



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

Semester Thesis

---

# Protein Docking Using Local Shape Distributions

---

Micha Riser

riserm@student.ethz.ch

Department of Computer Science  
Swiss Federal Institute of Technology Zurich

September 2004

Supervisors:

Jan Remy

Prof. Dr. Angelika Steger

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The Docking Problem . . . . .	3
1.2	Related Work . . . . .	3
1.3	Motivation . . . . .	3
1.4	Overview . . . . .	4
<b>2</b>	<b>Shape Distributions of Proteins</b>	<b>5</b>
2.1	Global Shape Distribution . . . . .	5
2.2	Discretisation . . . . .	6
2.3	From Global to Local . . . . .	6
2.4	Surface Complementarity . . . . .	8
2.5	Histogram Distances . . . . .	9
<b>3</b>	<b>The Algorithm</b>	<b>11</b>
3.1	Outline . . . . .	11
3.2	Running Time . . . . .	12
3.3	Parameters . . . . .	12
<b>4</b>	<b>Experimental Results</b>	<b>14</b>
4.1	Test Data . . . . .	14
4.2	Results . . . . .	16
4.2.1	$R_{box}, rsize = 1$ . . . . .	16
4.2.2	$R_{box}, rsize = 1.5$ . . . . .	16
4.2.3	$R_{box}, rsize = 2$ . . . . .	17
4.2.4	$R_{geom}, rsize = 1$ . . . . .	17
4.2.5	$R_{geom}, rsize = 1.5$ . . . . .	17
4.2.6	$R_{geom}, rsize = 2$ . . . . .	17
4.2.7	$R_{gauss}, rsize = 1$ . . . . .	18
4.2.8	$R_{gauss}, rsize = 1.5$ . . . . .	18
4.2.9	$R_{gauss}, rsize = 2$ . . . . .	18
4.3	Discussion . . . . .	19

---

<b>5</b>	<b>Conclusions</b>	<b>21</b>
5.1	Summary . . . . .	21
5.2	Open Issues . . . . .	21
5.3	Comparison with Other Protein Docking Algorithms . . . . .	22
<b>A</b>	<b>Command Line Arguments</b>	<b>23</b>

# Chapter 1

## Introduction

### 1.1 The Docking Problem

When biochemical reactions take place it is a characteristic process that proteins interact with each other. The so-called receptor connects to the ligand. It is a challenge to predict whether and how a pair of proteins might interact. This is referred to as the *docking-problem* which has been fascinating biophysical chemists and computer scientists since the late 1970s [13]. Given the structure of two interacting proteins a docking algorithm aims to find the orientation of the docked complex. At the interface, two docking proteins show high degrees of steric and chemical complementarity. Besides the large number of atoms, a key difficulty of the docking problem is that the interfaces of both the receptor and the ligand may undergo conformational changes during the docking process.

### 1.2 Related Work

There have been various approaches to reduce the running time for the docking-problem. A common way is to perform initial geometric analysis of each surface and locate cavities, local knobs and holes [8, 3]. Grid-based Fourier correlation approaches [7] have been used to calculate the degree of overlap efficiently. However, high resolution correlations still take hours or days of computing time. And working with large grids which is required to get accurate results is a memory-intensive process. Ritchie and Kemp [11] approached these problems using a method based on spherical polar correlations of protein surface representations.

### 1.3 Motivation

We use *shape distributions* to describe the proteins. Shape distributions are not new and have already been used to classify 3D models [9] and to find

protein similarity [2]. A distribution can be seen as the fingerprint for a part of the surface.

Our approach to the docking-problem is to generate a collection of shape distributions to describe the possible docking positions of a protein. Having generated the fingerprints, i. e. the shape distributions, in the preprocessing, the docking problem is then solved by searching for two fingerprints that are very similar. The advantage of shape distributions is that they can be compared very fast. Furthermore, they are translation and rotation invariant and therefore they are independent of the protein's orientation in space. As shape distributions have successfully been used to find protein similarity, the hope is that they are also suitable to work despite the conformation changes the proteins undergo in the docking.

## 1.4 Overview

The next chapter introduces the definition of a shape distribution and shows how it has been adapted to suit our purposes. Then we present the algorithm in detail and do the running time analysis. Chapter 4 presents the results of the semester-thesis. Finally we compare our algorithm with others and reason about open problems.

## Chapter 2

# Shape Distributions of Proteins

### 2.1 Global Shape Distribution

Canzar and Remy [2] used the simple model for shape similarity, introduced by Osada, Funkhouser, Chazelle and Dobkin in [9] for general objects, to find proteins of similar shape. In this method a shape distribution characterizing the protein's geometry is calculated as follows:

Let  $S$  be the set of points on the surface of the protein,  $|S| = \text{area}(S) = \int_S dS$  its area. We look at the random experiment of choosing uniformly a point on the surface. Let  $E$  be the random variable which maps the outcome of the experiment to the surface point. Hence, the associated probability density function  $\phi$  is given by

$$\phi_E(p) = \begin{cases} 1/|S| & , p \in S \\ 0 & , p \notin S \end{cases} \quad (2.1)$$

Then, for the global shape distribution we perform this experiment twice independently from each other and calculate the random variable  $G$  as a function of the two independent random variables  $E_1, E_2$ :

$$\phi_G(d) = Pr( f(E_1, E_2) = d ) \quad (2.2)$$

where  $f$  is the so-called shape function which measures a geometric property that depends on  $S$ , e. g. the Euclidean distance between two points. Osada et al. claim that this resulting shape distribution is very characteristic for the shape of the object. So, the problem of matching geometric objects can be reduced to comparing probability distributions.

## 2.2 Discretisation

In order to be able to compute the distribution we approximate it by discretising the set of surface points we choose from. In [2] the middle points of all atoms located at the surface of the protein were considered as sampling points. Here, we first create a triangulation of the solvent excluded surface and then use the vertices of the triangulation as finite set of points  $S$ . We do so to have a finer resolution of the geometry and later on we use the graph of the triangulation to determine the connectivity between surface points.

Like that, we can evaluate the probabilities for the random variable  $G$  by summing over all pairs of points in our finite set  $S$  of  $N$  points. Let  $E_1, E_2$  be the random variables of the experiment defined as previously. Then,

$$\Pr(G = d) = \sum_{p_1 \in S} \sum_{p_2 \in S} \Pr( f(p_1, p_2) = d \mid E_1 = p_1 \wedge E_2 = p_2 ) \cdot \Pr(E_1 = p_1 \wedge E_2 = p_2) \quad (2.3)$$

$$= \frac{1}{N^2} \sum_{p_1 \in S} \sum_{p_2 \in S} \Pr( f(p_1, p_2) = d \mid E_1 = p_1 \wedge E_2 = p_2 ) \quad (2.4)$$

We divide the values of  $G$  into bins of interval length  $l$  to get a histogram of the shape distribution. So the value  $b_k$  of the  $k$ -th bin can be calculated as

$$b_k = \sum_{lk \leq x < (l+1)k} Pr(G = x) dx, \quad k = 0, \dots, M \quad (2.5)$$

Using the Euclidean distance  $f(p_1, p_2) = \|p_1 - p_2\|$  as shape function, the diameter  $d$  of the protein gives us an upper bound for the value of  $G$  and therefore  $M = \lfloor d/l \rfloor$ .

Like that, the protein's shape is characterized by a sequence  $(b_1, \dots, b_M)$  of values,  $\sum b_i = 1$ ,  $b_i$  describing the probability that the distance between two independently uniformly chosen vertices falls into the range of bin  $i$ .

## 2.3 From Global to Local

The protein docking can occur where two protein surfaces match locally. Therefore we try to adapt the idea of the global shape histogram to create histograms  $L_{c_0}, \dots, L_{c_n}$  of which each describes the shape in a local area around the point  $c_i$  only. So, if the docking center is at  $c_i$  only the surface around that point has to match. The docking is independent from the surface far away from  $c_i$  and therefore only surface points that are actually involved in the docking should contribute to the local histograms. This leads to choosing the points non-uniformly, i. e. preferring points near to

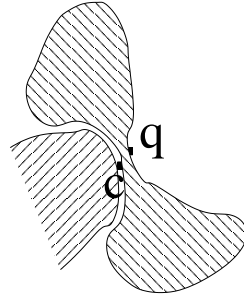


Figure 2.1: Nearness problem: Even though the point  $q$  is close to the docking center  $c$  in space, it is completely irrelevant for the docking.

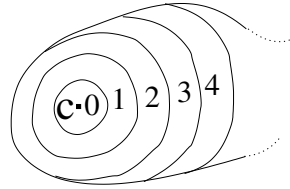


Figure 2.2: Division of the surface into rings with respect to the center point  $c$

the potential docking center and neglecting points on the other side of the protein.

However, nearness of surface points in space does not necessarily mean that the points are related in the docking situation, as depicted in figure 2.1. Therefore, we define a distance measure over the surface:

$$d_S(p, q) = \text{distance on the surface } S \text{ from } p \text{ to } q$$

As an approximation to this we look at the graph with the vertices as nodes and the triangle edges as edges and calculate the shortest path in this graph.

The random experiment  $E_{c_i}$  of choosing a point of the surface around the center  $c_i$  now works in the following way: First, we classify the vertices  $v_j \in S$  into rings according to their distance from the center point  $c_i$  of the local histogram  $L_{c_i}$  (fig. 2.2). That is, we put the vertex  $v_j$  into ring number  $r(v_j) = \lfloor d_S(v_j, c_i) / rsize \rfloor$  where  $rsize$  is a parameter that defines the width of the rings. Then, we pick the number of the ring with distribution  $R(i)$ . Among all vertices in the picked ring we take one vertex uniformly at random.

$$\Pr(E_{c_i} = v) = \frac{\Pr(R = r(v))}{|\{v' | r(v) = r(v')\}|} \quad (2.6)$$

For calculating the local shape histograms we proceed in the same way as for the global histograms, i. e. we take two independent instances of  $E_{c_i}$  and define:

$$\Pr(L_{c_i} = d) = \Pr( f(E_{c_i}, E'_{c_i}) = d ) \quad (2.7)$$

$$= \sum_v \sum_{v'} \Pr( f(v, v') = d | E_{c_i} = v \wedge E'_{c_i} = v' ) \\ \cdot \Pr(E_{c_i} = v) \Pr(E'_{c_i} = v') \quad (2.8)$$

Due to the locality of the histograms that we want to achieve, the distribution  $R(i)$  prefers smaller ring numbers, that is points which can be reached by shorter paths on the surface. One possibility is to choose uniformly among those vertexes which are not more than a certain distance away from the center. This can be modeled by a box-shaped  $R$ -function. For other possible  $R$ s see figure 3.1.

## 2.4 Surface Complementarity

To dock two proteins, it is not sufficient that we find pieces of the proteins' surfaces that are approximatively the same, but we have to find complementary surface patches. That is, a concave patch of the ligand has to match a convex patch of the receptor and so on. So far, our histograms do not respect this fact. They only depend on the surface geometry but do not distinguish between convexity and concavity, i. e. which side of the surface is inside the protein and which is outside.

We extended the histogram with a second dimension to classify the points, or more precisely the pairs of points, with respect to their location towards the interior of the protein. In more detail, this works as following: Given the two points  $v$  and  $v'$  we look at the line from  $v$  to  $v'$  and the surface normals (always pointing away from the protein) at the points. We shoot a ray from  $v$  into direction  $d = v' - v$  and test if the ray intersects the protein surface before arriving at  $v'$ . Depending on the angle between the normal vector and the direction and the result of the intersection test we classify the point pair into three possible categories (depicted in figure 2.3):

- 1 **"convex" situation:** The angle between the surface normal and the line is larger than  $\pi/2$  at both vertices and the ray does not intersect the surface.
- 0 **"intersect" situation:** The ray intersects the surface at least once. (If the angle between normal vector and line is at one vertex  $> \pi/2$  and at one  $< \pi/2$  then the ray must intersect the surface certainly due to its steadiness.)

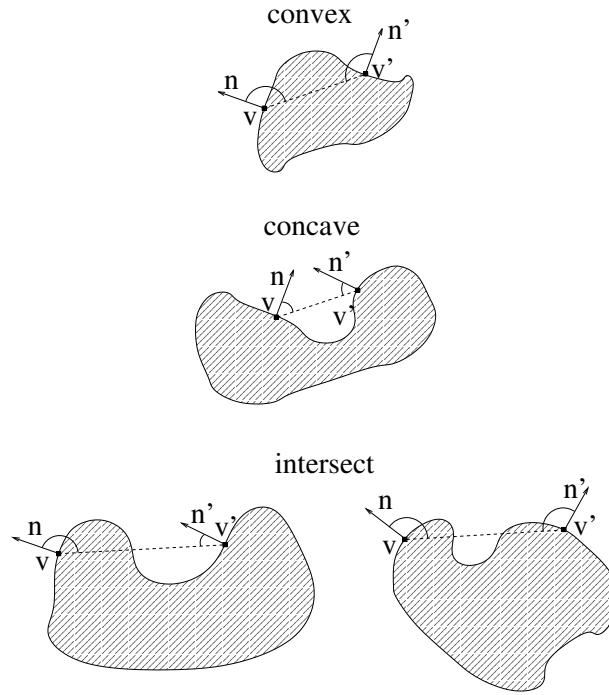


Figure 2.3: Classification of point pairs into "convex", "concave" and "intersect" (with one or two intersections).

- 1 **"concave" situation:** The angle between the surface normal and the line is smaller than  $\pi/2$  at both vertices and the ray does not intersect the surface.

## 2.5 Histogram Distances

The similarity of two surface patches can now be judged by comparing their shape distributions, i. e. the local histograms under an arbitrary metric. Let  $f_{i,j}$  be the  $i$ -th entry of the histogram in the classification  $j \in \{-1, 0, 1\}$ . We have chosen the Minkowski norm

$$D(f, \tilde{f}) = \left( \sum_{j \in \{-1, 0, 1\}} \sum_{i=1}^B |f_{i,j} - \tilde{f}_{i,-j}|^N \right)^{1/N} \quad (2.9)$$

with parameter  $N = 2$ .

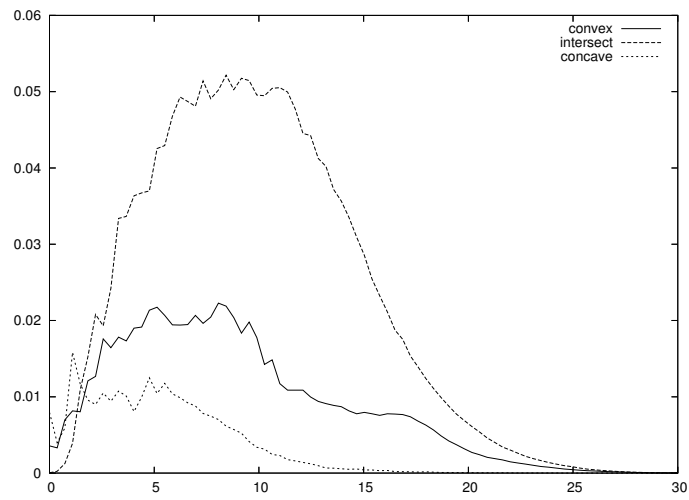


Figure 2.4: Example of a local histogram with the probability values for the distance between a point pair, classified to enable complimentary matchings.

## Chapter 3

# The Algorithm

### 3.1 Outline

The algorithm performs the following steps:

- **Preprocessing**

For each protein of which we want to find docking positions we do the following preprocessing steps:

1. Calculate the triangulation of the solvent surface.
2. Classify each vertex pair according to the method described in section 2.4 and store the result in a matrix.
3. For each vertex  $v_i$ : Consider  $v_i$  as the center of a possible docking area and calculate a histogram  $L_{v_i}$  characterizing the local surface around the center. To do this we first use Dijkstra's algorithm [5] to find the shortest path from  $v_i$  to all other vertices on the edge graph of the triangulation. Then, we divide the vertices into rings and calculate the histogram by summing over all pairs of vertices according to equation 2.8. Finally the histogram is written to a file.

- **Similarity Finding**

To find possible docking positions between a receptor and a ligand we first read all the local histograms  $L_1, \dots, L_n$  and  $\tilde{L}_1, \dots, \tilde{L}_{\tilde{n}}$  of both proteins from the file system. Then, for each pair of histograms  $(L_i, \tilde{L}_j)$  we calculate the distance between them and record the top  $K$  pairs with smallest distance.

Note that the preprocessing depends on the individual protein itself solely. Therefore it has to be performed only once for each protein even if the docking candidates change.

For the first step of the preprocessing we used Michel Sanner’s tool *msms* [12] which efficiently calculates the triangulation out of the information provided by the Protein Data Bank.

## 3.2 Running Time

**Preprocessing** According to Sanner et al. [12], the first step is done in  $O(m \log m)$  where  $m$  is the number of atoms in the protein. Let  $n$  be the number of vertices of the triangulation. The second step works in  $O(n^3)$  because there are  $O(n^2)$  pairs of vertices and for each pair we have to cast at most one ray. In the simplest form, the ray-casting works by testing the ray for intersection against all triangles of the surface, resulting in  $O(n)$  for one test. We can speed up the ray-casting by initially building up a bounding hierarchy over the triangles.

In the third step we have Dijkstra’s algorithm which runs in  $O(e \log n) = O(n \log n)$  as the number of edges  $e$  is a constant multiple of the number of vertices in our triangulation. The last step has running time  $O(n^3)$  because equation 2.8 loops over all pair of vertices and we do this for each vertex as possible docking center. Because we have previously calculated the classification of the point pair we do not have to cast any rays in this step and we can process each pair in constant time  $O(1)$ .

So, all in all, we arrive at the running time  $O(m \log m + n^3)$  for the preprocessing.

**Similarity Finding** The comparison of a single histogram pair is very fast. We have  $n$  histograms for the receptor and  $\tilde{n}$  for the ligand. Therefore the total running time is  $O(n + \tilde{n} + n \cdot \tilde{n}) = O(n \cdot \tilde{n})$  for reading the previously calculated histograms and finding the pairs with the lowest distance.

## 3.3 Parameters

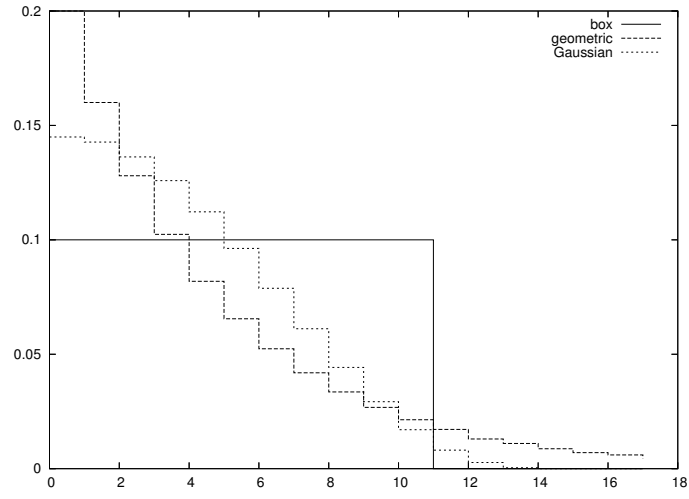
There are two parameters which influence the definition of the locality of the shape histogram: The distribution  $R$  for choosing the ring number and *rsiz*e which defines the width of the rings.

Possible distributions for  $R(i) := Pr(R = i)$ ,  $i \in \mathbb{N}$  are the box function, which weights rings 0 up to  $h - 1$  the same and then drops to zero at once:

$$R_{box}(i) = \begin{cases} 1/h, & i < h \\ 0, & \text{else} \end{cases}, \quad (3.1)$$

the geometric distribution, which drops the weight of the ring by a constant factor per ring as one gets farther away from the center:

$$R_{geom}(0) = 1 - h, \quad R_{geom}(i) = R_{geom}(0) \cdot h^i \quad (3.2)$$

Figure 3.1: Different distributions for  $R$ .

or an approximation to the Gaussian function which makes – in contrast to the geometric distribution – only small steps in the weights near the center.

$$R_{gauss}(i) = \begin{cases} c(h^2 - i^2)^3, & i < h \\ 0, & \text{else} \end{cases} \quad (3.3)$$

## Chapter 4

# Experimental Results

### 4.1 Test Data

For the test runs we used proteins that are listed in the RCSB Protein Data Bank (PDB). In particular we used the case DHB which has as receptor Hemoglobin  $\alpha_1$  and as ligand Hemoglobin  $\beta_1$ . The corresponding PDB code is 2DHB.

We chose this particular protein pair because it is already known how the interaction takes place [1]. That is, the file from the PDB already gives the atom locations in docked position. We extracted the two strings of amino acids as the input of the docking test. From that, the triangulation of the solvent surface was created. Note that there was no need to randomly displace the proteins away from their docking position before using them as input as it is necessary with other methods to avoid that the algorithm has an artificially good starting point. In our case the generation of the fingerprints is completely translation and rotation invariant so it would not have made any difference.

Our program supports two methods of docking search:

1. One full protein and a patch of surface with center point as input. The program finds the place on the protein where the given patch matches best.
2. Two full proteins as input. The program tries to find the best docking positions of the given proteins.

The test results shown here were all made using the second variant. That is, the input to the program was the full triangulation of the two proteins of the DHB case. We systematically varied the parameters *rsize* and the distribution for *R* which both together define the locality of the fingerprints.

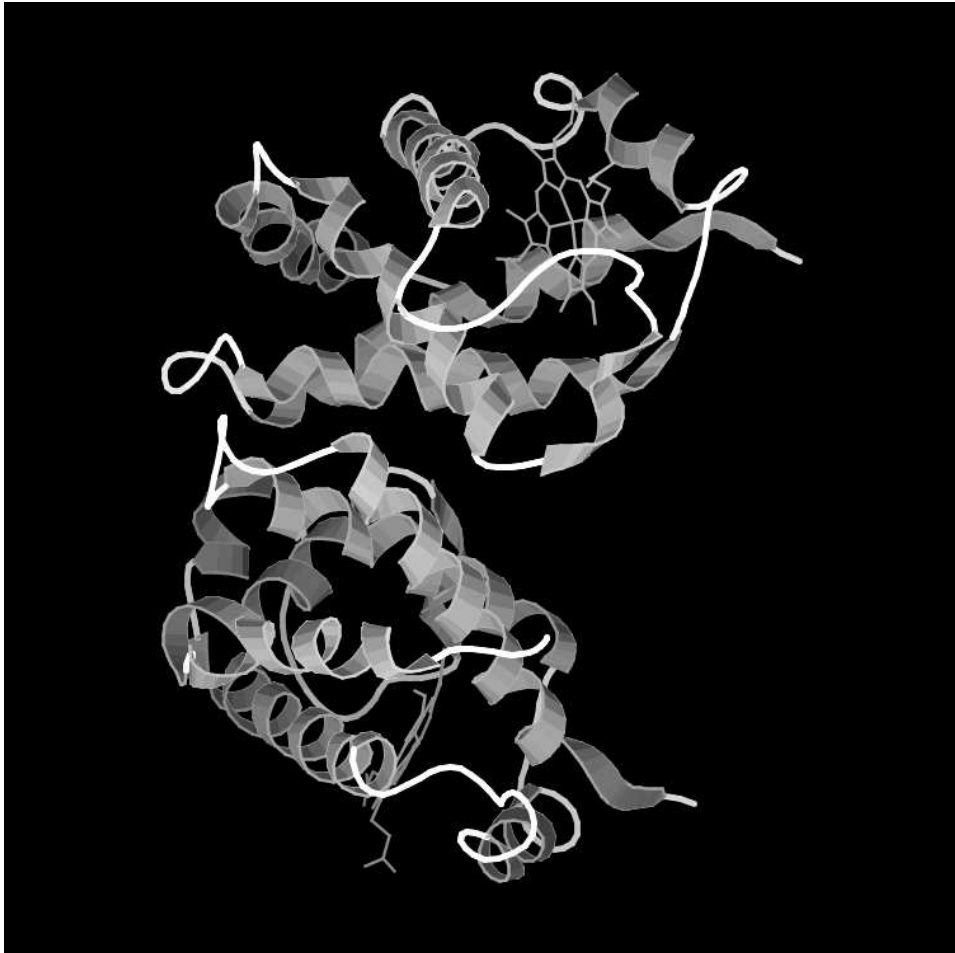


Figure 4.1: Molecular structure of 2DHB with coils and helices.

## 4.2 Results

The following tables present the test results for the different parameters.  $h$  is the parameter for the random variables  $R_{box}$  (eq. 3.1),  $R_{geom}$  (eq. 3.2) and  $R_{gauss}$  (eq. 3.3) resp. The program returns a list of the 5000 best matching fingerprints. This list is ordered by distance of the shape distributions. We considered a pair of vertices as correct docking location if the vertices were less than 2 Å away from each other in the original docking position. Top match records the rank of the first fingerprint in the output that was such a docking match. # matches counts the number of docking matches among the 5000 best matching fingerprints.

The tests were run on a Pentium4/2.5GHz with 1.5GB RAM.  $t_{pre}$  is the time that the preprocessing step needed.  $t_{match}$  states how many seconds that the actual search for the docking positions lasted.

### 4.2.1 $R_{box}$ , $rsize = 1$

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
4	N/A	0	2408s	972s
5	N/A	0	2428s	975s
6	1281	1	2440s	969s
7	329	4	2462s	968s
8	24	9	2479s	970s
9	87	16	2518s	961s
10	105	21	2570s	961s
11	1219	17	2618s	960s
12	1937	5	2703s	961s

### 4.2.2 $R_{box}$ , $rsize = 1.5$

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
3	N/A	0	2420s	962s
4	4703	1	2435s	971s
5	772	11	2452s	963s
6	50	25	2511s	969s
7	94	22	2582s	962s
8	N/A	0	2528s	962s
9	N/A	0	2698s	961s
10	N/A	0	2833s	963s

**4.2.3**  $R_{box}$ ,  $rsize = 2$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
3	401	3	2424s	962s
4	26	12	2467s	961s
5	13	14	2559s	962s
6	N/A	0	2714s	967s
7	N/A	0	2881s	963s
8	N/A	0	3223s	961s
9	N/A	0	3571s	966s
10	N/A	0	4102s	961s

**4.2.4**  $R_{geom}$ ,  $rsize = 1$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
0.50	980	3	2483s	962s
0.60	817	3	2475s	970s
0.70	1599	4	2556s	960s
0.75	722	4	2694s	961s
0.80	84	5	2928s	960s
0.85	262	7	3365s	961s
0.90	N/A	0	4775s	961s

**4.2.5**  $R_{geom}$ ,  $rsize = 1.5$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
0.50	N/A	0	2506s	963s
0.60	N/A	0	2695s	960s
0.70	2325	4	3024s	963s
0.75	1637	9	3568s	962s
0.80	N/A	0	4393s	969s
0.85	N/A	0	5253s	960s
0.90	N/A	0	11743s	962s

**4.2.6**  $R_{geom}$ ,  $rsize = 2$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
0.50	698	3	2683s	960s
0.60	357	11	3167s	970s
0.70	1252	9	4091s	960s
0.75	N/A	0	5589s	961s
0.80	N/A	0	7805s	963s
0.85	N/A	0	12646s	960s
0.90	N/A	0	23750s	968s

4.2.7  $R_{gauss}, rsize = 1$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
9	2124	1	2505s	963s
10	603	1	2562s	963s
11	177	3	2610s	962s
12	62	7	2697s	961s
13	33	8	2786s	961s
14	28	10	2892s	963s
15	28	12	3026s	961s
16	44	14	3175s	962s
17	76	14	3354s	961s
18	234	15	3639s	962s

4.2.8  $R_{gauss}, rsize = 1.5$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
6	N/A	0	2505s	962s
7	1434	4	2587s	964s
8	355	12	2683s	963s
9	207	18	2836s	962s
10	187	22	3033s	963s
11	254	21	3254s	960s
12	402	16	3559s	961s
13	1682	5	3947s	961s

4.2.9  $R_{gauss}, rsize = 2$ 

width $h$	top match	# matches	$t_{pre}$	$t_{match}$
4	2035	6	2466s	961s
5	312	9	2564s	961s
6	165	12	2687s	965s
7	26	25	2888s	961s
8	34	15	3171s	964s
9	435	9	3569s	965s
10	N/A	0	4075s	963s

### 4.3 Discussion

The tests have shown that with the right parameters for the ring weighting distribution the algorithm produces usable results. I. e. the fingerprints of the correct docking positions match to a large extent. However there are also fingerprints that match in a same amount but the associated centers are rather far away from the correct docking position.

For all distributions, box, geometric and approximated gauss, we see that the ring size is not the key parameter: for all *rsiz*e we find values *h* for which the algorithm performs well. However, setting the parameter *h* carefully is important. If we look at the results from the box distribution we can estimate how large the surface patch is where the proteins touch each other: The performance of the algorithm suddenly drops when the maximal distance from the center for a point to contribute to the histogram *rsiz*e·*h* is larger than about 11Å. This suggests that the contacting patch has a similar radius.

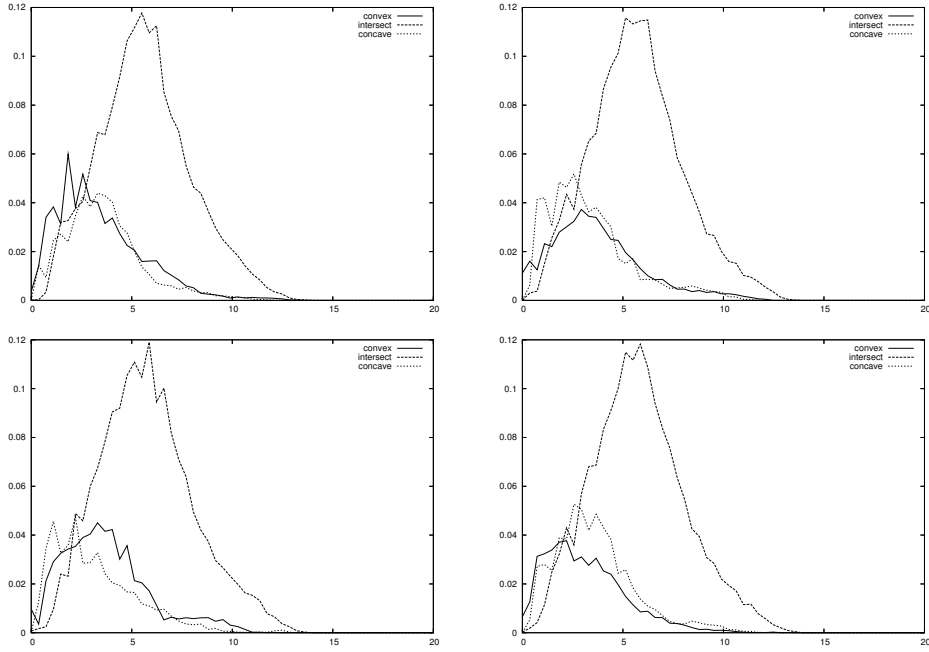
The results from the geometric weighting are rather ambiguous. We find some correct docking positions among the top 5000 for different parameters but for none we get as good results as with the box function. A problem with the geometric weighting is that near the center there are a few vertices that have rather large weights. This leads to the situation that already a small deviation between the two surfaces near the center can cause a big difference in the histograms. This difference could also been caused by a different triangulation of otherwise identical surfaces.

The approximation to the Gaussian function as distribution diminishes that problem as it is flatter for small values. The tests using  $R_{gauss}$  seem to confirm this and show good results for a large range of *h*, especially with a ring size of 1Å: The correct docking position is among the top 50 of the best matching fingerprints.

The results show that the running time of the actual matching search is independent of the ring weighting. This is easy to explain as the number of histograms which are generated and which need to be compared with each other is always the same. The time for the preprocessing depends mainly on the "width" of the weighting function, i. e. how many vertices contribute to each histogram. Due to their nature, the geometric and gauss functions need a larger width to perform well and therefore, they need more computing time than the box function in the preprocessing.

We have also tried to match triples of histograms that result from looking at the histograms associated to the three edges of each triangle. However, this resulted in a significant increase of the computation time in the docking search ( $t_{match}$ ), about a factor 5 in our implementation. On the other hand, the results with this enhancement were not much better than the plain comparison of single histograms.

Figure 4.2: Matching fingerprints. Top: The two histograms reported as best match in the test case  $R_{box}$ ,  $h = 8$ ,  $rsize = 1$ . Bottom: The two histograms of the correct docking position reported at place 24 in the same test.



# Chapter 5

## Conclusions

### 5.1 Summary

We have adapted the idea of global shape distributions for our purpose and presented a method to describe proteins with a set of fingerprints where each fingerprint catches the geometric properties of a part of the protein surface around a center point. We have shown a way to distinguish between convex and concave surface patches and integrated this into the fingerprints. This set of fingerprints then allowed a fast search for possible docking positions between a pair of proteins.

### 5.2 Open Issues

There are some ways how the results of the algorithm might be improved: One problem of the current histograms are that they are rather unsteady. This is due to the fact that the number of sampling points is not so large and the samples have different weights. One could extend the algorithm so that it does not assign weights to the vertices but to the faces instead. Then, Monte Carlo sampling could be used to create the histograms by randomly picking points on the faces.

Currently, we have many "false matches" among the best matching fingerprints, i. e. fingerprints that match good but are not valid docking positions. This can happen because the shape distributions reduce the 3D structure of the surface to a one-dimensional curve. One might try to include different properties of the protein into the histogram like normal vectors (which we already partly have) or electrostatic potentials.

To actually find possible matching positions we do a "brute force" search on the fingerprints, i. e. each fingerprint of one protein is compared with each fingerprint of the other protein. This search might be speed up by creating a data structure with the set of histograms which allows a fast lookup of similar histograms.

### 5.3 Comparison with Other Protein Docking Algorithms

When we compare the results of our algorithm with the docking algorithm using spherical polar Fourier correlations (Ritchie and Kemp [10], who report an improvement of their method in comparison to the geometric hashing method [4] and the combined steric/electrostatic FFT correlation method [6]) we see that our algorithm performs worse, it does not find the correct docking in the top 10. However, with calculation times around 20 minutes for the docking search is our algorithm noticeably faster than [10] where calculation times of about 2 hours are reported for the global search. However, they used a different CPU which is certainly slower than what we have used for our tests. So, the calculations times are hardly comparable.

## Appendix A

# Command Line Arguments

**Global Compare** This mode creates the global histograms of the given two protein files and compares them to each other.

```
proteins --global prefix1 prefix2
```

Here, `prefix $n$`  denotes the prefix of the vertices (`prefix $n$ .vert`) and the faces (`prefix $n$ .face`) files corresponding to the two proteins.

**Protein Docking Search** This mode tries to find the docking positions of the given proteins:

```
proteins --wf box|geom|gauss <width> --rw <width>
        [--read] prefix1 prefix2
```

Here, `--wf box|geom|gauss <width>` determines the type and width of the ring weighting function, `--rw` sets the ring width in Å. If the `--read` flag is set the program reads the histograms from the filesystem (saved from a previous run).

If you do not want to do the docking search on two full proteins but find the best matching segment on a protein then use the following parameters:

```
proteins --wf box|geom|gauss <width> --rw <width>
        [--read] prefix1 -p <num> prefix2
```

Where `prefix1` points to the protein, `prefix2` to the segment files and the number after `-p` gives the vertex index of the segment center.

# Bibliography

- [1] W. Bolton and M. F. Perutz. Three dimensional fourier synthesis of horse deoxyhaemoglobin at 2.8 Å resolution. *J. Mol. Biol.*, 228(271):551–552, Nov 1970.
- [2] S. Canzar and J. Remy. Shape distributions and protein similarity. 2003.
- [3] M. L. Connolly. Shape complementary at the hemoglobin  $\alpha_1\beta_1$  subunit interface. *Biopolymers*, 25:1229–1247, 1986.
- [4] t. l. W. D. Fischer, S. Liang Lin and R. Nussinov. A geometry-based suite of molecular docking processes. *J. Mol. Biol.*, 248:159–177, 1995.
- [5] E. Dijkstra. *Numerische Mathematik*, volume 1, chapter A note on two problems in connection with graphs, pages 269–271. 1959.
- [6] H. A. Gabb, R. M. Jackson, and M. J. Sternberg. Modelling protein docking using shape complementary, electrostatics and biochemical information. *Journal of Molecular Biology*, 272(1):106–120, September 1997.
- [7] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, and A. A. Friesem. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. In *Proceedings of the National Academy of Sciences*, volume 89, pages 2195–2199, March 1992.
- [8] R. Norel, S. L. Lin, H. J. Wolfson, and R. Nussinov. Molecular surface complementary at protein-protein interfaces: The critical role played by surface normals at well placed, sparse, points in docking. *Journal of Molecular Biology*, 252(2):263–273, 1995.
- [9] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.
- [10] D. W. Ritchie and G. J. Kemp. Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces. *Journal of Computational Chemistry*, 20(4):383–395, March 1999.

- 
- [11] D. W. Ritchie and G. J. Kemp. Protein docking using spherical polar fourier correlations. *Proteins: Structure, Function, and Genetics*, 39(2):178–194, May 2000.
- [12] M. F. Sanner, A. J. Olson, and J.-C. Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, March 1996.
- [13] S. Wodak and J. Janin. Computer analysis of protein-protein interaction. *J. Mol. Biol.*, 124(2):323–342, Sep 1978.