A Contact-assisted Approach to Protein Structure Prediction and Its Assessment in CASP10

Badri Adhikari, Xin Deng, Jilong Li, Debswapna Bhattacharya, and Jianlin Cheng
Department of Computer Science, University of Missouri, Columbia, MO 65211 USA
chongji@missouri.edu

14th July 2013
What is a **contact** in a protein structure?

Distance(7,41) = 5.8 Å
The problem

**Objective:**

1. select good models from the predicted model pool and,
2. improve the models (if possible) with this contacts information

**Sources:**

(a) wet lab experiments,
(b) contact prediction tools
Method Overview

• **Rank models in the pool**
  – Using existing MQAP tool APOLLO
  – Using contacts satisfaction score
  – Combined approach

• **Refine selected models**
  – Using 3Drefine to refine the selected models

• **Remodel the refined models**
  – Using Modeller to refold the models with contacts as distance restraints
Model Quality Assessment using APOLLO

pairwise comparison of models

### Scores

<table>
<thead>
<tr>
<th>Model</th>
<th>GDT-TS</th>
<th>MaxSub</th>
<th>TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>model1</td>
<td>0.5770</td>
<td>0.0458</td>
<td>0.0647</td>
</tr>
<tr>
<td>model2</td>
<td>0.6863</td>
<td>0.0530</td>
<td>0.0679</td>
</tr>
<tr>
<td>model3</td>
<td>0.7020</td>
<td>0.0551</td>
<td>0.0733</td>
</tr>
<tr>
<td>model4</td>
<td>0.5013</td>
<td>0.0551</td>
<td>0.0733</td>
</tr>
<tr>
<td>model5</td>
<td>0.2017</td>
<td>0.0551</td>
<td>0.0733</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Adding contacts component to rank models

\[
Total \ Score = \underbrace{GDT-TS \ score + MaxSub \ score + TM-score}_{\text{APOLLO component}} + (\% \ of \ contacts \ satisfied) - 0.1 \times (\% \ of \ no-contacts \ satisfied) \underbrace{\}}_{\text{Contacts component}}
\]
Remodeling using MODELLER

Input model as template  ➔  Contacts as distance restraints  ➔  Modeller

Program for Comparative Protein Structure Modelling by Satisfaction of Spatial Restraints

 ➔  Refolded model
Score models by performing model quality assessment using APOLLO

Score models based on how well they satisfy the input contacts or no-contacts

Compute Total Score for each model using the scores and rank the models

Select top models and refine them using 3Drefine

Remodel using MODELLER with contacts as distance restraints
Results Overview

• Data
  – 15 CASP10 Contact assisted targets

• Evaluation of
  – top 1 models, step by step
  – the model selection formula

• A case study
Results: CASP10 contact-assisted targets

- 15 target proteins
- 3 to 34 known contacts provided
- several long-range contacts
- mostly missed contacts were provided
- data is challenging to our method

<table>
<thead>
<tr>
<th>Target #</th>
<th>Target</th>
<th># of residues</th>
<th># of contacts / no contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T0649</td>
<td>184</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>T0653</td>
<td>383</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Tc658-D1</td>
<td>166</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>Tc666</td>
<td>180</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>Tc673</td>
<td>62</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Tc676</td>
<td>173</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Tc678</td>
<td>154</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Tc680</td>
<td>96</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>Tc684-D1</td>
<td>73</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Tc684-D2</td>
<td>168</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>T0691</td>
<td>141</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>Tc705-D2</td>
<td>344</td>
<td>34</td>
</tr>
<tr>
<td>12</td>
<td>Tc717-D2</td>
<td>166</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>Tc719-D6</td>
<td>163</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Tc734</td>
<td>212</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>Tc735-D1</td>
<td>233</td>
<td>28</td>
</tr>
<tr>
<td>15</td>
<td>Tc735-D2</td>
<td>88</td>
<td>7</td>
</tr>
</tbody>
</table>
## Results: CASP10 contact-assisted targets

<table>
<thead>
<tr>
<th>#</th>
<th>Target</th>
<th>Selected Model</th>
<th>Refined Model</th>
<th>Modeller Model</th>
<th>Final Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSD</td>
<td>GDT-TS</td>
<td>RMSD</td>
<td>GDT-TS</td>
</tr>
<tr>
<td>1</td>
<td>Tc649</td>
<td>13.880</td>
<td>0.3026</td>
<td>13.790</td>
<td>0.3026</td>
</tr>
<tr>
<td>2</td>
<td>Tc653</td>
<td>18.780</td>
<td>0.2190</td>
<td>18.770</td>
<td>0.2203</td>
</tr>
<tr>
<td>3</td>
<td>Tc658</td>
<td>14.220</td>
<td>0.2078</td>
<td>14.220</td>
<td>0.2078</td>
</tr>
<tr>
<td>4</td>
<td>Tc666</td>
<td>8.355</td>
<td>0.2889</td>
<td>8.366</td>
<td>0.2917</td>
</tr>
<tr>
<td>5</td>
<td>Tc673</td>
<td>9.384</td>
<td>0.3566</td>
<td>9.399</td>
<td>0.3525</td>
</tr>
<tr>
<td>6</td>
<td>Tc676</td>
<td>11.480</td>
<td>0.2425</td>
<td>11.470</td>
<td>0.2455</td>
</tr>
<tr>
<td>7</td>
<td>Tc678</td>
<td>6.980</td>
<td>0.3842</td>
<td>6.966</td>
<td>0.3826</td>
</tr>
<tr>
<td>8</td>
<td>Tc680</td>
<td>2.979</td>
<td>0.7473</td>
<td>2.956</td>
<td>0.7446</td>
</tr>
<tr>
<td>9</td>
<td>Tc684</td>
<td>20.460</td>
<td>0.1250</td>
<td>20.460</td>
<td>0.1239</td>
</tr>
<tr>
<td>10</td>
<td>Tc691</td>
<td>9.619</td>
<td>0.3358</td>
<td>9.597</td>
<td>0.3376</td>
</tr>
<tr>
<td>11</td>
<td>Tc705</td>
<td>13.440</td>
<td>0.1853</td>
<td>13.450</td>
<td>0.1875</td>
</tr>
<tr>
<td>12</td>
<td>Tc717</td>
<td>18.050</td>
<td>0.1747</td>
<td>18.070</td>
<td>0.1747</td>
</tr>
<tr>
<td>13</td>
<td>Tc734</td>
<td>17.620</td>
<td>0.1392</td>
<td>17.660</td>
<td>0.1403</td>
</tr>
<tr>
<td>14</td>
<td>Tc719</td>
<td>23.140</td>
<td>0.1411</td>
<td>23.170</td>
<td>0.1380</td>
</tr>
<tr>
<td>15</td>
<td>Tc735</td>
<td>25.150</td>
<td>0.2269</td>
<td>25.140</td>
<td>0.2269</td>
</tr>
</tbody>
</table>

**Average Change:** 2.302 -0.002
Evaluation of Ranking

All models grouped by targets

<table>
<thead>
<tr>
<th>Ranking Method</th>
<th>Average Correlation</th>
<th>Average Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Score Formula</td>
<td>0.601</td>
<td>0.088</td>
</tr>
<tr>
<td>APOLLO Component only</td>
<td>0.559</td>
<td>0.087</td>
</tr>
<tr>
<td>Contacts Component only</td>
<td>0.390</td>
<td>0.088</td>
</tr>
</tbody>
</table>
Case study: target Tc734

native in green

Model Ranked 1

Refined with 3Drefine

Refolded with MODELLER

<table>
<thead>
<tr>
<th></th>
<th>RMSD</th>
<th>GDT-TS</th>
<th>GDT-HA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>17.33</td>
<td>0.3079</td>
<td>0.1738</td>
</tr>
<tr>
<td>Refined</td>
<td>17.33</td>
<td>0.3090</td>
<td>0.1717</td>
</tr>
<tr>
<td>Refolded</td>
<td>7.789</td>
<td>0.3498</td>
<td>0.1835</td>
</tr>
</tbody>
</table>
But, mostly

1. we do NOT have true contacts
2. we may not have a quality model pool

Ongoing Research

Two approaches to build 3D structures from scratch:
• Contact-assisted protein structure prediction using Distance Geometry Simulated Annealing (CNS)
• Contact-assisted fragment replacement based protein structure prediction
Contact-assisted prediction using Distance Geometry Simulated Annealing

The pipeline:

1. Sequence
2. Secondary structure prediction using PSPro, SSPro, and PSIPRED
3. Contact prediction using DNcon, SVMcon, and NNcon
4. Contacts filtering
5. Prepare different set of contacts
6. Run Distance Geometry Simulated Annealing using CNS
## Results

Green is predicted
Gray is native

<table>
<thead>
<tr>
<th>RMSD</th>
<th>GDT-TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>0.60</td>
</tr>
<tr>
<td>4.5</td>
<td>0.55</td>
</tr>
<tr>
<td>4.4</td>
<td>0.58</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compound</th>
<th>RMSD</th>
<th>GDT-TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0665</td>
<td>3.6</td>
<td>0.60</td>
</tr>
<tr>
<td>T0560</td>
<td>4.5</td>
<td>0.55</td>
</tr>
<tr>
<td>T0716</td>
<td>4.4</td>
<td>0.58</td>
</tr>
</tbody>
</table>
Contact-assisted fragment replacement based protein structure prediction

Randomly pick a fragment of (length 9)

Predict features (SS and SA) for this fragment

Lookup in the fragment database to find the most matching fragment

Convert structure into angular space and replace the fragment

Compute energy of the new structure (using contact satisfaction score)

Convert structure back to Cartesian co-ordinates

Extended structure

Final Structure

Monte-Carlo Simulated Annealing
Results

Green is predicted
Gray is native
Protein folding process for target T0716 demonstrated through a movie

http://www.youtube.com/watch?v=HBONCqN9U4k
Conclusion

• Known contacts can be incorporated into existing protein structure prediction techniques to improve protein model ranking and generation.
• Initial results demonstrate that folding proteins using predicted contacts and predicted secondary structures is a promising direction.
• Quality of top contacts is the key to good structures.
• Filtering contacts from predicted contacts list is the most important as well as the most challenging problem for accurate structure predictions.
Acknowledgements

Advisor:
• Dr. Jianlin Cheng

Group:
• Sharif Ahmed
• Debswapna Bhattacharya
• Renzhi Cao
• Xin Deng
• Jilong Li
• Matt Spencer
• Tuan Anh Trieu

Dr. Cheng’s Bioinformatics, Data Mining, Machine Learning Laboratory (BDML) Computer Science Department, University of Missouri, Columbia, MO
Thank You
Supplementary Slides
Weight calculation for Total Score formula
Primary filtering algorithm

1. Sort all the contacts predicted according to the contact score
2. Pick the first contact $x \leftrightarrow y$ and insert 3 contacts into a list $L$:
   i. $x \leftrightarrow y$
   ii. $x \leftrightarrow (y-1)$
   iii. $x \leftrightarrow (y+1)$
3. Pick second pair and check if it exists in list $L$,
   i. if it does, do nothing
   ii. if it does not, repeat step (b) for this new pair
4. Continue, for all predicted contacts

Q. Why not use complete neighborhood size of 5 instead of 3 on one side?
Inserting 5 sets decreases the total number of filtered contacts very less.
Different set of contacts used

1) Total 0.4L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts
2) Total 0.5L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts
3) Total 0.8L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts
4) Total 1.0L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts
5) Total 1.5L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts
6) Total 2.0L number of contacts consisting of top 33.33% SR, 33.33% MR, and 33.33% LR contacts

L

Sequence length

SR
Short Range contacts (sequence separation <= 11 and >= 6)

MR
Medium Range contacts (sequence separation <= 23 and >= 12)

LR
Long Range contacts (sequence separation >= 24)
Planned pipeline

**protein_structure_predict.sh**
- Inputs:
  a) Fasta
  b) Contact prediction + threshold
  c) Sulphide bonds prediction
  d) β-sheet detection
  e) Secondary structure prediction
  f) Native pdb (for evaluation)

**cns_dgsa.sh**
- CNS script that does not involve prediction
  - Input: all standard CNS parameters
  - Output: x structures (default 20)
Transforming contacts into distance restraints for DGSA

- Transforming contacts (8A) into distance restraints:

  Residues in Contact: A - B

  Distance btw A and B: 6
  +- deviation: -3 and +2

- Current implementations can be improved.
Transforming Secondary Structure prediction to NOE distance restraints, Hydrogen bonds, and angular restraints

- **Helix**

  \[
  \begin{align*}
  \Phi & = -57.8; \\
  \Phi_{dev} & = 0.1; \\
  \Psi & = -47.0; \\
  \Psi_{dev} & = 0.1;
  \end{align*}
  \]

- **Strand**

  \[
  \begin{align*}
  \Phi & = -139.0; \\
  \Phi_{dev} & = 1.0; \\
  \Psi & = 135.0; \\
  \Psi_{dev} & = 1.0;
  \end{align*}
  \]

  most important restraints

- detect if sheets form strands
- parallel/anti-parallel
How many contacts do we need?

Now that we can make secondary structures roughly

\( N_c \) that is up to 100% of the sequence length. We calculated 20 structures for \( N_c = 30 \) up to \( L \) in steps of 10

where \( L \) is the length of the protein in the PFAM alignment. For instance, for protein of length 100, we calculated 160 structures (20 structures per bin size). This range is comparable to the number of true distances needed as constraints to reconstruct a known protein structure, which is between 15-30% of the number of residues in the protein [2,3] as discussed above. We extended the analysis to calculate structures for very few constraints (\( \leq 20 \)) and for larger numbers of constraints to test whether with

Our algorithm computes sparse subsets capable of determining the tertiary structure at approximately 4.8 Å Ca RMSD with as little as 8% of the native contacts (Ca-Ca and Cb-Cb). At the same time, a randomly chosen subset of native contacts needs about twice as many contacts to reach the same level of accuracy.

With as little as 40% of the contacts reasonably good models can be produced. On the contrary the
### Evaluation of predicted structures (comparison with native structure):

<table>
<thead>
<tr>
<th>MODEL</th>
<th>RMSD</th>
<th>TM</th>
<th>MaxSub</th>
<th>GDT-TS</th>
<th>GDT-RA</th>
</tr>
</thead>
<tbody>
<tr>
<td>extended.pdb</td>
<td>32.72</td>
<td>0.1079</td>
<td>0.1197</td>
<td>0.1569</td>
<td>0.0943</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_12.pdb</td>
<td>9.310</td>
<td>0.3199</td>
<td>0.3420</td>
<td>0.4104</td>
<td>0.2972</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_2.pdb</td>
<td>9.143</td>
<td>0.3733</td>
<td>0.3898</td>
<td>0.4675</td>
<td>0.3349</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_4.pdb</td>
<td>9.131</td>
<td>0.3353</td>
<td>0.3503</td>
<td>0.4057</td>
<td>0.3019</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_17.pdb</td>
<td>8.991</td>
<td>0.3379</td>
<td>0.3480</td>
<td>0.4009</td>
<td>0.2972</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_17.pdb</td>
<td>8.687</td>
<td>0.3205</td>
<td>0.3514</td>
<td>0.4151</td>
<td>0.2877</td>
</tr>
<tr>
<td>cns_1.5L_0.35R_0.3MR_0.3LR_17.pdb</td>
<td>5.843</td>
<td>0.3452</td>
<td>0.3378</td>
<td>0.4717</td>
<td>0.2925</td>
</tr>
<tr>
<td>cns_2.0L_0.35R_0.3MR_0.3LR_8.pdb</td>
<td>5.842</td>
<td>0.3407</td>
<td>0.3230</td>
<td>0.4764</td>
<td>0.2972</td>
</tr>
<tr>
<td>cns_2.0L_0.35R_0.3MR_0.3LR_10.pdb</td>
<td>5.636</td>
<td>0.3561</td>
<td>0.3488</td>
<td>0.4811</td>
<td>0.3019</td>
</tr>
<tr>
<td>cns_2.0L_0.35R_0.3MR_0.3LR_3.pdb</td>
<td>5.763</td>
<td>0.3080</td>
<td>0.3009</td>
<td>0.4811</td>
<td>0.2877</td>
</tr>
<tr>
<td>cns_2.0L_0.35R_0.3MR_0.3LR_12.pdb</td>
<td>5.753</td>
<td>0.4003</td>
<td>0.4156</td>
<td>0.5283</td>
<td>0.3396</td>
</tr>
<tr>
<td>cns_1.5L_0.35R_0.3MR_0.3LR_6.pdb</td>
<td>5.672</td>
<td>0.3985</td>
<td>0.3967</td>
<td>0.5047</td>
<td>0.3302</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_10.pdb</td>
<td>5.660</td>
<td>0.4122</td>
<td>0.4325</td>
<td>0.5377</td>
<td>0.3679</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_7.pdb</td>
<td>5.622</td>
<td>0.3716</td>
<td>0.4161</td>
<td>0.5377</td>
<td>0.3726</td>
</tr>
<tr>
<td>cns_2.0L_0.35R_0.3MR_0.3LR_7.pdb</td>
<td>5.622</td>
<td>0.3517</td>
<td>0.3227</td>
<td>0.4611</td>
<td>0.2877</td>
</tr>
<tr>
<td>cns_0.4L_0.35R_0.3MR_0.3LR_9.pdb</td>
<td>5.543</td>
<td>0.3670</td>
<td>0.3713</td>
<td>0.5000</td>
<td>0.3302</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_9.pdb</td>
<td>5.500</td>
<td>0.3065</td>
<td>0.3600</td>
<td>0.4953</td>
<td>0.3113</td>
</tr>
<tr>
<td>cns_1.0L_0.35R_0.3MR_0.3LR_7.pdb</td>
<td>5.390</td>
<td>0.3285</td>
<td>0.3132</td>
<td>0.5047</td>
<td>0.3019</td>
</tr>
<tr>
<td>cns_1.0L_0.35R_0.3MR_0.3LR_13.pdb</td>
<td>5.330</td>
<td>0.3937</td>
<td>0.4023</td>
<td>0.5142</td>
<td>0.3302</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_16.pdb</td>
<td>5.366</td>
<td>0.3564</td>
<td>0.3447</td>
<td>0.4966</td>
<td>0.3019</td>
</tr>
<tr>
<td>cns_1.5L_0.35R_0.3MR_0.3LR_13.pdb</td>
<td>5.355</td>
<td>0.3013</td>
<td>0.3070</td>
<td>0.5283</td>
<td>0.3349</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_3.pdb</td>
<td>5.336</td>
<td>0.3229</td>
<td>0.3988</td>
<td>0.5142</td>
<td>0.3255</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_13.pdb</td>
<td>5.321</td>
<td>0.3035</td>
<td>0.3020</td>
<td>0.5094</td>
<td>0.3019</td>
</tr>
<tr>
<td>cns_1.5L_0.35R_0.3MR_0.3LR_6.pdb</td>
<td>5.259</td>
<td>0.3877</td>
<td>0.4041</td>
<td>0.5330</td>
<td>0.3632</td>
</tr>
<tr>
<td>cns_1.5L_0.35R_0.3MR_0.3LR_16.pdb</td>
<td>5.204</td>
<td>0.4023</td>
<td>0.4155</td>
<td>0.5330</td>
<td>0.3443</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_19.pdb</td>
<td>5.259</td>
<td>0.3922</td>
<td>0.4196</td>
<td>0.5283</td>
<td>0.3443</td>
</tr>
<tr>
<td>cns_1.0L_0.35R_0.3MR_0.3LR_1.pdb</td>
<td>5.227</td>
<td>0.3474</td>
<td>0.3401</td>
<td>0.5094</td>
<td>0.3066</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_17.pdb</td>
<td>5.223</td>
<td>0.3384</td>
<td>0.4270</td>
<td>0.5330</td>
<td>0.3349</td>
</tr>
<tr>
<td>cns_0.5L_0.35R_0.3MR_0.3LR_2.pdb</td>
<td>5.110</td>
<td>0.3441</td>
<td>0.3464</td>
<td>0.5047</td>
<td>0.3113</td>
</tr>
<tr>
<td>cns_1.0L_0.35R_0.3MR_0.3LR_15.pdb</td>
<td>4.993</td>
<td>0.3773</td>
<td>0.3664</td>
<td>0.6142</td>
<td>0.3160</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_7.pdb</td>
<td>4.850</td>
<td>0.3084</td>
<td>0.3092</td>
<td>0.4983</td>
<td>0.3113</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_16.pdb</td>
<td>4.858</td>
<td>0.3698</td>
<td>0.4464</td>
<td>0.5602</td>
<td>0.3821 (maximum GDT-TS)</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_14.pdb</td>
<td>4.728</td>
<td>0.3558</td>
<td>0.3639</td>
<td>0.5084</td>
<td>0.3113</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_19.pdb</td>
<td>4.772</td>
<td>0.3556</td>
<td>0.3664</td>
<td>0.5226</td>
<td>0.3349</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_14.pdb</td>
<td>4.559</td>
<td>0.3537</td>
<td>0.3408</td>
<td>0.5000</td>
<td>0.3066</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_15.pdb</td>
<td>4.668</td>
<td>0.3520</td>
<td>0.3615</td>
<td>0.5377</td>
<td>0.3302</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_13.pdb</td>
<td>4.327</td>
<td>0.3974</td>
<td>0.4473</td>
<td>0.5613</td>
<td>0.3505</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_13.pdb</td>
<td>4.055</td>
<td>0.3950</td>
<td>0.4215</td>
<td>0.5425</td>
<td>0.3349</td>
</tr>
<tr>
<td>cns_0.6L_0.35R_0.3MR_0.3LR_20.pdb</td>
<td>3.810</td>
<td>0.4302</td>
<td>0.5001</td>
<td>0.5802</td>
<td>0.3774 (maximum GDT-TS and minimum RMSD)</td>
</tr>
</tbody>
</table>

Averages: GDT-TS 0.446 RMSD 7.187
Results

Green is predicted
Gray is native
Monte Carlo Simulated Annealing

Initialization:

# n is iterations, T is temperature, L is sequence length
pdb=extended structure
E_i=energy(pdb)
T_i=L
T_f=0.001

while (t < T_f){
    for(100 times, r = random number between 1 and L){
        fragment_replace()
        if(count_clash() == 0){
            reject move
        }
        E_r=energy(pdb)
        if(E_r < E_i){
            accept move
        }
        else{
            ΔE=E_r-E_i;
            α = exp(-ΔE/t)
            if(r <= α){
                accept move
            } else{
                reject move
            }
        }
    }
    t = next_temperature()
}

next_temperature()

    t = L, L-10, L-20, .., 10, 8, .., 1, 0.9, 0.8, 0.7, .., 0.1, 0.09, 0.08, .., 0.01
1. **Targets**

   For fragment assembly based protein structure prediction we choose 3 CASP10 targets: T0737-D1, T0693-D1 and T0756-D2.

<table>
<thead>
<tr>
<th>Target</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0737-D1</td>
<td>117</td>
</tr>
<tr>
<td>T0693-D1</td>
<td>100</td>
</tr>
<tr>
<td>T0756-D2</td>
<td>86</td>
</tr>
</tbody>
</table>

2. **Building fragment database**

   We used “PISCES” PDB database to build our fragment database. The PDB database had 4057 native structures.

   One fragment consists of 9 residues.

   Three steps used for building the fragment database:

1. **Extracting features from native pdb**
   - The tool “stride” was used to extract the features for each native pdb.
   - Following five features were extracted:
     i. Secondary structure
     ii. Solvent accessibility
     iii. \( \Phi \) (phi angle)
     iv. \( \varphi \) (psi angle)
     v. Residue type
   - Wrote a script to parse the “stride” results.

2. **Splitting the native pdb files into fragments**
   - Wrote a script to cut the native pdb into fragment
   - Issue: in some native pdb files, residue numbers are not continuous, as shown in Figure 1
   - We skipped these missing residues when building the fragment database

   ![Figure 2 Screenshot of fragment database file](image)

3. **Combining fragments and feature files to build fragment database**
   - Wrote a script to combine the fragment cut from native structures (from step 2) and the features generated (from step 1)
   - A screenshot of the fragment database is shown in Figure 3
3 Fragment search algorithm

3.1 Prediction of secondary structure and solvent accessibility of input fragment

We used SSpro4.0 to predict the secondary structure and solvent accessibility of the three target proteins, as shown in Figure 3.

Figure 3 Predicted secondary structure and solvent accessibility of targets

3.2 Extracting segments from targets

For each residue of the input target file containing the target sequence, predicted secondary structure, and predicted solvent accessibility, one segment with 9 residues centered by itself is extracted.

If there are less than 4 residues at left or right, some “X” will be added into the segment.

For example, for the target T0693-D1:

The segment for 1st residue:
SEQ: XXXXMIbLA
SS: XXXXCCCH
SOL: XXXXeeebb

The segment for 5th residue:
SEQ: MIDLAPlvR
SS: CCCCHHHHHH
SOL: eeebbebbe

The segment for the last residue:
SEQ: VHiDTXXX
SS: EECXXXXX
SOL: bbeebbeXXX

3.3 Searching each segment against the fragment database

![Flowchart]

Formula for scoring database fragments against input segment:
\[
\text{score} = \frac{2(\text{residue match count}) + (\text{ss match count}) + (\text{sa match count})}{4}
\]
Each extracted segment of the target is searched against the fragment database. For each fragment in the fragment database, we compared its residues, secondary structure, and solvent accessibility to those of the extracted segment of the target. One matched score will be calculated based on this comparison.

This score ranges from 0 to 9 if the length of segment is 9, and it is based on residue match count, secondary structure match count, and solvent accessibility match count.

After comparing the segment of the target to all the fragments in the fragment database, **top 100 fragments in the fragment database with highest scores are picked**. The matched score, phi angles, and psi angles of each of the top 100 fragments are written into the output file.

### 3.4 Output of fragment search algorithm

The output file of fragment search contains:

- residue number, matched score, and corresponding phi angles and psi angles in the window size of 9

For each residue number, top 100 group of phi angles and psi angles are copied to the output file, and each group of phi angles and psi angles is in one line.

```
Resid,Score:phi1 psi1, phi2 psi2, phi3 psi3, phi4 psi4, ..., phi9 psi9
1, 4: 999 999, 999 999 999 999, 999 999 999, ..., -125.70 174.83
..., ...
17, 7: -7.675 -8.555 -10.215 -121.48 -97.93 -158.73, -32.41 -41.76, ..., -57.34 -44.99
..., ...
100, 4: -106.02 109.83, -91.76 -149.18, -128.97 163.44, -67.64 -15.55, ..., 999 999
```

**Figure 4 Screenshot of output file.** Resid means the central residue number of each fragment in the target. Score means matched score, and phi and psi mean $\Phi$ and $\Phi$ of residues in the fragment with highest matched score.

---

### 4 Fragment Replacement Strategy

- a. Get the $\phi$-$\psi$ angles from the Cartesian system using rama.c program in Crankite package.
- b. Randomly pick a fragment from the top 100 fragments for the given 9-residue window and get the $\phi$-$\psi$ angles.
- c. We noticed that conversion from $\phi$-$\psi$ angle-space to Cartesian space is not ideal with lipa.c.
- d. We convert the torsion space attributes to Spherical coordinate system and perform the fragment replacement. (shown in diagram below)
- e. Finally, we convert the spherical coordinate system back to Cartesian coordinates.
5 Contact Based Potential Energy Function

5.1 Contact Prediction

We first predict residue-residue contacts using nncns tool and convert the contacts into a matrix

\[
M_{ij} \text{ (1 = contact, 0 = not contact) distance threshold = 8Å}
\]

And \( K_g(M_{ij}) \) is a two-valued function defined as:

\[
K_g(M_{ij}) = w_{\text{reward}} \text{ if } M_{ij} = 1
\]
\[
K_g(M_{ij}) = w_{\text{penalty}} \text{ if } M_{ij} = 0
\]

\( w_{\text{reward}} \) is to reward true positive contacts.

\( w_{\text{penalty}} \) is to penalize false positive contacts.

5.2 Energy Function

General form of Potential Energy Function:

\[
E_{\text{total}} = \sum_{j=1}^{N} K_g(M_{ij}) \cdot H[d - |c_i - c_j|]
\]

where, \( d = 8 \text{ Å} \) and \( H[n] \) is the Heaviside step function with the form:

\[
H[n] = \begin{cases} 
0, & n < 0, \\
1, & n \geq 0,
\end{cases}
\]

5.3 Greedy Reward and Penalty Approaches

We adopted the following 4 strategies:

<table>
<thead>
<tr>
<th>#</th>
<th>Description of Greedy Approach</th>
<th>( w_{\text{reward}} )</th>
<th>( w_{\text{penalty}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Give more weight to long range True Positive contacts Penalize False Positive contacts</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Give more weight to long range True Positive contacts No Penalty for False Positive contacts</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Give equal weight to all True Positive contacts No Penalty for False Positive contacts</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Give equal weight to all True Positive contacts Penalize False Positive contacts</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>