Reza Jacob
BioChem218
Final Project

Docking Algorithms of Virtual Ligand Screening

Abstract

Recent efforts in structural biology have lead and will lead to an exponential growth in the number of structures publicly available; many of these structures are potential drug targets. Because of recent advances in docking algorithms, *in silico* screening provides an attractive alternative to traditional screening for drug leads, since *in vitro* high-throughput screening of compounds is costly and relatively inefficient. The virtual ligand screening process consists of selecting a target protein, mathematically modeling the protein surface, choosing or building a library of ligands, flexibly docking those ligands, scoring the binding of each ligand, and selecting candidate compounds for further studies. The computational crux of virtual ligand screening resides in the docking and scoring algorithms, which are usually distinct. The most successful and the most widely used virtual screening programs employ one of various docking algorithms and often combine several of various scoring functions. Though there are many examples in which virtual ligand screening significantly aided drug development, the upshot of this paper will be to describe several representative docking algorithms and to illustrate their strengths and weaknesses.

Introduction

In the post-genomic area, scientific interest in the biomedical community is now focused on proteomics. The Structural Proteomic Initiative reflects this, as does the exponential growth of the number of structures in the Protein Data Bank (Hol 2000). One potential use for this wealth of structural information is structure-aided drug development. The ability to accurately predict or simulate *in silico* the interaction of a protein with a ligand is critical for the success of any form of structure-aided drug design. In order to screen a library of compounds for the ability to bind to a protein of interest, high-throughput docking algorithms are necessary. Along with these algorithms, a scoring function is needed to prioritize hits in order of relevance for chemical synthesis and *in vitro* or studies. The process of virtual ligand screening attempts to sift through a list of compounds in a library using a docking algorithm to discriminate a small number of binders from a very large number of non-binders and subsequently rank the binders according to their potential as candidate drugs (Shoichet et al 2000).

There are several advantages of virtual ligand screening over traditional *in vitro* screening. One advantage of *in silico* screening is that a very large number of compounds can be screened without synthetically generating each compound. A typical corporate chemical library designed for use with high-throughput screening *in vitro* contains 0.2 – 1.0 million compounds, yet often no viable candidate drugs emerge from such a screen (Abagyan et al 2001). Because of the costly nature of such *in vitro* screens (Birve et al 2002), virtual screening reduces the cost of drug discovery as well as eliminates time spent in assaying a library from which no hits emerge. In addition, the

chemical composition of a virtual library can be boundlessly complex and diverse, unlike libraries produced by combinatorial chemistry (Abagyan et al 2001). Thus, many academic groups as well as corporate entities have pursued research in developing virtual ligand screening technologies.

Phases of Virtual Ligand Screening:

To perform virtual ligand screening on a target, one must first select a target of interest. This is, after all, perhaps the most important step: from a list of structures, one must chose a biomedically relevant protein with a structure that has pockets of shape and contour suitable to drug design. Because of the lack of such features, the number of possible drug targets is far less than the number of unique proteins of known structure (Brive et al 2002). For a protein to be "drugable," it must have a small molecule binding site, and if one knows the location of the binding site of interest *a priori* to the *in silico* experiment, the chances of success of structure-aided drug design is much greater (Brive et al 2002). The simplest way to find a drugable protein and locate such a binding site is to select a structure with a ligand bound. Though I will not discuss them, there are algorithms that locate potential solvent-exposed binding sites given the input of structural coordinates, and these algorithms are based on contour maps of the protein surface (Brive et al 2002). Once a target is selected, the virtual ligand screening process can be broken down into five phases: modeling of the target, generating a virtual library, docking each member in the library, scoring each docked ligand, and selecting candidate ligands for further investigation (Abagyan et al 2001).

Having an accurate representation, or model, of the drug target (the protein under investigation) is essential to any virtual ligand screening process. High-resolution X-ray crystallographic structures are routinely used for protein modeling. In addition, structures derived from NMR experiments or those predicted by homology modeling may also be used. If the model of the target protein is extracted from a structure in which a ligand is bound to the protein, then docking experiments using that model are termed "bound docking." Otherwise, those experiments are termed "unbound docking" (Halperin et al 2002). Because of the conformational changes that occur between the liganded and unliganded forms of many a protein, performing bound docking is preferable to performing unbound docking. In other words, having a protein modeled in its liganded state typically yields better results through most docking algorithms over having a protein modeled in its unliganded state. Not surprisingly, the most important part of the model is the protein's binding site; if the binding site is modeled poorly, virtual ligand screening will not yield useful results (Halperin et al 2002). Though not necessary for all docking algorithms, some require the user to specify the location of the binding site of interest. Details of the modeling process, including choice of coordinate system and surface representation, will be described below.

After modeling a target, one must create or select a library of compounds to be used in the virtual screen. Ideally, the library should consist of small, orally bioavailable compounds which are not toxic and do not provoke an immune response. Because of digestibility and because of immunological concerns, peptides and proteins are usually not included in libraries used for virtual ligand screening. The library can be a set of

commercially available compounds or a collection of drug-like hits from previous *in vitro* high-throughput screening studies; alternatively, the library can be generated *in silico* based on scaffolds of known ligands or drugs or based upon knowledge of the binding site structure. However it is formed, the library should be large, diverse, and chemically feasible for synthesis (Abagyan et al 2001, Brive et al 2002).

In docking each member of the library to the target, the goal is to quickly and correctly predict the binding geometry of the ligand complexed with the protein. This goal is achieved by "finding the global minimum of an ill-behaved energy function of hundreds of variables," (Brive et al 2002). There are several programs available which attempt to do this, and they differ in complexity, theoretical orientation, and implementation. The docking algorithm is the most time-consuming step in the virtual ligand screening process, and algorithms that take more than 3 minutes per ligand per processor are considered too lengthy for use in a high throughput screen (Abagyan et al 2001); all the programs covered here are sufficiently fast so that they meet this requirement. To correctly predict the geometry of the bound ligand, the ligand must be modeled as flexible. All modern docking algorithms used in structure-aided drug design model the ligand as flexible, though the receptor does not need to be modeled as flexible (Abagyan et al 2001). Flexible docking algorithms can be easily categorized (Bissantz et al 2000). Examples of flexible docking algorithms and some programs that feature them include:

- Fast Shape Matching
  - DOCK
  - Eudock
- Monte Carlo methods
  - Internal Coordinate Mechanics (ICM)
  - QXP
  - MCDOCK
- Genetic algorithms
  - AutoDock
  - GOLD
  - Gambler
- Incremental Construction
  - FlexX
  - SLIDE
  - Hammerhead

Selected Docking Algorithms for use in Virtual Ligand Screening

Program	**Designer/company**	**Web site**
Affinity	Accelrys	http://www.accelrys.com/insight/affinity.html
AutoDock	David Goodsell	http://www.scripps.edu/pub/olson-web/doc/autodock/
BioMedCAChe	CAChe Software	http://www.cachesoftware.com/biomedcache
CAChe for Medicinal Chemists	CAChe Software	http://www.cachesoftware.com/medcache/
DOCK	Irwin Kuntz	http://www.cmpharm.ucsf.edu/kuntz/
DockVision	Steven Ness and	http://www.dockvision.com

	Trevor Hart	
FlexX	Thomas Lengauer and Matthias Rarey	http://cartan.gmd.de/flexx/
Glide	Schrödinger Inc.	http://www.schrodinger.com/Products/glide.html
GOLD	Cambridge Crystallographic Data Centre	http://www.ccdc.cam.ac.uk/prods/gold/index.html
Hammerhead	Ajay Jain	http://cc.ucsf.edu/jain/index.html
PRO_LEADS	Protherics plc	http://www.protherics.com/crunch/section5.html
SLIDE	Leslie Kuhn	http://www.bch.msu.edu/labs/kuhn/web/projects/screening.html
VRDD	Zhiping Weng	http://bme.bu.edu/faculty/weng.html

Table 1 (adapted from Willis 2001) lists some docking programs along with their authors and features hyperlinks to the homepages of various docking programs.

After a ligand has been docked, a scoring function is applied to compute the tightness of fit between the target and the ligand. The scoring phase is the most difficult phase in virtual ligand screening, and there is much room for improvement of scoring functions (Halperin et al 2002). If a ligand is incorrectly docked, calculation of a correct score is impossible (Abagyan et al 2001). The most accurate scoring functions are based on conformational sampling methods such as free energy perturbations and linear interaction energies approximations, and these methods cannot be used in virtual ligand screening because they are too computationally expensive (Bissantz et al 2000). Scoring functions that are fast enough to be used in virtual ligand screening are based one of three methods: force-field methods (Dock, GOLD), empirical free energy scoring functions (Ludi, Chemscore, Score, Fresno, FlexX, Plp), or knowledge-based methods (Pmf, Drugscore). All of these programs can effectively predict binding free energies within 3 kcal/mol of the true binding free energy (Bissantz et al 2000). The choice of which scoring algorithm to use depends on the intended use of the algorithm: some scoring algorithms are best suited to discriminate a small number of binders from a very large number of non-binders, while others are best used to prioritize a small number of chemically related binders for further investigation *in vitro* or *in vivo*. Most scoring functions fail to predict a good score for experimentally verified ligand-target interactions that were not represented in a training set (Abagyan et al 2001). Because many scoring programs over-fit data to particular training cases or families, many researchers resort to programs (such as CScore) that feature a combination of scoring functions to rank compounds (Bissantz et al 2000). Others have shown that a combination of independent scoring functions performs superior to any single scoring function in discriminating binders from non-binders (Charifson et al 1999).

In selecting leads for further research, scoring obviously plays the critical role. Sometimes a particular chemical scaffold dominates the high-ranking hits (Abagyan et al 2001). Metabolic considerations such as undesirable absorption, distribution, metabolism, excretion, and toxicity properties can hamper the development of leads from the virtual ligand screening process. Alternately, the lead compounds might be synthetically infeasible. It seems that many of the problems associated with the output of virtual ligand screening programs actually derive from problems associated with the

ligand library. If one could screen for poor bioavailability or high toxicity *in silico*, one could eliminate compounds from the library. However, knowing that a highly toxic ligand binds well may still be of use to drug design because it might be possible to later modify the lead to yield lower toxicity, perhaps at the expense of binding efficiency. If one could detect an overabundance of a chemical scaffold in the library, one could perhaps eliminate some of these compounds or otherwise modify the docking algorithm to use information gained during the docking of one compound to speed up the docking of a closely related compound. For example, if two similar compounds differ only by an R- group, then knowing the docked conformation of one might greatly facilitate the docking of the other. In any case, researchers suggest manual screening of the top 300 or so hits from a virtual ligand screening run.

Discussion

The three phases of virtual ligand screening that are the most computationally intense are modeling, docking, and scoring. These three aspects are interrelated, in that any given choice of a surface model suggests a conformational search strategy, which then implies how to rank ligands to be pursued for further study (Halperin et al 2002). For each of several representative virtual ligand screening programs, each of these three aspects will be discussed in detail below, though docking algorithms will be given the most attention.

Internal Coordinate Mechanics: ICM-Dock and Monte Carlo Methods

The ICM-Dock program (Abagyan et al 1994) is part of the ICM Bioinformatics and Computational Structural Biology suite. In ICM, a force field is used to model the protein surface. ICM-Dock features an internal coordinate representation of the system in torsional space, rather than in Cartesian space (Brive et al 2002). The user can select whether the protein is modeled rigidly on a grid or flexibly; the ligand is explicitly modeled in torsion space. Variables such as bond length and bond angle that oscillate with a high frequency are held fixed: this simplification, along with the torsional coordinates, reduces the degrees of freedom by seven-fold compared to a Cartesian space without fixed bond length and bond angles (Brive et al 2002). Moreover, if Valence Shell Electron Pair Repulsion Theory is applied to compute the geometry of all atoms, a ten-fold decrease in the degrees of freedom is observed, compared to the Cartesian space with no restrictions (Brive et al 2002). One main advantage gained by such a representation lies in the speedup achieved by reducing the degrees of freedom of the system. The other advantage of representing the system in torsional space is the ease at which conformational space can be sampled in the docking algorithm by manipulation torsion angles.

ICM-Dock essential uses a biased Monte Carlo algorithm to minimize an energy function in torsional space. The algorithm begins by pseudorandomly selecting a set of torsion angles in the ligand and subsequently finding the local energy minimum about those angles. The selection of torsion angles to manipulate during Monte Carlo steps is not completely random but rather somewhat biased to increase the speed of the search (Brive et al 2002). By partially biasing the exploration of conformational space, rare ligand conformations can be explored rapidly, and high efficiency is maintained with an order of

magnitude gain in speed (Abagyan et al 2001). Once the angles to be manipulated have been chosen, a new conformation is then adopted according to the Metropolis criteria. The change in free energy, ΔG, between the current state of the system and the state of the proposed move is calculated. The move is accepted with probability $\min(1,\exp[-\Delta G/RT])$, where R is the universal gas constant and T is the absolute temperature of the simulation (Abagyan et al 1994). The Monte Carlo steps are then repeated, in conjunction with methods to foster sampling of unexplored conformational space: during this Monte Carlo process, a stack of low energy conformations is created, and if the same conformation is visited a certain number of times, the simulation temperature T is doubled (Brive et al 2002). Thus, the algorithm allows for the escape from local minima while maintaining a record of them. Once the global minimum is found, ICM-Dock uses consensus scoring, combining several independent scoring functions.

DOCK and Shape-Matching Methods

The program DOCK (Kuntz et al 1996), developed at the University of California, San Francisco, uses a geometry-based search to explore conformational space in docking a ligand. DOCK and other geometric algorithms treat the protein target as rigid, and its surface model is based upon the Connolly surface, which is obtained by rolling a probe sphere with the radius of a water molecule over the van der Waals surface of the protein (Halperin et al 2002). The program superimposes ligand atoms onto precomputed critical points on the target surface, which are either local extrema or saddle points (Kuntz et al 1996). The collection of critical points maps out the *negative image* of the target's binding site: if the target surface has a knob or protrusion, its negative image will have a depression or hole and vise versa. A matching algorithm determines which atoms of the ligand are superimposed upon which critical points (Willis 2001). Different possible alignments of the ligand with the target's negative image are scored using a function that assesses intermolecular interactions such as van der Waals forces, electrostatics, hydrophobic interactions, and solvation effects. The highest scoring alignment is retained.

In choosing critical points on both the target and the ligand, the sparseness of the points is important, since algorithmic complexity depends on the number of critical points (Halperin et al 2002). The geometric matching algorithm also computes surface normals of the target's negative image and the ligand at their critical points. The problem, in essence, reduces to finding a rigid transformation of the ligand (rotation plus translation) that achieves the pairing of critical points on the ligand with those on the target's negative image such that the surface normals are parallel. Thus a geometric algorithm detects pairs of critical points in both the ligand and the target's negative image that share the same internal distance and have parallel surface normals after superpositioning. Because the ligand and target are held rigid, it suffices to align three pairs of non-collinear points, or equally, congruent triangles (Halperin et al 2002). Some recent versions of DOCK and other geometric alignment algorithms incorporate flexibility of the ligand (Willis 2001). This can be accomplished if the docking procedure is repeated with a different conformation of the ligand. Although during a run of the matching algorithm the ligand is held rigid, the overall program can perform flexible docking because different rigid conformations of the same ligand can be threaded through the

matching algorithm.

AutoDock and Genetic Algorithms/Lamarckian Genetic Algorithms

The Genetic Algorithm

Genetic algorithms borrow themes and terms from genetics and evolution. A force field model of the target surface is used in these algorithms, similar to those used in ICM; the receptor is held rigid and fixed in space. In genetic algorithms, a set of *state variables*, which encode the ligand's translation, orientation, and conformation, correspond to the ligand's genotype; the ligand's atomic coordinates correspond to the ligand's phenotype (Morris et al 1998). Each state variable is a gene, and from these genes the phenotype of the ligand can be computed based on a free energy function. Genes are represented linearly on a chromosome and can be mutated and recombined. The algorithm begins with a population of the same ligand, in which each of a user-specified number of individual is randomly or pseudorandomly assigned genes within physical bounds. Once the starting population is determined, the algorithm begins (Morris et al 1998).

A *mapping* step converts the genotype of each individual in the population into a phenotype (Morris et al 1998). The *fitness* of a ligand denotes the sum of the free energy of intermolecular interaction between the ligand and the protein with the intramolecular interaction energy within the ligand, as evaluated by the free energy function. A *selection* event then occurs in which individuals with poor fitness die. Next in this scheme, individuals exchange genes through mating, or *crossover*, by which new individuals are created through the combination of genetic information from two parents (Morris et al 1998). In the crossover, no breaks occur within a gene, and for each mating, exactly two crossover events occur in AutoDock's implementation; if one individual has genotype *ABCDE* and another has genotype *abcde*, one possible offspring would have genotype AbcdE. Moreover, individuals with greater fitness mate more than those of lesser fitness. Additionally, a certain percentage of offspring undergo random *mutation* events, in which genes are changed by a random amount, taken from a Cauchy distribution (Morris et al 1998). The use of the Cauchy distribution is sound because all genes are encoded as real numbers and because the Cauchy distribution attains values more distant from the mean more often than the normal distribution. The crossover and mutation rates are user-specified. After mating, all the individuals of the older generation die of old age, except a certain user-specified *elitist* fraction of individuals with the highest fitness (Morris et al 1998). The process is then repeated, starting at the mapping phase; the algorithm loops over *generations* until either the maximum number of generations exceeds the user-defined limit or the maximum number of energy calculations exceeds a user-defined limit. Throughout the algorithm, birth rates and selection are adjusted such that the population size is held constant (Morris et al 1998).

The Lamarckian Genetic Algorithm

The genetic algorithm is efficient at globally exploring the conformational space, but it is not specifically suited to finding local minima (Morris et al 1998). Therefore, the authors of AutoDock have included local search function, designed to discover local minima, which is called just before the genetic algorithm loops back to the mapping phase. The

genetic algorithm with the addition of the local search function is termed the Lamarckian genetic algorithm (Morris et al 1998), because the phenotype of each individual that undergoes the local search becomes reverse transcribed into that individual's genotype before the genetic algorithm resumes. A user-specified, random fraction of individuals undergoes the local search. If the local search function were implemented so that the coordinates of the atoms were directly perturbed by search steps in order to find a local minimum, the Lamarckian nature of the search demands the existence of an inverse map from the phenotype space to the genotype space. To circumvent the need for this inverse map, the search occurs by taking search steps in the genotype space (Morris et al 1998), rather than taking steps in the phenotype space as occurs in ICM.

The local search derives from an algorithm by Solis and Wets (Morris et al 1998), in which steps of a given size are taken, and the conformation that is to become the new conformation is evaluated with the energy function. The change in energy between the two conformations is calculated, as in the Monte Carlo algorithm. If the change is favorable it is accepted, otherwise the algorithm examines taking a step in the exact opposite direction. If that step is favorable, then it is accepted. Otherwise, a counter of the number of failures is incremented and the algorithm repeats. After a user-specified number of consecutive failures, the step size is doubled. Also, after a user-specified number of consecutive successful steps, the step size is halved. (Thus, the search adapts to its environment, consistent with the theme of evolution in these algorithms.)

The scoring function used by AutoDock is based on an estimate of the free energy of binding, derived from the sum of independent contributions from a van der Waals term, a Hydrogen-bond term, a screened Columbic electrostatic term, an entropic term based on torsional strain, and a solvation term (Morris et al 1998).

Conclusion

The literature seems mixed as to which docking algorithm(s) perform the most efficiently and accurately. Because of the impact of the scoring function on the results, the ability of an algorithm to dock ligands correctly is dwarfed by the importance of a robust scoring function, and a combination of independent scoring functions outperforms any single scoring function (Charifson et al 1999). Below, I comment on some of the weaknesses in the docking and scoring algorithms described above.

All docking algorithms are based on minimizing some sort of energy function (Halperin et al 2002), and thus the accuracy and broad applicability of each term in the energy function is important for proper docking, regardless of which algorithm is used. Over-training the parameters of the energy function to specific cases or families can cause the function to fail when applied to novel data. For the DOCK algorithm, which relies heavily on shape matching, water molecules involved in any ligand-protein interaction are not taken into account. For both the Monte Carlo (ICM) and the Genetic/Lamarckian Genetic (AutoDock) algorithms, the search for a global minimum may fail, instead yielding a local minimum. Moreover, even if the global minimum is attained, it may not reflect the *in vivo* binding because either the ligand does not have enough time to find its global minimum or energy barriers that are too steep to be overcome at physiological

temperature are overcome during the simulation. For all of the docking algorithms, the ligand binding pocket might be modeled incorrectly, or the user may have selected the wrong region of the surface as the binding pocket. And, especially for the DOCK algorithm, the ligand may not be flexible enough to accurately simulate *in vivo* binding. Finally, some conformation change in the protein may be necessary for docking to occur, and DOCK and AutoDock do not permit target flexibility.

If docking fails, scoring is meaningless (Abagyan et al 2001). But if docking succeeds in correctly prediction the conformation of the ligand-target complex, then the scoring function must accurately sort binders from non-binders. Many scoring functions are over-trained to certain data, especially knowledge-based scoring functions, which consult tables of how often particular interactions occur in complexes of known structure (Abagyan et al 2001). If a scoring function is based on free energy, over-fitting the function's parameters to specific cases can cause the function to fail. The accuracy of terms in scoring functions is critical to suppress false positives. Most scoring functions don't explicitly account for water molecules (Halperin et al 2002), and dielectric constants for Columbic electrostatic interactions are difficult to predict.

Bibliography

Abagyan R, Totrov M. Biased Probability Monte Carlo Conformational Searches and Electrostatic Calculations for Peptides and Proteins. *J Mol Biol*. 235: 983-1002

Abagyan R, Totrov M. High-throughput Docking for Lead Generation. *Curr Opin Chem Biol*. 2001 5(4): 375-82

Brive L, Abagyan R. Computational Structural Proteomics. *Ernst Schering Res Found Workshop*. 2002 38: 149-66

Bissantz C, Folkers G, Rognan D. Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations. *J Med Chem*. 2000 43(25): 4759-67

Charifson PS, Corkery JJ, Murcko MA, Walters WP. Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. *J Med Chem*. 1999 42: 5100-09

Halperin I, Ma B, Wolfson H, Nussinov R. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins*. 2002 47(4): 409-43

Hol WG. Structural genomics for science and society. *Nat Struct Biol*. 2000 7: 964-6

Morris GM, Goodsell DS, Halliday RS, Huey R, Hart WE, Belew RK, Olson AJ. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function *J Comp Chem*. 1998 19(14): 1639-62

Shoichet BK, Bussiere DE. Macromolecular crystallography and lead discovery: possibilities and limitations. *J Mol Biol*. 2000 295: 337-356

Willis RC. 2001: A Dock Odyssey. *Modern Drug Discovery*. 2001 4(9):26-32