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.

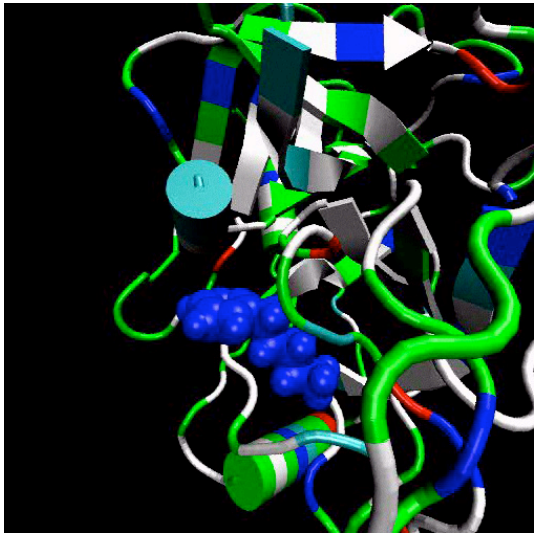
• The energy of a ligand configuration is then the sum of interpolations of each ligand atom to the eight nearest grid points plus the ligand internal energy.

• 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

D - dock_div T - torsion drive

A - anchor search U - Automated Matching

Methods predicting a conformation <1.0 Ang rmsd to xtl

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

Docking Results

D - dock_div
A - anchor search
T - torsion drive
U - Automated Matching

Methods predicting a conformation <1.0 Ang rmsd to xtl

Protein	Best RMSD	Method	Protein	Best RMSD	Method
2ak3	0.40	DU	5sga	0.29	DU
2cgr	0.91	A	6abp	0.31	DATU
2gbp	0.29	DTU	6rnt	17.50	none
2qwb	0.34	DAU	6tim	0.32	DAU
2qwc	0.40	DAU	7abp	0.30	DATU
2qwd	0.35	DTU	7est	0.40	DAU
2qwe	0.43	DAU	8abp	0.35	DATU
2qwg	0.60	DAU	9abp	0.37	DATU
2sns	0.90	DAU			
2xim	0.27	DAU			
2xis	1.43	none			
3ptb	0.36	DATU			
4sga	0.28	DAU			
5abp	0.30	DATU			

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 *catch* and *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 *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.