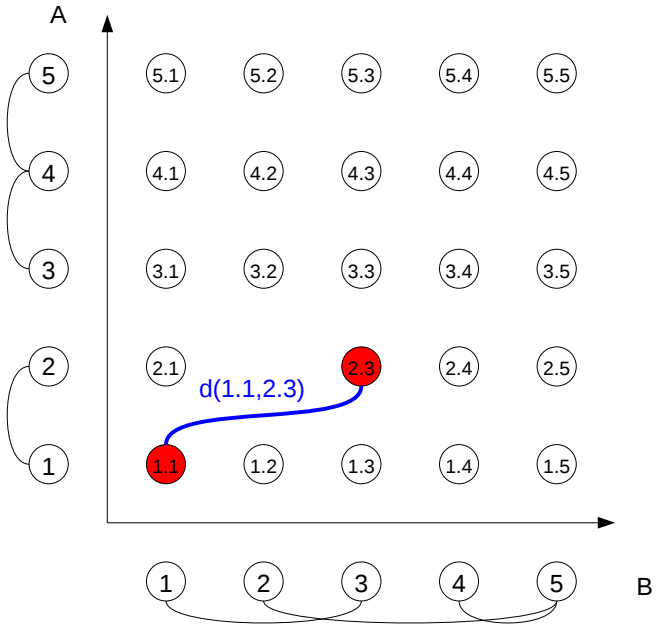
Alignment graph formalism : Edges and weights

Classical representation

Distance A(1,2) is alligned with distance B(1,3)



Alignment graph

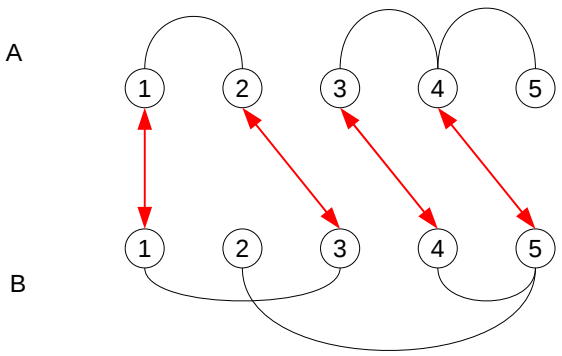


- Aligning distance A_{ij} with distance B_{kl} (i.e. matching pairs $\mathbf{i} \leftrightarrow \mathbf{k}$ and $\mathbf{j} \leftrightarrow \mathbf{l}$) is modeled by the edge $(\mathbf{i.k}, \mathbf{j.l})$
- Its weight, $\mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}})$, is given by the DALI elastic similarity function

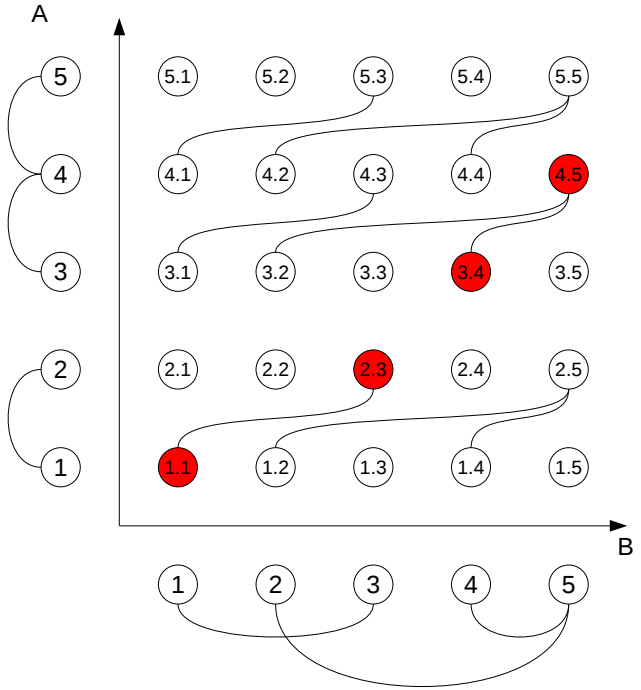
Modeling feasible alignment

A feasible alignment is order-preserving one-to-one amino-acid matching

Classical representation

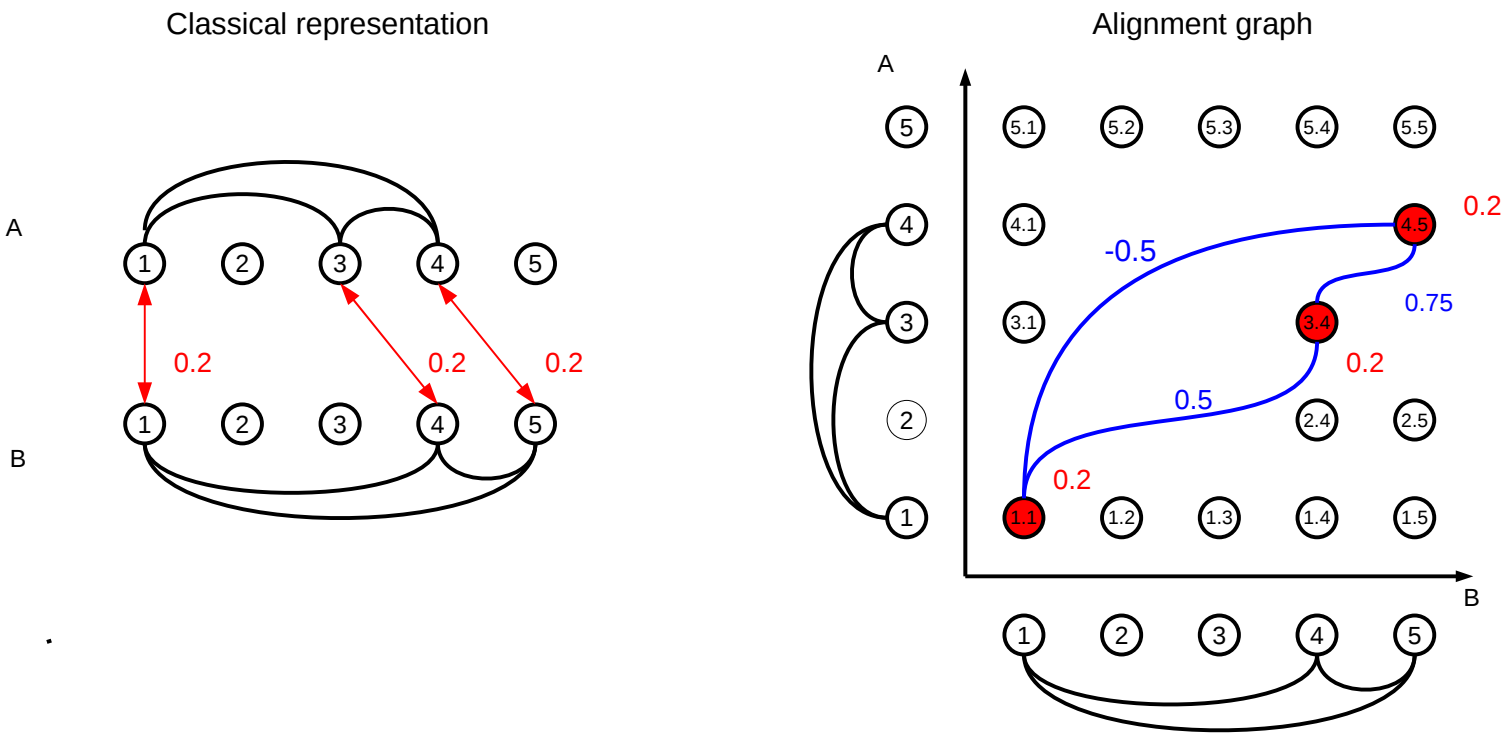


Alignment graph



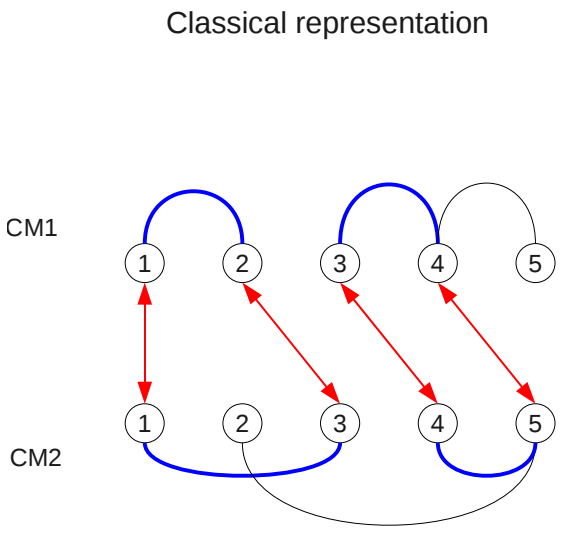
- A feasible matching $\Leftrightarrow$ an **Increasing Subset of Vertices (ISV)** in the alignment graph

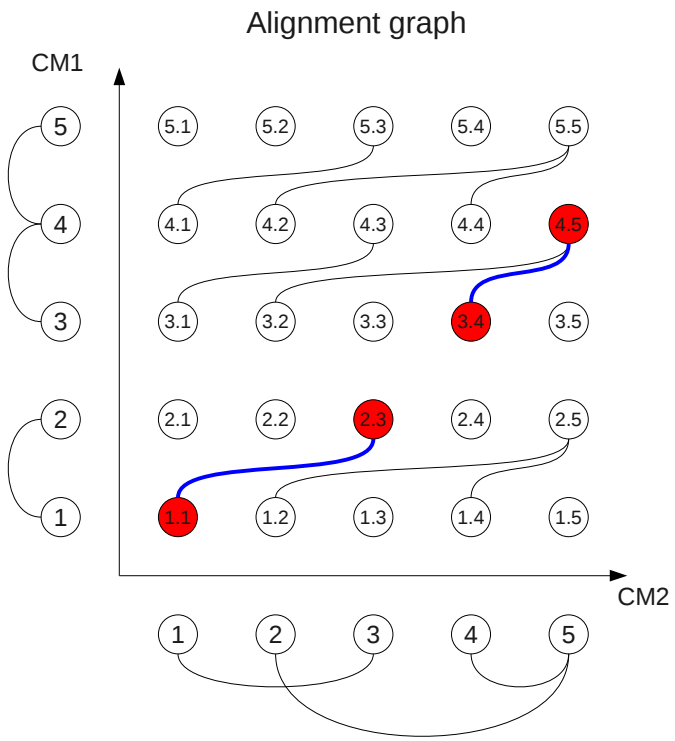
Score of a feasible alignment



- The score of an alignment = the weight of the induced graph (sum of vertices and edges)
  (here : **0.75+0.5-0.5** + **0.2+0.2+0.2**= 1.35)
- The optimal alignment = the one with the **maximum score**

CMO, the alignment graph approach : recall

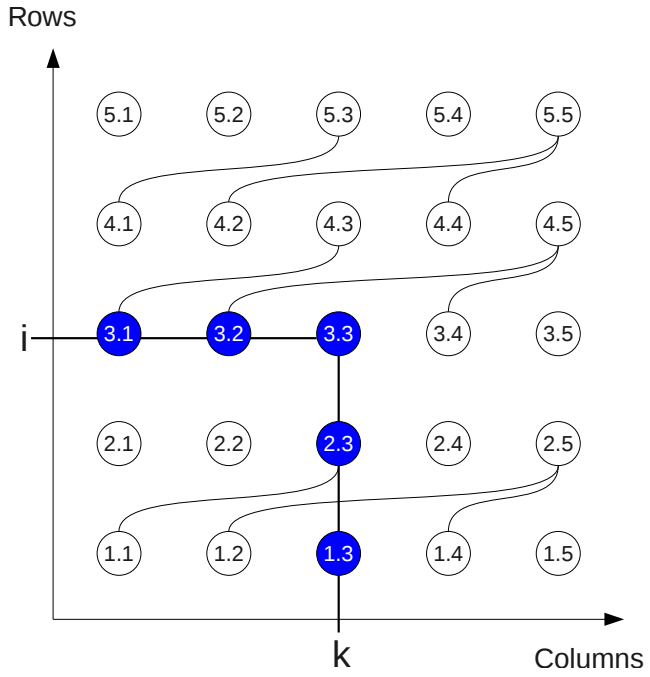




- An optimal alignment $\Leftrightarrow$ an **Increasing Subset of Vertices** having a maximum number of edges
- CMO can be viewed as a subcase of DALI scoring function

Mathematical model : constraints

Let *SOS* (Special Order Set) denote a set of mutually exclusive vertices (x variables). The notion Increasing Subset of Vertices (ISV) allows to define various *SOS* : any row, any column, ... The most important is as follows :

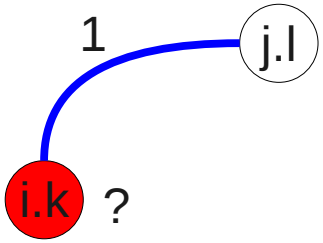


- $$\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \qquad \forall \text{ row } i, \forall \text{ col } k.$$
- n rows and m columns $\rightarrow$ $\mathbf{n} \times \mathbf{m}$ equations
- Determine the vertex feasible solutions set (ISV)

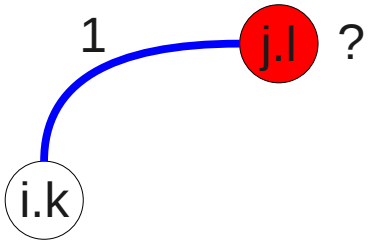
Principle of modeling : binding edges and vertices together

Edge-driven tail vertex activation

$$x_{ik} \geq y_{ikjl}, \forall \text{ edge } (ik.jl).$$



Principle of modeling : binding edges and vertices together

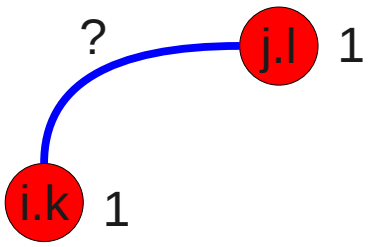


$$x_{ik} \geq y_{ikjl}, \forall \text{ edge } (ik.jl).$$

Edge-driven head vertex activation

$$x_{jl} \geq y_{ikjl}, \forall \text{ edge } (ik.jl).$$

Principle of modeling : binding edges and vertices together



$$x_{ik} \geq y_{ikjl}, \forall \text{ edge } (ik.jl).$$

$$x_{jl} \geq y_{ikjl}, \forall \text{ edge } (ik.jl).$$

Vertices-driven edge activation

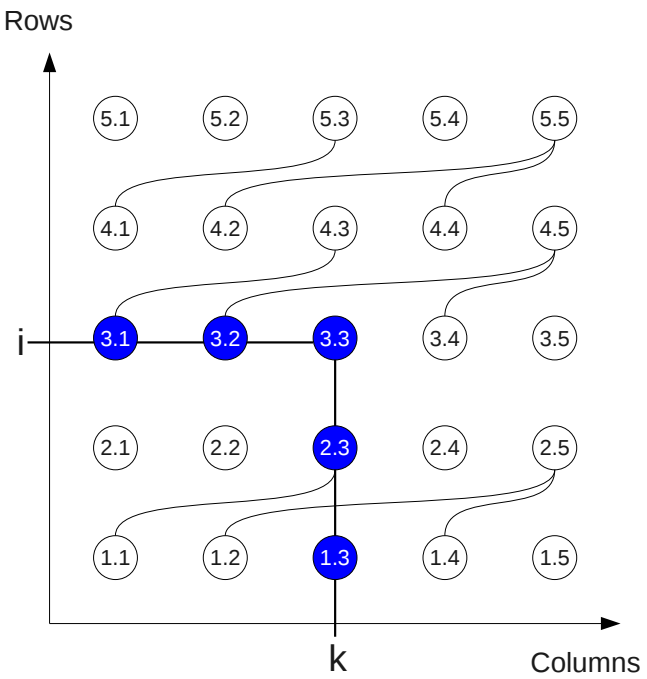
$$x_{ik} + x_{jl} - y_{ikjl} - 1 \leq 0, \forall \text{ edge } (ik.jl).$$

Needed because of negative edge weights.

Mathematical model : constraints

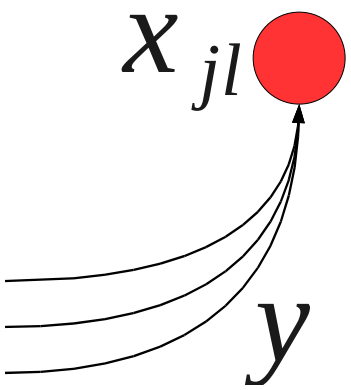
- Feasible alignment (ISV) :

$$\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k.$$



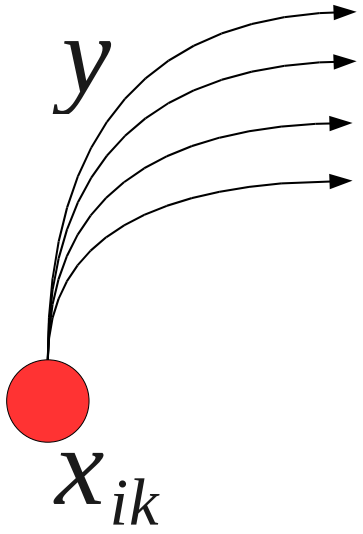
Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$



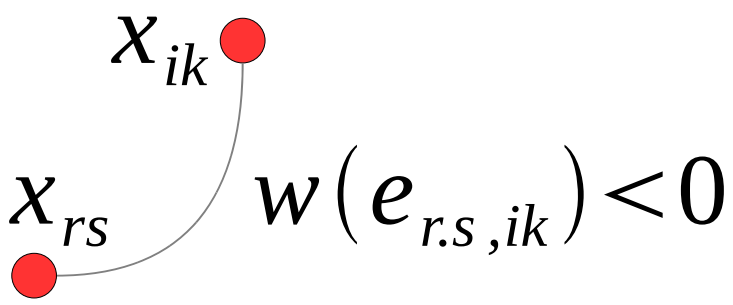
Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$
- Edge-driven head vertex activation : $x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl}$



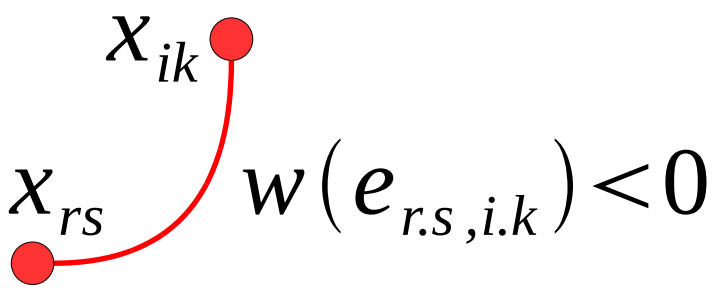
Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$
- Edge-driven head vertex activation : $x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl}$
- an edge $(r.s, i.k)$ is activated if its both extremities (rs) and (ik) are activated : $x_{ik} + x_{rs} - y_{rsik} \leq 1$



Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$
- Edge-driven head vertex activation : $x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl}$
- an edge $(r.s, i.k)$ is activated if its both extremities (rs) and (ik) are activated : $x_{ik} + x_{rs} - y_{rsik} \leq 1$



Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$
- Edge-driven head vertex activation : $x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl}$
- an edge $(r.s, i.k)$ is activated if its both extremities (rs) and (ik) are activated : $x_{ik} + x_{rs} - y_{rsik} \leq 1$
- The last constraint is lifted to :

Mathematical model : constraints

- Feasible alignment (ISV) : $\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall$ row $i, \forall$ col k.
- Edge-driven tail vertex activation : $x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl}$
- Edge-driven head vertex activation : $x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl}$
- an edge $(r.s, i.k)$ is activated if its both extremities (rs) and (ik) are activated : $x_{ik} + x_{rs} - y_{rsik} \leq 1$
- The last constraint is lifted to :
- Vertices-driven edge activation
  $x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1$ for $w(e_{rsik}) < 0$

Mathematical model : the entire Linear Integer Program

$$\max \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

Subject to :

$$\text{Feasible alignment (ISV)} : \sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k. \tag{3}$$

$$\text{Edge-driven tail vertex activation } : x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl} \tag{4}$$

$$\text{Edge-driven head vertex activation } : x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl} \tag{5}$$

$$\text{Vertices-driven edge activation} : x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1 \text{ for } w(e_{rsik}) < 0 \tag{6}$$

Lagrangian relaxation principle

IP problem P :

$$
\begin{array}{lcll}
Z_P & = & \max cx & \\
s.t. & & x \in X & \text{— “easy” constraints} \\
 & & Ax \leq b & \text{— “complicating” constraints}
\end{array}
$$

Lagrangian relaxation principle

IP problem P :

$$\begin{array}{lcll} Z_P & = & \max cx & \\ s.t. & & x \in X & \text{— “easy” constraints} \\ & & Ax \leq b & \text{— “complicating” constraints} \end{array}$$

Lagrangian relaxation $LR(\lambda)$:

$$Z_{LR(\lambda)} = \max \{cx + \lambda(b - Ax) | x \in X\}$$

Lagrangian relaxation principle

IP problem P :

$$\begin{array}{lll} Z_P & = & \max cx \\ s.t. & & x \in X \quad — \quad \text{“easy” constraints} \\ & & Ax \leq b \quad — \quad \text{“complicating” constraints} \end{array}$$

Lagrangian relaxation $LR(\lambda)$:

$$Z_{LR(\lambda)} = \max \{cx + \lambda(b - Ax) | x \in X\}$$

- $LR(\lambda)$ is also an IP problem, but easier to solve than P
- $LR(\lambda)$ is relaxation (upperbound) of P for *any* λ (i.e. $Z_P \leq Z_{LR(\lambda)}$)

Lagrangian relaxation principle

IP problem P :

$$\begin{array}{lll} Z_P & = & \max cx \\ s.t. & & x \in X \quad — \quad \text{“easy” constraints} \\ & & Ax \leq b \quad — \quad \text{“complicating” constraints} \end{array}$$

Lagrangian relaxation $LR(\lambda)$:

$$Z_{LR(\lambda)} = \max \{cx + \lambda(b - Ax) | x \in X\}$$

- $LR(\lambda)$ is also an IP problem, but easier to solve than P
- $LR(\lambda)$ is relaxation (upperbound) of P for *any* λ (i.e. $Z_P \leq Z_{LR(\lambda)}$)

Improving bounds / solving P :

- Lagrangian dual Z_{LD} : $\mathbf{Z_{LD}} = \min_{\lambda} \mathbf{Z_{LR(\lambda)}}$

Mathematical model : the entire Linear Integer Program

$$\max \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

$$\text{Feasible alignment (ISV)} : \sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k. \tag{7}$$

$$\text{Edge-driven tail vertex activation } : x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl} \tag{8}$$

$$\text{Edge-driven head vertex activation } : x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl} \tag{9}$$

$$\text{Vertices-driven edge activation} : x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1 \text{ for } w(e_{rsik}) < 0 \tag{10}$$

Mathematical model : the entire Linear Integer Program

$$\max \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

$$\text{Feasible alignment (ISV)} : \sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k. \tag{7}$$

$$\text{Edge-driven tail vertex activation } : x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl} \tag{8}$$

$$\text{Edge-driven head vertex activation } : x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl} \tag{9}$$

$$\text{Vertices-driven edge activation} : x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1 \text{ for } w(e_{rsik}) < 0 \tag{10}$$

Note : All constraints except (10) are the same as in CMO.

Mathematical model : the entire Linear Integer Program

$$\max \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

$$\text{Feasible alignment (ISV)} : \sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k. \tag{7}$$

$$\text{Edge-driven tail vertex activation } : x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl} \tag{8}$$

$$\text{Edge-driven head vertex activation } : x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl} \tag{9}$$

$$\text{Vertices-driven edge activation} : x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1 \text{ for } w(e_{rsik}) < 0 \tag{10}$$

Note : All constraints except (10) are the same as in CMO.

Equations (8) and (10) will be relaxed in order to apply a Lagrangian relaxation similar to the one for CMO approach. The relaxed problem can be solved by double dynamic programming in time $O(|V| + |E|)$ where $|E|$ is much larger than in CMO.

Mathematical model : the entire Linear Integer Program

$$\max \sum_{i,j \in A} \sum_{k,l \in B} \mathbf{d}(\mathbf{A_{ij}}, \mathbf{B_{kl}}) y_{ikjl} + \sum_{i \in A, k \in B} \mathbf{s}(\mathbf{A_i}, \mathbf{B_k}) x_{ik}$$

$$\text{Feasible alignment (ISV)} : \sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1, \forall \text{ row } i, \forall \text{ col } k. \tag{7}$$

$$\text{Edge-driven tail vertex activation } : x_{jl} \geq \sum_{i.k \in SOS_{jl}} y_{ikjl} \tag{8}$$

$$\text{Edge-driven head vertex activation } : x_{ik} \geq \sum_{jl \in SOS_{ik}} y_{ikjl} \tag{9}$$

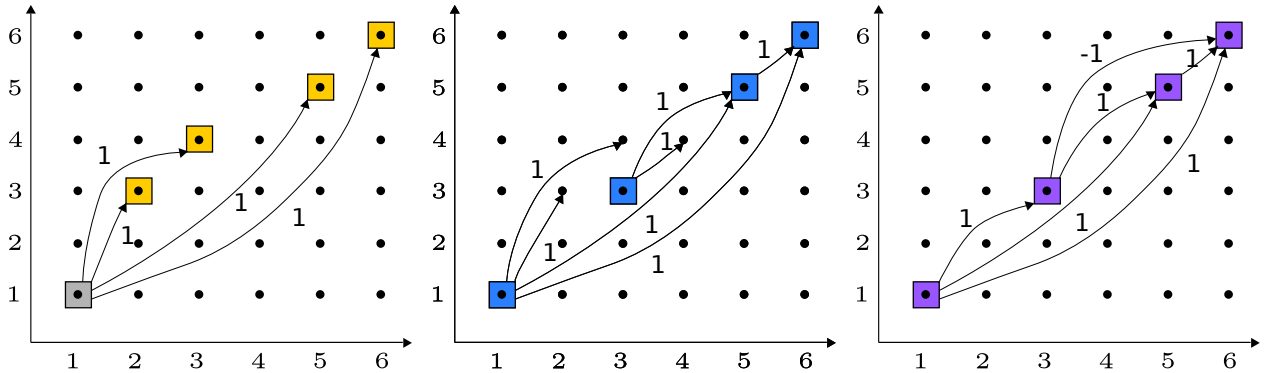
$$\text{Vertices-driven edge activation} : x_{ik} \leq \sum_{(r,s) \in SOS_{ik}} (y_{rsik} - x_{rs}) + 1 \text{ for } w(e_{rsik}) < 0 \tag{10}$$

Equations (8) and (10) will be relaxed in order to apply a Lagrangian relaxation similar to the one for CMO approach. The relaxed problem can be solved by double dynamic programming in time $O(|V| + |E|)$ where $|E|$ is much larger than in CMO.

Although theoretical not very different from CMO version, the Lagrangian relaxation implementation required significant programming effort.

Visualization of the computations in the relaxed problems

The relaxed problem is solved by dynamic programming in $O(|V| + |E|)$.



- **Left :** Local profit computation. Node 1.1 picks its best set of outgoing edges (i.e. maximizing this node's profit).
- **Center :** The solution of the relaxed problem. It is composed of the increasing path that is the solution of the global problem, colored in blue, together with the outgoing edges that these nodes picked in their respective local problem. The relaxed solution maximizes the sum of profits. Its score is $LR(\lambda) = 7$ and an upper bound on the optimal score.
- **Right :** The feasible solution that can be deduced from the relaxed solution. It is composed of the nodes that are activated in the relaxed solution together with all induced edges. Its score is $Z_{lb} = 4$ and represents a lower bound on the optimal score.

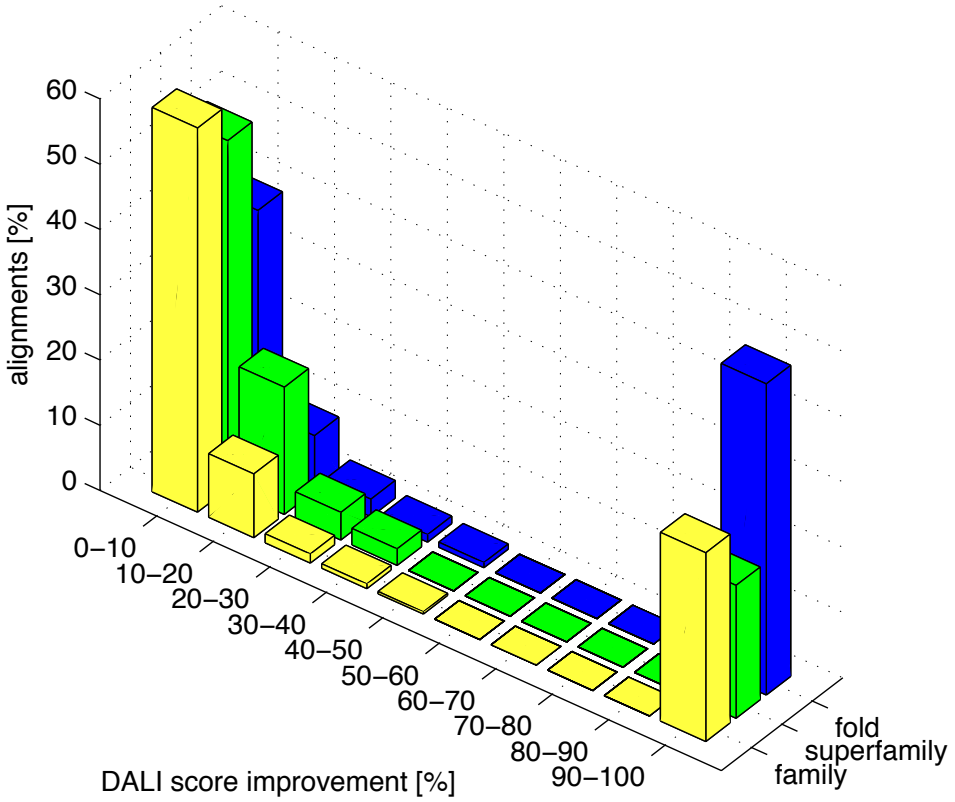
Computational results : DALI (Heuristic) vs DALIX (Exact)

Computations are done on cluster nodes each with two quad core 2.26 GHz Intel Xeon processors. The DALIX computation time limit for SCOPCath alignments is 30 CPU minutes per instance and for all other data sets 30 CPU hours per instance. In each branch-and-bound node are computed 1000 Lagrangian iterations.

	SKOLNICK		**SCOPCath**		**SISY**	**RIPC**
		Family	**Superfamily**	**Fold**		
Alignments	164	386	151	926	62	11
Positive z-score	164	359	141	302	61	11
DALIX optimal	136	143	14	31	11	2
DALI optimal	38	50	5	5	3	0
DALIX better	123	287	118	258	31	6
DALI better	3	16	14	30	27	5
missed by DALI		83	24	123		

Computational results I : Exact vs Heuristic solution

	SKOLNICK	SCOPCath			SISY	RIPC
		family	superfamily	fold		
Alignments	164	386	151	926	62	11
DaliX optimal	136	143	14	31	11	2
Dali optimal	38	50	5	5	3	0
Missed by Dali	0	**83**	**24**	**123**	0	0



Computational results I : DALI score improvement

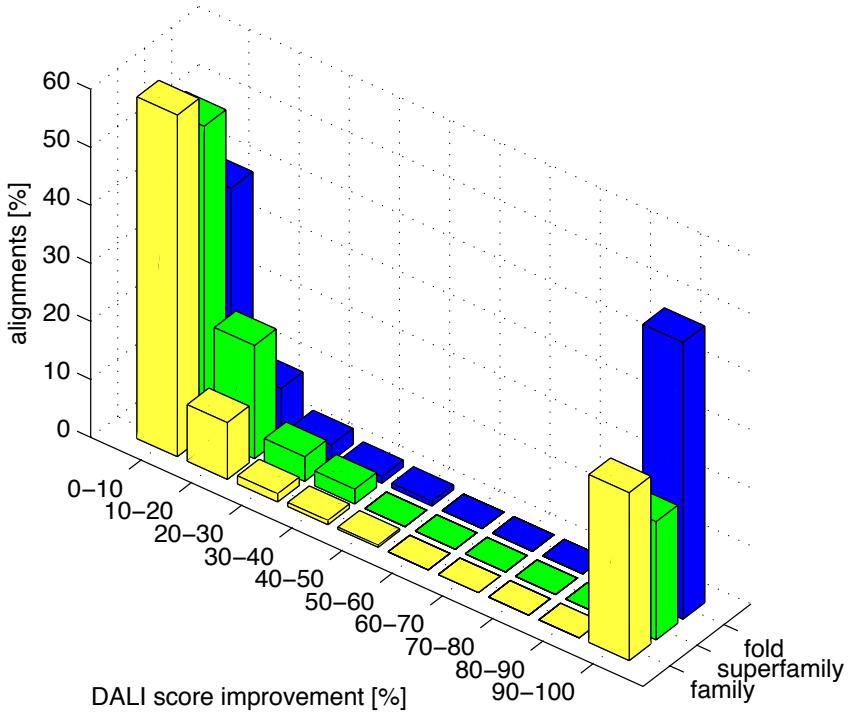


FIGURE: The barplot bins the percentages of DALI score improvement for the cases in which the DALIX alignment has positive z-score and is better than the DALI alignment. On family level, these are 278, on superfamily level 118 and on fold level 258 alignments. The improvement is computed with respect to the DALI alignment. The DALIX computation time limit is 30 CPU minutes. For most alignments, the score improvement is small. There is furthermore a large percentage of protein pairs that are entirely missed by DALI , i.e. for which DALI falsely reports that there is no structural similarity.

Conclusions

- First exact general algorithm for distance matrix alignments
- It is applicable to any distance matrix-based scoring scheme (i.e. is able to consider scoring functions with positive and negative values)
- The new tool allows to evaluate the precision of DALI-one of the most popular heuristic structural alignment method
- Some anomalies of the DALI heuristic (cases for which DALI entirely misses structural similarities)
- But globally the exact computations confirmed the high quality of the DALI heuristic (they are almost always very close to the optimum).