Implementation of a Flexible Docking Code into the CMDF Project
## Overview of the ModMSCDock Method

### Objectives

- Accurately describe small molecule binding sites on proteins of arbitrary structure
- Predict non-bond interaction energies between macromolecules and small, drug-like ligand candidates.
- Streamline the drug design process to rely mostly on cheap and fast computational models over expensive and slow laboratory experiments.

### Approach

- Convert the protein into a grid of points which each hold the values for the electrostatic and vdW summations at that point.
- The energy of a ligand configuration is then the sum of interpolations of each ligand atom to the eight nearest grid points plus the ligand internal energy.
- Perform biased Monte-Carlo and molecular dynamics searches using algorithms designed here at MSC, Caltech.

### Milestones and Achievements

- Completed 51,500 lines of python code to manage ligands and proteins in several different file formats, perform five different methods for docking and several algorithms for post-dock work up, including molecular minimization and dynamics and PBF, SGB, or AVGB solvation.
- Next milestone: the python code will be bridged to the main modules of the CMDF project, which uses standardized data structures such as OpenBabel and gOpenMol.

Example:

Ligand 1tni entering the binding pocket of trypsin
Docking Uses a Wide Selection of Software

- **Tools designed at Caltech**
  - **bgf2fsm** - atom typing for AVGB solvation
  - **anchor_dock** - Dock 4.01 enabled to perform large scale anchor searches
  - **dock_div** - Dock 4.01 with a run-time diversity filter to ensure a search of a known completeness.
  - **solvation** - fast computation of solvent accessible surface area (SASA’s) of ligands, proteins, & protein complexes
  - **2pt** - analysis of the velocity autocorrelation of MD trajectories for estimating the contribution of entropy to the binding of protein - ligand complexes
  - **mpsim** - mechanics and dynamics using the cell multipole method (CMM)

- **Other Software**
  - **NAMD** - efficient parallel molecular dynamics with periodic boundary conditions and fast particle mesh Ewald for calculations with explicit water solvation (now integrated directly into CMDF)
  - **Amber** - protein building and trajectory analysis
  - **APBS** - PBF implicit solvation models (energies & forces)
  - Utilities from the Dock suite of programs (sphgen, connolly_ms, convsyb etc.)
All Controlled from Python Core Scripts: ModMSCDock

- Master mscd.py calls other python scripts as needed
  - Overview of different types of scripts:
    - Specifying and preparing the receptor protein for docking
    - Specifying and checking ligand files for atom typing
    - Generating the molecular surface and spheres
    - Running the docking job
  - Performing post-docking work up:
    - Post_Diversity: removes ligand orientations that are very similar
    - BuriedSurface: removes ligands with an insufficient amount of surface area buried by the protein (a crude estimate of solvation effects)
    - Mpsim_Rescore: rescores orientations using MPSim for accuracy
  - Performing final scoring
    - Level 1 - minimization with protein fixed; ligand movable
    - Level 2 - minimization with protein & ligand both movable
Docking Methods

- **DockDiv**
  - Uses a run time geometric clustering algorithm to ensure a diverse pool of conformations (also called “docking with diversity”)

- **Anchor Search**
  - Plants several anchors (groups of atoms with no rotatable bonds) and uses piecewise construction to grow the remainder of the ligand into the protein potential

- **Torsion Drive**
  - Randomly generates ligand conformations and proceeds to dock them one by one using automated matching

- **Automated Matching**
  - Generates conformations by matching distances between ligand atoms to distances between spheres

- **TorsionDrive with DockDiv**
  - Performs dock with diversity on each conformation generated by torsion drive
Docking Results

Methods predicting a conformation <1.0 Ang rmsd to xtl

<table>
<thead>
<tr>
<th>Protein</th>
<th>Best RMSD</th>
<th>Method</th>
<th>Protein</th>
<th>Best RMSD</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1abe</td>
<td>0.30</td>
<td>DATU</td>
<td>1ppc</td>
<td>0.28</td>
<td>DAU</td>
</tr>
<tr>
<td>1abf</td>
<td>0.32</td>
<td>DATU</td>
<td>1rbp</td>
<td>0.46</td>
<td>DATU</td>
</tr>
<tr>
<td>1apb</td>
<td>0.46</td>
<td>DATU</td>
<td>1rgk</td>
<td>0.43</td>
<td>DAU</td>
</tr>
<tr>
<td>1apw</td>
<td>0.33</td>
<td>DU</td>
<td>1rgl</td>
<td>0.64</td>
<td>D</td>
</tr>
<tr>
<td>1bap</td>
<td>0.30</td>
<td>DATU</td>
<td>1rnt</td>
<td>0.59</td>
<td>DAU</td>
</tr>
<tr>
<td>1bra</td>
<td>0.61</td>
<td>DATU</td>
<td>1sre</td>
<td>0.40</td>
<td>DATU</td>
</tr>
<tr>
<td>1dhf</td>
<td>0.46</td>
<td>DU</td>
<td>1tet</td>
<td>1.10</td>
<td>D</td>
</tr>
<tr>
<td>1dr1</td>
<td>0.35</td>
<td>DU</td>
<td>1tnh</td>
<td>0.30</td>
<td>DTU</td>
</tr>
<tr>
<td>1drf</td>
<td>0.44</td>
<td>DAU</td>
<td>1tni</td>
<td>0.72</td>
<td>DAU</td>
</tr>
<tr>
<td>1ela</td>
<td>3.44</td>
<td>none</td>
<td>1tnj</td>
<td>0.64</td>
<td>DU</td>
</tr>
<tr>
<td>1etr</td>
<td>0.35</td>
<td>DU</td>
<td>1tnk</td>
<td>0.98</td>
<td>D</td>
</tr>
<tr>
<td>1exw</td>
<td>10.96</td>
<td>none</td>
<td>1tnl</td>
<td>0.18</td>
<td>D</td>
</tr>
<tr>
<td>1hvr</td>
<td>0.41</td>
<td>DA</td>
<td>1yyy</td>
<td>0.47</td>
<td>DU</td>
</tr>
<tr>
<td>1inc</td>
<td>0.54</td>
<td>DAU</td>
<td>1zzz</td>
<td>0.48</td>
<td>DU</td>
</tr>
</tbody>
</table>
**Docking Results**

Methods predicting a conformation <1.0 Ang rmsd to xtl

<table>
<thead>
<tr>
<th>Protein</th>
<th>Best RMSD</th>
<th>Method</th>
<th>Protein</th>
<th>Best RMSD</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>2ak3</td>
<td>0.40</td>
<td>DU</td>
<td>5sga</td>
<td>0.29</td>
<td>DU</td>
</tr>
<tr>
<td>2cgr</td>
<td>0.91</td>
<td>A</td>
<td>6abp</td>
<td>0.31</td>
<td>DATU</td>
</tr>
<tr>
<td>2gbp</td>
<td>0.29</td>
<td>DTU</td>
<td>6rnt</td>
<td>17.50</td>
<td>none</td>
</tr>
<tr>
<td>2qwb</td>
<td>0.34</td>
<td>DAU</td>
<td>6tim</td>
<td>0.32</td>
<td>DAU</td>
</tr>
<tr>
<td>2qwc</td>
<td>0.40</td>
<td>DAU</td>
<td>7abp</td>
<td>0.30</td>
<td>DATU</td>
</tr>
<tr>
<td>2qwd</td>
<td>0.35</td>
<td>DTU</td>
<td>7est</td>
<td>0.40</td>
<td>DAU</td>
</tr>
<tr>
<td>2qwe</td>
<td>0.43</td>
<td>DAU</td>
<td>8abp</td>
<td>0.35</td>
<td>DATU</td>
</tr>
<tr>
<td>2qwg</td>
<td>0.60</td>
<td>DAU</td>
<td>9abp</td>
<td>0.37</td>
<td>DATU</td>
</tr>
<tr>
<td>2sns</td>
<td>0.90</td>
<td>DAU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2xim</td>
<td>0.27</td>
<td>DAU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2xis</td>
<td>1.43</td>
<td>none</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3ptb</td>
<td>0.36</td>
<td>DATU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4sga</td>
<td>0.28</td>
<td>DAU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5abp</td>
<td>0.30</td>
<td>DATU</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary of Results for 50 Globular Proteins

- Complete Misses
  - 1ela - elastase
  - 1exw - palmitoyl protein thioesterase I
  - 2xis - xylose isomerase
  - 6rnt - ribonuclease T1

- DockDiv - 46 / 50
- Automated Matching - 40 / 50
- Anchor Search - 32 / 50
- Torsion Drive - 16 / 50
Future Work

- Port DockDiv & AnchorDock to C++ in order to use \textit{catch} and \textit{throw} to handle memory exceptions

- Create the scoring grids using the Dreiding FF

- Change the object data structure in DockDiv & AnchorDock to the OpenBabel structure in order to have complete compatibility with all of the CMDF tools

- Use the method of analyzing explicit water dynamics of protein complexes using velocity autocorrelation and \textit{2pt} to predict the entropic contribution to binding for a set of about 150 carbonic anhydrase inhibitors with known experimental data - i.e. continue our research of obtaining binding constants using explicit water dynamics
Acknowledgements

- William A. Goddard III and the MSC Biogroup

- The various executables used by MSCDock have been designed by years of research at Caltech and elsewhere.

- The immediate python code for using these tools for docking was written by John Wendel and Jiyoung Heo.