

MASARYK UNIVERSITY  
FACULTY OF INFORMATICS



# Algorithms searching for discontinuous motifs on the surface of protein molecules

MASTER THESIS

Bc. Zuzana Jiroušková

Brno 2006

## Declaration

I declare that this thesis was composed by myself and all presented results are my own unless stated otherwise. All sources and literature that I have used are cited with reference to the corresponding source quoted in the text.

.....

## Acknowledgements

I would like to express my sincere thanks to my supervisor Mgr. Radka Svobodová Vařeková who introduced the world of computational chemistry to me. I am grateful for her enthusiasm, valuable guidance and help with every detailed question.

My gratitude also belongs to my family members for their love and constant encouragement. Special thanks to Bc. Luděk Novotný for his patience and support in hard moments.

## Abstract

This thesis is focused on the algorithms searching for discontinuous motifs on the surface of protein molecules. A surface is the most important part of a molecule, because all reactions between molecules take place on the surface. Therefore, it is favourable to know which atoms or motifs (continuous or discontinuous groups of amino acids) occur on the surface and which lie buried in the internal part of the molecule.

From the algorithms for identification of the surface atoms three suitable ones were chosen and studied in detail: NIN approach (a classical simple approach for identifying surface atoms), UCSF approach (which uses heuristics) and a sophisticated SAS approach. The algorithms are described within this thesis and analysed from many points of view. Using the basis of these algorithms, an algorithm searching for discontinuous motifs on surface of protein molecules was developed.

Described algorithms were implemented and successfully tested for searching surface atoms and discontinuous motifs (specifically biologically important motifs from the ELM database). These atoms and motifs were searched on the the surfaces of real protein molecules from the PDB database. The programs `samie` (for surface motifs) and `sad` (for surface atoms) were created by the implementation of the above mentioned algorithms. For the calculation of the solvent-accessible surface was designed a script `sasa`.

The developed programs will be used in a larger bioinformatics project realised in co-operation between University College Dublin and ANF Data company. This project is focused on determining geometrically similar motifs in molecules of drugs and proteins. Specifically, the identification of the surface protein motifs will help to reduce the amount of studied data.

## **Keywords**

surface atoms, van der Waals surface, solvent-accessible surface SAS, contact/re-entrant surface, Number of intersecting neighbours NIN, University of California at San Francisco UCSF, proteins, protein motifs

# Contents

<b>I</b>	<b>Introduction</b>	<b>9</b>
<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Goals of the thesis . . . . .	11
1.2	Structure of the thesis . . . . .	12
<b>II</b>	<b>Theoretical part</b>	<b>13</b>
<b>2</b>	<b>Molecular surfaces</b>	<b>14</b>
2.1	Definitions of molecular surfaces . . . . .	14
2.1.1	Van der Waals surface . . . . .	14
2.1.2	Solvent accessible surface . . . . .	15
2.1.3	Contact/re-entrant surface . . . . .	16
2.2	Definition of a surface atom . . . . .	16
<b>3</b>	<b>Computational methods for identifying surface atoms</b>	<b>18</b>
3.1	NIN approach . . . . .	18
3.2	UCSF approach . . . . .	20
3.3	SAS approach . . . . .	23
<b>4</b>	<b>Proteins</b>	<b>25</b>
4.1	Amino acids . . . . .	25
4.1.1	Peptide bond . . . . .	26
4.2	Structure of proteins . . . . .	26
4.3	Protein motifs . . . . .	28
<b>III</b>	<b>Methods</b>	<b>30</b>
<b>5</b>	<b>Methods</b>	<b>31</b>
5.1	Hardware and software . . . . .	31
5.2	PBD format . . . . .	31
5.3	ELM . . . . .	33

5.4	VMD	33
5.5	RasMol	34
<b>IV</b>	<b>Implementation</b>	<b>35</b>
<b>6</b>	<b>Implementation</b>	<b>36</b>
6.1	Program sad	36
6.1.1	Functionality	36
6.1.2	Programming language	36
6.1.3	Interface	36
6.1.4	Output file	37
6.2	Program samie	38
6.2.1	Functionality	38
6.2.2	Programming language	38
6.2.3	Interface	39
6.2.4	Input motif	39
6.2.5	Output file	40
6.3	Script sasa	40
6.3.1	Functionality	40
6.3.2	Programming language	40
6.3.3	Interface	40
<b>V</b>	<b>Results and discussion</b>	<b>42</b>
<b>7</b>	<b>Testing</b>	<b>43</b>
7.1	Testing molecules	43
7.2	Surface atoms	44
7.2.1	NIN approach	44
7.2.2	UCSF approach	45
7.2.3	SAS approach	46
7.2.4	Comparison of approaches	47
7.3	Surface motifs	49
<b>VI</b>	<b>Conclusion</b>	<b>54</b>
<b>8</b>	<b>Conclusion</b>	<b>55</b>

## VII Appendices

57

### A List of amino acids in proteins

58

# Part I

## Introduction

# Chapter 1

## Introduction

Computational chemistry [1, 2] is considered to be a branch of classical chemistry which started its development in the early sixties of the last century. Both disciplines are derived from the same chemical theory, but they are also very different in many ways. Computational chemistry is so much influenced by informatics that it can be regarded as a discipline on the boundary between chemistry and informatics.

One of the things in which both disciplines differ is the way of applying chemical theories. Classical chemistry looks for the theoretical models and natural relations which are verified by experimentally received data. Computational chemistry also uses theoretical models, but in a different way: for implementation and molecular modeling on real systems of molecules.

Co-operation between classical chemistry and computational chemistry is very profitable for both sides. Computational chemistry can be a powerful help, especially in such courses of study, where experimental solving of given problem is almost impossible or very difficult, for example working with unstable molecules and so on.

Among the most actively-studied groups of molecules belong the proteins – large biopolymers, which consist of amino acids. Proteins have an important role in biochemistry and perform a wide variety of biological functions [3]. Some of them are building blocks of tissues, organs, hair, nails and so on, other catalyze biochemical processes (enzymes), coordinate chemical processes in the organisms (hormones) or play structural or mechanical roles. Because of their big importance, proteins are also studied by a variety of computational methods.

From the biochemical point of view, the most important part of a molecule is its surface. Its importance comes from the fact that all reactions between molecules take place on the surface. Therefore, it is necessary to know which atoms or motifs (continuous or discontinuos groups of amino acids) occur on the surface and which lie buried in the internal part of the molecule. This is also one of the reasons why the algorithms searching for discontinuos motifs on the surface of protein molecules are the subject of this thesis. The objectives of this elaboration are to study existing algorithms for identifying surface atoms, choose the most appropriate of them and adapt them to the proteins. The obtained results are subsequently used for searching motifs. The developed software will be tested for searching biologically important motifs from the ELM database [4] on the surfaces of real protein molecules from the PDB database [5].

The programs developed within this thesis will be used in a larger bioinformatics project realised in co-operation between University College Dublin and ANF Data Company. Specifically, this project is focused on determining geometrically similar motifs in molecules of drugs and proteins, because the active sites in proteins can be used as templates for the drug design. While searching for the geometrical similarities it is reasonable to work only with the surface protein motifs, which can radically reduce the amount of studied data. Determining the surface motifs is allowed by the programs created within this thesis.

A note: The thesis is realised in co-operation with ANF Data Company, a Siemens group member.

## 1.1 Goals of the thesis

- Learn the basic conceptions of computational chemistry
- Focus on the problematics of computer representation of molecules
  - learn the PDB format for a data storage of molecules
  - develop a suitable data structure for storing information received from the PDB file
  - implement a program for reading and storing necessary information from the PDB file into a designed data structure
- Focus on the problematics of the molecular surface and surface atoms
  - study the best-known algorithms for identifying surface atoms
  - describe three suitable algorithms
    - \* NIN approach
    - \* UCSF approach
    - \* SAS approach
- Focus on implementation
  - implement algorithms NIN, UCSF, SAS
  - implement algorithm for searching discontinuous motifs
- Focus on testing and analyses
  - test algorithms of NIN, UCSF and SAS approaches
  - compare and discuss results of these algorithms
  - test algorithm for searching discontinuous motifs on the surface of protein molecules
  - compare and discuss how the results of this algorithm depend on the input information about surface atoms

## 1.2 Structure of the thesis

The thesis is structured into seven parts. The entire thesis begins with the introductory part, but the centre of the work are the following Chapters 2, 3 and 4 which contain the theoretical part of the thesis. This part summarizes knowledge about the molecular surfaces and surface atoms, includes the best-known algorithms and discusses their time complexity. It also provides a brief description of proteins and their motifs and allows an easy insight into the problematics of searching discontinuous motifs on the surface of protein molecules.

The next part of the thesis is devoted to the methods and software tools used during the elaboration of this work and to the description of the used file format for molecule storage. This is developed in Chapter 5.

The subsequent part deals with the implementation of the mentioned algorithms. Chapter 6 includes the descriptions of the functionality of the implemented algorithms, gives reasons for the choice of programming language and introduces the user interface of all programs.

The following part of my thesis is devoted to the results and discussion, where the received pieces of knowledge are summarised, the weaknesses of algorithms are discussed and some particular examples are shown.

Finally, part Conclusion closes this thesis, summarizes the results and points out some directions for further research.

This work also contains appendices, which make the seventh part: these chapters contain some additional information, which the thesis refers to and which is above the scope of this thesis to deal with.

The text of this thesis and all files of source codes can be found on the enclosed CD.

Part II

Theoretical part

# Chapter 2

## Molecular surfaces

### 2.1 Definitions of molecular surfaces

Classical chemistry works directly with molecules whereas computational chemistry substitutes real molecules for their appropriate representations.

In the real world a molecule is a system of moving particles (nuclei and electrons). Although the electron density in atoms has no strictly defined boundaries, a molecule is usually represented as a solid figure. This conception has been proved to be a good representation by many empirical and semiempirical applications.

We are interested in atoms which lie on the surface of a molecule, so first of all we must define a suitable representation of a molecular surface. Many different definitions of the molecular surface are presented in literature, but following models are usually considered to be the most important and have found wide use not only in the computational chemistry [6, 7, 8]:

- van der Waals surface
- solvent accessible surface
- contact/re-entrant surface

All the above mentioned models of molecular surface are described in detail in the following sections.

#### 2.1.1 Van der Waals surface

The conception of the **van der Waals surface** [6] assumes the following model of a molecule: each atom in the molecule is represented by a sphere. The center of such a sphere is the same as three-dimensional coordinates of the represented atom. The radius of the sphere is the van der Waals radius<sup>1</sup> of the relevant atom. Although the van der Waals radius is unequivocally defined,

---

<sup>1</sup>The van der Waals radius is a distance at which there is a balance between the van der Waals attractive and repulsive forces in the formation of chemical bonds.

its particular values for the same atom can differ. This is caused by the fact that the values of van der Waals surface are determined experimentally. The first values were given by A. Bondi in [9], but later many other values appeared. Specifically, in this thesis are used the van der Waals radii mentioned in [10].

The van der Waals surface of the molecule is defined as the surface of a molecule, which is represented by using the model described above. The van der Waals surface of a hypothetical molecule is illustrated in Figure 2.1.

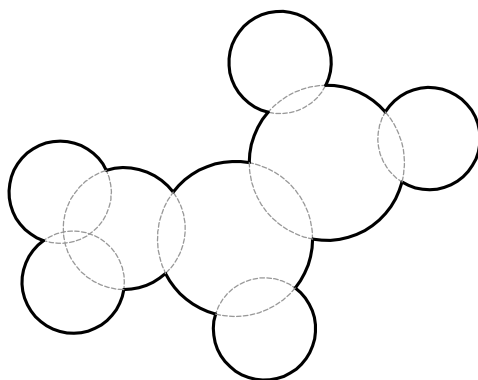


Figure 2.1: Cross-sectional view of the van der Waals surface

### 2.1.2 Solvent accessible surface

The biggest disadvantage of the van der Waals surface is, that it is not able to provide information about accessibility of the specific part of the molecular surface to the solvent. Specifically, the problem appears when the studied atom has its molecular surface hidden in some cavities. In spite of the fact that a solvent molecule will obviously never be able to touch it, according to the van der Waals surface conception this parts of the atom are considered to be parts of a molecular surface.

The first scientist who tried to solve this problem and improved the van der Waals surface was Hermann, who defined the **solvent accessible surface (SAS)** [7, 11]. As shown in Figure 2.2, SAS is a molecular surface which is defined as a set of all the centers of the solvent (probe) sphere, when the solvent is rolled over the entire van der Waals surface of a molecule. This is equivalent to a van der Waals surface in which the atomic radii are extended by the solvent radius.

The choice of the solvent radius is not strictly defined and it may differ depending on the concrete software implementation. Typically, the solvent is modeled as a sphere with the radius of  $1.40 \text{ \AA}$ <sup>2</sup>.

---

<sup>2</sup> $\text{\AA}$  (angström) is a traditional unit used for a description of a molecular structure. The relation between  $\text{\AA}$  and SI units is following:  $1 \text{ \AA} = 10^{-10} \text{ m}$ .

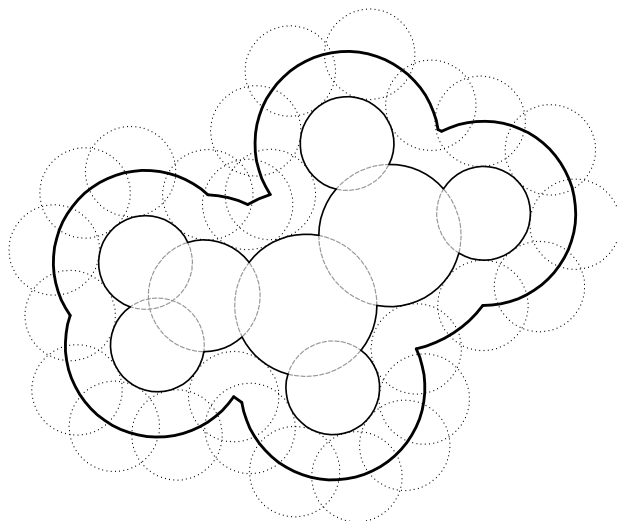


Figure 2.2: Cross-sectional view of the solvent accessible surface

### 2.1.3 Contact/re-entrant surface

According to Richards [8], the molecular surface is a combination of two parts. The first part is called **contact surface** and is defined as the van der Waals surface which is able to be in contact with the solvent surface. The second part is called **re-entrant surface** and it is a composition of interior-facing pieces of the solvent sphere touching simultaneously two or more atoms.

This model of surface is similar to the van der Waals surface, but excludes regions of the van der Waals surface which are unreachable to the solvent (see Figure 2.3).

**Contact/re-entrant surface** is regarded to be the most exact surface from all of the above mentioned descriptions. But it should be pointed out that calculation of this surface is CPU-consuming, and that is why it is not so widely used as SAS.

## 2.2 Definition of a surface atom

Using different models of the surface of the molecule may cause that the studied atom will be considered as a surface atom in one model and the other model will classify it as an internal atom. So it is important to define what we consider to be a surface atom.

Specifically, all three algorithms described in the following text are interested in the atoms which are accessible to the solvent, so the solvent-accessible surface model will be the most suitable choice. Due to the further use of solvent-accessible surface model, it will be useful to describe it in detail.

In case of the solvent accessible surface, we distinguish two types of surface [11]: true and effective. The **true surface atom** is an atom whose solvent accessible area  $SAS$  is greater than

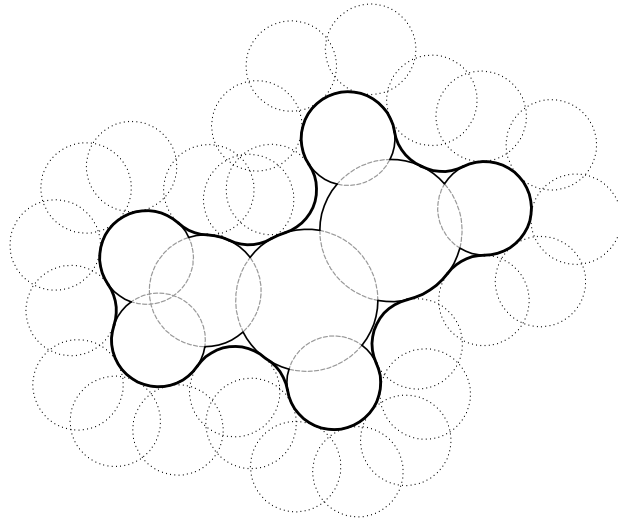


Figure 2.3: Cross-sectional view of the contact/re-entrant surface

zero. This concept of surface atom has one disadvantage: If  $SAS$  is an extremely small value, it could be desirable for us to mark this atom also as an internal atom. That is the reason why another type of surface atom was defined: The atom is considered to be an **effective surface atom** if its  $SAS$  is greater than a user-specified minimum threshold value.

The effective surface atoms are for the purpose of this thesis more favourable than the true surface atoms, and for that reason I am using the effective surface atoms.

# Chapter 3

## Computational methods for identifying surface atoms

There are many algorithms for identifying surface atoms in macromolecular structures. According to the complexity of approach we can divide them into two main groups:

First group contains **simple algorithms**, which are based on rather intuitive notions. These algorithms consider all atoms in the molecule and apply simple observations to them. A typical representative of this group of algorithms is the **Number of intersecting neighbours** (NIN) approach.

The second main group of algorithms are **sophisticated algorithms**, which are based strictly on mathematical foundations and have lower time-complexity than the previously mentioned algorithms. Typical representatives of these algorithms are the **Solvent-accessible surface** (SAS) approach or the **UCSF** approach.

In my thesis I used algorithms from both groups in order to better distinguish different approaches, which each particular algorithm uses. Each of them will be described in detail in the following text.

### 3.1 NIN approach

Number of intersecting neighbours (NIN) approach [11] is a characteristic representative of the first group of algorithms – the simple ones. For these algorithms it is typical that they are based on elementary observations. The observation of the NIN approach claims that atoms lying in the internal part of the molecule are covered by other atoms from all sides.

In this approach an atom is represented as a sphere with its radius counted as a sum of two radii: the van der Waals radius and the probe radius<sup>1</sup>.

The model of the molecule is the same as presented in section 2.1, the only difference is that

---

<sup>1</sup>the radius of the probe sphere, where the probe is a molecule of solvent, which is most often a water molecule

the atomic radius does not equal only the van der Waals radius, but the van der Waals radius plus the probe radius.

As the name indicates, this algorithm is based on a number of intersecting neighbours  $N^{int}$  for each atom in a molecule. Let's define  $r_{ij}$  as the distance between the centers of atomic spheres  $i$  and  $j$ ,  $R_i$  and  $R_j$  the atomic radii (it means the van der Waals radius plus the probe radius).

The definition of intersecting neighbours is following: two atoms  $i$  and  $j$  are intersecting neighbours if and only if

$$r_{ij} < R_i + R_j$$

This condition is illustrated in Figure 3.1.

In other words, if the investigated atom lies in the internal part of the molecule, the number of adjacent atoms which intersect it, is greater than in case of a surface atom. It means that its  $N^{int}$  will be quite high for the internal atoms. On the contrary it is obvious that an atom, which lies on the surface of the molecule (surface atom) has lower  $N^{int}$ .

Hence  $N^{int}$  is used as a basis for sorting atoms into two categories: the surface and the internal atoms.

The NIN approach algorithm is described by the following pseudocode:

*Pseudocode of algorithm*

FOR EACH atom  $i$

1.  $N_i^{int} = 0$
2. calculate its atomic radius  $R_i$  (= the van der Waals radius + the probe radius)
3. FOR EACH atom  $j \neq i$ 
  - (a) calculate its atomic radius  $R_j$  (= the van der Waals radius + the probe radius)
  - (b) calculate the distance  $r_{ij}$  between atoms  $i$  and  $j$
  - (c) IF  $r_{ij} < R_i + R_j$  THEN  $N_i^{int} = N_i^{int} + 1$
4. FOR EACH atom  $i$

IF  $N_i^{int} < level$  THEN  $i$  is a surface atom

ELSE  $i$  is an internal atom

From the above mentioned pseudocode of algorithm it is obvious that the time complexity of the NIN approach is in  $\Theta(N^2)$ , where  $N$  is the number of atoms in the molecule.

The major advantage of this approach is the simplicity of the implementation. On the other hand, distinguishing between internal and surface atoms can be problematic, consequently this approach does not provide as exact results as we would like to. The whole approach is based on identifying neighbours and investigating mutual intersections, but this piece of information depends on the considered atomic radii. The radius used in this algorithm must also include the

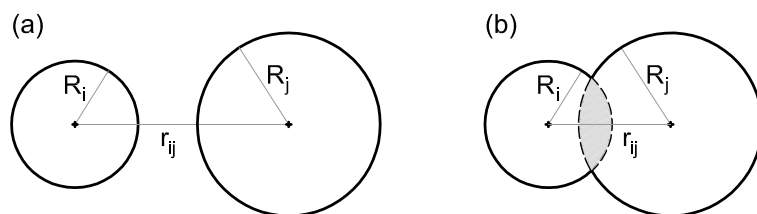


Figure 3.1: (a) Spheres  $i$  and  $j$  are nonintersecting; (b) Spheres  $i$  and  $j$  are intersecting

probe radius. Depending on the curvature of the molecular surface, most of the surface atoms are intersected in such parts of their molecular surface which were created by increasing the van der Waals radius of the probe radius. In case that the probe radius was 0, the explored atom would not be intersected and thus it would be correctly marked as a surface atom. This fact affects the results of this approach very much.

Another uneasy thing is to determine the *level*, because if we want to find it, we need another algorithm for a comparison.

## 3.2 UCSF approach

University of California at San Francisco (UCSF) approach [11] is named after the university, where it was developed by Bash et al. [12]. This method was developed because of the need to reduce CPU-time necessary for computer-graphic displays of macromolecules, but the principle is also very useful for identifying of surface atoms.

The approach is based on a manipulation with a molecule placed in a grid, that is why it is sometimes also called a Grid approach.

In this approach both the atom and the molecule have the same representations as mentioned in the previous algorithm.

First of all, the three-dimensional grid is prepared for the particular molecule, which means that the lowest and the greatest X, Y and Z coordinates of all the atoms in the molecule are found. These values determine the total range of the grid.

Then the molecule is put into the grid. Bash et al. used spacing between the grid points 1.60 Å in their work [12], and so I use the same value in my thesis. This value is not a fixed figure and can be adapted for a particular objective.

The size of the grid is calculated individually for the particular molecule according to the coordinates of its atoms. Due to the fact that each molecule has a finite number of atoms, the size of molecule is finite, and as a result the molecule can be placed into the finite grid with a finite number of grid points.

After placing the molecule into the grid, each atom becomes a component of one cube in the

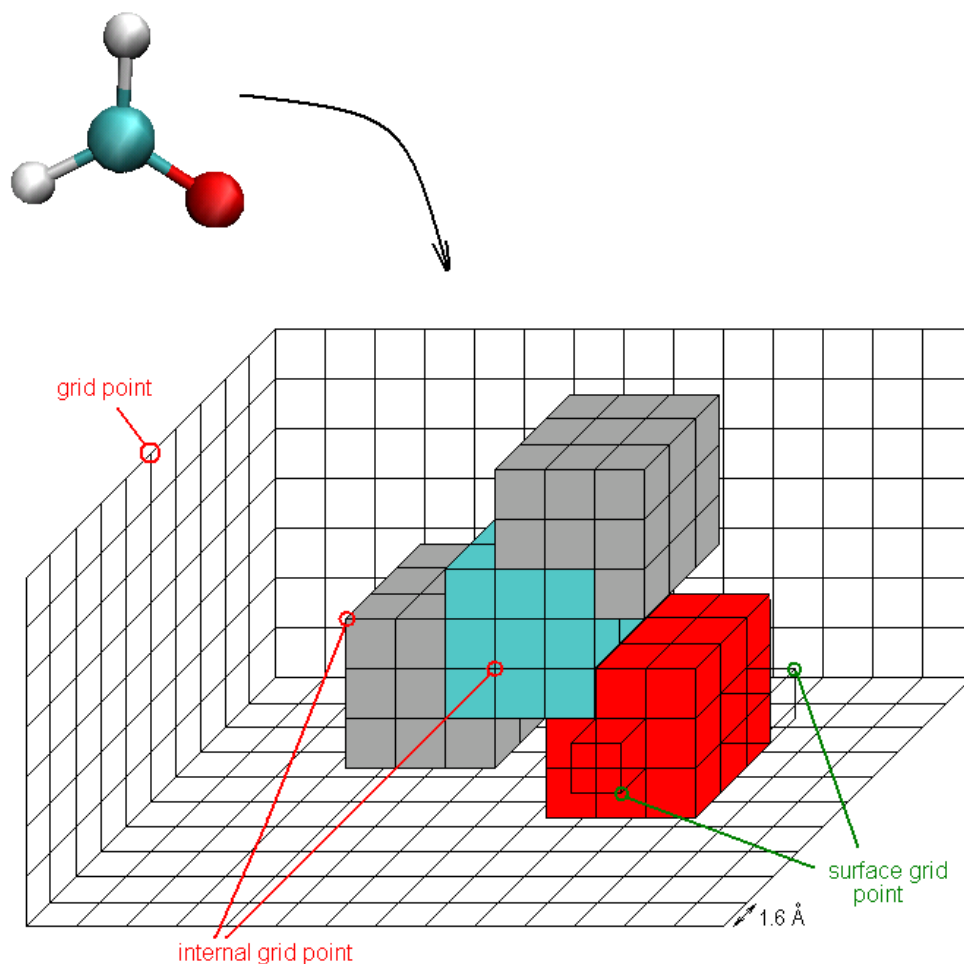


Figure 3.2: Principle of UCSF approach

grid. If the atom lies directly on the edge between the cubes, it becomes a part of the cube with smaller Cartesian coordinates. But the probability of such a case is very small.

In the next steps we will use the grid to find out whether the particular atom is surrounded by neighbouring atoms or not.

Each atom in the grid is represented by 27 cubes ( $3 \times 3 \times 3$  cubes). Specifically, the atom is considered to be localized in the central cube and all the 26 cubes around are used for calculation of heuristics which describe the positions of neighbouring atoms around the explored atom.

In my thesis I use heuristics which obtains information about the number of atoms in the neighbourhood of the explored atom by increasing auxiliary variables. Each cube in the grid has its auxiliary variable **cube score**. The value of the cube score at the beginning equals zero and its increasing follows a strictly defined rule: If the cube contains some atom, its cube score and cube scores of all twenty-six cubes in its three-dimensional neighbourhood are increased.

After increasing all the cube scores of grid cubes, we need to calculate a score for each grid

point (the point in the corner of the cube). The **grid point score** is defined as a sum of the cube scores of all the adjacent grid cubes<sup>2</sup>.

The next step divides the grid points into two parts according to the established grid point score: If score = 0 then the grid point is marked as a **surface grid point**, otherwise the grid point is marked as an **internal grid point**.

Finally, we determine the surface and internal atoms. To achieve this division we use results obtained from the previous steps. 64 grid points (4 layers contain 4 lines with 4 grid points in each, which makes 64 grid points together) belong to one atom. If we want to determine whether the investigated atom is a surface atom or not, we must examine all neighbouring grid points of these 64 grid points. In total, we must inspect 216 grid points (6 layers, where each of them contains 6 lines with 6 grid points in each line). An atom is marked as a **surface atom**, if at least one of those 216 grid points in its neighbourhood is marked as the surface grid point, otherwise the atom is marked as an **internal atom**.

All principles of the UCSF approach are described in the following pseudocode of algorithm:

*Pseudocode of algorithm*

1. put the molecule into the prepared grid (fitting the molecule)
2. FOR EACH atom  $a$   
    store the coordinates of the cube which contains this atom
3. FOR EACH atom  $a$   
    increase the cube scores of all 27 grid cubes which represent the atom
4. FOR EACH atom  $a$   
    FOR EACH grid point  $g$  relevant to  $a$   
        calculate the grid point score of  $g$ , which is defined as a sum of cube scores of all adjacent grid cubes.
5. FOR EACH atom  $a$   
    FOR EACH grid point  $g$  relevant to  $a$   
        determine whether  $g$  is a surface point or not:  
        IF *grid point score of  $g$*  = 0 THEN  $g$  is marked as a surface grid point  
            ELSE  $g$  is marked as an internal grid point
6. FOR EACH atom  $a$   
    decide whether it is a surface atom or not:  
    IF at least one of 216 grid points in neighbourhood of  $a$  is marked as a surface grid point THEN  $a$  is a surface atom  
        ELSE  $a$  is an internal atom.

---

<sup>2</sup>Grid point  $g$  has 8 adjacent grid cubes, all of which have  $g$  as a corner point.

From the above mentioned description of the algorithm, it follows that the time complexity of the UCSF approach lies in  $\Theta(N)$ , where  $N$  is the number of atoms in the molecule.

Among the advantages of this approach belongs its time complexity, which is lower than in case of the NIN approach. The biggest advantage of this algorithm is the fact that except for the grid spacing, no other parameters are needed to decide whether the atom is on the surface of a molecule or not.

But we must also admit that the results received from this approach are strongly dependent on the grid spacing<sup>3</sup>, which is at the same time the biggest disadvantage. The choice of the grid spacing is very important, because it influences everything in the UCSF approach. This choice is discussed in [11], where after series of experiments following observation was made: For grid spacings less than 1.60 Å the data were even more unsatisfactory (the number of atoms marked as surface rapidly increased). Surprisingly for grid spacings greater than 1.60 Å the results were also unsatisfactory, for example some of the atoms previously correctly classified were afterwards classified in a wrong way.

Among the disadvantages also belongs the fact, that the UCSF approach uses constant grid spacing without respect to the examined atoms, although each atom has its own van der Waals radius. Another disadvantage of the UCSF approach is its tendency to mark quite a big number of atoms as surface atoms.

In spite of its disadvantages, this algorithm is good and provides better results than the NIN approach.

### 3.3 SAS approach

Solvent-accessible surface (SAS) approach [11] is a typical representative of the second group of algorithms – the sophisticated ones, which are strictly based on the mathematical foundations.

There are many methods which have been proposed for the calculation of the SAS. The original method was introduced by Lee & Richards [13, 14] in 1971 and it is based on slicing the molecule. The principle of this approach is following: A set of collinear planes with a fixed spacing divides the molecule into sections. For each section the solvent molecule is rolled along the van der Waals surface. The solvent accessible area for an atom is then calculated from the length of the arc drawn on a particular section. The solvent accessible surface of the molecule is calculated as a sum of solvent accessible surfaces of individual atoms. More details can be found in [13].

In 1973 Shrake & Rupley [15] developed a new method, in which a finite set of dots is regularly distributed on the surface of the atoms and then it is counted how many points are within the radius of other atoms.

Later new analytical methods were discovered, which use different approaches. One of them is based on the Voronoi procedure. This approach calculates SAS from the intersection of Voronoi polyhedra with spheres surrounding each atom (details are described in [16]).

Other approaches which are commonly used for determining the solvent accessible surface

---

<sup>3</sup>The grid spacing is the distance between the grid points.

are the  $\alpha$ -shapes [17], a generalization of the convex hull. The method for computing the  $\alpha$ -shape of a three-dimensional point set  $S$  is based on triangulation, which is decomposition of  $S$  into non-intersecting tetrahedra.

The number of different approaches for calculating the solvent-accessible surface is quite high and still new approaches appear. One of the reasons are the advantages of a solvent-accessible surface representation, which are described in section 2.1.2. In favour of this approach also spoke its low time complexity, which is in  $\Theta(N \cdot \log N)$ .

I did not implement this approach in my thesis, because there are many software tools, which implement this method effectively, so I rather chose one of them (see section 5.4).

# Chapter 4

## Proteins

Proteins [3, 18] are considered to be the fundamental building blocks of all living beings and have a wide variety of functions. Some of them are building blocks of tissues, organs, hair, nails and so on, other catalyze biochemical processes (enzymes), other coordinate chemical processes in organisms (hormones) or have other different functions. But all of them have their important roles in living organisms and that is why they belong to the most actively-studied molecules in biochemistry.

From the chemical point of view a protein is a macromolecule, which consists of more than 100 amino acids joined by the peptide bonds.

### 4.1 Amino acids

Amino acid [19] is a molecule, which contains an **amino group** ( $-NH_2$ ), a **carboxyl group** ( $-COOH$ ) and a **R group** (which is a specific functional group).

All of these groups are attached to an alpha carbon atom<sup>1</sup> as illustrated in Figure 4.1. The R group determines the particular amino acid and all its unique properties, which differ with every single amino acid. In a protein, the R group is also called a **side chain**.

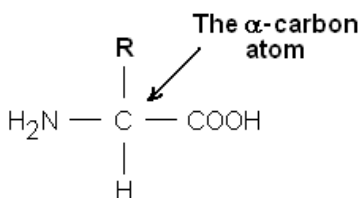


Figure 4.1: Structure of amino acids

Despite the fact, that there are over 30 amino acids occurring in nature [20], all proteins are synthesized from only 20 different amino acids (see Appendix A). These amino acids are so

---

<sup>1</sup>The alpha carbon atom is the closest carbon atom to the carboxyl group of amino acid.

important and so often used in biochemistry that for each of these amino acids the three-letter and one-letter abbreviations are given.

### 4.1.1 Peptide bond

A **peptide bond** is the linkage between two amino acids, formed by the condensation reaction as shown in Figure 4.2.

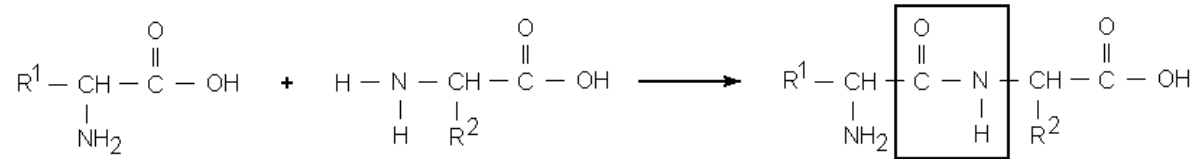


Figure 4.2: Peptide bond

## 4.2 Structure of proteins

All important attributes of proteins are derived from their structure [18] which can be divided into four levels:

- A **primary structure** (see Figure 4.3) of a protein refers to its linear sequence of amino acids and the location of any disulfide ( $-\text{S}-\text{S}-$ ) bridges. The amino acid which is bound in a protein (it means that this amino acid has  $-\text{NH}-$  and  $-\text{CO}-$  groups instead of  $-\text{NH}_2$  and  $-\text{COOH}$ ) is also called a **residue**.

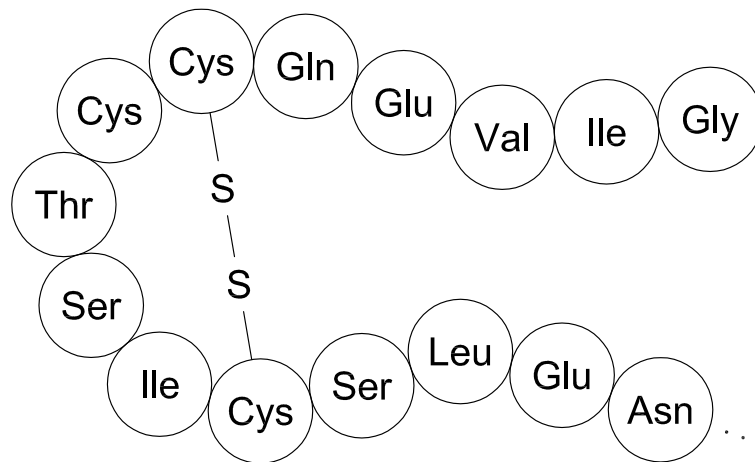


Figure 4.3: Primary structure of a hypothetical protein

- A **secondary structure** occurs when the sequence of amino acids is linked by hydrogen bonds. There are two main types of secondary structures:  $\alpha$ -helix and  $\beta$ -sheet (or pleated sheet).

An  $\alpha$ -**helix** is a helix which is formed out of the protein chain. This chain makes up the central structure and the side chains extend out and away from the helix. An idealized shape of  $\alpha$ -**helix** is illustrated in Figure 4.4 a.

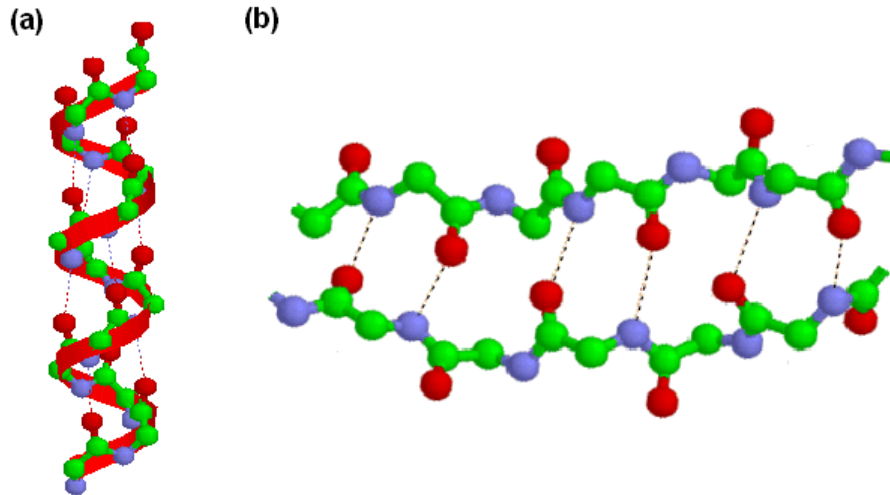


Figure 4.4: An ideal shape of (a)  $\alpha$ -helix and (b)  $\beta$ -sheet<sup>2</sup>

Another type of secondary structure is a  $\beta$ -**sheet**, which consists of a pair of chains lying side-by-side. The chains are stabilized by the hydrogen bonds between the carboxyl oxygen atom on one chain and the  $-NH$  group on the adjacent chain.

An ideal shape of a  $\beta$ -sheet is shown in Figure 4.4 b.

The parts of a protein chain which do not show any easily categorized secondary structure components are referred to as a **random coil**.

- A tertiary structure is the full three-dimensional (3D) folded structure of the entire protein chain. It is described by the Cartesian co-ordinates of all atoms in a molecule. The tertiary structure is very important, because the whole function of a protein depends on it. If the tertiary structure is broken, the protein loses its biological function.

The main experimental techniques for determining the 3D structures of macromolecules are the X-ray crystallography and the nuclear magnetic resonance (NMR). Very important site of tertiary protein structures is the Protein Data Bank [5].

---

<sup>2</sup>In order to achieve transparency, Figure 4.4 contains only backbone atoms and excludes hydrogens. However, it can be done without any contradiction to universality, because only the backbone atoms are involved in the secondary structure (not the side chains).



Figure 4.5: Tertiary structure of 1GWR

- If a protein molecule consists of more than one protein unit then this molecule has also the **quaternary structure**, which specifies the relative positions among the subunits in a protein.

A typical representative of a protein which has a quaternary structure is hemoglobin (see Figure 4.6), a protein with four protein units – two  $\alpha$ -globins and two  $\beta$ -globins.

### 4.3 Protein motifs

The term **motif** [21], which means a characteristic sequence or structure, is used in two different ways in computational chemistry:

The first kind of motifs is the **structural motif**. This point of view on the motifs is through the protein structure, so the structural motif access refers to a set of continuous secondary structure elements of proteins. An example of a structural motif is the helix-turn-helix motif. These motifs are usually not able to exist separately from the rest of the protein.

The structural motifs sometimes suggest a certain function, but more often they do not. It is caused by the fact that many different amino acid sequences are compatible with the same secondary structure motif. Also the number of structural motifs is so big that even additional aminoacidic chains can be inserted into the motif without disrupting the structure. As a result, the identification of structural motifs is very difficult and despite the fact that bioinformatics and computational chemistry are interested in it, they also search for other ways which can determine the particular function.

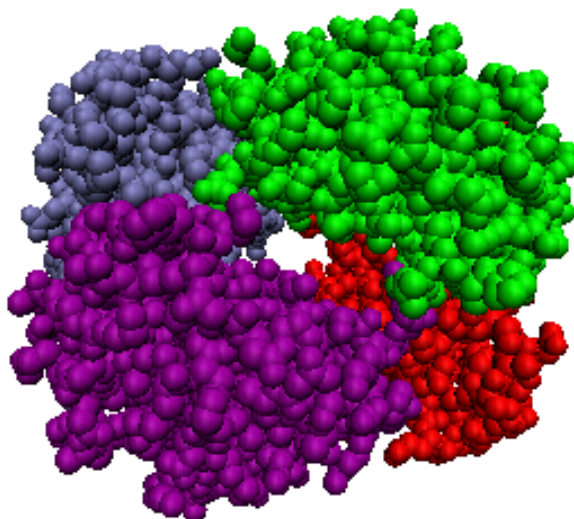


Figure 4.6: Quarternary structure of hemoglobin

Another kind of motifs is the **sequence motif**<sup>3</sup>. This point of view on the motifs is concerned mainly with the primary structure of proteins. It can be very useful for determining concrete biochemical function and that is one of the reasons why this thesis is focused on this approach. Specifically, in spite of the fact that active sites of proteins are various (we can find active sites with the same function, but very different primary structure), some motifs correspond to the particular function. The number of such motifs is relatively small – only several tens – and some of them are described in the ELM database [4] of biologically important motifs.

Provided that we know, that some motif is a biologically important motif, it is favourable to detect its presence on the surface of some other proteins. The geometry of the found motifs can be compared with the geometry of motifs described in the ELM database or with the geometry of this motif in some other molecules (for example drugs).

An example of a sequence motif is the nuclear receptor box (LIG\_NRBOX) motif L--LL to which the nuclear receptors are bound.

---

<sup>3</sup>in the following text will be referred to only as motif

**Part III**  
**Methods**

# Chapter 5

## Methods

### 5.1 Hardware and software

This thesis was elaborated on the operating system UNIX (specifically LINUX Red Hat 9) and MS Windows 2000 and XP.

The implementation of the algorithms was done in the C language. All programs were compiled by the gcc compiler (version 3.4.2), which is distributed under the GNU licence. The script for calculation of the solvent-accessible surface was written in Tcl/Tk language.

For the testing purposes a set of real motifs and protein molecules was created: The motifs were used from the ELM [4] database and the protein molecules from the PDB [5] database.

The molecules were visualized by the programs VMD [26] and Rasmol [29]. Structural formulas of molecules were drawn by the program ACD/ChemSketch [32]. Figures for this thesis were drawn using AutoCAD [33].

The text of this thesis was typeset by using the program L<sup>A</sup>T<sub>E</sub>X.

### 5.2 PDB format

Protein Data Bank [23] (PDB) is a database of large biological molecules, including proteins and nucleic acids. The PDB was established in 1971 at Brookhaven National Laboratory as a repository for resolved crystallographic structures. Since 1998, the PDB is maintained by the Research Collaboratory for Structural Bioinformatics (RCSB).

The Protein Data Bank uses a uniform format (called PDB format [24]) to store a variety of information associated with each structure, for example sequence details, atomic coordinates or partial bond connectivities as derived from crystallographic studies. The PDB files are free of charge through the RCSB PDB portal [5].

PDB file has a strictly defined structure, which has its origin in the period of punched cards. The whole file is divided into lines. Each line determines one PDB record and consists of 80 columns. The first six columns of every line contain a record name and columns 7 to 70 contain

data, columns 71 to 80 are blank or may contain information added by library management programs.

PDB file contains many types of records and the order of all records in the PDB file is strictly defined. According to the type of information, PDB records can be divided into several sections:

- The first section contains general information about a molecule. Here can be found records containing information about primary and secondary structure, heterogens, coordinate transformation operators, data received from crystallographic studies and so on. Such data are presented by many types of records. The most significant are for example these: HEADER, COMPND, AUTHOR, JRNL, SEQRES, HELIX, SHEET or CRYST1.
- The middle part of the PDB file comprises atomic coordinates, which are the most important for our purposes, because they describe positions of each atom in the molecule. This section is the largest section in the whole PDB file, because each atom of the molecule is described by one line beginning with ATOM or HETATM.

The ATOM record contains the atomic coordinates for standard residues and in case of proteins the ATOM records are listed from amino to carboxyl terminus.

The format of ATOM record is following:

Columns	Data type	Field <sup>1</sup>	Definition
1 – 6	Record name	"ATOM "	Record name
7 – 11	Integer	serial	Atom serial number
13 – 16	Atom	name	Atom name
17	Character	altLoc	Alternative location indicator
18 – 20	Residue name	resName	Residue name
22	Character	chainID	Chain identifier
23 – 26	Integer	resSeq	Residue sequence number
27	AChar	iCode	Code for insertion of residues
31 – 38	Real (8.3) <sup>2</sup>	x	X Cartesian coordinate of atom <sup>3</sup>
39 – 46	Real (8.3)	y	Y Cartesian coordinate of atom <sup>3</sup>
47 – 54	Real (8.3)	z	Z Cartesian coordinate of atom <sup>3</sup>
55 – 60	Real (6.2)	occupancy	Occupancy
61 – 66	Real (6.2)	tempFactor	Temperature factor
73 – 76	LString (4)	segID	Segment identifier, left-justified
77 – 78	LString (2)	element	Element symbol, right-justified
79 – 80	LString (2)	charge	Charge on the atom

Table 5.1: Description of the ATOM record format

<sup>1</sup>name of the field

<sup>2</sup>it means real number with 8 digits together, 3 are after decimal point

<sup>3</sup>in Angström units

Each record is divided into fields by a column number and for each of these fields its name, data type and definition are rigorously defined. Detailed description is given in Table 5.1, which is borrowed from the PDB Format Guide [25].

Figure 5.1 demonstrates an example of ATOM record structure.

	1	2	3	4	5	6	7	8	
	1234567890123456789012345678901234567890123456789012345678901234567890								
ATOM	158	N	PRO	20	-9.525	28.593	34.637	1.00 19.20	N
ATOM	159	CA	PRO	20	-10.337	29.108	35.741	1.00 16.29	C
ATOM	160	C	PRO	20	-9.691	30.307	36.428	1.00 16.88	C
ATOM	161	O	PRO	20	-8.497	30.299	36.732	1.00 17.29	O
ATOM	162	CB	PRO	20	-10.425	27.908	36.690	1.00 13.88	C
ATOM	163	CG	PRO	20	-10.204	26.722	35.800	1.00 13.98	C
ATOM	164	CD	PRO	20	-9.089	27.213	34.927	1.00 19.44	C

Figure 5.1: Example of PDB ATOM record format

The HETATM record contains atomic coordinate records for heterogens. HETATM records are formatted in the same way as ATOM records.

In this section the TER record also should be mentioned, because it indicates a chain terminator.

- The last section of the PDB file contains information about chemical connectivity (the CONECT record), summary information and end-of-file marker END.

In my thesis I suppose that one PDB file describes only one structure, so I use only three main types of records: 'ATOM', 'HETATM' and 'TER'. That is why I focused on these entries in the above mentioned description. More details are described in [24].

## 5.3 ELM

The Eukaryotic Linear Motif (ELM) database [31], which was established by the ELM consortium, is a resource for investigating short functional motifs in eukaryotic proteins.

This server provides more than 80 motif descriptions and access to basic annotation. During the elaboration of this thesis the ELM server was used for choosing some suitable discontinuous motifs.

The ELM database is available through the Internet (see [4]).

## 5.4 VMD

The VMD [26] is a software developed for an interactive graphical display and analysis of molecular systems. It is designed especially for biomolecules like proteins and nucleic acids. Among its biggest strengths we can find the ability to visualize data obtained by the molecular dynamics

and the ability to calculate some properties of molecules (for example SAS). Another advantage is the fact that it provides a wide variety of different rendering methods and coloring schemes.

The VMD is implemented in C++ and requires either the Silicon Graphics GL library or the OpenGL library.

The input file for this software is the molecular coordinate file, which contains the positions of all the atoms that make up the molecule, for example the PDB file (see section 5.2).

This program offers graphical user interface (GUI) as well as a text console, which provides an interface for the text-oriented user. Such a user can write both embedded VMD commands and his own scripts using the Tcl/Tk library package, which contains a parser for the Tcl language.

The Tcl (Tool Command Language) [27] is an open source language developed by professor John Ousterhout. This language is suitable for many purposes, but primarily for issuing commands to interactive programs. Tk (a Tcl extension) is a toolkit of commands for building window-based Tcl applications. The main advantage of Tcl is its programability, the Tcl language can be easily extended with the additional commands specific to the particular application.

During the elaboration of this thesis, the program VMD was mainly used for calculating the solvent-accessible surface. In the embedded Tcl/Tk language was written an additional script for calculating the solvent-accessible surface for each atom, which is described in section 6.3. Afterwards this program served as a software tool for visualization of the protein molecules.

More information contains [28], where the software is freely available for download.

## 5.5 RasMol

RasMol [29] is a software tool used for displaying molecules. This software was written by Roger A. Sayle at the Bioinformatics Research Unit at the University of Edinburgh. RasMol is primarily devoted to the visualization of structural information from the PDB database, although the PDB format is not the only one which is supported.

The qualities of this tool are almost the same as the qualities of VMD, but there are some features in which both programs differ and for that reason RasMol was used in this thesis too. One of these features is the fact that RasMol enables displaying labels of atoms directly into the figure, so the identification of the atoms is easier than in case of the VMD.

Another advantage of this program is its ability to translate the images into a variety of formats for display or alternation by other graphics programs. Very useful is the EPSF format, which can be easily embedded into the PostScript files.

More information can be found at [30], where the software is freely available for download.

# Part IV

## Implementation

# Chapter 6

## Implementation

### 6.1 Program `sad`

#### 6.1.1 Functionality

The program `sad` (Surface Atom Detector) is intended for identifying surface atoms in the given molecule. For these purposes it contains the implementations of the following algorithms: NIN approach, UCSF approach and SAS approach.

The program works with a protein molecule given as a file in PDB format and also requires that this input file corresponds to the description introduced in [5.2](#).

#### 6.1.2 Programming language

The C language was chosen as the programming language for the program `sad` as it fulfils the following conditions:

- The C language is a universal programming language, which is quite common in computational chemistry.
- It is a sufficiently powerful tool, which enables writing effective and fast programs.
- The C language is easily portable among different operating systems.

#### 6.1.3 Interface

The program `sad` has a command line user interface, which is common in the UNIX-oriented operating systems, so the pieces of information needed for its control are given as parameters.

The advantage of such an approach is the fact, that the program can be used in batch processing, because it needs no other parameters after starting.

Calling of the program `sad` has the following structure:

```
sad -i input_file -o output_file (parameters)
```

Possible parameters of the program `sad` :

- Mandatory parameters:
  - **input file**
    - `-i input_file`   specification of the input file in the PDB format
  - **method of surface atoms searching**
    - (only one of the following options is permitted)
    - `-NIN level`    NIN approach
      - where a level is the maximal number of neighbouring atoms for a surface atom (see section 3.1)
    - `-UCSF`        UCSF approach
    - `-SAS file`     SAS approach
      - where a file is a name of an output file of the program `sasa` (see section 6.3). This file contains precounted data.
  - **output file**
    - `-o output_file`   specification of the output PDB file name
- Optional parameters:
  - **silent mode**
    - `-q`   prohibits writing outputs on the standard output

Help, which displays a list of available parameters with short comments, is called by the command `sad -help`.

An example of calling the program:

```
sad -i 1TTG.pdb -o results.pdb -NIN 48
```

This command causes that the program `sad` identifies the surface atoms by the NIN approach with the *level* set to value 48. This information is written into the output file `results.pdb`

### 6.1.4 Output file

The program `sad` writes its output into a file which is very similar to the file in a PDB format described in section 5.2. The difference between these two files appears only in the ATOM records. Specifically, each ATOM record in the output file contains the same information as the original ATOM record except for its residue name entry, which is changed in this way: If the relevant atom was identified as a surface atom, its residue name entry contains abbreviation SUR. Otherwise,

the atom was identified as an internal atom and its residue name entry contains abbreviation INN.

This adaptation of PDB file can be favourably used during the display of a molecule, because most of the visualization programs provide functions which can colourfully distinguish different residua.

An example of the output file is illustrated in Figure 6.1.

```

          1         2         3         4         5         6         7         8
1234567890123456789012345678901234567890123456789012345678901234567890
ATOM    510  N   SUR  A 375      1.206  16.990  13.300  1.00 40.54      N
ATOM    511  CA  SUR  A 375      0.888  15.935  14.257  1.00 37.65      C
ATOM    512  C   INN  A 375     -0.331  15.149  13.786  1.00 34.64      C
ATOM    513  O   INN  A 375     -0.162  13.935  13.653  1.00 33.10      O
ATOM    514  CB  INN  A 375      0.674  16.487  15.660  1.00 38.36      C
ATOM    515  CG  INN  A 375      1.947  17.070  16.231  1.00 38.82      C
ATOM    516  CD  SUR  A 375      1.813  17.502  17.675  1.00 40.01      C
ATOM    517  OE1 INN  A 375      1.438  16.720  18.534  1.00 40.40      O
ATOM    518  NE2 SUR  A 375      2.132  18.746  17.993  1.00 39.69      N

```

Figure 6.1: Example of a part of the output file

## 6.2 Program **samie**

### 6.2.1 Functionality

The program `samie` (Surface Atoms and Motifs IdentifiEr) is intended for identifying all instances of a given motif in the given molecule. For each instance of a motif the program also determines whether the instance lies on the surface of the given molecule or not.

The whole program can be divided into two logical parts: The first part of this program has the same functionality as the above mentioned program `sad`. Therefore, after processing of this part the information about the surface and internal atoms in the molecule is known.

In the second part of this program all pieces of information received in the first part of the program are used for determination, if the input motif is on the surface of the molecule or not.

The program works with a protein molecule given as a file in the PDB format and also supposes that this input file corresponds to the description introduced in 5.2.

### 6.2.2 Programming language

The program `samie` was implemented in the C programming language (for the same reasons as the program `sad`).

### 6.2.3 Interface

The program `samie` has a command line user interface (for the same reasons as the above described program `sad`).

Calling of the program `samie` has the following structure:

```
samie -i input_file -o output_file (parameters)
```

Possible parameters of `samie` are:

- Mandatory parameters:
  - **input file**<sup>1</sup>
  - **method of surface atoms searching**<sup>1</sup>
  - **output file**<sup>1</sup>
  - **searched motif**
    - m motif    specification of the searched motif
- Optional parameters:
  - **type of motif detection**
    - f    default option, which determines that during the identification of surface motifs the full amino acid chains are used
    - sc    during identification of surface motifs only the side chains are used
  - **silent mode**<sup>1</sup>

Help, which displays a list of available parameters with short comments, is called by the command `samie -help`.

An example of calling the program:

```
samie -i 1TTG.pdb -o results.pdb -SAS sas.txt -m L--LL
```

This command causes that the program `samie` first of all identifies the surface atoms by the SAS approach from the data described in `sas.txt` and subsequently determines the location of all instances of `L--LL` motif. Then, for all instances of this motif the program decides, whether the instance is on the surface of the molecule (or not) and writes this decision to the output file `results.pdb`

### 6.2.4 Input motif

The input motif is given as a parameter and its definition is following:

The motif is given as a string which is composed from one-letter amino acid abbreviations (see Appendix A) and a special character ‘-’. This character represents the empty position which may contain any amino acid abbreviation.

---

<sup>1</sup>specification is the same as in case of program `sad` (see subsection 6.1.3)

## 6.2.5 Output file

The specification of the output file is similar to the the output file of the program `sad` (see subsection 6.1.4). The only difference between these two output files is the definition of the residue name entry for each `ATOM` record. In this case it contains three types of abbreviations: `SUR`, `INN` and `UNK`.

If the relevant atom is not a part of any instance of the given motif, its residue name entry contains abbreviation `UNK`. Otherwise, the residue name entry contains either `SUR` or `INN` abbreviation.

Let us consider that the relevant atom is a part of some instance of the searched motif. If this instance of the motif is analysed to be on the surface of a molecule, the considered atom (and all atoms which participate in the motif) has its residue name entry filled with abbreviation `SUR`. Otherwise, the motif is analysed to be in the internal part of the molecule and the considered atom (and all atoms which participate in the motif) has in its residue name entry abbreviation `INN`.

This adaptation of PDB file was selected for the same reasons as described in subsection 6.1.4.

## 6.3 Script `sasa`

### 6.3.1 Functionality

The `sasa` is a script for the program `VMD`, based on the embedded `Tcl/Tk` language. It is an extension of the standard `VMD` command `measure_sasa` which is able to calculate the solvent-accessible surface. Specifically, the script calculates the solvent-accessible surface for all atoms of the molecule. Afterwards the program writes the results into the file `sas.txt`.

### 6.3.2 Programming language

The `Tcl` language was chosen as the programming language for the script `sasa` as it fulfils the following conditions:

- `Tcl` is a universal scripting language suitable for embedded development.
- It is a standard part of the program `VMD`.

### 6.3.3 Interface

The script is called through the command-line of the program `VMD` by the following sequence of commands:

```
source C:/sasa.tcl
measureSAS
```

The meaning of these commands is following:

- command *source* specifies the path to the script
- command *measureSAS* calls the main function of the script and starts the calculation of the solvent-accessible surface for each atom of the molecule which is actually loaded in the program VMD.

The output file *sas.txt* contains as many lines as the molecule has atoms. Each line includes two pieces of information: the number of an atom and its solvent-accessible surface.

## Part V

# Results and discussion

# Chapter 7

## Testing

The testing part of this thesis can be divided into two sections. The first of them was aimed at the surface atoms. The target was to detect a number of identified surface atoms for each approach and compare the obtained results. The second field of the testing was devoted to the surface motifs. The objective was to determine a number of identified instances of the given motif and to decide for each instance whether it is on the surface of the molecule or not. Finally, the obtained results were discussed.

### 7.1 Testing molecules

During the testing part of this thesis and according to the purposes of the particular tests, a set of suitable testing molecules was created (see Table 7.1).

PDB ID	number of atoms	motif	
		name	amino acids <sup>1</sup>
1FNA	675	LIG_RGD	RGD
1TTG	702	LIG_RGD	RGD
1EDU	1289	LIG_AP2alpha.2	DPW
1H0A	1611	LIG_AP2alpha.2	DPW
1T63	2307	LIG_NRBOX	L--LL
1GWR	3987	LIG_NRBOX	L--LL
1GWQ	4192	LIG_NRBOX	L--LL
1JSU	5183	LIG_CYCLIN_1	R-L
1AXC	7038	LIG_CYCLIN_1	R-L
1H25	9037	LIG_CYCLIN_1	K-L

Table 7.1: Set of testing molecules

---

<sup>1</sup>Sequence of amino acids which determine the particular motif.

This group of 10 different protein molecules was selected from the PDB database very carefully in order to receive a representative sample.

The described set of molecules was used in both testing parts, so the specification of the molecules described in Table 7.1 contains the pieces of information, which was used during the tests. For each molecule is given its PDB ID (identification code from the PDB database), a total number of atoms and the ELM motif which the molecule contains.

## 7.2 Surface atoms

This section, which is devoted to the testing of surface atoms, is divided into the subsections according to the used approach. In each subsection all the molecules from the set of testing molecules were analysed and the results were summed up.

### 7.2.1 NIN approach

The NIN approach is based on an intuitive notion that the number of intersecting neighbours is greater for the internal atoms than in case of the surface atoms. So the first thing which must be observed is the boundary value **level**, which divides the atoms in the molecule to internal and surface atoms.

The rough boundaries of the level were determined from the proportion of identified surface atoms to the total number of atoms, because it is obvious that the number of the surface atoms will be greater than the number of the internal atoms.

PDB ID	number of atoms	level 46		level 47		level 48		level 49		level 50	
		sur <sup>2</sup>	% <sup>3</sup>	sur	%	sur	%	sur	%	sur	%
1FNA	675	520	77.1	538	79.8	557	82.6	575	85.2	596	88.3
1TTG	702	622	88.7	636	90.6	652	92.9	666	94.9	677	96.5
1EDU	1289	785	60.9	831	64.5	871	67.6	921	71.5	958	74.4
1H0A	1611	882	54.8	931	57.8	993	61.7	1050	65.2	1102	68.5
1T63	2307	1317	57.1	1411	61.2	1518	65.8	1628	70.6	1723	74.7
1GWR	3987	2336	58.6	2518	63.2	2710	68.0	2909	73.0	3083	77.4
1GWQ	4192	2217	52.9	2409	57.5	2607	62.2	2797	66.8	2973	71.0
1JSU	5183	2671	51.6	2837	54.8	3049	58.9	3266	63.1	3479	67.2
1AXC	7038	3717	52.9	3994	56.8	4252	60.5	4512	64.2	4767	67.8
1H25	9037	5185	57.4	5532	61.3	5901	65.3	6269	69.4	6605	73.1

Table 7.2: Results of the **NIN approach** depending on the choice of the level

<sup>2</sup>Number of atoms which were classified as the surface atoms.

<sup>3</sup>Proportion of the surface atoms to all atoms in the molecule.

From the results presented in Table 7.2 and the visualization of these results, number 48 was selected as the most suitable value of the level. According to the performed observations, this figure provides the most exact results. The outcomes of the NIN approach with the level set to 48 were studied in detail (see Table 7.3) and discussed. The figures of 1FNA and 1TTG (see Table 7.4) are presented to provide the visual assessment of the NIN approach results.

PDB ID	number of atoms	NIN approach		
		sur	%	CPU-time [s]
1FNA	675	557	82.6	0.25
1TTG	702	652	92.9	0.27
1EDU	1289	871	67.6	0.83
1H0A	1611	993	61.7	1.30
1T63	2307	1518	65.8	2.71
1GWR	3987	2710	68.0	8.04
1GWQ	4192	2607	62.2	8.07
1JSU	5183	3049	58.9	13.53
1AXC	7038	4252	60.5	24.63
1H25	9037	5901	65.3	40.68

Table 7.3: Results of the **NIN approach** for the level set to 48

From the results in Table 7.2, it can be seen that the proportion of the surface atoms to the total number of atoms grows linearly with the increasing value of the NIN level. So it is obvious, that this property causes that the level provides too rough estimate for division of the atoms into the surface and internal atoms, which is one of the biggest disadvantages of this approach.

The algorithm works quite well, but sometimes an incorrect classification of the atoms appears: An internal atom can be incorrectly classified as a surface atom, because of the low atom density in its neighbourhood, which can be caused for example by a cavity. On the contrary, a surface atom can be incorrectly classified as an internal atom, because of the heighten atom density in its neighbourhood.

In general, this algorithm is quite fast and provides sufficiently accurate results, which corresponds to the description in the literature.

## 7.2.2 UCSF approach

This approach provides another method for identifying the surface atoms and it is based on more sophisticated theoretical foundations than the above discussed approach. Unlike the NIN approach, there is no level to be set. The molecule is put into the grid and then the strictly defined rules are followed. Thus there is no parameter whose setting could be explored.

The results obtained from this approach are presented in Table 7.5. Some visualized molecules are shown in Table 7.6.

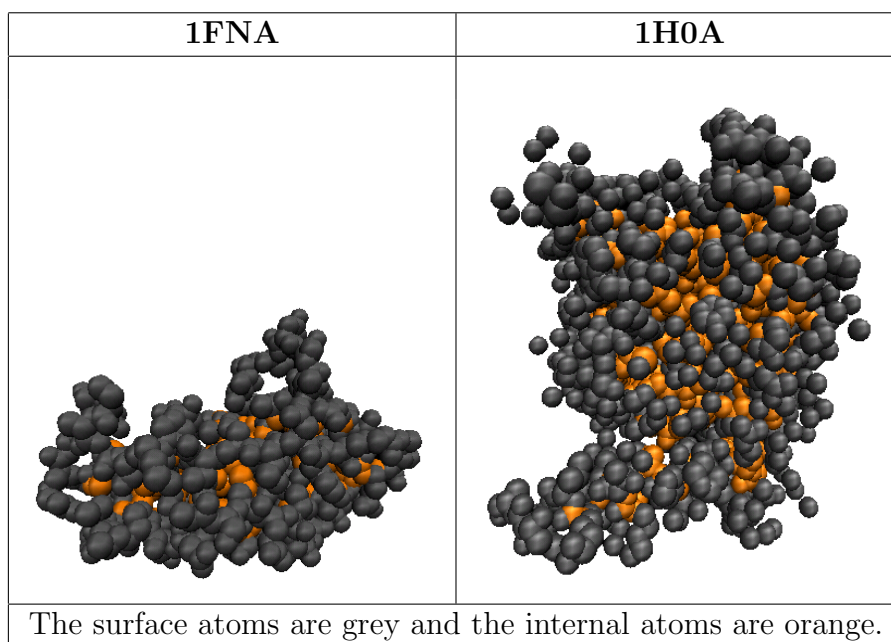


Table 7.4: 1FNA and 1H0A classified by the **NIN approach** with the level set to 48

The advantage of this approach is the fact, that there is no parameter which could influence the results. So this method could be used as a comparative approach and can help to set the optional parametres of the other methods. But it should be pointed out, that this approach has also some disadvantages. The biggest disadvantage of the algorithm is the fact, that the atoms are not always identified correctly, which is caused by the spacing between the grid points.

Despite using the recommended value of the spacing between the grid points, it can be observed (for example from the figures shown in Table 7.6) that this algorithm tends to classify a great number of atoms as the surface atoms. This is caused by the fact that each atom in the molecule is represented by a large three-dimensional figure.

In spite of the disadvantages, this algorithms is quite fast and provides sufficient results, which is in accordance with the conclusions described in the literature.

### 7.2.3 SAS approach

The SAS approach is the last algorithm for identifying the surface atoms, which is studied within this thesis.

Specifically, the SAS approach was not implemented, because there are many software tools, which implement this method effectively. I chose the VMD program, which was adapted by the script `sasa` for this purpose.

The outcomes received from the VMD program were further processed into the PDB file suitable for the visualization, so the exact determining of CPU-time is in this case very difficult and it could be inaccurate. For this reason it was not surveyed. The results obtained from this approach are presented in Table 7.7. Some visualized molecules are shown in Table 7.8.

PDB ID	number of atoms	UCSF approach		
		sur	%	CPU-time [s]
1FNA	675	575	85.2	0.001
1TTG	702	649	92.5	0.002
1EDU	1289	927	72.0	0.003
1H0A	1611	1040	64.6	0.005
1T63	2307	1493	64.8	0.007
1GWR	3987	2681	67.3	0.011
1GWQ	4192	2618	62.5	0.011
1JSU	5183	3166	61.1	0.014
1AXC	7038	4699	66.8	0.019
1H25	9037	6238	69.1	0.026

Table 7.5: Results of the **UCSF approach**

In case of this approach, it is not strictly defined the value of the level from which the relevant atom is classified as the surface atom. The level actually determines the threshold of the effective surface atoms and for the reasons presented in section 2.2, its value was set to 0.01. This choice of the level prevents from the incorrect classification of an internal atom as a surface atom.

Although the CPU-time is not specified and the time of processing depends on the particular molecule, the estimated time was greater than 10 minutes in case of all the molecules which have more than 3500 atoms.

The literature considers this algorithm for identifying the surface atoms to be a very precise approach. Indeed, the results presented in the above mentioned tables show, that the number of incorrectly classified atoms is very small. For these reasons the SAS approach is the most widely used algorithm for identifying the surface atoms in a molecule.

In spite of the fact that this approach classifies the surface atoms correctly, it is very slow, which is the biggest disadvantage.

### 7.2.4 Comparison of approaches

The algorithms presented within this thesis are very various. Some of them are based on the intuitive notion (the NIN approach), other use the grid (the UCSF approach) or other mathematical foundations (the SAS approach). Although these algorithms are built on various methods, all of them have the same objective: to identify the atoms on the surface of the molecule. Therefore, it could be interesting to compare these methods mutually.

From the results presented in Table 7.9 it is evident, that all three approaches take notes of the size of the molecule. Smaller molecules (such as 1FNA) have greater percentage of atoms classified as the surface atoms than the larger molecules (such as 1H25). This can be easier observed in case of the NIN and the UCSF approach, but also the SAS approach has such

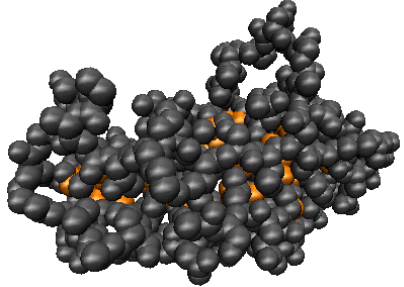
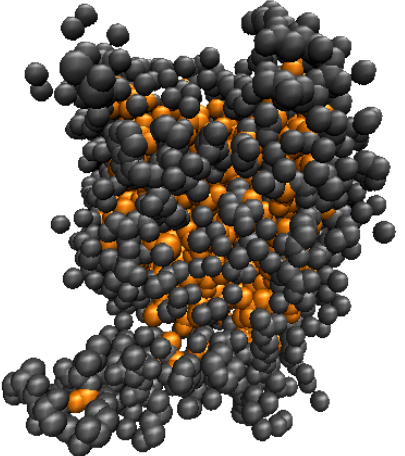
1FNA	1H0A
	
<p>The surface atoms are grey and the internal atoms are orange.</p>	

Table 7.6: 1FNA and 1H0A classified by the **UCSF** approach

features. This observation is in accordance with the general structure of a molecule. It is obvious that the larger molecules could bury inside themselves some of their atoms more easily than the smaller molecules.

In some cases (such as 1FNA), the results of the SAS approach differ from the other approaches very much. I suppose that this is caused by the dissimilarity of the approaches. Undoubtedly, the UCSF approach tends to classify the greatest percentage of atoms as the surface atoms, but not all such classified atoms are really the surface atoms.

The power of the algorithms is illustrated in Table 7.10. This table contains three figures of the same molecule, but each time another approach is used.

All the above presented approaches can be used for identifying the surface atoms, because they provide adequate results. The most precise is the SAS approach, whose results are the most exact. At the same time, this approach is the slowest algorithm of all. The reasonable compromise between the quality of the classification of the surface atoms and the CPU-time is the UCSF approach. This algorithm produces sufficient results and it is undoubtedly the fastest algorithm from all the algorithms presented within this thesis. Finally, the NIN approach can be also used for determining the surface atoms, but it tends to make many mistakes. In spite of this fact, the results received from the NIN approach are still sufficient.

PDB ID	number of atoms	SAS approach	
		sur	%
1FNA	675	476	70.6
1TTG	702	598	85.2
1EDU	1289	800	62.1
1H0A	1611	928	57.7
1T63	2307	1366	59.3
1GWR	3987	2535	63.6
1GWQ	4192	2472	59.0
1JSU	5183	2698	52.1
1AXC	7038	3973	56.5
1H25	9037	5577	61.8

Table 7.7: Results of the **SAS approach** for the level set to 0.01

### 7.3 Surface motifs

The second field of the testing was devoted to the surface motifs. The objective was to determine a number of identified instances of the given motif and to decide for each instance whether it is on the surface of the molecule or not.

For these tests the same set of molecules was used which was applied in the previous testing of surface atoms. The developed program `samie` was used during all tests.

Although this thesis is devoted to the discontinuous motifs, the program `samie` is written universally so it allows to identify also continuous motifs. This feature of the program was applied and the tests were done also for the continuous motifs.

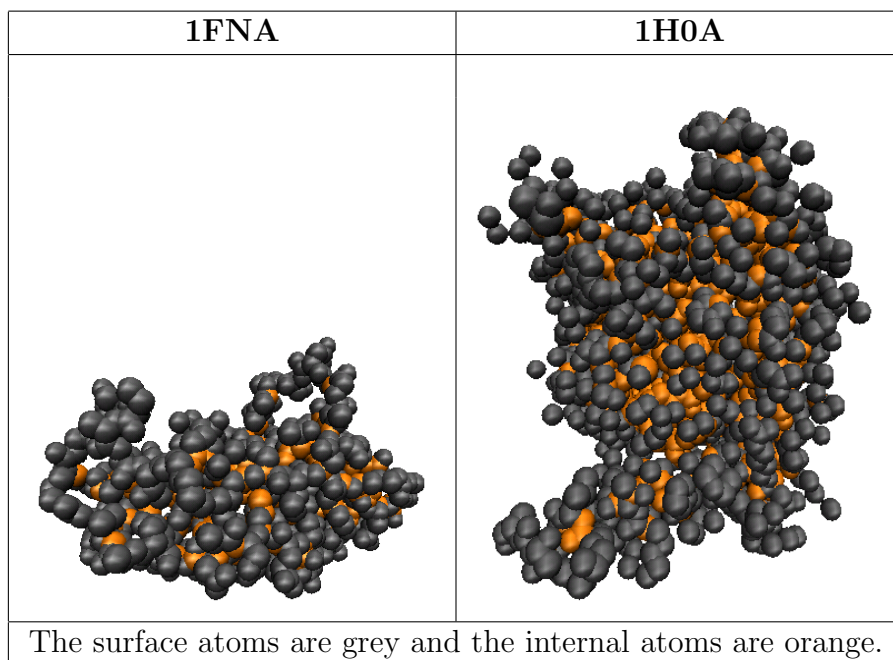
The following observations were found out from the Table 7.11 and Table 7.12: The different method of identifying the surface atoms has an impact on further deciding whether the particular instance of the given motif is on the surface of the molecule or not.

In case of the molecules which contain the continuous motifs (1FNA, 1TTG, 1EDU and 1H0A), the results were not so influenced by the chosen approach as in the case of the molecules with the discontinuous motifs.

The fact that the differences between the individual approaches influence the results of the searching for the motifs is demonstrated in Table 7.13. This table contains three figures of the same molecule, but each time another approach is used.

In the particular case, which is illustrated in Table 7.13, the UCSF approach was the most suitable, because it found almost all surface motifs correctly.

In general features, all three algorithms are able to divide the motifs into the internal motifs and the surface motifs. Thus, they can be favourably used for the first approximation during identifying the motifs. The NIN approach is the least exact method whereas the UCSF and the SAS approaches provide more exact results.

Table 7.8: 1FNA and 1H0A classified by the **SAS approach**

PDB ID	number of atoms	NIN approach		UCSF approach		SAS approach	
		sur	%	sur	%	sur	%
1FNA	675	557	82.6	575	85.2	476	70.6
1TTG	702	652	92.9	649	92.5	598	85.2
1EDU	1289	871	67.6	927	72.0	800	62.1
1H0A	1611	993	61.7	1040	64.6	928	57.7
1T63	2307	1518	65.8	1493	64.8	1366	59.3
1GWR	3987	2710	68.0	2681	67.3	2535	63.6
1GWQ	4192	2607	62.2	2618	62.5	2472	59.0
1JSU	5183	3049	58.9	3166	61.1	2698	52.1
1AXC	7038	4252	60.5	4699	66.8	3973	56.5
1H25	9037	5901	65.3	6238	69.1	5577	61.8

Table 7.9: Comparison of the **NIN approach**, the **UCSF approach** and the **SAS approach**

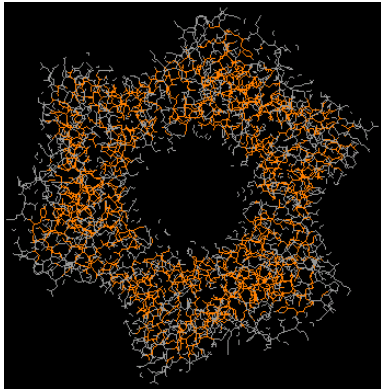
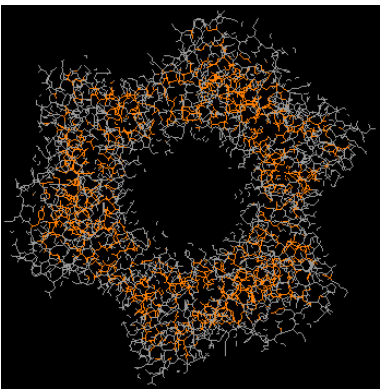
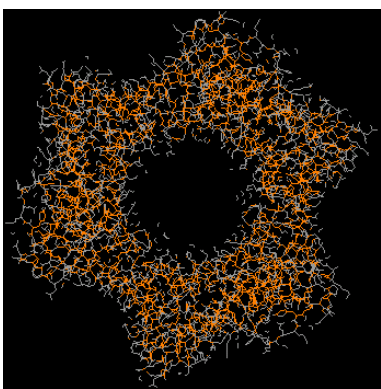
1AXC		
NIN approach	UCSF approach	SAS approach
		
The surface atoms are grey and the internal atoms are orange.		

Table 7.10: Visual comparison of approaches

PDB ID	motif description	position in sequence	NIN approach	UCSF approach	SAS approach
1FNA	RGD	72	surface	surface	surface
1TTG	RGD	77	surface	surface	surface
1EDU	DPW	19	surface	surface	surface
1H0A	DPW	30	surface	surface	surface
1T63	L--LL	190	internal	internal	internal
		254	internal	internal	internal
1GWR	L--LL	192	surface	internal	surface
		224	surface	surface	surface
		433	surface	internal	surface
		465	surface	surface	surface
		482	surface	surface	surface
1GWQ	L--LL	491	surface	surface	surface
		198	internal	internal	surface
		230	surface	surface	surface
		441	internal	internal	surface
		473	surface	internal	surface
1GWQ	L--LL	489	surface	surface	internal
		498	surface	internal	internal

Table 7.11: Results of motif testing – part I

<b>PDB ID</b>	<b>motif description</b>	<b>position in sequence</b>	<b>NIN approach</b>	<b>UCSF approach</b>	<b>SAS approach</b>
1JSU	R-L	109	internal	internal	internal
		113	internal	surface	internal
		187	surface	surface	surface
		204	surface	surface	surface
		247	internal	surface	surface
		352	internal	internal	internal
		549	surface	surface	surface
		562	surface	surface	surface
1AXC	R-L	63	surface	surface	surface
		146	internal	surface	internal
		203	surface	surface	surface
		261	surface	surface	surface
		330	surface	surface	surface
		413	internal	surface	internal
		469	surface	surface	surface
		527	surface	surface	surface
		596	surface	surface	surface
		681	internal	surface	internal
		740	internal	internal	surface
798	surface	surface	surface		
1H25	K-L	56	surface	surface	surface
		65	surface	surface	surface
		419	surface	surface	surface
		468	surface	surface	surface
		608	surface	surface	surface
		617	surface	surface	surface
		956	surface	surface	surface
		1005	surface	surface	surface
		1090	surface	surface	surface
1093	surface	surface	surface		

Table 7.12: Results of motif testing – part II

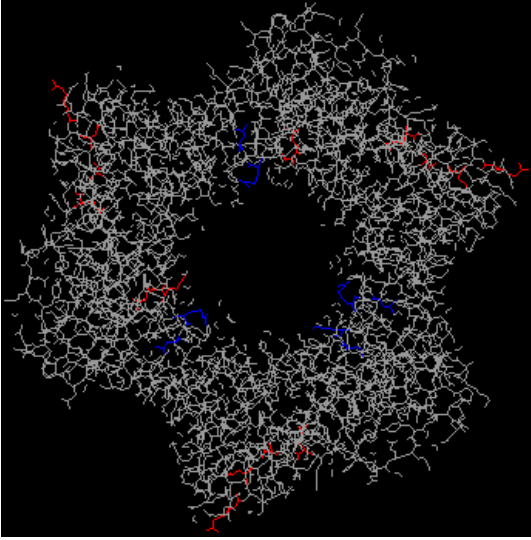
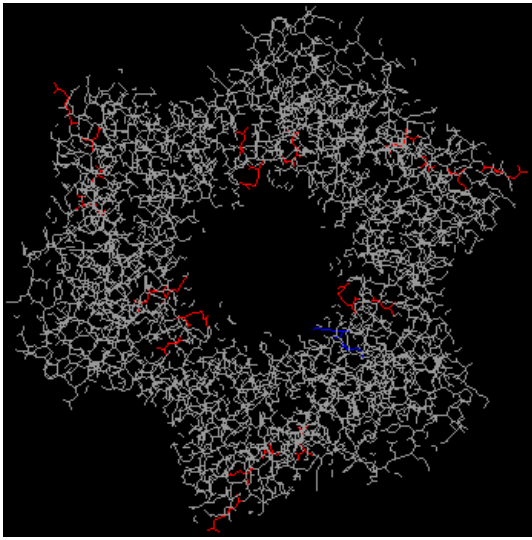
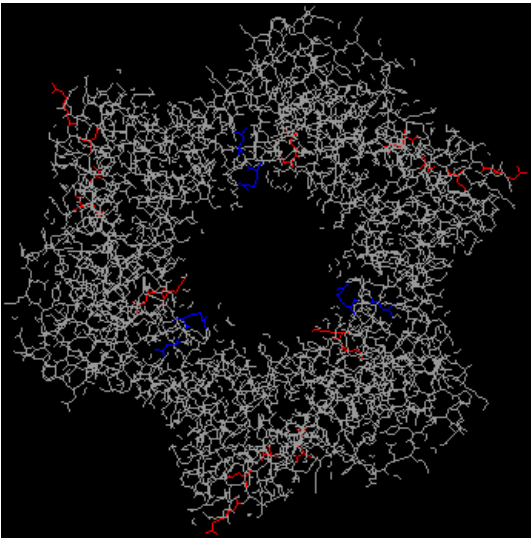
1AXC	
NIN approach	UCSF approach
	
SAS approach	
	
The surface motifs are red, the internal motifs are blue and other atoms in the molecule are grey.	

Table 7.13: Visual comparison of identified motifs

**Part VI**  
**Conclusion**

# Chapter 8

## Conclusion

This thesis is focused on the algorithms searching for discontinuous motifs on surface of protein molecules and can be divided into several parts:

First of all, the foundations of computational chemistry was studied and the basic conceptions about molecular surfaces, surface atoms, proteins and their motifs were learned. Subsequently, the thesis dealt with the algorithms for identification of surface atoms. These algorithms were examined and three suitable ones were chosen and studied in detail: NIN approach (a classical simple approach for identification of surface atoms), UCSF approach (which uses heuristics) and a sophisticated SAS approach. As far as the time complexity is concerned, the UCSF approach belongs to  $\Theta(N)$ , the SAS approach has its time complexity in  $\Theta(N \cdot \log N)$  and the NIN approach is in  $\Theta(N^2)$ . On the basis of these algorithms, an algorithm for searching discontinuous motifs on surface of protein molecules was designed.

The realization of these algorithms required to develop suitable data structures for storing essential information about each atom in the molecule. These pieces of information are available through the PDB format, therefore it was needful to learn it too.

The second part of the thesis was devoted to the implementation of the described algorithms into the programs `samie` (for identification of surface motifs) and `sad` (for identification of surface atoms) and the script `sasa` for the calculation of the solvent-accessible surface. These algorithms were implemented and the created programs were successfully tested.

The last part of this thesis was devoted to the comparison of the implemented algorithms. In this part I used a set of 10 different molecules from the PDB database.

The tests showed that in the general features all three algorithms are able to divide both the atoms and motifs into internal and surface. The NIN approach is the least exact method, whereas the UCSF and the SAS approaches provide more exact results.

The developed programs will be used in a larger bioinformatics project realised in co-operation between University College Dublin and ANF Data Company. This project is focused on determining geometrically similar motifs in molecules of drugs and proteins. Specifically, the identification of the surface protein motifs will help to reduce the amount of studied data.

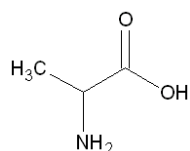
This thesis could be enlarged by another work which could deal with the optimization of

source codes, and possibly with implementation of other approaches used for identification of surface atoms.

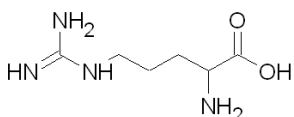
**Part VII**  
**Appendices**

# Appendix A

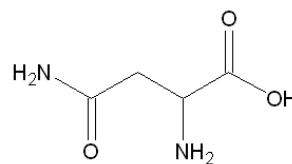
## List of amino acids in proteins



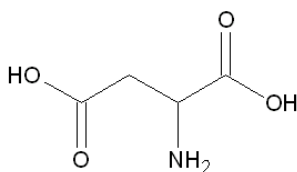
Alanine (Ala, A)



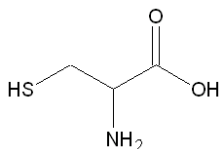
Arginine (Arg, R)



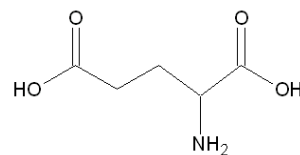
Asparagine (Asn, N)



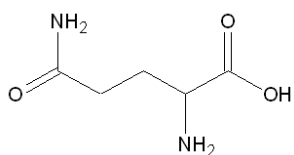
Aspartic acid (Asp, D)



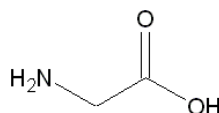
Cysteine (Cys, C)



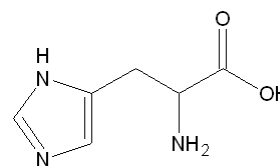
Glutamic acid (Glu, E)



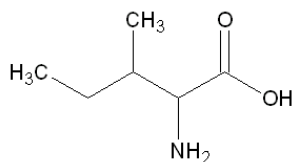
Glutamine (Gln, Q)



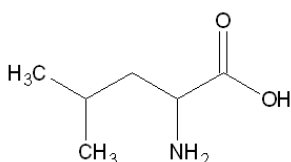
Glycine (Gly, G)



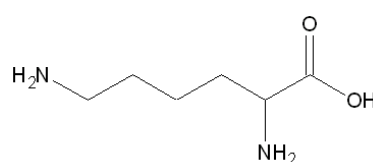
Histidine (His, H)



Isoleucine (Ile, I)



Leucine (Leu, L)

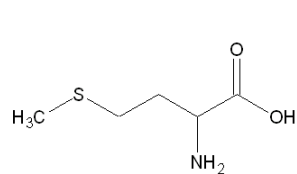


Lysine (Lys, K)

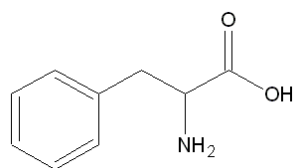
Table A.1: Amino acids and their three- and one-letter abbreviations – part I

APPENDIX A. LIST OF AMINO ACIDS IN PROTEINS

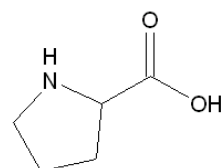
---



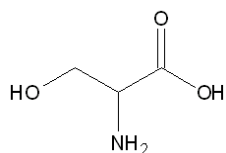
Methionine (Met, M)



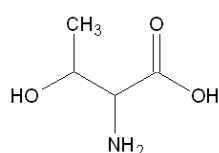
Phenylalanine (Phe, F)



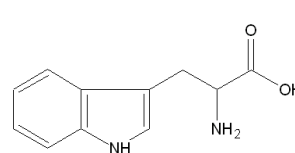
Proline (Pro, P)



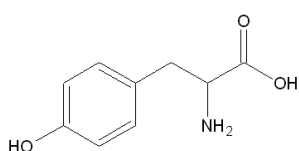
Serine (Ser, S)



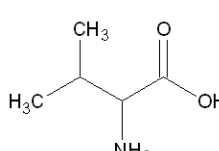
Threonine (Thr, T)



Tryptophan (Trp, W)



Tyrosine (Tyr, Y)



Valine (Val, V)

Table A.2: Amino acids and their three- and one-letter abbreviations – part II

# List of Figures

2.1	Cross-sectional view of the van der Waals surface . . . . .	15
2.2	Cross-sectional view of the solvent accessible surface . . . . .	16
2.3	Cross-sectional view of the contact/re-entrant surface . . . . .	17
3.1	(a) Spheres $i$ and $j$ are nonintersecting; (b) Spheres $i$ and $j$ are intersecting . . . . .	20
3.2	Principle of UCSF approach . . . . .	21
4.1	Structure of amino acids . . . . .	25
4.2	Peptide bond . . . . .	26
4.3	Primary structure of a hypothetical protein . . . . .	26
4.4	An ideal shape of (a) $\alpha$ -helix and (b) $\beta$ -sheet . . . . .	27
4.5	Tertiary structure of 1GWR . . . . .	28
4.6	Quarternary structure of hemoglobin . . . . .	29
5.1	Example of PDB ATOM record format . . . . .	33
6.1	Example of a part of the output file . . . . .	38

# List of Tables

5.1	Description of the <b>ATOM</b> record format . . . . .	32
7.1	Set of testing molecules . . . . .	43
7.2	Results of the <b>NIN approach</b> depending on the choice of the level . . . . .	44
7.3	Results of the <b>NIN approach</b> for the level set to 48 . . . . .	45
7.4	1FNA and 1H0A classified by the <b>NIN approach</b> with the level set to 48 . . . . .	46
7.5	Results of the <b>UCSF approach</b> . . . . .	47
7.6	1FNA and 1H0A classified by the <b>UCSF approach</b> . . . . .	48
7.7	Results of the <b>SAS approach</b> for the level set to 0.01 . . . . .	49
7.8	1FNA and 1H0A classified by the <b>SAS approach</b> . . . . .	50
7.9	Comparison of the <b>NIN approach</b> , the <b>UCSF approach</b> and the <b>SAS approach</b> . . . . .	50
7.10	Visual comparison of approaches . . . . .	51
7.11	Results of motif testing – part I . . . . .	51
7.12	Results of motif testing – part II . . . . .	52
7.13	Visual comparison of identified motifs . . . . .	53
A.1	Amino acids and their three- and one-letter abbreviations – part I . . . . .	58
A.2	Amino acids and their three- and one-letter abbreviations – part II . . . . .	59

# Content of enclosed CD

- Basic information:
  - web page `index.html`, which represents a finger-post with a list of files placed on the enclosed CD
- The source codes of a implemented script and programs:
  - script `sasa`
  - program `sad`
  - program `samie`
- PDB files used for testing
- Files received during the calculations:
  - text files with results of the script `sasa`
  - output files of the programs `sad` and `samie`
- Text of the thesis in the PDF format

# Bibliography

- [1] Jensen F.: *Computational chemistry*. Wiley, 1999
- [2] Leach A.R.: *Molecular modelling*. Longman, 1996
- [3] Wikipedia (<http://www.wikipedia.org>): *Protein*.  
<http://en.wikipedia.org/wiki/Protein> (May 2006)
- [4] The Eukaryotic Linear Motif resource for Functional Sites in Proteins (<http://elm.eu.org/links.html>): *Elementary sites in Proteins*  
<http://elm.eu.org/links.html> (May 2006)
- [5] RCSB (<http://www.rcsb.org>): *Protein Data Bank*.  
<http://www.rcsb.org/> (May 2006)
- [6] Pearlman R. S.: *Molecular Surface Areas and Volumes and Their Use in Structure/Activity Relationships*
- [7] Hermann R. B.: *Theory of hydrophobic bonding II. The correlation of hydrocarbon solubility in water with solvent cavity surface area*, J. Phys. Chem. 76, 1972, 2754-2759
- [8] Richards F. M.: *Areas, volumes, packing, and protein structure*, Ann. Rev. Biophys. Bioeng. 6, 1977, 151-176
- [9] Bondi A.: *van der Waals Volumes and Radii*, J. Phys. Chem. 68, 441-451
- [10] Flower D. R.: *SERF: A program for accessible surface area calculations*, J. Mol. Graphics Mod. 15, 1997, 238-244
- [11] Deanda F., Pearlman R. S.: *A novel approach for identifying the surface atoms of macromolecules*, Journal of Molecular Graphics and Modelling 20, 2002, 415-425
- [12] Bash P. A., Pattabiraman N., Huang C., Ferrin T. E., Langridge R.: *van der Waals surfaces in molecular modeling: implementation with real time computer-graphics*, Science 222, 1983, 1325-1327
- [13] Lee B., Richards F. M.: *Interpretation of protein structures: estimation of static accessibility*, J. Mol. Biol. 55, 1971, 379-400

- [14] Manavalan P., Ponnuswamy P. K.: *Solvent Accessibilities in Glycyl, Alanyl and Seryl Dipeptides*, *Bioinformatics* 18, 2002, 1365-1373
- [15] Shrake A., Rupley J. A.: *Environment and exposure to solvent of protein atoms. Lysozyme and insulin.*, *J. Mol. Biol.* 79, 1973, 351-371
- [16] McConkey B. J., Sobolev V., Edelman M.: *Quantification of protein surfaces, volumes and atom-atom contacts using a constrained Voronoi procedure*, *Bioinformatics* 18, 2002, 1365-1373
- [17] Connexions (<http://cnx.org/>): *Molecular Shapes and Surfaces*.  
<http://cnx.org/content/m11616/latest/#AlphaShapes> (May 2006)
- [18] Voet D., Voet J.: *Biochemistry*. Wiley, 2. edition, 1995
- [19] Vodráška, Zdeněk. *Biochemie* 2. rev. edition, Praha, 1996
- [20] Šípál Z. a kol.: *Biochemie*, Praha, 1992
- [21] Petsko G. A., Ringe D.: *Protein Structure and Function*. New Science Press, 2003
- [22] Schoolscience homepage (<http://www.schoolscience.co.uk>): *Why are proteins important and what are they?*  
<http://www.schoolscience.co.uk/content/5/chemistry/proteins/index.html>  
(May 2006)
- [23] RCSB (<http://www.rcsb.org>): *About the PDB*.  
[http://www.rcsb.org/pdb/static.do?p=general\\_information/about\\_pdb/index.html](http://www.rcsb.org/pdb/static.do?p=general_information/about_pdb/index.html)  
(May 2006)
- [24] RCSB (<http://www.rcsb.org>): *PDB format*.  
[http://www.rcsb.org/pdb/file\\_formats/pdb/pdbguide2.2/guide2.2\\_frame.html](http://www.rcsb.org/pdb/file_formats/pdb/pdbguide2.2/guide2.2_frame.html)  
(May 2006)
- [25] RCSB (<http://www.rcsb.org>): *PDB Format Guide*.  
[http://www.rcsb.org/pdb/docs/format/pdbguide2.2/part\\_62.html](http://www.rcsb.org/pdb/docs/format/pdbguide2.2/part_62.html) (May 2006)
- [26] Humphrey W., Dalke A., Schulten K.: *VMD – Visual Molecular Dynamics*. *J. Mol. Graph.* 14, 1996, 33-38
- [27] Ousterhout J.: *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, 1994
- [28] Theoretical and Computational Biophysics Group (<http://www.ks.uiuc.edu>): *VMD – Visual Molecular Dynamics*.  
<http://www.ks.uiuc.edu/Research/vmd/> (May 2006)
- [29] Sayle R., Milner-White J. E.: *RasMol: Biomolecular graphics for all*. *Trends Biochem Sci (TIBS)* 20(9), 1995, 374-376

- [30] University of Massachusetts Amherst (<http://www.umass.edu>): *RasMol Home Page*.  
<http://www.umass.edu/microbio/rasmol/> (May 2006)
- [31] Puntervoll P., Linding R., Gemünd C., Chabanis-Davidson S., Mattingsdal M., Cameron S., Martin D. M. A., Ausiello G., Brannetti B., Costantini A., Ferre F., Maselli V., Via A., Cesareni G., Diella F., Superti-Furga G., Wyrwicz L., Ramu C., McGuigan C., Gudavalli R., Letunic I., Bork P., Rychlewski L., Kster B., Helmer-Citterich M., Hunter W. N., Aasland R. and Gibson T. J.: *ELM server: a new resource for investigating short functional sites in modular eukaryotic proteins*. *Nucleic Acids Research* 31, 2003, 3625-3630
- [32] Advanced Chemistry Development, Inc.: *ChemSketch 5.12*  
<http://www.acdlabs.com/company.html> (May 2006)
- [33] Autodesk (<http://www.autodesk.com>): *AutoCAD 2007 trial*.  
<http://www.autocad.com> (May 2006)