Comparing the solvent-accessible surfaces of proteins

Dr Kim Evans

Outline of talk

1. Statement of the protein surface matching problem

2. Tools for comparing small unlabelled point sets

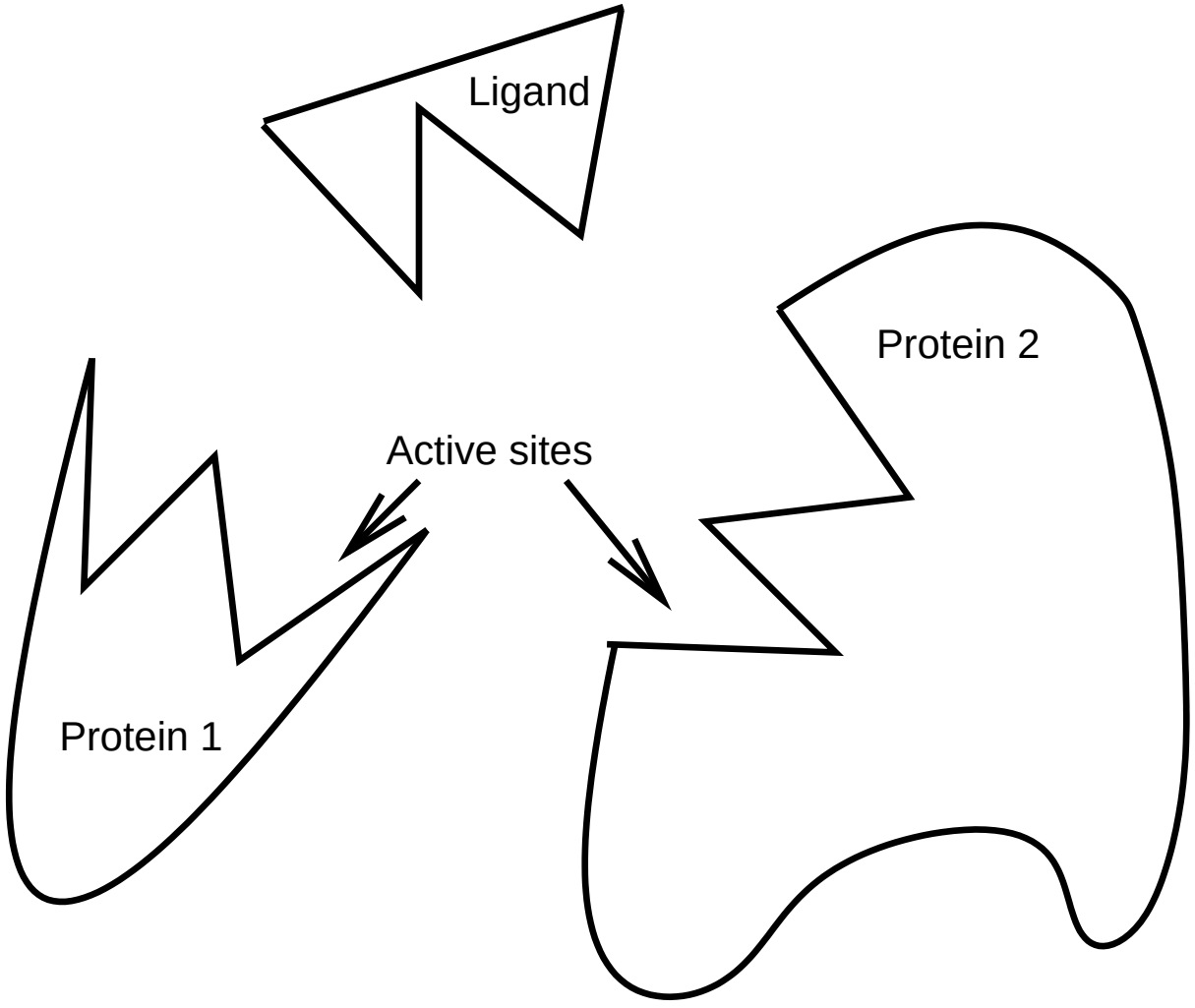
3. Markov Chain Monte Carlo simulation

4. Some simple statistics to help with the search

5. Results

The problem

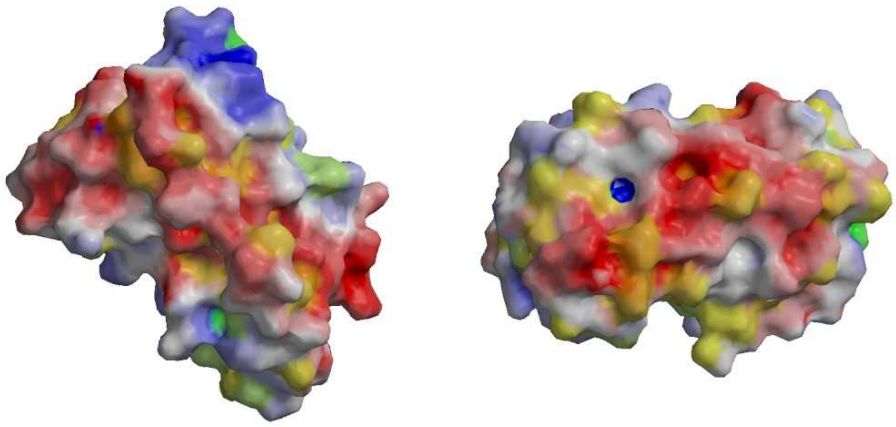
Comparing the geometry of protein surfaces to look for binding sites (or active sites) the proteins have in common



Why is this difficult?

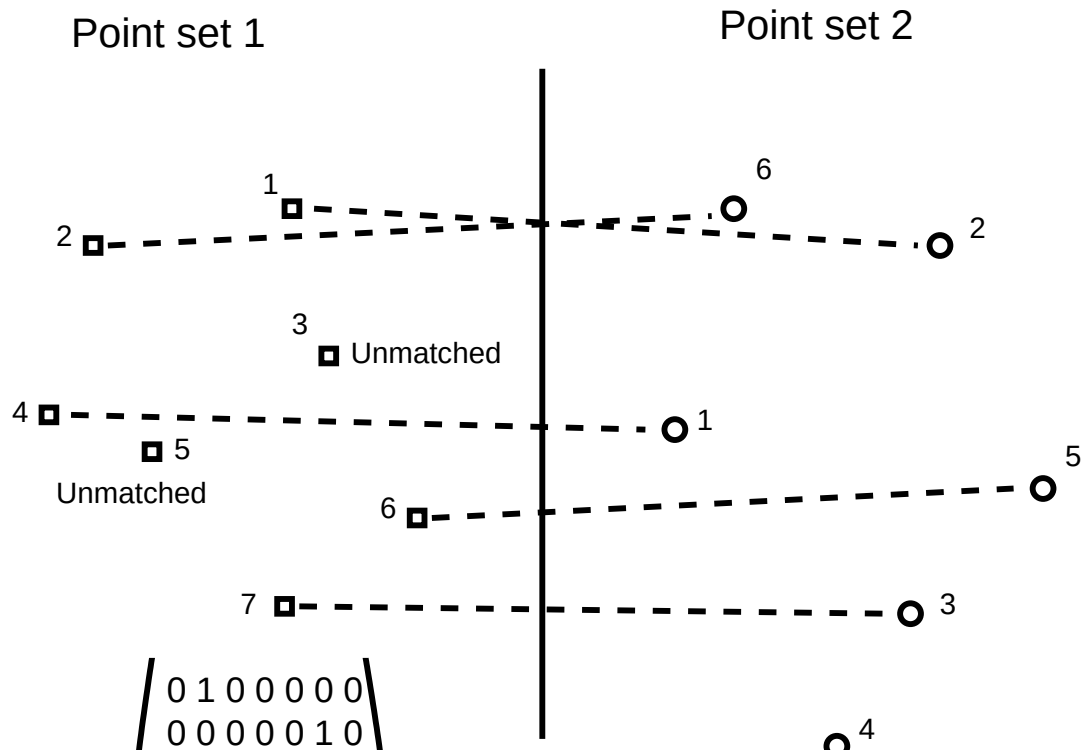
1. Protein surfaces are typically very large (tens of thousands of atoms defining the surface is not uncommon)

2. The surfaces are very ‘knobbly’



Comparing small unlabelled point sets

Suppose we had found two regions, one on each protein, that we thought might be the same shape. There is still a problem - the points are unlabelled (i.e. arbitrarily labelled). How do we decide which atoms to match?



$$\Lambda = \text{(match matrix)} \quad \begin{pmatrix} 0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0 \end{pmatrix}$$

Inference on the match matrix

We use Markov Chain Monte Carlo (MCMC) simulation to draw inference about the optimal match matrix.

The steps involved are as follows:

1. Update match matrix by choosing one row at random and moving the 1 to a different position

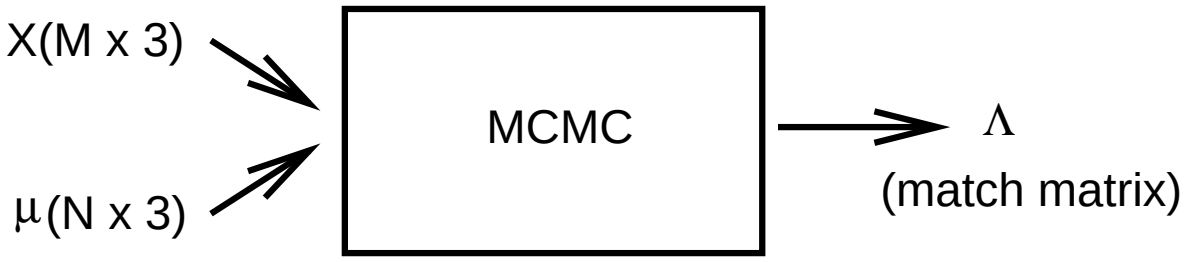
2. Set $\alpha = \min\left(1, \frac{\pi(\Lambda^*|\sigma^2,X,\mu)}{\pi(\Lambda|\sigma^2,X,\mu)}\right)$, where σ^2 is a variance parameter

3. Accept new match matrix with probability α

4. Repeat for large number of iterations

Summary of MCMC procedure

Input: Two unlabelled point sets,X and μ represented as matrices



Looking for possible candidates for binding sites in common - Introducing the HPC

Choose an atom, i, in protein 1 and an atom, j in protein 2

Let $B_1(i)$ be the nearest N points to atom i in protein 1

Let $B_2(j)$ be the nearest N points to atom j in protein 2

We write $d_{1(1)}, d_{1(2)}, \ldots, d_{1(N)}$ for the ordered distances of atoms in $B_1(i)$ from atom i in protein 1

We write $d_{2(1)}, d_{2(2)}, \ldots, d_{2(N)}$ for the ordered distances of atoms in $B_2(j)$ from atom j in protein 2

TT statistics

Having chosen atoms i and j, we define a statistic TT to compare the vectors $(d_{1(1)} \dots d_{1(N)})$ and $(d_{2(1)} \dots d_{2(N)})$ of distances from the atoms i and j.

$$TT(i,j) = \sum_{k=1}^{N} |d_{1(k)} - d_{2(k)}|^2$$

Rationale: If the atoms in the balls $B_1(i)$ and $B_2(j)$ are in a similar arrangement one would expect TT to be small.

BUT a small TT statistic does not guarantee that the arrangements of atoms are the same.

The S statistic to improve the search for candidates for the MCMC algorithm

The idea of the S statistic is that we choose two atoms in each ball of points that are close to the central atom (i or j) and align the balls based on the positions of those four atoms.

Let us label the two atoms in ball 1 i_1 and i_2 and the two atoms in ball 2 j_1 and j_2. The alignment process is as follows:

1. Centre both balls of points

2. Rotate ball 1 so that the vector from the origin to atom i_1 is pointing in the direction of the north pole. Rotate ball 2 similarly for atom j_1

3. Rotate ball 1 again so that the vector from the origin to atom i_2 lies in the $x - z$ plane. Rotate ball 2 similarly for atom j_2.

This alignment brings atom i_1 and j_1 as close together as possible and atoms i_2 and j_2 as close together as possible given the condition on atoms i_1 and j_1.

We then define the S statistic, $S(i,j)$, as the sum of squares of distances from each point in the rotated ball 1 to its nearest point in the rotated ball 2

$$S(i,j) = \sum_{k=1}^{N} \min_{1 \leq l \leq N} \parallel [B_1(i)]_k - [B_2(j)]_l \parallel^2$$

Embedding the balls for which both TT(i,j) and S(i,j) are small in larger point sets

If for a particular choice of atoms i and j we find that both TT(i,j) and S(i,j) are small, then there is a good chance that the balls $B_1(i)$ and $B_2(j)$ are the same shape. We therefore need to use the MCMC algorithm to draw inference about the corresponding match matrix.

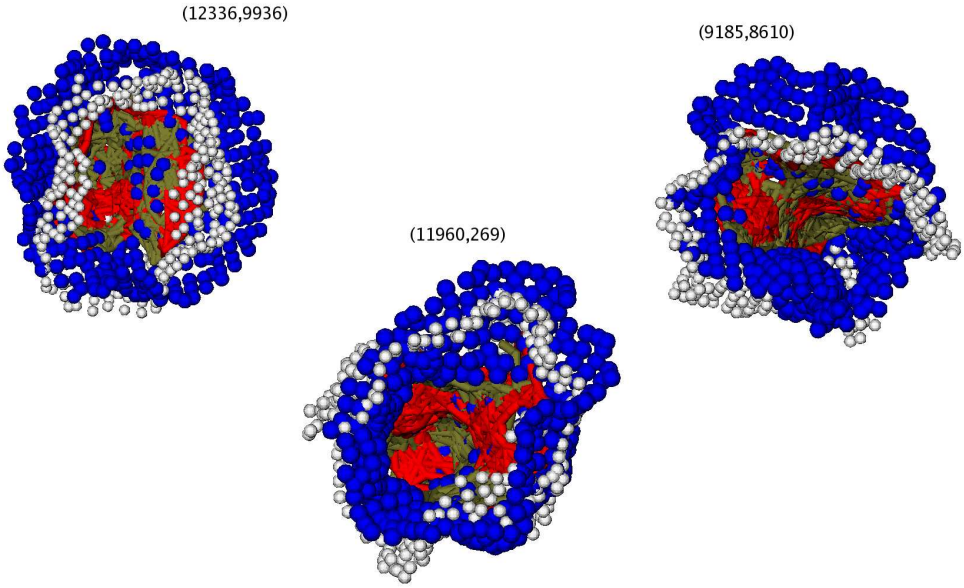
We embed each ball in a larger point set, centered at atom i for $B_1(i)$ and atom j for $B_2(j)$. To start us off on the MCMC algorithm, we initially match each atom in the rotated $B_1(i)$ to the nearest point in the rotated $B_2(j)$. We consider all the other atoms in the larger point sets to be unmatched.

Our starting point for the match matrix is given on the next slide.

The initial match matrix for the MCMC algorithm

$$\hat{\Lambda} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ 0 & & & & 0 & \dots & 0 & 0 \\ \vdots & & \Lambda_{small} & & \vdots & \vdots & \vdots & \vdots \\ 0 & & & & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

Results



Here we have three pictures generated by applying the MCMC algorithm to pairs of balls for which the values of both S(i,j) and TT(i,j) were small.

Protein 1 = red surface (matching atoms) and white atoms (unmatched atoms)

Protein 2 = tan surface (matching atoms) and blue atoms (unmatched atoms)

The final root mean square errors were around 0.6 for each of these three final ‘solutions’ to the problem.

Conclusions

We set out to tackle the problem of looking for regions of the same shape on the surfaces of proteins. These regions might have biological significance (active sites).

The MCMC algorithm worked well to compare unlabelled point sets of up to a couple of hundred atoms. However, the protein surfaces are large (the surfaces of the proteins we used each contained about 12000 atoms).

Given an atom i in protein 1 and atom j in protein 2, consider the balls of points centered on those atoms. We used the HPC to calculate some simple statistics (TT(i,j) and S(i,j)) to help look for regions that might be the same shape. We then applied the MCMC algorithm to find optimal matches in those regions.