MMRefiner
A Deep Learning Guided Protein Complex Refinement Protocol

MOHIMENUL KARIM
Protein Complex
Motivation

Experimental methods are expensive and time consuming

Computational methods can generate structures with deviation from the native

Orientation of the individual monomers are not always optimal

No deep learning method to refine the near native protein complexes
Method

A deep learning method to predict the refined protein complex structure

Predict the transformation matrix of individual chains
- Translation matrix
- Rotation matrix

Apply the transformation on the individual chains

Generate the refined protein complex

Considered only chain 2 in this study
Dataset

Dockground

35 complexes (deviation spanning from 1 Å to 6 Å)

10 decoys for each

Training: 1647 complexes

Test: 480 complexes

No overlap in training and test dataset
Features

Relative position (residue number/chain length)
Residue type (one hot encoded)
Secondary structure (three state; one hot encoded)
Solvent accessibility
Phi (both sine and cosine)
Psi (both sine and cosine)
Translation

\( t_i \in t_x, t_y, t_z \)

\[ s_{\text{score}} = \frac{1}{1 + \left( \frac{t_i}{d_0} \right)^2} \text{ where, } t_i \in t_x, t_y, t_z \text{ and } d_0 = 10 \quad \text{........................................} (1) \]

*If* \( t_i < 0 \), *then* \( s_{\text{score}} \) **is negative**

After prediction

Calculate \( t_i \) using equation 1

*If* \( s_{\text{score}} \) **is negative**, *then* \( t_i \) **is negative**
Rotation

Calculate the angles of rotation along x, y and z axis from the rotation matrix.

Predict sine and cosine of the angles of rotation.

Calculate the angles back from the prediction using the following equation:

\[ \theta = \tan^{-1} \left( \frac{\sin \theta}{\cos \theta} \right) \]

Calculate the rotation matrix from the angles.
Architecture 1

Conv2D (input channels=60, output channels=128)
ReLU
Max pool (2x2)
Conv2D (input channels=128, output channels=256)
ReLU
Max pool (2x2)
Fully connected layer (256)
Output layer (3 for translation and 6 for rotation)
Architecture 2

Conv2D (input channels=60, output channels=128)
ReLU
Max pool (2x2)
Conv2D (input channels=128, output channels=256)
ReLU
Max pool (2x2)
Dropout (0.25)
Fully connected layer (256)
Dropout (0.5)
Output layer (3 for translation and 6 for rotation)
Architecture 2

Input 160x160 → 2D CNN → Max pool 2x2 → 2D CNN → Max pool 2x2 → Dropout 0.25 → Dropout 0.5 → Features
Input and Training

Dimension: 160x160

Each pixel \((i, j)\)
- Feature of residue \(i\) from chain 1
- Feature of residue \(j\) from chain 2
- Concatenate the features of \(i\) and \(j\)

Training
- Learning rate: 0.001
- Optimize MSE loss
Experiments

Exp1
- Architecture 1
- Epochs: 500
- Batch size: 100

Exp2
- Architecture 2
- Epochs: 500
- Batch size: 100

Exp3
- Translation from Exp1
- Rotation from Exp2
Experiments

Exp4
- Resnet18
- Epochs: 35

Exp5
- Translation from Exp4
- Rotation (Exp4; stops training if loss<0.2 and at least 20 epochs)
# Result

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Average TM score before</th>
<th>Average TM score after</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>0.52258</td>
<td>0.4991</td>
</tr>
<tr>
<td>Exp2</td>
<td>0.52258</td>
<td>0.4957</td>
</tr>
<tr>
<td>Exp3</td>
<td>0.52258</td>
<td>0.4986</td>
</tr>
<tr>
<td>Exp4</td>
<td>0.52258</td>
<td>0.4965</td>
</tr>
<tr>
<td>Exp5</td>
<td>0.52258</td>
<td>0.4972</td>
</tr>
</tbody>
</table>
## Result

<table>
<thead>
<tr>
<th>Experiment</th>
<th>TM increased</th>
<th>TM decreased</th>
<th>TM same</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>136</td>
<td>276</td>
<td>68</td>
</tr>
<tr>
<td>Exp2</td>
<td>107</td>
<td>301</td>
<td>72</td>
</tr>
<tr>
<td>Exp3</td>
<td>103</td>
<td>303</td>
<td>74</td>
</tr>
<tr>
<td>Exp4</td>
<td>101</td>
<td>300</td>
<td>79</td>
</tr>
<tr>
<td>Exp5</td>
<td>104</td>
<td>301</td>
<td>75</td>
</tr>
</tbody>
</table>
Significant improve -> TM is increased by at least 0.04

<table>
<thead>
<tr>
<th>Experiment</th>
<th>#Significant Improve (out of 480)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>75</td>
</tr>
<tr>
<td>Exp2</td>
<td>68</td>
</tr>
<tr>
<td>Exp3</td>
<td>70</td>
</tr>
<tr>
<td>Exp4</td>
<td>70</td>
</tr>
<tr>
<td>Exp5</td>
<td>69</td>
</tr>
</tbody>
</table>
## Result

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Average IDDT before</th>
<th>Average IDDT after</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>0.678</td>
<td>0.669</td>
</tr>
<tr>
<td>Exp2</td>
<td>0.678</td>
<td>0.66784</td>
</tr>
<tr>
<td>Exp3</td>
<td>0.678</td>
<td>0.66785</td>
</tr>
<tr>
<td>Exp4</td>
<td>0.678</td>
<td>0.668</td>
</tr>
<tr>
<td>Exp5</td>
<td>0.678</td>
<td>0.66783</td>
</tr>
</tbody>
</table>
# Result

<table>
<thead>
<tr>
<th>Experiment</th>
<th>IDDT increased</th>
<th>IDDT decreased</th>
<th>IDDT same</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>48</td>
<td>386</td>
<td>46</td>
</tr>
<tr>
<td>Exp2</td>
<td>2</td>
<td>420</td>
<td>58</td>
</tr>
<tr>
<td>Exp3</td>
<td>0</td>
<td>422</td>
<td>58</td>
</tr>
<tr>
<td>Exp4</td>
<td>11</td>
<td>411</td>
<td>58</td>
</tr>
<tr>
<td>Exp5</td>
<td>1</td>
<td>422</td>
<td>57</td>
</tr>
<tr>
<td>Experiment</td>
<td>#Significant Decrease (out of 480)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp1</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp2</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp3</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp4</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp5</td>
<td>23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Result**

Significant decrease -> Decrease by at least 0.05
Findings

Refinement is sensitive to the prediction of sign

Chain 2 was taken far away in many of the complex structures

Convolutional neural networks may not work

Many of the structures had the same TM score after the transformation, because only chain 1 was considered during the alignment and chain 2 was ignored (chain 2 was too far to be aligned)

A few number of structures had significant improvement and therefore, the proposed method has potential to work better
Future work

Graph Neural Network (on progress)
More decoys for training and testing
Flexible refinement to capture conformational changes
Questions