Algorithms for Protein Structure Analysis: Alignment and Classification

Sourangshu Bhattacharya
sourangshu@cse.iitkgp.ernet.in

Computer Science and Engineering,
IIT Kharagpur - 721302.
Outline

Protein Structure Alignment
   Introduction
   Matrix and Graph Representation
   Graph Matching
   A new method
   Results

Protein Sub-Structure Alignment
   Neighborhoods
   Results

Protein Structure Classification
   Preliminaries
   Kernels on Protein Structures
   Results

Learning with Uncertainty
   Resolution-aware Protein Structure Classification
Outline

Protein Structure Alignment
  Introduction
  Matrix and Graph Representation
  Graph Matching
  A new method
  Results

Protein Sub-Structure Alignment
  Neighborhoods
  Results

Protein Structure Classification
  Preliminaries
  Kernels on Protein Structures
  Results

Learning with Uncertainty
  Resolution-aware Protein Structure Classification
What is a Protein?

- Amino acids form peptide bonds to polymerize.
- Proteins are poly-peptide molecules.
- Represented by sequence of residues.
- Poly-peptide chains fold to form 3D structures.
Protein Structure

Myoglobin
(1DWT):
Tertiary Structure
Protein Structure

Simplification: $C^\alpha$ atoms and topology.
Loss:
- Side chain
- Secondary structure
Gain: Simplicity
Past uses: SSAP, DALI, CE, etc.
Problem with Topology: non-topological similarities are not detected.
New model: Pointset.
Gain: Generality (Active sites ?)
Past uses: $C^\alpha$ match.
A protein structure $X$ having $n$ residues is represented as

$$X = \{x_1, \ldots, x_n\} \text{ where } x_i \in \mathbb{R}^3, 1 \leq i \leq n.$$ 

Each $x_i$ gives position of $C^\alpha$ atom of the $i^{th}$ residue with respect to some arbitrary coordinate system.
Structural Alignment

Alignment between two proteins, 2PEL and 5CNA, showing circular permutations.
Alignment is defined by a set of *equivalences*.
Optimal superposition can be calculated easily.
Structural Alignment

A structural alignment between two proteins $X^A$ and $X^B$ is a 1-1 mapping $\phi : \{i | x^A_i \in \bar{X}^A\} \rightarrow \{j | x^B_j \in \bar{X}^B\}$, where $\bar{X}^A \subseteq X^A$ and $\bar{X}^B \subseteq X^B$. 
A structural alignment between two proteins $X^A$ and $X^B$ is a 1-1 mapping $\phi: \{i|\mathbf{x}^A_i \in \bar{X}^A\} \rightarrow \{j|\mathbf{x}^B_j \in \bar{X}^B\}$, where $\bar{X}^A \subseteq X^A$ and $\bar{X}^B \subseteq X^B$.

Root Mean Square Deviation

$$RMSD(\phi) = \sqrt{\frac{1}{|\bar{X}^A|} \sum_{(i,j) \in \Phi} (\mathbf{x}^A_i - T(\mathbf{x}^B_j))^2}$$

where $T$ is the optimal transformation.

Can we use this as a score function?
Structural Alignment

A structural alignment between two proteins $X^A$ and $X^B$ is a 1-1 mapping $\phi : \{i | x_i^A \in \overline{X}^A\} \rightarrow \{j | x_j^B \in \overline{X}^B\}$, where $\overline{X}^A \subseteq X^A$ and $\overline{X}^B \subseteq X^B$.

Root Mean Square Deviation

$$\text{RMSD}(\phi) = \sqrt{\frac{1}{|\overline{X}^A|} \sum_{(i,j) \in \Phi} (x_i^A - T(x_j^B))^2}$$

where $T$ is the optimal transformation.

Problem: Both $\phi$ and $T$ are unknown and interdependent.
Another Score

Distance Root Mean Square Deviation

\[
RMSD_D(\phi) = \sqrt{\frac{1}{|P^A|^2} \sum_{x_i^A, x_j^A \in \tilde{X}^A} (d_{ij}^A - d_{\phi(i)\phi(j)}^B)^2}
\]

where, \(d_{ij}^A\) is the distance between residues \(x_i^A\) and \(x_j^A\).
Graph and Distance Matrix

Protein Structure Graph

Residues

D(i,j)

Residues

Distance / Adjacency Matrix
# Graph and Distance Matrix

## DALI Scoring function

Known: Neighboring residues interact with greater force than far away ones.

\[
S_{DALI}(\phi) = \sum_{x_i^A, x_j^A \in \bar{X}^A} \left( 0.2 - \frac{|d_{ij}^A - d_{\phi(i)\phi(j)}^B|}{\bar{d}_{ij}} \right) \exp \left( - \left( \frac{\bar{d}_{ij}}{20} \right)^2 \right)
\]

Maximize \( S_{DALI} \) over all \( \phi \).

DALI uses heuristics which degrade it’s performance. Also, not amenable to theoretical analysis.

## Observation from DALI score

Neighboring residues affect the score more than far away ones. So, use nearness instead of distance function.
Graph and Distance Matrix

Nearness matrix

The adjacency or nearness matrix $A$ of a given protein $X = \{x_1, \ldots, x_n\}$ is defined as:

$$A_{ij} = e^{-\frac{d_{ij}}{\alpha}}, \; \alpha > 0$$

- An exponentially decreasing function of $d$ between 0 and 1.
- A continuous and invertible function.

Scoring function

$$S(\phi) = \sum_{x_i^A, x_j^A \in \bar{X}^A} T - (A^A_{ij} - A^B_{\phi(i)\phi(j)})^2$$

Maximize $S(\phi)$ over all $\phi$. $T$ is a known threshold.
Scoring function

\[ S(\phi) = \sum_{x_i^A, x_j^A \in \bar{X}^A} T - (A_{ij}^A - A_{\phi(i)\phi(j)}^B)^2 \]

Maximize \( S(\phi) \) over all \( \phi \). \( T \) is a known threshold.

Graph Matching

Given two weighted graphs \( G^A \) and \( G^B \), find their maximal subgraphs \( \bar{G}^A \) and \( \bar{G}^B \) and a mapping \( \phi \) between vertices of \( \bar{G}^A \) and \( \bar{G}^B \) such that

\[ |A_{ij}^A - A_{\phi(i)\phi(j)}^B| < T, \ i, j \in \bar{G}^A \]
Graph Matching
Graph Matching

Given two weighted graphs $\mathcal{G}^A$ and $\mathcal{G}^B$, find their maximal subgraphs $\overline{\mathcal{G}}^A$ and $\overline{\mathcal{G}}^B$ and a mapping $\phi$ between vertices of $\overline{\mathcal{G}}^A$ and $\overline{\mathcal{G}}^B$ such that

$$|A_{ij}^A - A_{\phi(i)\phi(j)}^B| < T, \ i, j \in \overline{\mathcal{G}}^A$$

Intractable

This is the optimization version of the well known NP-Hard problem *subgraph isomorphism*. Thus a polynomial time algorithm to find an exact solution of this problem does not exist unless $P = NP$. 
Graph Matching

Assumption

Two structures have same number of residues, and all of them are aligned.

Weighted Graph Matching (Umeyama 88)[4]

\[ S(P) = \| PA^T P - A^B \|^2 \]

Minimize \( S(P) \) over all permutation matrices \( P \).
Motivation (Umeyama 88)[4]

**Theorem 1** Let $A^A$ and $A^B$ be full rank adjacency matrices, with eigenvalue decompositions

$$A^A = U^A \Lambda^A U^{AT}$$
$$A^B = U^B \Lambda^B U^{BT}$$

$Q = U^B S U^{AT}$ minimizes $\|Q A^A Q^T - A^B\|^2$ for all orthogonal matrices $Q$. Here $S \in S = \{\text{diag}(s_1, \ldots, s_n) | s_i = 1 \text{ or } -1\}$.

**Theorem 2** Let $\tilde{U}^A$ and $\tilde{U}^B$ be matrices having absolute values of the entries in matrices $U^A$ and $U^B$. Let $\hat{P}$ be the optimal permutation matrix in the case of a perfect match, then $\hat{P}$ maximizes

$$\text{tr}(P^T \tilde{U}^B \tilde{U}^{AT})$$
Corollary

Permutation $\hat{\pi}$ corresponding to $\hat{P}$ can be obtained by:

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} \| (\bar{U}^A)_i - (\bar{U}^B)_{\pi(i)} \|^2$$

where $(A)_i$ is the $i^{th}$ row of matrix $A$. 
Neighborhood Preserving Projections

Projection

We are interested in projecting the residues on real line such that neighborhoods are preserved optimally.

\[
\max_{\mathbf{f} \in \mathbb{R}^n} \sum_{i=1}^{n} \sum_{j=1}^{n} [A_{ij}(f_i + f_j)^2 - A_{ij}(f_i - f_j)^2]
\]

Observations

- Second term: \(|f_i - f_j|\) low whenever \(A_{ij}\) is high.
- First term: \(|f_i + f_j|\) high whenever \(A_{ij}\) is low. So, \(f_i\) and \(f_j\) should be far apart.
- Unbounded solution. Constrain by adding \(\|\mathbf{f}\|^2 = n\).
Neighborhood Preserving Projections

Final formulation

\[
\max_{\mathbf{f} \in \mathbb{R}^n} \mathbf{f}^T \mathbf{A} \mathbf{f}
\]

Subject to

\[\|\mathbf{f}\|^2 = n\]

This is same as finding the eigenvector corresponding to maximum eigenvalue of the matrix \( \mathbf{A} \).

Absolute Value

If \( \mathbf{f} \) is a eigenvector, so is \(-\mathbf{f}\). Thus, we define \textit{neighborhood preserving projections}, \( f_i \) as \( |f_i^*| \).
## Scoring function

### Similarity score

Given two proteins $X^A$ and $X^B$, and their neighborhood preserving projections $f^A$ and $f^B$, we define the similarity between residue $i$ of $X^A$ and residue $j$ of $X^B$ as:

$$s(i, j) = T - (f^A_i - f^B_j)^2$$

The similarity score of an alignment $\phi$ is:

$$S(\phi) = \sum_{x_i \in \bar{X}^A} s(i, \phi(i))$$

Maximize $S(\phi)$ w.r.t. $\phi$. 
Spectral Similarity

By considering only the leading eigenvector, the spectral similarity score becomes:

$$\min_{\pi \in \Pi} \sum_{i=1}^{n} ((\bar{U}_G^1)_i - (\bar{U}_H^1)_{\pi(i)})^2$$

or

$$\max_{\pi \in \Pi} \sum_{i=1}^{n} T - ((\bar{U}_G^1)_i - (\bar{U}_H^1)_{\pi(i)})^2$$
Connection

**Projection Similarity**

If all residues of the two proteins are aligned, i.e. $\bar{X}^A = X^A$ and $\bar{X}^B = X^B$, we solve,

$$\max_{\phi} \sum_{x_i \in X^A} T - (f_i^A - f_j^B)^2$$

**Unequal residues**

The above problem can be solved even in case of unequal number of residues in the two structures.
Greedy Fragment Pair Search

### Topology
- The above problem is an instance of *assignment problem*. We could solve it in polynomial time.
- But we use information in protein sequence to solve the problem more efficiently.

### Basic Idea
- The scoring function \( s(i, j) \) is analogous to the sequence similarity function.
- Use sequence alignment algorithms, e.g. local alignment algorithm.
**Greedy Fragment Pair Search**

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Initialize alignment to null.</td>
</tr>
<tr>
<td>2. Calculate the local alignment matrix of incremental fragment similarity.</td>
</tr>
<tr>
<td>3. Find the maximum element in the matrix and traceback to find the high scoring fragment pair.</td>
</tr>
<tr>
<td>4. Add the currently found fragment pair to the alignment and delete the rows and columns corresponding to the currently added residues from local alignment matrix.</td>
</tr>
<tr>
<td>5. Go to step 3. If no positive scoring entry is found, terminate.</td>
</tr>
</tbody>
</table>
## Benchmark Datasets

Comparison between Matchprot (MP) and DALI using benchmark datasets.

<table>
<thead>
<tr>
<th>Data set / Classification</th>
<th>Total Pairs</th>
<th>Better</th>
<th>Worse</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fischer</td>
<td>68</td>
<td>17</td>
<td>18</td>
<td>33</td>
</tr>
<tr>
<td>Novotny et. al.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.10.40</td>
<td>21</td>
<td>8</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>1.10.164</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1.25.30</td>
<td>21</td>
<td>3</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>2.30 110</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2.40.100</td>
<td>28</td>
<td>4</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>2.100.10</td>
<td>15</td>
<td>5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3.10.70</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3.40.91</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.70.10</td>
<td>15</td>
<td>1</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>2.40.20</td>
<td>21</td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

**Better:** MP has lower RMSD higher length of alignment (Lali).

**Worse:** DALI has lower RMSD higher Lali.

**Level:** MP has either both higher or lower RMSD and Lali than DALI.
Non-topological Similarities

Alignment between 2PEL and 5CNA showing circular permutation
Comparison with CE (Shindyalov and Bourne) [1]

Retrieval of domains having similar folds from ASTRAL 95% non-redundant dataset.

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Matchprot</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(TP/FP/prec./rec.)</td>
<td>(TP/FP/prec./rec.)</td>
</tr>
<tr>
<td>d101m__</td>
<td>93 / 0 / 1 / 0.95</td>
<td>96 / 2 / 0.97 / 0.99</td>
</tr>
<tr>
<td>d1htia__</td>
<td>272 / 56 / 0.82 / 0.83</td>
<td>307 / 29 / 0.91 / 0.93</td>
</tr>
<tr>
<td>d1jzba__</td>
<td>23 / 0 / 1 / 0.1</td>
<td>33 / 270 / 0.1 / 0.14</td>
</tr>
<tr>
<td>d2pela__</td>
<td>70 / 50 / 0.58 / 0.8</td>
<td>61 / 36 / 0.62 / 0.70</td>
</tr>
<tr>
<td>d7rsa__</td>
<td>18 / 0 / 1 / 1</td>
<td>17 / 1 / 0.94 / 0.94</td>
</tr>
</tbody>
</table>

TP: True positive
FP: False positive

\[
\text{prec} = \frac{TP}{TP + FP}
\]

\[
\text{rec} = \frac{TP}{\text{Actual}}
\]
Time Comparison
Summary

- Fast $O(n^3)$ deterministic algorithm for comparing protein structure.
- New score function using neighborhood preserving projections.
- State of the art performance for structure retrieval on SCOP.
Outline

Protein Structure Alignment
  Introduction
  Matrix and Graph Representation
  Graph Matching
  A new method
  Results

Protein Sub-Structure Alignment
  Neighborhoods
  Results

Protein Structure Classification
  Preliminaries
  Kernels on Protein Structures
  Results

Learning with Uncertainty
  Resolution-aware Protein Structure Classification
Problem of Indels

Problem

- The above algorithm was designed for similarly sized proteins.
- It still works for many cases with up to 40% indels.
- However, it gives wrong answers for proteins having higher indels (Roughly half of the residues are absent in the other protein).

Main Idea

Align conserved substructures called *neighborhoods*, and “grow” neighborhood alignments to entire structure.
## Neighborhoods

### Observation

Spatial neighborhoods are more preserved even in evolutionarily distant proteins.

### Reasons

- The site crucial for functioning remains structurally preserved.
- Many a times, additions are in terms of separate domains.

### Solution

- Compare spatial neighborhoods instead of entire structures using spectral method.
- “Grow” the neighborhood alignments to get a good overall alignment.
## Neighborhoods

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The $k$-structure neighborhood</strong> of a residue of a protein is defined as the set of $k$ residues nearest to the given residue in 3D.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The $k$-sequence neighborhood</strong>, $N_{\text{seq}}^A(i)$ starting from residue $i$ of structure $A$ is defined as $N_{\text{seq}}^A(i) = {x_i, \ldots, x_{i+k-1}}$.</td>
</tr>
</tbody>
</table>
Neighborhoods

Sequence Neighborhood

Structure Neighborhood

Myoglobin (1DWT)
# Alignment using Neighborhoods

## Overall Scheme

1. Calculate a spanning set of neighborhoods.
2. Align all pairs of neighborhoods.
3. Grow neighborhood alignments to entire structure.

## Spanning set of Neighborhoods

- Set of neighborhoods should span the entire protein, and should not be very high.
- For structure neighborhoods, choose one around every residue.
- For sequence neighborhoods, choose one starting at every residue.
Alignment using Neighborhoods

### Neighborhood Alignment

- For sequence neighborhoods, use the spectral algorithm developed above.
- For structure neighborhoods, solve *maximal common subgraph*.
- Restrict sizes of structure neighborhoods.

### Growing Neighborhood Alignments

- Calculate optimal transformation based on neighborhood alignment.
- Re-calculate similarity measure based using transformed coordinates.
- Calculate final alignment using revised similarity measure.
Algorithm

Protein A
- Calculate list of spanning neighborhoods

Protein B
- Calculate list of spanning neighborhoods

For each pair of neighborhoods:
  - Calculate neighborhood alignments
  - Calculate the optimal transformation for neighborhood alignments and store the transformations in a list

Cluster Transformations

For each of the mean transformations:
  - Calculate new similarity score between residues based on transformations
  - Calculate alignment between structures based on new similarity score

Report best alignment as final
Comparison with existing methods: Difficult Cases

10 difficult pairs mentioned in (Shindyalov and Bourne)[1]

<table>
<thead>
<tr>
<th>PDBid1(size) - PDBid2(size)</th>
<th>Seq Nbhd LAli / RMS</th>
<th>Struct Nbhd LAli / RMS</th>
<th>DALI Len / RMS</th>
<th>CE Len / RMS</th>
<th>SSM Len / RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1fxiA(96) - 1ubq(76)</td>
<td>54 / 2.18</td>
<td>56 / 2.16</td>
<td>60 / 2.6</td>
<td>100 / 3.82</td>
<td>60 / 2.86</td>
</tr>
<tr>
<td>1ten(90) - 3hhrB(185)</td>
<td>84 / 1.58</td>
<td>82 / 1.39</td>
<td>86 / 1.9</td>
<td>87 / 1.90</td>
<td>73 / 2.09</td>
</tr>
<tr>
<td>3hlaB(270) - 2rhe(114)</td>
<td>70 / 2.26</td>
<td>68 / 2.26</td>
<td>75 / 3</td>
<td>85 / 3.46</td>
<td>78 / 3.08</td>
</tr>
<tr>
<td>2azaA(129) - 1paz(120)</td>
<td>72 / 2.46</td>
<td>79 / 2.20</td>
<td>81 / 2.5</td>
<td>85 / 2.90</td>
<td>79 / 2.41</td>
</tr>
<tr>
<td>1cewl(108) - 1molA(94)</td>
<td>68 / 1.80</td>
<td>79 / 1.94</td>
<td>81 / 2.3</td>
<td>81 / 2.34</td>
<td>79 / 2.12</td>
</tr>
<tr>
<td>1cid(177) - 2rhe(114)</td>
<td>91 / 2.05</td>
<td>91 / 2.06</td>
<td>97 / 3.2</td>
<td>98 / 2.97</td>
<td>89 / 2.32</td>
</tr>
<tr>
<td>1crl(534) - 1ede(310)</td>
<td>160 / 2.50</td>
<td>174 / 2.49</td>
<td>211 / 3.5</td>
<td>220 / 3.91</td>
<td>188 / 3.81</td>
</tr>
<tr>
<td>2sim(381) - 1nsbA(390)</td>
<td>262 / 2.72</td>
<td>262 / 2.63</td>
<td>222 / 3.8</td>
<td>276 / 2.99</td>
<td>271 / 2.86</td>
</tr>
<tr>
<td>1bgeB(159) - 2gmfA(121)</td>
<td>85 / 2.48</td>
<td>87 / 2.22</td>
<td>94 / 3.3</td>
<td>102 / 4.02</td>
<td>44 / 2.49</td>
</tr>
<tr>
<td>1tie(166) - 4fgf(124)</td>
<td>105 / 2.20</td>
<td>106 / 2.27</td>
<td>114 / 3.1</td>
<td>115 / 2.86</td>
<td>114 / 2.85</td>
</tr>
</tbody>
</table>
## Overall Results on Benchmark Datasets

### Comparison with DALI [3]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Better / Worse / Level</td>
<td>Better / Worse / Level</td>
</tr>
<tr>
<td>Fischer's</td>
<td>4 / 4 / 60</td>
<td>5 / 2 / 61</td>
</tr>
<tr>
<td>Novotny’s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.10.164</td>
<td>1 / 0 / 9</td>
<td>2 / 0 / 8</td>
</tr>
<tr>
<td>1.10.40</td>
<td>11 / 0 / 10</td>
<td>5 / 0 / 16</td>
</tr>
<tr>
<td>1.25.30</td>
<td>10 / 0 / 11</td>
<td>5 / 0 / 16</td>
</tr>
<tr>
<td>2.30.110</td>
<td>0 / 0 / 6</td>
<td>0 / 0 / 6</td>
</tr>
<tr>
<td>2.40.100</td>
<td>0 / 0 / 28</td>
<td>0 / 0 / 28</td>
</tr>
<tr>
<td>2.100.10</td>
<td>5 / 3 / 7</td>
<td>5 / 0 / 10</td>
</tr>
<tr>
<td>3.10.70</td>
<td>0 / 0 / 10</td>
<td>0 / 0 / 10</td>
</tr>
<tr>
<td>3.40.91</td>
<td>0 / 0 / 6</td>
<td>0 / 0 / 6</td>
</tr>
<tr>
<td>3.70.10</td>
<td>0 / 0 / 15</td>
<td>2 / 0 / 13</td>
</tr>
<tr>
<td>2.40.20</td>
<td>0 / 3 / 18</td>
<td>0 / 0 / 21</td>
</tr>
</tbody>
</table>
## Overall Results on Benchmark Datasets

### Comparison with CE [1]

<table>
<thead>
<tr>
<th>Data set/ classifn.</th>
<th>Align. sequence nbhd. Better / Worse / Level</th>
<th>Align. structure nbhd. Better / Worse / Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fischer’s</td>
<td>2 / 1 / 65</td>
<td>2 / 0 / 66</td>
</tr>
<tr>
<td>Novotny’s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.10.164</td>
<td>0 / 0 / 10</td>
<td>0 / 0 / 10</td>
</tr>
<tr>
<td>1.10.40</td>
<td>0 / 0 / 21</td>
<td>0 / 0 / 21</td>
</tr>
<tr>
<td>1.25.30</td>
<td>1 / 0 / 20</td>
<td>0 / 0 / 21</td>
</tr>
<tr>
<td>2.30.110</td>
<td>0 / 0 / 6</td>
<td>0 / 0 / 6</td>
</tr>
<tr>
<td>2.40.100</td>
<td>6 / 0 / 22</td>
<td>4 / 0 / 24</td>
</tr>
<tr>
<td>2.100.10</td>
<td>4 / 0 / 11</td>
<td>4 / 0 / 11</td>
</tr>
<tr>
<td>3.10.70</td>
<td>1 / 0 / 9</td>
<td>1 / 0 / 9</td>
</tr>
<tr>
<td>3.40.91</td>
<td>0 / 0 / 6</td>
<td>0 / 0 / 6</td>
</tr>
<tr>
<td>3.70.10</td>
<td>0 / 0 / 15</td>
<td>0 / 1 / 14</td>
</tr>
<tr>
<td>2.40.20</td>
<td>1 / 1 / 19</td>
<td>0 / 0 / 21</td>
</tr>
</tbody>
</table>
### Overall Results on Benchmark Datasets

#### Comparison with SSM

<table>
<thead>
<tr>
<th>Data set/ classifn.</th>
<th>Align. sequence nbhd. Better / Worse / Level</th>
<th>Align. structure nbhd. Better / Worse / Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fischer's</td>
<td>13 / 10 / 45</td>
<td>23 / 5 / 40</td>
</tr>
<tr>
<td>Novotny's</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.10.164</td>
<td>3 / 1 / 6</td>
<td>4 / 0 / 6</td>
</tr>
<tr>
<td>1.10.40</td>
<td>9 / 0 / 12</td>
<td>8 / 0 / 13</td>
</tr>
<tr>
<td>1.25.30</td>
<td>9 / 0 / 12</td>
<td>3 / 0 / 18</td>
</tr>
<tr>
<td>2.30.110</td>
<td>1 / 1 / 4</td>
<td>1 / 1 / 4</td>
</tr>
<tr>
<td>2.40.100</td>
<td>1 / 0 / 27</td>
<td>2 / 1 / 25</td>
</tr>
<tr>
<td>2.100.10</td>
<td>1 / 4 / 10</td>
<td>3 / 1 / 11</td>
</tr>
<tr>
<td>3.10.70</td>
<td>2 / 0 / 8</td>
<td>3 / 0 / 7</td>
</tr>
<tr>
<td>3.40.91</td>
<td>2 / 0 / 4</td>
<td>1 / 0 / 5</td>
</tr>
<tr>
<td>3.70.10</td>
<td>0 / 1 / 14</td>
<td>1 / 2 / 12</td>
</tr>
<tr>
<td>2.40.20</td>
<td>0 / 6 / 15</td>
<td>3 / 2 / 16</td>
</tr>
</tbody>
</table>
## Structure Retrieval

### 5 SCOP Folds

<table>
<thead>
<tr>
<th>SCOPid (tot. num.)</th>
<th>Method</th>
<th>cutoff (% / Z)</th>
<th>True +ve</th>
<th>False +ve</th>
</tr>
</thead>
<tbody>
<tr>
<td>d101m__ (37)</td>
<td>seq nbhd</td>
<td>50% 45% 4.0 CE</td>
<td>34 35 35</td>
<td>9 38 95</td>
</tr>
<tr>
<td>d1htia_ (253)</td>
<td>seq nbhd</td>
<td>50% 45% 4.0 CE</td>
<td>190 231 233</td>
<td>4 13 224</td>
</tr>
<tr>
<td>d1jzba__ (119)</td>
<td>seq nbhd</td>
<td>50% 45% 4.0 CE</td>
<td>28 48 2</td>
<td>56 172 0</td>
</tr>
<tr>
<td>d2pela_ (48)</td>
<td>seq nbhd</td>
<td>50% 45% 4.0 CE</td>
<td>41 45 36</td>
<td>9 21 8</td>
</tr>
<tr>
<td>d7rsa__ (4)</td>
<td>seq nbhd</td>
<td>50% 45% 4.0 CE</td>
<td>4 4 4</td>
<td>0 13 0</td>
</tr>
</tbody>
</table>
Summary

- A robust algorithm for protein structure alignment.
- Idea of neighborhood alignments and growing of neighborhood alignments to entire structures.
- Outperformed state of the art techniques on benchmark datasets.
Outline

Protein Structure Alignment
  Introduction
  Matrix and Graph Representation
  Graph Matching
  A new method
  Results

Protein Sub-Structure Alignment
  Neighborhoods
  Results

Protein Structure Classification
  Preliminaries
  Kernels on Protein Structures
  Results

Learning with Uncertainty
  Resolution-aware Protein Structure Classification
Automatic Structure Classification

Problem
Classify given protein structures into SCOP superfamilies.

Approach
Define kernels on protein structures and use kernel methods.

Motivation
- Support vector machines (SVMs) are one of the most popular classifiers.
- SVMs cannot be directly used with protein structures.
- Kernels on protein structures will allow SVMs and many other methods to be applied.
Kernel Methods

Definition

A kernel $K$ on a set $\mathcal{X}$ is a real valued function on $\mathcal{X} \times \mathcal{X}$ satisfying the following properties:

- $K(x, y) = K(y, x)$ (Symmetric)
- $K(x, x) \geq 0$, and 0 only if $x = 0$
- $\sum_{i,j} c_i c_j K(x_i, x_j) \geq 0 \forall c_i, c_j \in \mathbb{R}$ (Positive semidefinite)

RKHS

Kernels can be thought of as dot products in a higher dimensional space called reproducing kernel hilbert space (RKHS).
Kernel Methods

Input space

Kernel function

RKHS

No Space

Mapping of data to a suitable space using kernel functions
Kernel Methods

### Geometry
- Kernel functions define a geometry in the RKHS.
- Angles can be measured using the kernels.
- Distances can be defined as
  \[ d(x_i, x_j) = \sqrt{\mathcal{K}(x_i, x_i) + \mathcal{K}(x_j, x_j) - 2 \times \mathcal{K}(x_i, x_j)}. \]

### Kernelized Algorithms
Many machine learning techniques can be modified to be used with kernels rather than vectorial data.
- Support Vector Machines.
- K-means clustering.
- Gaussian process regression, Principal component analysis, etc.
Building New kernels

If $k_1(x, y)$ and $k_2(x, y)$ are two valid kernels, then the following kernels are valid:

- **Linear Combination:**
  \[ k(x, y) = c_1 k_1(x, y) + c_2 k_2(x, y) \]

- **Exponentiation:**
  \[ k(x, y) = \exp(k_1(x, y)) \]

- **Product:**
  \[ k(x, y) = k_1(x, y) k_2(x, y) \]

- **Polynomial Transformation:**
  \[ k(x, y) = Q(k_1(x, y)) \]

- **Function product:**
  \[ k(x, y) = f(x) k_1(x, y) f(y) \]
### Motivation

- Many types of data processed by learning algorithms cannot be naturally represented as vectors.
- Kernelized learning algorithms can be used, if appropriate kernels are defined on those data.

### Examples

- Strings, trees, graphs, etc.
- Protein structures.
## Kernels on Structured Data

### Intuition

- Kernels can be thought of as similarity measures since $d(x_i, x_j)$ is a decreasing function of $K(x_i, x_j)$.
- Define similarity measures on structured data satisfying properties of kernels.
- Generally, positive-semidefiniteness is most difficult to ensure.

### Other Kernels on Proteins

- Graph Kernels.
- Sequence based kernels.
- Alignment Kernels using empirical kernel maps.
## Scheme

<table>
<thead>
<tr>
<th>Problem</th>
<th>Define kernels capturing similarity between protein structures.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideas</td>
<td>▶ Kernels should capture the notion of structural alignment.</td>
</tr>
<tr>
<td></td>
<td>▶ Define kernels on neighborhoods and extend them to entire protein structures.</td>
</tr>
</tbody>
</table>
### Kernels on Neighborhoods

**Convolution Kernels (Haussler 99)**

- $x \in X$ is a composite object, parts from $X_1, \ldots, X_m$.
- $R$ is a relation over $X_1 \times \cdots \times X_m \times X$ such that $R(x_1, \ldots, x_m, x)$ is true if $x$ is composed of $x_1, \ldots, x_m$.
- $K^1, \ldots, K^m$ be kernels on $X_1, \ldots, X_m$, respectively.

It can be showed that $K$ is a kernel on $X$.

$$K(x, y) = \sum_{(x_1, \ldots, x_m) \in R^{-1}(x), (y_1, \ldots, y_m) \in R^{-1}(y)} \prod_{i=1}^{m} K^i(x_i, y_i)$$

where $R^{-1}(x) = (x_1, \ldots, x_m) \in X_1 \times \cdots \times X_m | R(x_1, \ldots, x_m, x) = \text{true}$.
## Kernels on Neighborhoods

### Spectral Kernel

- $X$ is set of all neighborhoods.
- $X_1, \ldots, X_m$ are sets of residues.
- $R(x_1, \ldots, x_m, N)$ is true if $\{x_1, \ldots, x_m\} \in N$.
- $K^1, \ldots, K^m$ are RBF kernels comparing spectral projections.

Spectral kernel is defined as:

$$K_{SS}(N_i, N_j) = \sum_{\pi \in \Pi} e^{-\frac{||f^i - \pi(f^j)||^2}{\beta}}$$
Pairwise Distance Kernel

- $X$ is set of all neighborhoods.
- $X_1, \ldots, X_m$ are sets of all pairs of residues.
- $R(d_1, \ldots, d_m, N)$ is true if $d_1, \ldots, d_m$ are pairwise distances in $N$.
- $K^1, \ldots, K^m$ are RBF kernels comparing pairwise distances.

Pairwise distance kernel is defined as:

$$K_{PDS}(N_i, N_j) = \sum_{\pi \in \Pi} e^{-\frac{||d_i - \pi(d_j)||^2}{\sigma^2}}$$
Connection with Spectral Score

**Theorem**

Let $N_i$ and $N_j$ be two sub-structures with spectral projection vectors $f^i$ and $f^j$. Let $S(N_i, N_j)$ be the score of alignment of $N_i$ and $N_j$, obtained by solving assignment problem. For large enough value of $T$ such that all residues are matched.

$$\lim_{\beta \to 0} \mathcal{K}_{SS}(N_i, N_j)^{\beta} = e^{S(N_i, N_j) - kT}$$

**Non-psd Kernel**

$$\mathcal{K}_{LSS}(N_1, N_2) = \lim_{\beta \to 0} (\mathcal{K}_{SS}(N_1, N_2))^{\beta}$$
Kernels on Protein Structures

Kernels on Structures

For a set of proteins $X^1, \ldots, X^n$, define kernels:

\[
K_1(X^i, X^j) = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} K_{SS}(N^i_a, N^j_b)
\]

\[
K_2(X^i, X^j) = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} K_{PDS}(N^i_a, N^j_b)
\]
Kernels on Protein Structures

More Accurate Kernels

\[
K_3(X^i, X^j) = \sum_{a,b=1}^{n_i} \sum_{c,d=1}^{n_j} K_{SS}(N^i_a, N^i_b) \times K_{SS}(N^i_c, N^i_d) \times K_{norm}((N^i_a, N^i_b), (N^i_c, N^i_d))
\]

\[
K_4(X^i, X^j) = \sum_{a,b=1}^{n_i} \sum_{c,d=1}^{n_j} K_{PDS}(N^i_a, N^i_b) \times K_{SS}(N^i_c, N^i_d) \times K_{norm}((N^i_a, N^i_b), (N^i_c, N^i_d))
\]

where, \( K_{norm}((N^i_a, N^i_b), (N^i_c, N^i_d)) = e^{- \frac{(\|x^i_a - x^i_b\| - \|x^i_c - x^i_d\|)^2}{\sigma^2}}. \)
Alignment Kernels

- Increase the accuracy of these kernels by using alignment information.
- Add neighborhood kernels for aligned residues:

\[
\mathcal{K}^{Al}_1(X^i, X^j; \phi_{ij}) = \sum_{a | x^i_a \in \bar{X}^i} \mathcal{K}_{SS}(N^i_a, N^j_{\phi_{ij}(a)})
\]

- \( \mathcal{K}^{Al}_2 \) and \( \mathcal{K}^{Al}_3 \) are defined analogously using \( \mathcal{K}_{LSS} \) and \( \mathcal{K}_{PDS} \).
### Kernels on Protein Structures

#### Alignment Kernels

- Make alignment kernels positive semidefinite:

\[
\mathcal{K}_{4Al}(P^i, P^j) = \begin{cases} 
\sum_{a \mid p^i_a \in \bar{P}^i} \mathcal{K}_{SS}(N^i_a, N^j_{\phi_{ij}(a)}) & \text{if } i \neq j \\
\sum_{b=1}^{M} \sum_{a \mid p^i_a \in \text{dom}(\phi_{ib})} \mathcal{K}_{SS}(N^i_a, N^i_{\phi_{ib}(a)}) & \text{if } i = j
\end{cases}
\]

- \(\mathcal{K}_{5Al} \) and \(\mathcal{K}_{6Al} \) are defined analogously using \(\mathcal{K}_{LSS} \) and \(\mathcal{K}_{PDS} \).
## Structure Kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Positive Acc.</th>
<th>Negative Acc.</th>
<th>Total Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>69.67%</td>
<td>54.87%</td>
<td>62.27%</td>
</tr>
<tr>
<td>$K_2$</td>
<td>68.73%</td>
<td>61.33%</td>
<td>65.03%</td>
</tr>
<tr>
<td>$K_3$</td>
<td>56.13%</td>
<td>54.93%</td>
<td>55.53%</td>
</tr>
<tr>
<td>$K_4$</td>
<td>64.00%</td>
<td>60.93%</td>
<td>62.45%</td>
</tr>
<tr>
<td>CE</td>
<td>96.47%</td>
<td>63.33%</td>
<td>79.90%</td>
</tr>
</tbody>
</table>
## Alignment Kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Positive Acc.</th>
<th>Negative Acc.</th>
<th>Total Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1^{A_l}$</td>
<td>74.33%</td>
<td>83.47%</td>
<td>78.90%</td>
</tr>
<tr>
<td>$K_2^{A_l}$</td>
<td>79.13%</td>
<td>86.47%</td>
<td>82.80%</td>
</tr>
<tr>
<td>$K_3^{A_l}$</td>
<td>73.87%</td>
<td>82.67%</td>
<td>78.27%</td>
</tr>
<tr>
<td>$K_4^{A_l}$</td>
<td>91.87%</td>
<td>75.93%</td>
<td><strong>83.90%</strong></td>
</tr>
<tr>
<td>$K_5^{A_l}$</td>
<td>80.67%</td>
<td>76.07%</td>
<td>78.37%</td>
</tr>
<tr>
<td>$K_6^{A_l}$</td>
<td>88.53%</td>
<td>80.20%</td>
<td><strong>84.37%</strong></td>
</tr>
<tr>
<td>CE (NN)</td>
<td>96.47%</td>
<td>63.33%</td>
<td>79.90%</td>
</tr>
</tbody>
</table>
Summary

Problem
Build a protein structure classifier which takes resolution information into account.

Motivation

▶ Coordinates of atoms in protein structures are resolved to a particular accuracy.
▶ For example 1biaa1: 2.3Å, 1repc1: 2.6Å
▶ RMSDs of alignment between proteins are sometimes lower than the resolution.
▶ Example: 1biaa1 - 1repc1: 2.2Å
▶ Kernel values are perturbed due to perturbation in structure within resolution.
SVM Classification with uncertain kernels

SVM dual form

\[
\max_{\alpha \in S, t} \alpha^\top e - \frac{1}{2} t \quad \text{s.t.} \quad \alpha^\top YK Y \alpha \leq t
\]

where \( S = \{ \alpha | 0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0 \} \) and \( Y = \text{diag}(y_i) \).

SVM chance constrained form

\[
\max_{t, \alpha \in S} \alpha^\top e - \frac{1}{2} t \\
\quad \text{s.t.} \quad \text{Prob} \left( \alpha^\top Y(\mathbf{K} + Z) Y \alpha \leq t \right) \geq 1 - \epsilon
\]

where \( Z \) is a matrix of random noise.
Gaussian Uncertainty

Theorem

- $Z$ is an $n \times n$ random matrix.
- $Z_{ij} \sim N(0, \sigma^2_{ij})$.
- $K = \bar{K} + Z$, where $\bar{K}$ is kernel matrix.

For such a $K$, the chance constraint in previous formulation is satisfied if the following holds.

$$\sum_{ij} y_i y_j \alpha_i \alpha_j \bar{K}_{ij} - \Phi^{-1}(\epsilon) \| \Sigma \ast (\alpha \alpha^\top) \|_F \leq t$$
### Theorem

- $Z$ be a $n \times n$ random matrix with $E(Z_{ij}) = 0$.
- $P(a_{ij} \leq Z_{ij} \leq b_{ij}) = 1$.
- $K = \overline{K} + Z$, where $\overline{K}$ is kernel matrix.

For such a $K$, the chance constraint in previous formulation is satisfied if the following holds.

$$
\sum_{ij} y_i y_j \alpha_i \alpha_j K_{ij} + \sqrt{2 \log(1/\epsilon)} \sqrt{\sum_{ij} \beta_{ij} \alpha_i^2 \alpha_j^2} \leq t
$$

where, $\beta_{ij} = l_{ij}^2 \gamma_{ij}^2$, $l_{ij} = \frac{b_{ij} - a_{ij}}{2}$, $\gamma_{ij}$ is a function of $a_{ij}$ and $b_{ij}$. 
Robust SVM

Deterministic Optimization Problem

The chance constraint program proposed earlier for learning SVMs with uncertain kernels can be posed as:

\[
\min_{t, \alpha \in S_n} \quad \frac{1}{2} t - \sum_i \alpha_i
\]

s.t. \[\sum_{ij} y_i y_j \alpha_i \alpha_j \overline{K}_{ij} + \kappa \sqrt{\sum_{ij} \beta_{ij} \alpha_i^2 \alpha_j^2} \leq t\]

where \(\kappa\) depends on \(\epsilon\). This problem is applicable for both Gaussian and interval uncertainties.
Solution of the above problem

- The solution method depends on the matrix $\beta = [\beta_{ij}]$.
- When $\beta$ is rank one, the solution boils down to solving SVM with modified kernel.
- When $\beta$ is PSD, the problem is a second order cone program (SOCP), and can be solved using SOCP solver.
- In the general case, the problem is non-convex and can be solved using a standard descent algorithm.
## Results

<table>
<thead>
<tr>
<th></th>
<th>RSVM</th>
<th>SVM</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QP</td>
<td>SOCP</td>
<td>QN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MajErr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA</td>
<td>72.67</td>
<td>73.56</td>
<td>82.78</td>
</tr>
<tr>
<td>F1</td>
<td>73.49</td>
<td>74.35</td>
<td>82.95</td>
</tr>
<tr>
<td>RobustErr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA</td>
<td>27.11</td>
<td>50.33</td>
<td>66.44</td>
</tr>
<tr>
<td>F1</td>
<td>26.81</td>
<td>50.28</td>
<td>66.36</td>
</tr>
<tr>
<td>NomErr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA</td>
<td>66.50</td>
<td>66.65</td>
<td>76.00</td>
</tr>
<tr>
<td>F1</td>
<td>65.13</td>
<td>65.16</td>
<td>75.80</td>
</tr>
</tbody>
</table>
Results

Observations

- Robust SVM performs better than nominal SVM on synthetic datasets generated using Gaussian, Uniform and Beta noise.
- Robust error RSVM-SOCP and RSVM-QN increases less rapidly than nominal SVM, as the uncertainty is increased.
- For resolution-aware protein structure classification, RSVM-QN outperforms nominal SVM, and SVM with multiple instance kernel on 15 SCOP superfamilies.
P E Bourne and I N Shindyalov. 
Protein structure alignment by incremental combinatorial extension of optimal path. 

David Haussler. 
Convolution kernels on discrete structures. 
Technical report, University of California, Santa Cruz, 1999.

Liisa Holm and Chris Sander. 
Protein structure comparison by alignment of distance matrices. 

Shinji Umeyama. 
An eigendecomposition approach to weighted graph matching problems. 
Publications

Discussed here


Thank you!

Questions?