A Comparison of Global and Local Search Methods in Drug Docking

Christopher D. Rosin
University of California, San Diego
crosin@cs.ucsd.edu

R. Scott Halliday
Hewlett Packard
roberth@sdd.hp.com

William E. Hart
Sandia National Labs
wehart@cs.sandia.gov

Richard K. Belew
University of California, San Diego
rik@cs.ucsd.edu

Abstract

Molecular docking software makes computational predictions of the interaction of molecules. This can be useful, for example, in evaluating the binding of candidate drug molecules to a target molecule from a virus. In the Autodock docking software [11], a physical model is used to evaluate the energy of candidate docked configurations, and heuristic search is used to minimize this energy. Previous versions of Autodock used simulated annealing to do this heuristic search. We evaluate the use of the genetic algorithm with local search in Autodock. We investigate several GA-local search (GA-LS) hybrids and compare results with those obtained from simulated annealing. This comparison is done in terms of optimization success, and absolute success in finding the true physical docked configuration. We use these results to test the energy function and evaluate the success of the application.

1 The Docking Problem

When two molecules are in close proximity, it can be energetically favorable for them to bind together tightly. The molecular docking problem is the prediction of energy and physical configuration of binding between two molecules. A typical application is in drug design, in which we might dock a small molecule that is a proposed drug design to an enzyme we wish to target. For example, HIV protease is an enzyme in the AIDS virus that is essential to its replication. The chemical action of the protease takes place at a localized active site on its surface. HIV protease inhibitor drugs are small molecules that bind to the active site in HIV protease and stay there, so that the normal functioning of the enzyme is prevented. Docking software allows us to evaluate a drug design by predicting whether it will be successful in binding tightly to the active site in the enzyme. Based on the success of docking, and the resulting docked configuration, designers can refine the drug molecule.

1.1 Autodock

Autodock docks small flexible substrate molecules to large rigid macromolecules (such as proteins) [4, 11]. A candidate docking gives specific positions and orientations for the protein and small molecule. Autodock uses an approximate physical model to compute the free energy of a candidate docking, and uses a heuristic search to minimize this energy. This method makes most sense when there is a single docked configuration that is at a much lower energy than other configurations, so that we expect this low-energy configuration to be the consistent result of physical interaction between the two molecules. If the prediction of this configuration is to be accurate, the energy function must have its global minimum at or near this physical configuration.

Heuristic search operates on the configuration of the small molecule, assuming (without loss of generality) a fixed position for the protein. The small molecule can take any position around the protein, and can have any orientation. Global orientation is expressed as a quaternion, which can be thought of as a vector giving an axis of rotation, along with an angle of rotation about this axis. The small molecule may also have several internal rotatable bonds so that its shape is somewhat flexible. The representation of a candidate docking consists of 3 coordinates giving the position of the small molecule, followed by the 4 components of the quaternion specifying the overall orientation of the small molecule, followed by one angle for each of the rotatable bonds.

The energy evaluation in Autodock takes a specified candidate docking and begins by calculating the absolute position of each atom in the small molecule. The total energy is the sum of the energy contribution from each atom in the small molecule. The energy contribution of an atom is obtained by summing the potential energy resulting from interactions between it and each atom in the fixed macromolecule. The pairwise interatomic potentials used in this sum account for short range repulsive forces and long-range weak van der Waals attractive forces, using a Lennard-Jones 12-6 potential:

$$\frac{C_{12}}{r^{12}} - \frac{C_6}{r^6}$$

where r is the distance between the atoms. C_{12} and C_6 are chosen appropriately for the atom types involved in the pairwise interaction (for example, carbon with nitrogen or hydrogen with oxygen); see [12] for details. Hydrogen bonds use a longer-range 12-10 potential. Finally, an additional contribution is added for electrostatic interactions. Due to bond asymmetries, atoms in a molecule act as if they have partial charges. Electrostatic potentials are based on these partial charges.

To account for internal energy in a flexible small molecule with internal rotatable bonds, we calculate the same energy contributions summed over all pairs of atoms within the small molecule. This sum is added to the total energy evaluation. This helps discard conformations of the small molecule that are energetically unfavorable independently of interaction with the macromolecule.

To save time when computing energy of interaction with the macromolecule, 3-D potential grids are computed for each atom type before optimization begins. Interaction energy is computed as described above at each point in the grid. Then, when calculating total energy during optimization, the energy contribution of an atom is obtained via trilinear interpolation of its position within the grid specific to its atom type, based on the values at the nearest 8 points in the grid. Energy due to pairwise interactions within the small molecule does not make use of these grids.

Computation of the grids for energy evaluation requires knowledge of the (assumed fixed) 3-D positions of each atom in the protein; these positions are usually obtained by X-ray crystallography. We also require the structure of the small molecule, along with the locations of internal rotatable bonds. Small molecules tend to be chemically simple, so that we can determine their structure (at least up to the degrees of freedom represented by the rotatable bonds) from their chemical composition alone. Partial charges are required to calculate electrostatic interaction potentials, but these partial charges can be computed from the structure with molecular modelling software such as MOPAC [10]. So, it is possible to use Autodock to test many candidate small molecules against a single target protein, after obtaining the structure of this protein experimentally. This makes Autodock an important computational tool in the initial stages of drug design.

1.2 Prior Work

Molecular docking has received much attention in computational biology. The DOCK program [8] was one of the first approaches to this problem, and current versions of it are still used. It attempts to find binding sites based on physical properties of the docking molecules, without doing a complete heuristic search of possible docked configurations. More recent efforts often employ an energy function such as that described for Autodock, and use heuristic search to minimize energy. The genetic algorithm is becoming a popular choice for the heuristic search method in docking applications; see [1] for references.

Earlier versions of Autodock used simulated annealing for heuristic optimization [11, 4]. There have been several successful applications of Autodock with simulated annealing [3, 9, 15, 14], and it is a good testbed for comparison with the genetic algorithm because of the effort that has gone into optimizing simulated annealing parameters. A preliminary comparison of genetic algorithms and simulated annealing in Autodock appears in [5].

2 Global and Local Search Methods

Docking is a difficult optimization problem, and successful search requires efficient *local* search of each attractor basin, as well as effective *global* sampling across the entire range of possible docking orientations. Earlier versions of Autodock relied exclusively on an optimized variant of simulated annealing. Simulated annealing can be viewed as including both global and local search aspects, depending on whether it rejects a (locally) inferior alternative or jumps (globally) to a new, random location. It does a more global search early in a run, when high temperature allows transitions over energy barriers from one valley to another. Late in a run, low temperature places more focus on doing local optimization in the current valley. In the simulated annealing implementation used here, we use uniform deviates to do extensive exploration. During the run, deviate sizes are reduced. This is consistent with doing global sampling early in the run and focused local search late in the run.

Here we also test GAs that explicitly include a local search operator: the GA is used to allocate samples across the entire search domain, but (some fraction of) these also go on to perform a local search. An important distinction is between *Lamarckian* and *non-Lamarckian.* In Lamarckian GA-LS hybrids, the genetic representations of individuals' local search *starting points* are replaced by the *results of this local search.* This requires the use of a (biologically unrealistic!) mapping from the "acquired, phenotypic" end-point of local search back to the genetic representation used by the GA. In Autodock, the GA and local search operate on the same representation, so this mapping is trivial. Our previous work [6] leads us to use Lamarckian GA-LS hybrids for the experiments presented here.

In a GA-LS hybrid, mutation plays a somewhat different role than it does in a GA without explicit local search [5]. Without an explicit local search operator, it must be the mutation operator that makes small, refining moves that are not efficiently made using crossover and selection alone. With an explicit local search operator, however, the local-refinement requirement becomes unnecessary. Mutation is still necessary for its other role in the GA: replacing alleles that might have disappeared. With only this function, mutation can take on a more exploratory role. Following Hart, we use Cauchy deviates for mutations in our GA-LS hybrids [5]. Cauchy deviates are a compromise between radical jumps to completely arbitrary sections of the solution space and exploring the local richness of the current point.

There are good reasons to believe that global-local search hybrids may perform better on optimization problems than either type separately [5, 7]. This is due in part to the smoothing performed by local search. Global search methods are good at identifying promising areas of the solution space, while local search methods do well at refining a good solution. When combined in a non-Lamarckian way, this can be seen as transforming the landscape. That is, a point x in the original space is mapped to $x*$, the local minimum with x in its neighborhood. The global operator then manipulates this, an operation it's fairly good at.

The sort of global-local hybrid search performed by a

GA-LS hybrid seems potentially more powerful than the simple combination of global and local search performed by simulated annealing. SA focuses on local search late in a run and loses its ability to make long jumps, whereas a GA-LS hybrid can do global search along with careful local search as long as its population has not converged. Furthermore, SA executes a search corresponding to a single individual's search. SA is therefore unable to compare search results across an entire population, as the GA does. Nor can it combine information across multiple local search basins, as the GA does when crossover combines mating individuals.

Global-local search hybrids may be especially effective for docking. We believe that there are multiple locations on the surface of the macromolecule where the small molecule could dock, and multiple orientations of the small molecule that are energetically plausible. Local search can reveal which of these locations and orientations is best, by fitting the small molecule as closely as possible to the macromolecule within a small local neighborhood of a coarse location and orientation. But we do not expect smooth hills in energy from one location and orientation to a very different one, so that global search is required to choose among these. A global-local search hybrid can effectively sample distant choices of location and orientation with global search, and get accurate evaluations of each of these using local search.

2.1 Simulated Annealing

In the original version of Autodock, simulated annealing was the only optimization method available. For this paper, it was run one hundred times on each test case using between 1.5 and 1.8 million function evaluations each time. The implementation was tuned carefully based on prior experience with Autodock. A run consists of 50 cycles, with each cycle terminating after 25000 moves are accepted or 25000 moves are rejected (whichever comes first). Temperature is 616 cal mol^{-1} during the first cycle, reduced linearly after each cycle so that it reaches 0 on the last cycle. Each cycle begins with the lowest-energy configuration from the previous cycle. Translational maximum step sizes are reduced from 3.0 to 0.2 angstroms linearly during from first cycle to last cycle, and angle maximum step sizes are similarly reduced from 24 to 5 degrees during the run [11].

2.2 Solis and Wets' Algorithm

Solis and Wets describe a class of local and global search algorithms [13] with proofs of convergence in the limit of infinite search time. Following Hart [5], we use a Solis & Wets local search algorithm in our GA-LS hybrid. This local search algorithm is a randomized hillclimber with an adaptive step size. Each step starts with a current point x. A deviate d is chosen from a normal distribution whose standard deviation is given by a parameter ρ. If either $x + d$ or $x - d$ is better, a move is made to the better point and a "success" is recorded. Otherwise a "failure" is recorded. After several successes in a row, ρ is increased to move more quickly. After several failures in a row, ρ is decreased to focus the search. Additionally, a bias term is included to give the search momentum in directions that yield success. The pseudocode for our implementation of Solis & Wets local search is given in Figure 1.

```
Choose initial point x
Set b (bias vector with dimensionality of search space) to 0
Initialize ρ

while max iterations not exceeded and ρ not too small
    for each dimension i of the solution space
        add deviate D_i = normal deviate with mean b_i
                and standard deviation ρ

    if new solution is better
        failures = 0
        successes = successes + 1
        b = 0.4D + 0.2b
    else
        for each dimension i
            add −D_i to the original solution
        if new solution is better
            failures = 0
            successes = successes + 1
            b = b − 0.4D
        else
            failures = failures + 1
            successes = 0
            b = 0.5b

    if successes >= max successes
        failures = 0
        successes = 0
        ρ = expansion factor *ρ

    if failures >= max failures
        failures = 0
        successes = 0
        ρ = contraction factor *ρ
```

Figure 1: Solis & Wets local search

An important feature of this type of local search is that it doesn't rely on gradient information. This is useful in cases like Autodock where gradient information is non-existent, unreliable, or poorly specified. Autodock's grid-based intermolecular energy has a gradient that is undefined whenever an atom is on a grid boundary, and has discontinuities as atoms move across grid boundaries. This would make gradient-based local search a poor choice for Autodock.

2.3 Genetic Algorithm and Global-Local Search Hybrids

We combine both GA and local search techniques into hybrids which perform a generation of the GA followed (at some frequency) by (some number of iterations of) local search [5]. The GA-LS hybrids here are Lamarckian. We have previously illustrated how applying local search to a small fraction of the population can sometimes improve the efficiency of GA-LS hybrids [5]. The GA-LS hybrids used on the docking problems apply local search to 7% of the

Ligand/Protein Complex	PDB Shorthand	# of torsions	# of Dimensions
β-Trypsin/Benzamidine	3ptb	0	7
Cytochrome P-450cam/Camphor	2cpp	0	7
McPC-603/Phosphocholine	2mcp	4	11
Streptavidin/Biotin	1stp	5	12
HIV-1 protease/XK263	1hvr	10	17
Influenza Hemagglutinin/sialic acid	4hmg	11	18

Table 1: Test cases summary

population in each generation. Local search is done just before each GA generation (except for the initial one).

We use a fairly standard genetic algorithm with stochastic remainder selection. Fitness is scaled inversely to the free energy of the conformation, so that we are minimizing energy. The worst fitness is tracked each generation, and an average of this worst value over the last 10 generations is maintained. This average worst value is subtracted from fitness before reproduction, so that an individual with fitness equal to or less than this worst receives 0 offspring. Two-point crossover is used. On the genome, global position is first, followed by the quaternion specifying global orientation, followed by the torsion angles. Since each gene is real-valued, the standard bit-flip type mutation is no longer appropriate. In this implementation, mutation is performed by adding a Cauchy deviate to the particular gene to be mutated [5]. The Cauchy distribution has the form:

$$C(x; \alpha, \beta) = \frac{\beta}{\pi\beta^2 + (x-\alpha)^2}$$

3 Test Methods and Experiments

A test suite of six cases was used in all of the experiments. Each test case consists of a macromolecule and a small substrate molecule. The salient features of the six test cases are summarized in Table 1. The different test cases were selected to test various aspects of the energy function [11].

The number of torsions is very important because it sets the dimensionality of the search space. The representation used in each experiment consisted of a triple of Cartesian coordinates, a four dimensional quaternion, and the torsions. So, the dimensionality of the search space is 7+(number of torsions). The initial individuals had each parameter generated randomly in its range. Position coordinates are expressed in angstroms and are constrained to lie within the grid (23 angstroms long on each side of the cube), and angles and quaternion coordinates lie in the interval $[-100, 100]$ (angles are in radians). The main effect of these ranges is to set the relative size of jumps taken; in this representation, a 0.1 radian rotation is comparable to a 0.1 angstrom translation.

For each of the experiments, unless otherwise mentioned, the genetic algorithm operated on a population of 50 individuals, used a (two-point) crossover rate of 80%, and a mutation rate of 2%. Mutation took the form of Cauchy deviates with $\alpha = 0$ and $\beta = 1$. Simple elitism was also used.

When GA-LS hybrids were used, local search was performed with 7% frequency. The local search operator was Solis & Wets using normal deviates. The initial value of ρ was 1, and the contraction and expansion factors were, respectively, 0.5 and 2.0. The lower bound on ρ was 0.01. After four consecutive successes, ρ was expanded. After four consecutive failures, ρ was contracted.

To allow comparisons, each GA run was limited to 1.5 million function evaluations (the lower end of the number of evaluations required for simulated annealing). This is so that each run used just as much information and took roughly the same amount of time. Using function evaluations provides the best metric. This is because the energy evaluation function is computationally expensive and is the major determinant of how long optimization will take.

We tested several versions of the genetic algorithm on this problem, and compared results to those obtained with simulated annealing. As a baseline, we used a genetic algorithm without local search. Several of the molecules with multiple torsion angles have a branched structure, so that the torsion angles do not have a natural linear ordering to use for crossover. Because of this, we also tried the genetic algorithm without crossover (with mutation rate raised to 20%).

Adding local search, we tried Solis & Wets for 3000 steps, applied with 7% frequency to each generation's population. It may not be necessary to do such a complete local search since partial local searches can give a good approximation to the fitness that a full search would yield, and Lamarckian search may make it unnecessary to do a complete local search every time. So, we tried Solis & Wets for only 30 steps as well.

For each of the GA-based search methods, 20 runs were done on each test case. Simulated annealing was run 100 times on each test case. For each method on each test case, we consider the minimum energy produced by the search, averaged over all runs. Because we have crystallographic structures of the true docked complex for each test case, we can also measure the absolute accuracy of the final docked configuration. This is done by taking the square root of the average squared deviation of the atoms in the predicted configuration from the crystallographic configuration.

Note that because we hold population size constant, the runs that use a long local search get far fewer generations. This is a concern because if many generations are required, the runs with long local search are hampered. If many generations are not required, the runs without long local search are wasting time. Preliminary results, however, indicated that neither number of generations nor

	SA	GA-NoXover	GA	GA-SW30	GA-SW3000
SA	X	****-*	***--*	******	******
GA-NoXover		X	**---*	******	******
GA			X	*****-	******
GA-SW30				X	-*----
GA-SW3000					X

Table 2: Significant Differences ($p < 0.05$) in Energy

	SA	GA-NoXover	GA	GA-SW30	GA-SW3000
SA	X	*****-	**-***	-*****	-*****
GA-NoXover		X	-----*	-*****	--****
GA			X	-*****	-*****
GA-SW30				X	------
GA-SW3000					X

Table 3: Significant Differences ($p < 0.05$) in RMSD

local search length is an overriding determiner of success. This suggests that holding the population size constant is reasonable (or as reasonable as other possible choices) for these experiments.

For notational convenience, throughout this paper we will use several shorthands. "GA" will refer to a genetic algorithm with standard settings. "SW3000" will refer to the Solis & Wets algorithm with standard settings and performing a maximum of 3000 iterations during local search. "SW30" refers to the Solis & Wets algorithm with standard settings and performing a maximum of 30 iterations during local search. For the GA-LS hybrids, these tags are concatenated, e.g. GA-SW30 refers to a genetic algorithm - Solis & Wets (with at most 30 local search iterations) hybrid.

4 Results

Figure 3 graphs results for each of the search methods. Each graph has a bar for each method, showing its average performance and one standard deviation from this average. Test cases are ordered from top to bottom by increasing number of torsions. The left column of graphs shows energy, and the right column shows RMS deviation; in both cases, shorter bars are better.

The nonparametric Kruskal-Wallis test was used to verify that there exist significant differences among the optimization methods (at the 5% level), in both energy and RMSD, for each test case. A directional post hoc comparison was used to locate these significant differences at the 5% level of significance [2]. This rank-based test gives an indication of which of two compared methods would be most likely to give the better result if a single run of each was performed. The significant differences in energy and RMSD are shown in Tables 2 and 3. Each table entry has 6 symbols, showing a "*" if the column method is significantly better than the row method, or a a "-" for no significant difference, for each of the 6 test cases in the order they are shown in Figure 2.

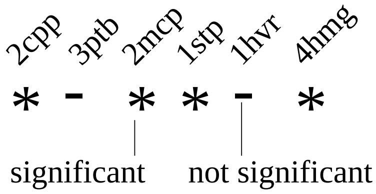

Figure 2: Significance Table Entries

5 Discussion

Even without local search, the GA (both with and without crossover) performed better than SA on all but one test case, with most performance improvements being significant. When augmented by local search (GA-SW30, GA-SW3000), the GA does significantly better than SA on all 6 test cases. GA-SW3000 always had the best average energy, and always had average RMSD at least close to the best. The RMSD of GA-SW3000 was significantly better than that of SA on the hardest 5 out of 6 test cases, showing that the substantial improvement in search also leads to an improvement in true performance, despite approximations made in the energy calculation.

Any RMSD below 0.5 angstroms is considered excellent. GA-SW3000 beats this standard or comes close to it on most problems. Even on the largest problem, RMSD was still only around 1 angstrom for GA-SW3000. These results indicate that Autodock is largely successful at predicting docked complexes when it uses a powerful search technique.

The genetic algorithm with crossover beats the genetic algorithm without crossover on 4 out of 6 test cases, with the rank test showing significant superiority on 3 out of 6 test cases[1]. But crossover does not yield a major improve-

[1] For 3ptb, the rank test shows crossover to be superior even

Test	Mean Energy	Energy S.D.	Mean RMSD	RMSD S.D.	Lowest Energy	RMSD at Lowest
2cpp	-36.90	0.05	0.45	0.09	-36.98	0.57
3ptb	-50.29	0.26	0.17	0.04	-50.80	0.20
2mcp	-45.90	0.84	0.71	0.14	-47.01	0.68
1stp	-53.61	0.38	0.38	0.14	-54.79	0.47
1hvr	-103.08	0.32	0.30	0.06	-103.98	0.35
4hmg	-47.11	0.29	0.24	0.04	-47.69	0.21

Table 4: Results of pure local search on the crystal structure

ment in performance that is consistent across all test cases. Crossover's role in docking may be limited due to the small number of genes, and the concern described above about linear orderings of torsion angles. But the GA with crossover performed better than without on the two problems with the largest number of torsions, so crossover may still play an important role in complex docking problems. Since the genetic algorithm with crossover did do significantly better on the largest problem, and didn't do vastly worse on any problem, the choice of including crossover for the remaining experiments is reasonable.

Adding local search to the GA produced substantial improvements in optimization performance. GA-SW3000 had significantly better energy than the GA on all test cases. Without local search, GA performance was very inconsistent on several test cases, yielding a large standard deviation in resulting energy. GA-LS hybrids gave more consistent results with a lower standard deviation.

GA-SW3000 outperformed GA-SW30 on all test cases. By adding local search to the GA, we hope to get solutions that are locally optimal on a fine scale. GA-SW30 may be unable to provide this. Solis & Wets local search starts with a large value of ρ and gradually decreases it. With a short local search, ρ never has a chance to become small, so fine-scale local optimization may never be done.

Better performance can be obtained from simulated annealing with more computation time. Individual SA runs can give unreliable results, but performance is often much better if we take the *best* energy from the 100 SA dockings on each problem. GA-SW3000 performance is more consistent than that of SA from run to run, in addition to being better on average. So, we expect that a very small number of runs of GA-SW3000 on new docking problems will give us high confidence that we have accurate results. With simulated annealing, we may do many runs and still be unsure that the next run won't yield a much lower energy. For the largest two problems, *average* GA-SW3000 performance is still substantially better than *best* SA performance, showing limits to performance gains realized by doing more restarts of an inferior optimization method.

6 Evaluating the Evaluation Function

Even if we had a perfect search method, we have to account for the accuracy of the energy function when evaluating the success of Autodock. For improved search performance to result in improved docking accuracy, lower-energy dockings must have smaller deviations from the true structure.

though its average energy was worse due to a single very bad run.

Looking at the correspondence between energy and RMS deviation produced in the experiments, we see that successful runs that produce low-energy dockings generally have good RMS deviations from the crystallographic configuration. Minima with energy closer to the best energy found for the landscape generally correspond better to the correct structure. This correlation breaks down somewhat at the lowest energies. For example, GA-SW3000 always produced the best average energy, but did not always produce the best average RMS deviation from the crystallographic structure. Methods that produced adequately low energies typically had reasonable agreement with the crystallographic structure. But because there is not a detailed correlation between energy and RMS deviation at the best energies, it seems that better optimization will not produce greater physical accuracy without improvements to the energy function's accuracy.

As another way to examine the energy landscape, for each of the six test cases, the Solis & Wets algorithm was run for a maximum of 300 iterations on a population of 50 individuals, each seeded with the crystal structure. The initial ρ was set at one, and the lower bound was set at 0.01. By locally optimizing the crystal structure, we should get some idea of how closely the "real" solution matches the nearest minima predicted by the energy function. Table 4 gives the average and lowest energies, and the average RMSD and RMSD at the lowest energy, for each of the test problems.

The average RMSD of nearby minima is usually within the 0.50 angstrom standard. Most of the lowest energies also beat the 0.50 RMSD standard, and the two exceptions follow close behind. These results indicate that, while the crystallographic structure is not at a local energy minimum, the surrounding local minima have a structure that is close to being correct. These surrounding local minima are not global minima: GA-SW3000 often found lower energies than these. But the structures corresponding to these lower energies are also close to the correct structure. This is sufficient for the application to be successful, since it means that successful optimization of the model energy will produce physically accurate results.

7 Conclusions and Future Directions

GA-LS hybrids offer great performance advantages over simulated annealing on the molecular docking problem. The local search component plays a major role in this advantage. The accuracy of the energy function is adequate, so that low-energy minima have structures close to the true docked configuration. Combined with the efficient search performed by the GA-LS hybrid, the application is successful.

Work is currently underway (by Garrett Morris, David Goodsell, Ruth Huey, and Arthur Olson at the Scripps Research Institute) to add energies of solvation and desolvation to the evaluation function, and preliminary results suggest that this makes energies very accurate. In addition, future versions of Autodock will allow some flexibility in the protein. Currently, evaluations are done with rigid protein structures obtained from a docked complex. For some real proteins, structure changes during docking. For example, HIV protease has flaps that enclose the active site once a substrate is docked. By allowing flexibility in the protein, we may be able to eliminate the need for crystallographic structure of a docked complex (so that the structure of the undocked protein would suffice), and allow elements of protein structure to be fine-tuned to a particular small molecule during docking.

GA-LS hybrids can achieve accurate docking results in a small fraction of the time required by simulated annealing to achieve comparable accuracy. The time factor is very important for docking. Currently, Autodock requires 5-30 minutes on a fast workstation to do a run, but eventually docking should be fast enough to be used interactively during drug design. Also, we can consider doing automated drug design by evaluating candidates with a docking simulation. In this case, it is essential that docking be very fast so that many evaluations can be done in the outer loop search for molecular designs. We are currently looking at using a genetic algorithm for this outer loop.

Acknowledgements

Garrett Morris, David Goodsell, Ruth Huey and Arthur Olson wrote the simulated annealing version of Autodock at the Scripps Research Institute, and did the SA runs used in this paper [11].

References

[1] Clark, D.E., and Westhead, D.R. (1996). "Evolutionary algorithms in computer-aided molecular design." Journal of Computer-Aided Molecular Design 10:337-358.

[2] Freund, R.J., and Wilson, W.J. (1997). Statistical Methods. Academic Press.

[3] Goodsell, D.S., Lauble, H., Stout, C.D., and Olson, A.J. (1993). "Automated Docking in Crystallography: Analysis of the Substrates of Aconitase." Proteins: Structure, Function, and Genetics 17:1-10.

[4] Goodsell, D.S., and Olson, A.J. (1990). "Automated Docking of Substrates to Proteins by Simulated Annealing." Proteins: Structure, Function, and Genetics 8:195-202.

[5] Hart, W.E. (1994). Adaptive Global Optimization with Local Search. PhD thesis, Computer Science & Engineering Department - University of California, San Diego. ftp://ftp.cs.sandia.gov/pub/papers/wehart/thesis.ps.gz

[6] Hart, W.E., and Belew, R.K. (1996). "Optimization with Genetic Algorithm Hybrids that Use Local Searches." In R.K. Belew and M. Mitchell, editors, Adaptive Individuals in Evolving Populations, pp. 483-496. Addison-Wesley.

[7] Hart, W.E., Kammeyer, T.E., and Belew, R.K. (1994). "The Role of Development in Genetic Algorithms." In D. Whitley and M. Vose, editors, Foundations of Genetic Algorithms III, pp. 315-332. Morgan Kaufman.

[8] Kuntz, I.D., Blaney, J.M., Oatley, S.J., Langridge, R., and Ferrin, T.E. (1982). "A Geometric Approach to Macromolecule-Ligand Interactions." Journal of Molecular Biology 161:269-288.

[9] Lunney, E.A., Hagen, S.E. et al. (1994). "A Novel Nonpeptide HIV-1 Protease Inhibitor: Elucidation of the Binding Mode and its Application in the Design of Related Analogs." Journal of Medicinal Chemistry 37:2664-2677.

[10] MOPAC quantum chemistry software. Available from Quantum Chemistry Program Exchange, Indiana University.

[11] Morris, G.M., Goodsell, D.S., Huey, R., and Olson, A.J. (1996). "Distributed Automated Docking of Flexible Ligands to Proteins: Parallel Applications of Autodock 2.4." The Journal of Computer-Aided Molecular Design 10:293-304.

[12] Olson, A.J., Goodsell, D.S., Morris, G.M., and Huey, R. (1995). Autodock User Guide: Automated Docking of Flexible Ligands to Receptors, Version 2.4. Scripps Research Institute, Department of Molecular Biology. http://www.scripps.edu/pub/olson-web/doc/autodock/documentation.html

[13] Solis, F.J., and Wets, R.J-B. (1981). "Minimization by Random Search Techniques." Mathematical Operations Research 6:19-30.

[14] Stoddard, B.L., and Koshland, D.E. (1992). "Prediction of the Structure of a Receptor-Protein Complex Using a Binary Docking Method." Nature 358:774-776.

[15] Vara Prasad, J.V.N., Para, K.S. et al. (1994). "Novel Series of Achiral, Low Molecular Weight, and Potent HIV-1 Protease Inhibitors." Journal of the American Chemical Society 116:6989-6990.

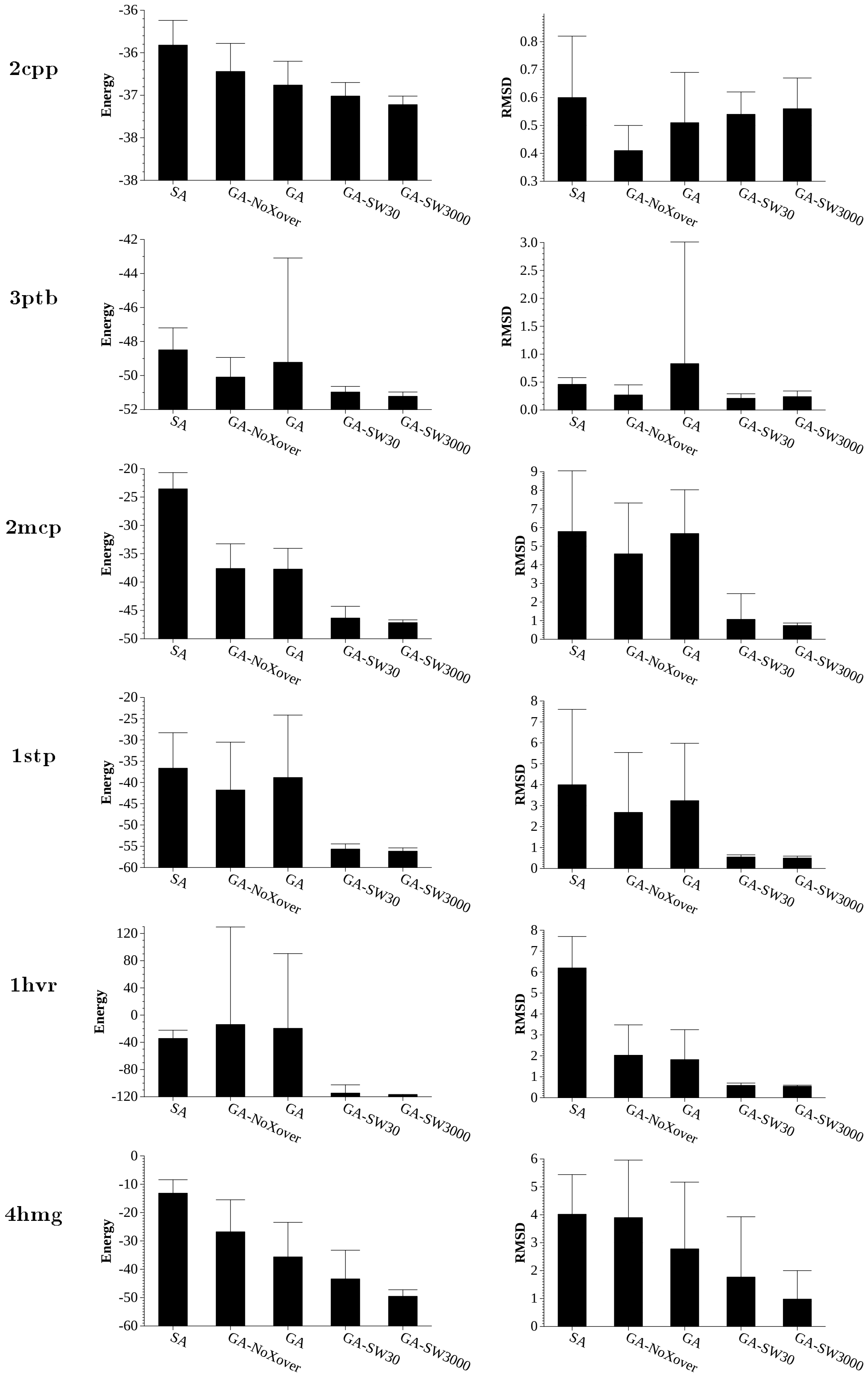

Figure 3: Average Energy and RMSD