Multiple Structure Alignment

1. What is the meaning of “flexible” in the paper by Ye and Godzik, “Multiple flexible structure alignment using partial order graphs”?

2. What is the multiple alignment strategy (inspired from multiple sequence alignment heuristics) that is employed by the POSA algorithm described in the paper by Ye and Godzik, “Multiple flexible structure alignment using partial order graphs”?

3. What is the accuracy measure employed by CBA algorithm when assessing the accuracy of the generated alignments with respect to a ground truth alignment?

Protein structures

Protein structure databases

- PDB
- 3D structures
- SCOP
  - Murzin, Brenner, Hubbard, Chothia
  - Classification
    - Class (mostly alpha, mostly beta, alpha/beta (interspersed), alpha+beta (segregated), multi-domain, membrane)
    - Fold (similar structure)
    - Superfamily (homology, distant sequence similarity)
    - Family (homology and close sequence similarity)

Protein structure comparison

- Levels of structure description
  - Atom/atom group
  - Residue
  - Fragment
  - Secondary structure element (SSE)
- Basis of comparison
  - Geometry/architecture of coordinates/relative positions
  - Sequential order of residues along backbone, …
  - Physio-chemical properties of residues, …

How to compare?

- Key problem: find an optimal correspondence between the arrangements of atoms in two molecular structures (say A and B) in order to align them in 3D
- Optimality of the alignment is determined using a root mean square measure of the distances between corresponding atoms in the two molecules
- Complication: It is not known a priori which atom in molecule B corresponds to a given atom in molecule A (the two molecules may not even have the same number of atoms)
**Root Mean Square Deviation (RMSD)**

\[ RMSD = \sqrt{\frac{\sum d_i^2}{n}} \]

- \( n \) = number of atoms
- \( d_i \) = distance between 2 corresponding atoms \( i \) in 2 structures

**Unit of RMSD** => e.g. Ångstroms
- identical structures => RMSD = “0”
- similar structures => RMSD is small (1 – 3 Å)
- distant structures => RMSD > 3 Å

**Multiple Structure Alignment**

- The idea is similar to Multiple Sequence Alignment:
  - Find regions that are conserved among a set of input proteins
- The difference:
  - We do not use sequence information but atomic coordinate positions (3D structures of proteins) to determine conserved regions

**Pairwise vs. Multiple and Sequence vs. Structure**

- Optimum pairwise sequence alignment can be found in \( O(n^2) \) time.
- Multiple sequence alignment is exponential time.
- Pairwise structure alignment problem is NP-complete (Lathrop, 1994)
- So, Multiple Structure Alignment is a difficult problem
**Serpins**

First 6 molecules (core highlighted in red) 11 molecules Core alone of 11 molecules

Multiple structure alignment result of MUSTA algorithm

**Globins**

Running time = 1min (average)

Multiple structure alignment result of MUSTA algorithm

**Cal-binding**

Running time = 8 sec

Multiple structure alignment result of MUSTA algorithm

**Reading 1**

- Multiple flexible structure alignment using partial order graphs by Ye and Godzik, Bioinformatics 2005.
- Progressive multiple alignment guided by a guide tree
- Information on unaligned regions are kept and displayed as a partial order graph with the final alignment

**Flexible vs. rigid body alignment**

- Rigid body alignment:
  - Proteins treated as static rigid 3D objects
  - If we are to report a final single alignment, combining two local alignments may produce inconsistencies when two proteins are superpositioned using a single rotation/translation matrix
- Flexible: allow certain parts of the protein structure to twist (rotate and translate) so that we get a better alignment of matched parts

**Progressive alignment**
**Partial order graphs**

- We can have both
  
  \[
  \begin{align*}
  &ab-de- \quad a-bd-e \\
  &\quad a-cd-f \quad \text{and} \quad \quad ac-df-
  \end{align*}
  \]

  No order defined between b and c (and e and f)

**The algorithm**

- Start with flexible pairwise alignments (FATCAT)
  
  - Each pairwise alignment is going to induce a partial ordered graph (POG)
  
  - At each step combine the previous alignments (represented by POGs) using dynamic programming
    
    - Consider all the combinatorial pairings of Aligned Fragment Pairs (AFP)
    
    - Do not consider flexibility of the alignment when combining POGs. The flexibility is considered in the final stage

**Results**

- Calmodulin proteins with two calcium binding states

- tRNA synthetases

- Rossmann fold

**Discussion**

- Questions? Unclear points to discuss?
- Pros and Cons?
- Quantitative results?
- Improvements?
- Validity of the tests?
**Reading 2**

- Development and validation of a consistency based multiple structure alignment algorithm by Ebert and Brutlag in Bioinformatics 2006.
- CBA: A seven stage iterative algorithm
- Uses progressive alignment strategy to combine pairwise alignments into a multiple structural alignment
- Similar to POSA, CBA uses a pairwise structural alignment algorithm (LOCK 2)

**Introduces a new benchmark dataset**

- Use conserved sequence motifs as gold standards
- From a non-redundant protein database (<40% sequence similarity) construct family and superfamily validation sets containing same PROSITE and eMOTIF patterns.
  - 478 family validation sets
  - 197 superfamily validation sets

**LOCK 2 alignment method**

- Aligns secondary structures first
- Then, residue alignment phase with a dynamic programming algorithm that scores residues pairs based on the distances between beta carbons and the angles between the Calpha-Cbeta vectors.

**The algorithm**

1. Run pairwise structural alignments (LOCK 2)
2. Superimpose structures
3. Extract new pairwise residue alignment (LOCK 2)
4. Cluster to obtain residue correspondences
5. Realign individual secondary structure elements
6. Cluster to obtain residue correspondences
7. Consistency-based alignment of local regions

**Step 1**

1. Run pairwise structural alignments (LOCK 2)

- Pairwise alignments with LOCK 2. (Rigid body alignments)
- Each alignment produces a set of aligned residue pairs and a rotation/translation matrix that superimposes the two protein structures aligned. The rotation/translation matrix is determined based on the aligned residue pairs.

**Step 2**

2. Superimpose structures

- Produce a multiple structure alignment using a progressive approach
- Build a guide tree using pairwise LOCK 2 scores
- When combining two alignments use the superimposition matrix of the pair of descendants which produce the highest alignment score
- All structures are transformed into the same frame of reference at the end
- Fixes proteins’ positions in 3d spaces for the following steps
Step 3

• The new superimposition obtained in step 2 changes relative positions of amino acids so, rerun the residue alignment phase of LOCK 2
• This will give a new set of all-to-all pairwise set of aligned residues

Step 4

• Obtain a consistent multiple alignment of residue pairs using the pairwise matches obtained in Step 3.
• Use Markov cluster algorithm (MCL) to determine a consistent group of matched residues (i.e., determine the columns of multiple alignment)

Step 5

• Use secondary structure information to realign the matched residue pairs to get a more biologically correct multiple alignment
• The realignment only changes the residue pairings, it does not change the 3D positions of proteins

Step 6

• Repeat step 4 and run the MCL algorithm to find multiply aligned residue groups.

Step 7

• The final stage is used to identify and realign small regions that could not be resolved in the previous steps.
  • Ordering inconsistencies
  • Gaps in secondary structures
  • Columns that contain more than one residue from the same protein

Results

<table>
<thead>
<tr>
<th>Method</th>
<th>CBA Method</th>
<th>CEMC Method</th>
<th>MultiProt Method</th>
<th>MASS Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>in (%)</td>
<td>in (%)</td>
<td>in (%)</td>
<td>in (%)</td>
<td>in (%)</td>
</tr>
<tr>
<td>cMOTIF families</td>
<td>96.2</td>
<td>93.4</td>
<td>75.1</td>
<td>87.7 (77.5)</td>
</tr>
<tr>
<td>eMOTIF families</td>
<td>94.8</td>
<td>87.7</td>
<td>47.3</td>
<td>84.9 (51.3)</td>
</tr>
<tr>
<td>PROSITE families</td>
<td>97.5</td>
<td>95.9</td>
<td>81.7</td>
<td>90.7 (72.9)</td>
</tr>
<tr>
<td>PROSITE superfamilies</td>
<td>95.4</td>
<td>89.4</td>
<td>55.3</td>
<td>87.8 (57.3)</td>
</tr>
</tbody>
</table>
Discussion

- Questions? Unclear points to discuss?
- Pros and Cons?
- Improvements?
- Validity of the tests?
  - Do you agree with the use of sequence motifs?

- No class next week.
- Ramazan Bayraminiz kutlu olsun.
- Think about your project ideas during the break and all of you should have finalized your project topics by November 1
  - You will e-mail me a short description of your project.
- You will have about 1.5 months until project presentations and 2 more weeks to finish project reports.