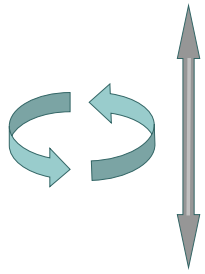
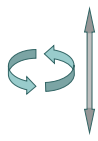


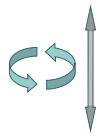
Protein Structure Overlap

Maximizing Protein
Structural Alignment
in 3D Space



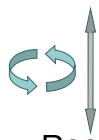
Motivation (1)

- As mentioned several times, we want to know more about protein function by assessing protein structure.
  - Similar structure often implies similar function.
- A frequent concern is whether two proteins have the same or very similar structure.
  - An assessment of this can be done by attempting to superimpose the two proteins in 3D space.
  - The proteins may have the same residues or they may be very similar (homologs, for example).



Motivation (2)

- Recap: A frequent concern is whether two proteins have the same or very similar structure.
- There are various applications:
  - The proteins may have the same sequence but differ in conformation.
    - The difference might be due to a different energy state or may be due to a change in conformation when a ligand is present in a binding site.
  - The proteins may have almost the same sequence; there are minor variations.
    - For example, a mutation has caused some amino acid to change. What is the effect on conformation?



Motivation (3)

- Recap: A frequent concern is whether two proteins have the same or very similar structure.
- Applications (continued):
  - The proteins may have more extensive differences in their sequences but it is possible that they nonetheless show a lot of similarity in conformation.
    - The question is “How similar are these conformations?”.
  - The proteins may have considerable differences in their sequences but it is possible that they share similar structure in various regions.
    - We would want to compare these similar regions.

Motivation (4)

- The possibility of similar structure despite differences in sequence is not surprising since it has been observed that:

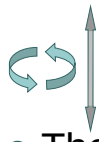
  Structure is more conserved than sequence.

  - Ultimately, it is protein functionality that is most important.
  - Evolutionary processes do not *read* sequences.
    - Evolutionary processes will tend to “observe functionality” (as determined by structure) rather than observing sequence.
    - They essentially use the “duck test” on protein function:
      "If it walks like a duck and quacks like a duck, it must be a duck".

5

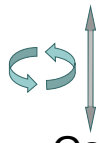
Introduction

- Our strategy in these applications is to do a structural alignment or overlap of the proteins in the 3D space.
- In our case, this will involve modifying the coordinates of atoms so that:
  - One protein is moved (translated) in the space so that the centroids of the two proteins coincide.
  - An optimal rotation is then done to get the maximal amount of overlap.
    - That is, the **maximal structural alignment**.



Structure Alignment (1)

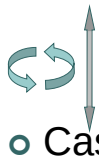
- There are three possible cases to consider based on assumptions about molecular flexibility and sequence similarity:
- Case 1: Same Sequence | Rigid Proteins
  - Find the translation and rotation that minimizes the RMSD of the two proteins.
- Case 2: Different Sequence | Rigid Proteins
  - First find the matching amino acid pairs that are to be brought into 3D alignment.
  - Then translate and rotate to maximally align these amino acid pairs in the 3D space.



Structure Alignment (2)

- Case 3: Different Sequence | Flexible Proteins
  - This is more difficult.
  - We get the matching amino acids as in Case 2.
  - Then try to get the translation and rotation that will give a maximal structure alignment with some conformational changes allowed.
    - We try to limit the amount of conformational change or at least go from one energy minimum to another.
    - Some researchers try to find “hinges” in the more flexible regions of the protein.

The case of same sequence and flexibility is not considered since, trivially, the proteins are simply assumed to be capable of full overlap.



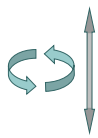
Alignment for Structural Comparison

- Case 1 is often employed to see how the same proteins may have different conformations due to the presence of ligands.
  - Translation and rotation is necessary just to get an alignment of all protein regions outside the binding site.
- Note how three files for HIV protease may have different coordinates for the same atoms:

ATOM	1	N	PRO	A	1	-12.600	38.218	3.719	1IZI
ATOM	2	CA	PRO	A	1	-12.444	38.367	2.244	
ATOM	1	N	PRO	A	1	0.421	40.709	18.682	1MSN
ATOM	2	CA	PRO	A	1	-0.422	39.511	18.905	
ATOM	1	N	PRO	A	1	29.101	40.309	5.484	1EBW
ATOM	2	CA	PRO	A	1	30.105	39.343	4.986	

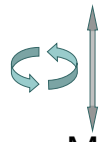
PDB IDs

The PDB does not put proteins in any "standard" orientation.



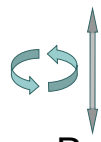
Simple Structural Alignment (1)

- We will consider structural overlap of rigid proteins (dealing with flexibility is much more difficult).
- We will deal with Case 2.
  - Different sequence | Rigid proteins
  - Note that Case 1 is just a *special case* of Case 2.
- Recall the objectives:
  1. First find the matching amino acid pairs that are to be brought into alignment.
  2. Then translate and rotate to maximally align these amino acid pairs.



Simple Structural Alignment (2)

- Matching amino acids
  - The objective is to find a correspondence or matching between 3D features.
    - Not easy if the proteins are not similar.
    - Trivial if we have the same sequence.
  - Strategies for matching:
    1. Use a sequence alignment and so derive the matching.
    2. Specify a matching for particular amino acid pairs when it is determined that their 3D structures should be in structural alignment.
       - May require special biological expertise.
    - Both of these strategies must contend with the issue of how we deal with the amino acids that do *not* match.

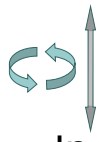


Simple Structural Alignment (3)

- Before discussing how we will use the matching, let us formalize the problem:
  - We will assume that we are trying to overlap the proteins in the 3D space (i.e. superimpose them) by having the $C\alpha$ (alpha carbons) of matching residues overlap as much as possible.
  - We are given two sequences of alpha carbon 3D coordinates:

$$P = \left\{ p^{(i)} \right\}_{i=1}^{|P|} \qquad Q = \left\{ q^{(i)} \right\}_{i=1}^{|Q|}$$

where $|P|$ and $|Q|$ are the number of residues in protein P and protein Q respectively.

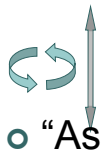


Simple Structural Alignment (4)

- In the Case 1 scenario we would have $|P| = |Q|$. We could then specify the problem as follows:
  - Find a 3D rotation matrix R and a translation vector T such that when R and T operate on all the $C\alpha$ coordinates of P we end up with a new set of alpha carbon coordinates:

$$P_{transformed} = \left\{ Rp^{(i)} + T \right\}_{i=1}^{|P|}$$

    that are as close as possible to the $C\alpha$ coordinates of Q.
    - What do we mean by "as close as possible"?



Simple Structural Alignment (5)

- "As close as possible":
  - Our measure of success will be to minimize the sum of squares of norms that measure the distance between matching alpha carbons.
    - This is minimizing in the *Least Squares* sense:

Find R and T that will minimize E.

$$E(P_{transformed}, Q) = \frac{1}{2} \sum_{i=1}^{|P|} \left\| Rp^{(i)} + T - q^{(i)} \right\|^2.$$

After the minimum E is calculated, we evaluate the Root Mean Square Deviation to see how well we have done:

$$RMSD(P_{transformed}, Q) = \sqrt{\frac{1}{|P|} \sum_{i=1}^{|P|} \left\| Rp^{(i)} + T - q^{(i)} \right\|^2}.$$

Simple Structural Alignment (6)

- Dealing with overlap when $|P| \neq |Q|$.
  - When the proteins have different lengths we will have to decide which pairs of residues contribute to the calculation of E in the previous slide.
    - Recall that this is our Case 2.
- Some definitions:
  - An **equivalence** is a set of pairs

    $$\left\{\left(p^{(\alpha_1)}, q^{(\beta_1)}\right), \left(p^{(\alpha_2)}, q^{(\beta_2)}\right), \ldots, \left(p^{(\alpha_N)}, q^{(\beta_N)}\right)\right\}$$

    indicating the correspondence between the amino acids in P and Q.

Simple Structural Alignment (7)

- Dealing with overlap when $|P| \neq |Q|$ (continued).
  - An **alignment** M for P and Q is an equivalence such that $\alpha_1 < \alpha_2 < \ldots < \alpha_N$ and $\beta_1 < \beta_2 < \ldots < \beta_N$.
  - We extract the alpha carbons from each list in the alignment: $M(P) = \left(p^{(\alpha_1)}, p^{(\alpha_2)}, \ldots, p^{(\alpha_N)}\right)$

    $$M(Q) = \left(q^{(\beta_1)}, q^{(\beta_2)}, \ldots, q^{(\beta_N)}\right)$$
  - These become the alpha carbons that are used in the least squares sum to be minimized.

Simple Structural Alignment (8)

- Dealing with overlap when $|P| \neq |Q|$ (continued).
  - In summary, we do Case 2 by minimizing

$$E(M(P_{transformed}), M(Q)) = \frac{1}{2}\sum_{i=1}^{N}\left\| Rp^{(\alpha_i)} + T - q^{(\beta_i)} \right\|^2.$$

  - Recall that Case 1 is just:

$$\alpha_i = \beta_i = i \qquad \forall i.$$

  - *RMSD* changes:

$$RMSD(P_{transformed}, Q) = \sqrt{\frac{1}{|N|}\sum_{i=1}^{|N|}\left\| Rp^{(\alpha_i)} + T - q^{(\beta_i)} \right\|^2}.$$

17

Deriving the R and T Transforms (1)

- We start by defining the centroids of the alpha carbons used in the superimposition.
  - Let:

$$p^{(c)} = \frac{1}{N}\sum_{i=1}^{N} p^{(\alpha_i)} \qquad q^{(c)} = \frac{1}{N}\sum_{i=1}^{N} q^{(\beta_i)}.$$

Centroid for P

Centroid for Q

  - Then let: $x^{(i)} = p^{(\alpha_i)} - p^{(c)} \qquad y^{(i)} = q^{(\beta_i)} - q^{(c)}.$
  - We will now consider $x^{(i)}$ and $y^{(i)}$ $i = 1, 2, \ldots, N$ to be the coordinates of the matching alpha carbons in proteins P and Q respectively.

Deriving the R and T Transforms (2)

- Why did we let $x^{(i)} = p^{(\alpha_i)} - p^{(c)}$ $\quad y^{(i)} = q^{(\beta_i)} - q^{(c)}$?
  - We have essentially translated the entire protein so that its centroid is at the origin. In other words:
- If we now use these new coordinates in the computation of the centroids we see that they are at the origin.
  - In fact: $\sum_{i=1}^{N} x^{(i)} = \sum_{i=1}^{N} \left(p^{(\alpha_i)} - p^{(c)} \right) = \sum_{i=1}^{N} p^{(\alpha_i)} - \sum_{i=1}^{N} p^{(c)}$
    $= Np^{(c)} - Np^{(c)} = 0.$
  - This is also true for the $y^{(i)}$.
  - In summary: $\boxed{\sum_{i=1}^{N} x^{(i)} = 0}$ and $\boxed{\sum_{i=1}^{N} y^{(i)} = 0.}$

19

Deriving the R and T Transforms (3)

- So, working with our new coordinate system, we see that we want to find R and T to minimize:

$$E = \frac{1}{2} \sum_{i=1}^{N} \left\| Rx^{(i)} + T - y^{(i)} \right\|^2.$$

- Expanding this we get:

$$E = \frac{1}{2} \sum_{i=1}^{N} \left(Rx^{(i)} + T - y^{(i)} \right)^{\mathrm{T}} \left(Rx^{(i)} + T - y^{(i)} \right)$$
$$= \frac{1}{2} \sum_{i=1}^{N} \left(\left(Rx^{(i)} - y^{(i)} \right)^{\mathrm{T}} + T^{\mathrm{T}} \right) \left(\left(Rx^{(i)} - y^{(i)} \right) + T \right)$$
$$= \frac{1}{2} \sum_{i=1}^{N} \left[\left\| Rx^{(i)} - y^{(i)} \right\|^2 + \left(Rx^{(i)} - y^{(i)} \right)^{\mathrm{T}} T + T^{\mathrm{T}} \left(Rx^{(i)} - y^{(i)} \right) + T^{\mathrm{T}} T \right]$$
$$= \frac{1}{2} \sum_{i=1}^{N} \left[\left\| Rx^{(i)} - y^{(i)} \right\|^2 + T^{\mathrm{T}} T \right].$$

Because of the previous slide (centroids at origin) both these terms become zero in the sum!

Deriving the R and T Transforms (4)

- Under these conditions (centroids at origin) we have just seen that:

$$E = \frac{1}{2}\sum_{i=1}^{N}\left[\left\|Rx^{(i)} - y^{(i)}\right\|^2 + \|T\|^2\right].$$

- If we wish to get a minimum E it is clear that we want $T = 0$ since this will zero out the $\|T\|^2$.
  - Note that we could also get this result by computing $\frac{\partial E}{\partial T} = T$ and setting this to zero.

Now we see why centroids at the origin are beneficial!

- So, finally, we see that we want to find the rotation matrix R that will minimize: $E = \frac{1}{2}\sum_{i=1}^{N}\left\|Rx^{(i)} - y^{(i)}\right\|^2.$

Rotation Matrices

- To find the 3D rotation matrix R that will minimize

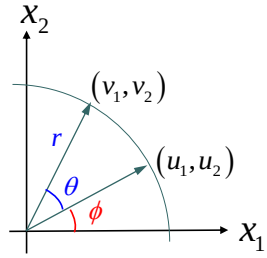
$$E = \frac{1}{2}\sum_{i=1}^{N}\left\|Rx^{(i)} - y^{(i)}\right\|^2$$

we need to know more about rotation matrices.

- Over the next few slides we discover the important attributes of such matrices.

A Rotation Matrix in 2D (1)

- We now derive a matrix that transforms a point (u_1, u_2) in the (x_1, x_2) plane so that its vector is rotated by an angle of θ.
  - The final position of (u_1, u_2) after rotation will be (v_1, v_2).
  - We assume that the distance of (u_1, u_2) from the origin is r. With these assumptions we can write:



$$u_1 = r\cos\phi \qquad v_1 = r(\cos\phi\cos\theta - \sin\phi\sin\theta)$$
$$u_2 = r\sin\phi \qquad v_2 = r(\cos\phi\sin\theta + \sin\phi\cos\theta)$$

$$v_1 = r\cos(\phi+\theta) \qquad v_1 = u_1\cos\theta - u_2\sin\theta$$
$$v_2 = r\sin(\phi+\theta) \qquad v_2 = u_1\sin\theta + u_2\cos\theta.$$

A Rotation Matrix in 2D (2)

- These last equations:

$$v_1 = u_1\cos\theta - u_2\sin\theta$$
$$v_2 = u_1\sin\theta + u_2\cos\theta$$

  can be written in matrix form as:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = R_\theta \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

- Note that $R_\theta^{\mathrm{T}} R_\theta = I.$
  - In fact, the columns of this rotation matrix are **orthonormal**:

$$C^{(i)} \bullet C^{(i)} = \sin^2\theta + \cos^2\theta = 1 \quad i = 1,2$$
$$C^{(i)} \bullet C^{(j)} = -\cos\theta\sin\theta + \cos\theta\sin\theta = 0 \quad i \neq j.$$

A Rotation Matrix in 2D (3)

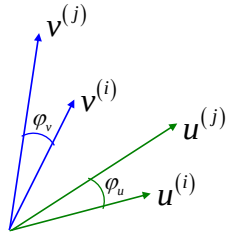
- The equation $R_\theta^{\mathrm{T}} R_\theta = I$ is an important property of the rotation matrix.
  - Consider the norm $\|u\|$ of a vector u in the (x_1, x_2) plane.
    - When a rotation transformation is applied to u we get $v = R_\theta u$.
    - Calculating the norm of v:

$$\|v\|^2 = v^{\mathrm{T}} v = \left(R_\theta u\right)^{\mathrm{T}} R_\theta u = u^{\mathrm{T}} R_\theta^{\mathrm{T}} R_\theta u = u^{\mathrm{T}} u = \|u\|^2 .$$

    - So, the property $R_\theta^{\mathrm{T}} R_\theta = I$ means that the transformation matrix does not change the length of a vector.
      - Length is "invariant" under this transformation.

A Rotation Matrix in 2D (4)

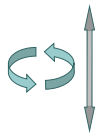
- Angles between vectors are preserved when the transformation R_θ is applied to both vectors:
  - Suppose we have two vectors $u^{(i)}$ and $u^{(j)}$ that are both subject to the transformation R_θ .
    - After the transformation we get $v^{(i)} = R_\theta u^{(i)} \quad v^{(j)} = R_\theta u^{(j)}$.
    - We compare the angle between $u^{(i)}$ and $u^{(j)}$ with the angle between the transform images $v^{(i)}$ and $v^{(j)}$:



$$\cos\varphi_v = \frac{v^{(i)\mathrm{T}} v^{(j)}}{\left\|v^{(i)}\right\| \left\|v^{(j)}\right\|} = \frac{u^{(i)\mathrm{T}} R_\theta^{\mathrm{T}} R_\theta u^{(j)}}{\left\|v^{(i)}\right\| \left\|v^{(j)}\right\|}$$

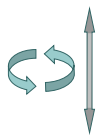
$$= \frac{u^{(i)\mathrm{T}} u^{(j)}}{\left\|u^{(i)}\right\| \left\|u^{(j)}\right\|} = \cos\varphi_u .$$

So the angle is preserved.



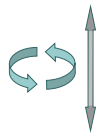
A Rotation Matrix in 2D (5)

- A reasonable comment on the last slide would be: “Well, it is obvious that angles are preserved because both vectors move through an angle theta.”
  - This is true. However, the important point is that we showed preservation of the angles by using an argument that relied only on the property $R_\theta^{\mathrm{T}} R_\theta = I$.
    - No mention was made about the sin, cos structure of the transformation matrix.
  - Consequently, we can simply state that R is a rotation matrix as long as $R^{\mathrm{T}} R = I$.
    - It will preserve both lengths and angles when used as a transformation matrix.



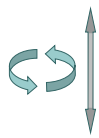
Rotation Matrices in 3D (1)

- We could develop the 3D version of the R_θ matrix.
- This is more of a challenge because in our 3D space, a rotation matrix is defined by three angles.
  - Think of the roll, pitch, and yaw angles that specify the angular position of an aircraft.
  - The 3 by 3 rotation matrix for 3D space is quite complicated with lots of sines and cosines…
- However, to pursue our objective of maximal overlap of proteins we do not really need to know the explicit angles of rotation.
  - A “generic” rotation matrix will do.



Rotation Matrices in 3D (2)

- What do we mean by generic?
  - As long as the rotation matrix R satisfies the rule that $R^{\mathrm{T}}R = I$ we will be guaranteed that both lengths and angles are preserved since the equations for these properties are the same in 3D as they are in 2D except for the change in dimension.
- There is one more issue:
  - The generic rotation matrix could preserve lengths and angles while introducing a mirror image reflection.
  - We must avoid this if we are to maintain the chirality of our rotated molecule.
    - We will address this issue later.



Max. Overlap & Lagrange Multipliers

- Recall: our earlier objective was to get maximum structural overlap by minimizing E, where

$$E = \frac{1}{2}\sum_{i=1}^{N}\left\| Rx^{(i)} - y^{(i)} \right\|^2 .$$

- We now realize that we want to find the matrix R that will minimize E but it must be subject to the rotation *constraint* that $R^{\mathrm{T}}R = I$.
  - Lagrange multipliers can take care of this.
    - But before we derive a Lagrangian, there is still another simplification that can be made:

Restating the Problem

- Note that since $R^{\mathrm{T}}R = I$ we can write:

$$E = \frac{1}{2}\sum_{i=1}^{N}\left\|Rx^{(i)} - y^{(i)}\right\|^2 = \frac{1}{2}\sum_{i=1}^{N}\left(Rx^{(i)} - y^{(i)}\right)^{\mathrm{T}}\left(Rx^{(i)} - y^{(i)}\right)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\left(x^{(i)\mathrm{T}}R^{\mathrm{T}} - y^{(i)\mathrm{T}}\right)\left(Rx^{(i)} - y^{(i)}\right)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\left(x^{(i)\mathrm{T}}R^{\mathrm{T}}Rx^{(i)} - x^{(i)\mathrm{T}}R^{\mathrm{T}}y^{(i)} - y^{(i)\mathrm{T}}Rx^{(i)} + y^{(i)\mathrm{T}}y^{(i)}\right)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\left(\left\|x^{(i)}\right\|^2 - x^{(i)\mathrm{T}}R^{\mathrm{T}}y^{(i)} - y^{(i)\mathrm{T}}Rx^{(i)} + \left\|y^{(i)}\right\|^2\right)$$

$$= \frac{1}{2}\sum_{i=1}^{N}\left(\left\|x^{(i)}\right\|^2 + \left\|y^{(i)}\right\|^2\right) - \sum_{i=1}^{N} y^{(i)\mathrm{T}}Rx^{(i)}.$$

Both of these are equal to the inner product of $y^{(i)}$ and $Rx^{(i)}$.

Independent of R.

So we can minimize E by maximizing this last sum!

Formulating the Lagrangian (1)

- The last slide tells us that we want to maximize

$$H = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}}Rx^{(\gamma)}$$

where R is the 3 by 3 matrix: $R = \begin{bmatrix} r_1^1 & r_1^2 & r_1^3 \\ r_2^1 & r_2^2 & r_2^3 \\ r_3^1 & r_3^2 & r_3^3 \end{bmatrix}$

subject to $R^{\mathrm{T}}R = I$ a constraint that we will rewrite as:

$$\sum_{\gamma=1}^{3} r_\gamma^\alpha r_\gamma^\beta = \delta_\alpha^\beta = \begin{cases} 0 & \text{if } \alpha \neq \beta \\ 1 & \text{if } \alpha = \beta. \end{cases}$$

  - Since $\alpha = 1,2,3$ and $\beta = 1,2,3$ there are nine of these constraints.

Formulating the Lagrangian (2)

- The Lagrangian will be $G = H - F$ where:

$$H = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}} R x^{(\gamma)} \quad \text{and} \quad F = \frac{1}{2}\sum_{\alpha=1}^{3}\sum_{\beta=1}^{3}\lambda_{\beta}^{\alpha}\left[\left(\sum_{\gamma=1}^{3} r_{\gamma}^{\alpha} r_{\gamma}^{\beta}\right) - \delta_{\alpha}^{\beta}\right].$$

  - Recall how multiple constraints are set up in a Lagrangian: Use a linear combination of all the constraints.
    - The λ_{β}^{α} represent the 9 Lagrange multipliers.
    - We have chosen to index them with α and β.
      - Useful later when representing the equations in matrix form.
    - Note that the constraint does not change when we interchange α and β.
    - This symmetry implies $\lambda_{\beta}^{\alpha} = \lambda_{\alpha}^{\beta}$.

Solving for R (1)

- How does H depend on the components of R?
  - We fully expand

$$H = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}} R x^{(\gamma)}.$$

  - $Rx^{(\gamma)}$ is just a 3D vector:

$$Rx^{(\gamma)} = \begin{bmatrix} \sum_{\beta=1}^{3} r_1^{\beta} x_{\beta}^{(\gamma)} \\ \sum_{\beta=1}^{3} r_2^{\beta} x_{\beta}^{(\gamma)} \\ \sum_{\beta=1}^{3} r_3^{\beta} x_{\beta}^{(\gamma)} \end{bmatrix}$$

  - So:

$$H = \sum_{\gamma=1}^{N}\sum_{\alpha=1}^{3} y_{\alpha}^{(\gamma)} \sum_{\beta=1}^{3} r_{\alpha}^{\beta} x_{\beta}^{(\gamma)}.$$

Solving for R (2)

- We will need to take the partial derivatives of G with respect to all 9 components of the R matrix.
  - First working with H:

$$\frac{\partial H}{\partial r_i^{\,j}} = \sum_{\gamma=1}^{N} \frac{\partial}{\partial r_i^{\,j}} \left(\sum_{\alpha=1}^{3} y_\alpha^{(\gamma)} \sum_{\beta=1}^{3} r_\alpha^{\beta} x_\beta^{(\gamma)} \right) = \sum_{\gamma=1}^{N} y_i^{(\gamma)} x_j^{(\gamma)}.$$

Only the term with $\beta = j$ and $\alpha = i$ contributes.

  - To simplify our equations we set:

$$\sum_{\gamma=1}^{N} y_i^{(\gamma)} x_j^{(\gamma)} = c_i^{\,j} \qquad \Rightarrow \qquad C = \sum_{\gamma=1}^{N} y^{(\gamma)} x^{(\gamma)\mathrm{T}}.$$

  - There will be nine of these $c_i^{\,j}$ values, all derived from the input data.

Solving for R (3)

- The partial derivatives of F are a bit more complicated since the R matrix entries appear in a quadratic fashion: $r_\gamma^{\alpha} r_\gamma^{\beta}$
  - We have to apply the product rule when taking derivatives.

$$\frac{\partial F}{\partial r_i^{\,j}} = \frac{1}{2} \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} \lambda_\beta^{\alpha} \left[\left(\sum_{\gamma=1}^{3} \frac{\partial}{\partial r_i^{\,j}} r_\gamma^{\alpha} r_\gamma^{\beta} \right) - \delta_\alpha^{\beta} \right]$$

$\alpha = j \quad \gamma = i$ $\qquad$ $\beta = j \quad \gamma = i$

$$= \frac{1}{2} \sum_{\beta=1}^{3} \lambda_\beta^{j} r_i^{\beta} + \frac{1}{2} \sum_{\alpha=1}^{3} \lambda_j^{\alpha} r_i^{\alpha} = \sum_{\beta=1}^{3} r_i^{\beta} \lambda_\beta^{j}.$$

Change this index to β and then use $\lambda_\beta^{j} = \lambda_j^{\beta}$.

Solving for R (4)

- Finally, since the Lagrangian is $G = H - F$:

$$\frac{\partial G}{\partial r_i^j} = 0 \quad \Rightarrow \quad \frac{\partial H}{\partial r_i^j} = \frac{\partial F}{\partial r_i^j} \quad \Rightarrow$$

$$\sum_{\gamma=1}^{N} y_i^{(\gamma)} x_j^{(\gamma)} = c_i^j = \sum_{\beta=1}^{3} r_i^\beta \lambda_\beta^j \quad \forall i, j.$$

  - By considering these variables to be entries in arrays R, λ, and C we can rewrite this last equation as:

$$C = R\lambda.$$

Notation:

$$R = \begin{bmatrix} r_1^1 & r_1^2 & r_1^3 \\ r_2^1 & r_2^2 & r_2^3 \\ r_3^1 & r_3^2 & r_3^3 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} \lambda_1^1 & \lambda_1^2 & \lambda_1^3 \\ \lambda_2^1 & \lambda_2^2 & \lambda_2^3 \\ \lambda_3^1 & \lambda_3^2 & \lambda_3^3 \end{bmatrix}$$

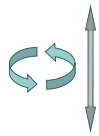
$$C = \begin{bmatrix} c_1^1 & c_1^2 & c_1^3 \\ c_2^1 & c_2^2 & c_2^3 \\ c_3^1 & c_3^2 & c_3^3 \end{bmatrix}$$

Solving for R (5)

- From the previous slide: $C = R\lambda.$
  - We know C. How do we solve for λ and then R?
  - We have used the equation $R^{\mathrm{T}}R = I$ to do various simplifications before we created the Lagrangian but this constraint has not yet been used as a constraint for the Lagrangian analysis itself.
  - So, note: $C^{\mathrm{T}}C = \lambda^{\mathrm{T}}R^{\mathrm{T}}R\lambda = \lambda^{\mathrm{T}}\lambda.$
  - Since $C^{\mathrm{T}}C$ is a square symmetric matrix, we can do an eigen-decomposition:

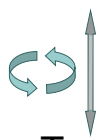
$$\lambda^{\mathrm{T}}\lambda = C^{\mathrm{T}}C = VS^2V^{\mathrm{T}}.$$

  - If we can use this to find an appropriate λ then we set $R = C\lambda^{-1}$ and we are done.



Optimal Proper Rotations (1)

- What is meant by an "appropriate" λ ?
- Selection of λ must be made with due attention to two issues that have not yet been addressed:
  1. The rotation matrix must not introduce a reflection that changes chirality.
     - Preservation of angles and lengths will still allow this!
  2. Although we wanted to minimize E, there is nothing in the Lagrange strategy that guarantees this. The procedure could also lead to an R that maximizes E!
     - The Lagrange strategy only gets you critical rotations that produce extreme values of E.



Optimal Proper Rotations (2)

- To ensure a proper rotation we insist that the determinant of the rotation matrix is +1, that is: $\det(R) = 1$.
- To be sure that E is minimized by our choice of R we must look more deeply into the quantity

$$H = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}} R x^{(\gamma)}$$

to see how its value is determined by choice of R.

  - In particular, the construction of λ will involve the determination of signs of the square roots of the three entries on the diagonal matrix within VS^2V^{T}.

Optimal Proper Rotations (3)

- A very elegant strategy for the computation of R starts with the singular value decomposition of C:
  - Our SVD theory tells us that we can write:

$$C = USV^{\mathrm{T}} \quad \Rightarrow \quad C^{\mathrm{T}}C = \lambda^{\mathrm{T}}\lambda = VS^2V^{\mathrm{T}}$$

where $S^2 = \mathrm{diag}\left(s_1^2, s_2^2, s_3^2\right)$.

  - It is easy to show that:

$$\lambda = V\mathrm{diag}\left(\sigma_1 s_1, \sigma_2 s_2, \sigma_3 s_3\right)V^{\mathrm{T}} \quad \text{with} \quad \sigma_i = \pm 1 \quad i = 1,2,3.$$

  - Since $R\lambda = C$ we can write:

$$R = C\lambda^{-1} = USV^{\mathrm{T}}V\mathrm{diag}\left(\sigma_1 s_1^{-1}, \sigma_2 s_2^{-1}, \sigma_3 s_3^{-1}\right)V^{\mathrm{T}}$$

$$= U\mathrm{diag}\left(\sigma_1, \sigma_2, \sigma_3\right)V^{\mathrm{T}}$$

Optimal Proper Rotations (4)

- So $C = USV^{(\mathrm{T})} \quad \Rightarrow \quad R = U\mathrm{diag}\left(\sigma_1, \sigma_2, \sigma_3\right)V^{\mathrm{T}}$.
  - Now we put this into our H quantity:

$$H = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}} R x^{(\gamma)} = \sum_{\gamma=1}^{N} y^{(\gamma)\mathrm{T}} U\mathrm{diag}\left(\sigma_1, \sigma_2, \sigma_3\right)V^{\mathrm{T}} x^{(\gamma)}$$

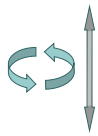
Dot products

$$= \sum_{\gamma=1}^{N}\sum_{k=1}^{3} \sigma_k y^{(\gamma)\mathrm{T}} u^{(k)} v^{(k)\mathrm{T}} x^{(\gamma)} = \sum_{\gamma=1}^{N}\sum_{k=1}^{3} \sigma_k \left(y^{(\gamma)} \bullet u^{(k)}\right)\left(v^{(k)} \bullet x^{(\gamma)}\right)$$

Last two slides explain this:

$$= \sum_{\gamma=1}^{N}\sum_{k=1}^{3} \sigma_k \left(u^{(k)} \bullet y^{(\gamma)}\right)\left(x^{(\gamma)} \bullet v^{(k)}\right) = \sum_{\gamma=1}^{N}\sum_{k=1}^{3} \sigma_k u^{(k)\mathrm{T}} y^{(\gamma)} x^{(\gamma)\mathrm{T}} v^{(k)}$$

$$= \sum_{k=1}^{3} \sigma_k u^{(k)\mathrm{T}} \left[\sum_{\gamma=1}^{N} y^{(\gamma)} x^{(\gamma)\mathrm{T}}\right] v^{(k)} = \sum_{k=1}^{3} \sigma_k u^{(k)\mathrm{T}} C v^{(k)}.$$



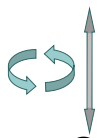
Optimal Proper Rotations (5)

- The SVD of C tells us that $Cv^{(k)} = u^{(k)} s_k$ and so we get a very concise value for H:

$$H = \sum_{k=1}^{3} \sigma_k u^{(k)\mathrm{T}} C v^{(k)} = \sum_{k=1}^{3} \sigma_k u^{(k)\mathrm{T}} u^{(k)} s_k = \sum_{k=1}^{3} \sigma_k s_k .$$

- Recall: E was minimized when H was maximized, so the best E occurs when $\sigma_i = +1 \quad i = 1,2,3.$
  - This gives us:

$$R = U \mathrm{diag}\left(\sigma_1, \sigma_2, \sigma_3\right) V^{\mathrm{T}} = U V^{\mathrm{T}} .$$



Optimal Proper Rotations (6)

- Our previous line: $R = U V^{\mathrm{T}} .$
- So are we finally done?
- Not quite. Remember that we stated that we must have: $\det(R) = 1$.
  - It is possible that the matrix C has a singular value decomposition that leads to $\det(UV^{\mathrm{T}}) = -1$.
    - This is called an *improper* rotation and it introduces a reflection.
  - We can get still get a proper rotation by defining R as:

$$R = U \mathrm{diag}\left(1,1,-1\right) V^{\mathrm{T}} .$$

Optimal Proper Rotations (7)

- Our previous line:

$$R = U\,\mathrm{diag}(1,1,-1)\,V^{\mathrm{T}}.$$

- Why does this work?
  - The determinant of R has changed sign because the determinant of the diagonal matrix now has value -1.
  - So it is a proper rotation.
  - But the value of H is now $s_1 + s_2 - s_3$ and so it is not as large as $s_1 + s_2 + s_3$.
    - So we have somewhat compromised E to get a proper rotation.
  - NOTE: to get the minimal E under these circumstance we make sure that s_3 is the smallest of the three values.
    - That is to say, we are assuming s_1, s_2, s_3 are in descending order.

Summary (1)

- Steps for 3D alignment of proteins P and Q:
  1. Determine the subsequences of alpha carbons to be used in the 3D alignment:

$$M(P) = \left(p^{(\alpha_1)}, p^{(\alpha_2)}, \ldots, p^{(\alpha_N)}\right)$$
$$M(Q) = \left(q^{(\beta_1)}, q^{(\beta_2)}, \ldots, q^{(\beta_N)}\right)$$

  2. Calculate centroids $p^{(c)}$ and $q^{(c)}$.
  3. Shift the proteins so that centroids are at the origin.
     We are then working with $x^{(i)}$ and $y^{(i)}$ coordinate sets.
  4. Calculate the C matrix and compute its SVD.
     This gives $C = USV^{\mathrm{T}}$.
     If necessary reorder the singular values so that $s_1 \geq s_2 \geq s_3$.

Summary (2)

- Steps for 3D alignment (continued):

5. Compute the rotation matrix

$$R = UV^{\mathrm{T}}.$$

6. Check to see if $\det(R) = 1$.
   If this determinant is negative then we must redefine the rotation matrix to be

$$R = U\mathrm{diag}(1,1,-1)V^{\mathrm{T}}.$$

7. Apply the rotation matrix to the $x^{(i)}$ coordinates.

U diag V^{T} Alternate Representation (1)

- Here is a useful matrix manipulation.
  - This is in the linear algebra review notes, but we now make a special note of it.
- Given three matrices:
  - U dimension m x l, with columns $u^{(k)}$ $k = 1, 2, \ldots, l$
  - V dimension n x l, with columns $v^{(k)}$ $k = 1, 2, \ldots, l$ and
  - the diagonal matrix $diag(d_1, d_2, \ldots, d_l)$

then the m x n matrix $U\mathrm{diag}(d_1, d_2, \ldots, d_l)V^{\mathrm{T}}$ can be written as:

$$U\mathrm{diag}(d_1, d_2, \ldots, d_l)V^{\mathrm{T}} = \sum_{k=1}^{l} d_k u^{(k)} v^{(k)\mathrm{T}}.$$

U diag V^{T} Alternate Representation (2)

$$U\mathrm{diag}\left(d_1, d_2, \ldots, d_l\right)V^{\mathrm{T}} = \sum_{k=1}^{l} d_k u^{(k)} v^{(k)\mathrm{T}}.$$

$$\text{L.H.S} = \begin{bmatrix} d_1 u^{(1)} & d_2 u^{(2)} & \cdots & d_l u^{(l)} \end{bmatrix} \begin{bmatrix} v^{(1)\mathrm{T}} \\ v^{(2)\mathrm{T}} \\ \vdots \\ v^{(l)\mathrm{T}} \end{bmatrix}$$

Matrix: i^{th} row is: $d_k u_i^{(k)}$ $k = 1, 2, \ldots, l.$

Matrix: j^{th} column is: $v_j^{(k)\mathrm{T}}$ $k = 1, 2, \ldots, l.$

Product of these has entry at row i and column j equal to:

$$\sum_{k=1}^{l} d_k u_i^{(k)} v_j^{(k)\mathrm{T}}.$$

R.H.S. is a sum of matrices such that entry at row i and column j is equal to:

Calculation of the RMSD

- We can compute the squared distance between each rotated $x^{(i)}$ point and its corresponding $y^{(i)}$ point:

$$d^{(i)2} = \left\| Rx^{(i)} - y^{(i)} \right\|^2$$

  - Then we can compute the Root Mean Square Deviation for the set of corresponding points:

$$\mathrm{RMSD}(P,Q) = \sqrt{\frac{\sum_{i=1}^{N} d^{(i)2}}{N}}$$

$\mathrm{RMSD}(P,Q)$ close to zero $\Rightarrow$ P and Q identical

$1\text{Å} < \mathrm{RMSD}(P,Q) < 3\text{Å} \Rightarrow$ P and Q very similar

$3\text{Å} < \mathrm{RMSD}(P,Q) \Rightarrow$ P and Q have little or no similarity.

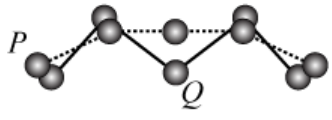
These comments are applied only to the atoms in the alignment.

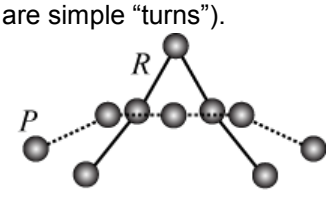
RMSD Issues (1)

- The significance of the RMSD may vary with protein length.
  - For example: It has been observed that two lengthy proteins P and Q (say 500 residues in length) with a vague evolutionary relationship may produce an RMSD that is over 4 Å while two shorter proteins (say 100 residues in length) may produce an RMSD that is less than 3 Å even though they have the same evolutionary distance between them.

RMSD Issues (2)

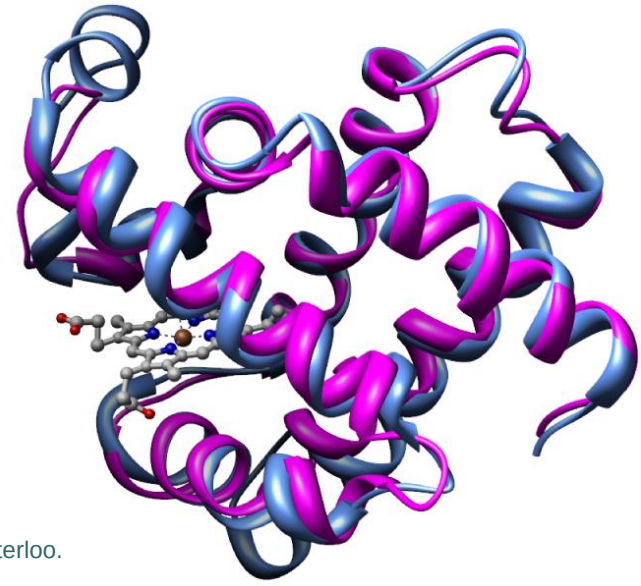
- The best structural alignment is not always achieved by the lowest RSMD.
  - Consider the figure below:
    - Suppose we have "2D molecules" P, Q, and R and we wish to use an $RMSD$ calculation to determine which one of Q or R is most similar to P.
    - The overlap of P and Q may be almost exact because the corresponding atoms have a similar physical alignment.
    - The overlap of P and R has a higher RMSD but the overall shape of R is more like P (both are simple "turns").





Showing Structural Alignment

- A structural alignment of 1MBN and 1JEB:



Acknowledgement:
This image kindly provided by
Shuo (Alex) Xiang,
a graduate student
at the University of Waterloo.

Dealing with Weaker Sequence Similarity

- The problem of similarity evaluation becomes much more of a challenge when a local sequence alignment becomes sketchy, for example, when the two proteins have a distant evolutionary relationship.
  - Since we know that structure is more conserved than sequence, it is reasonable to strive for algorithms that determine structural alignment with little or no help from a preliminary sequence alignment.

Low Sequence Similarity

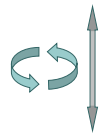
- Complicating Issues:

1. Lengthy sequences of mismatches
   - Mismatches may reside in loop regions while the hydrophobic core contains residues that show a higher percentage of matches in the sequence alignment.
   - It may be reasonable to simply remove the loop region from consideration and try to maximize the overlap of atoms in the hydrophobic core.
2. Presence of gaps in the alignment
   - Gaps in the sequence alignment pose a difficult problem because they indicate a break in continuity of the structural alignment.

Strategies for Low Sequence Similarity

- There are various strategies that can be employed:
  - By breaking up a protein into fragments we can try to derive separate structural alignments of fragments on either side of a gap.

  - We can redefine the pairs of atoms that are to be put into maximal overlap.

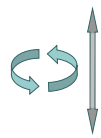


STAMP (1)

- This structural alignment algorithm is due to Russell and Barton (1992):
  1. Perform a local sequence alignment of P and Q to get a set of atom pairs that will be used to define the overlap function.

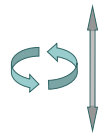
     Work with alpha carbons from a sequence of aligned positions with no gaps.

  2. Derive the translation and rotation matrices that ensure maximum overlap for this set of atom pairs.



STAMP (2)

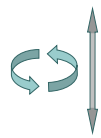
- Continued:
  3. Construct a distance matrix D with axes corresponding to residue positions in each protein and cell $D(i, j)$ holding the distance between alpha carbon i in protein P and alpha carbon j in protein Q.
  4. Compute a similarity matrix for P and Q by subtracting all values in D from the largest distance in D.
  5. Use dynamic programming to compute a high score path through the matrix.



STAMP (3)

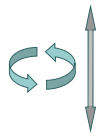
- The optimal path defines a new alignment for P and Q.
- Continue with Steps 2, 3, 4, and 5, repeating until convergence is reached (when there is no change in the path computed in Step 5).

- Several variations of this algorithm can be easily developed.
  - For example, in the dynamic programming step we might alter the recursion so that more weight is given to the part of an alignment that corresponds to helix or strand regions.



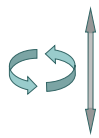
Algorithms Comparing Intramolecular Relationships (1)

- If the proteins P and Q have primary sequences that are the same or very similar, then the basic superposition algorithm using a rotation matrix can be used for the structure comparison.
- If the *structures* are very similar then this strategy still works well, even with low sequence similarity.
  - For example, members of the globin family may have low sequence similarity but can show a significant amount of structure alignment when the STAMP algorithm is applied.
  - There may be some missing segments in the alignment but there is still a large percentage of structural overlap.



Algorithms Comparing Intramolecular Relationships (2)

- When the structural similarity of the proteins is less obvious, it becomes much more difficult to specify the equivalent residues in the comparison.
  - For example, there may be two domains in P that are structurally similar to two domains in Q even though the sequence similarity is weak.
  - If the physical separation of these two domains in P is quite different from the separation of the corresponding domains in Q, or if the domains have a very different spatial orientation, then the superposition strategy will not do well since there will be a poor overall fit between the topologically equivalent substructures.



Algorithms Comparing Intramolecular Relationships (3)

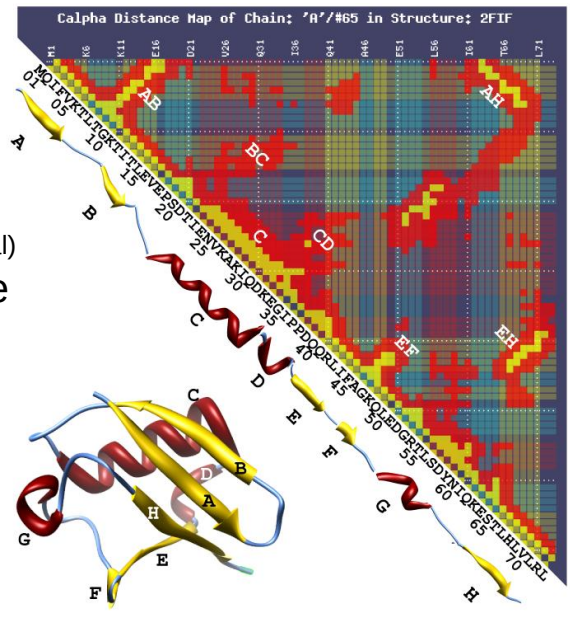
- We need a strategy that works with local *structural* alignments just as the local sequence alignment algorithm depends on local sequence matching.
- We look at two algorithms:
  - DALI (Distance ALIgnment) does an optimal pair wise structural alignment of protein structures based on the similarity of local patterns extracted from distance maps.
  - SSAP (Secondary Structure Alignment Program) produces a structural alignment using double dynamic programming to generate an alignment of local "views" that are common to both proteins.

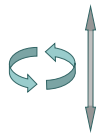
Distance Maps (1)

- Before discussing DALI we consider *distance maps*.
- The map is a square matrix of cells that are indexed by the residues of the protein being studied.
  - In simple versions of a distance map the cell $D[i, j]$ at row i and column j is colored black if the distance between alpha carbon $[i]$ and alpha carbon $[j]$ is less that some particular threshold (say 10 Å), otherwise it is left as white.
  - Note that cell $D[i, j]$ will have the same coloration as cell $D[i, j]$ and so the matrix is symmetric.
    - A more informative map is shown next.

Distance Maps (2)

- Distance map (above the main diagonal) generated by the Moltalk server.
- http://i.moltalk.org.
- PDB id: 2FIF

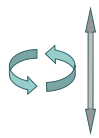




DALI (1)

1. Distance maps (stored as matrices) are computed for both P and Q.
2. Extract a full set of overlapped hexapeptide submatrices from each matrix.

   Each submatrix is a square 6 by 6 array taken from the distance map. These are also called contact patterns.
3. Each of the $(|P|-5)^2$ contact patterns obtained from the distance matrix for P is compared with the $(|Q|-5)^2$ contact patterns obtained from the distance matrix for Q. Each contact pattern of P is paired with its most similar partner in Q. This produces a *pair list*.



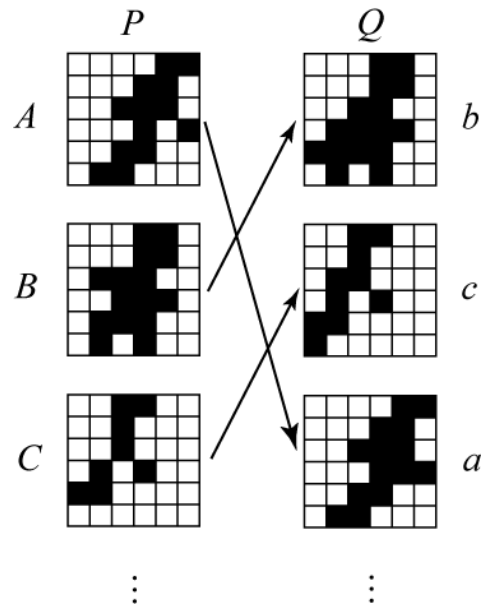
DALI (2)

4. The list is sorted with respect to strength of contact pattern pair similarity. Pairs with low similarity are eliminated.
5. Contact patterns from P are connected to form chains and contact patterns from Q are also connected to form chains.

   Chain forming connections are not made arbitrarily. The connections are generated so that the two chains represent a more extended structural alignment of P and Q.

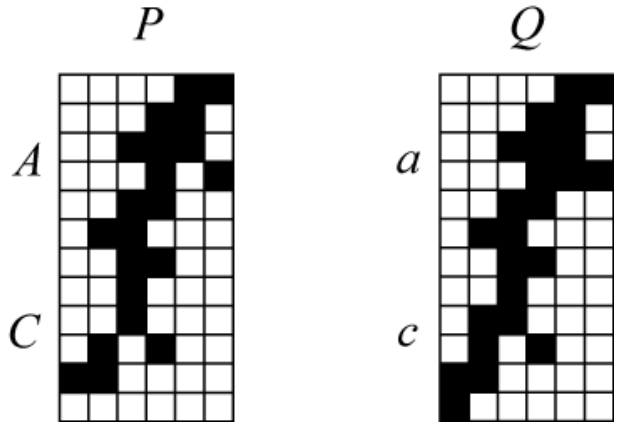
DALI (3)

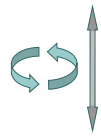
- Matching DALI contact patterns:



DALI (4)

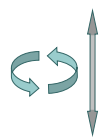
- Chain formation:





DALI (5)

- Monte Carlo:
  - The complexity of the processing arises from the possibility of generating chains in many different ways not all of them useful in the pursuit of a structural alignment.
  - To increase the chances of meaningful chain formation, the algorithm favors the utilization of contact patterns that are ranked high in the sorting step.
  - Nonetheless, the algorithm must search a very large space of possibilities and this is facilitated by means of a Monte Carlo algorithm.



DALI (6)

- Monte Carlo (continued):
  - The Monte Carlo optimizing strategy involves a type of random walk exploration of the search space containing all the chaining possibilities.
  - Moves in this space are randomly chosen.
  - A move corresponds to a change of chain formation and can be evaluated by means of a scoring function.

DALI (7)

- Monte Carlo (continued):
  - The probability p of accepting a move is given by

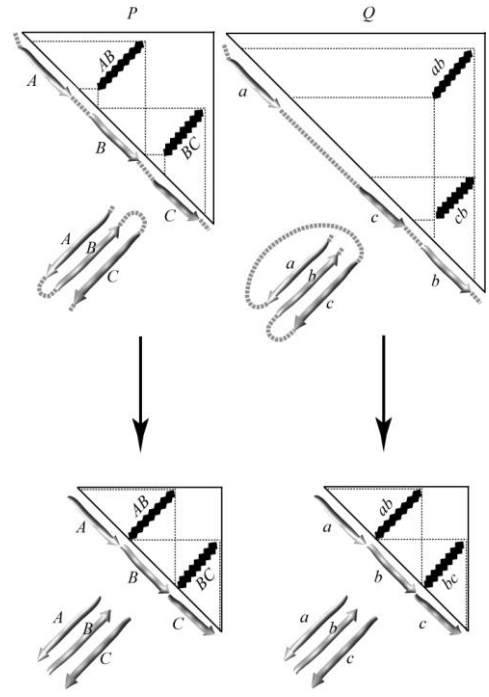
$$p = \exp\left(\beta\left(S' - S\right)\right)$$

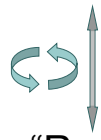
where S' is the new score and S is the old score.

- Parameter β must be carefully chosen.
- Moves with higher scores are always accepted.
- Moves with poor scores are sometimes accepted and this helps the algorithm to get out of local minima that may trap the procedure resulting in less satisfactory structural alignments.

DALI (8)

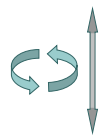
- Reorganizing contact patterns:





DALI (9)

- "Rewiring" (Reorganizing contact patterns):
  - DALI has the ability to detect the similarity of two hydrophobic cores even though the secondary structure components of a core have been "rewired".
  - DALI will analyze the extended contact pattern and recognize that an interchange of fragments c and b in Q will give a better alignment.
  - After this is done, both sets of patterns are collapsed to a representation that reveals the final structural alignment of the three strands in P with the three strands in Q.



SSAP (1)

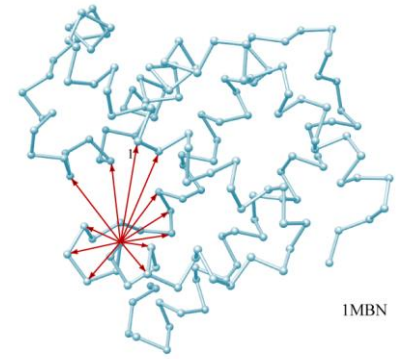
- An approach to the structural alignment problem, that cleverly handles insertions and deletions of residues, was developed by Taylor and Orengo in 1989.
- Their SSAP (Secondary Structure Alignment Program) algorithm relies on the notions of *local views*.
  - These views are used to create an overall structural alignment by means of *double* dynamic programming.

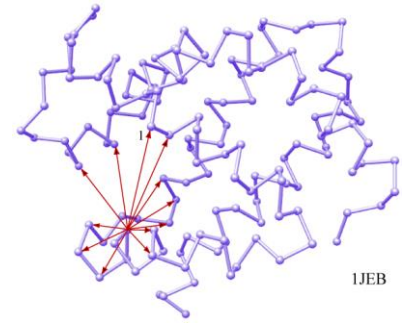
SSAP (2)

- A visual example of views:

	1JEB		1MBN	
	HIS	87	HIS	93
1	LYS	61	THR	67
2	VAL	62	VAL	68
3	ASN	97	TYR	103
4	VAL	96	LYS	102
5	PRO	95	ILE	101
6	LEU	83	LEU	89
7	LYS	82	PRO	88
8	TYR	89	THR	95
9	ILE	90	LYS	96
10	LEU	91	HIS	97
11	TYR	42	LYS	42
12	THR	39	THR	39

The list for a particular view includes all the alpha carbons in a protein.
For clarity, these illustrations only have partial lists.





75

SSAP (3)

- The idea of double dynamic programming is to use a *high level* dynamic programming algorithm to find a structural alignment that is comprised of the largest number of pairs of similar views.
  - However, the evaluation of the similarity of two particular views is itself an optimization problem.

    This is solved by using a "*low level*" dynamic programming strategy.
  - The score matrix used at this low level will be called a *view level matrix*.

    There will be many of these matrices; at most $(|P|*|Q|)$.
  - The single high level score matrix will be called the *consensus matrix*.

76

SSAP: Overview (1)

- Again we assume that P and Q are represented by:

$$\left\{p^{(i)}\right\}_{i=1}^{|P|} \qquad \left\{q^{(j)}\right\}_{j=1}^{|Q|}$$

- SSAP steps:

1. Calculate a *view* for each alpha carbon atom of both P and Q.

   For a particular alpha carbon $p^{(i)}$ the view is a list of vectors.

   Each vector in the list goes from $p^{(i)}$ to another alpha carbon of the same protein.

   Formally, the view for $p^{(i)}$ is the set $\left\{p^{(i,r)}\right\}_{r=1}^{|P|}$ where $p^{(i,r)}$ designates the vector going from $p^{(i)}$ to $p^{(r)}$.

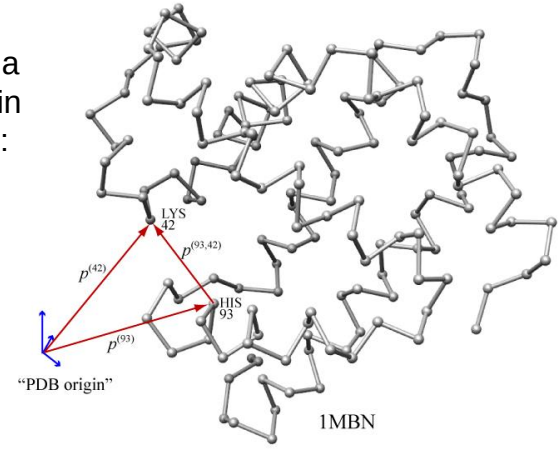
   We use the same notation for $q^{(j)}$.

SSAP: Overview (2)

- Example: LYS 42 as a typical alpha carbon in the view from HIS 93:

LYS 42 is designated by $p^{(93,42)} = p^{(42)} - p^{(93)}$ in the view from HIS 93.

$$p^{(i,r)} = p^{(r)} - p^{(i)}$$



SSAP: Overview (3)

- SSAP steps (continued):

2. Build the $|P|x|Q|$ view matrices:

For each combination of $p^{(i)}$ and $q^{(j)}$, $1 \le i \le |P|$ $1 \le j \le |Q|$, compare vector views using a dynamic programming strategy that fills in a view level matrix with values that are based on the "similarity of vectors".

We refer to the view level matrix as: $V^{(i,j)}$.

- The entry in cell (r, s) is denoted by $V_{r,s}^{(i,j)}$.
- This entry specifies the similarity of vectors $p^{(i,r)}$ and $q^{(j,s)}$.
  - We will discuss "similarity" later.

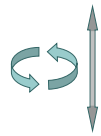
SSAP: Overview (4)

- SSAP steps (continued):

3. Build a consensus matrix

For each view matrix, a dynamic program is used to compute an optimal path score equal to the sum of all view similarity evaluations along the path minus any gap penalties. (Details given later).

If the total path score is above a specified threshold, then the alignment scores on the path are added to accumulating similarity evaluations of the consensus matrix.



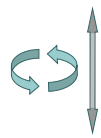
SSAP: Overview (5)

- SSAP steps (continued):

4. Compute an optimal path in the consensus matrix

   Using dynamic programming, derive a set of equivalent residues by finding an optimal path in the consensus matrix.

   - There are many variations on these ideas.
   - We now provide more details:



SSAP: Building the consensus matrix (1)

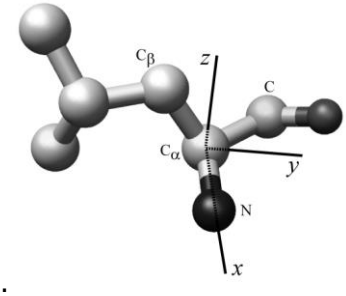
- Step 2 asks for $|P||Q|$ matrices.
- We can reduce this by filtering the potential pairs.
  - For example, we would not compare the views of $p^{(i)}$ and $q^{(j)}$ if $p^{(i)}$ is in a helix and $q^{(j)}$ is in a strand.
  - However, this type of secondary structure filtering requires that we have a reliable assessment of the secondary structure status of a residue.
  - To avoid the discretization error that this implies, one may resort to a filtering strategy that compares the dihedral angles on either side of $p^{(i)}$ with the corresponding dihedral angles of $q^{(j)}$.
    - Authors of SSAP also filtered with respect to solvent accessibility.

SSAP: Building the consensus matrix (2)

- Recall: the view level matrix is: $V^{(i,j)}$.
  - The entry in cell (r, s) is denoted by $V_{r,s}^{(i,j)}$.
  - This entry specifies the similarity of vectors $p^{(i,r)}$ and $q^{(j,s)}$.
    - Initially the authors simply compared the lengths of these two vectors but abandoned this idea because it was not sensitive to direction of the vectors involved.
    - Instead they establish $p^{(r)} - p^{(i)}$ in a frame of reference that has the alpha carbon $p^{(i)}$ at the origin.
      - The same is done for $q^{(s)} - q^{(j)}$.

SSAP: Building the consensus matrix (3)

- Frame of reference at the alpha carbon $p^{(i)}$:
  - The three atoms: C_α along with the C and N atoms bonded to it define a plane.
  - With the origin at C_α, we let the C_α – N bond be the x-axis.
  - Then the y-axis is in the plane and perpendicular to the x-axis.
    - Choose the positive direction of y to lie in the same direction as C.



  - The z-axis is perpendicular to the plane and in the same direction as the C_β atom of the residue attached to C_α.
    - We use the hydrogen atom that replaces C_β in the case of glycine.
  - Each axis is represented by a normalized vector.

SSAP: Building the consensus matrix (4)

- By calculating the inner product of $p^{(r)} - p^{(i)}$ with each these orthonormal vectors we get the coordinates of $p^{(r)} - p^{(i)}$ in this local frame of reference.
- In the same fashion, a local frame of reference is constructed for $q^{(j)}$ and the coordinates of $q^{(s)} - q^{(j)}$ are calculated with respect to this frame of reference.
- We can now treat these newly computed coordinates as if they are relative to the ***same*** frame of reference.

SSAP: Building the consensus matrix (5)

- Suppose $p^{(r)} - p^{(i)}$ in this computed frame of reference is represented by the column vector

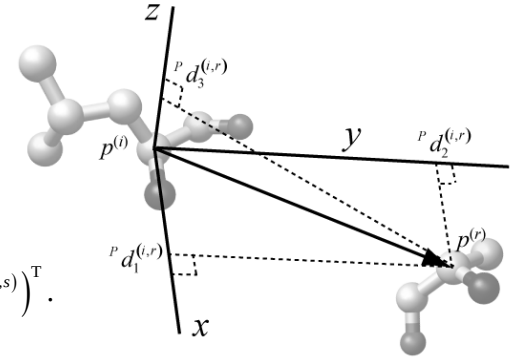
$${}^{P}d^{(i,r)} = \left({}^{P}d_1^{(i,r)} \quad {}^{P}d_2^{(i,r)} \quad {}^{P}d_3^{(i,r)} \right)^{\mathrm{T}}.$$

- Suppose

$$q^{(s)} - q^{(j)}$$

is

$${}^{Q}d^{(j,s)} = \left({}^{Q}d_1^{(j,s)} \quad {}^{Q}d_2^{(j,s)} \quad {}^{Q}d_3^{(j,s)} \right)^{\mathrm{T}}.$$



SSAP: Building the consensus matrix (6)

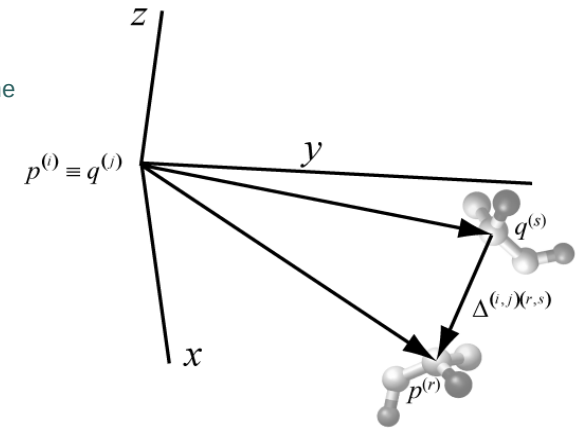
- Define: $\Delta^{(i,j)(r,s)} = {}^{P}d^{(i,r)} - {}^{Q}d^{(j,s)}$

$$= \left({}^{P}d_1^{(i,r)} - {}^{Q}d_1^{(j,s)} \quad {}^{P}d_2^{(i,r)} - {}^{Q}d_2^{(j,s)} \quad {}^{P}d_3^{(i,r)} - {}^{Q}d_3^{(j,s)} \right)^{\mathrm{T}}.$$

Note: A small norm for $\Delta^{(i,j)(r,s)}$ would indicate that the view at $p^{(i)}$ is similar to the view at $q^{(j)}$ at least as far as the vectors $p^{(r)}$ and $q^{(s)}$ are concerned.

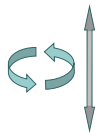
Note, for later:

$$\Delta^{(i,j)(i,j)} = {}^{P}d^{(i,i)} - {}^{Q}d^{(j,j)}$$
$$= 0 - 0 = 0.$$



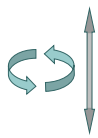
SSAP: Building the consensus matrix (7)

- Our goal is to establish more indications of vector similarity for several other values of r and s.
- The dynamic program at this view level will essentially extract an alignment of the residues in such a way as to maximize the number of similar pairs of vectors $p^{(r)}$ and $q^{(s)}$.
  - NOTE: we never have to rely on getting a sequence alignment first!



SSAP: Building the consensus matrix (8)

- In the original paper, the authors decided to avoid the square roots involved in the calculation of a Euclidean distance in order to save computation time.
- Consequently, the norm squared is used in place of the vector length.
- They also needed to convert a measure of difference into a measure of similarity.
  - This was done by using a hyperbolic formula for the view matrix entries (see next slide).

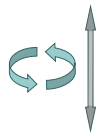


SSAP: Building the consensus matrix (9)

- The entry at cell (r, s) in view matrix $V^{(i,j)}$ is given by:

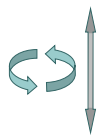
$$V_{r,s}^{(i,j)} = \frac{a}{b + \left\| \Delta^{(i,j)(r,s)} \right\|^2}$$

- Experiments have determined that $a = 50$ and $b = 2$ give good results.
  - Note that with these parameter settings, $V_{i,j}^{(i,j)}$ is always $a/b = 25$.



SSAP: Building the consensus matrix (10)

- Recall that the view matrix $V^{(i,j)}$ was built assuming that $p^{(i)}$ would be put into an equivalence with $q^{(j)}$.
- So: all residues prior to $p^{(i)}$ can only be aligned with residues prior to $q^{(j)}$ and all residues after $p^{(i)}$ can only be aligned with residues after $q^{(j)}$.
- This gives the view matrix $V^{(i,j)}$ a rather peculiar appearance.
  - An entry in cell (r,s) with $r < i$ and $s > j$ will be undefined.
  - As well, an entry in cell (r,s) with $r > i$ and $s < j$ will be undefined.



SSAP: Building the consensus matrix (11)

- A typical view matrix:
  - There will be many of these…

In this example we use the peptide sequences:

P = "NEEDLEMAN"
and
Q = "WATERMAN".

	W	A	T	E	R	M	A	N
N	1	1	2	2				
E	14	2	1	2				
E	1	4	12	1				
D					25			
L						3	1	2
E						8	2	1
M						11	1	3
A						1	11	3
N						2	3	14

- The dynamic program to compute an optimal path in the view matrix is an adaptation of the dynamic program used for the global sequence alignment problem.

SSAP: Building the consensus matrix (12)

- We start at location (i, j) of the score matrix $S^{(i,j)}$ and fill in the bottom right submatrix using the following recursion which is valid for $r > i$ and $s > j$:

Base cases:

$$S_{i,s}^{(i,j)} = \frac{a}{b} - g(s-j) \qquad S_{r,j}^{(i,j)} = \frac{a}{b} - g(r-i).$$

Recurrence for cell (r, s):

$$S_{r,s}^{(i,j)} = \max \begin{cases} S_{r-1,s-1}^{(i,j)} + V_{r,s}^{(i,j)} \\ S_{r-1,s}^{(i,j)} - g \\ S_{r,s-1}^{(i,j)} - g. \end{cases}$$

93

SSAP: Building the consensus matrix (13)

- We start at location (i, j) of the score matrix $S^{(i,j)}$ and fill in the top left submatrix "going in the opposite direction" using the following recursion which is valid for $r < i$ and $s < j$:

Base cases:

$$S_{i,s}^{(i,j)} = \frac{a}{b} - g(j-s) \qquad S_{r,j}^{(i,j)} = \frac{a}{b} - g(i-r).$$

Recurrence for cell (r, s):

$$S_{r,s}^{(i,j)} = \max \begin{cases} S_{r+1,s+1}^{(i,j)} + V_{r,s}^{(i,j)} \\ S_{r+1,s}^{(i,j)} - g \\ S_{r,s+1}^{(i,j)} - g. \end{cases}$$

94

SSAP: Building the consensus matrix (14)

- After the matrix is filled we locate the maximum element in the upper left submatrix and initiate a trace-back from this cell to cell (i, j).
- Similarly, starting at the maximum element is the lower right submatrix we initiate another trace-back that ends in cell (i, j).
  - In the next figure, each score in the trace-back path is set in bold font and enclosed by an ellipse.
  - In the examples given we are working with a gap penalty of $g = 4$.

SSAP: Building the consensus matrix (15)

- Score matrix for the previous example:

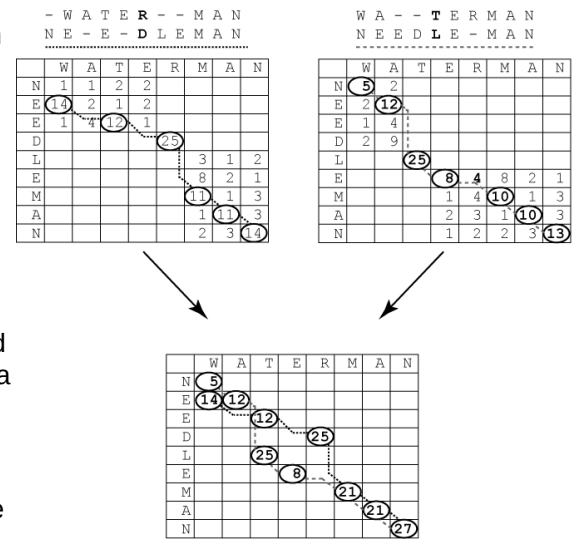
	W	A	T	E	R	M	A	N
N	39	31	25	19	13			
E	**43**	35	29	23	17			
E	25	**29**	**33**	26	21			
D	9	13	17	**21**	**25**	21	17	13
L					**21**	28	24	20
E					**17**	29	30	26
M					13	**28**	30	33
A					9	24	**39**	35
N					5	20	35	**53**

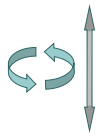
SSAP: Building the consensus matrix (16)

- The score for this path is the sum of the two maximum scores (96 in this score matrix).
- If this path score is above a pre-selected threshold, then the path elements that represent matches in the alignment are added to the corresponding elements of the consensus matrix.
  - This is shown in the next figure.

SSAP: Building the consensus matrix (17)

- The first matrix is $V^{(4,5)}$, from the views for $p^{(4)}$ and $q^{(5)}$.
- The alignment to get the optimal path (marked by a black dotted line) is given just above the matrix.
- The second matrix represents $V^{(5,3)}$.
- Its optimal alignment is also shown above the matrix and the path is designated with a gray dashed line.
- Of course we will need many such additions to the consensus matrix before we are ready for the next step.





SSAP: Computing the alignment

- Compute the optimal path in the consensus matrix:
  - After the consensus matrix has been constructed, an optimal path is derived using a conventional Smith-Waterman algorithm.
  - The result of the SSAP algorithm is an alignment that gives an equivalence set for the various segments of P and Q that are presumed to have structural similarity.