MPI
INFORMATIK

An Algorithm for the Protein Docking Problem

Hans-Peter Lenhof

MPI–I–95–1-023 August 1995

FORSCHUNGSBERICHT ■ RESEARCH REPORT

MAX-PLANCK-INSTITUT
FÜR
INFORMATIK

Im Stadtwald ■ 66123 Saarbrücken ■ Germany

The *Max-Planck-Institut für Informatik* in Saarbrücken is an institute of the *Max-Planck-Gesellschaft*, Germany.

ISSN: 0946 - 011X

Forschungsberichte des

Max-Planck-Instituts für Informatik

Further copies of this report are available from:

Max-Planck-Institut für Informatik
Bibliothek & Dokumentation
Im Stadtwald
66123 Saarbrücken
Germany

An Algorithm for the Protein Docking Problem

Hans-Peter Lenhof

MPI–I–95–1-023 August 1995

An Algorithm for the Protein Docking Problem

Hans-Peter Lenhof
Max-Planck-Institut für Informatik
D-66123 Saarbrücken, Germany

Abstract

We have implemented a parallel distributed geometric docking algorithm that uses a new measure for the size of the contact area of two molecules. The measure is a potential function that counts the "van der Waals contacts" between the atoms of the two molecules (the algorithm does not compute the Lennard-Jones potential). An integer constant c_a is added to the potential for each pair of atoms whose distance is in a certain interval. For each pair whose distance is smaller than the lower bound of the interval an integer constant c_s is subtracted from the potential ($c_a < c_s$). The number of allowed overlapping atom pairs is handled by a third parameter N. Conformations where more than N atom pairs overlap are ignored. In our "real world" experiments we have used a small parameter N that allows small local penetration. Among the best five dockings found by the algorithm there was almost always a good (rms) approximation of the real conformation. In 42 of 52 test examples the best conformation with respect to the potential function was an approximation of the real conformation. The running time of our sequential algorithm is in the order of the running time of the algorithm of Norel *et al.* [NLW+]. The parallel version of the algorithm has a reasonable speedup and modest communication requirements.

1 Introduction

Docking reactions play an important role in a large number of biochemical processes. Although the mechanisms of docking reactions are not well understood, two complementarity principles seem to be important for the recognition and binding of docking partners. The first principle is the *shape complementarity principle*: The shapes of the molecules that build a docking complex are (locally geometrically) complementary, that is, there is a large fit between the surfaces of the docking partners. The second complementarity principle is the *chemistry principle*. It states that there is a strong chemical complementarity (with respect to hydrogen bonds, electrostatic interactions, hydrophobicity and so on) between the sites of docking partners.

Although the second principle is the more important one, it is possible to identify many docking sites solely with the help of the shape complementarity principle. In order to find these sites for two proteins A and B with n and m atoms (w.l.o.g. $n > m$), the following $3D$ matching problem has to be solved: Determine all transformations (rigid motions) of B such that there is a large fit between the surface of A and the surface of B and no penetration of B into the interior of A. We will call the parts of the surfaces that match for a special conformation the *common surface* of the conformation. For all candidates with a good geometric fit the potential energy difference of the docking conformation and the molecules A and B has to be

computed. The best candidates with respect to potential energy difference are possible docking conformations.

In the above description of the geometric $3D$ matching problem, two strong assumptions were made: (1) The two proteins are rigid. (2) There is no penetration of the rigid bodies. Of course, proteins are not rigid. They have certain dynamics that have strong influence on their chemical reactivity. Some parts of the molecules are very flexible, others are more or less rigid. Molecular dynamics simulations of some proteins indicate that the receptor or docking sites of the proteins are not very flexible. But "small" local changes of the shape of the docking sites happen during the docking reactions. Hence, if we work with rigid bodies, the algorithm is not allowed to ignore conformations with local overlappings. That means, we need a fitness function that can handle local penetration.

Fischer *et al.* [FNN+] and Lin *et al.* [LNF+] use the following measure for the size of the common surface: The algorithm computes for each protein a set of points on its contact surface [Con1, Con2]. The points of A are stored in a $3D$ grid. Boxes (voxels) of the grid that contain surface points are called *surface boxes*, boxes in the interior of the molecule *inner boxes* and boxes outside the molecule *outer boxes*. Given a transformation of molecule B, the algorithm computes for each surface point of B the box of the grid of A that contains the point. The algorithm counts the number of surface boxes that contain points of B. The number of such boxes (that contain points of A and B) is the measure for the size of the common surface.

In this paper we present a new approach for measuring the size of the common surface. Intuitively the idea behind this measure was motivated by the observation that the contact area of most docking complexes is densely packed with respect to the van der Waals hulls of the atoms. We count the number of *"van der Waals contacts"* between atoms of molecule A and atoms of (a conformation of) molecule B ($conf(B)$; A is fixed and B will is movable). Our fitness function is defined as follows:

$$\begin{aligned} FIT(conf(B)) \;:=\; & c_a * \#\{(a,b) \| a \in A, b \in B, d_l \leq d(a,b) \leq d_u\} \\ & - c_s * \#\{(a,b) \| a \in A, b \in B, d(a,b) < d_l\}. \end{aligned}$$

Here, $c_a < c_s$ are integer constants and d_l and d_u are distance parameters (default : $d_l = 2.85$ Å, $d_u = 4.0$ Å). The first part of the fitness function FIT counts the number of atom pairs that have a "van der Waals contact." The second part represents a negative score for the "overlapping" atom pairs. We presently do not take into account that atoms have different van der Waals radii, but we could easily refine our fitness function with a modest increase of running time and space requirements.

We call a conformation of B *feasible* if the number of overlapping atom pairs, i.e., pairs (a,b) with $a \in A$, $b \in B$ and $d(a,b) < d_l$, is less than a prescribed constant N. The goal is to compute the feasible conformations with the highest fitness values, say the top 1000 (or any other prescribed number).

In Section (2) we describe a data structure for molecule A that allows to approximately compute the fitness value of a fixed conformation of B and we also describe a technique to identify promising conformations for B. In Section (3) we show how to parallelize our algorithm. In Section (4) we present a few docking results. We have accurately docked proteins where other programs had difficulties. In 42 out of 52 tested examples the best element of the fitness list was close to the real conformation. In 4 out of 52 examples a good approximation of the real conformation was among the best five elements of the fitness list. As far as we know, there is only one other docking system that produces results of similar quality, namely the docking

system that is being developed at the GBF (Gesellschaft für Biotechnologische Forschung mbH, Braunschweig) in the group of Prof. Schomburg. It uses correlation techniques (see [KSE+]). The running times of this docking system seem to be much larger than the running times of our program.

The fitness function is only a rough measure for the potential energy contribution of the van der Waals interactions. We can give the measure a stronger chemical taste, by changing the constant c_a for pairs of atoms that can build hydrogen bonds. This can be realized with a modest increase of the computational requirements. Approaches to refine the modell and some future research directions will be dicussed in Section (5).

For other (geometric) protein docking techniques see [Con3, KSE+, KCF, KBO+, FNN+, HCT, EKS+].

2 The Sequential Algorithm

First, we describe the algorithm for the *fitness test*, i.e, the algorithm that computes the size of the common surface of a given conformation using the measure defined above. Second, we outline the technique for selecting a discrete set of conformations that will be tested.

For a point p define its *contact value*

$$\begin{aligned}\text{contact\_value}(p) \;:=\; & c_a * \#\{(a,p) \| a \in A, d_l \leq d(a,p) \leq d_u\} \\ & -c_s * \#\{(a,p) \| a \in A, d(a,p) < d_l\},\end{aligned}$$

i.e, the contact value of p is simply the value of our fitness function for a molecule consisting of a single atom which is placed at point p of the three-dimensional space.

We describe two data structures that allow to efficiently determine an approximation of the contact_value(p) for any p. The second data structure is faster than the first, but uses more space. For both data structures a $3D$ grid that contains molecule A is computed. The boxes of the grid have a side length of 4 Å. If all points in a box have the same contact value, then we store the contact value with the box. Otherwise we store the value "Undefined" and a pointer to a local data structure for this box. The two data structures for the fitness test differ by the local data structure that is added to boxes with value "Undefined." In the first data structure this local data structure is a simplified octree [FVF+] with a constant number (default:4) of hierarchy levels. The leaves of the octree store the maximum of the contact values of the eight corners of the corresponding cube.

The second data structure has a $3D$ grid (array of contact values) as local data structure. The approximation of the contact-value that is stored for a cell of the grid is the maximum of the contact values of its eight corners. It enables faster tests, but requires more storage.

Given a conformation of A and B the fitness test can be carried out in the following way: For each atom of B we determine the box of the grid that contains the atom. If the value of the box is not "Undefined", then we add this value to the fitness function. Otherwise we search in the local data structure of the box for a smaller box that contains the atom and has a defined contact value. This value is added to the fitness value. The sum of all contact values is the fitness value of the conformation. Instead of considering all atoms of B, we compute only the contact values of the atoms of B that belong to the Connolly (contact) surface of B [Con1, Con2]. These atoms can be easily computed in a preprocessing step. The rationale behind only looking at atoms in the Connolly surface is that atoms of B that do not belong to the Connolly surface of B have contact value 0 in most feasible conformations.

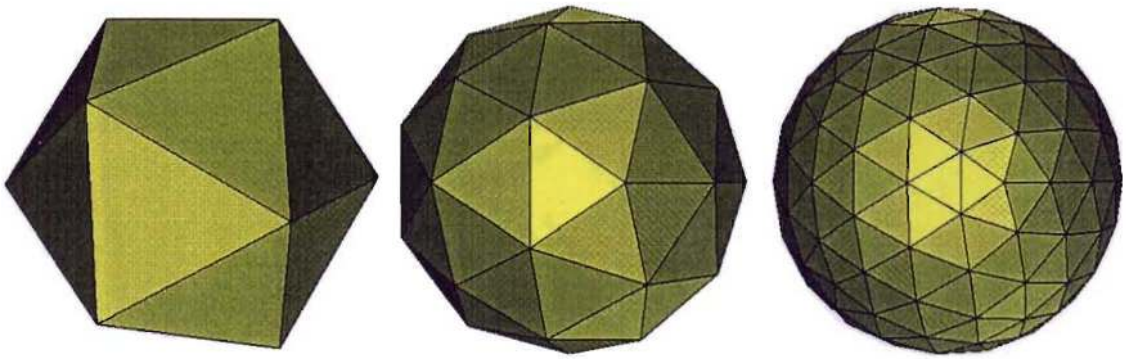

Figure 1: An icosahedron and the first two recursive refinements.

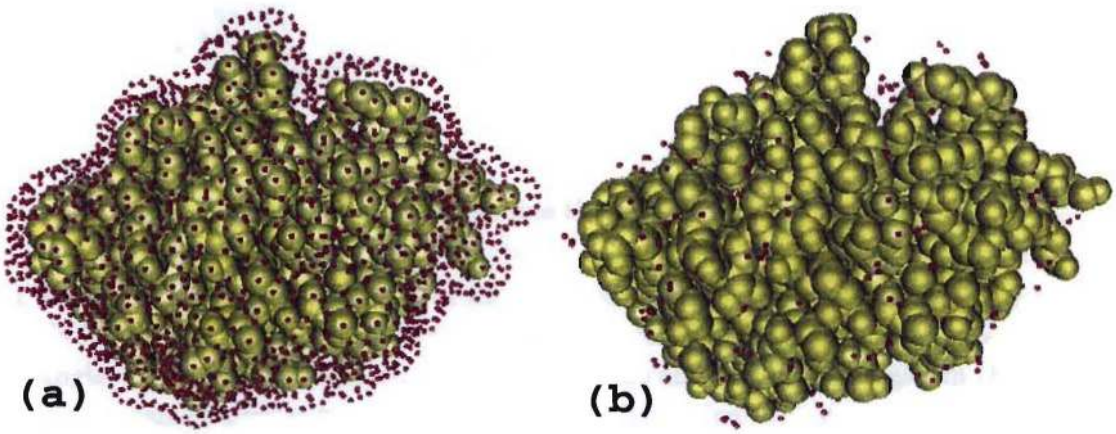

Figure 2: (a) All points on the probe sphere surface. (b) The points with contact value greater or equal to 12.

Now we describe the method for selecting the conformations that have to be tested: We compute an almost uniformly distributed point set on the surface of a sphere s. We can get such a point set by recursively refining an icosahedron (see Figure 1). For our purpose we take a sphere of radius 3.5 Å.

For each atom a of A we carry out the following test: We move the center of the sphere s to the center of atom a. For each point of the discrete surface point set of sphere s the algorithm checks if the point belongs to the so called *probe center surface*. A point belongs to this surface if the smallest distance to any atom in $A - a$ is greater or equal to 3.5 Å. We store all the points that belong to the probe center surface in a list L. For each point p in the list L contact_value(p) is computed. We select the points with "large" contact values (default: ≥ 12) and store them in a second list BL (see Figure 2). The points that have such large contact values are usually located in invaginations of the surface of A.

Using geometric hashing [LW] the set of test transformations can be computed as follows (see Figure 3): We compute all triangles between points of BL, whose side lengths are larger than a lower bound l_l and smaller than an upper bound l_u, and store them in a hash table H.

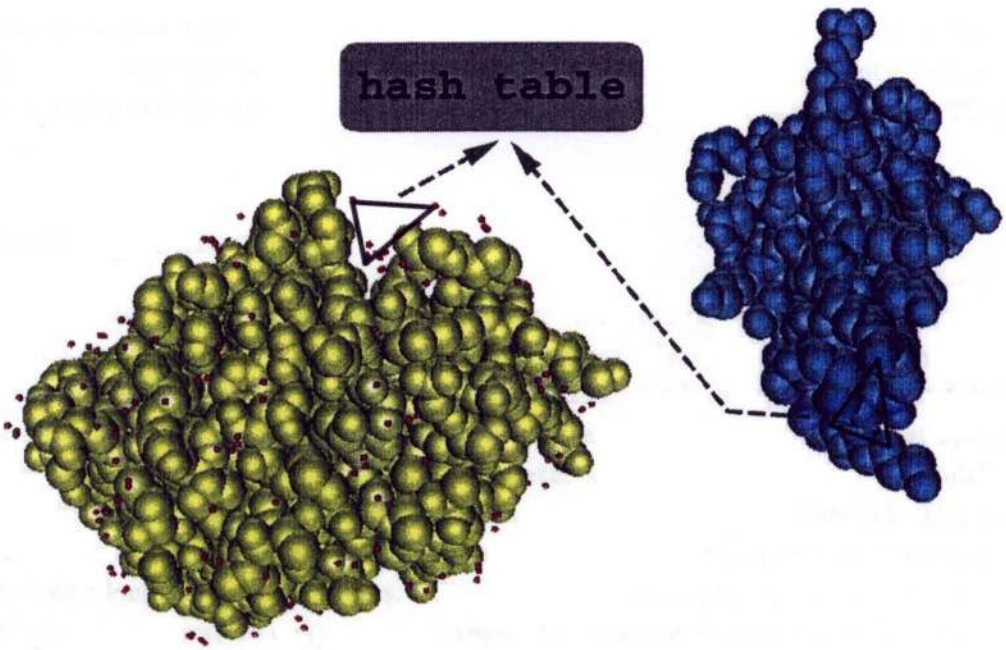

Figure 3: How to determine the transformation test set.

Then we do the same for the centers of the atoms of B that belong to the Connolly surface of molecule B, i.e., we compute all triangles that fulfill the above length conditions. For each of the triangles between atom centers of B, we determine all "similar" triangles in the hash table H. For each pair of similar triangles (t_1, t_2) a transformation is computed, that maps t_1 onto t_2. Since the triangles are similar but not equal, there are different ways to map the triangles. We use the centers of gravity, the normals of the triangles and angle bisectors to determine (choose) a transformation.

Using local complementarity criteria we can reduce the number of transformations that will be tested.

3 The Parallel Algorithm

The sequential algorithm can be easily parallized, by splitting the list of fitness tests. A master processor distributes the work between a set of clients and coordinates the clients. Each client builds the data structure for fitness tests in a preprocessing step. After that the master processor informs the client which part of the transformation list he should work on, by sending him an integer i. This integer is the list number where the client should start. The client stops at $i+STEP$, where $STEP$ is a small integer. The client informs the master that he has carried out his work, by sending an integer. Either all work has been done — in this case the master informs the client that he should send his list of the best transformations — or there is a rest of the transformation list — then the master sends a new start number to the client. The master collects all results from the clients and computes a list of the best transformations. There is no communication between the clients. The message passing is handled by PVM routines [Sun].

By choosing a suitable small constant $STEP$, the load of the clients is well balanced, but

the communication overhead is still modest. The first version runs on a cluster of workstations with processors that have different performance values (SUN and SGI workstations). Hence, it is difficult to prove precisely how the speedup behaves, but our experience seems to imply that the speedup will be greater than 90 % for a small number of processors (< 32) (see also Section 5).

4 Docking Examples

We now summarize our "real world" experiments. We have tested 52 docking examples that can be found in the PDB. We have also tested dockings where B is a small ligand (see Table (1)). The best 1000 geometric dockings were optimized by a local optimizer that "shakes" the molecule (only translations, no rotations). Since we are still working on parameter optimization, we did three experiments with different parameter sets for some of the "difficult" docking examples. The best result of these three experiments can be found in Table (1). The parameter sets of the three experiments differ only in the sizes of the lower bound l_l and the upper bound l_u for the edge length of the triangles. All other parameters (c_a, c_s, d_l, d_u, N, tolerance for "similar" edge length and so on) did not change. The standard lower bound l_l is 6.5 Å and the standard upper bound l_u is 10.5 Å. The two other parameter sets are "small" = ($l_l = 4.0$ Å, $l_u = 9.5$ Å) and "large" = ($l_l = 8.5$ Å, $l_u = 12.0$ Å). Norel *et al* [NLW+] did only one experiment for each docking example. Their docking program did not carry out local optimizations that can significantly improve the docking results.

The algorithm did almost always determine good approximations of the real conformations. The algorithm did not dock 1LYM, that is there was no approximation of the real conformation among the 1000 elements of the score list. We have used the local optimizer to compute the "real fitness" of the natural conformation (see column 8). The large difference between the "real fitness" and the maximal fitness that has been found by the algorithm implies that there are a few thousand conformations with better fitness value than the real conformation (see Table (1)).

In almost all examples where the algorithm succeeded, an approximation of the real conformation was among the five best conformations, more precisely: The worst example was 4XIA. It was number 290 of the score list. If doubles are eliminated, its rank is below 50. The second worst was 2HFL, which was the 54th element of the score list. On the ranking list of Norel *et al* [NLW+] it was number 6792 (see Column 4 of Table (1)). Using our local optimizer we "shaked" the real conformation of 2HFL, in order to see what fitness values can be expected for approximations of the real conformation. The "real fitness" 723 (see column 8 of Table (1)) would be number 7 on the score list. 3HFM was number 13 (17637 on the list of Norel *et al*) and 4CPA was number 11 (161 on the list of Norel *et al*).

All other examples had approximations among the first five elements. In most examples the best geometric fit was an approximation of the real conformation, for instance 4SGB, which was number 13691 on the ranking list of Norel *et al.*

The results above show that the new fitness function is a promising new measure for the size of the docking sites. Many docking sites can be determined by shape complementarity, but there are examples (perhaps a large percentage of all docking examples) where the common surface of the docking complex is much smaller than the common surface of the best geometric fits. The best approximations of these difficult examples will have large ranks (> 1000). Since energy evaluations have to be carried out for all potential solutions, we cannot claim that these

Results obtained by our docking algorithm

PDB	# A	# B	no.	fit.	best	real	rms Å	[NLW+] (min)	dock. (min)	proc.
1AAR	601	601	4	492	528	576	2.82		4.25	6(136.5)
6ADH	2834	2835	1	1381	1381	1554	1.57		59.43	4(139.0)
3APR	2403	57	1	611	611	551	3.04		4.57	1(53.0)
2CCY	972	972	1	744	744	810	1.85		19.42	5(165.5)
1CHO	1750	400	1(2)	492	492	606	2.96	12.4(44.8)	3.21	2(59.5)
4CPA	2437	289	11(106)	517	574	604	3.31	5.4(11.9)	5.27	4(165.5)
5CSC	3303	3303	1	2651	2651	2752	2.12		496.32	5(161.5)
4CTS	3444	3444	1	3041	3041	3230	1.93		88.04	4(139.0)
3DFR	1294	81	1	906	906	960	1.31		1.23	1(53)
3EST	1822	31	1	469	469		1.55		1.33	1(53.0)
3FAB	1551	1683	1	1319	1319	1350	3.65		203.29	3(112.5)
4FAB	1695	1700	1	1149	1149	1367	3.76		58.56	4(139.0)
1FAI	1657	1663	1	1074	1074	1712	4.00		40.47	4(139.0)
2FB4	1710	1602	3	965	999	1585	2.86		40.58	3(112.5)
2FBJ	1683	1636	1	1443	1443	1562	1.91		60.19	3(112.5)
3GCH	1048	700	1	2067	2067	2201	1.25		12.19	4(165.5)
3GPD	2577	2577	1	1123	1123	1143	5.89		794.31	4(165.5)
	2577	2577	4	985	1123	1143	3.76		794.5	4(165.5)
2HFL	3227	1000	54 (6792)	646	767	723	3.16	52.6(230.8)	227.56	4(165.5)
3HPM	3295	1001	13 (17637)	649	777	602	1.59	60.5(275.4)	51.51	4(165.5)
3HLA	2189	829	1	1052	1052	1134	1.36		25.18	7(202.0)
4HVP	758	758	1(2)	1260	1260	1235	1.62	12.7(52.5)	23.31	1(53.0)
4HVP	1516	54	1	512	512	478	1.72		1.44	1(53.0)
	1516	54	3	493	512	478	0.69		1.44	1(53.0)
2KAI	1799	438	4	624	634	657	3.68		7.15	4(139.0)
2LTN	1786	1786	1	930	930	1106	3.01		27.50	7(189.5)
1LYM	1001	1001			661	235			31.59	4(165.5)
4MBN	1205	44	1	536	536	496	1.89		1.32	1(53.0)
2MCG	1606	1606	1	895	895	1205	1.75		108.30	4(165.5)
1MCP	1709	1692	1	1333	1333	1587	1.42		114.45	3(112.5)
2MCP	1720	1692	1	1397	1397	1544	1.73		48.26	4(139.0)
1MEE	1948	530	1	917	917	965	1.83		10.40	5(161.5)
2MHB	1178	1113	1(49)	408	408	659	2.16	34.0(174.2)	7.07	6(124.0)
1MVP	872	867	1	1066	1066	1324	3.13		16.50	5(161.5)
1PP2	946	946	1	989	989	1043	3.89		37.21	3(112.5)
2PTC	1629	454	1(161)	596	596	623	4.35	9.9(28.4)	4.32	2(59.5)
1P01	1391	27	2	344	344	281	1.67		0.59	1(53.0)
1P02	1351	23	1	351	351	246	1.52		1.04	1(53.0)
2RSP	890	890	1	1336	1336	1304	1.60		13.56	5(161.5)
3RUB	3471	1029	1	1225	1225	1790	2.82		32.48	5(161.5)
2SEC	1920	530	1	735	735	692	1.02		13.37	4(165.5)
3SGB	1310	380	4	607	630	620	1.28		6.25	4(165.5)
4SGB	1310	380	1(13691)	613	613	638	4.53	3.5(7.6)	4.34	4(165.5)
	1310	380	4(13691)	596	613	638	2.24	3.5(7.6)	4.34	4(165.5)
2SNI	1938	513	1(81)	636	636	643	1.28	12.4(39.2)	9.46	1(53.0)
1TEC	2004	522	1(95)	712	712	640	1.25	8.8(27.7)	5.00	5(114.0)
2TGP	1629	454	1(180)	665	665	661	3.65	8.4(22.8)	5.00	5(114.0)
1TGS	1646	416	1(552)	690	690	646	1.91	12.9(32.2)	4.22	5(114.0)
1TIM	1870	1870	1	1201	1201	1235	2.24		94.09	4(139.0)
3TIM	1889	1889	1	1385	1385	1454	2.44		81.58	6(184.0)
3TPI	1629	454	1	543	543	638	2.26		3.51	6(136.5)
4TPI	1629	471	1(11)	836	836	880	1.52	7.9(23.3)	10.28	1(53.0)
	1629	471	2(11)	786	836	880	3.78	7.9(23.3)	10.28	1(53.0)
	1629	471	3(11)	769	836	880	2.83	7.9(23.3)	10.28	1(53.0)
2TSC	2256	2146	1	2625	2625	2597	1.40		56.23	5(161.5)
2UTG	548	548	1	734	734	1032	2.22		4.11	6(136.5)
4XIA	3040	3040	290	700	3904	1337	3.44		1163.23	4(165.5)

Table 1: Columns: (1) PDB code of the molecular complex. (2) The number of atoms of A (without hydrogen atoms). (3) The number of atoms of B (without hydrogen atoms). (4) The rank of the best approximation of the real conformation (the ranking of the algorithm of Norel et al. [NLW+] is given in brackets). (5) The fitness value of the best approximation. (6) The fitness value of the best geometric fit. (7) The "real fitness" of the natural conformation, determined by the local optimizer. (8) The RMS deviation of the approximation in Å. (9) The sequential running time of the docking program of Norel et al. (the time to carry out the scoring) on a SUN SPARC ??. (10) Preprocessing+docking time of our algorithm. The sequential running times have been measured on a SGI POWER CHALLENGE M. The times for the distributed version have been measured on a non-homogeneous workstation cluster. (11) The number of processors (since we are working on a non-homogenous workstation cluster, we computed a performance number for each processor; total performance value = sum of the performance values of the processors).

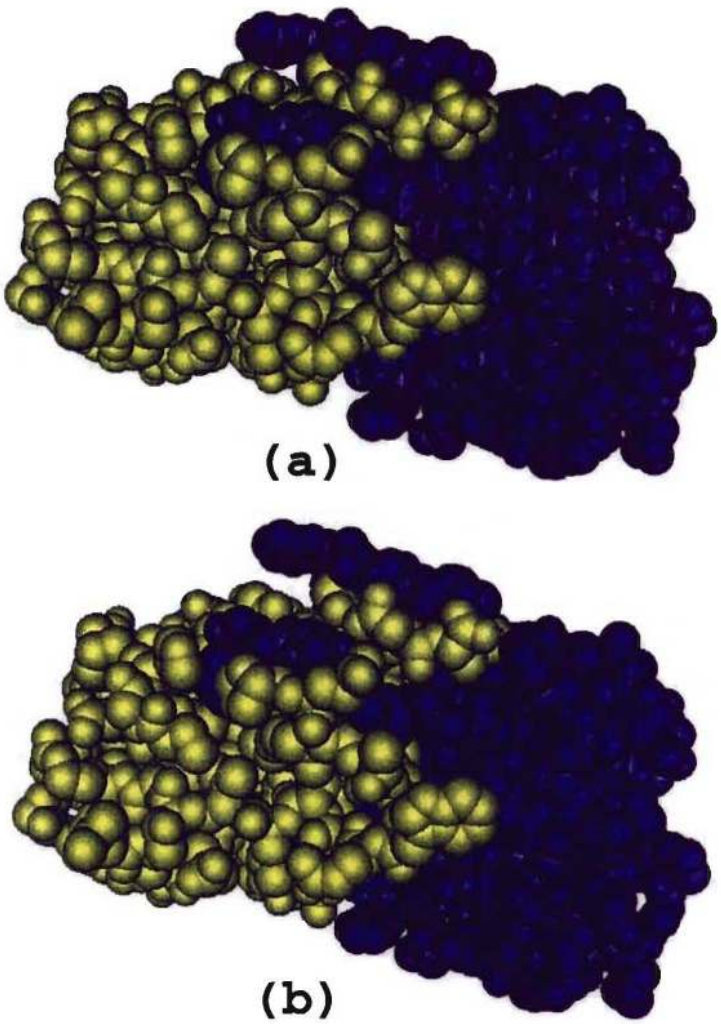

Figure 4: Example: (a) The natural docking conformation of the HIV-1 protease (dimer). (b) The best geometric fit (1.62 Å rms-deviation).

docking experiments were successful. Hence, there will be docking examples, where the docking sites cannot be identified by shape complementarity. We have to search for such examples, in order to learn more about the docking reactions. These examples will be the test set for our future research (refinement of the fitness function).

5 Future Research

First, we will check all docking examples in the Brookhaven Protein Data Bank and search for examples that cannot be docked by our algorithm. These examples will be our test set for future research.

In the above docking experiments the docking tests have been carried out with the docking conformations of the two molecules A and B, i.e, the conformations of A and B in the docking complex. We also did a few successful docking experiments with "native" conformations, for example with the HIV protease. But there is a lack of examples, where the "native" conformations of A and B and the conformation of the docking complex are known. We have to search for such examples, because the results of docking tests with the "native" conformations are the real quality measure for a docking program.

We will expand our model of conformation valuation. We will add a hydrogen bond component to our fitness functions. More precisely: Hydrogen bond building atom pairs (a, b), that are in contact, will get a larger constant weight. We hope that this refinement of the model will result in a better separation of the good docking sites from the docking sites that are randomly geometrically complementary. By refining the model in this way, we leave of course the pure geometric consideration of the docking problem. Note that we do still not compute energy values. Furthermore, we will try to reduce the number of fitness tests with the help of hydrogen bonds (i.e., by asking for at least one pair of atoms that can build a hydrogen bond in each matching that will be tested).

Besides we will try to test other methods to generate the points on the probe sphere surface and other methods to compute the orientations that have to be tested. We observed that the worst results with respect to RMS deviation are caused by "large" deviations of the rotation angles. We will try to decrease the RMS deviations by changing the point set of surface points or by modifying the way the triangles are matched. Our program has a simple local fitness optimizer that shakes molecule B locally. This optimizer has to be improved.

Since the sequential running time is in the order of the fastest known sequential algorithm [LNF+] and since the parallel algorithm shows a good speedup, we hope that we will be able to handle a list of docking candidates $\{B_1, B_2, \cdots, B_k\}$ for a molecule A with a high performance multi-processor system in reasonable time. The possibility to handle lists of docking candidates will be added to the system.

The preprocessing step will be parallelized. Our program is still lacking the ability to repair (substitute) problems that are caused by a break-down of a processor. We will implement the following straightforward repair method: The master checks if a processor has problems to carry out his work. If this is the case, the master redistributes this work. Furthermore, the speedup of the parallel version will be measured on a SGI POWER CHALLENGE multi-processor system.

Finally, we will implement a filter that removes transformations which are very similar (same docking site). We observed that our score list (1000 transformations) contains very often more than hundred almost identical transformations. We will implement a simple solution that

builds lists of similar transformations. Hence the score list will become a list of transformation lists.

Acknowledgements: I thank Prof. Dr. Kurt Mehlhorn and Priv. Doz. Dr. Michiel Smid for their comments on earlier versions of the paper. Their proposals significantly improved the readability of the paper. I am also grateful to Prof. Dr. Dietmar Schomburg and Dr. Michael Meyer for placing docking examples at our disposal.

References

[Con1] M.L. Connolly: "Analytical Molecular Surface Calculation.", J. Appl. Cryst., vol. 16, 1983, pp. 548–558.

[Con2] M.L. Connolly: "Solvent Accessible Surface of Proteins and Nucleic Acid.", Science, vol. 221, 1983, pp. 709–713.

[Con3] M.L. Connolly: "Shape Complementary at the Hemoglobin $\alpha_1\beta_1$ Subunit Interface.", Biopolymers, vol. 25, 1986, pp. 1229–1247.

[EKS+] M. Ester, H.P. Kriegel, T. Seidl and X. Xu: "Formbasierte Suche nach komplementären 3D-Oberflächen in einer Protein-Datenbank.", in Datenbanksysteme in Büro, Technik und Wissenschaft, GI-Fachtagung 1995, Georg Lausen (Hrsg.), Springer Verlag, pp. 373–382.

[FNN+] D. Fischer, R. Norel, R. Nussinov and H.J. Wolfson: "3-D Docking of Protein Molecules." Combinatorial Pattern Matching, 1993, LNCS 684, Springer Verlag, pp. 20–43.

[FVF+] J. Foley, A. van Dam, S. Feiner and J. Hughes: "Computer Graphics: Principles and Practice." Addison Wesley, 1990.

[HCT] M. Helmer-Citterich and A. Tramontano: "PUZZLE: A New Method for Automated Protein Docking Based on Surface Shape Complementarity.", J. Mol. Biol., vol. 235, 1994, pp. 1021–1031.

[KSE+] E. Katchalski-Katzir, I. Sharir, M. Eisenstein, A.A. Friesem, C. Aflalo and I.A. Vakser: "Molecular Surface Recognition: Determination of Geometric Fit between Protein and their Ligands by Correlation Techniques.", Proc. Natl. Acad. Sci. USA, 1992, vol. 89, pp. 2195–2199.

[KBO+] I.D. Kuntz, J.M. Blaney, S.J. Oatley, R. Langridge and T.E. Ferrin: "A Geometric Approach to Macromolecule-Ligand Interactions." J. Mol. Biol., vol. 161, 1982, pp. 269–288.

[KCF] F.S. Kuhl, G.M. Crippen and D.K. Friesen: "A Combinatorial Algorithm for Calculating Ligand Binding.", J. Comp. Chem., vol. 5 (1), 1984, pp. 24–34.

[Lew] R.A. Lewis: "Clefts and Binding Sites in Protein Receptors.", in Methods in Enzymology, Editor: J.J. Langone, vol. 202, 1991, pp. 126–156.

[LNF+] S.L. Lin, R. Nussinov, D. Fischer and H.J. Wolfson: "Molecular Surface Representations by Sparse Critical Points.", PROTEINS: Structure, Function, and Genetics, vol. 18, 1994, pp. 94–101.

[LW] Y. Lamdan and H.J. Wolfson: "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme." In Proceedings of the IEEE Int. Conf. on Computer Vision, 1988, pp. 238–249.

[NLW+] R. Norel, S.L. Lin, H.J. Wolfson and R. Nussinov: "Shape Compementarity at Protein-Protein Interfaces.", Biopolymers, vol. 34, 1994, pp. 933–940.

[Sun] V.S. Sunderam: "PVM: A Framework for Parallel Distributed Computing.", Concurrency: Practice & Experiment, vol. 2, 1990, pp. 315–339.