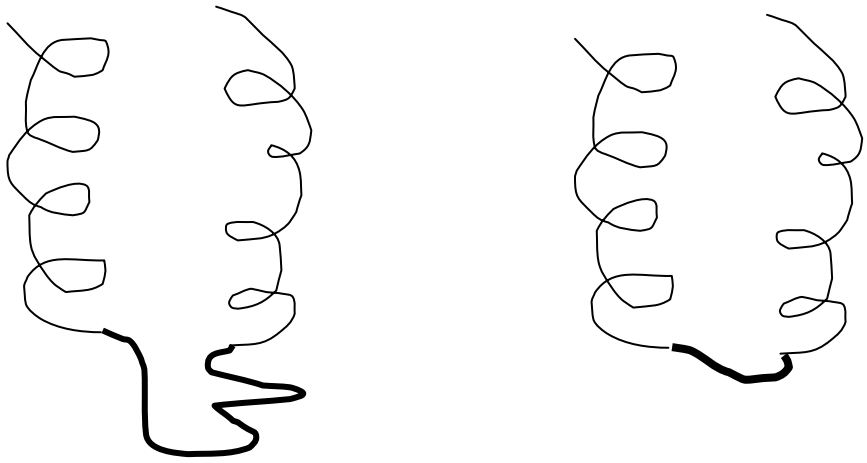
Structural Overlap

Consider the two shapes below



We should be able to detect the similarity between the fragments (helices), and point to the loop as the structural segment that deviates the most, regardless of the sequence. Since the identities of the amino acids are not used (only the C_α positions) highly remote evolutionary connections may be observed. This is our final goal. We start however with the (much) simpler case of overlapping proteins of the same length (no alignment is necessary just proper measure of their distance).

Computing the distance between protein structures

We consider two proteins A and B with the same number of amino acids n (the question of alignment of two structures with different number of amino acids will follow the simpler case of overlap). The coordinate vectors of protein A and B are denoted by X_A and X_B respectively. Each of these vectors is of length $3n$ including the (x,y,z) (Cartesian) positions of the C_α-s of the amino acids. The rank 3 vector of amino acid i in structure A is denoted by r_i^A. The distance between the two structures D is defined (and written explicitly as)

$$D^2 = \sum_{i=1}^{n} \left(r_i^A - r_i^B \right)^2$$

Hence we think on the two proteins as a collection of points, or alternatively as a point in $3n$ space for which we compute norm two of the vector difference $\left\| X_A - X_B \right\|_2$

Since the coordinates are defined in Cartesian space, it is possible to translate or rotate one of the structures with respect to the other without changing any of the internal

distances between the points that belong to the same object, the protein. That is, maintaining its rigid shape. For simplicity we will always move structure A.

We will consider the translation and the rotation separately. A translation is defined by adding to each of the r_i^A vector a single constant vector t. A rotation is defined by multiplying a coordinate vector by a 3x3 matrix U (e.g. Ur_i^A). U satisfies $UU^t = 1$ and $\det(U) = 1$ the usual condition on a rotation matrix that we discussed earlier.

Let us start with the simpler problem, that of translation. We wish to determine a vector of translation t that will be added to each of the atom in protein A so that D^2 is minimal. This is trivial

$$D^2 = \sum_{n=1}^{N} \left(r_n^A + t - r_n^B \right)^2 = \text{minimum}$$

$$2\frac{dD}{dt_\eta} = 2\sum_{n=1}^{N} \left(r_n^A + t - r_n^B \right)_\eta = 0$$

$$t_\eta = \frac{1}{N}\sum_{n=1}^{N} \left(r_{\eta n}^B - r_{\eta n}^A \right) = \frac{1}{N}\sum_{n=1}^{N} r_{\eta n}^A - \frac{1}{N}\sum_{n=1}^{N} r_{\eta n}^B = r_{\eta(gc)}^A - r_{\eta(gc)}^B$$

η=x,y,z

Hence, all we need to do is to correct the position of r_i^A by the difference in the geometric centers of the two proteins $r_{\eta(gc)}^A$ and $r_{\eta(gc)}^B$. After doing this we will be ready to consider the more interesting problem of overlapping two structures, the problem of rotation.

In fact, to make sure that the next item on the agenda is pure rotation we will set the two geometric centers of the two proteins to zero. In the following derivation we assume that this was already done. We will keep the same notation of r_i^A and r_i^B for the vectors with the adjusted translation.

To correct for possible rotations we write yet another optimization problem. The unknown below is the rotation matrix U. The structures are known and are presumed rigid. The distance between the two structures is a function of the rotation matrix, and we need to pick such a rotation that makes the distance as small as possible (minimal). As we shall see this problem has a unique solution that will be extremely useful for further analysis. Of course the rotation matrix U cannot be any matrix it must satisfy the obvious conditions we stated earlier. It must keep the overall shape of the protein the same (hence the proteins must be rotated with respect to each other as rigid bodies). We therefore must have $UU^t = I$ to preserve all the internal distances in the protein . We also

must avoid reflection $\left(\det\left(U\right)=1\right)$ since reflection changes the so-called "chirality" of proteins and their chemical identity. We shall deal with distance conservation first ($UU^t = I$) and only later return to the reflection problem $\left(\det\left(U\right)=1\right)$.

After the lengthy introduction here is the optimization task that we are facing: Minimize D^2 as a function of the matrix U . U is a rotation matrix.

$$D^2 = \sum_{n=1}^{N}\left(Ur_n^A - r_n^B\right)^2 = \text{minimum}$$

subject to the constraint: $UU^t = I$

or $\sum_{k=1}^{3} u_{ki}u_{kj} - \delta_{ij} = 0$

We have used the notation $u_{ki} = \left(U\right)_{ki}$ and $\delta_{ij} = \begin{Bmatrix} 1 & i = j \\ 0 & i \neq j \end{Bmatrix}$

Note that the condition $UU^t = I$ is a constraint on a matrix, or alternatively 9 different equations (an equation for each of the elements in the matrix). Some of the conditions are redundant, how many?

Using the "mechanic" of the Lagrange's multipliers we introduced earlier, we add the constraints to the target function that we wish to optimize.

$$F = D^2 + \sum_{i,j}\Lambda_{ij}\left(\sum_k u_{ki}u_{kj} - \delta_{ij}\right)$$

The unknowns that we wish to determine are all the elements of the U matrix (9 in all). However, the constraints reduce the number of unknown to 3. This must be the case since are we argued earlier a rotation is completely determined once three parameters (three rotations angles) are given.

To find the minimum of D^2 subject to the constraint of unitary matrix U (another way of saying $UU^t = I$), we differentiate with respect to the matrix element u_{ij}, we have

$$\frac{\partial F}{\partial u_{ij}} = \sum_k u_{ik}\left(\sum_n r_{nk}^A r_{nj}^A + \lambda_{kj}\right) - \sum_n r_n^A r_n^B = 0$$

We now define two matrices

$$R_{ij} = \sum_n r_{ni}^B r_{nj}^A \qquad S_{ij} = \sum_n r_{ni}^A r_{nj}^A$$

More on overlapping shapes (part II)

Note that S_{ij} is a symmetric matrix while R_{ij} is not. It is also interesting to note that the matrix of the Lagrange multipliers -- $\lambda_{ij} = (\Lambda)_{ij}$ is symmetric too. Can you prove it?

With the help of the above definition we can write $\frac{\partial F}{\partial u_{ij}} = 0$ in a more compact form

$$U(S+\Lambda) - R = 0$$

We have one matrix equation with two unknown matrices (!) -- U and Λ. Of course, things are not so bad since we still have the constraint equation: $UU^t = 1$
Note also that $(S+\Lambda)$ is a symmetric matrix. On the other hand R is not symmetric which makes our problem a little more interesting. The following trick will eliminate some of our problems: Multiply the last equation by its transpose:
$$(S+\Lambda)^t U^t U (S+\Lambda) = R^t R$$

and using $U^t U = 1$ (our favorite constraint) eliminates U from the equation. This does not seem like a positive step since the rotation matrix U is what we are after… Nevertheless, some insight to the problem will be given from the equation below

$$(S+\Lambda)(S+\Lambda) = R^t R$$

The eigenvectors of $(S+\Lambda)$ -- a_k are the same as the eigenvectors of $R^t R$ (assuming no eigenvalue degeneracy for the symmetric matrix $(S+\Lambda)$). The eigenvalues of $R^t R$ are μ_k^2. The corresponding eigenvalues of $(S+\Lambda)$ are therefore

$(S+\Lambda)a_k = \pm\mu_k a_k$ (the eigenvalues of the symmetric matrix must be real but since we have only the eigenvalues of the square of the matrix, the eigenvalues themselves are determined only up to a sign)

We now use the eigenvectors a_k to reconsider the matrix equation after multiplying from the right with a_k
$$U(S+\Lambda)a_k = Ra_k$$
Since a_k is an eigenvector of $(S+\Lambda)$ we can also write
$$U(\pm\mu_k)a_k = Ra_k$$

We have three orthogonal eigenvectors $a_k \quad k=1,2,3$. The rotation matrix U transforms these three vectors to another set that we call $b_k \quad k=1,2,3$. Note that the b_k set is also a set of orthogonal vector. This is easy to appreciate as follows:

$$\left(b_i\right)^t \left(b_j\right) = \left(Ua_i\right)^t \left(Ua_j\right) = a_i^t U^t U a_j = a_i^t a_j = \delta_{ij}$$

Using the "b" notation we can also write

$$b_k = \pm \frac{1}{\mu_k} R a_k$$

Note that right hand side includes only known (by now) entities. So we can use R, a_k and μ_k to compute the b_k-s. Since we also knows that

$$U a_k = b_k$$

We can reconstruct the rotation matrix as a solution to the above equation, i.e.,

$$U = \sum_{k=1}^{3} b_k a_k^t$$

where we have used the orthogonality of the a_k-s. This "optimal" U can be plugged in the initial equation for the distance to compute the "optimal" distance. There is however a few more subtle points that are discussed below, and we postpone for the moment the calculation of the distance.

We note that we can also write an expression for the matrix R in terms of the two sets of vectors

$$R = \sum_{k=1}^{3} b_k \left(\pm \mu_k\right) a_k^t$$

A few more comments: The set of orthonormal vectors b_k is obtained by rotating the set a_k with the (unknown) U. However the b_k are also the "left" eigenvectors of R. The right and the left eigenvectors, and the eigenvalues can be obtained directly from Singular Value Decomposition (SVD) of the asymmetric matrix R. Using SVD (without going into details exactly what it means) is the simplest approach to our problem using the facilities and the resources of MATLAB.

Finally our optimal distance can be computed more directly without thinking on U at all (of course to make a nice plot of overlapping structures requires the rotation matrix. In contrast to the argument below the rotation matrix is also required to avoid inversion):

$$D^2 = \sum_n \left(Ur_n^A - r_n^B\right)^2 = \sum_n \left(r_n^A\right)^2 + \left(r_n^B\right)^2 - 2\sum_n r_n^B \left(Ur_n^A\right)$$

$$= \sum_n \left(r_n^A\right)^2 + \left(r_n^B\right)^2 - 2\sum_n \sum_k \left(b_k r_n^B\right)\left(r_n^A a_k\right)$$

$$= \sum_n \left(r_n^A\right)^2 + \left(r_n^B\right)^2 - 2\sum_k \left(b_k\right)\left(Ra_k\right)$$

$$= \sum_n \left(r_n^A\right)^2 + \left(r_n^B\right)^2 - 2\sum_k \pm\mu_k$$

where we have used the known forms of U and R in terms of the vectors a_k and b_k to arrive at the final amazingly simple expression in the last line. Since the sum $\sum_n \left(r_n^A\right)^2 + \left(r_n^B\right)^2$ is a constant, the only term that can make a difference is the $-2\sum_k \pm\mu_k$. If we wish (and we do!) to make the distance as small as possible we should only positive values for the μ_k. Hence to compute the minimal distance we need to compute only the eigenvalues of $R^t R$ and not the eigenvectors.

There is however one caveat. So far we made sure that the constraint $UU^t = I$ is satisfied, however, we did not take care of the second condition for a proper rotation $\det\left(U\right) = 1$. It is possible that the rotation matrix defined by $U = \sum_{k=1}^{3} b_k a_k^t$ has a determinant of –1. This can be tested for by explicit construction of the rotation matrix and calculation of the determinant.

What are we going to do if the determinant is negative?

We clearly need to modify the rotation matrix to have a determinant of +1. This is the place where we can go back and re-investigate the $\pm$ sign we have before the eigenvalue. The smallest possible distance between the structures will be obtained for all positive eigenvalues (choosing only the + sign). However, if the rotation is not proper (determinat is equal to –1), we may need to compromise on something else. We can change the sign of the determinant by changing the sign of the vector b_k to $-b_k$. A negative eigenvalue $-\mu_k$ means that we also change the sign of the "secondary" eigenvector b_k to $-b_k$. (Note $Ra_k = \mu_k b_k$ if we change the sign of the vector b_k we must change the sign of the eigenvalue to maintain the equality i.e. $Ra_k = \left(-\mu_k\right)\cdot\left(-b_k\right)$. Since the eigenvalues of U are all of norm 1, changing the sign of one the left eigenvectors $\left(b_k\right)$ changes the sign of the corresponding eigenvalue and the sign of the determinant (as desired). The distance between the two proteins will not be the shortest possible after the adjustment but this is the price we have to pay in order to obtain a proper rotation.