Algorithms and Learning for Protein Science

Structural alignments and analysis of static structures

Frederic.Cazals@inria.fr

Overview

- Theory/algorithms
  - Groups/Lie algebras $SO(3), \mathfrak{so}(3)$; $SE(3), \mathfrak{se}(3)$
  - Procruste problems – molecular distances
  - Spectral clustering
  - (Power diagrams and α-complexes)
- Protein science
  - Comparing static molecular structures
  - Rigid structural alignments (IRMSD)

    `https://sbl.inria.fr/doc/Molecular_distances-user-manual.html`

    `https://sbl.inria.fr/doc/Molecular_distances_flexible-user-manual.html`
  - Flexible structural alignments (`Kpax`)

    `https://sbl.inria.fr/doc/Kpax-user-manual.html`
  - Identification of rigid domains in proteins (`SPECTRALDOM`)

    `https://sbl.inria.fr/doc/group__Spectral__domain__explorer-package.html`
  - (Molecular surfaces, volumes, interfaces) (`Intervor`)

    `https://sbl.inria.fr/doc/Space_filling_model_surface_volume-user-manual.html`

    `https://sbl.inria.fr/doc/Space_filling_model_interface-user-manual.html`
  - (Cradle models)

    `https://sbl.inria.fr/doc/Molecular_cradle-user-manual.html`

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The Kpax flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

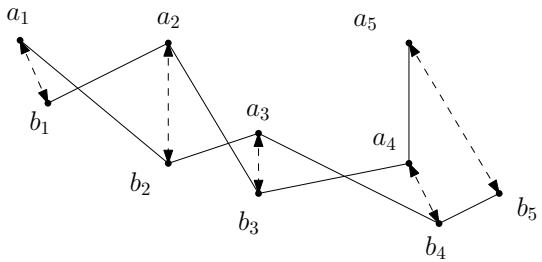
The combined RMSD

The AcrB efflux pump

The least Root Mean Square Deviation: lRMSD

– a geometric distance for two ordered point clouds

▷ Data: two point sets $A = \{a_i\}_{i=1,\dots,n}$, $B = \{b_i\}_{i=1,\dots,n}$ + alignment *i.e.* 1-1 correspondence $\mathcal{A} = \{a_i \leftrightarrow b_i\}$



$$\mathrm{RMSD}(A, B) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|a_i - b_i\|^2} \tag{1}$$

▷ least Root Mean Square Deviation:

$$\mathrm{lRMSD}(A, B) = \min_{g \in SE(3)} \mathrm{RMSD}(A, g \cdot B). \tag{2}$$

Importantly: the optimal rigid motion is obtained.

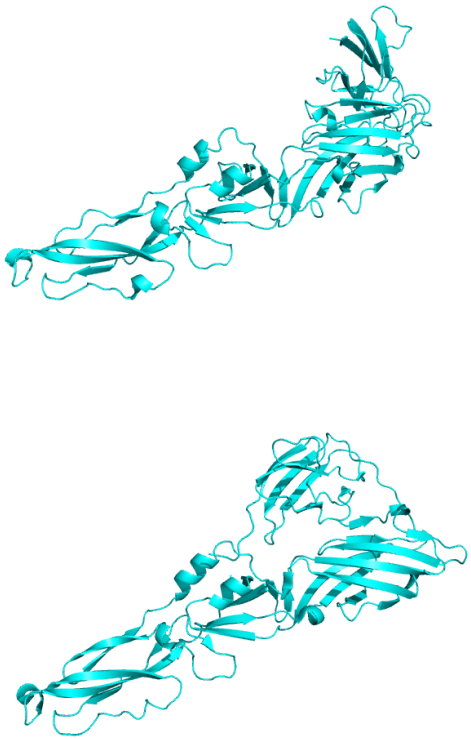
▷ Pros and cons:

- pro: easy to compute (quadratic problem, SVD)
- cons: medium range values for large structures tell nothing

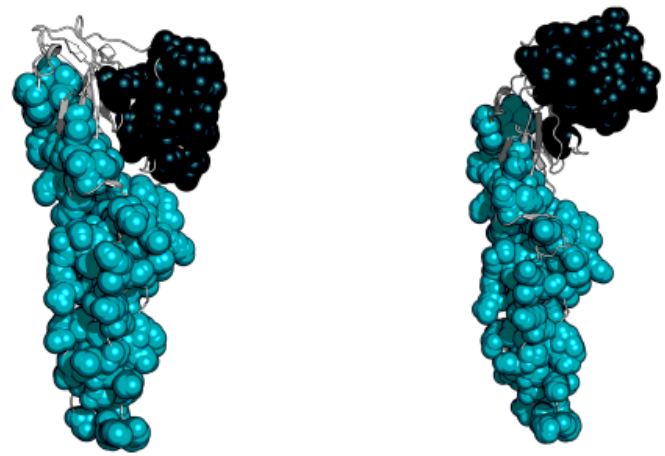
▷Ref: Kabsch, Acta Crystallographica Section A, 1976
▷Ref: Umeyama, IEEE PAMI 1991

Combined RMSD : TBEV glycoprotein in two different conformations pre and post fusion

▷ Classical analysis:



▷ Our motifs:



Motif	Alignment size	lRMSD
Large	88	1.69
Small	40	0.38

Statistics from Apurva:

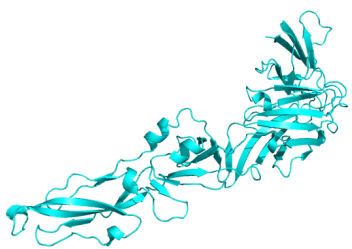
- 370 a.a. aligned
- lRMSD: 11.1Å

Structural similarity measures

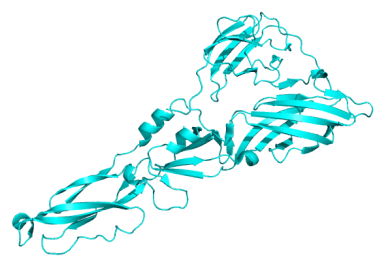
- Comparing conformations of:

  (PB1) the same molecule: mapping between atoms known (identical atoms)
  $\rightarrow$ a geometric problem
  (PB2) two related molecules (e.g. two polypeptide chains of different length)
  $\rightarrow$ a dual combinatorial (common contacts) + geometric problem (how similar?)

- (PB1) Comparing conformations:
  - issue #1: for large structures, small numbers ~ 1 are fine; medium / larger number are often meaningless.
  - issue #2 (related): a score does not give a mapping.
- (PB2) Comparing related molecules:
  - focus on topology (e.g. contact map overlap) or geometry (isometric regions)
  - in any case: the longer the alignment the worse the geometric measure



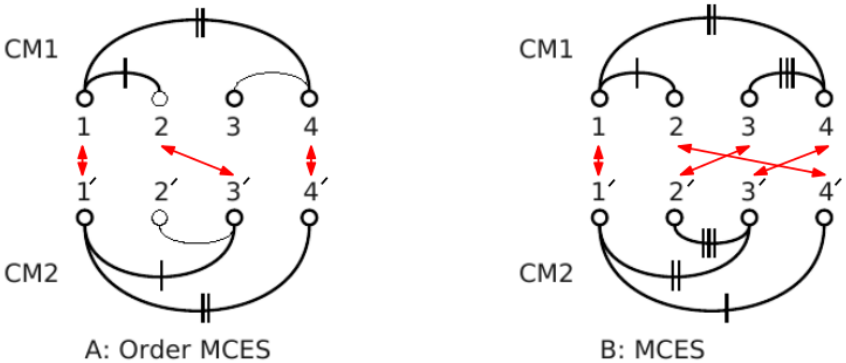
TBEV pre-fusion



TBEV post-fusion

Contact map overlap with Apurva

▷ Contact map of a polypeptide chain



A graph stating when two amino-acids (a.a.) are in close proximity (e.g. distance between their C_α carbons).

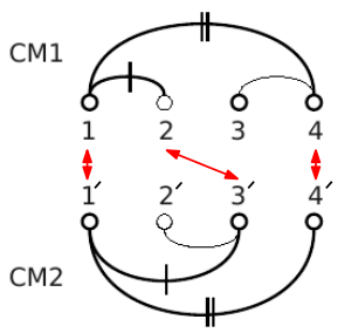
▷ Contact map overlap (CMO):

- Find subsets of vertices I and J yielding the largest set of common edges in their induced graphs
- Constraint: amino-acids are linearly ordered $\Rightarrow$ matched vertices must be so

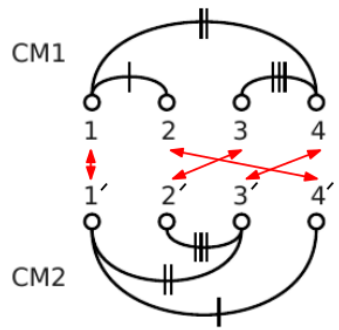
▷ Hardness: decision problem is NP-complete.
▷ Algorithm: integer programming model + branch-and-bound algorithm + Lagrangian relaxation.
▷Ref: Papadimitriou et al, FOCS 1999
▷Ref: R. Andonov, N. Malod-Dognin, and N. Yanev, J. of Computational Biology, 2011

Contact map overlap (CMO) as an Integer Linear Program

- Vertex sets: $I = (i_1, i_2, \ldots, i_s), i_1 < i_2 < \cdots < i_s$ and $J = (j_1, j_2, \ldots, j_s), j_1 < j_2 < \cdots < j_s$: arbitrary subsets of vertices from the first and second contact maps, respectively.
- Alignment: in the mapping $i_k \leftrightarrow j_k, k = 1, 2, \ldots, s$, the edge (k, l) is **common** if and only if both edges (i_k, i_l) and (j_k, j_l) exist in the two contact maps.
- CMO: find the sets I and J maximizing the number of common edges. Since amino-acids are linearly ordered, crossings are not allowed.



A: Order MCES



B: MCES

Pairwise structural alignments: iterative methods a-la Kpax

- Goal: find a structural alignment between two molecules/conformations
- Matching vs rigid alignment:
  - Finding the matching given the relative position: dynamic programming
  - Finding the relative position given the matching: quadratic program
- Iterative alignments:

(1) Compute a seed alignment
(2) Iterate until some fixed point:
  (3) Post-process the structures
  (4) Compute the rigid superposition induced by the alignment (procruste)
  (5) Perform Dynamic Programming to obtain a new alignment

▷Ref: Ritchie et al, Bioinformatics, 2012

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The Kpax flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

The combined RMSD

The AcrB efflux pump

Orthogonal matrices

- Def:
  - A matrix of $\mathbb{R}^{d\times d}$ such that U satisfies $U^{\mathsf{T}}U = \mathrm{I}_d$.
  - Premultiplying $UU^{\mathsf{T}} = \mathrm{I}_d$ by U^{-1} yields $U^{\mathsf{T}} = U^{-1}$.

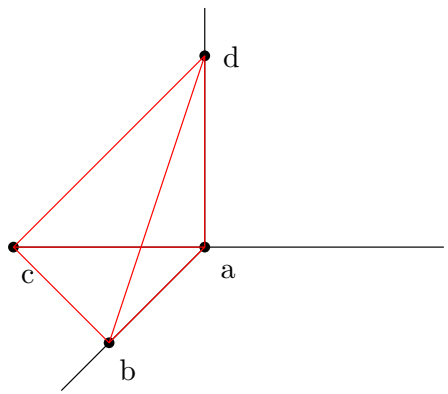
- Preservation of the dot product, whence distances and angles

$$\langle Qu, Qv,\rangle = u^{\mathsf{T}}Q^{\mathsf{T}}Qv = \langle u, v\rangle, \tag{3}$$

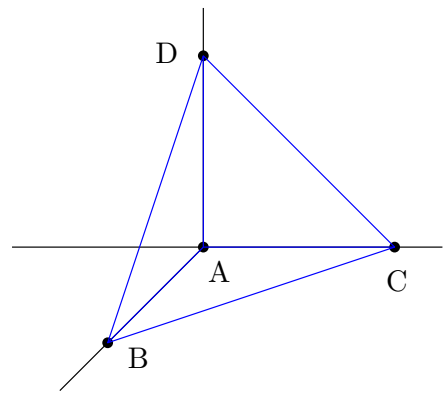
$$\langle Qu, Qu,\rangle = u^{\mathsf{T}}Q^{\mathsf{T}}Qu = \|u\|^2. \tag{4}$$

Since they preserve distances, they define *isometries*.

- Orientation preservation: two cases
  - $\det(U) = 1$: orientation preserving – rotations.
  - $\det(U) = -1$: involves a reflection – mix of rotations and one reflection.



(A)



(B)

Related (important) groups

- Def: A group is a non empty set G and a binary operation $\cdot : G \times G \to G$:
  - (i) Product is associative: $(AB)C = A \cdot (B \cdot C)$
  - (ii) There exists an identity I such that $I \cdot A = A \cdot I = A$
  - (iii) Each element has an inverse: $A \cdot A^{-1} = I$
- Groups of interest related to orthogonal matrices:
  - $O(3)$: the orthogonal group
  - $SO(3)$: subgroup of $O(3)$ with $\det(U) = 1$: the special orthogonal group aka rotation group
- Euclidean space $\mathbb{E}^d$: rigid transformations vs rigid motions
  - $E(3)$: the Euclidean group, defining **isometries** or **rigid transformations**:

$$\phi(v) = Uv + t, \text{ with } U^{\mathsf{T}}U = \mathsf{I}_3 \text{ and } t \text{ a translation.} \tag{5}$$

  - $SE(3)$ or $E^+(3)$: special Euclidean group, involving orthogonal matrices of det. 1. Define **rigid motions**.

Lie groups and associated algebras

- ▷ Groups which are also manifolds: *Lie groups*
  - $SO(3)$ and $\mathfrak{so}(3)$: for rotations
  - $SE(3)$ and $\mathfrak{se}(3)$: for rigid motions of $\mathbb{E}^3$
- ▷ Moving back and forth: the exp and log operators

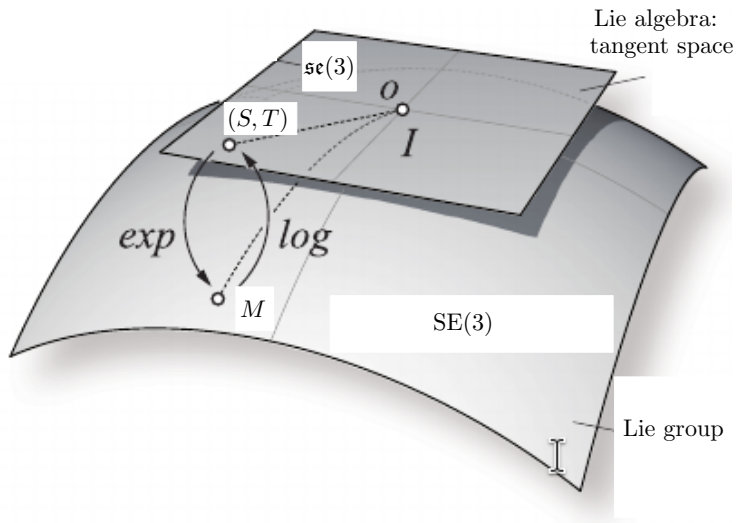


Figure: **Lie group $SE(3)$ and Lie algebra $\mathfrak{se}(3)$: mapping back and forth using the exponential map.** Fig. adapted from Hagemann et al.

Rotations in $\mathbb{E}^3$ and change of coordinate system

▷ Rotation: of angle θ is an orthonormal basis $\mathcal{B}' = (i, j, k)$ along the axis k

$$Rot_{\mathcal{B}'}(k, \theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

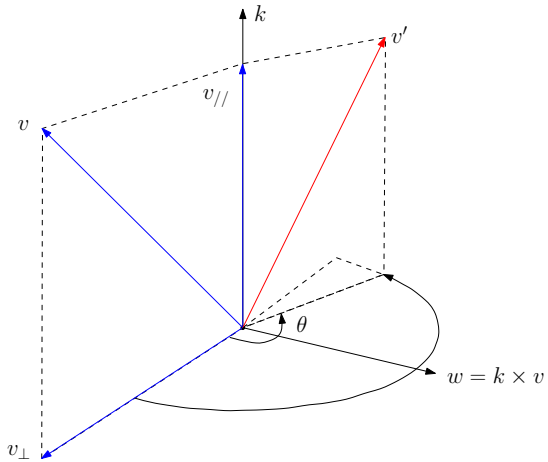
▷ In the reference coordinate system: denoting P the matrix of change of base from $\mathcal{B}'$ to the standard basis $\mathcal{B}$, the equation of the rotation is the std basis $\mathcal{B}$ is therefore

$$R = Rot(k, \theta) = PRot_{\mathcal{B}'}(k, \theta)P^{-1}. \tag{7}$$

▷ NB: applying $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ to Eq. (6): $\mathrm{Tr}(R) = 2\cos\theta + 1$.

Rodrigues' formula for rotations in $\mathbb{E}^3$

Proposition. 1. (Geometric version) Consider now a unit vector $s = (s_0, s_1, s_2)^T$, and the associated The rotation about k by the angle θ satisfies Eq. 8



$$v' = v\cos\theta + (k \times v)\sin\theta + k(\langle k, v\rangle)(1-\cos\theta). \quad (8)$$

NB: triple cross product formula

$a \times (b \times c) = \langle a, c\rangle)b - \langle a, b\rangle c$

Proposition. 2. (Linear algebra version) In matrix form, the rotation matrix is given by Rodrigues' formula:

$$R = \mathrm{I} + \sin\theta S + (1 - \cos\theta)S^2. \quad (9)$$

with S the skew symmetric matrix

$$S = \begin{pmatrix} 0 & -s_2 & s_1 \\ s_2 & 0 & -s_0 \\ -s_1 & s_0 & 0 \end{pmatrix} \quad (10)$$

Rotations: $SO(3) \Leftrightarrow \mathfrak{so}(3)$

- Consider $S \in \mathfrak{so}(3)$ and $R \in SO(3)$ with:

$$S = \begin{pmatrix} 0 & -s_2 & s_1 \\ s_2 & 0 & -s_0 \\ -s_1 & s_0 & 0 \end{pmatrix} \tag{11}$$

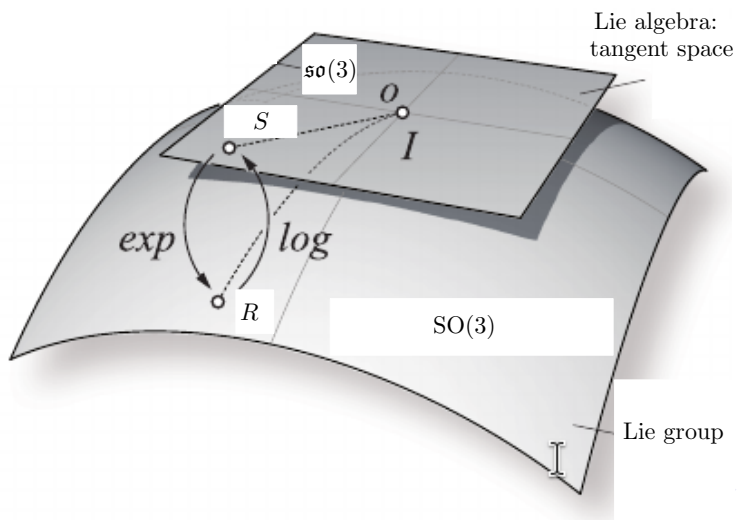
- Exponentiation of S yields R:

$$R = \exp(S) = \sum_{n \geq 0} \frac{S^n}{n!} = I + \frac{\sin(\theta)}{\theta} S + \frac{1 - \cos(\theta)}{\theta^2} S^2. \tag{12}$$

NB: the finite sum stems from the identity $S^3 = -\theta^2 S$.

- Logarithm of R yields S: subtracting Eq. 12 and its transpose yields

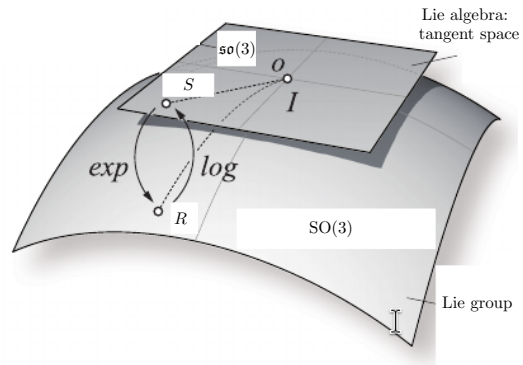
$$S = \frac{\theta(R - R^T)}{2sin(\theta)} = \log R. \tag{13}$$



Interpolating rotations via $SO(3)$ and $\mathfrak{so}(3)$

- Goal: interpolate between two rotations R_0 and R_1 – or I and $R_1 R_o^{-1}$
- From the theory of Lie groups/algebras:

$$F(t) = \exp(t \log(R_1 R_0^{-1}))R_0, t \in [0,1]. \quad (14)$$



- Application to interpolate between two rotations:
  1. Compute $R = R_1 R_0^{-1} = R_1 R_0{}^{\mathsf{T}}$
  2. Using $\mathrm{Tr}(R) = 1 + 2\cos\theta$ and compute $S = \log R = \frac{\theta}{2\sin\theta}(R - R^{\mathsf{T}})$
  3. Change $S \to tS$ and $\theta \to t\theta$, and take the exp
  4. Multiply by R_0, whence

$$F(t) = \left[I + \frac{\sin(t\theta)}{t\theta}(tS) + \frac{1-\cos(t\theta)}{(t\theta)^2}(tS)^2\right]R_0. \quad (15)$$

Interpolating rigid motions via $SE(3)$ and $\mathfrak{se}(3)$

▷ Rigid motion $M \in SE(3)$, representation: with rotation matrix R and a 3x1 translation vector T, consider the matrix

$$M = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{16}$$

The inverse matrix satisfies

$$M^{-1} = \begin{bmatrix} R^{\mathsf{T}} & -R^{\mathsf{T}} T \\ 0 & 1 \end{bmatrix} \tag{17}$$

We write a rigid transformation M as an exponential:

$$M = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} = exp\left(\begin{bmatrix} S & U \\ 0^T & 0 \end{bmatrix} \right) = \begin{bmatrix} exp(S) & V\ U \\ 0^T & 1 \end{bmatrix} \tag{18}$$

where S is a skew-symmetric matrix and U is a 3x1 vector. We use the equations for $SO(3)$ logarithm to find S.

▷ Interpolation between two motions M_0 and M_1:

$$H(t) = \exp(t \log(M_1 M_0^{-1}))M_0, t \in [0, 1] \tag{19}$$

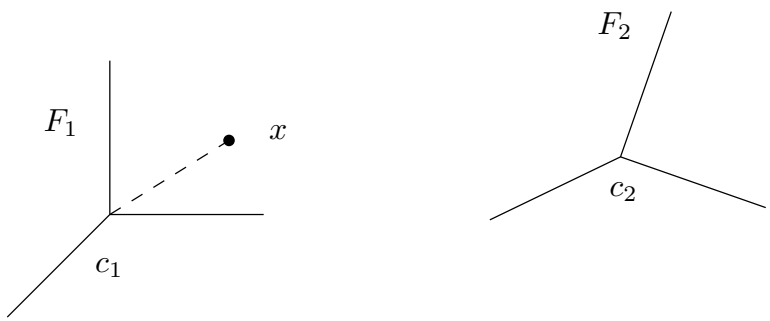
Rigid frames and point trajectories

▷ Change of coordinate system via translation and rotation: assume we are given two frames, represented by orthogonal matrices, and a point x in the first frame. The following linear transformation rewrites x into the second one:

$$x' = F_2 {F_1}^{\mathsf{T}} (x - c_1) + c_2. \tag{20}$$

The calculation reads as follows:

- ${F_1}^{\mathsf{T}}(x - c_1)$ rewrites x in the base F_1
- Multiplication by F_2 rewrites the previous vector in F_2
- Finally, we translate by c_2



▷ Motion of a point in a moving frame: x expressed in F_1, which is moved to F_2 as a rigid body. Using $R(t)$ and $T(t)$ the interpolated transformations between the two rotations and the translation, the trajectory of point x is given by

$$x(t) = R(t)(x(c) - c_1) + T(t). \tag{21}$$

NB: $R(0) = \mathsf{I}, R(1) = F_2 {F_1}^{\mathsf{T}}; T(0) = c_1, T(1) = c_2$

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the IRMSD

The Kpax flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

The combined RMSD

The AcrB efflux pump

Proc(r)uste

- Du grec ancien: "*Celui qui martèle pour allonger*"
- Wikipedia: ≪ **Brigand de la mythologie grecque qui torturait ses victimes ainsi: il les allongeait sur un lit et coupait leurs membres, ou les étirait, afin qu'elles correspondissent exactement aux mensurations de celui-ci.** ≫

`https://fr.wikipedia.org/wiki/Procuste`

Three optimization problems

Let P and Q be real $d \times n$ matrices – representing two point clouds of size n.

Problem 1. Find an orthogonal matrix U of size $d \times d$

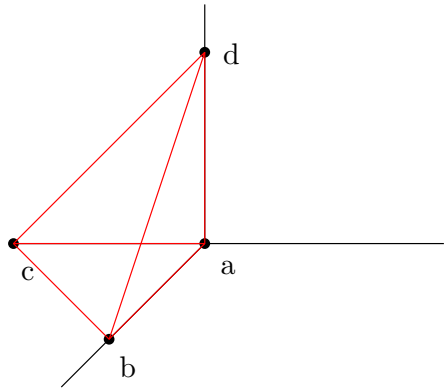
$$\arg\min_U \sum_i \|Uq_i - p_i\|^2 = \arg\min_U \|UQ - P\|_F^2 \tag{22}$$

Problem 2. (Constrained procruste) Solve Eq. (22), with U to be a rotation matrix, that is $\det(U) = 1$.

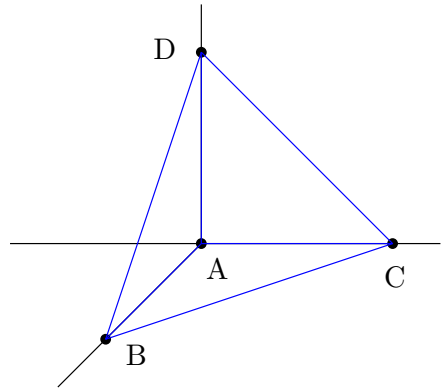
Problem 3. (Rigid motion) Let ϕ be an orientation rigid motion of $\mathbb{R}^d \mapsto \mathbb{R}^d$, that is $phi(q) = Uq + t$, with U an orthonormal matrix of determinant one, and t a translation. Find ϕ minimizing

$$\arg\min_\phi \sum_i \|\phi(q_i) - p_i\|^2 \tag{23}$$

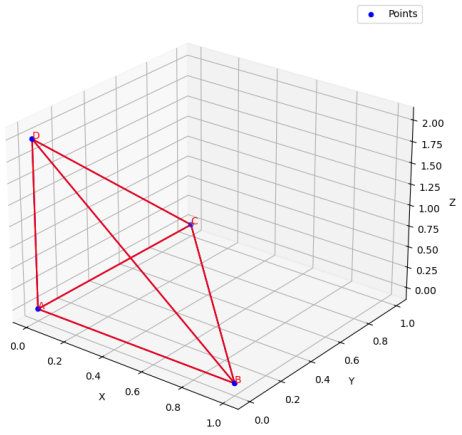
Orthogonal versus constrained problem: comparison



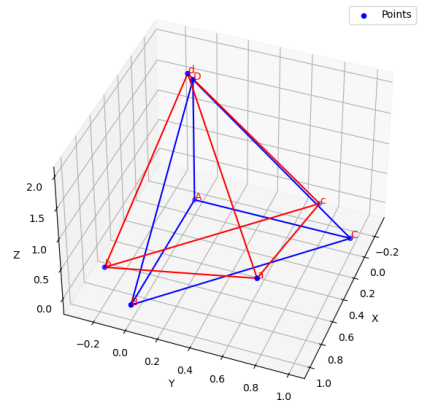
(A)



(B)



(C)



(D)

Figure: **Superimposition two 3D point clouds: (A,B)** Q and P **(C)** Via optimal procruste solution: translation + reflexion **(D)** Via optimal rotation

Orthogonal procruste problem: solution

Using $\|A\|_F^2 = \mathrm{Tr}(AA^\mathsf{T})$ and $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ and $\mathrm{Tr}(A) = \mathrm{Tr}(A^\mathsf{T})$ we get:

$$\begin{aligned}
\|UQ - P\|_F^2 &= \mathrm{Tr}((UQ - P)^\mathsf{T}(UQ - P)) && (24)\\
&= \mathrm{Tr}((Q^\mathsf{T}U^\mathsf{T} - P^\mathsf{T})(UQ - P)) && (25)\\
&= \mathrm{Tr}(Q^\mathsf{T}Q) + \mathrm{Tr}(P^\mathsf{T}P) - \mathrm{Tr}(Q^\mathsf{T}U^\mathsf{T}P) - \mathrm{Tr}(P^\mathsf{T}UQ) && (26)\\
&= \mathrm{Tr}(Q^\mathsf{T}Q) + \mathrm{Tr}(P^\mathsf{T}P) - \mathrm{Tr}(P^\mathsf{T}UQ) - \mathrm{Tr}(UQP^\mathsf{T}) && (27)\\
&= \mathrm{Tr}(Q^\mathsf{T}Q) + \mathrm{Tr}(P^\mathsf{T}P) - 2\mathrm{Tr}(UQP^\mathsf{T}). && (28)
\end{aligned}$$

Assume we wish to maximize $\mathrm{Tr}(UQP^\mathsf{T})$, and consider the SVD

$$QP^\mathsf{T} = V\Sigma W^\mathsf{T}. \quad (29)$$

We get

$$\begin{aligned}
\mathrm{Tr}(UQP^\mathsf{T}) &= \mathrm{Tr}(UV\Sigma W^\mathsf{T}) = \mathrm{Tr}(W^\mathsf{T}UV\Sigma) && (30)\\
&= \mathrm{Tr}(Z\Sigma) \text{ with } Z = W^\mathsf{T}UV. && (31)
\end{aligned}$$

Note that Z is orthogonal since it is the product of orthogonal matrices.
Note the following upper bound: $\mathrm{Tr}(Z\Sigma) = \sum_i z_{ii}\sigma_i \leq \sum_i \sigma_i$.
On the other hand, taking $U = WV^\mathsf{T}$ yields $\mathrm{Tr}(Z\Sigma) = \sum_i \sigma_i$.

▷Ref: G. Golub and C.F. Van Loan, Matrix computations, 2012

Problem 3/rigid motions: centering the data

Lemma 4. With ϕ be a rigid motion form $\mathbb{R}^d$, with $\phi(\overline{q}) \neq \overline{p}$, define the affine function

$$\tau : \mathbb{R}^d \mapsto \mathbb{R}^d, \quad \tau(y) = \phi(y) - \overline{q} + \overline{p}. \tag{32}$$

If $\phi(\overline{q}) \neq \overline{p}$, then:

$$\Delta(\tau) < \Delta(\phi). \tag{33}$$

(Proof sketch)
With U a $d \times d$ orthonormal matrix and $t \in \mathbb{R}^d$, assume that $\phi(q) = Uy + t$. Then $\tau(q) = Uq - U\overline{q} + \overline{p}$.

$$\|\phi(q_i) - p_i\|^2 - \|\tau(q_i) - p_i\|^2 \tag{34}$$

$$= (Uq_i + t - p_i)^{\mathsf{T}}(Uq_i + t - p_i) - Uq_i - U\overline{q} + \overline{p} - p_i{}^{\mathsf{T}}(Uq_i - U\overline{q} + \overline{p} - p_i) \tag{35}$$

$$= 2(Uq_i - p_i + t)^{\mathsf{T}}(U\overline{q} - \overline{p} = t) - (U\overline{qp} + t)^{\mathsf{T}}(U\overline{qp} + t) \tag{36}$$

Therefore

$$\Delta(\phi) - \Delta(\tau) = \sum_i \|\phi(q_i) - p_i\|^2 - \|\tau(q_i) - p_i\|^2 \tag{37}$$

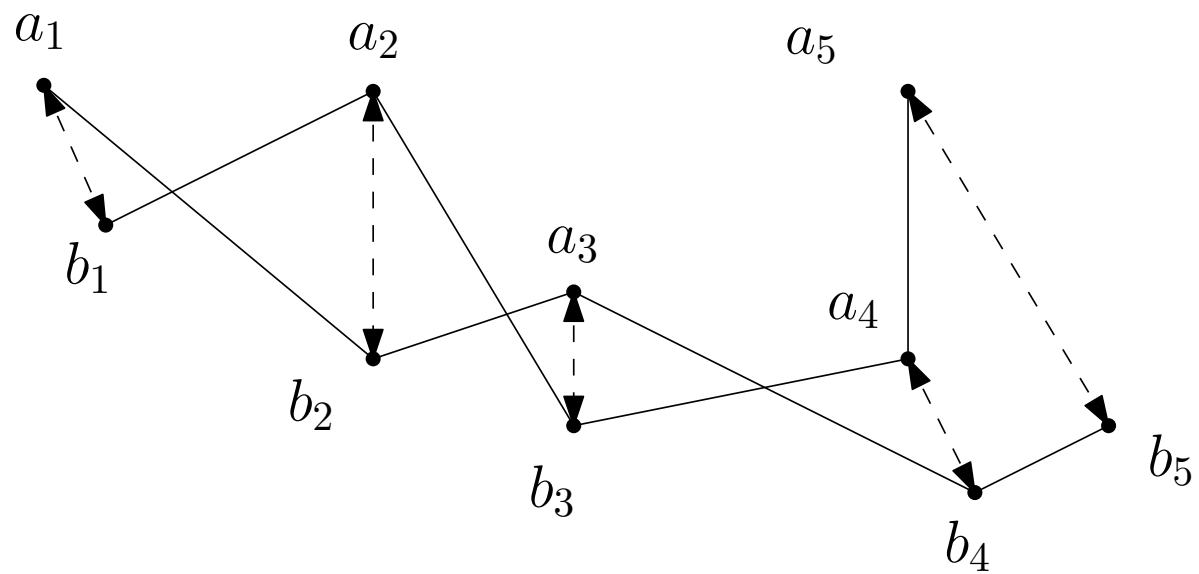
$$= \dots \tag{38}$$

$$= n\,\|U\overline{q} - \overline{p} + t\|^2 = n\,\|\phi(\overline{q}) - \overline{p}\|^2 > 0. \tag{39}$$

The Kbasch-Umeyama algorithm for centered data

1: **procedure** Kabsch-Umeyama(P: $d \times n$ matrix,Q: $d \times n$ matrix)
2: Compute the $d \times d$ matrix $M = QP^{\mathsf{T}}$
3: Compute the SVD $M = VSW^{\mathsf{T}}$
4: Set $s_1 = \ldots, d_{d-1} = 1$
5: Set $s_d = \text{sign}(|VW|)$
6: Set $\tilde{S} = \text{Diag}((s_1, \ldots, s_d))$
7: Return $U = W\tilde{S}V^{\mathsf{T}}$
8: **end procedure**

NB: Using the identity matrix I_d instead of $\tilde{S}$ yields the optimal rigid transformation, which may not be a rigid motion (involved a reflection)



▷Ref: J. Lawrence et al, J. of research of the National Institute of Standards and Technology, 2019

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The `Kpax` flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From `SPECTRUS` to `SPECTRALDOM`
- Gallery of results

The combined RMSD

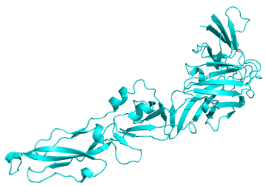
The AcrB efflux pump

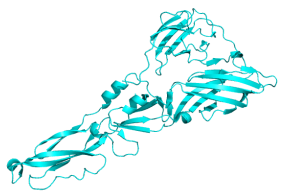
Conformational changes and rigid domains:

the example of fusion proteins

▷ Ex: TBEV glycoprotein (class II fusion): pre. versus post fusion

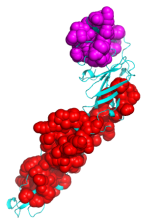
– Classical analysis:

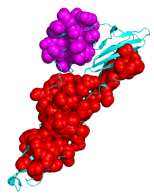




Statistics from Apurva:
370 a.a. aligned; lRMSD = 11.1Å

– Our motifs:





pre-fusion post-fusion

Motif	Alignment size	lRMSD
Red	88	1.69
Purple	40	0.38

Rigid domains and flexible linkers: intuition

Definition 5. Given two sets of a.a. $M_A = \{a_{i_1}, \ldots, a_{i_s}\} \subset S_A$ and $M_B = \{b_{i_1}, \ldots, b_{i_s}\} \subset S_B$, and a one-to-one alignment $\{(a_{i_j} \leftrightarrow b_{i_j})\}$ between them, we define the ***least RMSD ratio*** as follows:

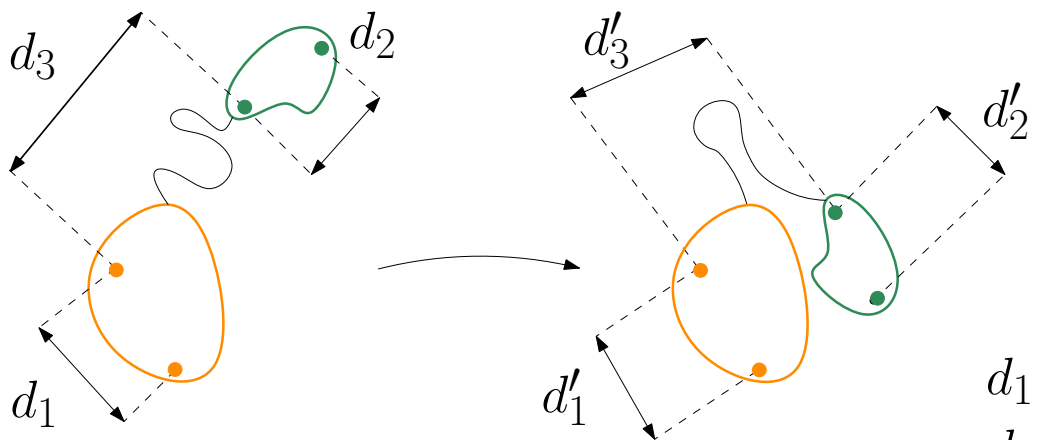
$$r_{\mathrm{lRMSD}}(M_A, M_B) = \mathrm{lRMSD}(M_A, M_B)/\mathrm{lRMSD}(S_A, S_B). \tag{40}$$

The sets M_A and M_B are called *structural motifs* provided that

$$|M_A| = |M_B| \geq s_0 \text{ and } r_{\mathrm{lRMSD}}(M_A, M_B) \leq r_0,$$

for appropriate thresholds s_0 and r_0.

▷ Quasi-isometric deformation: (selected) distances (almost) preserved



$$d_1 \sim d'_1$$
$$d_2 \sim d'_2$$
$$d_3 \neq d'_3$$

Flexible aligners in general

Illustration with `Kpax`

▷ Input: two polypeptide chains A and B, with n and m residues

Definition 6. Pose of A and B: a relative positioning of A and B.

▷ Goal: find two subsets of residues $S_A = \{a_i\}_{i=1,\ldots,r}$ and $S_B = \{b_j\}_{j=1,\ldots,r}$ of these chains and a one-to-one-mapping between them, which we denote $\mathcal{A} = \{(a_i, b_i)\}$ so as to maximize:

- The length r of the alignment,
- Some quality measure between the two subsets, *e.g.* the lRMSD .

NB: both objectives are contradicting.

▷ General strategy:

- Given fixed positions of the chains: used DP to find the alignment
- Given the alignment: find the best rigid motion
- Upon reaching some fixed point: stop

Kpax: pose dependent and independent scores

- Pose independent scores K_{ij}: defined later

Definition 7. (K-score, pose independent) Assume K_{ij} is a pose independent score defined for two amino acids $a_i \in A$ and $b_j \in B$. The K-score of an alignment $\mathcal{A}(A, B)$ is the sum $K = \sum_{(i,j)\in\mathcal{A}} K_{ij}$.

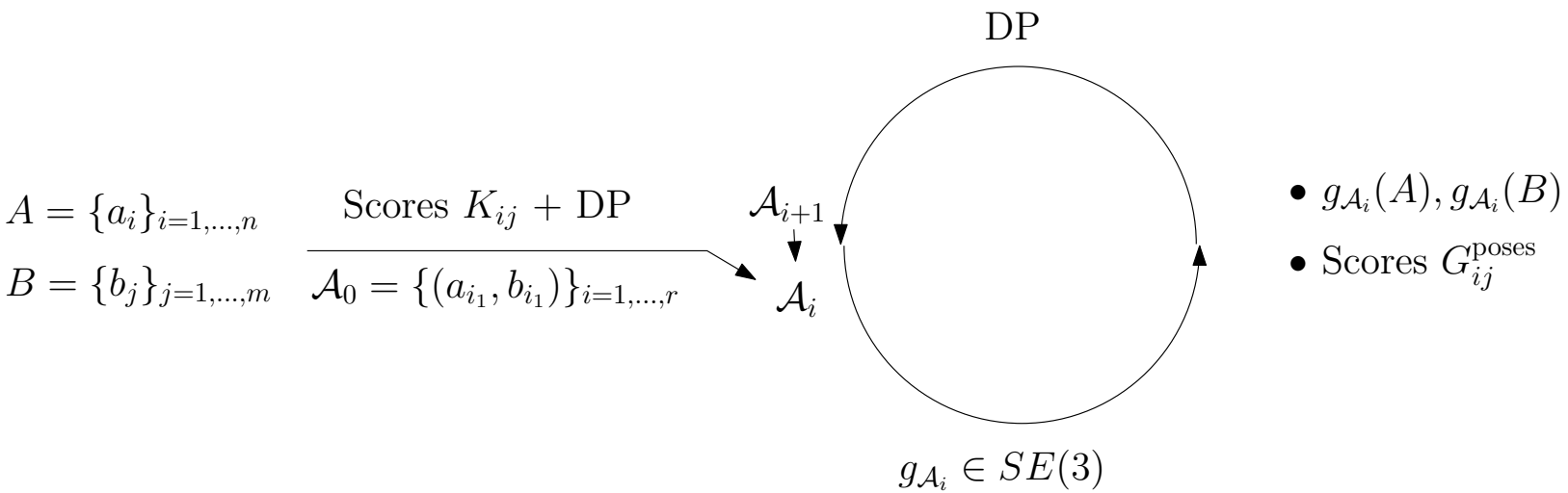
- Pose dependent scores G_{ij}^{pose}

Definition 8. (G-score, pose dependent) Given a pose of the two chains, let $R_{i,j}$ be the distance between the C_α carbons of residue i on chain A and residue j on chain B. The *G-score* of these residues is defined by The *G-score* of an alignment $\mathcal{A}$ is defined by:

$$G_{AB} = \sum_{(i,j)\in\mathcal{A}} G_{ij}^{pose} \text{ with } G_{ij}^{pose} = exp(-R_{i,j}^2/4\sigma_{pose}^2). \tag{41}$$

Kpax: generic algorithm

procedure `Kpax`(Chains P and Q)
  Compute the $n \times m$ local scores K_{ij} – Eq. (45)
  Use dynamic programming to obtain an initial alignment $\mathcal{A}_0$
  $i \leftarrow 0$
  Initialize the poses : $A^{(i=0)} \leftarrow A$, $B^{(i=0)} \leftarrow B$
  while True **do**
    Compute the opt rigid motion $g_{\mathcal{A}_i}$
    Compute the poses $A^{(i+1)} = g_{\mathcal{A}_i}(A)$ and $B^{(i+1)} = g_{\mathcal{A}_i}(B)$
    (Optional: prune long (say > 8Å) edges
    Compute the $n \times m$ pose dependent scores G_{ij}^{pose} – Eq. 41
    Use dynamic programming to obtain a new alignment $\mathcal{A}_{i+1}$
    if Combinatorial or Geometric criterion met **then**
      Break
    end if
  end while
  Return final statistics
end procedure

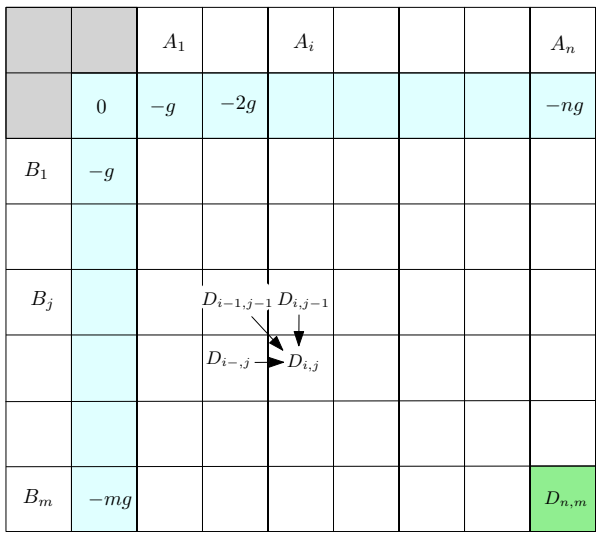


Dynamic programming based on a score – K_{ij} or G_{ij}^{pose}

- Fill the DP table from top left to bottom right, using the equations

$$D_{i,j} = \max\left(\begin{cases} D_{i-1,j-1} + s_{ij}, \\ D_{i,j-1} - P_i^A, \\ D_{i-1,j} - P_j^B \end{cases}\right). \tag{42}$$

- Trace back a path from (n, m) to the origin $(0, 0)$, following the arrow selected during step 1.



▷ Complexity (space and time): $O(n + m)$

▷ Importance of gaps: isolating independent/disconnected motifs

Canonized paths: definition

- ▷ Moving a.a. A_i or B_j into a canonical position
  - Translation to locate the C_α at the origin
  - Rotation to place the C along the negative z axis
  - Rotation about the z axis to plant the N atom in the xz plane with positive x coordinate

Definition 9. (Canonized n-path) Consider the i-th C_α carbon, and assume its polypeptide chain has been transformed using $\underline{K}_i$. The *canonized* n-path associated of this C_α is the set of $\leq 2n$ 3D points providing the coordinates of the C_α of index $k \in [-n, -(n-1), \ldots, -1, 1, \ldots, n-1, n]$, when they exist.

The *support* of a canonized n-path is the subset of $[-n, -(n-1), \ldots, -1, 1, \ldots, n-1, n]$ corresponding to existing C_α atoms. Supports denoted $I_A[i]$ and $I_B[j]$.

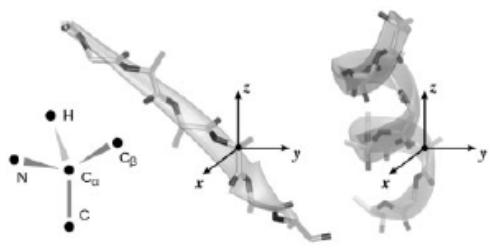
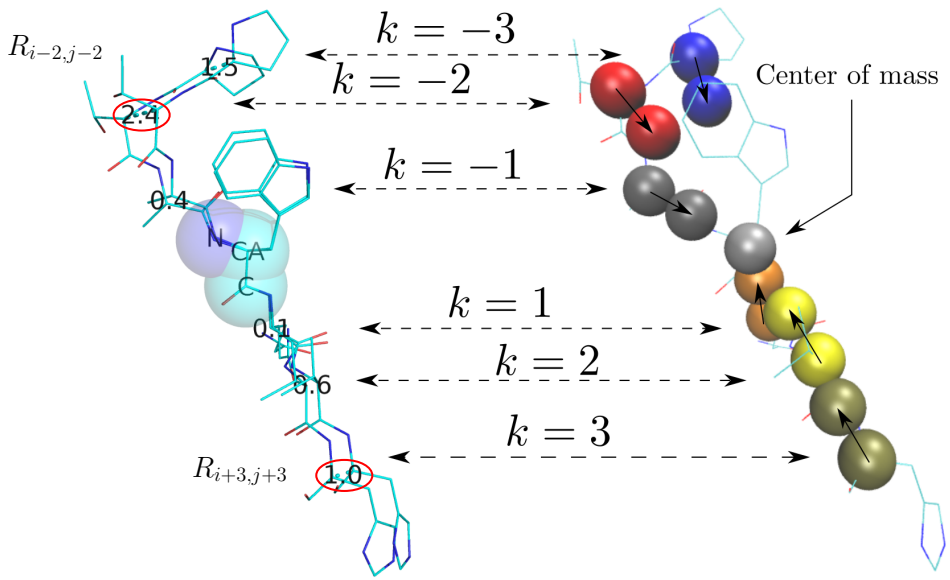


Figure: `Kpax`: **canonical frame**

Using the supports, define:

$$R_{i+k,j+k} = \left\| C^A_{\alpha;i+k} C^B_{\alpha;j+k} \right\|, k \in I_A[i] \cap I_B[j]. \quad (43)$$

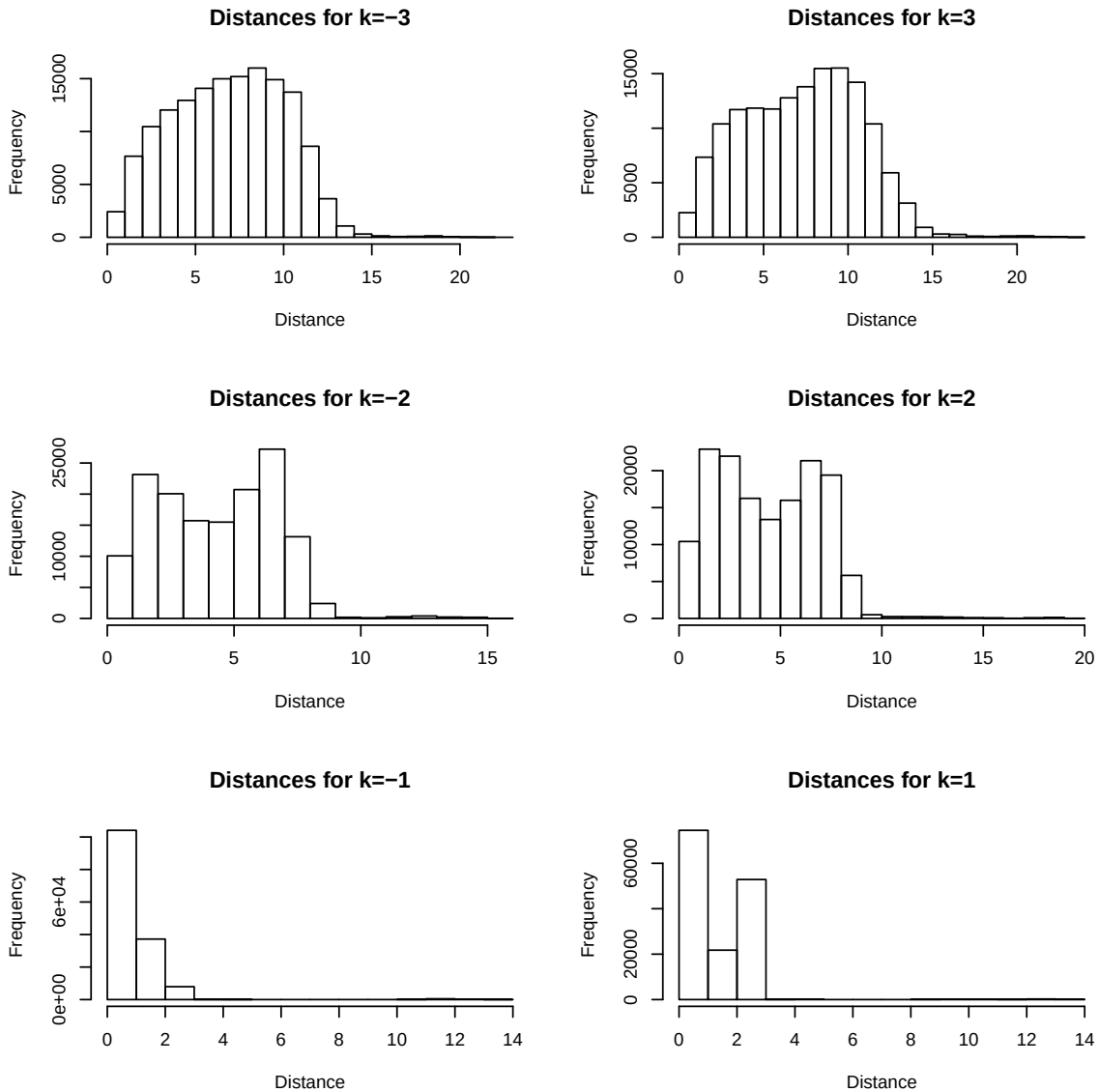
Canonized paths, pairwise distances and virtual atoms



- Pairwise distances: $R_{i+k,j+k}$
- Virtual atoms: given the center of mass of a chain – A or B:
  - To account for the overall shape of the chain
  - For each a.a.: atom place 2Å away from each C_α carbon in the direction of the center of mass

Distribution of pairwise distances

- Distribution of C_α distances in the local frame



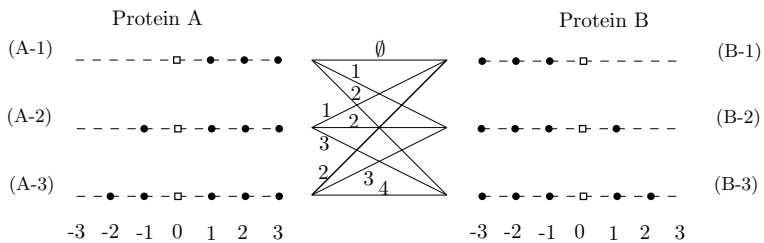
Canonized paths: boundary conditions and weights μ_{ij}

Definition 10. (Multiplicity of a pair) Consider $k \in [-n, -(n-1), \ldots, -1, 1, \ldots, n-1, n]$. The *multiplicity* η_k if

$$\begin{cases} \eta_k = 1 \text{ if } k \in I_A[i] \cap I_B[j] \text{ and } -k \in I_A[i] \cap I_B[j] \\ \eta_k = 0 \text{ otherwise.} \end{cases} \tag{44}$$

The *multiplicity* of the pair of a.a. (i, j) is defined as $\mu_{ij} = \frac{2n}{\sum_k \eta_k}$.

▷ Intersection of supports, illustration for $n = 3$



- Each (A-*) and (B-*) entry: C_α: empty square: its neighborhood: black dots
- Weights on edges of the bipartite graph: $|I_A[i] \cap I_B[j]|$
- Exple: $(A-1) \times (B-3)$: residues of indices 1 and 2 are common; since two common residues are available, the associated multiplicity is $\mu_{ji} = 6/2 = 3$.

Pose independent score K_{ij}

▷ Three components:

- Local score – based on distances $R_{i+k,j+k}$
- Spatial score – based on distances between virtual atoms
- Sequence similarity score
  - Obtained by DP on sequences, with the proper substitution nmatrix

$$K_{ij} = w_l K_{ij}^{\text{Loc.}} + w_s K_{ij}^{\text{Spa.}} + w_b K_{ij}^{\text{Blo.}}, \text{ with } w_l + w_s + w_b = 1. \quad (45)$$

Default values are $w_l = w_s = 0.5; w_b = 0$.

▷ NB: if both proteins have more than 180 residues, the spatial score is turned off (D. Ritchie, personal communication).

Local score $K_{ij}^{\text{Loc.}}$ and spatial score $K_{ij}^{\text{Spa.}}$

Using the coefficients μ_{ij} from cannonized paths (Def. 9), we define:

Definition 11. (Local score) Granted: hyper-parameters $\beta_k (= 1)$ and σ_k.

The *local score* $K_{ij}^{\text{Loc.}}$ of A_i and B_j via their canonized n-paths:

$$K_{ij}^{\text{Loc.}} = \exp\Big(-\mu_{ij} \sum_{k \in I_A[i] \cap I_B[j]} \beta_k R_{i+k,j+k}^2 / (4\sigma_k^2)\Big). \tag{46}$$

▷ Coefficients σ_k: since the larger the index k, the larger the distance $R_{i+k,j+k}$. Stdev of these distances.

▷ Spatial score: local score applied to virtual atoms

Definition 12. (Spatial score) The *spatial score* of two a.a. i (chain A) and j (chain B) is defined from the pairwise distances of their virtual atoms as follows:

$$K_{ij}^{\text{Spa.}} = \exp\Big(-\mu_{ij} \sum_{k \in I_A[i] \cap I_B[j]} \beta_k V_{i+k,j+k}^2 / (4\tau_k^2)\Big). \tag{47}$$

where $V_{i+j,j+k}$ is the distance between the two virtual atoms associated with residues $i + k$ on chain A and $j + k$ on chain B.

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The Kpax flexible aligner

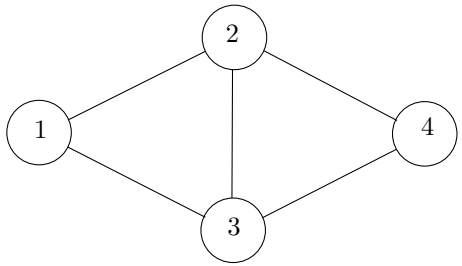
Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

The combined RMSD

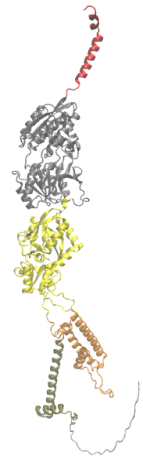
The AcrB efflux pump

Spectral clustering for domain identification: new insights



Laplacian matrix:

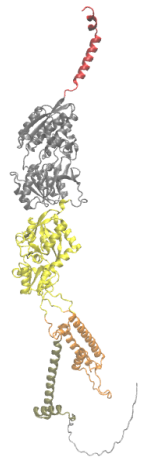
$$L = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}$$



F. Cazals, J. Herrmann, E. Sarti

Proteins, 2025 (in press)

Decomposing a biomolecule into rigid domains



- ▷ Rationale: find a segmentation of the chain / protein such that interactions within domains are dense, while those across domains are not
- ▷ Unsupervised approaches: maximizing intra-domain interactions and minimizing inter-domain interactions
- ▷ Demo. . .

K-means for a fixed value of k

- Rationale: to form homogeneous clusters, minimize $\phi \equiv$ the sum of squared distances to the center of masses of the clusters
- Hardness: NP-hard, but provides the randomized ++ initialization provides is such that

$$\mathbb{E}\left[\phi\right]/\phi_{OPT} \leq 8(\log k + 2) \tag{48}$$

- Cluster stability assessment: for a data point, compare the ratio of distances to the first and second nearest neighbors

$$\bar{r}^{12}_{k,n} =< d_{NN1}/d_{NN2} > \tag{49}$$

▷Ref: k-means++, Arthur and Vassilvitskii, ACM SODA 2007

▷Ref: SPECTRUS, Ponzoni et al, Structure, 2015

Graphs clustering and cuts

- Input: a weighted graph $G = (V, E)$, with symmetric weights $w_{ij} \geq 0$
  - Generalized degree of node i: $d_i = \sum_{j \neq i} w_{ij}$
  - (Sub-)Graph volume for $A \subset V$: $\sum_{i \in A} d_i$
- Goal: partition G into disjoint sets $A_1, \ldots, A_k$

$$\text{cut}(A, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A, \overline{A_i}) \text{ with } W(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{50}$$

- Avoiding trivial cuts via the minimization of:
  - Ratio cut: uses the size of clusters

$$\text{RatioCut}[A, \ldots, A_k] = \sum_i \frac{W(A_i, \overline{A_i})}{|A_i|}. \tag{51}$$

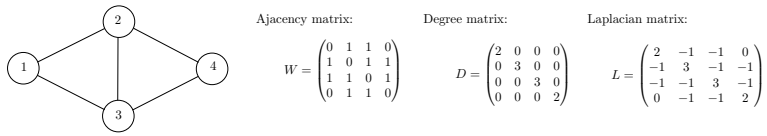
  - Normalized cut: uses the volume (sum of degrees) of clusters

$$\text{NCut}[A, \ldots, A_k] = \sum_i \frac{W(A, \overline{A_i})}{vol A_i}. \tag{52}$$

- Hardness: NP-hard problems ... for which efficient relaxations exist
- Ref: von Luxburg, Stat. Computing, 2007

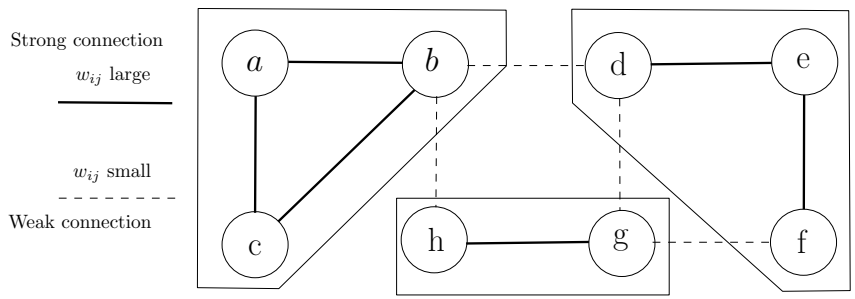
(Standard) Graph Laplacian: example and intuition

- Laplacian $L = D - W$



NB: The unit vector 1_4 is an eingenvector associated with the eigenvalue 0.

- Intuition



Graph Laplacians: std, normalized, random walk

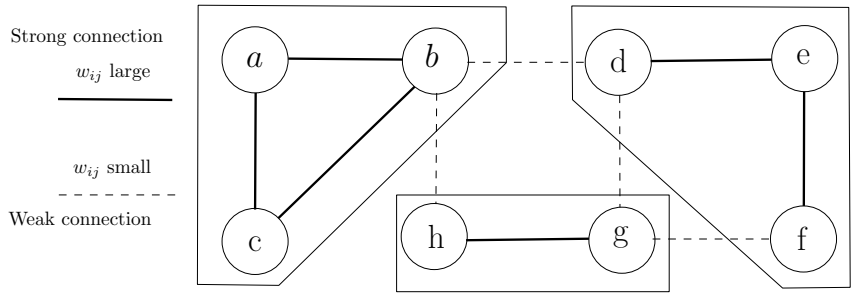
- Def:
  - unnormalized graph Laplacian $L = D - W$
  - normalized Laplacian $L_{\text{sym.}} = D^{-1/2} L D^{-1/2}$
  - Random walk Laplacian: $L_{\text{rw}} = D^{-1} L = I - P$–with P the random walk transition matrix (NB: G can be directed)
- Several important properties of L:
  - L yields a positive semi-definite quadratic form

$$\text{For a vector } f \in \mathbb{R}^d : f^{\mathsf{T}} L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2. \tag{53}$$

  - Eigenvalues are $\lambda_n \geq \ldots \lambda_1 = 0$;
  - The unit vector 1_n is an eigenvector of $\lambda_1 = 0$
  - Eigenspace of the (multiple) eigenvalue 0: indicator vectors $1_{A_1}, \ldots, 1_{A_k}$ for the connected components of G
- NB:
  - similar properties for $L_{\text{sym.}}$ and L_{rw}
  - L_{rw}: Perron-Frobenius and the Google page rank

▷Ref: von Luxburg, Stat. Computing, 2007

Spectral clustering using $L_{\text{sym.}}$



▷ Input:

- similarity matrix $\in \mathbb{R}^{n \times n}$
- k the num. of clusters

▷ Algorithm:

- Compute the Laplacian $L = D - W$
- Compute the normalized Laplacian $L_{\text{sym.}} = D^{-1/2} L D^{-1/2}$
- Compute the first (smallest) k eigenvectors of $L_{\text{sym.}}$
- Form $U \in \mathbb{R}^{n \times k}$ by truncating the eigenvectors to k columns
- Form matrix T by normalizing the rows of U
  NB: **ideally:** these are the indicator vectors of the clusters.
- Consider the n point $T[i :]$: points on the unit sphere S^k
- Perform k-means for these points

▷`Ref: Ng et al, NIPS, 2002`

Graph cuts, graph clustering, Laplacians

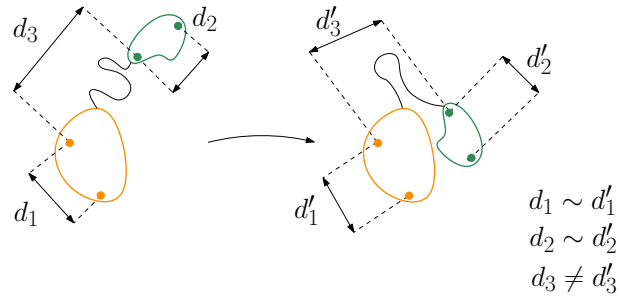
▷ Key points:

- graph cuts yield NP-hard discrete optimization problems
- spectral clustering corresponds to **relaxations** of these problems
- correspondences
  - RatioCut: L
  - NCut: $L_{\text{sym.}}$ or L_{rw}

▷`Ref: von Luxburg, Stat. Computing, 2007`

SPECTRUS: from atomic fluctuations to similarities

- Rationale to identify rigid domains:
  - split protein with spectral clustering
  - weight σ_{ij} (or w_{ij}): function of the variation of relative distances



- Distance fluctuation for each pair of a.a.:
  - option 1: from crystal structures

$$f_{ij} = \sqrt{< d_{ij}^2 > - < d_{ij} >^2}. \tag{54}$$

  - option 2: from theoretical model – atomic **normal modes**
- Similarity matrix/weight matrix from fluctuations–with suitable σ:

$$\sigma_{ij} = \exp(-f_{ij}/2\sigma^2), \tag{55}$$

- Rmk: distance threshold used to consider only pairs within $r_c = 10$Å.

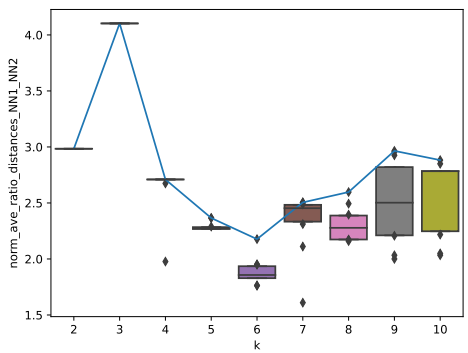
- `Ref: SPECTRUS, Ponzoni et al, Structure, 2015`

SPECTRUS: spectral clustering, k-means, renormalization

- ▷ Spectral clustering: reduces to k-means on the sphere S^k.
- ▷ SPECTRUS score for one run of `k-means++` at fixed value of k:
  - $\overline{r}^{12}_{k,n}$: average ratio d_{NN1}/d_{NN2} obtained from `k-means++`
  - $\overline{r}^{12}_{k,n}$[Ref]: average average such ratio for n random points on S^k
  - Associated SPECTRUS score:

$$\text{Score}_{k,n} = \frac{\overline{r}^{12}_{k,n}}{\overline{r}^{12}_{k,n}[\text{Ref}]}. \tag{56}$$

- ▷ Repeats: best $\text{Score}_{k,n}$ out of $N_c = 10$ repeats for a given k.

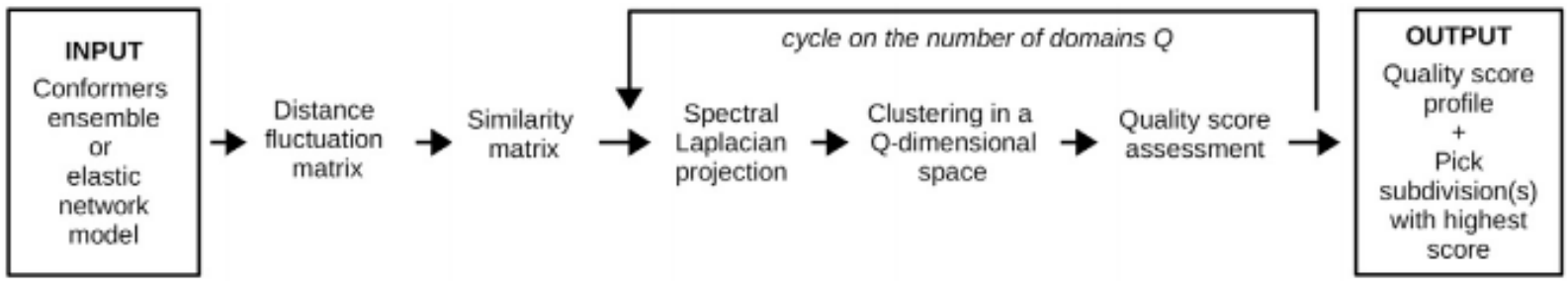


- ▷ Range for k: best k for $k \in [k_{min}, k_{max}]$.

▷Ref: SPECTRUS, Ponzoni et al, Structure, 2015

SPECTRUS: workflow

- Convert fluctuations into weights
- Apply spectral clustering
- Normalize the `k-means` score for spectral clustering



▷`Ref: SPECTRUS, Ponzoni et al, Structure, 2015`

From SPECTRUS to SPECTRALDOM

▷ Fluctuation / stdev for residues i and j:

$$f_{ij} = \sqrt{< d_{ij}^2 > - < d_{ij} >^2}. \quad (57)$$

▷ SPECTRUS:

- f_{ij} from normal modes
- Convert f_{ij} into weights w_{ij}
- Apply spectral clustering using the Laplacian $L = D - W$

▷Ref: Ponzoni et al, Structure, 2015

▷ In SPECTRALDOM: two options

- **Diffusion Map mode**: w_{ij} from (non-)covalent contacts

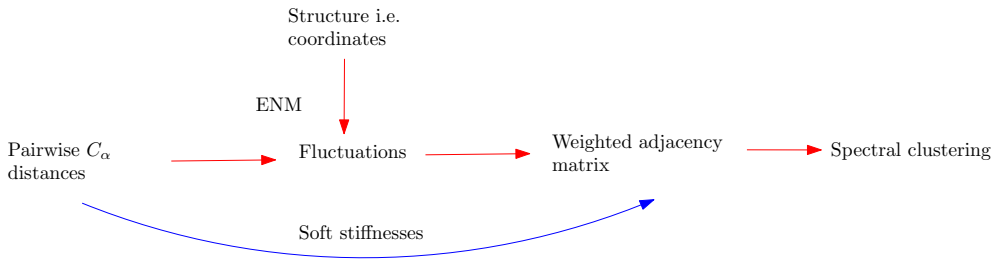
$$w_{ij} = \gamma_{ij} * exp(-\frac{\ln 2}{2} \mid d_{ij} - d_0 \mid).$$

- **Multiple Sequence Alignment mode**: from atomic positions

▷Ref: Cazals et al, Proteins, 2025

SPECTRUS to SPECTRALDOM– details

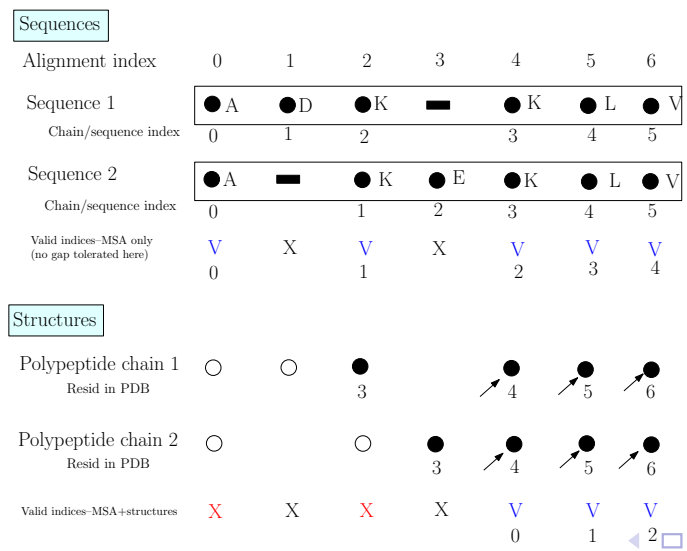
▷ A diffusion map mode: "*For these or even larger macromolecular assemblies, which may be too onerous to simulate with atomistic MD, we further show that the distance fluctuation matrix can be viably obtained from computationally effective elastic network models using only a single reference structure as input.*"



- No more normal modes useless: local vibrations encoded by contacts

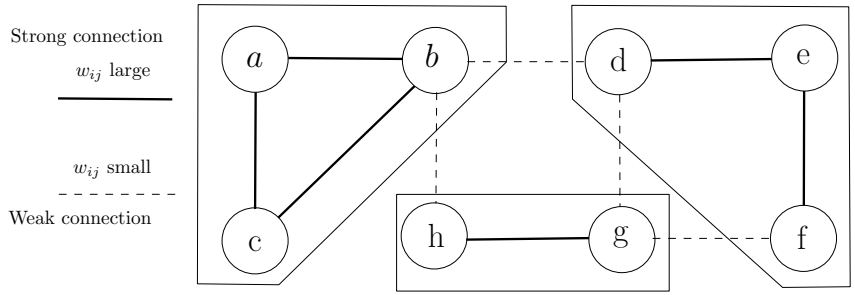
▷ A Multiple Sequence Alignment mode: homologous proteins

"~~*SPECTRUS takes as input multiple structures with no mis- matched sets of missing residues or a single conformation when used in the elastic network mode. . .*~~"



SPECTRALDOM: Diffusion Map mode

▷ Intuition: identify domains from *strong* connexions



▷ Def: non covalent C_α neighbors: two non consecutive a.a. having heavy atoms within distance threshold $r_c = (10\text{Å})$

▷ Consider the following stiffness constants–as in the harmonic model:

- covalent interactions: $\gamma_{ij}^{\text{Cov}}(= 10)$
- non covalent interactions: resp. $\gamma_{ij}^{\text{NCov}}(= 1)$
- and possibly: hydrogen bonds: γ_{ij}^{HB}; salt bridges: γ_{ij}^{SB}, etc.

▷ Direct def. of the similarity / weighted adjacency matrix:

$$w_{ij} = \gamma_{ij} * exp(-\frac{\ln 2}{2} \mid d_{ij} - d_0 \mid). \tag{58}$$

with $d_0 = 5\text{Å}$ and $\sigma = 1$.

Chainsaw– combining RNN and community detection

▷ Stochastic block models for community detection: probability to obtain the graph of domain connectivity

$$\mathbb{P}[A] = \Pi_{i<j} \hat{a}_{ij}^{a_{ij}} (1 - \hat{a}_{ij})^{(1-a_{ij})}. \tag{59}$$

▷ Chainsaw: main steps

- Residual convolutional NN yields C_α adjacency matrix,
- Structure is converted into five feature channels (pairwise C_α distances, and four channels for predicted SSE (helix vs strand, within vs boundary of SSE))
- Features converted into a pairwise probability matrix using a deep convolutional network
  NB: learning minimized the cross entropy between the predicted soft adjacency matrix and the one representing the probability of residue co-occurrence in the same domain.
- Optimization problem solved

$$D^* = \arg\max_{\{v_k\}} \mathbb{P}[A]. \tag{60}$$

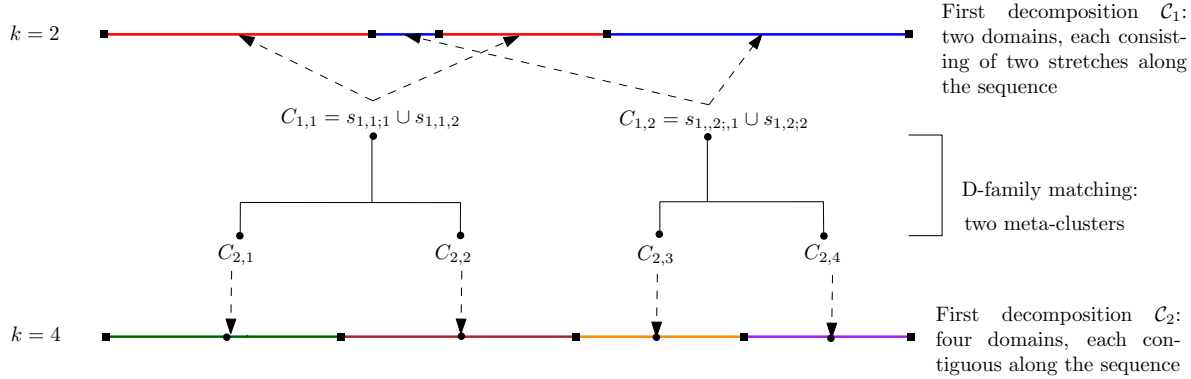
▷ Claim: superior results over unsupervised methods

▷Ref: Chainsaw, Orengo et al, Bioinformatics, 2024

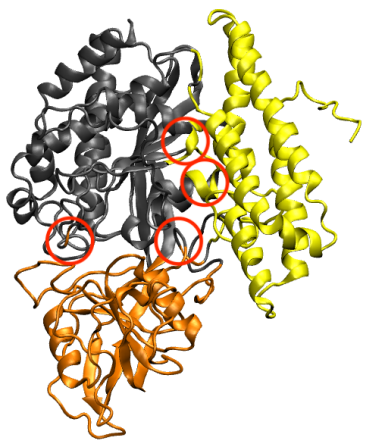
Reducing fragmentation

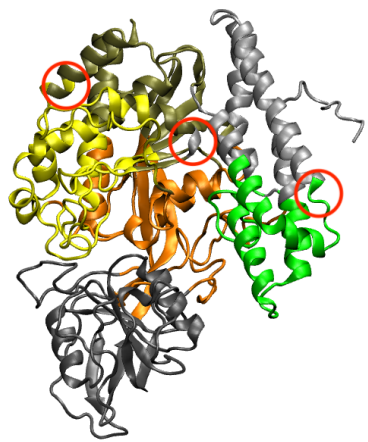
with the D-family matching algorithm

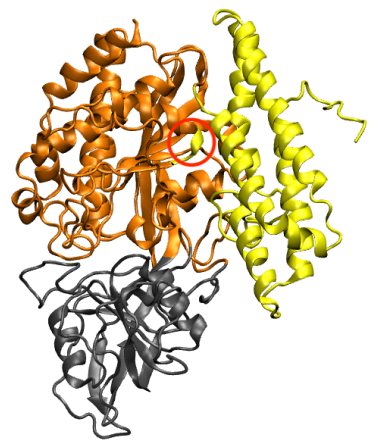
- Exple: chain split into 2 non contiguous domains for $k = 2$



- Fixing the fragmentation by combining two decompositions via D-Fam. maching:

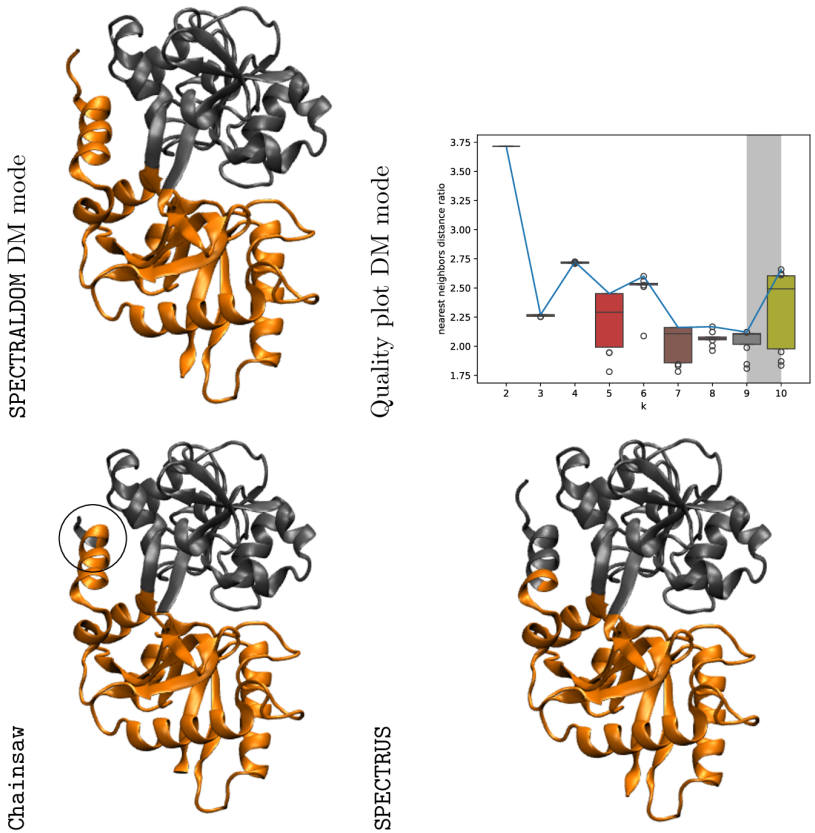






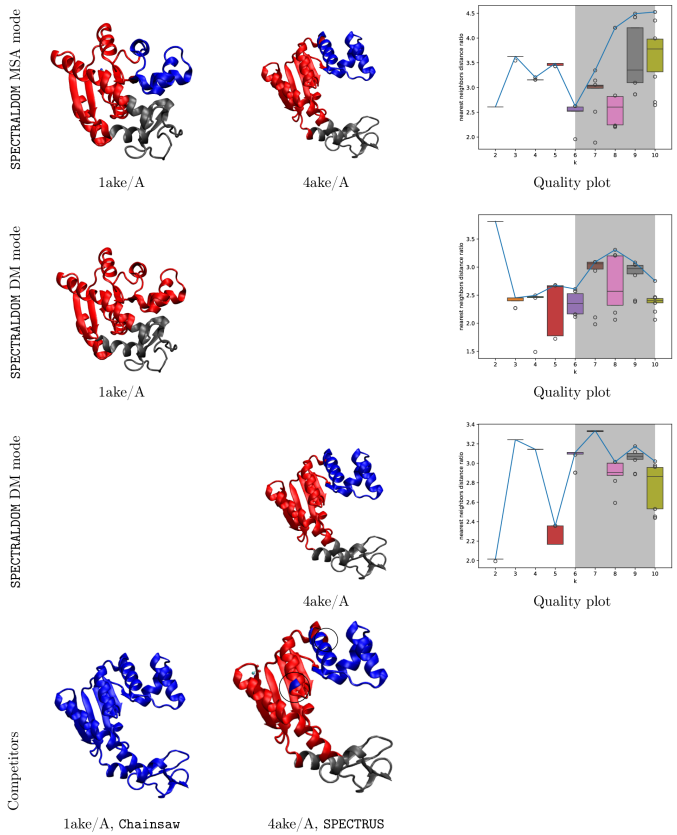
Human serum transferrin: Chainsaw vs SPECTRUS vs SPECTRALDOM

- Two domains easily found
- Chainsaw and SPECTRUS: difficulties in terminal helix
- NB: PDBid 1a8e/A



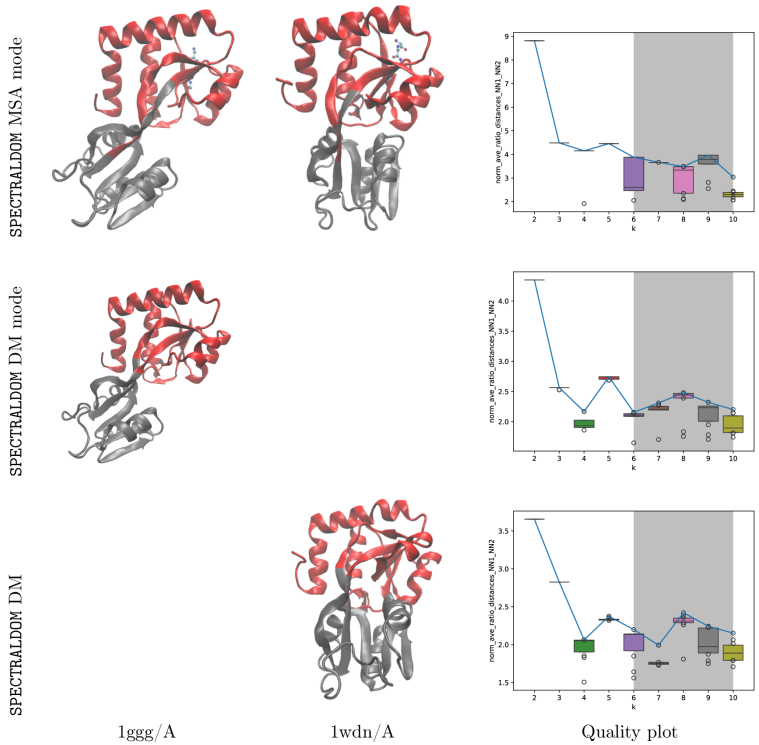
Escherichia coli adenylate kinase: SPECTRALDOM MSA and DM

- (Row 1) MSA mode applied to chains A of 1ake and 4ake.
- (Row 2) DM mode for 1ake/A.
- (Row 3) DM mode for 4ake/A.
- (Row 4) Comparison with SPECTRUS and Chainsaw
- NB: crystal structures: closed: 1ake/A; open: 4ake/A.



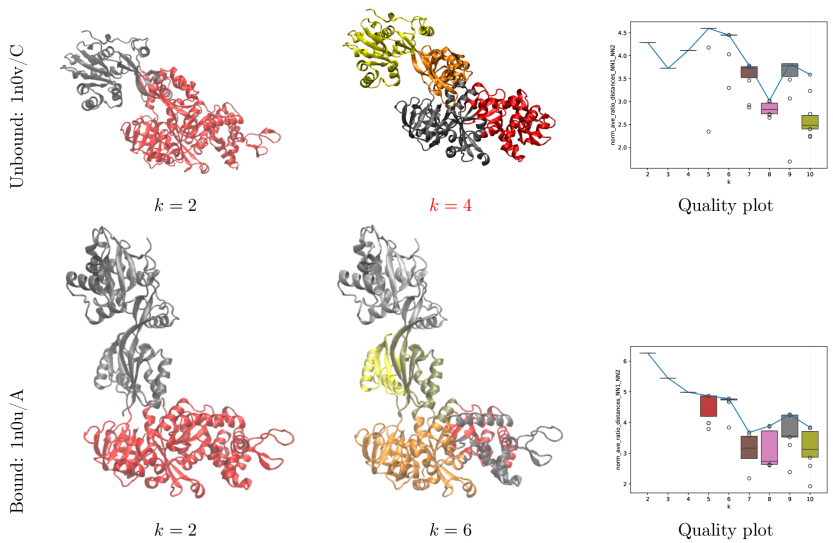
Glutamine-binding protein – GlnBP

- **(Top)** MSA mode applied to chains A of 1ggg and 1wdn.
- **(Middle)** DM mode for 1ggg/A.
- **(Bottom)** DM mode for 1wdn/A.
- NB:Crystal structures: unbound: 1ggg/A; bound: 1wdn/A.



Yeast elongation factor eEF2 : SPECTRALDOM MSA mode

- Values of k of interest: $k = 2, 4, 6$ – local maxima with low score variance.
- **(Top)** Reference domains (crystallography) for the unbound (1n0v) and bound structures (1n0u)
- **(Middle)** Domains of the unbound structure
- **(Bottom)** Domains of the bound structure
- NB: crystal structures: unbound: 1n0v/C; bound: 1n0u/A



Outlook

- SPECTRALDOM:
  - distances suffices (no coordinates) to identify domains
  - a simple and interpretable model
- Software in the Structural Bioinformatics Library

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The Kpax flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

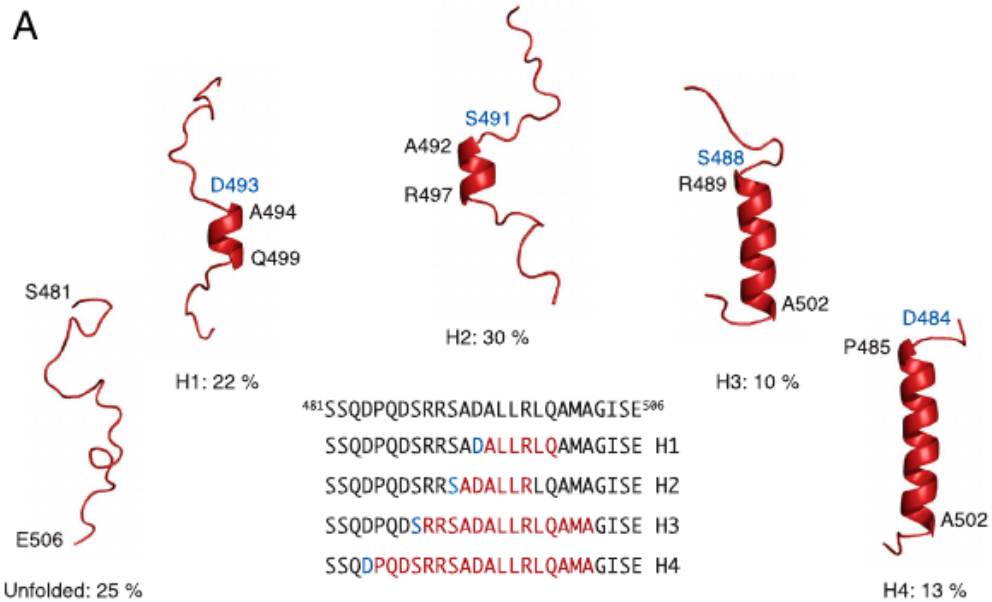
The combined RMSD

The AcrB efflux pump

Structural comparisons: beyond global comparisons

accounting for local features

▷ IDP: molecular recog. element (MoRE) ⊂ C ter domain of the measles nucleoprotein (N_{tail})



▷Ref: M. Blackledge et all, PNAS, 2011

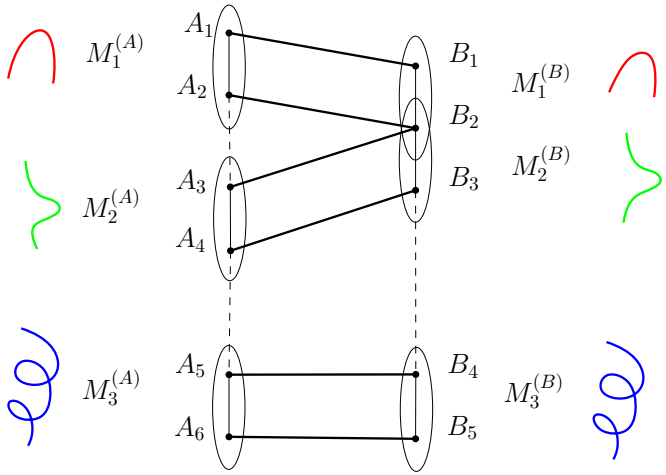
▷ Globular protein: class II fusion proteins before/after fusion

lRMSD	$RMSD_{Comb.}$
$11.32\mathring{A}$	$0.55\mathring{A}$

▷Ref: F. Rey et al, Cell, 2014

Comparing two molecules: the combined RMSD

▷ Rationale: use one rigid motion for each *rigid/structurally conserved* region



Given two molecules A and B:

- Identify *rigid* motifs (`SPECTRUS`, `Kpax`, etc)
- Define the *motif intersection graph*

Definition 13. Consider two structures A and B for which non-overlapping domains $\{C_i^{(A)}, C_i^{(B)}\}_{i=1,\dots,m}$ have been identified. Assume that a lRMSD has been computed for each pair $(C_i^{(A)}, C_i^{(B)})$. Let w_i be the weights associated with an individual lRMSD . The ***combined RMSD*** is defined by

$$\mathrm{RMSD}_{\mathrm{Comb.}}(A,B) = \sqrt{\sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} \mathrm{lRMSD}^2(C_i^{(A)}, C_i^{(B)})}. \tag{61}$$

▷ Rmk: comes into two guises, namely vertex weighted and edge weighted

▷`Ref: Cazals and Tetley, Proteins, 2019`

Upper and lower bounds

▷ Convexity inequalities:

Lemma 14. The *combined RMSD* satisfies the following upper and lower bounds:

$$\mathrm{RMSD}_{\mathrm{Comb.}}(A,B) \geq \sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} \mathrm{IRMSD}(C_i^{(A)}, C_i^{(B)}). \tag{62}$$

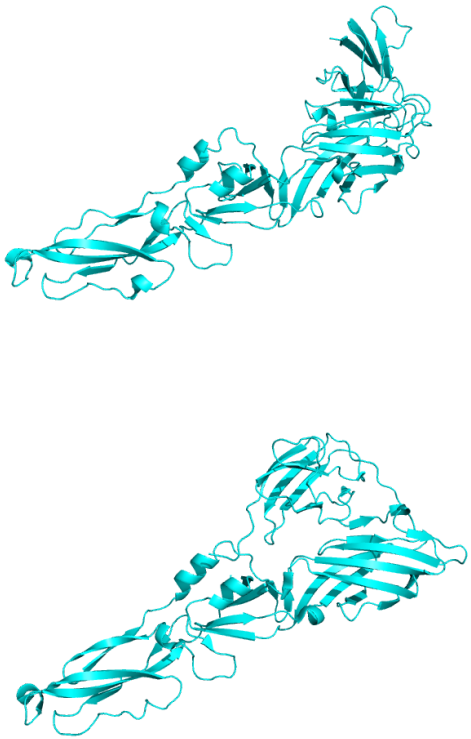
Let $l_{\mathrm{min}} = \min_i \mathrm{IRMSD}(C_i^{(A)}, C_i^{(B)})$ and $l_{\mathrm{max}} = \max_i \mathrm{IRMSD}(C_i^{(A)}, C_i^{(B)})$. One has

$$\mathrm{RMSD}_{\mathrm{Comb.}}(A,B) \leq \sum_{i=1}^{m} \frac{w_i}{\sum_i w_i} \mathrm{IRMSD}(C_i^{(A)}, C_i^{(B)}) + 2\left(\sqrt{\frac{l_{\mathrm{min}} + l_{\mathrm{max}}}{2}} - \frac{\sqrt{l_{\mathrm{min}}} + \sqrt{l_{\mathrm{max}}}}{2}\right). \tag{63}$$

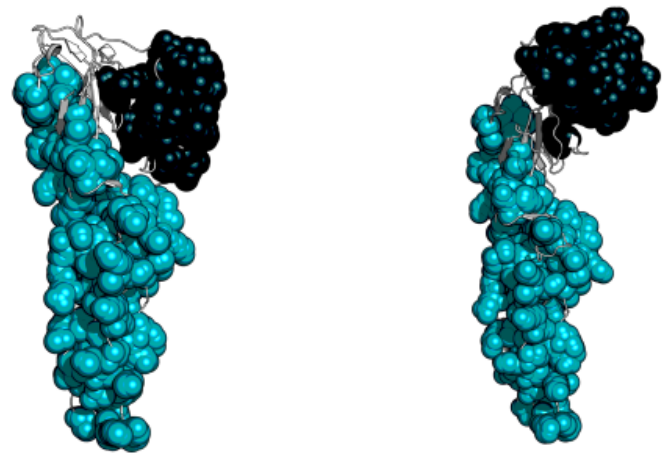
▷Ref: `Cazals and Tetley, Proteins, 2019`

Combined RMSD : TBEV glycoprotein in two different conformations pre and post fusion

▷ Classical analysis:



▷ Our motifs:



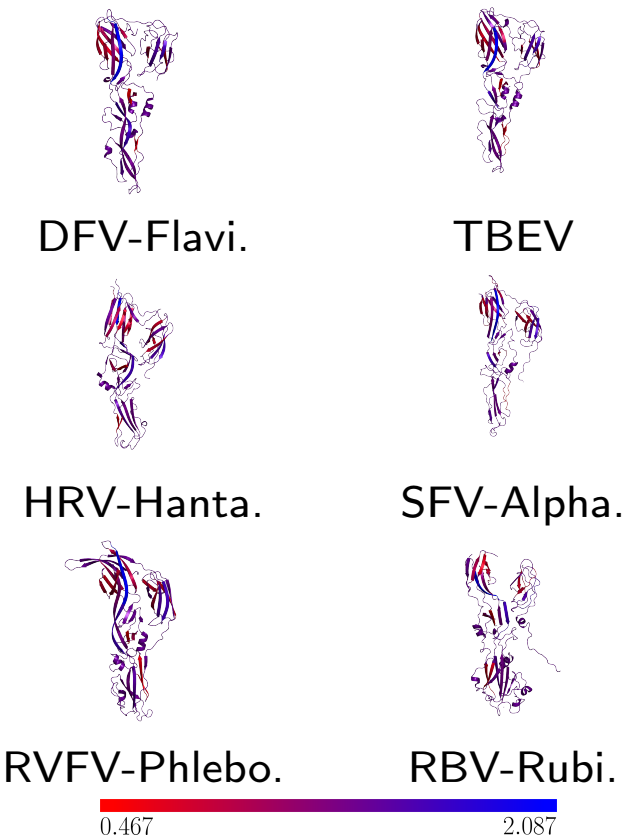
Motif	Alignment size	lRMSD
Large	88	1.69
Small	40	0.38

Statistics from Apurva:

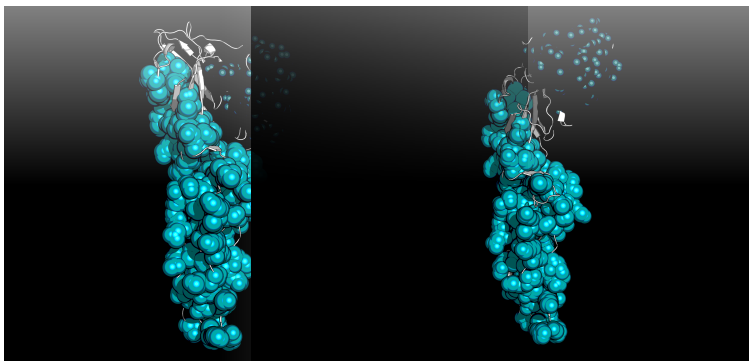
- 370 a.a. aligned
- lRMSD: 11.1Å

The lRMSD smoothes out local structural conservation

▷ Class II fusion proteins: SSE conservation



▷ Class II fusion protein in soluble and post-fusion conformation



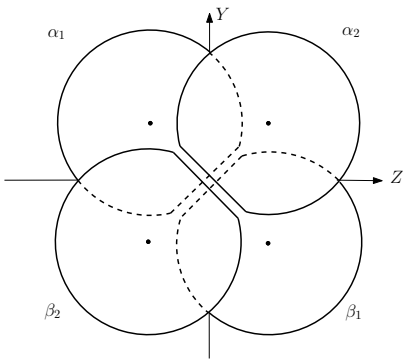
Motif	Alignment size	lRMSD
Large	88	1.69
Small	40	0.38

▷ Motifs and $RMSD_{Comb.}$:

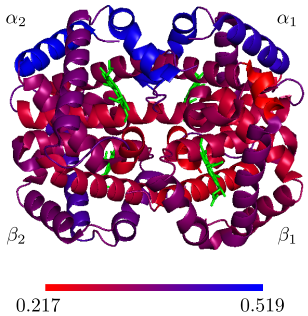
- `Align-Identity-SFD`: #a.a.: 152; $RMSD_{Comb.}$: 2.53 Å.
- `Align-Identity-CD`: #a.a.: 161; $RMSD_{Comb.}$: 1.26 Å.

$\text{RMSD}_{\text{Comb.}}$ identifies novel quaternary structures for hemoglobin

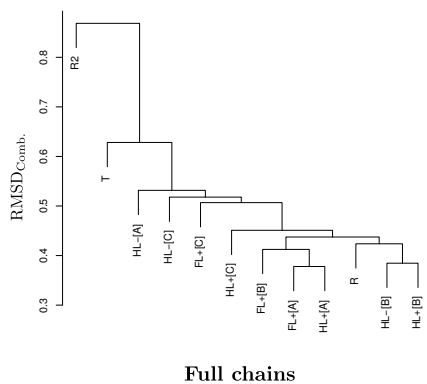
- Quaternary structure of hemoglobin



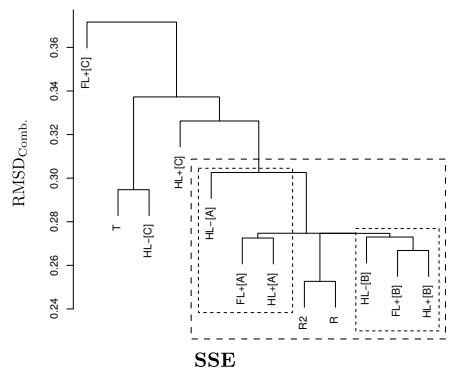
- Conservation of SSE



- $\text{RMSD}_{\text{Comb.}}$ on $\alpha_1\beta_1$ identifies novel quaternary structures



Full chains



SSE

- Ref: Shibayama et al, JACS, 2014
- Ref: Cazals et al, Proteins, 2019

Structural alignments and analysis

Structural alignments and similarities

Geometric prerequisites

Procruste problems and the lRMSD

The Kpax flexible aligner

Domain identification with spectra clustering

- Clustering: pre-requisites
- From SPECTRUS to SPECTRALDOM
- Gallery of results

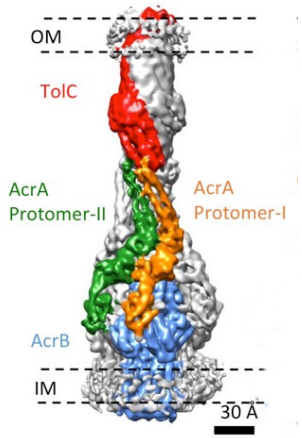
The combined RMSD

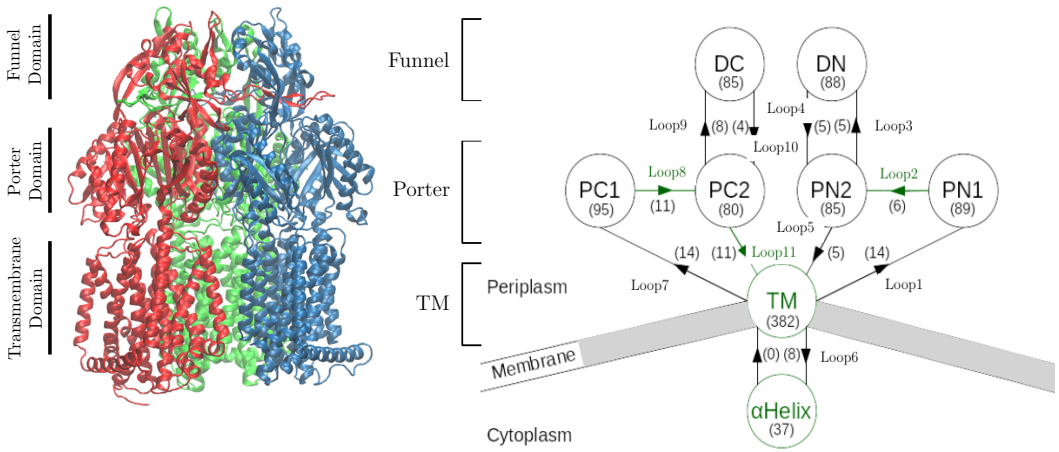
The AcrB efflux pump

RND proteins and drug efflux

▷ Resistance-nodulation-division (RND) proteins: mostly identified in Gram-negative bacteria, bacterial efflux pumps located in the cytoplasmic membrane.
Nb: homologous proteins involved in cancer chemo-resistance.

▷ Particular case: AcrA-AcrB-TolC transporter, with the AcrB trimer





AcrB involves 8 main subdomains and 12 linkers

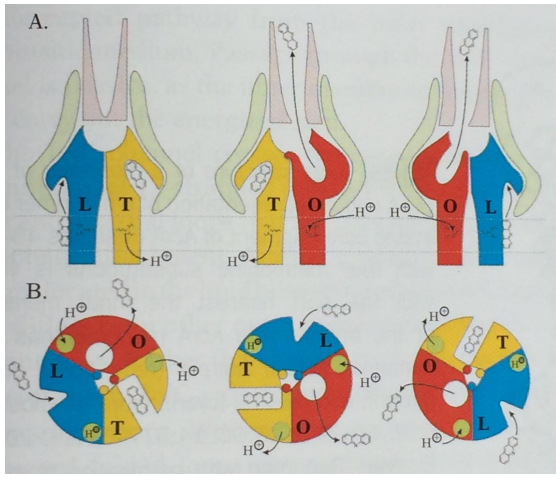
▷ Data available for AcrB : wild-type structures with median resolution 3.32 Å; 81 monomers gathered in from 32 (trimers) + 11 (monomers) PDB files

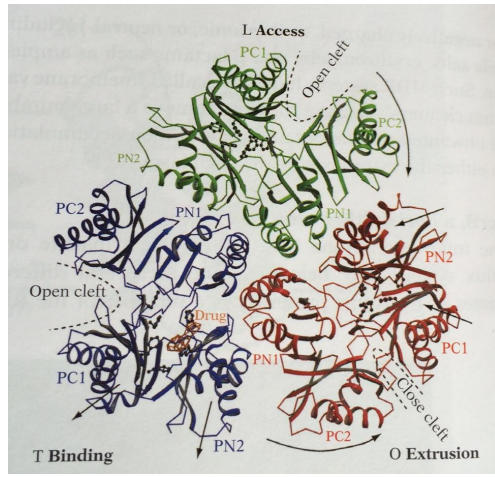
▷Ref: Yamaguchi et al, Frontiers in microbiology, 2015

Export by AcrB: overview of the mechanism

- ▷ Active transport using the proton motive force (PMF): AcrB trades
  - $1H^+$ flowing along the negative gradient – periplasm into the cytoplasm,
  - 1 substrate molecule – cytoplasm/inner membrane to the outside of the cell across AcrB - AcrA - TolC

- ▷ Mechanically: a peristaltic pump





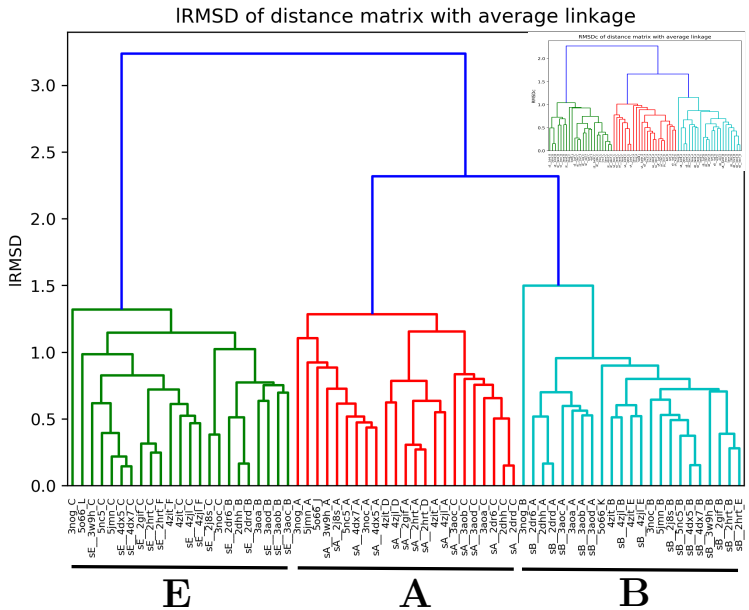
- ▷ Three step rotating mechanism based on 3 states of monomers

– Three states: A/L: access/loose; B/T: bound/tight; E/O: extrusiOn/Open
  A state: vestibule open on periplasm; binding pocket shrunk
  B state: molecule binds into pocket; blocked channel (central helix) opens
  E state: vestibule closed; exit open–central helix rotates
– Allostery: binding to AcrB → repacking AcrA → channel opening of TolC

▷Ref: Seeger et al., Science, 2006; Murakami et al, Nature 2006
▷Ref: Wang et al, eLife, 2017

Question 1: states for monomers and the trimer

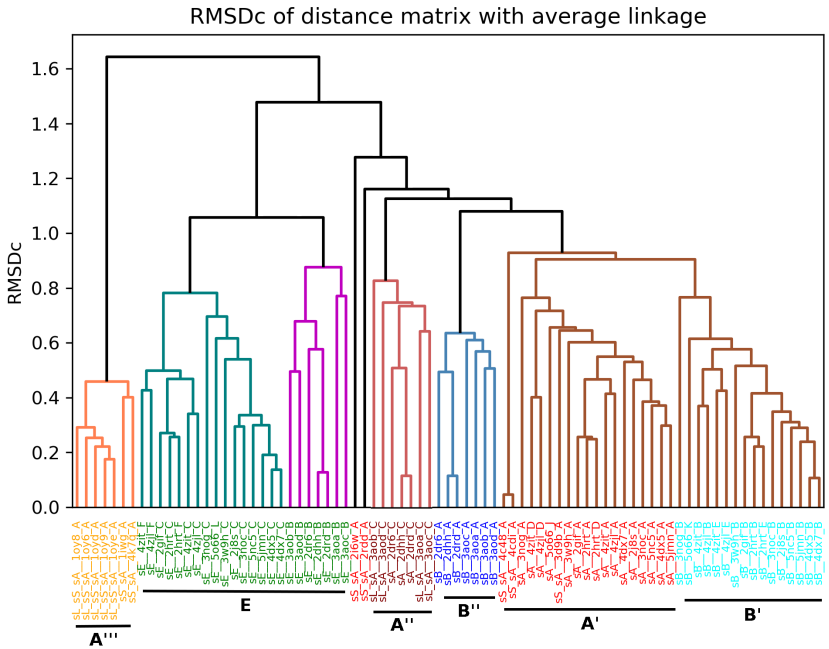
- Questions: are A, B, and E the only states? What about trimers?
- Method: hierarchical clustering of individual monomers

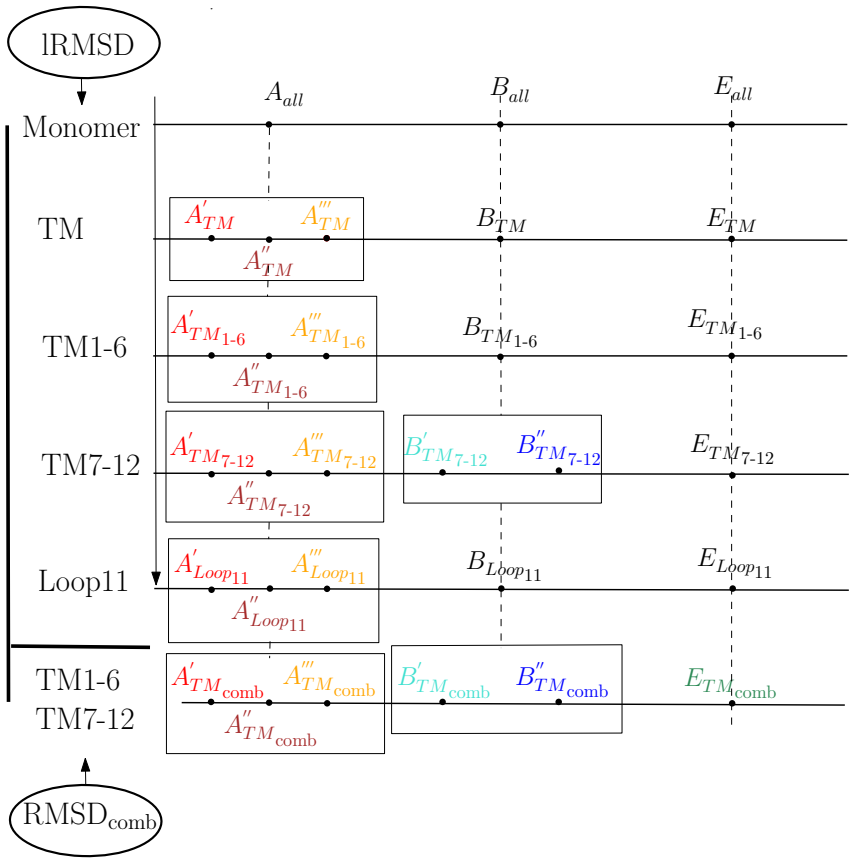


- Findings
  - labeling of unlabeled monomers as A, B or E (from the containing cluster)
  - only observed state for asymetric trimers: ABE

Question 2: sub-domains and sub-states

- Question: which subdomains (out of 8) and linkers (out of 12) account for the A, B and E states?
- Methods: combined lRMSD clustering of sub-domains

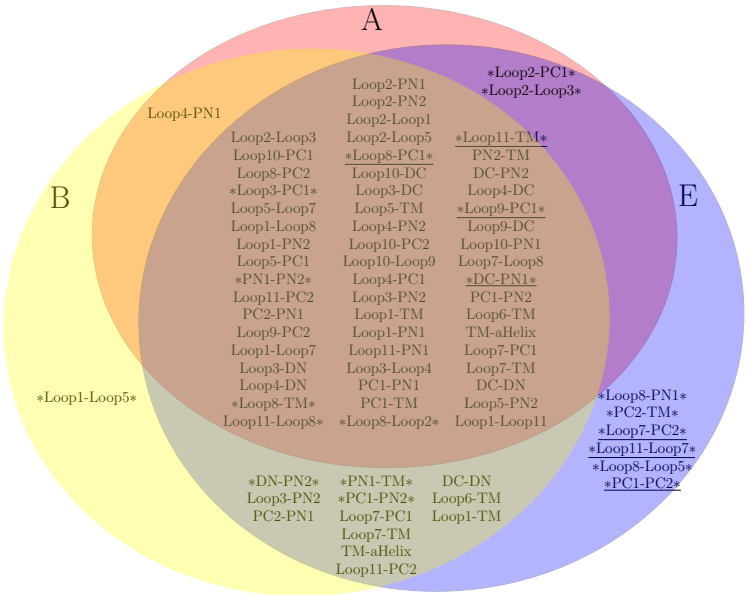




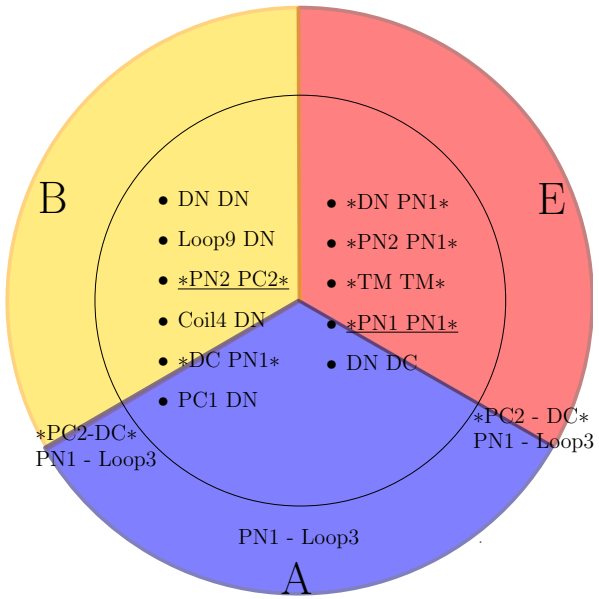
- Findings:
  - Subdomains compatible with the A, B, E clustering: Loop2, Loop8, Loop11, TM
  - Novel substates identified: $A \rightarrow A', A'', A'''; B \rightarrow B', B''$

Question 3: evolution of interfaces between sub-domains

- Question: identify subdomains whose relative positions change
- Method: using *Voronoi* interfaces between sub-domains



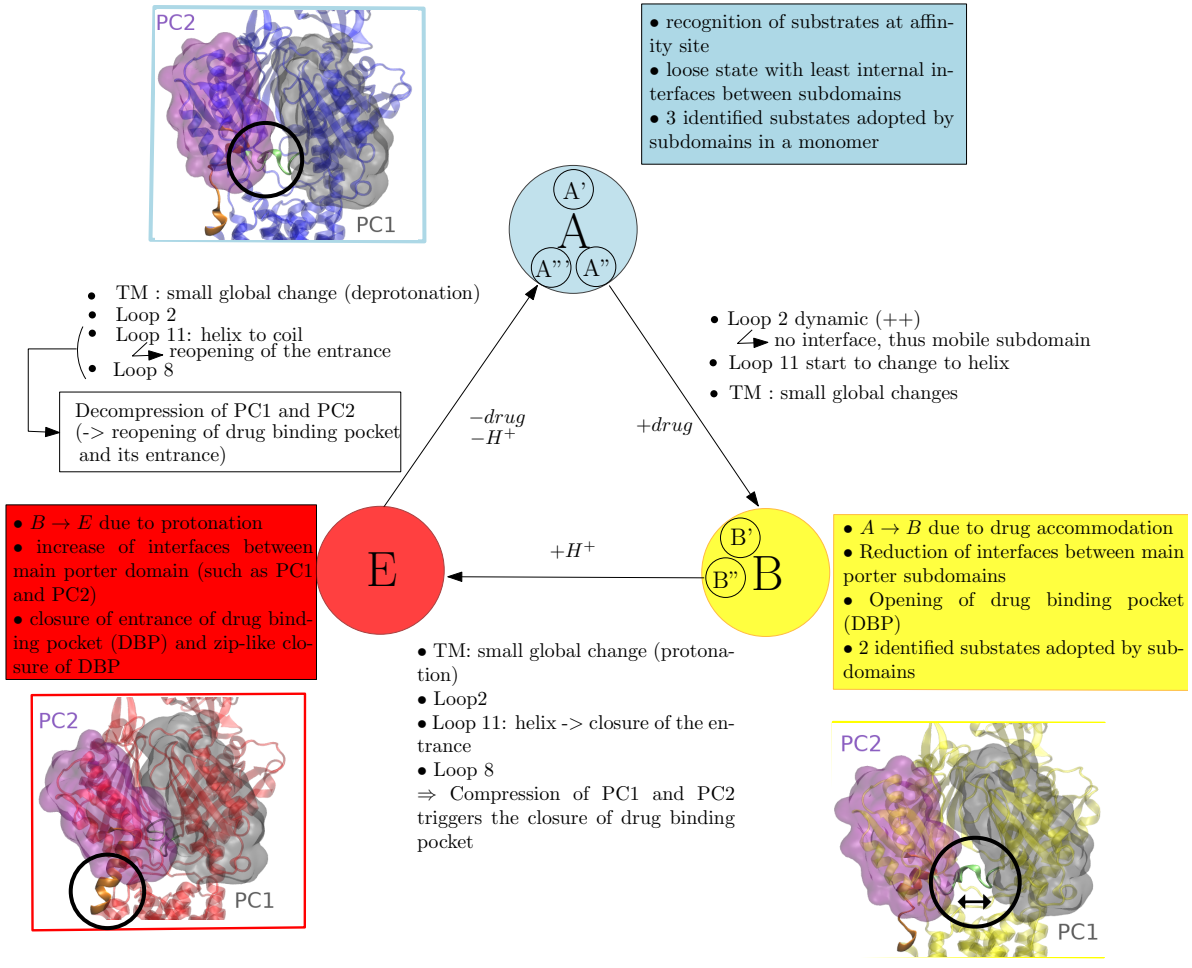
Interfaces: within monomer



Interfaces: across monomers

- Findings:
  - # of interfaces: 57 in A state, 68 in B state and 74 in E state
  - Characterization of interfaces specific to individual states

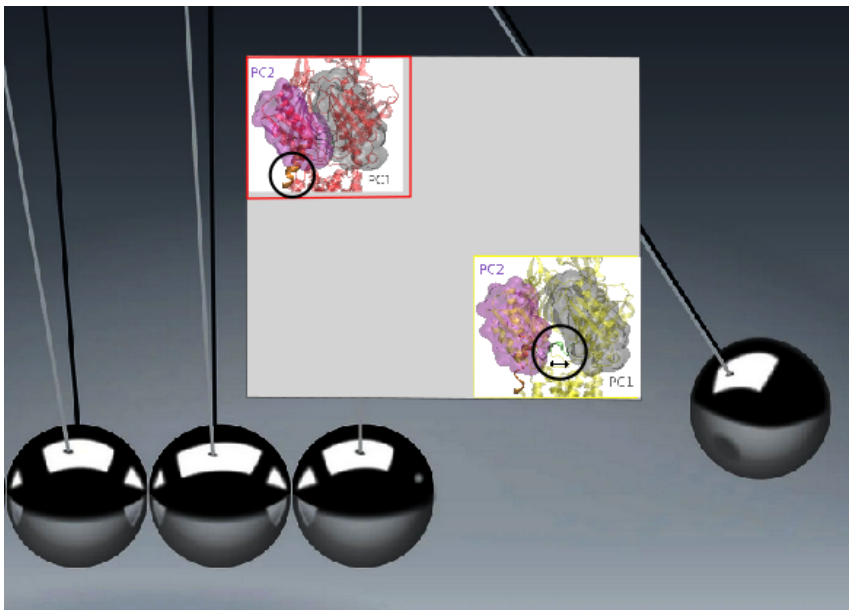
Better understanding of the AcrB cycle



▷Ref: Simsir et al, Proteins, 2021

Software: molecular craddle

▷ Rationale: model a complex molecular machine as a craddle of sub-domains whose relative positions change



▷ Package in the Structural Bioinformatics Library:
`https://sbl.inria.fr/doc/Molecular_cradle-user-manual.html`