YUCCA: An Efficient Algorithm for Small Molecule Docking

Vicky Choi

Department of Computer Science, Virginia Tech

Abstract. In this paper, we present a new algorithm, which is based on an efficient heuristic for local search, for rigid protein-small molecule docking. We tested our algorithm, called YUCCA, on the recent 100-complex benchmark [3], using the conformer generator OMEGA[16] to generate a set of low-energy conformers. The results showed that YUCCA is competitive both in terms of algorithm efficiency and docking accuracy.

Keywords. Protein-small molecule docking, virtual screening, drug design.

1 Introduction

The computational prediction of the three-dimensional structures of receptor-ligand complexes, where the receptor is a protein and the ligand is a small molecule, is called *protein-ligand* or *protein-small molecule* docking. Accurate and efficient protein-small molecule docking algorithm is of fundamental importance to structure-based drug design. During the drug design process, once the three-dimensional structure of a target protein (which is believed to be responsible for the disease) is determined, one can try to identify new drug candidates by *virtual screening* a database of known compounds through small molecule docking.

Docking problem. In general, there are two parts to the docking problem: a *scoring or evaluation function* that can discriminate correctly (i.e. experimentally observed) docking solutions (called poses) from incorrect ones; and a *search algorithm* that searches the configurational and conformational space for the candidate poses measured by the scoring function. The docking problem is challenging because the scoring function, which measures the binding affinity between ligand and receptor, is not completely understood; and the search space is high dimensional — besides six degree of freedom (DOF) of the configurational space, both molecules are flexible and might undergo conformational changes upon binding, which results in hundreds to thousands DOFs of conformational space. It is computationally infeasible to perform exhaustive conformational searches during docking. Thus far, the most commonly used approach in modelling protein-small molecule docking is to consider only the conformational space of the ligand, assuming the protein receptor is rigid. This is known as rigid-receptor flexible-ligand docking model.

Existing algorithms. In the last 20 years, many docking algorithms have been developed, see reviews [1, 2] and comparison studies [3, 4]. Broadly, these algorithms can be classified in three categories: *stochastic search*, *incremental construction* and *mutli-conformers docking* algorithms. The representatives for stochastic search algorithm are AutoDock[5], ICM [6], GOLD[7] etc. These algorithms are based on genetic algorithms and/or Monte Carlo simulated annealing. The incremental construction algorithms first dissect each molecule into a set of *rigid fragments* according to rotatable bonds, and then incrementally assemble the fragments around the binding pocket. Some examples of this class are DOCK[8], FlexX[9] and Surflex[10]. Unlike the incremental construction, multi-conformer docking algorithms separately generate a set of low-energy conformers, and then do rigid docking for each conformer. These include FLOG[11] and FRED (OpenEye Scientific Software). A brief description of FRED can be found in [4]. Among the existing algorithms, the more efficient algorithms are FRED, DOCK and FlexX.

Scoring function. Besides the search algorithm, critical to the accuracy of the docking algorithm is the scoring function it employs. The ideal scoring function would calculate the binding affinities between ligand and receptor, which include factors such as van der Waals interaction, hydrogen bonding, hydrophobicity, electrostatics etc. However, it is not well understood what is the proportion of the contribution of each factor. There are three main approaches to studying scoring function, namely, *force field-based*, *empirical-based* and *knowledge-based*; see [12] for a recent review. Force field-based method approximates the score by non-bond energy terms from the well-studied force field, such as AMBER or CHARMM. Empirical-based method uses a set of protein-ligand complexes which have the binding affinities determined experimentally to train the parameters in the scoring function. Knowledge-based method uses the Boltzmann hypothesis with the known structural database to compute the score. However, despite the extensive research, no scoring function comes close to the ideal and sometimes consensus scoring functions are used. See [13] for a recent comparative study of 11 popular scoring functions. Here we remark that most of the docking algorithms are capable of handling different (additive) scoring functions by interpolating the score through a grid, which can be precomputed and stored.

Docking accuracy & efficiency. Docking algorithms are evaluated based on their *docking accuracy*, *ranking accuracy* and *algorithm efficiency*. A solution pose is considered accurate if the root-mean-sqaure-deviation (RMSD) of the pose and the reference (experimentally determined) pose is less than 2 Å. Docking accuracy

requires one of the top 30 scored poses to be accurate while the ranking accuracy requires the top scored pose to be accurate. For a docking algorithm to be useful, it is required to be efficient and to achieve high docking accuracy. Recently, a comparative study of eight popular docking programs was conducted on a 100-complex benchmark [3].

New algorithm. In this paper, we will describe a new algorithm for *rigid* small-molecule docking. Thus our algorithm falls into multi-conformers docking category, which requires generating a set of low-energy conformers separately. The idea of the algorithm is based on an efficient heuristic for local search, which finds a low energy configuration within a local neighborhood. The global low energy (the binding) configuration is then identified by coarse sampling the input configuration. Using OMEGA 1.8.1 to generate a set of conformers (as FRED did), we tested YUCCA on the 100-complex benchmark. Performance of YUCCA is competitive with efficiency of average 4 seconds on a 3.0GHz Pentium IV computer running Linux Fedora 8.0 per docking complex, docking accuracy of 76% and ranking accuracy of 45% (see Section 3 for the comparison with other docking programs).

In Section 2, we describe our new algorithm YUCCA. We then report our experimental results on the 100-complex data set on Section 3. Finally, we conclude with discussion in Section 4.

2 YUCCA: The New Algorithm

2.1 Our scoring function.

There are two attributes to our scoring function: energy and bump. Here we define the energy of each atom pair to be its piecewise linear potential (PLP) [14] energy, which is one of the best and yet the simplest ([12, 13]); and the bump of an atom pair is 1 if its energy is greater than 0. The total energy (bump resp.) is the sum of energy (bump resp.) over all possible (heavy) atom pairs between ligand and receptor. Note that our objective is to find the lowest energy (with a small tolerated number of bumps) docking configuration.

2.2 High level description.

The algorithm coarsely samples a set of initial configurations. For each configuration, an efficient local search procedure heuristically finds a low energy configuration within a local neighborhood. The basic idea of the local search heuristic is similar to the idea used in [15] for protein-protein docking. But unlike the protein-protein docking case, for small molecules we precompute the scoring function in the grids which will then allow efficient look-up. Given a configuration of ligand around the active site of protein (which is held rigid and fixed), we want to move the ligand locally by a rigid motion such that the resulting complex configuration is of lower energy. Intuitively, imagine that each atom of the ligand is free to move locally, the atom will move to the *lowest* energy point within the local neighborhood, called its *attractor*, as shown in Figure 2. Since the atoms are not free from each other – there are intra-distance constraints between atoms, we apply least squares superposition for the atoms and their attractors to obtain a new (low energy expected) configuration. However, since the rigid motion does not restrict the moving range of each individual atom (because the objective function is RMSD), some of the ligand atoms in the new configuration might collide with the protein. We then reduce the collisions by again employing least squares superposition with *weights*. Intuitively, if an atom is bump-free, we prefer not to move it and thus match it with its current position; otherwise the atom is matched to its nearest bump-free grid point.

2.3 Details of the algorithm.

Conceptually, the algorithm consists of the following three steps:

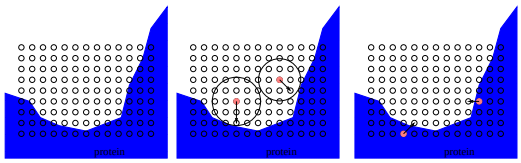
```
0. Preprocessing: precompute grids;
1. Sample a set of initial configurations;
2. Local search each configuration:
   Outer loop: lower energy,
       Inner loop: bump reducing.
```

In the following, we describe each step in detail.

Preprocessing. In the preprocessing step, we compute four different types of grids, each with four different atom types (hydrogen bond donor, hydrogen bond acceptor, hydrogen bond donor/acceptor and nonpolar) as classified in PLP[14], with total of 16 grids. There are energy grids, bump grids, attractor grids and bump-free grids (see Figure 1). The grid side size depends on the diameter of the reference ligand. It is set to be 1.5 times the diameter of the reference ligand. For energy, bump and bump-free grids, the grid spacing is set to 0.2Å, while it is set to 0.8Å for attractor grid. For each atom type, the value of a grid point of the energy grid is equal to the energy of a ligand atom (with its center sitting at the grid point) with the protein receptor (which is the sum of energies between the ligand atom and all protein atoms). Similarly, the value of a grid point of the bump grid is just the number of bumps of the ligand atom and protein. Both energy and bump grids are used to approximate the score of each computed configuration. Unique to our algorithm, there are two extra grids: attractor grid and bump-free grid. For attractor grid, we first compute

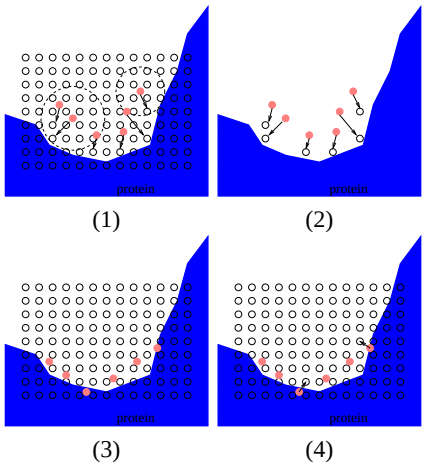
a local neighborhood distance, which equals to the distance between the grid point and the nearest protein atom center (limited by a maximum local neighborhood distance). Then each grid point points to the lowest energy grid point within the local neighborhood. For bump-free grid, each grid point points to the nearest bump-free grid point within the maximum local neighborhood allowed. See Figure 1 for an illustration.

Lower energy outer loop. As explained above, the main intuition of this step is to move the ligand to a lower energy configuration. This step was achieved by matching each atom center with its attractor (which is the lowest energy grid point within the neighborhood) and then apply the least squares fit to obtain a new (lower energy expected) configuration. See Figure 2.



(a) grid (b) attractor grid (c) bump-free grid

Fig. 1: (a) A grid around the active site of protein. For energy grid, each grid point stores the energy value of a ligand atom sitting at the grid point with protein. For bump grid, each grid point stores the number of bumps of a ligand atom sitting at the grid point with the protein. (b) Attractor grid: each grid point points to the lowest energy point within its local neighborhood. (c) Bump-free grid: each bump grid point points to its nearest bump free grid point.



(1) (2)

(3) (4)

Fig. 2: (1) Each ligand atom is "attracted" to the lowest energy grid point within its neighborhood. (2) There is a correspondence between the center of each ligand atom and its attractor (an instance of least-squares fit). (3) Apply the least squares fit to (2) and obtain the new configuration. (4) Match each bump atom with its nearest bump-free grid point.

Bump reducing inner loop. To reduce the number of bumps of the current configuration, we match each bump atom to its bump-free grid point by looking up from the bump-free grid, while matching the atom without bump with its current position. In order to effectively reduce the number of bumps, we set a larger weight to the bump (atom) pair. The weight depends on the distance of the pair. Intuitively, the closer the pair, the larger the weights should be. The weight is set to inverse proportional to the square distance of the pair. The algorithm iteratively reduces bumps in the inner loop until either the number of bumps is not greater than the tolerance or the movement is too small (measured by RMSD of two consecutive configurations). Then the algorithm iterates the outer loop to lower the energy. The inner loop is repeated at most 5 times while the outer loop is iterated at most 3 times (these numbers were determined through empirical tests). The lowest 30 energy configurations with at most 15 bumps are retained.

Sampling. First, a quasi-centroid was computed based on the low energy grid points. More precisely, quasi-centroid is the centroid of grid points with energy at most -2. Empirical results show that the quasi-centroid is at most 2.5Å away from the centroid of the bound ligand. Translations were then sampled with 8 cubical grid points around the quasi-centroid with distance 2Å This gave the centroid of a sampled configuration to be within 1.5Å from the centroid of the bound ligand empirically. Rotations were represented by unit quaternions: each rotation is specified by a rotation angle about a rotation axis. The axes are chosen to be the 20 uniformly distributed unit vectors on the unit sphere. The angle is inversely proportional to the diameter of the ligand (with minimum of 30 degrees). Again, these parameters were mined through extensive empirical tests.

3 Experimental Results

We tested YUCCA on the 100-complex benchmark [3]. The diversity of the benchmark was assessed according to several physicochemical descriptors, including molecular weights, number of rotatable bonds, number of H-bond donors and acceptors, volume of protein cavity and polar surface area. First, we use OMEGA 1.8.1 to generate a set of conformers with the same parameters as those used in [3] for FRED. That is, the maximum number of output conformers was 500; the upper bound relative to the global mimumum was 3 kcal/mol; the RMSD value below which two conformations are considered same was set to 0.8Å with 25 maximum rotatable bonds. The time required by OMEGA for generating conformers for YUCCA is on average 1.4 seconds. In total, 5967 conformers for the 100-complex benchmark were generated.

It took YUCCA on average 4 seconds on a 3.0GHz Pentium IV computer running Linux Fedora 8.0 per docking complex, with docking accuracy of 76% and ranking accuracy of 45%. The eight docking programs studied in [3] are: DOCK, FlexX, FRED, Glide, GOLD, Slide, Surflex and QXP. Among them, Glide, GOLD and Surflex achieved the highest docking accuracy of more than 80% and ranking accuracy of 50˜55%, while FRED is the fastest algorithm with average 18 seconds, followed by average 46 seconds by DOCK, on a 270 MHz SGI R12K processor running IRIX6.5. It should be noted that the programs were running on different computers and hence the running time are not directly comparable. Our algorithm is not yet optimized and we expect our algorithm to be at least as fast as FRED if running on the same computer.

4 Discussion and Future Work

With the rapid increase in the available 3D structures derived by large-scale structure-determination projects, effi cient and accurate docking algorithms will be even more important for studying molecular recognition. In particular, these docking algorithms will serve as a valuable tool for structure-based drug design. Despite of some successful applications, much of the challenges of an effi cient and accurate docking algorithm still remain. From an algorithmic point of view, besides the new Gaussian function-based docking algorithm FRED, the more effi cient approaches adopted by small-molecule docking are all based on template matching and incremental construction (e.g. DOCK, FlexX). In this paper, we proposed a new non-template non-incremental approach and the experimental results showed its promise to be a competitive docker. In the experimental tests above, we use OMEGA to generate conformers which might contribute to some failing docking cases. Currently, we are working on developing a conformer generator based on the correlated torsion angle database similar to [17]. We are also working on incorporating the preferred torsion angles into YUCCA to handle flexibility "on-the-fly" based on the "directed tweak" method[18]. Different scoring functions are under investigation to improve both docking and ranking accuracy. Finally, we are testing the *screening utility* of YUCCA and will report on this in the full version of the paper.

Acknowledgments

We would like to thank Dr. Didier Rognan for providing us the 100-complex benchmark in [3] and thank OpenEye Scientifi c Software for providing OMEGA. We would also like to thank Gavin Tsai, Navin Goyal, David Bevan, Joel Gillespic and Bradley Feuston for the discussion and comments.

References

1. D. B. Kitchen, H. Decornez, J.R. Furr & J. Bajorath, Docking and Scoring in virtual screening for drug discovery: methods and applications. *Nature Reviews Drug Discovery* 3, P.935-949 (2004).
2. R.D. Taylor, P.J. Jewsbury, J.W. Essex. A review of protein-small molecule docking methods, *J. Comput. Aided. Mol. Des.* 16 3 (2002), pp. 151-166.
3. E. Kellenberger, J. Rodrigo, P. Muller, D. Rognan. Comparative evaluation of eight docking tools for docking and virtual screening accuracy, *Proteins.* 2004, 57, 225-242.
4. T Schulz-Gasch, M Stahl. Binding site characteristics in structure-based virtual screening: evaluation of current docking tools. *J. Mol. Mod.* 2003; 9:47-57.
5. DS Goodsell, AJ Olson. Automated docking of substrates to proteins by simulated annealing. *Proteins.* 8:195-202, 1990.
6. M Totrov, R Abagyan. Flexible protein-ligand docking by global energy optimization in internal coordinates. *Proteins* 1997; Suppl 1: 215-220.
7. G Jones, P Willett, RC Glen, AR Leach, R Taylor. Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* 1997; 267: 727-748.
8. T.J.A. Ewing, S. Makino, A.G. Skillman, ID Kuntz. DOCK 4.0: Search strategies for automated molecular docking of flexible molecule databases, *J. Comput. Aided. Mol. Des.* 15 5 (2001), pp. 411-428.
9. M Rarey, B Kramer, T Lengauer and G Klebe. A Fast Flexible Docking Method using an Incremental Construction Algorithm *J. Mol. Biol.*, 261 3, (1996), 470-489.
10. A. Jain. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *J Med Chem.* 2003 Feb 13;46(4):499-511.
11. M.D. Miller, S.K. Kearsley, D.J. Underwood, R.P. Sheridan. FLOG: a system to select 'quasi-flexible' ligands complementary to a receptor of known three-dimensional structure. *J Comput Aided Mol Des.* 1994 Apr;8(2):153-74.
12. T. Schulz-Gasch, M. Stahl. Scoring functions for protein-ligand interactions: a critical perspective. *Drug Discovery Today: Technologies*, article in press, 2004.
13. R. Wang, Y. Lu and S. Wang. Comparative evaluation of 11 scoring functions for molecular docking. *J. Med Chem* **46**, 2287-2303, 2003.
14. G.M. Verkhivker, D. Bouzida, D.K. Gehlhaar et al. Deciphering common failures in molecular docking of ligand-protein complexes. *J. Comput. Aided. Mol. Des.* 14 8 (2000), pp. 731-751.
15. V. Choi, P. K. Agarwal, H. Edelsbrunner and J. Rudolph. Local Search Heuristic for Rigid Protein Docking. *The 4th Workshop on Algorithms in Bioinformatics (WABI 2004).*
16. J. Bostrm, JR Greenwood and J Gottfries. Assessing the performance of OMEGA with respect to retrieving bioactive conformations. *J. Mol. Graph. Mod.*, 2003, 21, 449-462.
17. BP Feuston, MD Miller, JC Culberson, RB Nachbar, SK Kearsley. Comparison of knowledge-based and distance geometry approaches for generation of molecular conformations. *J Chem Inf Comput Sci.* 2001 May-Jun;41(3):754-63.
18. Hurst, T.. Flexible 3D-search: the directed tweak method. *J. Chem. Inf. Comp. Sci.*, 34: 190, 1994.